UNIVERSITY OF MACEDONIA
SCHOOL OF INFORMATION SCIENCES
DEPARTMENT OF APPLIED INFORMATICS

The Applications of Rating Methods Combined with Machine Learning Techniques

A thesis presented for the degree of

Doctor of Philosophy

by

Kyriacos Talattinis

Thessaloniki, November 2023

THE APPLICATIONS OF RATING METHODS COMBINED WITH MACHINE LEARNING TECHNIQUES

Kyriacos Talattinis

BSc Applied Informatics, University of Macedonia, 2010
MSc Applied Informatics, University of Macedonia, 2012

PhD Thesis

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Supervisor

Retired Prof. George Stephanides

Advisory Committee

Retired Prof. George Stephanides	Prof. Alexander Chatzigeorgiou	Prof. Ioannis Refanidis

Examination Committee

1. George Stephanides (Retired Professor, Department of Applied Informatics, University of Macedonia)
2. Alexander Chatzigeorgiou (Professor, Department of Applied Informatics, University of Macedonia)
3. Ioannis Refanidis (Professor, Department of Applied Informatics, University of Macedonia)
4. Nikolaos Samaras (Professor, Department of Applied Informatics, University of Macedonia)
5. Aggelos Sifaleras (Professor, Department of Applied Informatics, University of Macedonia)
6. Maria Satratzemi (Professor, Department of Applied Informatics, University of Macedonia)
7. Efstratios Livanis (Assistant Professor, Department of Accounting and Finance, University of Macedonia)

Date of public defense: November 8th, 2023 at the University of Macedonia, Thessaloniki

Abstract

This dissertation examines the rating systems and their applications in various fields. Most of the systems studied were mainly developed for the sports field. Due to this fact, the main application developed in this thesis is related to the sports field and focuses on the predictions of the outcomes of English Premier League games by utilizing rating systems and machine learning techniques. The resulting prediction models from this application are derived either through team rankings, statistical methods, or a combination of machine learning techniques. Our research findings from the integration of rating systems with machine learning techniques are highly encouraging in terms of predictive quality and risk-adjusted investment opportunities.

Moreover, three distinct applications have been developed in other fields than sports, where the rating systems are utilized with or without combining machine learning approaches. The first application concerns the ranking of domain names, the second deals with financial management and optimization contexts, and the third focuses on user preference ratings and recommendations.

This thesis introduces two rating systems. The first is a novel rating system that deals with the rating and ranking of soccer teams by taking into account the outcome of games, the margin of victory, and the shooting accuracy of each team. The second system is more generalized and applicable in various fields for rating/ranking where the basic idea behind the method is the WSM (Weighted Sum Method) and in fact, it is a modified version of the MAUT (Multi-Attribute Utility Theory) / MAVT (Multi-Attribute Value Theory). The effectiveness of our proposed systems is evaluated in the main application of this thesis where they have been compared to other established systems with satisfactory results. Additionally, the second system is utilized in the second distinct application that focuses on optimization contexts in finance.

Finally, an open-source software dedicated to the implementation of rating/ranking systems with applications and examples in sports and other fields was developed and is provided.

Keywords: rating methods; ranking systems; machine learning; prediction models; rating applications

Περίληψη

Η παρούσα διατριβή εξετάζει τα συστήματα βαθμολόγησης και τις εφαρμογές τους σε διάφορα πεδία. Τα συστήματα που μελετήθηκαν, στην πλειοψηφία τους αναπτύχθηκαν κυρίως για τον αθλητικό τομέα. Για το λόγο αυτό, η κύρια εφαρμογή που αναπτύχθηκε στη διατριβή σχετίζεται με τον αθλητικό τομέα, εστιάζοντας στις προβλέψεις τελικών αποτελεσμάτων στο Αγγλικό Πρωτάθλημα Ποδοσφαίρου (English Premier League), με τη χρήση των συστημάτων βαθμολόγησης και τεχνικών μηχανικής μάθησης. Τα προβλεπτικά μοντέλα που προκύπτουν από την εφαρμογή, προέρχονται είτε μέσω των κατατάξεων των ομάδων, είτε από στατιστικές μεθόδους ή από το συνδυασμό με τεχνικές μηχανικής μάθησης. Τα ερευνητικά αποτελέσματα από το συνδυασμό των μεθόδων βαθμολόγησης με τεχνικές μηχανικής μάθησης είναι ενθαρρυντικά, τόσο για την ποιότητα των προβλέψεων, όσο και για την αξιοποίησή τους επενδυτικά.

Επίσης, αναπτύχθηκαν τρεις επιμέρους εφαρμογές σε πεδία εκτός αθλητισμού, όπου τα συστήματα βαθμολόγησης αξιοποιήθηκαν μεμονωμένα ή συνδυαστικά με τη μηχανική μάθηση. Η πρώτη εφαρμογή αφορά την κατάταξη διαδικτυακών ονομάτων, η δεύτερη ασχολείται με θέματα χρηματοοικονομικής διαχείρισης και βελτιστοποίησης, και η τρίτη εστιάζει σε βαθμολογίες προτιμήσεων χρηστών και συστάσεις.

Μέσω της διατριβής αναπτύχθηκαν δύο συστήματα βαθμολόγησης. Το πρώτο εστιάζει στη βαθμολόγηση και κατάταξη ομάδων ποδοσφαίρου, λαμβάνοντας υπόψη σε κάθε αγώνα το τελικό αποτέλεσμα, τη διαφορά των τερμάτων (goals) και την ακρίβεια των σουτ (shots) κάθε ομάδας. Το δεύτερο σύστημα είναι γενικά εφαρμόσιμο σε πεδία εκτός αθλητικών ομάδων και βασίζεται στην ιδέα του Σταθμισμένου Μέσου Όρου και αποτελεί μια τροποποιημένη έκδοση της Πολυκριτήριας Θεωρίας Χρησιμότητας/Αξίας. Η αποδοτικότητα των δυο συστημάτων εξετάστηκε στην κύρια εφαρμογή με ικανοποιητικά αποτελέσματα σε σχέση με τα υπόλοιπα συστήματα που μελετήθηκαν. Επιπλέον, το δεύτερο σύστημα έχει αξιοποιηθεί στη δεύτερη επιμέρους εφαρμογή, η οποία εστιάζει σε θέματα βελτιστοποίησης στο χρηματοοικονομικό τομέα.

Τέλος, προσφέρεται ένα εργαλείο λογισμικού ανοικτού κώδικα το οποίο περιέχει υλοποιήσεις των συστημάτων βαθμολόγησης με εφαρμογές και παραδείγματα.

Λέξεις Κλειδιά: μέθοδοι βαθμολόγησης, συστήματα κατάταξης, μηχανική μάθηση, μοντέλα πρόβλεψης, εφαρμογές βαθμολόγησης

To my parents

for their unconditional love and support.

Στους γονείς μου

για την άνευ όρων αγάπη και υποστήριξη τους.

Acknowledgments

I am deeply grateful to my supervisor Retired Professor George Stephanides for his guidance, inspiration, and constant encouragement throughout this journey. I am also immensely thankful for providing me with an invaluable opportunity to gain academic experience during my PhD. Moreover, I would like to extend my sincere appreciation to the members of my advisory committee, Professor Alexander Chatzigeorgiou and Professor Ioannis Refanidis, for their valuable suggestions. Also, I would like to thank my seven-member examination committee for their comments and suggestions.

I would like to express my deepest gratitude to Professor Ioannis Mavridis for providing me with the opportunity to participate in a European Union-funded project throughout my PhD. I am profoundly grateful for his trust in me and for giving me the privilege and honor to collaborate with him on this project.

My biggest thanks go to my parents for their support, encouragement, and unconditional love. Their unwavering belief in me has been the driving force behind my accomplishments.

Last but not least, I would like to thank all my other family members and friends who have supported me through this endeavor.

"My objective has always been to get better, no matter where my ranking is."

Luke Donald (English Professional golfer)

Table of Contents

List of Figures

List of Tables

Acronyms

A: Away (Team)

AWF: Anchored Walk-Forward

CS: Cost-Sensitive Learning

D: Draw

DT: Decision Trees

EPL: English Premier League

GA: Genetic Algorithm

GeM: Generalized Markov Method

H: Home (Team)

IMDb: Internet Movie Database

KNN: K-Nearest Neighbors

LR: Logistic Regression

MAUT: Muti-Attribute Utility Theory

MAVT: Muti-Attribute Value Theory

MCDM: Multi-Criteria Decision Making

ML: Machine Learning

NB: Naive Bayes

NN: Neural Networks

ODM: Offense-Defense Method

RF: Random Forest

RS: Rating System

RWF: Rolling Walk-Forward

SMA: Simple Moving Average

SVM: Support Vector Machine

TG: Total Goals

TS: Total Shots

TST: Total Shots on Target

TW: Total Wins

WL: Win-Loss Method

WSM: Weighted Sum Method

1 - Introduction

1.1 Overview of Rating and Ranking

Although the idea of rating and ranking a set of items dates back to the 13th century (Langville & Meyer, 2012), the need for accurate and efficient systems around these tasks acquired great interest over the last three decades, giving room to the development of a plethora of methods. Rating method or system is the process of evaluating each of a set's items based on some desirable attributes by assigning a numerical score whereas ranking refers to the need to arrange a group of items based on their importance. Essentially, ranking is the process of sorting a list containing the same type of items, based on their rating scores.

A noteworthy category that has attracted much scholars' interest is the pairwise comparison ranking systems. Those systems are based on the pairwise comparison method (Fechner, 1860; Fechner, 1966) which offers an easy way to rate and rank items by comparing them in pairs. A considerable number of methods in widespread use rely on pairwise comparisons and especially on the category of sport rating methods. Usually, ratings are generated by employing linear algebra and computational methods.

After rating, the ranking of an item yields its position relative to a set of items with similar attributes. This allows the observer to determine which item between two or more, is better, simply by checking their positions. In turn, this can allow quite accurate predictions in case two items of the given set are put against one another (Kyriakides, Talattinis, & Stephanides, 2015). Items can be for example sports teams (which one will win the game), movies (which one is more relevant), marathon runners (who will win the race), etc. Especially, for sports teams, many rating methods are also utilized for predictions of the match outcome. The challenge becomes even greater when sports outcome predictions can be utilized in betting. Considering the report of the European Gaming & Betting Association - EGBA (European Gaming & Betting Association, 2022) in 2022, "sports/other types of betting" is the second most popular product generating 35% (€13.6 billion) of the gross gaming revenue. Also, in 2021, sports betting is the first product that generates 46% (€5.3 billion) of the total online gross gaming revenue. There is no need to question why many scholars and professionals are trying to predict sports outcomes by employing rating systems, statistical methods, and other techniques from various fields.

Rating systems are also considered essential tools in decision-making and find applicability in many scientific fields. In decision theory, for example, many methods are implemented to generate rating lists, which are used to rank alternatives and ultimately choose the one that outperforms. In those problems, the purpose is to model a decision process that dictates the optimal solution under certain restrictions (resources, preferences, third-party reactions, etc.). In finance, evaluation metrics are used in order to rank potential investments. Rankings are also important in business market applications like product recommendation systems which utilize users' behavior to predict preferences and recommend products to potential customers. This is often the case in electronic shops, such as Amazon or eBay are indicative examples. In October 2006, the media service provider – Netflix announced a contest where a one million US dollar prize was offered to the best team that would improve its recommendation system (Bennett & Lanning, 2007). Another application that has seen a great rise in recent years is the ranking of social network users according to their popularity.

Depending on the problem's nature the ranking of items and the selection among alternative solutions can be achieved by integrating methods from various scientific fields such as Artificial Intelligence, Operations Research, Decision Theory, Game Theory, and Rating/Ranking. Especially, Artificial Intelligence is a rapidly growing field with Machine Learning being one of the key subfields that is also expanding quickly with new techniques and applications developed. Machine learning has become increasingly important due to the growing need for intelligent decision-making in various fields. In the context of Machine Learning, rating values can be utilized in the feature engineering process of machine learning algorithms in order to improve their target (i.e., predictions). Also, Learning To Rank (LTR) is a distinct and active area within Machine Learning that employs various algorithms and techniques for the development of models to rank items.

1.2 Problem and Motivation

From the preceding section, it is evident that rating and ranking comprise a scientific field of great interest to both academia and the business community. In spite of this, below we outlined some issues.

- Integration with techniques from other fields

Several studies have demonstrated various attempts to develop more accurate ranking systems under the target of their sufficient performance in predicting the relevant

importance and power among different alternatives. However, in many cases, ranking systems are unable to propose a globally optimal solution. Due to the complexity of real-world decision problems, there are usually multitudes of viable alternatives. To effectively tackle the rating task it is essential to incorporate techniques from other fields.

❖ Soccer outcome prediction oriented to betting opportunities

One notable category of rating methods is those that have their origin in sports (Colley, 2002; Massey, 1997; Keener, 1993) many of them are initially designed to rate and rank teams for National Collegiate Athletic Association (NCAA) football and National Football League (NLF). However, those methods have a main task to rate and rank "the best". For example, in sports team rankings, the term "best" usually refers to the top-performing team. While existing studies are mainly focused on the best rankings other studies have applied rankings to make predictions in sports outcomes in games such as football, basketball, and soccer (Govan A. Y., 2008; Kvam & Sokol, 2006; Hvattum & Arntzen, 2010; Lasek, Szlávik, & Bhulai, 2013). The challenge intensifies when the purpose is to predict the soccer match outcome. Compared to other popular sports such as football or basketball, soccer is quite different because it is a low-scoring sport and admits ties. For example, in the English Premier League (EPL), matches end up in ties almost a quarter of the time. In the NFL ties rarely happen while the NBA (National Basketball League) avoids ties, by giving extra time to declare a winner.

However, soccer is one of the more popular sports to bet on in Europe, and despite all the opposition and extensive research by numerous authors, there is still room for improvement in terms of developing profitable as well as risk-adjusted models. Evaluating the available literature, there is some evidence of inefficiencies in the soccer betting market (Dixon & Pope, 2004; Goddard & Asimakopoulos, 2004; Angelini & Angelis, 2019) which means that many interesting approaches could be utilized to define efficient models and strategies in terms of profitability. Also, our reason for believing in the potential for improvement is that many studies that utilize rating systems are mainly focused on their predictive ability, such as (Lasek, Szlávik, & Bhulai, 2013) study. Although few studies, such as (Hvattum & Arntzen, 2010) study, concentrate on the rating systems to generate prediction models oriented to profitability from betting, most of them do not evaluate the potential risk in a manner similar to investments. Furthermore, only a limited number of studies such as (Herbinet, 2018) utilize rating systems by combining machine learning techniques.

❖ Applications in other fields

The majority of sports rating systems are primarily used in the field of sports with few being applied in other areas. Most studies adopt the Elo system (Elo, 1978) that started from chess player rankings and is also applied in other fields. Nevertheless, many other sports rating systems have the potential to be applied in other sectors than sports.

❖ Software

Another issue is that the range of open-source software developed for the mentioned methods is very limited and existing packages are mainly focused on the implementation of the rating methods and not on the application part (Talattinis & Stephanides, 2022).

Overall, the rating and ranking problem from the perspective of this thesis has the following challenges:

- Development of effective rating and ranking systems.
- Combination of rating systems with machine learning techniques in order to improve the results.
- Utilization of ratings in prediction models. Although there are several fields in which predictions are important, our focus is oriented on soccer outcome prediction which as we explained is a more challenging task.
- Development of applications where the rating methods are integrated as components of larger processes such as recommendations, optimization procedures, etc.
- Implementation of rating and ranking methods with their various applications as an open-source project.

Our motivation in this thesis is to apply and examine the efficiency of such rating systems and combine them with machine learning techniques in the context of sports and other fields. Particularly for the sports field, we refer to the soccer team ratings and their utilization in the prediction of soccer match outcomes.

1.3 Aims - Objectives

This thesis has the following four objectives:

(1) To propose novel rating systems or altered approaches to existing ones.

(2) To examine the potential of utilizing rating methods in combination with machine learning techniques for predicting soccer match outcomes. The prediction models

can be focused on targeting high accuracy or profit and risk-adjusted performance when they are employed in betting.

(3) To conduct an empirical evaluation of rating methods in real-case problems in other fields than sports.

(4) To develop an open-source library that implements several rating methods and their applications.

1.4 Contributions

Our contribution is four-fold:

(1) Two ranking systems are proposed. The first one is a novel rating method for rating/ranking soccer teams by utilizing the game outcomes, margin of victory, as well as shooting accuracy of each competing team. The second is a modified rating system based on the Multi-attribute Utility/Value Theory (Keeney & Raiffa, 1976) that ranks and compares alternatives utilizing user preferences and it is a sophisticated and generalized rating method with applications in various fields. Both systems are tested for their performance and efficiency on the experimental part of this thesis and they have produced positive outcomes. Our proposed methods can be useful tools for researchers, decision-makers, data scientists, sports analysts, coaches, bettors, and other similar groups.

(2) A comprehensive application is developed and deals with soccer outcome prediction. By comparing multiple groups of prediction models, we have demonstrated that the combination of rating methods with machine learning techniques leads to superior performance in many cases. The prediction models have developed by focusing either on (1) high prediction accuracy or (2) better risk-adjustment models when they are employed in betting. Especially for the latter, our research indicates that it is worthwhile to employ cost-sensitive methods for the successful predictions of soccer match outcomes that imply risk-adjusted models in the betting market. Our findings are highly encouraging and provide valuable insights for soccer outcome prediction models.

(3) Real-world application cases are provided where our proposed methods and other popular ranking systems are applied. The contributions of the 3 applications are:

A. The case of the Domain Name Market: provides the utilization of rating systems in fields outside of their traditional use. By providing the feasibility

and effectiveness of rating systems in the domain names ranking, this application could inspire further exploration and innovation in the other alternative markets such as the NFT (Non-fungible Token) market or the virtual properties market in Metaverse.

B. Financial Management and Optimization: utilizes a rating system as a fitness function of a genetic algorithm where a financial trading strategy is targeted to be optimized based on the investor's profile. Our findings showed that the genetic algorithm produces results capable of satisfying different users' preferences. This implies that the genetic algorithm was guided correctly by the utilization of the rating system we examined. Thus, it can be effectively adopted in the optimization process. This application serves as a useful resource for similar problems and can help decision-makers, portfolio managers, and investors.

C. Users' preference ratings and recommendations: provides enhanced results for the recommendation process of movies. Specifically, by integrating ranking results of rating systems as a filter into the recommendation process better recommendations and predictions for user ratings are performed. This suggests that the rating systems we tested can be used as part of the recommendation process in order to improve the results.

(4) Finally, we offer the RatingsLib, an open-source library in Python dedicated to the implementation of rating/ranking systems with applications in sports and other fields. In this way, the proposed software significantly helps researchers, scientists, and professionals to rate and rank items. It can also be used to make predictions for the outcome of future sports games by combining a plethora of well-known rating methods with other approaches and techniques such as machine learning. Furthermore, the applications and other functionalities provided can serve as valuable tools for data scientists and other similar groups.

1.5 Outline and Publications

The remainder of this thesis is structured as follows:

❖ Chapter 2

An overview of the related literature is conducted. A range of significant studies associated with rating and ranking are discussed and according to the field they

belong to they have been divided into four categories: Rating/Ranking, Decision Theory, Performance Evaluation Metrics, and Machine Learning Approaches.

- Chapter 3

  The detailed theoretical background of popular rating systems along with rank and rating aggregation methods is studied in this chapter. Also, prediction techniques and comparison measures are discussed. An illustrative example is provided in order to show how to rate, rank, aggregate, predict, and compare the results with the discussed approaches.

- Chapter 4

  Our proposed rating systems namely AccuRATE and PointRATE are introduced. The AccuRATE is developed for rating soccer teams while the PointRATE is an alteration to the WSM and MAUT/MAVT methods and is more generalized. Examples of their application are provided and comparisons to the other methods discussed in Chapter 3 are conducted.

- Chapter 5

  The machine learning background is studied. Various well-known techniques and algorithms that are extensively applied for predictive modeling are presented. The combination of rating methods and machine learning techniques is discussed.

- Chapter 6

  The main application is provided and deals with soccer outcome prediction in the English Premier League by utilizing rating systems and machine learning methods. The related work, experimental design and procedure, results conducted, as well as many aspects of the application are covered in this chapter. The results are evaluated with a comparison to results derived from baseline models.

- Chapter 7

  Three real-world applications of rating systems are introduced and are intended to show the use of rating systems in other fields rather than sports. The first deals with the rankings in the domain name market, the second with financial management and optimization, and the third with user-preference ratings and movie recommendations.

- Chapter 8

  Our proposed open-source software library named RatingsLib is introduced. The architecture, functionalities, usage, and impact overview of RatingsLib are presented.

❖ Chapter 9

Our concluding remarks are presented. A discussion of our results is conducted by considering our initial objectives and our contributions. Limitations and future directions for this line of research are discussed.

For a better reading flow, we decided to place our proposed rating systems after Chapter 3 because they share the same illustrative example, and next the theoretical background of machine learning is placed in Chapter 5 due to its connection with the subsequent application chapters.

Overall, in the present thesis, there is one chapter for the literature review, two chapters for the theoretical background (rating methods, and machine learning techniques), one chapter for the proposed methods, two chapters devoted to applications (soccer outcome prediction, and other fields), and one chapter is centered to the implementation. Finally, the last section provides the conclusions. We have organized the objectives, contributions, and outline of this thesis into a schematic representation, shown in Figure 1-1, which indicates the chapter in which each contribution can be found.

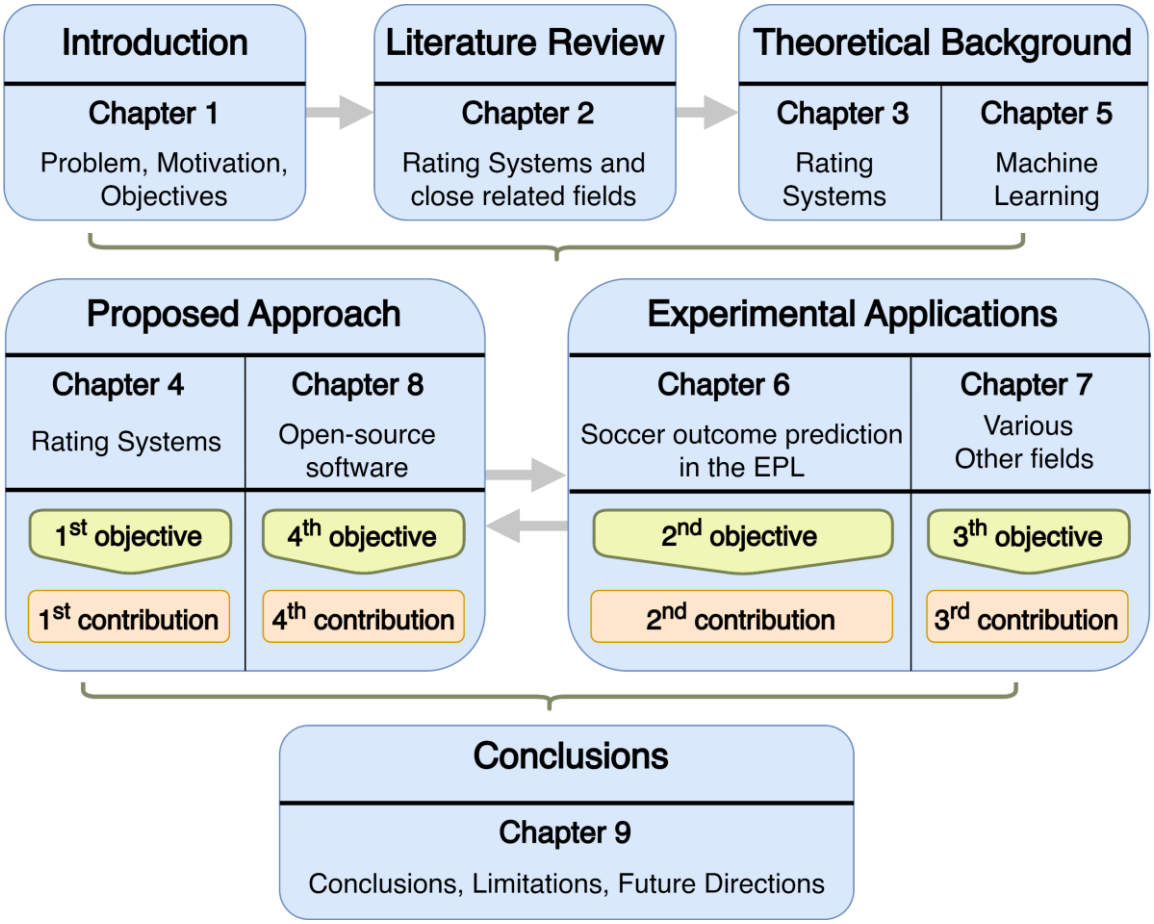


Figure 1-1: Thesis outline

This thesis has led to the following publications listed in Table 1-1 with their respective chapter.

Table 1-1: Publications

Publication	Type	Chapter
RatingsLib: A python library for rating methods with applications (Talattinis & Stephanides, 2022)	Journal	6, 7, 8
Forecasting Soccer Outcome Using Cost-Sensitive Models Oriented to Investment Opportunities (Talattinis, Kyriakides, Kapantai, & Stephanides, 2019)	Journal	6
A Hybrid Approach to Predicting Sports Results and an AccuRATE Rating System (Kyriakides, Talattinis, & Stephanides, 2017)	Journal	4, 6
Ranking Domain Names Using Various Rating Methods (Talattinis, Zervopoulou, & Stephanides, 2014)	Conference	7

In addition, the following publications made are also closely related to the main application of this thesis:

- Raw Rating Systems and Strategy Approaches to Sports Betting (Kyriakides, Talattinis, & Stephanides, 2015)
  Type: Conference
- Rating Systems Vs Machine Learning on the Context of Sports (Kyriakides, Talattinis, & Stephanides, 2014)
  Type: Conference

2 - Literature Review of Rating Methods

2.1 Introduction

In this chapter, we present some of the most common methods that are performed in different domains under the same objective - the comparison of similar objects/items. For the demands of this thesis, we studied methods of four major categories: Rating/Ranking, Decision Theory, Performance Evaluation Metrics, and Machine Learning Approaches. The remainder of this chapter is divided into five sections. In the first section (2.2), popular rating and ranking systems that mainly originate from the field of sports are included. Most of the studies showcased in section (2.2) employ linear algebra, computational and statistical techniques. In the second section (2.3) ranking methods from the decision theory are discussed. The third section (2.4) includes well-known performance evaluation metrics that are used in various fields. The fourth section (2.5) discusses machine learning techniques that are used for ranking. Finally, the last section (2.6) provides some conclusions.

2.2 Rating and Ranking

Ranking is the process of determining the importance of an item relative to the other items of a set it belongs to. Rating is a numerical value attributed to an item, based on some desirable criteria. Many researchers have proposed systems to rate and rank items of interest.

We begin with the pairwise comparison (or paired comparison) which was first introduced by Fechner in 1860 (Fechner, 1860; Fechner, 1966). Then in 1927, psychometrician Thurstone (Thurstone, 1927a; Thurstone, 1927b; Thurstone, 1927c) developed a model based on pairwise comparisons in order to rank items in terms of preferences or importance by utilizing an interval scale. The method of pairwise comparison is a way to rate and rank items by comparing them in pairs. In general, most of the methods we present below involve pairwise comparisons to produce a rating list and also have their origin in the sports field.

In 1951, Kenneth Arrow published the infamous Arrow's impossibility theorem (Arrow, 1951) that established the requirements of a fair voting method. There is a total of five principles. First, every voter's vote should be equally powerful (non-dictatorship). Second, the process must deterministically provide a complete ranking of the voters'

preferences (universality). Third, changes in individual rankings of irrelevant alternatives should not affect the ranking of the original subset (independence of irrelevant alternatives). Fourth, if a single individual changes their preference of A over B, by placing B higher than A, the process should place B at the same rank or higher, but not lower (monotonicity). Finally, every possible combination of ranks should be achievable by the process, given some set of individual preferences (non-imposition).

In 1952, Bradley and Terry proposed a model (Bradley & Terry, 1952) that is based on the idea that the probability of a pairwise outcome is directly related to their relative strength. The model has great importance and can be employed in applications to estimate the probability of a paired comparison outcome such as forecasting the winner of a sports match or the likelihood of a user buying a certain product. In 1960, Arpad Elo proposed the Elo system (Elo, 1978) which is based on the Bradley-Terry model and uses a modified form of the logistic function. The Elo was first used to rate and rank chess players and has been adopted by the United States Chess Federation (USCF). It is very popular and widely used in several fields.

Later, in 1990, the United Nations developed the Human Development Index, a system to rank the development level of a country, by taking into account people and their capabilities, as well as economic factors (UNDP, 1990). In 1993 James P. Keener introduced a novel rating approach based on the theory of nonnegative matrices and properties of the Perron-Frobenius theorem (Keener, 1993). A different approach was followed by Kenneth Massey in 1997 where a linear least squares methodology was proposed to rate and rank sports teams (Massey, 1997).

Sergey Brin and Larry Page pioneered introducing Google's PageRank in 1998 (Brin & Page, 1998), a graph-based algorithm to rank web pages by analyzing their links (Page, Brin, Motwani, & Winograd, 1999). PageRank is used in the popular Google search engine. One year later, Jon Kleinberg invented another method to rank linked documents named HITS from Hypertext Induced Topic Search, using the notions of hubs and authorities (Kleinberg, 1999). The HITS algorithm was used partly by the search engine Ask.com.

In the same year, Glickman developed the Glicko system (Glickman, 1999) an extension of the Elo system. The Glicko system takes into account the reliability of a player's rating, measures the rating deviation, and incorporates uncertainty into ratings. In 2002, Wesley Colley used a variation of Laplace's rule of succession to rank sports

teams (Colley, 2002). In 2003 a modified version of the Win-Loss rating system was published by C. Redmond (Redmond, 2003). This version considers the team's average dominance which is calculated by the point differentials and the number of games. Also, the method takes into account indirect comparisons and employs linear algebra in order to reach the final ratings.

In 2006, the LRMC (Logistic Regression / Markov chain) method was developed by Kvam and Sokol (Kvam & Sokol, 2006) to rank teams in American college basketball (NCAA). In their model, they use only basic scoreboard data and utilize a logistic regression model to estimate differences in teams' strength and then a Markov chain model to produce rank. One year later, in 2007, the Microsoft Research team developed the TrueSkill model (Herbrich, Minka, & Graepel, 2007) for rating players in multiplayer Xbox Live games. TrueSkill ranking system can be viewed as a generalization of the Elo system and additionally uses Bayesian inference for updating ratings after each game.

During 2007, Netflix hosted an open competition named "Netflix Prize", in order to find the best possible users' rating predictive algorithm for films (Bennett & Lanning, 2007). In the same year, Luke Ingram (Ingram, 2007) and Anjela Govan (Govan A. Y., 2008) used Markov chains to successfully rank NCAA basketball teams and NFL football respectively. The name of the method is known as Generalized Markov (GeM) and the main idea is based on the PageRank algorithm used by Google. Furthermore, in November of 2007, Callaghan et al. (Callaghan, Mucha, & Porter, 2007) introduced the Random Walker ranking method for ranking NCAA Division I-A Football teams. The method resembles the Markov method and the main idea behind is the voting by automated voters (random walkers) in which they cast their votes for the best team.

In 2008, a generalization of HITS was proposed for ranking sports teams, named "Offense-Defense" in Govan's PhD thesis (Govan A. Y., 2008; Govan, Langville, & Meyer, 2009). In 2012, McHale et al. (McHale, Scarf, & Folker, 2012) published the EA Player Performance Index for rating soccer players with a single score by taking into account their impact on winning performances. In 2017, we proposed AccuRATE (Kyriakides, Talattinis, & Stephanides, 2017), a rating system that rates and ranks soccer teams based on their performance and by considering the offensive opportunities.

We observe that a considerable number of methods we refer to belong to the sports rating systems. According to (Stefani R. T., 1999; Stefani & Pollard, 2007), sports rating systems can be divided into three categories: (1) subjective, (2) adjustive, and (3)

accumulative. The first refers to the rating given by experts. The second refers to the ratings that over a specific period increase in a cumulative manner. In the third category, the ratings can vary during a certain period, either getting higher or lower.

Having considered several key studies in the field, we will discuss another important category of methods called rank and rating aggregation methods. These methods aim to generate more accurate ratings/rankings by combining the results from multiple lists. In the case of multiple ranking lists, the rank aggregation method combines ranking results into a single ranking list. Also, it is possible to aggregate multiple rating lists into one, in this case, methods rely on the rating scores and we refer to rating aggregation methods. Some well-known rank aggregation approaches are the Borda count (Borda, 1784), Average Rank, and Copeland's method (Copeland, 1951). Markov chains are used by Dwork et al. (Dwork, Kumar, Naor, & Sivakumar, 2001a). Amy N. Langville and Carl D. Mayer in their book (Langville & Meyer, 2012) present two sophisticated approaches: (1) the Simulated Game Data rank aggregation technique, and (2) the Graph Theory method of rank aggregation. Additionally, the authors illustrate rating aggregation approaches such as employing the GeM method or utilizing the Perron eigenvector. Moreover, Govan et al. (Govan, Langville, & Meyer, 2009) used rating aggregation in their proposed system "Offense-Defense" in order to combine offensive and defensive rating scores into a single rating list.

Rank and rating aggregation techniques are useful in various fields. For example, in recommendation systems, multiple users rate items such as movies and the aggregation method can be used to generate a single list that reflects the preferences of all users. In sports competitions, the combination of various methods can determine the winner or the ranking of players. Also, in information retrieval, meta-search engines combine the results of multiple search engines. Those examples, as well as others, are presented in (Dwork, Kumar, Naor, & Sivakumar, 2001b; Langville & Meyer, 2012).

The application of rating and ranking systems in several fields constitutes another important topic. It is noteworthy that most applications in the literature of the methods discussed above are primarily centered on the sports field, as the majority of the methods developed for this field. The Bradley-Terry, Elo, and PageRank are the most popular choices for the development of applications in other fields than sports. The applications of rating systems in the sports field will be discussed in the subsequent chapters.

Consequently, the following is a selection of relevant studies that contribute to other fields than sports.

- Hastie and Tibshirany proposed the pairwise coupling model (Hastie & Tibshirani, 1997) that is similar to the Bradley-Terry model. Their approach is used in the machine learning field for the estimation of multi-class probabilities.
- Coulom utilized the Elo system (Coulom, 2007) in the "Go" game in order to compute pattern ratings. To compute them evaluate each move in the training database as a win for a particular pattern over the others. Then, Elo ratings of patterns can be used to calculate a probability distribution for all feasible moves in a new position.
- Gori and Gucci proposed the ItemRank (Gori & Pucci, 2007), an algorithm for recommendations of items based on user preferences. ItemRank is closely tied with PageRank.
- Chartier et al. applied the Colley system to rank movies (Chartier, Langville, & Simov, 2010).
- Talattinis et al. utilized various rating methods in order to rank domain names in the domain name market by considering various factors (Talattinis, Zervopoulou, & Stephanides, 2014).
- Pelánek employed the Elo rating system in education (Pelánek, 2016). In this application, the authors regard the response given by students to an item as a way to define the notion of a match between the student and an item. Elo system is used to efficiently estimate the skill of students in a dynamic manner and the difficulty of items.
- Talattinis and Stephanides introduced RatingsLib, an open-source software for rating methods with their applications (Talattinis & Stephanides, 2022).
- Numerous other applications are developed: Online social network users (Heidemann, Klier, & Probst, 2010); Twitter-like forums (Das Sarma, Das Sarma, Gollapudi, & Panigrahy, 2010); Information security (Pieters, van der Ven, & Probst, 2012); Wine tasters (London & Csendes, 2013).

2.3 Decision Theory

In decision theory, several methods have been developed to rank the alternatives and help the decision-makers to make decisions. For instance, when deciding among

options of a trip's means of transport, the alternatives could be a train, an airplane, or a bus. The bus may be the cheapest option, the train may be the most comfortable and the airplane may be the fastest. The best alternative is different for each traveler and as a result, methods originating from decision theory are more suitable.

Multiple-criteria decision-making (MCDM) is a sub-discipline of decision theory and aims to aid in such complex processes. As a methodological process usually, the problem is divided into smaller parts and described as a set of alternatives with their characteristics which are called criteria or attributes. Many different methods can be categorized as MCDM. Two of the simplest methods are the Weighted Sum Method (WSM) and the Weighted Product Model (WPM). Both methods are easily performed without the need for complex calculations. After defining weights for each criterion, these weights are used to sum or multiply the criteria with each other.

Multi-Attribute Value Theory (MAVT) and Multi-Attribute Utility Theory (MAUT) (Keeney & Raiffa, 1976) are both decision-making methods that rate alternatives in order to help decision-makers make choices. Specifically, MAVT and MAUT belong to the Multi-Attribute Decision Making (MADM) which is within the field of MCDM. The difference between the two methods is that MAUT is designed to take into account decisions under risk and uncertainty. MAVT uses value functions in order to rate and rank alternatives while MAUT is based on utility theory (Neumann & Morgenstern, 1947) and extends the MAVT. In particular, MAVT is a simplified form of MAUT. Moreover, MAUT has been utilized by numerous researchers in various fields, and it has the benefit of considering uncertainty as a factor. In both methods after determining the utility/value of each attribute, the attributes' weights are applied to compute the aggregated score of each alternative, and finally, the best alternative is selected based on the highest overall utility/value. Overall, MAUT and MAVT have been proven effective in addressing a wide range of problems in various fields due to their adaptability. MAUT and MAVT are also discussed in subsection 4.3.2.

Another widely used method is the Analytical Hierarchy Process (AHP) developed by Thomas Saaty. The method involves subdividing a problem into a structured hierarchy of criteria and alternatives. The user by utilizing Saaty's scale assigns a value to the relative importance between the criteria, as well as to the relative strength of each alternative with respect to each criterion. During this method, pairwise comparisons of alternatives against criteria are applied in order to compare the

performance of the alternatives. The problem of eigenvalues and eigenvectors is involved in deriving the weights that satisfy the consistency requirements of the method. To obtain the ratings, the score of each alternative is computed by summing the results of multiplying the weight of each criterion by the score of the alternative on that criterion. AHP has a connection with the methods of Keener and Massey mentioned in the previous section (Langville & Meyer, 2012). The AHP method has found applicability in a variety of fields. In sports, AHP is applied by (Bodin & Epstein, 2000) to rank the players in Major League Baseball (MLB). Also, (Sinuany-Stern, 1988) used AHP to rank 16 soccer teams of the Israeli National League.

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is a method developed by Hwang and Yoon in 1981 (Hwang & Yoon, 1981) for making multi-criteria decisions that rank and evaluate alternatives according to multiple criteria. The method compares the similarity of alternatives that are calculated by considering the distance between each alternative with the positive-ideal (best) and with the negative-ideal (worst) solution. The underlying idea is that the best alternative is one that is closest (minimum distance) to the positive-ideal solution and farthest (maximum distance) to the negative-ideal solution. TOPSIS is employed in a variety of fields. Among the studies that have been conducted one that concerns the sports field is from Kiani et al. (Kiani Mavi, Kiani Mavi, & Kiani, 2012) who have ranked football teams by combining AHP and TOPSIS methods.

ELECTRE (ELimination Et Choix Traduisant la REalité) is a family of MCDM methods that belongs to outranking methods and helps decision-makers to choose among alternatives. Bernard Roy's (Roy, 1968) paper was the first published paper on ELECTRE. It is classified as an outranking method which indicates that alternatives are ranked according to the utilization of the outranking relations. More specifically, ELECTRE builds a ranking of alternatives depending on their overall amount of preference by first establishing outranking relations between alternatives through pairwise comparisons. Several common characteristics can be observed between ELECTRE and the PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluation) ranking method initially proposed by J.P. Brans (Brans, 1982) which is considered to be a member of the outranking-methods family.

Another popular and different approach that is used in decision-making and belongs to the field of Operations Research is the Data Envelopment Analysis (DEA).

Farrell introduced the basic DEA model in 1958 (Farrell, 1957) and later in 1978 developed by Charnes et al. (Charnes, Cooper, & Rhodes, 1978). DEA is a non-parametric method in that the relative effectiveness of Decision Making Units (DMU) is evaluated based on their inputs and outputs. Various domains employed DEA, and several studies investigated its applicability in sports clubs (teams). Notably, (Barros & Leach, 2006) evaluated the performance of English Premier League teams based on sports and financial factors while (Haas, 2003) evaluated the technical efficiency of Major League soccer teams.

2.4 Performance Evaluation Metrics

Evaluation metrics are quantifiable measures that aim to compare, evaluate, and track progress. Usually, they provide a numerical representation of the score and they are used in many different contexts to measure the performance of models, methods, algorithms, items, etc. in order to make comparisons. In various fields, performance evaluation metrics are increasingly employed to define the final rating and ranking. Notably, we refer to those mainly used in the field of Machine Learning and Computational Finance because many of them are utilized in this dissertation.

In the field of Machine Learning, several evaluation metrics are used to measure the performance and the quality of the prediction models. In classification models, metrics such as accuracy, precision, recall, F1-score, etc. are the most common and easy to interpret. For example, the most straightforward way to rank prediction models is by their accuracy. However, it is common to use multiple evaluation metrics to gain a comprehensive understanding of the performance. We will provide an explanation of all these metrics, as well as many others in section 5.3. In regression models, popular metrics are the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Also, several metrics for evaluation that initially were used in other areas are adopted in machine learning. Brier score (Brier, 1950), and Matthews Correlation coefficient (Matthews, 1975) are used in classification, Fowlkes-Mallow score is used in clustering (Fowlkes & Mallows, 1983), Kendall's tau (Kendall, 1938) ranking correlation is used in learning to rank. Ranked Probability Score (RPS) is used to evaluate the performance of probabilistic predictions (Epstein, 1969; Murphy, 1969; Murphy, 1971).

In the field of Computational Finance, it is imperative to be able to choose from a multitude of different investing strategies. Each strategy may have its strengths and

weaknesses, but through careful selection, it is possible to create a portfolio that maximizes strengths and minimizes weaknesses. Performance evaluation metrics are used to identify these strengths and weaknesses. Each metric is designed to evaluate a certain aspect of a strategy, such as its profitability, risk, etc. It is difficult to design a metric that can fully describe a strategy, although by using certain different metrics, a potential investor can decide if a specific strategy possesses the desired characteristics. A useful common metric in terms of profitability is the ROI (Return on Investment) which measures the investment's profitability compared to its cost. Similarly, ROC (Return on Invested Capital) measures profitability relative to invested capital. Moreover, HPR (Hold Period Return) is a percentage of the total return from an investment in a given time period.

Value-At-Risk is a financial risk quantification measure over a defined time frame (Jorion, 2006), while Conditional-Value-At-Risk is used to quantify the probability that a specific loss will be larger than expected by Value-At-Risk. The Ulcer index was proposed by (Martin & McCann, 1989) and measures the intensity and duration of all past drawdowns. The lower value of the Ulcer index is better.

An important family of financial metrics is the risk-adjusted. Their role is essential in portfolio evaluation and selection because they consider risk and returns. The Sharpe ratio is a risk-adjusted metric developed by William Sharpe (Sharpe, 1994) and is used to assess an investment's returns compared to its risk level. Specifically, it measures the excess returns over the risk-free rate taking into account the risk level. The standard deviation of returns represents the level of risk and a higher Sharpe ratio value indicates a higher return with regard to the risk taken. The Sortino ratio (Sortino & Price, 1994) is also a useful metric for assessing the risk-adjusted return of investment by estimating the average return in relation to the downside deviation. It is a modification of the Sharpe ratio; however, as noted it measures risk using the downward deviation rather than the standard deviation. Another popular risk-adjusted metric is the Calmar ratio (Young, 1991) which considers the average annualized rate of return of an investment in relation to its maximum drawdown.

The development of performance evaluation metrics has been an important aspect of the software engineering field. Many different metrics and indicators are used with the aim of measuring the software development process or several characteristics of software projects. Moreover, in several other fields, different metrics are employed such as

environmental metrics, health metrics, etc. The metrics employed generally vary and the most suitable to utilize will depend on the field and the purpose. In the present thesis, we utilized metrics from machine learning and computational finance in order to evaluate methods and models' performance.

2.5 Machine Learning Approaches

Various statistical and machine learning approaches have been used in the field of ranking to measure the relative importance of items depending on the application and data characteristics. A wide range of machine learning techniques has been adapted and extended for use in ranking.

Learning to Rank (LTR) is a novel and popular subset of Machine Learning (Cao, Qin, Liu, Tsai, & Li, 2007) that implements supervised machine learning, which entails training and testing phases and aims to address issues related to ranking (Li, 2011) and develop an algorithm or a model for ranking objects (Cao, Qin, Liu, Tsai, & Li, 2007). The ranking is fundamentally based on Information Retrieval; as a consequence, the ultimate objective of LTR is not simply to construct a variety of novel algorithms and principles but also to significantly improve the ranking performance (Liu T.-Y. , 2009). The training data consists of documents and queries (Li, 2011). The ability to incorporate several distinct attributes is a crucial benefit of LTR methods (Liu T.-Y. , 2009). By integrating the model's output as one dimension of the features, any new developments in the retrieval model can be effortlessly included (Liu T.-Y. , 2009). Learning to rank is beneficial in various applications, including collaborative filtering and document retrieval (Cao, Qin, Liu, Tsai, & Li, 2007).

The fundamental variations between conventional supervised machine learning and learning to rank lie in the following facts: The traditional approach in machine learning addresses prediction challenges, including classification or regression, by analyzing individual instances at a time. The technique of LTR addressed ranking challenges, by analyzing lists of items. The objective of LTR is to optimize the ranking of items through the utilization of machine learning techniques. The primary concern of LTR focuses on the relative ordering of items, as opposed to the precise scores assigned to each item.

The primary distinction between LTR and traditional supervised machine learning is detected in the methods used for training and evaluating the model.

Traditional supervised machine learning strives to estimate a target variable, which is typically a categorical or continuous variable, using input variables. As opposed to traditional supervised machine learning, the purpose of learning to rank is to predict the relevance of items (such as documents, products, images, etc.) for a particular task or inquiry. Another characteristic that sets LTR apart from traditional supervised machine learning, is that LTR can be evaluated using metrics such as Mean Reciprocal Rank (MRR).

Recommendation systems are another important field that is connected with ranking and machine learning. By utilizing user preferences recommender systems produce rankings for products, services, etc. that are provided to the user as personalized recommendations. Ranking methods and machine learning techniques play a critical role in recommender systems. Ranking systems use a variety of techniques to rate items by including factors such as popularity, user utility, etc. Machine learning enables the system to learn from user data and adapt to user preferences changes over time. LTR is also used in recommender systems with the aim of improving their accuracy and effectiveness.

Collaborative filtering, content-based, demographic recommender, utility-based, popularity-based, and hybrid methods are some of the most common recommendation techniques. Those methods are also discussed in section 7.4.2.

2.6 Conclusions

The main conclusion is that in the last decades, there has been a great development in methods concerning rating and ranking. This great progress combines multiple fields. It is worth mentioning that it is impossible to develop a system that rates and ranks at the same time sets of different item types. However, the same system can be used in various fields. For example, the Elo system initially was developed for the ranking of chess players and is also extended to soccer teams.

The choice of method depends on the specific application and the purpose it serves and it requires a great deal of expertise to decide which one to use. In this chapter, we have focused on a specific aspect of these methods that is relevant to the present dissertation, however, some of them will be covered in greater depth in the subsequent chapters.

3 - Theoretical Background of Rating Methods

3.1 Introduction

In this chapter, the detailed theoretical background for rating methods underlying this thesis is presented. The aim is to offer the reader the necessary background to fully capture the details of each method utilized in the study. For this purpose, we initially present the theory of several popular rating methods and we applied them in a small illustrative example for better understanding. Next, rank and rating aggregation techniques are studied. Then we introduce various methods of comparison and we show how can we evaluate and compare the rating lists. Finally, we end this chapter by providing concluding remarks.

3.2 An Illustrative Example

For ease of explanation of rating methods, we apply them to a small example of soccer teams, in order to fully connect it with the main application of this thesis. The implementation details along with the steps of each rating method are provided in section 3.3. This example concerns the first 20 matches that took place during the 2018-2019 sports seasons of the English Premier League (EPL). Note that in a soccer match, there are three possible outcomes, namely, the win of the home team (Home-win), the win of the away team (Away-win), or the Draw. To demonstrate the example, the following statistics for each team in a soccer game are selected:

TG: the total goals scored by each team in the game.

TST: the total number of shots-on-target by each team in the game. A shot on target is a shot intended to score a goal, which is either successful or can be prevented by the goalkeeper or the last defender.

TS: the total number of shots by each team in the game. Equivalently, are the total number of team's shot attempts to score a goal, regardless if the shots are on or off target. Shots off-target are shots that are not directed toward the goal.

FO: the final outcome of the match, there are three different values 'H' = Home-win, 'A' = Away-win, and 'D' = Draw.

The same statistics were used in the main application of this thesis in Chapter 6.

The teams were ordered alphabetically in the rating and ranking results of each method. The example is summarized in the table below where in some columns the initial letter 'H' indicates the Home team, while 'A' denotes the Away team.

Table 3-1: The first 20 games of the English Premier League 2018-2019 season

Date	Home Team	Away Team	HTG	ATG	HTST	ATST	HTS	ATS	FO
		- 1st match week -							
10/8/18	Man United	Leicester	2	1	6	4	8	13	H
11/8/18	Bournemouth	Cardiff	2	0	4	1	12	10	H
11/8/18	Fulham	Crystal Palace	0	2	6	9	15	10	A
11/8/18	Huddersfield	Chelsea	0	3	1	4	6	13	A
11/8/18	Newcastle	Tottenham	1	2	2	5	15	15	A
11/8/18	Watford	Brighton	2	0	5	0	19	6	H
11/8/18	Wolves	Everton	2	2	4	5	11	6	D
12/8/18	Arsenal	Man City	0	2	3	8	9	17	A
12/8/18	Liverpool	West Ham	4	0	8	2	18	5	H
12/8/18	Southampton	Burnley	0	0	3	6	18	16	D
		- end of 1st match week -							
		- 2nd match week -							
18/8/18	Cardiff	Newcastle	0	0	1	6	12	12	D
18/8/18	Chelsea	Arsenal	3	2	11	6	24	15	H
18/8/18	Everton	Southampton	2	1	7	4	13	15	H
18/8/18	Leicester	Wolves	2	0	2	3	6	11	H
18/8/18	Tottenham	Fulham	3	1	11	3	25	10	H
18/8/18	West Ham	Bournemouth	1	2	5	5	11	12	A
19/8/18	Brighton	Man United	3	2	3	3	6	9	H
19/8/18	Burnley	Watford	1	3	3	6	8	9	A
19/8/18	Man City	Huddersfield	6	1	14	1	32	5	H
20/8/18	Crystal Palace	Liverpool	0	2	2	6	8	16	A
		- end of 2nd match week -							

3.3 Popular Rating Systems

In this section, we introduce the theoretical part of several rating systems and we also demonstrate their application to our illustrative example.

3.3.1 Win-Loss Method

The Win-Loss method is a traditional well-known and maybe the oldest method used to rate a team's performance according to wins and losses. This method is derived from the field of sports. We refer to the Win-Loss method as WL.

The main idea of the method is to rate the teams according to the total number of wins. The first-ranked team is the team with the most wins. In the case of ties, we use a second criterion to break the ties such as the total points scored by each team. The main advantage of the Win-Loss is its simplicity, while its main disadvantage is that the method only takes into account the total wins, regardless of the margin of victory of games.

In the case where not all teams play the same number of games, we simply normalize the result of total wins. For each team, we divide the number of wins by the number of games played by the team.

We present below the details of the Win-Loss method:

At first, we form the Win-Loss matrix W using:

$$W_{ij} = \begin{cases} 1/n_i & if\ team\ i\ beats\ team\ j \\ 0 & otherwise \end{cases},$$

where n_i is the number of games played by the team i.

Finally, we compute the rating vector

$$r = W \cdot e^T,$$

where e is a vector of 1's.

❖ Application of the Win-Loss Method to the example of EPL games

We compute the rating vector according to the total wins of each team. In our example, the normalization procedure is not required because all teams have played the same number of games. The rating vector r contains the sum of wins for each team.

	Arsenal	Bournemouth	Brighton	Burnley	Cardiff	Chelsea	Crystal Palace	Everton	Fulham	Huddersfield	Leicester	Liverpool	Man City	Man United	Newcastle	Southampton	Tottenham	Watford	West Ham	Wolves
r^T = (	0	2	1	0	0	2	1	1	0	0	1	2	2	1	0	0	2	2	0	0)

Table 3-2 gives the rating and ranking results.

Table 3-2: Win-Loss rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	0	3	Leicester	1	2
Bournemouth	2	1	Liverpool	2	1
Brighton	1	2	Man City	2	1
Burnley	0	3	Man United	1	2
Cardiff	0	3	Newcastle	0	3
Chelsea	2	1	Southampton	0	3
Crystal Palace	1	2	Tottenham	2	1
Everton	1	2	Watford	2	1
Fulham	0	3	West Ham	0	3
Huddersfield	0	3	Wolves	0	3

3.3.2 Colley Method

This method was proposed by astrophysicist Wesley Colley in 2001 for ranking sports teams (Colley, 2002). Colley's method is based on very simple statistical principles. In fact, it is a modified form of the Win-Loss method, which uses the percentage of wins of each team. This percentage is given by $r_i = \frac{w_i}{n_i}$, where w_i is the total number of wins of team i, and n_i is the total number of games played for team i. Colley's method makes use of an idea from probability theory, known as Laplace's rule of succession (Ross, 2010, p. 98), which transforms the standard winning percentage as

$$r_i = \frac{1 + w_i}{2 + n_i}, \tag{3.1}$$

where l_i is the number of losses by team i.

To consider the strength of schedule Colley rewrites w_i of (3.1) in the following way:

$$w_i = \frac{2w_i}{2} = \frac{w_i - l_i}{2} + \frac{w_i + l_i}{2} = \frac{w_i - l_i}{2} + \frac{n_i}{2}. \tag{3.2}$$

At the beginning of the season $\frac{n_i}{2}$ in (3.2) represents the opponents' ratings. Then, over the course of the season, the cumulative ratings of opponents can provide a good approximation of it. Thus (3.2) can be rewritten as

$$w_i = \frac{w_i - l_i}{2} + \sum_{j=1}^{n_i} \frac{1}{2} = \frac{w_i - l_i}{2} + \sum_{j \in O_i} r_j,$$

where O_i are the opponents of team i.

Colley describes the summation $\sum_{j \in O_i} r_j$ as the adjustment for the strength of schedule. Finally, the rating can be derived by solving the following system of linear equations:

$$(n_i + 2)r_i - \sum_{j \in O_i} r_j = \frac{w_i - l_i}{2} + 1,$$

which can be simply rewritten as $Cr=b$. Then follows a summary of Colley's rating method: At first, we can form the Colley matrix C using:

$$C_{ij} = \begin{cases} -n_{ij} & i \neq j \\ 2 + n_i & i = j \end{cases}, \quad (3.3)$$

where n_{ij} is the number of games played by team i against team j.

Then, we compute vector b given by:

$$b_i = \frac{1 + (w_i - l_i)}{2}. \quad (3.4)$$

If the matchup ends in a tie, then w and l vectors remain unaltered. Therefore, vector b will not be modified. The elements of coefficient matrix C_{ij}, C_{ji} will be decreased by 1, and C_{ii}, C_{jj} will be increased by 1. Finally, the linear system from (3.3) and (3.4) is

$$C \cdot r = b, \quad (3.5)$$

where r is the rating vector for the teams and the solution to the system. The sum of the rating vector is equal to the total number of teams divided by two.

As follows from the above, the only information used by this model is the wins, losses, and number of games each team played. Thus, the generated ratings are bias-free, which implies that certain points gained by each team in a game are not included (Langville & Meyer, 2012, p. 24). In other words, a win is more important regardless of the score. Due to the use of Laplace's rule of succession, Colley's method has several advantages over the traditional rating formula:

(1) At the beginning of the season, each team has a rating of $\frac{1}{2}$, instead of the preseason rating $\frac{0}{0}$ of the traditional system, which does not make any sense.

(2) Colley's method takes into consideration the strength of schedule, which is the strength of a team's opponents. This implies that a team should be rewarded more for winning against a strong opponent than the reward for winning against a weaker one (Langville & Meyer, 2012, p. 22).

❖ Application of the Colley Method to the example of EPL games

Initially, we form the matrix C and the vector b.

$$C = \begin{array}{r} \\ \text{Arsenal} \\ \text{Bournemouth} \\ \text{Brighton} \\ \text{Burnley} \\ \text{Cardiff} \\ \text{Chelsea} \\ \text{Crystal Palace} \\ \text{Everton} \\ \text{Fulham} \\ \text{Huddersfield} \\ \text{Leicester} \\ \text{Liverpool} \\ \text{Man City} \\ \text{Man United} \\ \text{Newcastle} \\ \text{Southampton} \\ \text{Tottenham} \\ \text{Watford} \\ \text{West Ham} \\ \text{Wolves} \end{array} \begin{pmatrix} \begin{array}{c} \text{Arsenal} \end{array} & \begin{array}{c} \text{Bournemouth} \end{array} & \begin{array}{c} \text{Brighton} \end{array} & \begin{array}{c} \text{Burnley} \end{array} & \begin{array}{c} \text{Cardiff} \end{array} & \begin{array}{c} \text{Chelsea} \end{array} & \begin{array}{c} \text{Crystal Palace} \end{array} & \begin{array}{c} \text{Everton} \end{array} & \begin{array}{c} \text{Fulham} \end{array} & \begin{array}{c} \text{Huddersfield} \end{array} & \begin{array}{c} \text{Leicester} \end{array} & \begin{array}{c} \text{Liverpool} \end{array} & \begin{array}{c} \text{Man City} \end{array} & \begin{array}{c} \text{Man United} \end{array} & \begin{array}{c} \text{Newcastle} \end{array} & \begin{array}{c} \text{Southampton} \end{array} & \begin{array}{c} \text{Tottenham} \end{array} & \begin{array}{c} \text{Watford} \end{array} & \begin{array}{c} \text{West Ham} \end{array} & \begin{array}{c} \text{Wolves} \end{array} \\ 4 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix}$$

$$b^{\mathrm{T}} = \begin{pmatrix} \begin{array}{c} \text{Arsenal} \end{array} & \begin{array}{c} \text{Bournemouth} \end{array} & \begin{array}{c} \text{Brighton} \end{array} & \begin{array}{c} \text{Burnley} \end{array} & \begin{array}{c} \text{Cardiff} \end{array} & \begin{array}{c} \text{Chelsea} \end{array} & \begin{array}{c} \text{Crystal Palace} \end{array} & \begin{array}{c} \text{Everton} \end{array} & \begin{array}{c} \text{Fulham} \end{array} & \begin{array}{c} \text{Huddersfield} \end{array} & \begin{array}{c} \text{Leicester} \end{array} & \begin{array}{c} \text{Liverpool} \end{array} & \begin{array}{c} \text{Man City} \end{array} & \begin{array}{c} \text{Man United} \end{array} & \begin{array}{c} \text{Newcastle} \end{array} & \begin{array}{c} \text{Southampton} \end{array} & \begin{array}{c} \text{Tottenham} \end{array} & \begin{array}{c} \text{Watford} \end{array} & \begin{array}{c} \text{West Ham} \end{array} & \begin{array}{c} \text{Wolves} \end{array} \\ 0 & 2 & 1 & 0.5 & 0.5 & 2 & 1 & 1.5 & 0 & 0 & 1 & 2 & 2 & 1 & 0.5 & 0.5 & 2 & 2 & 0 & 0.5 \end{pmatrix}$$

Then, solving the system (3.5), we obtain the rating vector. The results are shown below:

Table 3-3: Colley rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	0.3333	16	Leicester	0.4732	9
Bournemouth	0.6860	3	Liverpool	0.7128	2
Brighton	0.5625	6	Man City	0.6667	5
Burnley	0.4018	10	Man United	0.5089	7
Cardiff	0.3943	11	Newcastle	0.3914	12
Chelsea	0.6667	5	Southampton	0.3661	14
Crystal Palace	0.5015	8	Tottenham	0.6711	4
Everton	0.5625	6	Watford	0.7411	1
Fulham	0.2932	17	West Ham	0.3497	15
Huddersfield	0.3333	16	Wolves	0.3839	13

3.3.3 Massey Method

This method was proposed by Kenneth Massey in 1997 for ranking college football teams (Massey, 1997). By using a system of linear equations, apart from the numbers of wins and losses of each team, it considers also game scores in the ratings, i.e., the margin of victory. The method uses a linear least squares regression to solve a system of linear equations. A brief description of the least square method is given below.

Every model describing a relation between two variables say x and y, can be written as

$$y = f(a, x, b),$$

where a, b are the intercept and the slope of the regression line respectively, x is the independent variable, represented as the horizontal axis, and y is the dependent variable, represented as the vertical axis.

The goal of the least squares regression method is to calculate the parameters, a, b in order to achieve the best fit on the data, by minimizing the sum of squared residuals. A residual is the vertical distance between a data point and the regression line.

For most analytic techniques the reference curve is a straight line, so the model is linear. Thus, a linear least squares regression model is used. The relation between each (data point) i-th pair (x_i, y_i) is written as

$$y_i = a + bx_i + e_i,$$

where y_i is calculated from the linear model $a+bx_i$ and the residual e_i.

Massey's method is based on the mathematical theory of least squares, which can be represented by the following equation:

$$r_i - r_j = y_k, \tag{3.6}$$

where r_i and r_j are the ratings of teams i and j, respectively and y_k is the margin of victory for a game k between these teams. Each game k can be given by an equation of this form, so a system of m linear equations and n unknowns is created, where m is the number of the games that have already been played and n is the number of teams. This system can be written as

$$X \cdot r = y, \tag{3.7}$$

where X_{ki} takes the following values: 1 if team i won against team j in the k-th game, -1 if team i lost against team j in the k-th game, or 0 otherwise. As we notice the system in (3.7) is overdetermined, because $m \gg n$, i.e., there are more equations than unknowns. To deal with this problem, Massey proposed the use of a matrix $X^T \cdot X$ instead of X, therefore, a least squares solution is obtained (Massey, 1997):

$$X^T \cdot X \cdot r = X^T \cdot y. \tag{3.8}$$

The matrix $M = X^T \cdot X$ can be easily filled considering that every diagonal element M_{ii} is the total number of games played by team i and every off-diagonal element M_{ij}, for $i \neq j$, is the negation of the number of games played by team i against team j. More specifically, we can form the matrix *M* using

$$M_{ij} = \begin{cases} -n_{ij} & i \neq j \\ n_i & i = j \end{cases}, \tag{3.9}$$

where n_i is the number of games played by team i and n_{ij} is the number of games played by team i against team j.

Consequently, the Massey least squares system now becomes

$$M \cdot r = d, \tag{3.10}$$

where $M_{n\times n}$ is the Massey matrix described above,

$r_{n\times 1}$ is the vector of unknown ratings, and

$d_{n\times 1}$ is the total difference in scores for each team.

The vector *d* of the total difference in scores for the team *i* is given by the equation $d = X^T y$.

As we observe from (3.9), the formation of *M* is simple and does not require the computations of (3.8). However, the columns of matrix *M* are linearly dependent, which leads to $rank(M) < n$ and so, the solution to the linear system (3.10) is not unique (Langville & Meyer, 2012, p. 10). Massey addressed this problem by replacing one of the rows in M with *e* and the corresponding entry of d with a zero, where *e* is a vector of all 1's. Specifically, this change makes the rank of matrix *M* full. The row in M chosen by Massey is the last one. This implies that a constraint is added to the system (3.10) where the rating vector produced by the method sums to zero.

Summarizing Massey's rating method, firstly we have to form the Massey matrix M and the vector *d* that represents the total difference in scores for each team, after that we have to force matrix *M* to have full rank by making some replacements and finally, we compute the Massey rating vector r by solving the linear system generated by the previous replacement.

❖ Application of the Massey Method to the example of EPL games

After replacing the last row (or any other row) of M with *e* (vectors of 1's) and the last element of *d* with zero, we call them as $\bar{M}$ and $\bar{d}$ respectively.

$$\bar{M} = \begin{array}{r} \\ \text{Arsenal} \\ \text{Bournemouth} \\ \text{Brighton} \\ \text{Burnley} \\ \text{Cardiff} \\ \text{Chelsea} \\ \text{Crystal Palace} \\ \text{Everton} \\ \text{Fulham} \\ \text{Huddersfield} \\ \text{Leicester} \\ \text{Liverpool} \\ \text{Man City} \\ \text{Man United} \\ \text{Newcastle} \\ \text{Southampton} \\ \text{Tottenham} \\ \text{Watford} \\ \text{West Ham} \\ \text{Wolves} \end{array} \begin{array}{c} \begin{array}{cccccccccccccccccccc} \text{Arsenal} & \text{Bournemouth} & \text{Brighton} & \text{Burnley} & \text{Cardiff} & \text{Chelsea} & \text{Crystal Palace} & \text{Everton} & \text{Fulham} & \text{Huddersfield} & \text{Leicester} & \text{Liverpool} & \text{Man City} & \text{Man United} & \text{Newcastle} & \text{Southampton} & \text{Tottenham} & \text{Watford} & \text{West Ham} & \text{Wolves} \end{array} \\ \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

$$\bar{d}^{\mathrm{T}} = \begin{array}{c} \begin{array}{cccccccccccccccccccc} \text{Arsenal} & \text{Bournemouth} & \text{Brighton} & \text{Burnley} & \text{Cardiff} & \text{Chelsea} & \text{Crystal Palace} & \text{Everton} & \text{Fulham} & \text{Huddersfield} & \text{Leicester} & \text{Liverpool} & \text{Man City} & \text{Man United} & \text{Newcastle} & \text{Southampton} & \text{Tottenham} & \text{Watford} & \text{West Ham} & \text{Wolves} \end{array} \\ \begin{pmatrix} -3 & 3 & -1 & -2 & -2 & 4 & 0 & 1 & -4 & -8 & 1 & 6 & 7 & 0 & -1 & -1 & 3 & 4 & -5 & 0 \end{pmatrix} \end{array}$$

The rating vector r is the solution to the system $\bar{M}\, r = \bar{d}$. The rating and ranking results are shown in Table 3-4.

Table 3-4: Massey rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	2.5000	11	Leicester	-5.5313	16
Bournemouth	4.7812	3	Liverpool	7.2812	1
Brighton	-4.7813	14	Man City	4.7500	4
Burnley	-6.0312	17	Man United	-5.1563	15
Cardiff	3.0312	9	Newcastle	3.2812	7
Chelsea	3.2500	8	Southampton	-6.6562	19
Crystal Palace	5.0312	2	Tottenham	4.5312	5
Everton	-6.2812	18	Watford	-3.4063	13
Fulham	2.7812	10	West Ham	3.5312	6
Huddersfield	0.0000	12	Wolves	-6.9063	20

3.3.4 Elo Method

The Elo system was originally developed by Arpad Elo (Elo, 1978) to rank chess players and has been adopted by quite a lot of sports and organizations. The method uses a modified form of the logistic function and takes into account the rating difference between the players and the expected probability of the game's outcome. Particularly, for each participant (or team) *i*, their rating after a match against *j* is calculated as follows:

$$r_i(new) = r_i(old) + K \cdot \big(T(i,j) - E(i,j)\big), \tag{3.11}$$

where $r_i(old)$ is the rating before the game, $r_i(new)$ is the new rating after the game, *K* is a constant determined by the sport and organization, and *T* is a ternary variable taking values 0, 0.5, and 1, if team *i* lost, tied, or won, respectively. Finally, the expected outcome is computed as follows:

$$E(i,j) = \frac{1}{10^{-\frac{r_i(\text{old}) - r_j(old)}{\xi}} + 1}. \tag{3.12}$$

The above is the classic version of Elo which we referred to as Elo-Win. In another version of Elo that incorporates game scores (points) (Langville & Meyer, 2012, p. 59), the *T(i,j)* is modified to

$$T(i,j) = \frac{P_{ij} + 1}{P_{ij} + P_{ji} + 2}, \tag{3.13}$$

where P_{ij} is the points (or goals in the case of soccer) that team *i* scores against team *j*. We refer to this modified version as Elo-Point.

It is worth mentioning that the K-factor plays an important role since it controls the impact that the difference between the actual and expected outcome of the game has on ratings. In soccer, according to the type of tournament the following table represents the K-Factor value suggested by several Internet sites such as World Football Elo Ratings (EloRatings, 2023):

Table 3-5: K-Factor for soccer

Tournament Type	K factor
World Cup Finals	60
Continental Championship Finals and Major Intercontinental tournaments	50
World Cup Qualifiers and Major Tournaments	40
All other tournaments	30
Friendly matches	20

The parameter ξ is a constant that scales the rating difference and has an impact on the rating range. For chess and soccer games usually, ξ is set to 400. This means that if team A has 400 rating points higher than team B, then team A is expected to win over team B with a probability that is 10 times higher than that of team B.

Additionally, due to the fact that home teams tend to score more goals, a home-field advantage factor can be applied by adding it to the home team's rating. Therefore, the equation (3.12) is rewritten as follows:

$$E(i,j) = \frac{1}{10^{-\frac{r_i(\text{old})+\text{h}-r_j(old)}{\xi}}+1}.$$

where *h* is the home-field advantage. Many implementations of the Elo model for soccer, set the home-field advantage to 100 (EloRatings, 2023).

❖ Application of the Elo Method to the example of EPL games

We have used the following parameters: ξ=400 and *K*=40 (the EPL belongs to the category of "Major Tournaments"). The steps to calculate Arsenal's rating using the Elo-Win are shown below:

1st match week: Arsenal – Man City, Final Outcome: 0-2

$$r_{Man\ City}(old) = 0, \quad r_{Arsenal}(old) = 0, \quad \text{and } T = 0$$

$$r_{Arsenal}(new) = r_{Arsenal}(old) + 40 \cdot \left(0 - \frac{1}{10^{-\frac{r_{Arsenal}(old)-r_{Man\ City}(old)}{400}}+1}\right)$$

$$r_{Arsenal}(new) = 0 + 40 \cdot (-1/2) = -20$$

Similarly, Chelsea's rating is computed: $r_{Chelsea}(new) = 20$ and it will be used in the next step.

2nd match week: Chelsea – Arsenal, Final Outcome: 3-2

$$r_{Arsenal}(old) = -20, \quad r_{Chelsea}(old) = 20, \quad \text{and } T = 0$$

$$r_{Arsenal}(new) = r_{Arsenal}(old) + 40 \cdot \left(0 - \frac{1}{10^{-\frac{r_{Arsenal}(old)-r_{Chelsea}(old)}{400}}+1}\right)$$

$$r_{Arsenal}(new) = -20 - 17.7075 = -37.7075$$

The final rating for Arsenal is -37.7075. In a similar fashion, we calculate the ratings for the other teams. As for the Elo-Point version, we only change the value of *T* by applying equation (3.13). The ratings and rankings generated by Elo-Win and Elo-Point are depicted in Table 3-6 and Table 3-7 respectively.

Table 3-6: Elo-Win rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	-37.7075	12	Leicester	1.1500	6
Bournemouth	37.7075	3	Liverpool	40.0000	1
Brighton	2.2925	5	Man City	37.7075	3
Burnley	-18.8500	9	Man United	-2.2925	8
Cardiff	-20.0000	10	Newcastle	-20.0000	10
Chelsea	37.7075	3	Southampton	-20.0000	10
Crystal Palace	0.0000	7	Tottenham	37.7075	3
Everton	20.0000	4	Watford	38.8500	2
Fulham	-37.7075	12	West Ham	-37.7075	12
Huddersfield	-37.7075	12	Wolves	-21.1500	11

Table 3-7: Elo-Point rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	-11.5924	17	Leicester	6.2302	7
Bournemouth	12.6588	5	Liverpool	23.1415	1
Brighton	-6.3374	14	Man City	19.8464	2
Burnley	-6.0912	13	Man United	0.3374	9
Cardiff	-9.6546	15	Newcastle	-4.3454	12
Chelsea	13.5924	4	Southampton	-4.0000	11
Crystal Palace	0.1919	10	Tottenham	9.8612	6
Everton	4.0000	8	Watford	16.0912	3
Fulham	-15.8612	18	West Ham	-15.9922	19
Huddersfield	-21.8464	20	Wolves	-10.2302	16

3.3.5 Keener Method

This method has been proposed by James P. Keener in 1993 for ranking NCAA Division I-A football teams (Keener, 1993). Keener's method is based on the theory of nonnegative matrices.

Firstly, the Laplace's rule of succession (Ross, 2010, p. 98) is employed to compute a_{ij} element in a game between team *i* and team *j*:

$$a_{ij} = \frac{S_{ij} + 1}{S_{ij} + S_{ji} + 2}, \tag{3.14}$$

where S_{ij} is the points that team *i* scored and S_{ji} is the points scored by team *j*. The reason that Keener uses Laplace's rule of succession ratio is to ensure that even if a team scores 0 points, the other team will not be awarded whole points.

In contrast to Colley's method, Keener's method utilizes game scores. The fact that (3.14) uses points introduces bias into the method, implying that a team can boost its ranking by running up its score in a game. In other words, score points do matter. For this reason, Keener suggested a score smoothing function of (3.15) to minimize the possibility of bias.

$$h(x) = \frac{1}{2} + \frac{1}{2} sgn\left(x - \frac{1}{2}\right)\sqrt{|2x - 1|}\,. \tag{3.15}$$

Summarizing this method, we can form Keener matrix *K* using (3.14) and (3.15):

$$K_{ij} = h\left(a_{ij}\right), \tag{3.16}$$

In case of a different number of games among teams, we normalize each element of the *K* matrix by replacing $K_{ij} = \frac{K_{ij}}{n_i}$, where n_i is the number of games played by team *i*.

We can also select any other statistic than goals (e.g., total shots). The selection of any other statistic of the competition should be related to team strength. It is important to select the correct statistic since Keener's method is not bias-free, which means that all the points that succeeded in a matchup are taken into account for the final rating and ranking.

Finally, we can solve $K \cdot r = \lambda \cdot r$ to get the Perron vector of matrix K, i.e., rating vector *r*. In the linear system given above, λ is the spectral radius (dominant eigenvalue) of K.

The Perron-Frobenius theorem in linear algebra guarantees that a unique rating solution exists if the *K* matrix is nonnegative, irreducible, and primitive. Here, *K* is nonnegative ($K \geq 0$) because all the elements are equal to or greater than zero. Matrix *K* is irreducible if and only if its graph is strongly connected (Meyer, 2000, p. 209). In terms of sports teams, when the games graph is strongly connected this allows for comparison between any pair of teams regardless of whether they have played any game or not. Primitivity is achieved if *K* is nonnegative and irreducible, and has one eigenvalue on its spectral circle (Meyer, 2000, p. 674) or $K^m>0$ for some $m>0$ (Meyer, 2000, p. 678). If *K* is not irreducible one common technique is to perturb the matrix by adding a small

positive value ϵ to each entry (Langville & Meyer, 2012, p. 39). Also, by adding ϵ value to any element of the main diagonal primitivity will be satisfied (Meyer, 2000, p. 678).

❖ Application of the Keener Method to the example of EPL games

The matrix S is the ${V_{TG}}^T$ that is shown later in the GeM method in 3.3.7. By applying (3.16) and then by solving the system $Kr=\lambda r$, λ is the Perron value and the unique Perron vector r becomes the unique rating vector of our example. The ratings and rankings are depicted in Table 3-8.

Table 3-8: Keener rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	0.0472	17	Leicester	0.0507	8
Bournemouth	0.0529	4	Liverpool	0.0538	1
Brighton	0.0492	11	Man City	0.0535	2
Burnley	0.0486	14	Man United	0.0503	9
Cardiff	0.0482	16	Newcastle	0.0489	13
Chelsea	0.0528	5	Southampton	0.0490	12
Crystal Palace	0.0499	10	Tottenham	0.0526	6
Everton	0.0512	7	Watford	0.0533	3
Fulham	0.0468	18	West Ham	0.0467	19
Huddersfield	0.0461	20	Wolves	0.0483	15

3.3.6 Offense - Defense Method

The Offense-Defense method was developed by Anjela Govan (Govan A. Y., 2008; Govan, Langville, & Meyer, 2009). This method is inspired by the popular HITS algorithm (Kleinberg, 1999) used to rank web pages. We refer to the Offense-Defense method as ODM.

Govan separates the offensive and defensive strengths of each team and uses them to calculate its overall rating. Using the score a_{ij} that team j scored against team i, two column vectors, o and d are calculated, signifying the offensive and defensive strength respectively. For a given team i, o and d are calculated as follows:

$$o_i = \sum_{j=1}^{n} \frac{a_{ji}}{d_j}, \quad d_i = \sum_{j=1}^{n} \frac{a_{ij}}{o_j}.$$

The actual implementation demands that the first *d* is initialized as a vector of *1*'s. Then *o* is calculated and the process repeats in order to refine the results. In other words, given that $A_{n\,x\,n} = [a_{ij}]$, on the *k*-th iteration

$$o^{(k)} = A^T \frac{1}{d^{(k-1)}}, \quad d^{(k)} = A \frac{1}{o^{(k)}} .$$

The above equations are equivalent to a row-column scaling of matrix *A.* In order for them to converge, the matrix has to have total support. This is achieved by adding a constant ϵ to each of its elements. Given that ϵ is very small, it does not have any effect on the model. That means that *A* is replaced as follows:

$$P = A + \epsilon e e^T,$$

where *e* is a vector of 1's.

The final rating vector can be generated by combining offensive and defensive lists. The aggregation of two lists can be done easily by component-wise division of *o* and *d*. As it is evident by the way the rating is calculated, high *o* values mean strong offense and low *d* values mean strong defense. The overall rating vector can be written as

$$r = o/d . \tag{3.17}$$

❖ Application of the Offense-Defense Method to the example of EPL games

The matrix *A* is the ${V_{TG}}^T$ that is shown later in the GeM method (3.3.7). In our example, we started with $d^0 = e$ and a tolerance level of 0.0001. After *k* = 47711 iterations, we generate the final vectors *o* and *d*. The rating vector can be found by applying equation (3.17). The ratings and rankings are shown in the table below.

Table 3-9: Offense-Defense rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	1.9343	12	Leicester	1.876E+04	6
Bournemouth	1.313E+06	3	Liverpool	3.404E+12	1
Brighton	3.7124	11	Man City	5.2847	9
Burnley	4.103E+05	4	Man United	6.2235	8
Cardiff	0.0005	14	Newcastle	0.0000	18
Chelsea	4.5766	10	Southampton	0.0000	15
Crystal Palace	1049.8521	7	Tottenham	0.0000	17
Everton	0.0000	16	Watford	1.865E+06	2
Fulham	0.0000	20	West Ham	2.312E+05	5
Huddersfield	0.5584	13	Wolves	0.0000	19

3.3.7 Markov (GeM) Method

This method utilizes finite Markov chains theory and therefore, it is called the Markov Method. It was applied to the sports field by the graduate students Angela Govan and Luke Ingram to successfully rank NFL football and NCAA basketball teams respectively (Ingram, 2007; Govan A. Y., 2008). Markov's method is known as the Generalized Markov (GeM) ranking model and is, indeed, an adjustment of the famous PageRank algorithm (Brin & Page, 1998) that Google uses for webpage ranking. Similarly, to PageRank, GeM uses parts of finite Markov chains and graph theory in order to generate ratings of *n* items in a finite set. The method is not only oriented to sports team ratings but also to any problem that can be modeled as a weighted directed graph (Govan A. Y., 2008).

The main idea behind the Markov Method is voting. In every game between two teams, the weaker team casts a vote for the stronger team. There are many ways for a team to vote for another. The following three ways of voting schemes are illustrated by (Langville & Meyer, 2012, pp. 68-71) and we have utilized them in our examples and applications. The following schemes will be explained by assuming a game between team *i* and team *j* where the P_{ij} is the points (or goals in the case of soccer) that team *i* scores against team *j* and P_{ji} is the points/goals that team *j* scores against team *i*.

(1) Voting with Losses: The simplest method uses wins and losses, implying that a winning team gains a vote from the defeated team. In the case of a tie, both teams cast a half-vote. This can be done by:

$$V_{ij} = \begin{cases} 1 & if\ team\ i \text{ is beaten by team } j \\ 0.5 & tie \\ 0 & otherwise \end{cases}.$$

(2) Losers Vote with point differential: A better voting process would take into account game scores, namely, a winning team gets as many votes by a weaker opponent as the margin of victory in the game between them. This can be done by:

$$V_{ij} = \begin{cases} P_{ji} - P_{ij} & if\ team\ i \text{ is beaten by team } j \\ 0 & otherwise \end{cases}.$$

(3) Winners and Losers vote with points: In order to make the voting method even more advanced both teams should be allowed to cast votes equal to the number of points given up in the game. This can be done by:

$$V_{ij} = \begin{cases} P_{ji} & i \neq j \\ 0 & otherwise \end{cases}.$$

Markov's method has the advantage of combining more than one statistic to generate ratings. The wise selection of statistical information is a crucial part of the method. Thus, in order to get the GeM rating vector r, the voting matrices *V* for each statistic are formed first. Then each voting matrix is transformed to stochastic by dividing each matrix element by the sum of the elements in the corresponding row. Then, G is computed for the p statistics of interest as follows:

$$G = a_0 S_0 + \cdots + a_p S_p, \tag{3.18}$$

where $0 \leq a_i \leq 1$, $\sum_{i=1}^{p} a_i = 1$ and *p* is the total number of statistics.

Each stochastic matrix S_i is called a feature matrix and will be formed using different statistics. As expected, the tuning of the weights a_i plays an important role in the overall rating. Finally, we compute the rating vector r which is the stationary vector or dominant eigenvector of G. Especially, for the last step G is required to be irreducible (and aperiodic which is almost always met) (Langville & Meyer, 2012, p. 73). Irreducible means that the graph of *G* is strongly connected, or equivalently all teams are reachable. Irreducibility is required for the existence and uniqueness of the stationary vector (Meyer, 2000). By employing the adjustment from PageRank, if G is reducible, the irreducible $\bar{G}$ is computed by applying the (3.19):

$$\bar{G} = bG + (1-b)/nE, \qquad 0 < b < 1, \tag{3.19}$$

where *E* is the matrix of all 1's and *n* is the number of teams. If b < 1 then $\bar{G}$ is irreducible and aperiodic (Boldi, Santini, & Vigna, 2005). In terms of sports teams, the damping factor *b* regulates the transitioning probability of moving from undefeated teams. Thus, it helps to ensure that ratings converge and are stable over time. The choice of *b* depends on the application, for example (Brin & Page, 1998) used the value 0.85.

❖ Application of the GeM Method to the example of EPL games

As it is mentioned before, the Markov method has a vital difference from the other methods as allows the use of more than one statistic. Below is demonstrated how the Markov method can take advantage of the statistics *TW* (total wins), *TG*, *TS*, and *TST*. For the *TW* statistic which is the total wins by each team prior to the itinerary match, the first method (Voting with Losses) is applied where a winning team gains a vote from the defeated team. In the case of a tie, both teams cast a half-vote. The third method (Winners and Losers vote with points) is used for the other three statistics, where in every game, each team votes according to the number of points lost by the other team. The following matrices represent V_{TW} and V_{TG}:

$V_{TW} =$

	Arsenal	Bournemouth	Brighton	Burnley	Cardiff	Chelsea	Crystal Palace	Everton	Fulham	Huddersfield	Leicester	Liverpool	Man City	Man United	Newcastle	Southampton	Tottenham	Watford	West Ham	Wolves
Arsenal	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Bournemouth	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brighton	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Burnley	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	1	0	0
Cardiff	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0
Chelsea	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Crystal Palace	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Everton	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5
Fulham	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Huddersfield	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Leicester	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Liverpool	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Man City	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Man United	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Newcastle	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Southampton	0	0	0	0.5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Tottenham	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Watford	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
West Ham	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Wolves	0	0	0	0	0	0	0	0.5	0	0	1	0	0	0	0	0	0	0	0	0

$V_{TG} =$

	Arsenal	Bournemouth	Brighton	Burnley	Cardiff	Chelsea	Crystal Palace	Everton	Fulham	Huddersfield	Leicester	Liverpool	Man City	Man United	Newcastle	Southampton	Tottenham	Watford	West Ham	Wolves
Arsenal	0	0	0	0	0	3	0	0	0	0	0	0	2	0	0	0	0	0	0	0
Bournemouth	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Brighton	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0
Burnley	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
Cardiff	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chelsea	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Crystal Palace	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
Everton	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2
Fulham	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	3	0	0	0
Huddersfield	0	0	0	0	0	3	0	0	0	0	0	0	6	0	0	0	0	0	0	0
Leicester	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
Liverpool	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Man City	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Man United	0	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Newcastle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
Southampton	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Tottenham	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
Watford	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
West Ham	0	2	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
Wolves	0	0	0	0	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0

In a similar fashion, voting matrices V_{TS} and V_{TST} are formed. The next step is to

transform voting matrices to stochastic. S_{TW} is the only one presented below since the rest follow the same logic.

S_{TW} =

	Arsenal	Bournemouth	Brighton	Burnley	Cardiff	Chelsea	Crystal Palace	Everton	Fulham	Huddersfield	Leicester	Liverpool	Man City	Man United	Newcastle	Southampton	Tottenham	Watford	West Ham	Wolves
Arsenal	0	0	0	0	0	0.6	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0
Bournemouth	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Brighton	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0.5	0	0
Burnley	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Cardiff	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chelsea	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Crystal Palace	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Everton	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.33	0	0	0	0.67
Fulham	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0.6	0	0	0
Huddersfield	0	0	0	0	0	0.33	0	0	0	0	0	0	0.67	0	0	0	0	0	0	0
Leicester	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Liverpool	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Man City	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Man United	0	0	0.75	0	0	0	0	0	0	0	0.25	0	0	0	0	0	0	0	0	0
Newcastle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Southampton	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Tottenham	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0	0	0	0
Watford	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
West Ham	0	0.33	0	0	0	0	0	0	0	0	0	0.67	0	0	0	0	0	0	0	0
Wolves	0	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0	0	0	0	0	0	0

After the transformation of voting matrices to stochastic, we set the weights. In our example, for simplicity, equal weights (a_i) are set for each statistic. G matrix derived from equation (3.18): $G = a_{TW}S_{TW} + a_{TG}S_{TG} + a_{TS}S_{TS} + a_{TST}S_{TST}$,

where $a_{TW} = a_{TG} = a_{TS} = a_{TST} = 0.25$.

Finally, the matrix $\bar{G}$ is computed by the formula (3.19) where the value of damping factor b was set to 0.85. $\bar{G}$ depicted as follows:

$\bar{G}$ =

	Arsenal	Bournemouth	Brighton	Burnley	Cardiff	Chelsea	Crystal Palace	Everton	Fulham	Huddersfield	Leicester	Liverpool	Man City	Man United	Newcastle	Southampton	Tottenham	Watford	West Ham	Wolves
Arsenal	0.01	0.01	0.01	0.01	0.01	0.49	0.01	0.01	0.01	0.01	0.01	0.01	0.38	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Bournemouth	0.02	0.02	0.02	0.02	0.15	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.52	0.02
Brighton	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.26	0.01	0.01	0.01	0.6	0.01	0.01
Burnley	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.29	0.01	0.57	0.01	0.01
Cardiff	0.01	0.55	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.31	0.01	0.01	0.01	0.01	0.01
Chelsea	0.56	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.11	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Crystal Palace	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.22	0.01	0.01	0.65	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Everton	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.31	0.01	0.01	0.01	0.56
Fulham	0.01	0.01	0.01	0.01	0.01	0.01	0.36	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.51	0.01	0.01	0.01
Huddersfield	0.01	0.01	0.01	0.01	0.01	0.29	0.01	0.01	0.01	0.01	0.01	0.01	0.57	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Leicester	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.66	0.01	0.01	0.01	0.01	0.01	0.2
Liverpool	0.03	0.03	0.03	0.03	0.03	0.03	0.27	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.22	0.03
Man City	0.31	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.36	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Man United	0.01	0.01	0.54	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.33	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Newcastle	0.01	0.01	0.01	0.01	0.21	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.66	0.01	0.01	0.01
Southampton	0.01	0.01	0.01	0.29	0.01	0.01	0.01	0.57	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Tottenham	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.34	0.02	0.02	0.02	0.02	0.02	0.34	0.02	0.02	0.02	0.02	0.02
Watford	0.02	0.02	0.11	0.56	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
West Ham	0.01	0.35	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.51	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Wolves	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.44	0.01	0.01	0.42	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

The final rating will be the stationary vector or dominant eigenvector of $\bar{G}$. The ratings and rankings are listed in Table 3-10.

Table 3-10: GeM rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	0.0505	11	Leicester	0.0555	7
Bournemouth	0.0391	15	Liverpool	0.0569	6
Brighton	0.0515	10	Man City	0.0483	12
Burnley	0.0716	2	Man United	0.0611	4
Cardiff	0.0241	20	Newcastle	0.0358	18
Chelsea	0.0450	13	Southampton	0.0517	9
Crystal Palace	0.0377	16	Tottenham	0.0531	8
Everton	0.0663	3	Watford	0.0828	1
Fulham	0.0364	17	West Ham	0.0418	14
Huddersfield	0.0322	19	Wolves	0.0588	5

3.4 Rank and Rating Aggregation

In this section, we discuss how multiple rating/ranking lists can be combined into a single list. The methods discussed here are called heuristics which means that the final list may not be optimal (Langville & Meyer, 2012, p. 183). Their key benefit is their simplicity and that they are computationally fast. However, their main disadvantage is that the aggregation result may not reflect the optimal ranking which is the most accurate representation of the relative importance of the items among lists.

3.4.1 Rank Aggregation

Rank aggregation refers to the process where we combine ranking results from several methods into one list. Two well-known rank aggregation methods are explained below:

- Borda Count

  Borda count (Borda, 1784) is a rank aggregation method developed by Jean-Charles de Borda in 1770. Borda count assigns to each item a particular number of points, with the first-ranked item receiving the most points and the last-ranked item receiving the least. Specifically, for *n* items and *m* ranking lists, the first-ranked item in the first list gets *n*-1 points, the second gets *n*-2 points, and we continue until the last ranked item gets 0 points. This process is repeated in *m* ranking lists and then the scores for

each item are accumulated. The item with the highest score is the first-ranked, the second-highest score the second rank, etc. Borda count method is simple and many variations are proposed in the literature, however, one weakness is that the method is vulnerable to manipulation (Nitzan, 1985; Favardin, Lepelley, & Serais, 2002).

- Average Rank

  The average rank method is very simple. The aggregated ranking list of items in this method is defined by their average rank score across all ranking lists. The final ranking list is obtained by ordering the items based on their average ranking scores.

After the explanation of rank aggregation methods, we turn to our illustrative example where the ranking results from the rating systems presented in section 3.3 are aggregated in order to obtain a single ranking list for all teams. The aggregated ranking results are shown in Table 3-11 and as we can observe the two aggregation methods generate the same ranking lists for our illustrative example.

Table 3-11: Rank aggregation results

Team	Borda Count		Avg Rank		Team	Borda Count		Avg Rank	
	Count	#	Rating (avg)	#		Count	Rank	Rating (avg)	#
Arsenal	61	14	15.875	14	Leicester	99	7	11.125	7
Bournemouth	123	3	8.125	3	Liverpool	146	1	5.250	1
Brighton	87	11	12.625	11	Man City	122	4	8.250	4
Burnley	88	10	12.500	10	Man United	98	8	11.250	8
Cardiff	62	13	15.750	13	Newcastle	67	12	15.125	12
Chelsea	111	5	9.625	5	Southampton	67	12	15.125	12
Crystal Palace	98	8	11.250	8	Tottenham	110	6	9.750	6
Everton	96	9	11.500	9	Watford	134	2	6.750	2
Fulham	45	16	17.875	16	West Ham	67	12	15.125	12
Huddersfield	45	16	17.875	16	Wolves	58	15	16.250	15

#: Rank

3.4.2 Rating Aggregation

Rating aggregation is the procedure where multiple rating lists are combined to create a single rating list. Three methods that are suggested in (Langville & Meyer, 2012) and also mentioned in section 2.2 are presented in this subsection. The main issue is to turn ratings into the same scale which can be addressed by the normalization of values.

Firstly, in each method we compute the rating distances of teams and matrix *R* is formed as follows:

$$R_{ij} = \begin{cases} r_i - r_j & if\ r_i > r_j \\ 0 & otherwise \end{cases}, \tag{3.20}$$

where r_i is the rating score of team *i* and r_j is the rating score of team *j*.

Then, matrix *R* is normalized by dividing each element with the sum of all elements as follows:

$$\bar{R}_{ij} = \frac{R_{ij}}{e^T R e}, \tag{3.21}$$

where *e* is a vector of 1's.

Finally, the normalized rating average matrix $\bar{R}_{ave}$ can be computed by the weighted average of the normalized matrices (i.e., one normalized matrix per method) as

$$\bar{R}_{ave} = a_0\bar{R}_0 + \cdots + a_p\bar{R}_p, \tag{3.22}$$

where $0 \le a_i \le 1$, $\sum_{i=1}^{p} a_i = 1$ and *p* is the total number of methods.

The R_{WL} is shown below and the rest matrices are formed by following the same logic.

$R_{WL} =$

	Arsenal	Bournemouth	Brighton	Burnley	Cardiff	Chelsea	Crystal Palace	Everton	Fulham	Huddersfield	Leicester	Liverpool	Man City	Man United	Newcastle	Southampton	Tottenham	Watford	West Ham	Wolves
Arsenal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bournemouth	2	0	1	2	2	0	1	1	2	2	1	0	0	1	2	2	0	0	2	2
Brighton	1	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1
Burnley	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cardiff	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chelsea	2	0	1	2	2	0	1	1	2	2	1	0	0	1	2	2	0	0	2	2
Crystal Palace	1	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1
Everton	1	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1
Fulham	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Huddersfield	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Leicester	1	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1
Liverpool	2	0	1	2	2	0	1	1	2	2	1	0	0	1	2	2	0	0	2	2
Man City	2	0	1	2	2	0	1	1	2	2	1	0	0	1	2	2	0	0	2	2
Man United	1	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1
Newcastle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Southampton	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tottenham	2	0	1	2	2	0	1	1	2	2	1	0	0	1	2	2	0	0	2	2
Watford	2	0	1	2	2	0	1	1	2	2	1	0	0	1	2	2	0	0	2	2
West Ham	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wolves	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

After the normalization of all matrices, $\bar{R}_{ave}$ is computed by applying equal weights as follows:

$$\bar{R}_{ave} = 0.125\bar{R}_{WL} + 0.125\bar{R}_{Colley} + 0.125\bar{R}_{Massey} + 0.125\,\bar{R}_{Elo-win} + 0.125\bar{R}_{Elo-point} + 0.125\bar{R}_{Keener} + 0.125\bar{R}_{OD} + 0.125\bar{R}_{GeM}$$

Next, $\bar{R}_{ave}$ matrix can be used in the following methods to generate the aggregated rating list.

- Method Perron:

  The Perron-Frobenius theorem in linear algebra has been applied in several significant applications in various fields. As discussed previously, Keener's method (subsection 3.3.5) is based on the Perron theorem. The theorem can also be used to aggregate multiple rating lists. In this case, the $\bar{R}_{ave}$ matrix is nonnegative and as we explained in Keener's method, we can force the irreducibility and primitivity of $\bar{R}_{ave}$ matrix. The final rating vector *r* can be computed as

  $$\bar{R}_{ave} \cdot r = \lambda \cdot r, \qquad (3.23)$$

  where λ is the spectral radius (dominant eigenvalue) of $\bar{R}_{ave}$.

- Method ODM (Offense-Defense):

  In a similar manner as presented in the Offense-Defense method (subsection 3.3.6), the formula *r=o/d* is used in order to compute the final aggregated ratings where *o* and *d* are the offensive and defensive ratings respectively. Considering that offensive is represented by row sums while defensive is represented by column sums this can be done as follows:

  $$o = \bar{R}_{ave} \cdot e, \text{ and } d = e^T \cdot \bar{R}_{ave},$$

  where *e* is a vector of 1's.

- Method Markov:

  The aggregation technique here is performed based on the Markov method presented in subsection 3.3.7. The voting matrix is not required to be formed since the $\bar{R}_{ave}$ has the role of the voting matrix. Then, $\bar{R}_{ave}$ is transformed into a stochastic matrix and after that, all the subsequent steps in the Markov method remain the same. Also, to address the issue of irreducibility the formula (3.19) that uses the damping factor *b* can be applied.

The aggregated rating lists after applying the rating aggregation methods in our illustrative example (rating lists from section 3.3) are depicted in Table 3-12.

Table 3-12: Rating aggregation results

Team	Perron		ODM		Markov	
	Rating	Rank	Rating	Rank	Rating	Rank
Arsenal	0.0161	14	0.1838	14	0.0204	13
Bournemouth	0.0705	5	4.4494	6	0.0479	7
Brighton	0.0211	12	0.6233	11	0.0177	18
Burnley	0.0327	8	0.3756	12	0.0759	3
Cardiff	0.0142	16	0.1762	16	0.0191	16
Chelsea	0.0675	6	4.9180	5	0.0450	8
Crystal Palace	0.0319	9	0.9288	8	0.0289	10
Everton	0.0472	7	1.4095	7	0.0636	4
Fulham	0.0118	18	0.1037	19	0.0180	17
Huddersfield	0.0074	20	0.0561	20	0.0138	19
Leicester	0.0272	11	0.8590	9	0.0211	12
Liverpool	0.2997	1	99.1005	1	0.2453	1
Man City	0.0823	3	7.1606	2	0.0564	5
Man United	0.0310	10	0.8572	10	0.0378	9
Newcastle	0.0165	13	0.2698	13	0.0203	14
Southampton	0.0093	19	0.1752	17	0.0123	20
Tottenham	0.0712	4	6.3152	3	0.0482	6
Watford	0.1142	2	6.1807	4	0.1662	2
West Ham	0.0143	15	0.1434	18	0.0198	15
Wolves	0.0138	17	0.1837	15	0.0222	11

As we observe, there are differences in final ranking lists. This could be attributed to the limited number of games that rating systems used and as a consequence their results also affect the aggregation methods.

3.5 Outcome Probability and Predictions

The outcome prediction of a paired comparison is an interesting topic that received the attention of many researchers. Utilizing the results (ratings/rankings) from rating methods can be beneficial for this purpose and will also further enhance their usefulness. A simple way to predict the outcome of a sports game is by assuming that the

team with a better ranking position would win. We refer to this approach by the name "RANK" because predictions are based on rankings. This is an easy way to apply them in sports or other applications where ties do not frequently happen such as NFL, NBA, and NCAA. However, when ties occur more regularly this is a drawback to this approach because in order to predict efficiently draws both of the competing teams should have the same rating scores. Due to the fact that rating systems focus on determining which team is better equal rating scores are uncommon to happen. This suggests that predicting soccer outcomes in this way will not be efficient in terms of ties since soccer is a low-scoring sport and ties happen to a higher degree.

Another important topic is the probability associated with the pairwise comparison outcome which is valuable information that can be used to predict the outcome. The famous model by Bradley and Terry was developed in 1952 (Bradley & Terry, 1952) and aims to describe the probability of a pairwise outcome. Their model can also be used in the case of comparing two teams competing in a game where the probability of team *i* beating team *j* is described by the following formula:

$$\mathbb{P}(i > j) = \frac{\pi_i}{\pi_i + \pi_j},$$

where π_i, π_j is the overall strength of team *i* and team *j* respectively and the symbol > means that team *i* beats team *j*.

In simpler terms, the probability of team *i* winning the game against team *j* is directly related to the ratio of their strengths. To estimate the relative strength of $\pi_1, \pi_2, \ldots, \pi_n$ the Maximum Likelihood Estimation (MLE) can be used. This means that the teams' strengths maximize the likelihood of observed outcomes. Other variations of the model that include ties are proposed in other studies such as (Rao & Kupper, 1967; Davidson, 1970). Specifically, in 1970 Davidson (Davidson, 1970) proposed the following variation of the Bradley-Terry model in order to take into account ties:

$$\begin{aligned}
\mathbb{P}(i > j) &= \frac{\pi_i}{\pi_i + \pi_j + \nu\sqrt{\pi_i \cdot \pi_j}}, \\
\mathbb{P}(i < j) &= \frac{\pi_j}{\pi_i + \pi_j + \nu\sqrt{\pi_i \cdot \pi_j}}, \\
\mathbb{P}(i = j) &= \frac{\nu\sqrt{\pi_i \cdot \pi_j}}{\pi_i + \pi_j + \nu\sqrt{\pi_i \cdot \pi_j}},
\end{aligned} \tag{3.24}$$

where $v \geq 0$ is a parameter related to tie and varies depending on the specific problem and application.

Introducing the outcome probabilities into the prediction process can be utilized in a more useful way than using only the ranking positions of teams. Also, the use of probabilities is important for the risk management process that is integrated into various applications. In the context of soccer, the ratings can be turned into predictions by considering probabilities from outcomes. The probabilities can be computed from rating scores by applying the modified logistic function which is utilized by J. Lasek et al. (Lasek, Szlávik, & Bhulai, 2013). The authors relied on the logistic function that is used in many applications and aims to calculate the probability of player *i* winning a game against player *j*:

$$\mathbb{P}(i > j) = \frac{1}{1 + e^{-a(r_i - r_j)}} = \frac{e^{a(r_i - r_j)}}{1 + e^{a(r_i - r_j)}}, \quad (3.25)$$

where r_i, r_j are the rating of player *i* and player *j*, and *a* is an appropriate scaling factor.

For the case of soccer teams, the authors modified (3.25) and also incorporated the home-field advantage. The probabilities of team *i* winning or losing a game against team *j* are shown in (3.26) and (3.27) respectively.

$$\mathbb{P}(i > j) = \frac{\left(e^{a(r_i - r_j) + h \cdot \mathrm{t}}\right)^s}{1 + e^{a(r_i - r_j) + h \cdot \mathrm{t}}} = \frac{e^{a(r_i - r_j) + h \cdot \mathrm{t}}}{1 + e^{a(r_i - r_j) + h \cdot \mathrm{t}}}, \mathrm{s} = 1, \quad (3.26)$$

$$\mathbb{P}(i < j) = \frac{\left(e^{a(r_i - r_j) + h \cdot \mathrm{t}}\right)^s}{1 + e^{a(r_i - r_j) + h \cdot \mathrm{t}}} = \frac{1}{1 + e^{a(r_i - r_j) + h \cdot \mathrm{t}}}, \mathrm{s} = 0, \quad (3.27)$$

where *i* is the home team, *j* is the away team, r_i, r_j are the ratings of team *i* and team *j* respectively, *a* is an appropriate scaling factor, *h* is the home-field advantage, *s* is a binary variable for the game outcome (1 for win and 0 for loss), and *t* is the home-field indicator function.

Specifically, t = 𝕀[game played in home team's court] is an indicator function that takes on the value 1 when *i* and *j* teams are playing in the home team's court and 0 otherwise. This adaption was made in order to not consider the home-field advantage if the two teams play in a neutral field. Moreover, we can see that $\mathbb{P}(i > j) + \mathbb{P}(i < j) = 1$.

Also, J. Lasek et al. adopt the idea of Glickman (Glickman, 1999) for the computation of the Draw probability. In particular, the Draw probability can be modeled by assuming that is equal to the probability of two independent games between the two teams where in the first game team *i* wins team *j*, and in the second game team *i* is beaten by team *j*. Equally, a tie can be considered a half-win and a half-loss. Thus, the formula

for calculating the tie probability for a game outcome between team *i* and team *j* is the following:

$$\mathbb{P}(i=j) = \mathbb{P}(i>j)^{0.5} \cdot \mathbb{P}(i<j)^{0.5} = \frac{\sqrt{e^{a(r_i-r_j)+h\cdot \mathrm{t}}}}{1+\ e^{a(r_i-r_j)+h\cdot \mathrm{t}}}. \qquad (3.28)$$

From (3.28) the computation is performed by taking the square root of the product of the win probability of team *i* by the probability of losing. The fact that three outcome probabilities must sum to 1, the following constraint must be satisfied:

$$\mathbb{P}(i>\ j) + \mathbb{P}(i<\ j) + \mathbb{P}(i=j) = 1.$$

Therefore, the outcome probability function can be modified as follows in order to include ties is

$$\mathbb{P}\left(s \middle|\, r_i, r_j\right) = \frac{d^s}{1+d+\sqrt{d}}, \qquad d\ =\ e^{a(r_i-r_j)+h\cdot \mathrm{t}}. \qquad (3.29)$$

In order to include ties *s* must be set to 0.5. Thus, *s* is a ternary variable taking values 0, 0.5, and 1, if team *i* lost, tied, or won, respectively. The likelihood function can be written as the product of probabilities of each observed outcome which depends on the ratings of the teams involved:

$$\mathcal{L}(a,h) = \prod_{k=1}^{n} \mathbb{P}\left(s_k \middle|\, r_{ik}, r_{jk}\right), \qquad (3.30)$$

where *n* is the number of games, s_k is the observed outcome of the *k*-th game, r_{ik} and r_{jk} is the rating of team *i* and *j* respectively before the game *k*.

Then, the parameters *a* and *h* are determined by maximizing the likelihood function. As we can observe, the (3.29) equation is the Davidson model with *v*=1. Szczecinski and Djebbi also noted this observation in their study (Szczecinski & Djebbi, 2020). In the case we involve *v* in (3.29) and (3.30) then *v* will be determined by maximizing the likelihood function.

We have given the acronym “MLE” for this method since the required parameters for the probabilities are set to their Maximum Likelihood Estimation. In summary, this model computes the probabilities of soccer potential outcomes by including ties and also taking into account the home-field advantage.

3.6 Methods of Comparison

The methods of comparison are metrics for determining the quality of each ranking list. Several ways have been proposed in the literature to find the quality of the ranking lists, nevertheless, it is difficult to answer the question "Which method generates the best ranking list?", as it depends on the criterion, we evaluate the ranking lists.

Below we present some methods of comparing rating and ranking lists that will be used in subsequent chapters. First, the correlation between two ranking lists with Kendall's tau rank correlation coefficient is analyzed. Then, metrics and selection criteria for the lists' evaluation are presented.

3.6.1 Kendall's tau

Kendall's tau is a correlation measure developed by Maurice Kendall in 1938 (Kendall, 1938). It quantifies the agreement of two ranking lists and it is a non-parametric measure which means that does not require any assumption about the underlying distribution of the data. There is a version for full lists and partial lists (Langville & Meyer, 2012, p. 205). We will explain the version for full lists.

Kendall's correlation coefficient τ (tau), gives the degree to which one list agrees (or disagrees) with another and is computed as

$$\tau = \frac{n_c - n_d}{\frac{n(n-1)}{2}}, \tag{3.31}$$

where *n* is the number of items in the lists, n_c is the number of concordant pairs and n_d is the number of discordant pairs.

Kendall's tau value varies between -1 and 1, i.e., $-1 \leq \tau \leq 1$. If $\tau = 1$, then the two lists are in perfect agreement, while if $\tau = -1$, the two lists are totally opposite to each other.

Although this metric is very popular and finds wide applicability in various sectors, there are two disadvantages to mention. First, it is computationally expensive for very large lists. Second, the disagreements between the top of the lists have the same penalty as the bottom (Langville & Meyer, 2012, p. 206). However, in applications such as the sports team rankings the disagreements at the bottom of the lists are less important because they do not indicate the winner of a tournament.

In this thesis, we use the tau-b (Kendall M. G., 1945) which is a modified version of (3.31) and it makes adjustments for ties. The formula for two ranking lists *X, Y* is given below:

$$\tau = \frac{n_c - n_d}{\sqrt{(n_c + n_d + T_x) \cdot (n_c + n_d + T_y)}},$$

where T_x is the number of pairs tied on ranking list *X* only, and T_y is the number of pairs tied on ranking list *Y* only.

3.6.2 Other Metrics and Criteria

Although many systems have been developed to rate and rank items, using the appropriate system for a given application is of great importance. Also, a careful interpretation of the system's result is of paramount importance. For example, PageRank (Brin & Page, 1998) is efficient in ranking webpages, and so is the Elo system (Elo, 1978) to rank chess players. The other way around is not guaranteed to work. Comparison metrics of ranking lists are of paramount importance since they allow us to examine whether a ranking system can effectively respond to an application.

Sometimes the utilization of a rating system is intended for an indirect objective. A typical example is the utilization of team ratings in the prediction of game outcomes where the objective is to achieve a high prediction accuracy. This topic will also be discussed in subsection 4.3.5, however, some cases are provided here in order to facilitate comprehension. In the case of a movie ranking application, the selection criterion for a ranking system can be made after evaluating the future predictions in movie trends. In many cases, a ranking list can contribute to making the best decision in terms of profitability while in other cases in terms of cost efficiency. In terms of profitability, as an example, assume that an individual is willing to invest in the financial market by trading popular stocks based on their rankings. The comparison of rating systems in this example can be made by considering the profitability of the final trading strategy while all the other factors are fixed (e.g., risk management, etc.). In terms of cost-efficiency, a classic illustration is the car selection problem where the rankings of candidate cars can contribute to selecting a car with economical maintenance and low fuel consumption in the long run. However, these application examples might also need additional techniques and methods from other fields in order to be more comprehensive.

Obviously, in many cases, the comparison and evaluation of a ranking system are affected by the purpose and preferences of the stakeholder. In the context of sports team rankings, the rating systems can be utilized not only to forecast the outcome of a game but also for investment opportunities in betting markets. While a sports rating system

may be suitable for determining the final ranking, its efficiency in determining the outcome of future matches is not certain. Thus, sophisticated methods and more complex procedures are required for this goal.

Below some criteria and metrics are presented for the evaluation and comparison of rating methods related to specific purposes. Examples from various applications are also provided.

❖ Predictions Purposes

If we aim to use a rating system and the rating values produced are utilized to make predictions, accuracy is a popular metric for evaluation. Several other metrics that come from the machine learning field such as precision, recall, and F1-score are also important and will be analyzed in section 5.3. Also, RMSE and MAE are useful metrics that will be discussed in subsection 7.4.2.

- Hindsight Accuracy

  Hindsight prediction refers to predicting past games using the ratings of entire season games. The ratio of correct predictions to total games is called hindsight accuracy.

  Application examples:

  (1) Sports: At the end of each season, the final ratings and rankings of teams can be used to predict all the games of the season, thus seeing how well the model fits the available data.

  (2) Evaluation of a new ranking system: The assumption is that a rating system that ranks items effectively will also predict at least well enough the outcomes that were used to generate rankings.

- Foresight Accuracy

  Foresight prediction refers to predicting upcoming games using the ratings of previous weeks. The ratio of correct predictions to total games, we call it foresight accuracy. Foresight accuracy is usually lower than hindsight accuracy.

  Application examples:

  (1) Sports: The current ratings of teams are used in order to make predictions for future games.

  (2) Elections: The current rating scores of candidates are used in order to predict the winner of the elections.

❖ Investment Purposes

It is defined as the benefit that a rating system can provide if its results are utilized for investment decisions. The metrics and criteria may differ in each problem. Examples of metrics are the Return on Investment (ROI), Sharpe ratio, etc.

Application Examples:

(1) Sports: The ratings of sports teams are utilized for predictions of outcomes and then exploiting betting opportunities.

(2) Financial Markets (or other types of alternative markets): A rating system can be used to rate and rank portfolios, trading strategies, or financial instruments for trading (e.g., stocks). Other types of markets such as the real estate market for property selection or alternative markets can also be considered such as the domain name market for domain name selection.

3.7 Comparison Results for the Illustrative Example

The table below compares the ranking lists generated by the rating methods for our example. The lower diagonal elements represent Kendall's tau values of each pair, while the upper diagonal elements are the p-values of each pair from the two-sided hypothesis test, whose null hypothesis is an absence of association.

Table 3-13: Kendall's tau values and p-values for the illustrative example

	WL	**Colley**	**Massey**	**Elo_{win}**	**Elo_{point}**	**Keener**	**ODM**	**GeM**
WL	1.000	5.9E-06	0.063	4.9E-06	1.6E-05	6.1E-06	0.039	0.194
Colley	0.831	1.000	0.111	7.1E-08	6.3E-06	1.8E-06	0.035	0.074
Massey	0.339	0.260	1.000	0.140	0.186	0.186	0.186	0.047
Elo_{win}	0.859	0.907	0.247	1.000	5.2E-07	1.8E-07	0.028	0.082
Elo_{point}	0.786	0.737	0.221	0.839	1.000	1.4E-13	0.034	0.064
Keener	0.824	0.780	0.221	0.872	0.937	1.000	0.034	0.064
ODM	0.377	0.345	0.221	0.367	0.347	0.347	1.000	0.288
GeM	0.236	0.292	-0.326	0.291	0.305	0.305	0.179	1.000

Due to the small amount of data used this affects the final ranking lists so it is quite difficult to draw any general conclusions. From Table 3-13, we notice that ranking lists produced by methods Win-Loss, Colley, Elo-Win, Elo-Point, and Keener are very similar because *tau* correlation coefficients are significantly differentiated from zero.

Also, all p-values are less than 0.05 (<0.05), therefore we reject the null hypothesis (*tau*=0). Moreover, for each pair that contains the GeM or Massey ranking list, we observe that p-values are higher than 0.05 (≥0.05). This indicates that we do not reject the absence of association. This can be explained by the fact that Massey and GeM may require more games to generate stable ranking results. Moreover, the ranking list by the GeM method is differentiated due to the use of more than one statistic that is provided. Further research for their ranking correlation has been conducted in section 4.4.

Then we compare ranking lists by their ability to predict the outcome of games. Hindsight accuracy and foresight accuracy are examined. The foresight predictions will be made on future games. In particular, the ratings of teams are based on the first two weeks of our example and then predictions are made in the games of the third week. Therefore, the total games of hindsight predictions are 20 (first two match weeks of the 2018-2019 EPL season presented in Table 3-1) while the total games of foresight predictions are 10 (3rd match week of the 2018-2019 EPL season presented in Table 3-14).

Table 3-14: The 3rd match week of the English Premier League 2018-2019 season

Date	**Home Team**	**Away Team**	**Home Goals**	**Away Goals**	**FT**
		- 3rd match week -			
25/8/2018	Arsenal	West Ham	3	1	H
25/8/2018	Bournemouth	Everton	2	2	D
25/8/2018	Huddersfield	Cardiff	0	0	D
25/8/2018	Liverpool	Brighton	1	0	H
25/8/2018	Southampton	Leicester	1	2	A
25/8/2018	Wolves	Man City	1	1	D
26/8/2018	Fulham	Burnley	4	2	H
26/8/2018	Newcastle	Chelsea	1	2	A
26/8/2018	Watford	Crystal Palace	2	1	H
27/8/2018	Man United	Tottenham	0	3	A
25/8/2018	Arsenal	West Ham	3	1	H

The "RANK" and "MLE" approaches presented in section 3.5 are utilized for predictions (hindsight and foresight). In order to compute the parameters from (3.30) in

the MLE, the teams' ratings of at least one match week are required. In this example, for foresight predictions, the first two match weeks will be used to predict the third week. For this reason, for the second week, it is required to compute the ratings based on the observed outcomes of the first week. However, this was possible for all methods except Massey. Particularly, in our example due to the limited amount of data in the first week (only 10 games), the Massey method was unable to generate ratings of teams. Hence, for the Massey method foresight accuracy is not available in the case of the MLE method. Nevertheless, in hindsight results, it is possible to start rating teams from the second week when the number of games in the first week is not enough. Table 3-15 provides the hindsight and foresight accuracy results from predictions for each method.

Table 3-15: Hindsight and Foresight prediction accuracy

Rating Method	Hindsight Accuracy		Hindsight Correct Games		Foresight Accuracy		Foresight Correct Games	
	RANK	MLE	RANK	MLE	RANK	MLE	RANK	MLE
WL	0.85	0.85	17	17	0.6	0.7	6	7
Colley	0.85	0.85	17	17	0.5	0.5	5	5
Massey	0.85	0.85	17	17	0.4	-	4	-
Elo_{win}	0.85	0.85	17	17	0.5	0.5	5	5
Elo_{point}	0.75	0.85	15	17	0.6	0.5	6	5
Keener	0.75	0.85	15	17	0.6	0.5	6	5
ODM	0.7	0.55	14	11	0.4	0.4	4	4
GeM	0.6	0.6	12	12	0.5	0.4	5	4

As we mentioned before there are few games in our example, as a result, we cannot conclude about the accuracy rates. However, the results shown in Table 3-15 paralleled the theoretically expected results. Thus, we notice that hindsight accuracy is higher than foresight for each method. Finally, both methods performed similarly in terms of hindsight and foresight accuracy. Further research for prediction accuracy has been conducted in Chapter 6.

3.8 Conclusions

In this chapter, seven different rating methods and the theory behind them are described. Additionally, two rank aggregation methods and three rating aggregation

methods are outlined. Those rating methods are applied to rate and rank soccer teams to an illustrative example that consists of the first 20 games (the two first match weeks) of the EPL in the 2018-2019 season. After generating rating and ranking lists by each method, we compared them in terms of rank correlation. Additionally, we compared them for their ability to predict the outcome of the soccer games, and for their evaluation, we considered their hindsight and foresight accuracy. Particularly, the foresight accuracy is calculated from the predictions made on the games obtained from the 3rd match week of the EPL in the 2018-2019 season. The predictions were generated by two different methods. The first takes into account the ranking positions of teams while the second calculates the probabilities of outcomes based on ratings of teams and then selects the outcome with the highest probability.

In general, it is important to gather a sufficient number of games (many studies suggest more than 30 games) to produce better rating lists and draw conclusions. Also, a common conclusion is that the selection of statistics and the tuning of parameters for some rating systems are very important as they significantly affect the quality of the results.

Finally, we note that, depending on the application, some rating methods work better than others. Although most of the methods analyzed in this chapter have been initially proposed to rate teams or players in sports such as the NCAA and NFL, they can be applied successfully to rate and rank soccer teams.

4 - Proposed Rating and Ranking Systems

4.1 Introduction

This chapter introduces our two proposed rating systems. The first deals with the rating and ranking of soccer teams by taking into account the outcomes of games, the margin of victory, and the shooting accuracy of teams. The second method is more generalized and applicable in various fields for rating/ranking and consequently, comparing and selecting from a set of alternatives/items. A detailed description of the two methods with examples is presented in sections 4.2 and 4.3. Comparisons with the other systems are also made in section 4.4. The last section summarizes the conclusions.

4.2 The AccuRATE Method for Rating and Ranking Soccer Teams

4.2.1 Introduction

In this section, we present a novel rating system to rate and rank soccer teams. We call our system AccuRATE (Accuracy + Rate) triggered by its functionality to use the shooting accuracy of the teams in order to rate and rank them. This rating system is based on our paper (Kyriakides, Talattinis, & Stephanides, 2017) with small modifications and some extensions such as the perfect season example and sensitivity analysis.

Rating a team based on the outcome of a game even an advantageous measure is considered to be more quantitative than qualitative approach. For example, assuming that a team scores seven goals after achieving 10 shots on target, whereas another team scores 1 goal with 5 shots on target, then apparently in the second case, the efficiency rate is higher. Moving a step forward, we could also examine the ratio of the total shots on target (*TST*) to the total number of shots (*TS*) for every team along with the net difference in the final score (goals), i.e., margin of victory, of every game.

Moreover, several research studies have shown that more shots on target have a positive effect on the outcome of a soccer game. Szwarc (Szwarc, 2004) proved that TST is one of the main factors that make the difference between successful and unsuccessful teams in the game of finalists of the 2002 WorldCup. Rampinini et al (Rampinini, Impellizzeri, Castagna, Coutts, & Wisløff, 2009) worked on 416 individual games of the Italian Serie A league and they concluded that TS and TST were higher in the group of

more successful teams compared to those belonging to the less successful group. Lago et al in their study (Lago-Peñas, Lago-Ballesteros, Dellal, & Gómez, 2010) showed that the winning teams in the Spanish Men's Professional League of 2008/09 season had significantly higher TST and TS. Castellano et al (Castellano, Casamichana, & Lago, 2012) conducted a statistical analysis in 177 games of three Word Cup Tournaments (2002, 2006, and 2010) where they revealed that TST and TS are the only statistics that discriminated between successful and unsuccessful teams. Also, Liu et al (Liu, Hopkins, & Gómez, 2016) they have analyzed the games of season 2012/13 in the Spanish First Division Professional Football League, where they found a positive effect of TST and TS in the match outcome.

In the present section, we first introduce the AccuRATE rating system, then we apply the method in our illustrative example that comes from the previous chapter, next the distribution of ratings for EPL is examined, and after that the perfect season example is presented and a sensitivity analysis is carried out. Finally, some conclusions are drawn.

4.2.2 AccuRATE Rating System

Considering the above information, we propose the formulation of a system that uses the mentioned data as input to rate teams. Specifically, for a given match between two teams, *i*, *j*, and a final score of S_i, S_j the new rating of team *i*, r_i' is computed as

$$r_i' = \begin{cases} r_i + d^k, & \text{if team } i \text{ beats team } j \\ r_i - d^{1-k}, & \text{if team } j \text{ beats team } i \end{cases},$$

$$d = |S_i - S_j| \text{ and } k = \frac{TST_i}{TS_i},$$

where TST_i is the total shots on target that team *i* scored, TS_i is the total shots that team *i* scored against *j*, *d* is the net difference of the score achieved by the two teams (i.e., the margin of victory), and *k* represents the shooting accuracy of the team or in other words, the quality of the shots.

As the total shots on target cannot be more than total shots, *k* is always less or equal to 1 and greater or equal to 0 which means that the total net difference in score is reduced as the shooting accuracy decreases. The variable *k* punishes teams when the number of shots on target is low compared to the total shots. Also, if the difference $d = 1$ (one goal) then the winning team will gain 1 point while the losing team will lose 1 point because $d^1=d$. This signifies that for each team that wins or loses a game, its rating will be increased or reduced by at least 1 point respectively.

For better understanding, suppose a soccer league where only two teams participate, the team “Good” and the team “Better” and their initial rating scores are 0. Team “Good” scored 1 goal, with 6 shots of which 2 shots landed on target and 1 of them resulted in a goal (the rest 4 resulted in corners). Team “Better” scored 3 goals, with 4 shots, of which 3 landed on target and only 1 resulted in a corner. Below are the vectors for scores (*S*), total shots (*TS*), and total shots on target (*TST*).

$$S = \begin{matrix}\text{Good} & \text{Better}\\ (\ 1 & 3\ )\end{matrix} \qquad TS = \begin{matrix}\text{Good} & \text{Better}\\ (\ 6 & 4\ )\end{matrix} \qquad TST = \begin{matrix}\text{Good} & \text{Better}\\ (\ 2 & 3\ )\end{matrix}$$

The rating for each team will be:

$$r'_{Good} = r_{Good} - (|1-3|)^{1-\frac{2}{6}} = -1.59$$

$$r'_{Better} = r_{Better} + (|3-1|)^{\frac{3}{4}} = 1.68$$

The extracted values of the rating metric indicate that:

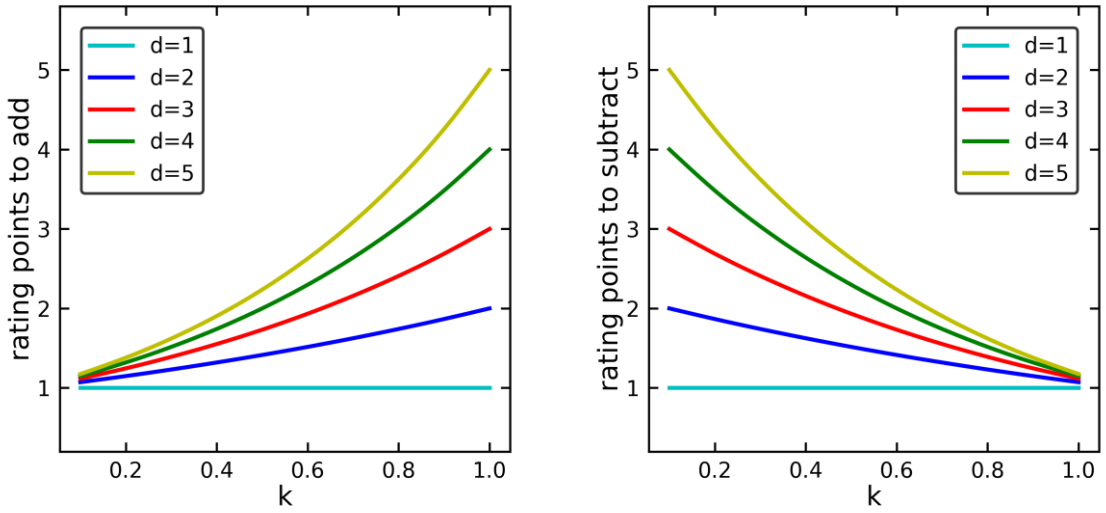
1. The advantage of strong teams to boost their ranking only by goals in the games with weak teams is now limited since they must also achieve good shooting accuracy.
2. The team that wins or loses the game with more than 1 goal difference then the total shots on target and total shots achieved would affect its total rating positively or negatively respectively.
3. Teams are rewarded or penalized for a game by assessing the significance of the outcome, which includes the margin of victory and shooting accuracy. Also, teams take great advantage of scoring a goal. Regardless of the conditions that are exploited to achieve a goal (i.e., great cooperation within the team, a highly skilled player, etc.), the team is rewarded with more points.
4. The offensive opportunities are evaluated. For example, if team A has a lot of shots against B but not many shots on target, team’s A offense may be weak since it has many opportunities but does not take advantage of them.

The points added or subtracted to the team in the case of winning or losing the game respectively, are presented in Table 4-1 for various values of *k* and *d*. Figure 4-1 shows the graphical representation of rating points and as is evident the points are increased (Figure 4-1 a) or decreased (Figure 4-1 b) exponentially in the case of win or loss respectively.

Table 4-1: Points added/subtracted to r_i for various *d* and *k* values

win	lose	rating points to add/subtract[1]				
k	***1-k***	***d*=1**	***d*=2**	***d*=3**	***d*=4**	***d*=5**
0.1	0.9	1	1.0718	1.1161	1.1487	1.1746
0.2	0.8	1	1.1487	1.2457	1.3195	1.3797
0.3	0.7	1	1.2311	1.3904	1.5157	1.6207
0.4	0.6	1	1.3195	1.5518	1.7411	1.9037
0.5	0.5	1	1.4142	1.7321	2.0000	2.2361
0.6	0.4	1	1.5157	1.9332	2.2974	2.6265
0.7	0.3	1	1.6245	2.1577	2.6390	3.0852
0.8	0.2	1	1.7411	2.4082	3.0314	3.6239
0.9	0.1	1	1.8661	2.6879	3.4822	4.2567
1	0	1	2	3	4	5

1: when the team loses a game the points are subtracted



a: team wins a game (points to add) **b: team loses a game (points to subtract)**

Figure 4-1: Points added/subtracted to r_i for various values of *k* and *d*

4.2.3 Illustrative Example

In order to illustrate the method, we have applied the AccuRATE method to our example in section 3.2. The steps to calculate Arsenal's rating are shown below:

1st match week: Arsenal – Man City, Final Outcome: Away-win, Score: 0-2

$$r_{Arsenal} = 0$$

$$r'_{Arsenal} = r_{Arsenal} - (|0-2|)^{1-\frac{3}{9}} = -1.5874$$

2nd match week: Chelsea – Arsenal, Final Outcome: Home-win, Score: 3-2

$$r_{Arsenal} = -1.5874$$

$$r'_{Arsenal} = r_{Arsenal} - (|2 - 3|)^{1-\frac{6}{15}} = -2.5874$$

The final rating for Arsenal is -2.5874. In a similar fashion, we calculate the ratings for the other teams. The final rating and ranking results generated by AccuRATE for the illustrative example of section 3.2 appear in the following table:

Table 4-2: AccuRATE rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	-2.5874	15	Leicester	0.2599	8
Bournemouth	2.2599	6	Liverpool	3.1486	2
Brighton	-1.0000	11	Man City	3.4078	1
Burnley	-1.5422	12	Man United	0.0000	10
Cardiff	-1.8661	14	Newcastle	-1.0000	11
Chelsea	2.4022	4	Southampton	-1.0000	11
Crystal Palace	0.1843	9	Tottenham	2.3566	5
Everton	1.0000	7	Watford	2.7875	3
Fulham	-3.1402	16	West Ham	-3.2974	17
Huddersfield	-6.1219	18	Wolves	-1.6555	13

To compare the correlation of AccuRATE's ranking list with the lists from the other methods, we use Kendall's tau distance. In Table 4-3 *tau* values show that AccuRATE has a high correlation with Keener, Elo-Point, Elo-Win, Win-Loss, and Colley. Also, based on the p-values (<0.001) of those pairs we reject the null hypothesis (*tau*=0). However, in Massey, GeM, and ODM comparisons there are low *tau* values possibly due to the small number of games. The ranking correlation of all pairs has been further examined in section 4.4.

Table 4-3: Kendall's tau and p-values for AccuRATE

	WL	Colley	Massey	Elo_{win}	Elo_{point}	Keener	ODM	GeM
tau	0.818	0.738	0.207	0.868	0.928	0.950	0.313	0.302
p-value	8.2E-06	7.0E-06	0.205	2.5E-07	1.3E-08	5.9E-09	0.055	0.064

Hindsight and foresight accuracy are listed in Table 4-4. In hindsight prediction, we predict the past games and we aim to find the maximum prediction accuracy achieved

when all information about the games is available, while in foresight we predict future game outcomes (in our example we use the 3rd week of the EPL 2018-2019 as future games). As mentioned in section 3.5, in the Rank-based method the prediction is the team receiving a higher rating (or draw if ratings of teams are equal) while in the MLE method, the probabilities from ratings are utilized (i.e., the outcome with the highest probability is selected). As expected, the results confirm that foresight accuracy is lower than hindsight. Due to the small example that we have used, it is difficult to draw any conclusion. Further research for prediction accuracy has been conducted in Chapter 6.

Table 4-4: Hindsight and Foresight prediction accuracy of AccuRATE

Rating Method	Hindsight Accuracy		Hindsight Correct Games		Foresight Accuracy		Foresight Correct Games	
	RANK	MLE	RANK	MLE	RANK	MLE	RANK	MLE
AccuRATE	0.75	0.85	15	17	0.6	0.4	6	4

4.2.4 Ratings Distribution for the English Premier League

To provide an analysis of ratings generated by AccuRATE we examine their distribution in the EPL during the seasons 2005/2006 to 2017/2018. The ratings are computed separately for each season on a weekly basis and they are started from zero in the first week of each season. Figure 4-2 shows the distribution of ratings and Figure 4-3 shows the cumulative distribution for the same period.

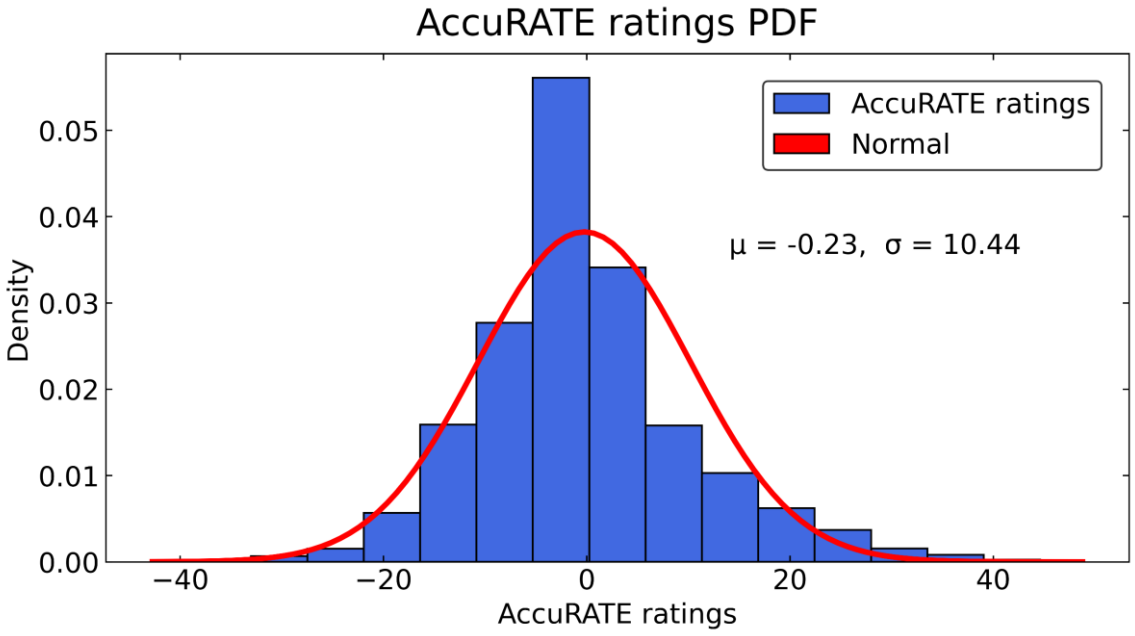


Figure 4-2: AccuRATE's distribution (EPL 2005/06-2017/18)

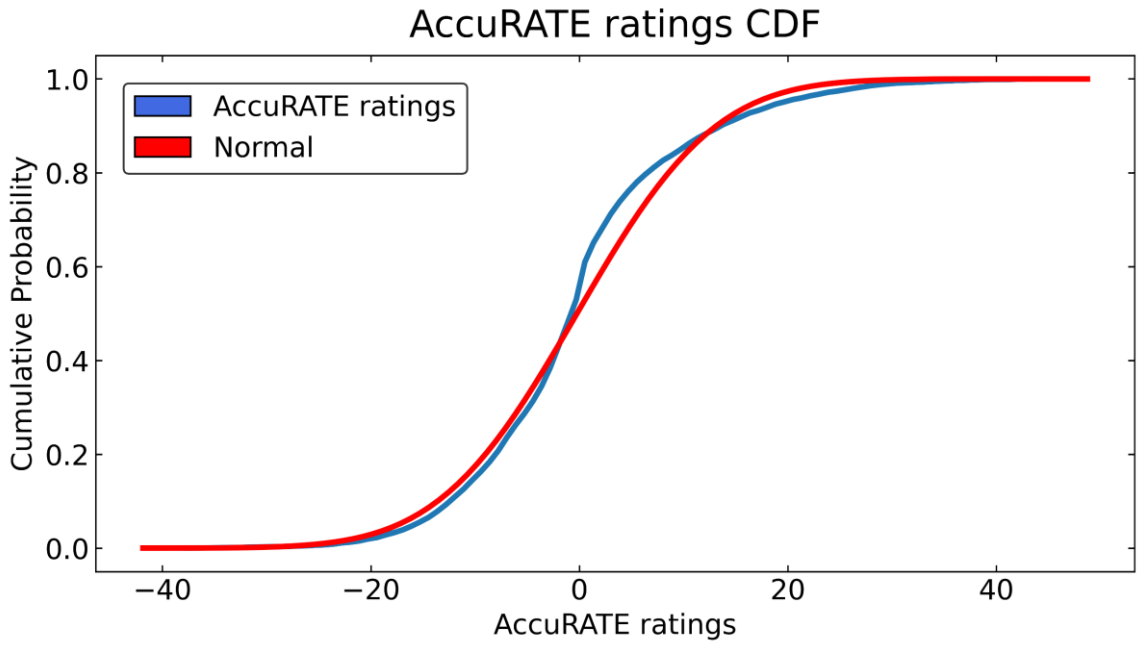


Figure 4-3: AccuRATE's cumulative probability (EPL 2005/06-2017/18)

As we can see from the above figures the distribution of ratings is close enough to normal. Moreover, there are many outliers since the teams have rating scores of zero in the first match week of every season.

4.2.5 Perfect Season Example

The perfect season example is a hypothetical example where all the teams perform perfectly related to their offensive and defensive characteristics. Our perfect season example is inspired by (Chartier, Kreutzer, Langville, & Pedings, 2011). Vaziri (Vaziri, 2016) also examined the perfect season scenario for the Markov-based ranking methods. One of the purposes of the perfect season example is to test the behavior of the method and if it ranks the teams based on their strength and as expected on a theoretical basis. Below we introduce and present our perfect season example of a competition with five teams. Note that all games are played in a neutral venue. We suppose that the rank of each team is the following:

$rank_{Team1} < rank_{Team2} < rank_{Team3} < rank_{Team4} < rank_{Team5}$, and

$rank_{Team1}=1$ $rank_{Team2}=2$ $rank_{Team3}=3$ $rank_{Team4}=4$ $rank_{Team5}=5$.

Table 4-5 shows the games, the winners, and the difference *d* in goals. The logic is that in every game the winner is the team with the better ranking position and their ranking difference is the value of *d*. For example, Team1 and Team2 have 1 position difference, thus $d=1$. Note that for each game that ended with one goal difference, the shooting

accuracy does not play a role. Another example is the matchup between Team1 and Team5 where the d=4 because they have 4 positions difference.

Table 4-5: Perfect season example

Home Team	Away Team	d	Winner	Home Team	Away Team	d	Winner
Team1	Team2	1	Team1	Team2	Team4	2	Team2
Team1	Team3	2	Team1	Team2	Team5	3	Team2
Team1	Team4	3	Team1	Team3	Team4	1	Team3
Team1	Team5	4	Team1	Team3	Team5	2	Team3
Team2	Team3	1	Team2	Team4	Team5	1	Team4

We will first examine the maximum and minimum rating values that each team can achieve. Based on the fact that the goal difference in all games is stable, the range of rating of each team depends on its shooting accuracy. The maximum rating requires perfect accuracy (k=1) while the minimum must be near zero (k=0). Note that the k=0 is impossible for Team1, Team2, Team3, and Team4 because they have scored at least one goal which is counted as a shot and shot on target. The table below shows the rating ranges of each team.

Table 4-6: AccuRATE's rating ranges for the perfect season

Team	Min rating (k=0)	Max rating (k=1)	Rating range
Team1	1 + 1 + 1 + 1 = 4	1 + 2 + 3 + 4 = 10	(4, 10]
Team2	-1 + 1 + 1 + 1 = 2	-1 + 1 + 2 + 3 = 5	(2, 5]
Team3	-2 -1 +1 +1 = -1	-1 -1 + 1 + 2 = 1	(-1, 1]
Team4	-3 -2 -1 + 1 = -5	-1 -1 -1 + 1 = -2	(-5, -2]
Team5	-4 -3 -2 -1 = -10	-1 -1 -1 -1 = -4	[-10, -4]

As we can observe from Table 4-6, the rating ranges indicate that two pairs of teams can change their initial ranking we have presented above. The first is the pair of (Team1, Team2) while the second is the (Team4, Team5). Those two pairs can change their ranking due to the common rating ranges: (4,5] for the first pair and (-5, -4] for the second. Therefore, there are the following four possible ranking lists:

1. $rank_{Team1} < rank_{Team2} < rank_{Team3} < rank_{Team4} < rank_{Team5}$, (the initial ranking)
2. $rank_{Team2} < rank_{Team1} < rank_{Team3} < rank_{Team4} < rank_{Team5}$,
3. $rank_{Team2} < rank_{Team1} < rank_{Team3} < rank_{Team5} < rank_{Team4}$,
4. $rank_{Team1} < rank_{Team2} < rank_{Team3} < rank_{Team5} < rank_{Team4}$.

This prompts the question of whether the first-ranked team (Team1) can be affected if its shooting accuracy k is reduced while d remains unchanged. The same question is also raised for the fourth-ranked team (Team4). To answer this question and find the critical value of k is difficult because there are multiple scenarios. For this reason, we will only analyze a simple scenario where Team1 has a lower shooting accuracy while Team2 achieves the maximum rating. In this scenario, Team1 will move from the first position if the following condition is satisfied (during its rating formation):

$$1 + 2^{k_{1,3}} + 3^{k_{1,4}} + 4^{k_{1,5}} < 5$$

$$\Longrightarrow 2^{k_{1,3}} + 3^{k_{1,4}} + 4^{k_{1,5}} < 4$$

where k_{ij} = the shooting accuracy of team i in the game between team i and team j.

There are many cases where the condition is satisfied but for simplicity, we only show two cases:

- In the first case, we considered that $k_{i,j}$ of Team1 is the same in every game. The critical point where Team1 moves from the first position is for $k \leq 0.26$.
- Another case is when Team1 has the worst accuracy ($k_{i,j} \approx 0$) in the games with Team3 and Team4. Then in the game with Team5, the $k_{1,5}$ must be greater than 0.5 because $1 + 1 + 4^{0.5} > 4$.

In other words, the above shows that the stronger team in order to maintain its ranking position should be able to achieve good shooting accuracy in the games against the weaker teams. However, by taking into account that Team1 is the strongest team, the shooting accuracy cannot affect its ranking easily because goals also play an important role. In a similar manner, we can examine the rankings for Team4 and Team5.

In summary, since the perfect example is a hypothetical example and does not refer to reality, the purpose here is only to examine the behavior of the method theoretically and if the rankings generated are consistent with the example. Therefore, it is difficult to draw any further conclusions and for this reason, a sensitivity analysis should be applied to a real dataset.

4.2.6 Sensitivity Analysis - English Premier League

In this subsection, sensitivity analysis for the method is performed in order to test the stability of ranking vectors on a real soccer dataset which is the EPL for the seasons 2005/06 to 2017/18. In particular, the sensitivity analysis is performed for each season separately in three periods: at the beginning of the season (1st - 3rd match weeks are

included), at the first half of the season (1st - 19th match weeks), and for the entire season (all match weeks are included). This allows us to draw conclusions about the behavior of the method in different phases of the season. In addition, the periods of the first half and at the end of the season are tested for a different number of games that are randomly selected. The purpose here is to find out how many games affect the ranking list. All results are computed (for each period and a specific number of games) as mean values over the total number of seasons and teams. Our sensitivity analysis includes two scenarios. The first scenario focuses on shooting accuracy (*k*) while the second on goal difference (*d*). The pseudocode and description of the procedure are given below:

- Pseudocode of sensitivity analysis:

```
Input: season_start, season_end, period, games, scenario
Output: pct_teams, avg_places
improvement = 0
counter_teams = 0
seasons = season_end-season_start
teams = 20
for season = season_start; season <= season_end; season++:
      initial_rank = accurate(season, period, games)
      for team = 1; team <= teams; team++:
            new_ranking = sensitivity(season, period, games, scenario)
            improvement = initial_ranking[team] - new_ranking[team]
            if improvement > 0:
                  counter_teams += 1
            improve += improvement
pct_teams = 100*(counter_teams/seasons)/teams)
avg_places = improve / counter_teams
```

- Procedure description
  - Inputs: The `season_start` is 2005 and indicates the 2005/06 season. The `season_end` is 2018 and indicates the 2017/18 season. The `period` represents one of the three periods (begin, first half, end), the `games` input indicates the number of games (1 game, 2 games, etc.), and `scenario` determines which variable is tested (*k* or *d*).
  - Outputs: The `pct_teams` is the percentage of teams that improve their ranking position and `avg_places` is the average number of places of improvement. For example, if `pct_teams` is 10 this means that 10% of total

teams, i.e., 2 teams (10% of 20 teams competing in the EPL = 2 teams) improve their ranking position. If `avg_places` is 1 this means that the teams improve in average their ranking by 1 position.

- Functions: The `accurate` function is the AccuRATE method that rates the teams and returns their rankings. The `sensitivity` is the procedure of sensitivity that returns the new ranking list.

❖ Scenario 1

The purpose of this scenario is the perturbation of *k* and then to measure the number of changes in the ranking list. The value of *k* is increased for one team every time. When all teams are tested, we compute the average number of teams that improve their ranking position and by how many places. The aim is to identify how varying levels of improving variable *k* can improve the ranking position of a team. We are also interested to see whether the period of the season has an impact on the ranking and how many games are needed to improve the ranking position of a team. Computed results are depicted in Table 4-7, Table 4-8, and Table 4-9 for the three different season periods.

Table 4-7: Changes in *k* at the beginning of the season

Games	**Description**	***k***					
	Increased %:	**5%↑**	**10%↑**	**15%↑**	**20%↑**	**25%↑**	**30%↑**
3-games	pct teams %:	6.92	14.62	18.46	21.92	24.23	27.69
	avg places:	1	1.03	1.08	1.23	1.27	1.31

Table 4-8: Changes in *k* at the first half of the season

Games	**Description**	***k***					
	Increased %:	**5%↑**	**10%↑**	**15%↑**	**20%↑**	**25%↑**	**30%↑**
1 game	pct teams %:	0	0.38	1.92	1.54	4.23	5.77
	avg places:	-	1	1	1	1	1
2 games	pct teams %:	1.54	1.15	4.62	5.77	6.54	7.31
	avg places:	1	1	1	1	1.18	1
3 games	pct teams %:	1.54	3.85	6.54	10	10.38	11.54
	avg places:	1	1.1	1.12	1.04	1.15	1.07
4 games	pct teams %:	2.31	5	7.69	10.77	14.62	15.38
	avg places:	1	1	1.05	1.07	1.18	1.1

Table 4-9: Changes in *k* at the end of the season

Games	**Description**	***k***					
	Increased %:	**5%↑**	**10%↑**	**15%↑**	**20%↑**	**25%↑**	**30%↑**
1 game	pct teams %:	1.15	1.54	1.15	1.54	1.15	1.54
	avg places:	1	1	1	1	1	1
2 games	pct teams %:	0.77	1.15	2.31	1.54	4.23	5.38
	avg places:	1	1	1	1	1.09	1
5 games	pct teams %:	2.69	3.85	5.38	7.69	10.77	10.38
	avg places:	1	1	1	1.05	1.11	1.15
10 games	pct teams %:	3.85	8.85	8.85	15.77	16.92	22.69
	avg places:	1	1.04	1.04	1.12	1.16	1.17

Before proceeding to the analysis of the results it is important to explain the table structure of the results. The first column denotes the number of games where the variable *k* is improved. The second column denotes the two metrics that are explained in the procedure description, i.e., "pct teams" and "avg places". The rest of the columns represent the level of improvement. For example, in Table 4-9 the result (10 games, *k*=5%) means that in every test one team has improved its *k* by 5% in 10 random games of the entire season.

As a side note, when a random match is selected between two teams, *i* and *j*, the *k* is improved as follows:

$$\bar{k}_{i,j} = k_{i,j} \cdot (1 + l)$$

where $k_{i,j}$ is the shooting accuracy of team *i* before improvement, *l* is the level of improvement (i.e., 5%, 10%, etc.), and $\bar{k}_{i,j}$ is the improved shooting accuracy.

We continue with the example (10 games, *k*=5%), where the metric values are 3.85% and 1 for "pct teams" and "avg places" respectively. The results are interpreted as follows: given that in every test one team has improved its *k* by 5% in 10 random games at the end of each season, when all teams are tested separately only 3.85% of teams manage to improve their ranking by 1 place on average.

As we can observe, the ranking of teams is not sensitive to small perturbations of *k*. As it was expected, higher *k* values in more games resulted in a better ranking for a

team. The most sensitive period is when the season starts while the other two periods are more stable due to more games being played.

Our general conclusion is that the ranking of a team is difficult to change if its shooting accuracy is not improved significantly for a large number of matches. Nevertheless, the ratings are improved from good shooting accuracy and the more accurate ratings can be exploited if they are utilized for other purposes and applications such as outcome prediction and betting.

❖ Scenario 2

In the present scenario, the sensitivity is examined by modifying the goal difference *d* for one team every time. When the team is the winner then *d* is increased in its favor for 1 or 2 goals. The procedure is performed in the same way as in Scenario 1. The positive effect on the team's ranking position is examined and the results are depicted in Table 4-10, Table 4-11, and Table 4-12 for the three different season periods.

Table 4-10: Changes in *d* at the beginning of the season

Games	**Description**	***d***	
	Improved:	**1↑**	**2↑**
3 games	pct teams %:	91.92	93.85
	avg places:	4.44	6.73

Table 4-11: Changes in *d* at the first half of the season

Games	**Description**	***d***	
	Improved:	**1↑**	**2↑**
1 game	pct teams %:	29.23	49.62
	avg places:	1.47	1.59
2 games	pct teams %:	48.46	68.85
	avg places:	1.63	2.03
3 games	pct teams %:	64.23	78.85
	avg places:	1.92	2.59
4 games	pct teams %:	69.62	86.54
	avg places:	2.15	3.11

Table 4-12: Changes in *d* at the end of the season

Games	Description	*d*	
	Improved:	**1↑**	**2↑**
1 game	pct teams %:	19.23	30.77
	avg places:	1.12	1.29
2 games	pct teams %:	34.23	54.23
	avg places:	1.36	1.62
5 games	pct teams %:	65.38	79.23
	avg places:	1.85	2.66
10 games	pct teams %:	81.15	92.31
	avg places:	2.77	4.13

The columns "1↑" and "2↑" under the *d* denote that the goal difference is increased by 1 or 2 goals respectively. It is clear from the results obtained that the goals scored by each team affect the ranking results most significantly. As entailed both from the above tables even one more goal scored by the winning team in one (random) game plays a vital role. Notably, the most sensitive period is when the season starts where 91.92% of teams can improve their ranking by 4.44 positions on average. In the periods of the first half and at the end of the season, 29.23% and 19.23% of the teams can improve their ranking by 1.47 and 1.12 positions on average respectively.

4.2.7 Conclusions

In this section, a novel rating system was proposed, taking into account the game outcomes, margin of victory, and shooting accuracy of each team against their opponents. This allows teams to be rewarded or penalized for a game by assessing the significance of the offensive opportunities. By utilizing the ratio of *TST* over the *TS*, as an exponent of the two teams' net goal difference in a game, an accurate team is awarded more points. Additionally, in a game between a strong and weak team, the strong team cannot boost its ranking only by goals but must also achieve good shooting accuracy.

The method is presented with the aid of an illustrative example. We have also examined the distribution of ratings generated by AccuRATE's method for the EPL, during the seasons 2005-2018, and it is close enough to normal. Moreover, the case of the perfect season example is examined for two purposes: (1) to validate the method if it ranks the teams as expected when their rankings are clear, and (2) to examine the

behavior of the method theoretically. Next, a sensitivity analysis is performed to test the stability of ranking vectors on a real dataset. The results of the sensitivity analysis showed that the difference in goals has the main effect on the formation of the rating and ranking of the team. Nevertheless, each team aiming to maintain its ranking position must achieve satisfactory shooting accuracy.

The fact that AccuRATE's orientation toward soccer teams' rating field adds an extra degree of difficulty in applying it to other domains since the method considers the outcomes of games, the margin of victory, and the shooting accuracy. Essentially, the difficult part which is the key to using it in other fields is to determine the term of shooting accuracy for the items to be ranked. In some other sports, this may be effortless, but in completely different contexts, it can be more challenging. A more generalized method is proposed in the next section and it overcomes this limitation.

Finally, the performance study of the AccuRATE method will be discussed and compared with other rating systems in the main application (Chapter 6) of this document.

4.3 PointRATE: The MAUT/MAVT Approach for Rating and Ranking

4.3.1 Introduction

Although this work originally started as an extension of our rating system (Kyriakides, Talattinis, & Stephanides, 2017) described in the previous section, a sophisticated and generalized rating method with applications in various fields emerged along the way. Our initial aim was to combine multiple teams' statistics and generate reliable ratings and rankings that reflect the overall performance of teams. One possible approach would be to employ the WSM (Weighted Sum Method) where weights are assigned to multiple performance values and then the scores are aggregated into a single one. Also, after exploring the possibility of adopting a utility-based rating system the MAUT/MAVT approach was considered a feasible option. The latter gives us the flexibility to involve utility or value functions. However, the case of soccer team ratings must be as objective as possible. To make the rating modeling more comprehensive and generalized to other fields, we have selected a point system to construct the value function by synthesizing the partial functions on every single attribute. We call our proposed method "PointRATE" because takes into account reward function points

defined by the user in order to rate items so the name seemed quite fitting (Points + RATE). This section is devoted to the development and description of this method.

As stated, the basic idea behind the method is the WSM (Weighted Sum Method) and in fact, it is based on MAUT (Multi-Attribute Utility Theory) / MAVT (Multi-Attribute Value Theory). The fact that part of the overall method is based on the MAUT/MAVT naturally shares similarities with other common methods. However, in this work, while we propose a modified method for ranking and comparing alternatives/items by utilizing user preferences, our focus is different from that of the bulk of the literature regarding experiments and applications. In comparison with prior works, our work examines the application of the method in two different sectors where slight research activity has been observed until now. The first application is the rating of the EPL soccer teams and then the ratings are used to perform predictions of game outcomes. This application is included in the experimental section of Chapter 6. The second application concerns the ability of the method to guide correctly a genetic algorithm for optimization of the parameters of a financial trading strategy intended for investors with different preferences. The second application is also deployed as a part of method validation and could also be seen as an extension of the sensitivity analysis. Section 7.3 deals with the second application.

In this section, in the first place, we introduce the MAUT/MAVT, then we describe our proposed method and how to deal with special cases, and after that, we discuss how the ratings generated by the method are connected with the user's objectives. Next, we demonstrate the model parameters for rating soccer teams which are applied in our illustrative example from section 3.2. Moreover, a sensitivity analysis is carried out to evaluate the stability of the method by utilizing the soccer team ratings model's parameters. Finally, the section closes with some conclusions.

4.3.2 MAUT / MAVT

In this section, a brief description of MAUT/MAUV is provided. As already mentioned in section 2.3, MAUT and MAVT belong to the area of MADM. Those methods have been used in many different problems and purposes, where the decision-makers express their preferences in the form of utility or value functions (Keeney & Raiffa, 1976). In economics usually, we call the value functions utility functions. Particularly, the value function represents the worth of an alternative based on the user's

preferences. In decision theory, utility functions are referred to as the decisions under risk. Therefore, the main difference between MAUT and MAVT is that the first is designed for decisions under risk and uncertainty, whereas the latter does not take uncertainty into account. In addition, MAVT can be considered as a simplified form of MAUT since the latter is an extension of the first.

An attribute represents a criterion as a performance measure (Ramanathan, 2004). For example, to rate a soccer team based on its strength, the offensive ability is the criterion while the total goals statistic is the attribute. The MAUT/MAVT aims to aggregate attribute values into a single value and as a result, first we aggregate and then compare. The additive form is very simple (Keeney R. L., 1971) and one of the most widely used forms of the multi-attribute value function. By using utility or value functions for each attribute the main requirement is to transform them into one common scale. In each attribute, two points are necessary to be defined (Belton & Stewart, 2002). These reference points are the most and least preferred values. On the global scale are defined by the decision maker while on the local scale by the alternatives.

In the additive model, the value function involves compensation among attributes which means that a bad performance in one attribute can be compensated by the good performance of other attributes. Also, basic conditions are required for the additive function (Keeney R. L., 1971; Fishburn, 1982; Keeney R. L., 1996). Conditions such as preferential independence and utility independence among attributes are very crucial. Preferential independence implies that the preference for any attribute must not be affected by the values of other attributes. Utility independence implies that the utility for a given attribute must be independent of the level of other attributes. Other important conditions are the difference independence and additive independence.

In MAUT/MAVT the overall utility or value function must be assessed according to the user's preferences. A variety of techniques exists to elicit the single-attribute value function (von Winterfeldt & Edwards, 1986). According to (Belton & Stewart, 2002) for score elicitation, there are three ways: (1) Definition of a partial value function, (2) Construction of a qualitative value scale, and (3) Direct ratings of each alternative. Some well-known and often-used methods are the bisection and the differences methods which belong to the first way.

The importance of attributes is an essential step in the process which aims to determine their weight. Many techniques have been proposed in the literature dealing

with the elicitation of weights. The weighting types can be subjective, objective, or integrated methods (Jahan, Mustapha, Sapuan, Ismail, & Bahraminasab, 2012). In the subjective type, weights depend on the decision maker while in the objective type, the decision maker has no role in the importance of weights.

Some popular methods of subjective type are direct rating, point allocation, rank-order, swing, trade-off, interval methods, and pairwise comparisons (Riabacke, Danielson, & Ekenberg, 2012). As for the first two methods, Bottomley et al. in their study show that weights derived from direct rating are more reliable than those in point allocation (Bottomley, Doyle, & Green, 2000).

A well-known method that belongs to the objective type is the mean weighting method where equal weights are assigned to each attribute (Deng, Yeh, & Willis, 2000). Other important approaches that produce objective weights are the entropy and the standard deviation methods. Moreover, Diakoulaki et al. (Diakoulaki, Mavrotas, & Papayannakis, 1995) proposed the CRITIC method where they used the inter-criteria correlation that is based on the standard deviation method to assign objective weights. Also, Paradowski et al. (Paradowski, Shekhovtsov, Bączkiewicz, Kizielewicz, & Sałabun, 2021) conducted an analysis of several objective weighting methods and introduced their implementation for comparing different weighting methods.

The integrated type refers to a combination of weighting methods. Two noteworthy that belong to the integrated type methods proposed by (Ma, Fan, & Huang, 1999), and (Wang & Parkan, 2006).

As pointed out before, after aggregation and calculation of the alternatives' ratings we choose the alternative with the highest rating score. The final step here is to perform a sensitivity analysis. Usually, a different weighting method can be employed to evaluate the importance of attributes. Also, other weighting schemes can be applied, such as equal weights to test the final ranking results. After the sensitivity analysis, recommendations can be made.

4.3.3 PointRATE Rating System

The detailed steps of the PointRATE method are illustrated as follows:

❖ Step 1 - Alternatives/Items and Attributes selection:

In the first step, we identify the alternatives/items we need to rate and the most important attributes that play a crucial role in them.

Assuming a set of *m* alternatives/items

$$A = \{A_1, A_2 \dots A_m\},$$

and a set of *n* attributes

$$X = \{X_1, X_2 \dots X_n\} \text{ with domains } D = \{D_1, D_2 \dots D_n\}.$$

Each alternative A_i corresponds to a set of observed attributes' values

$$A_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n}\},$$

where $x_{i,j}$ is the observed value (outcome) of attribute *j* of *i*-th alternative.

In notation, when we are not referring to a particular (*i*th) alternative, we omit *i* index and we refer to attributes' outcome as $\{x_1, x_2 \dots x_n\}$.

Sometimes many attributes can be grouped into higher-level, abstract attributes which in MCDA (Multi-Criteria Decision Analysis) are called criteria. As such, a set of abstract attributes $G = \{G_1, G_2 \dots G_q\}, q < n$ is generated and each attribute can participate in one abstract attribute.

❖ Step 2 - Assess the utility/value function based on the user's preferences:

In MAUT, we assess a real-value function $U(X_1, X_2 \dots X_n)$ for each attribute which is called utility. In the case of MAVT, we call it a value function $V(X_1, X_2 \dots X_n)$. Therefore, after determining the alternatives and attributes in the previous step, the goal is to build the user's utility/value function. Here, we suppose that the outcome is riskless, thus we have a value function. The value of each attribute is transformed on a common scale. A point system is described below in four parts (A, B, C, D), where we can specify the user's value function. The main idea is based on point allocation, where the attribute's domain is split into sub-intervals and then we allocate for each one the points of our preference. Then, the value function of each attribute is computed. We also referred to a single-attribute value function as a reward function since it awards points (formally utility or value) to the attribute.

For each *i* attribute:

A. Define points (b_i) of the Domain (D_i):

The user must provide $k>1$ points of interest in *i*'s attributes' domain

$D_i = [x,y]$ such that

$$b_i = \{b_{i,1}, b_{i,2} \dots b_{i,k}\},$$
$$\forall j \in [2 \dots k] \; b_{i,j-1} < b_{i,j}, \qquad b_{i,j} \in D_i \text{ and } b_{i,1} = x, b_{i,k} = \text{y}. \qquad (4.1)$$

The number of points depends on the user's preferences and may differ for each attribute. Furthermore, the points represent the critical values of each attribute

according to the user's preferences. For example, if we have an attribute of students' grades (from 0 to 10), we can define k=4 points: 0, 5, 8.5, 10. The goal of this procedure is to divide the attribute's domain into classes. Therefore, in this example, the points that are added in the b sequence are based on the fact that grades below 5 are considered a failure, grades between 5 and not equal to 8.5 are satisfactory, and grades above 8.5 belong to the top students.

The sequence b_i is strictly increasing due to the constraints of (4.1).

After defining points b_i, the resulting set of classes is

$$C_i = \{C_{i,1}, C_{i,2} \dots C_{i,k-1}\}, \quad (4.2)$$

where $C_{i,j} = [b_{i,j}, b_{i,j+1}) \; \forall \; j \in [1 \dots k-2]$ and $C_{i,k-1} = [b_{i,k-1}, b_{i,k}]$

For better understanding, we consider again the example of students' grades, and by applying (4.1) and (4.2) we have

$$b_i = \{0, 5, 8.5, 10\} \text{ and } C_i = \{[0, 5), [5, 8.5), [8.5, 10]\}.$$

B. Define reward points p_i for each class:

In this part, we define the reward points in order to utilize them later when we assess the value function. The MAVT has the requirement that all attribute values should be on the same scale and usually, the normalized range [0,1] is used. For our purposes, we use the range from 0 – 100 due to its simplicity and easy interpretation. Thus, we assume that the lowest value of the value function is 0 and the highest is 100. Specifically, first, we assign a starting point $p_{i,0} \in [0,100]$ and then for each set C_{ij} the user must specify the number of reward points to distribute between $b_{i,j}$, $b_{i,j+1}$ as $p_{i,j}$. The assignment of values is based on the fact that each reward point $p_{i,j}$ represents how many points we need to reward the class $C_{i,j}$. Moreover, the type of attribute plays an important role in the way we distribute points. Attributes can be classified into three types (Belton & Stewart, 2002). For each type, we explain how to define the reward points. For simplicity, we referred to them as "Type-1", "Type-2" and "Type-3".

- Type-1: In this type, the higher values are considered better and, therefore, the user's value function for this attribute is strictly increasing. For this reason, we set the starting point to be the lowest value of the value function. For example, the student's grades belong to this type.

The reward points can be defined as

$$p_i = \{p_{i,0}, p_{i,1}, p_{i,2} \dots p_{i,k-1}\},$$

$$p_{i,0} = 0, \quad \forall j \in [1 \dots k-1] \; 0 < p_{i,j} \leq 100, \quad \sum_{j=0}^{k-1} p_{i,j} = 100, \qquad (4.3)$$

where $p_{i,0}$ indicates that we start from the lowest value of the value function.

- Type-2: In this type, the attribute's lower values are preferred to the higher ones and, therefore, the value function is strictly decreasing. For example, attributes that refer to cost. Therefore, the reward points can be defined as follows

$$p_i = \{p_{i,0}, p_{i,1}, p_{i,2} \dots p_{i,k-1}\},$$

$$p_{i,0} = 100, \quad \forall j \in [1 \dots k-1] \; -100 \leq p_{i,j} < 0, \quad \sum_{j=0}^{k-1} p_{i,j} = 0, \qquad (4.4)$$

where $p_{i,0}$ indicates that we start from the highest value of the value function.

- Type-3: The user's value function for the attribute is non-monotonic.

$$p_i = \{p_{i,0}, p_{i,1}, p_{i,2} \dots p_{i,k-1}\},$$

$$p_{i,0} \in [0,100], \quad \forall j \in [1 \dots k-1] \; -100 \leq p_{i,j} \leq 100. \qquad (4.5)$$

To summarize, the assignment of $p_{i,j}$ depends on the kind of attribute and the user's evaluation of the importance of each attribute's range.

C. Compute cumulative reward points for each range:

At this point, we compute the cumulative points *s* for each range as given by the equation below:

$$s_i = \{s_{i,0}, s_{i,1}, s_{i,2} \dots s_{i,k-1}\}, \qquad (4.6)$$

where $s_{i,0} = p_{i,0}$ and $s_{i,j} = \sum_{l=1}^{j} p_{i,l}$.

For instance, if we examine an attribute of Type-1 and we choose k=3, then $p_{i,0}$=0. If we set $p_{i,1}$=80, and $p_{i,2}$=20 then the reward points for $C_{i,1}$ are distributed from 0 to 80 (not equal), and for $C_{i,2}$ from 80 to 100. As a result, we have

$p_i = \{0, 80, 20\}$ and $s_i = \{0, 80, 100\}$.

In equations (4.3) and (4.4), the constraints satisfy the requirement of having 0 and 100 as the minimum and maximum value of the reward function respectively. Since no constraints have been added in (4.5), in the case where 0 and 100 are not included in s_i then we normalize s_i in the range from 0 to 100.

D. Compute reward (value) function for *i* attribute:

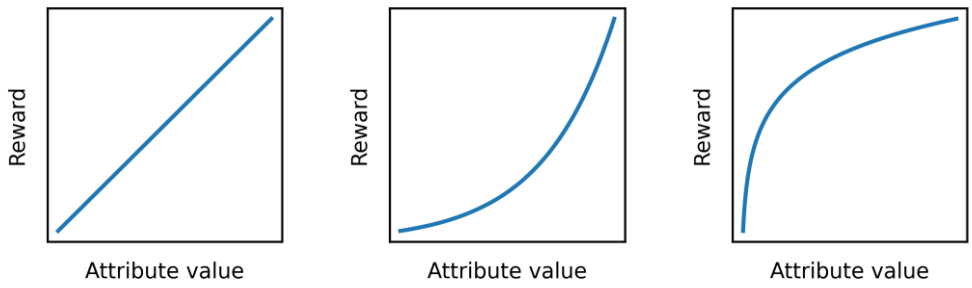
There are several ways that we could assess the user's value function of each attribute. The first option is to build a piecewise function. The user can apply a

value function that can take on a variety of forms and, therefore, may differ in each reward range. The main benefit of this approach is that the user is not required to specify a plethora of points s_i, p_i.

The reward function can be described by piecewise function f_i as follows:

$$v_i(x_i) = \begin{cases} v_{i,1}(x_i), & if\, x_i \in C_{i,1} \\ v_{i,2}(x_i), & if\, x_i \in C_{i,2} \\ \dots, & \dots \\ v_{i,\mathrm{k}-1}(x_i), & if\, x_i \in C_{i,\mathrm{k}-1} \end{cases}, \qquad (4.7)$$

where $v_i(x_i)$ are the total points awarded to attribute X_i, for the specific observed value x_i. Each $v_{i,j}$, where $j \in [1 \dots k-1]$ can be defined by the user, or computed using function approximation, by using the relation $\{(b_{i,j}, s_{i,j}), (b_{i,j+1}, s_{i,j+1})\}$. Though there are numerous reward schemes and utility functions that have been proposed by literature, in order to make the procedure simple, we have selected the $v_{i,j}$ to be one of the following basic functions (depicted in Figure 4-4): linear, exponential, and logarithmic. For simplicity, we only show the strictly increasing functions. The same reward function schemes can also be applied in a strictly decreasing form.



a: linear **b: exponential** **c: logarithmic**

Figure 4-4: Three basic functions

The second option is to do a curve fitting of the reward function. This approach is selected in some cases, where due to the complexity of the function, it may be more effective to use a plethora of points s_i, p_i and then approximate the function using interpolation methods, such as Polynomial and Spline Interpolation or Radial Basis Function.

❖ Step 3 – Determine the importance of Attributes:

As pointed out in subsection 4.3.2 there are several approaches developed to determine the importance of the attributes. Here, for the subjective type, the direct rating (DR) technique has been chosen as it is very simple and easily applicable. In the DR method,

the form of determining weights is made by two steps. In the first step, a number from a certain scale is assigned to each attribute, and in the second step, the attributes' numbers are normalized. The main advantage of the DR method is the possibility of choosing the importance of an attribute separately without taking into account the others. Although a scale from 0 to 100 is generally used, here for simplicity a 10-point Likert scale is used to classify the grade of importance for every single attribute.

A. Rating:

The user must define the importance of each attribute X_i (by selecting a value on a scale of 1 to 10), as a value h_i=[1..10].

The scale has the following meaning:

1-2: Slightly Important,

3-4: Moderately Important,

5-6: Important,

7-8: Very Important,

9-10: Extremely Important.

As each attribute is selected by the user, it is meaningless to start the scale from "unimportant". Other scales can also be considered; e.g., from 1 to 3 where 1 = worst, 2 = medium, and 3 = best.

B. Normalization:

We normalize ratings and transform them into weights for each attribute. The weight w_i of each attribute, X_i can be calculated as follows:

$$w_i = \frac{h_i}{\sum_{i=1}^{n} h_i} . \quad (4.8)$$

In the case of abstract attributes, the same process takes place for each abstract attribute G_i resulting in a value h_i, as well as for each regular attribute $X_{ij} \in G_i = \{X_{i1}, X_{i2}, \ldots X_{il}\}$ resulting in a value h_{ij}, where l is the total number of regular attributes that are grouped in G_i.

In this case, the final weight for each attribute is computed as

$$w_{ij} = \frac{h_{ij}}{\sum_{k=1}^{l} h_{ik}} \cdot \frac{h_i}{\sum_{i=1}^{q} h_i} . \quad (4.9)$$

❖ Step 4 - Compute the rating for each alternative/item:

The final rating for each A_i alternative/item is a classical additive value model and is calculated as a weighted sum

$$r_i(A_i) = \sum_{j=1}^{n} w_j \cdot v_j(x_{i,j}), \qquad (4.10)$$

where n is the total number of attributes, w_j and v_j is the weight and reward function for the j attribute respectively, and $x_{i,j}$ is the observed value (outcome) for the j attribute of i-th alternative/object. Since (4.10) is a weighted sum and the reward function of each attribute can take values from 0 to 100, therefore, this forces the final rating to be in the same range.

Finally, the overall procedure steps are illustrated schematically below:

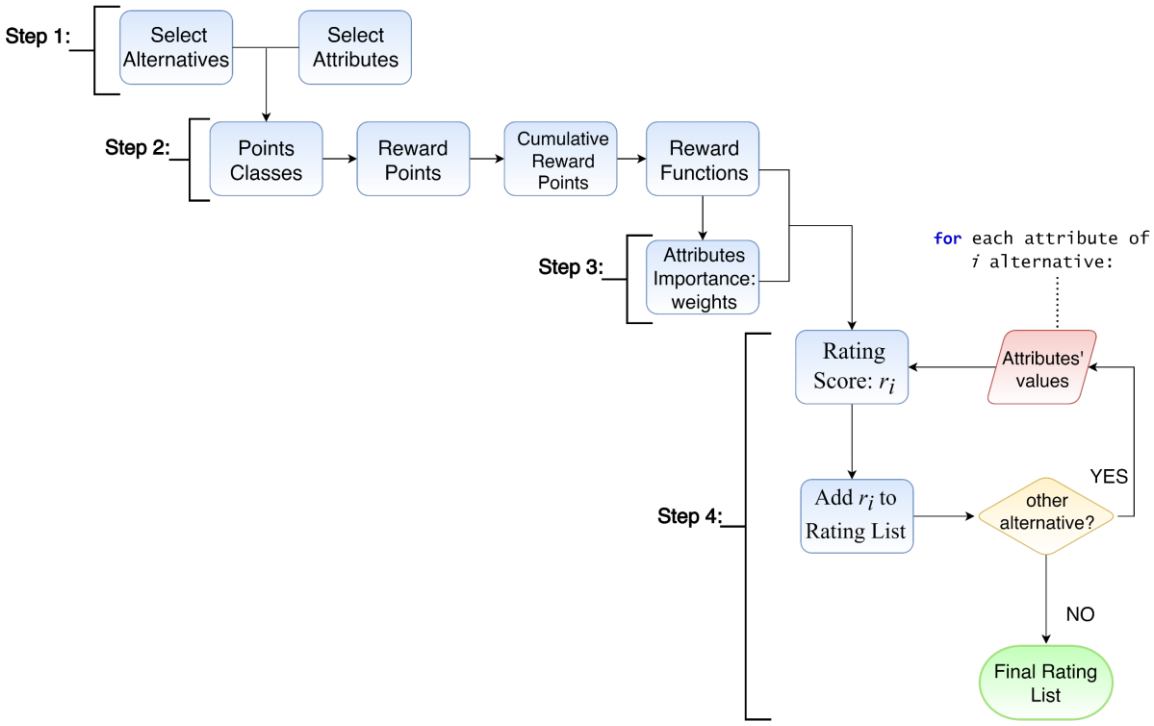


Figure 4-5: PointRATE procedure steps

4.3.4 Special Cases - Example with Non-Monotonic Reward Functions

Some special cases require non-monotonic reward functions for some attributes (Type-3). One such example is the age of a start-up's CEO. Being too young has the disadvantage of a lack of experience while being too old does not give the candidate enough time to execute long-term plans himself. In this special case, we can define points of interest

b_{age} = {20, 30, 40, 50, 60, 70} and

C_{age} = {[20, 30), [30, 40), [40, 50), [50, 60), [60, 70]},

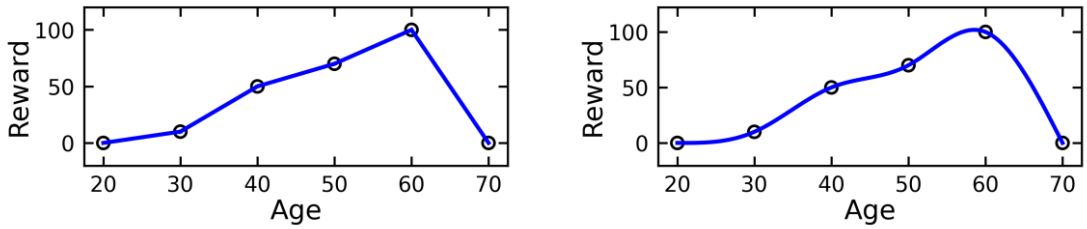
the reward points that indicate a decline in the later ranges, such as

p_{age} = {0, 10, 40, 20, 30, -100}

and reward cumulative points

s_{age} = {0,10,50,70,100,0}.

We have selected to assess the reward function by curve fitting and in particular, we estimated it through interpolation. In our example, we show two alternative types of interpolation. Finally, whatever type of interpolation is chosen, the reward function passes exactly through the set of data points p_{age} ,s_{age}. The graph of the reward function is depicted in Figure 4-6 (a) and (b), approximated using linear interpolation and spline interpolation respectively.



a: reward function approximated using linear interpolation

b: reward function approximated using spline interpolation

Figure 4-6: CEO age reward function

4.3.5 Ratings and Objectives

As we have pointed out previously, the objective is to select the alternative that maximizes the total utility of the user. Thus, the ratings generated reflect the preferences of the user. For instance, a simple problem could be a selection of a car where a user is considering the criteria of cost, design, safety, and fuel economy. In this example, the typical objective is to find the best car related to user preferences. For this problem, the final ratings have a direct impact on the final decision. However, in some cases, problems are more complex and their objective is indirectly related to ratings. For instance, in soccer, we rate teams every match week of the season based on the prior results from the games played and such ratings will be used for making predictions of game outcomes, and then we make decisions for betting. In this case, the modeling for ratings is a part of the entire process and if we want to achieve high profits then our model must be different from that of finding the best offensive team. Figure 4-7 demonstrates the method when the selection of attributes, points, classes, functions, etc. reflects the final target directly.

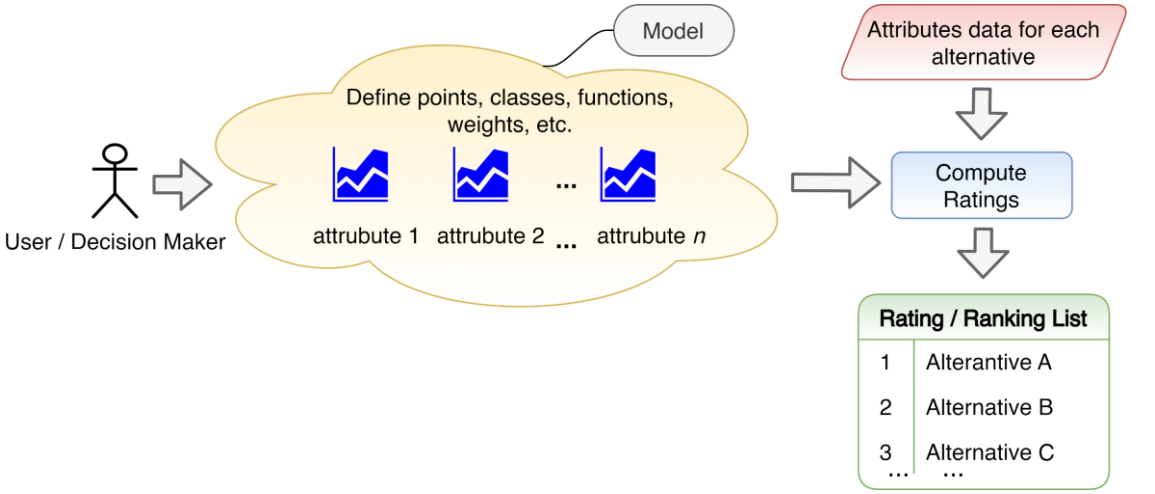


Figure 4-7: PointRATE for direct objective

For indirect objectives, the general process is demonstrated in Figure 4-8.

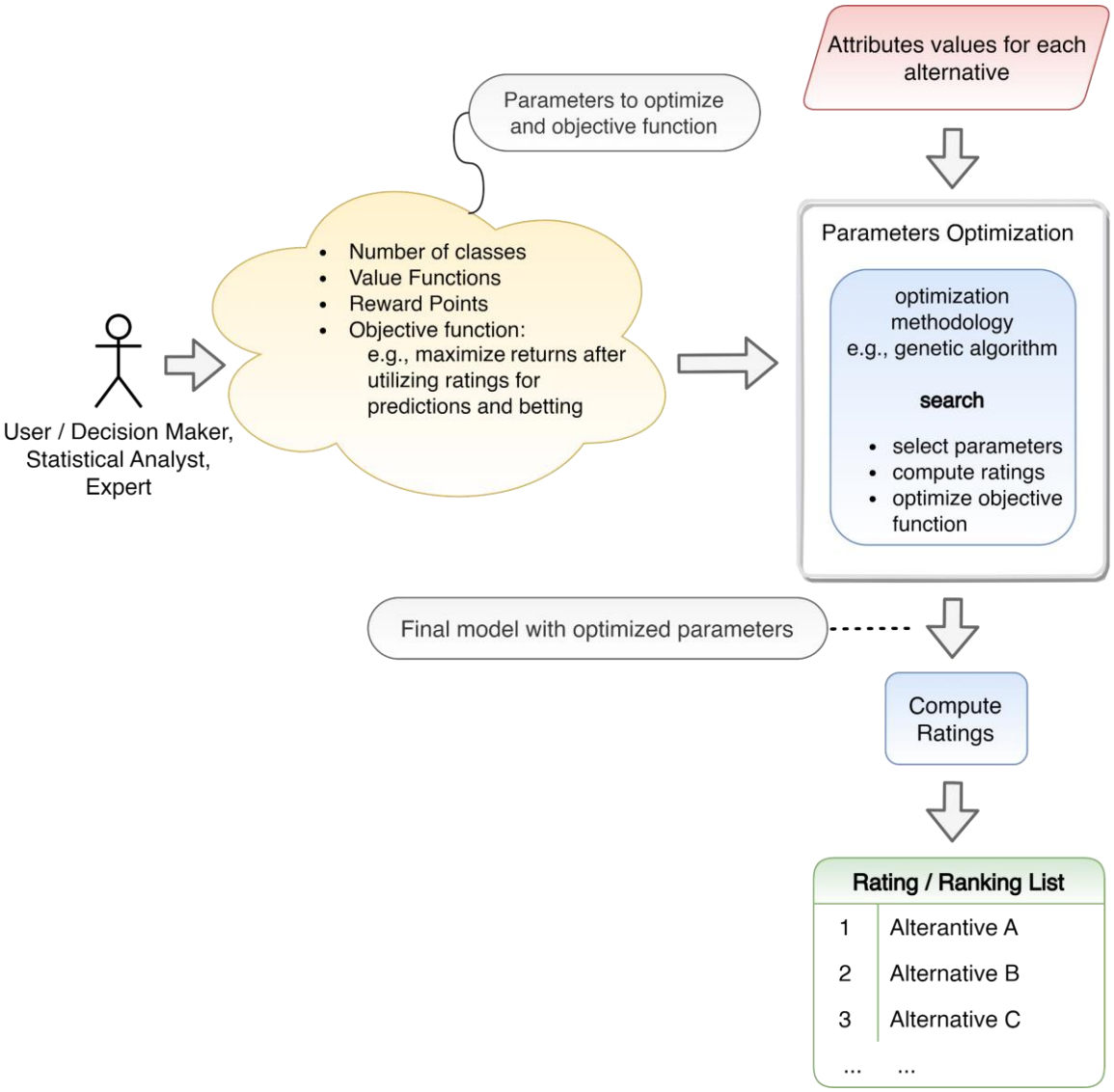


Figure 4-8: PointRATE for indirect objective

When there are more complex and indirect objectives the process is different. The user can give a range of inputs for points, functions, weights, etc., and then the purpose is to optimize them according to the final objective. For example, suppose that a bettor pursues high returns by placing bets on the outcome of soccer matches by utilizing the rating values of the soccer teams. In this case, the weights of the attributes can be optimized to the final returns. It is worth mentioning that the selection of optimization methodology depends on the problem.

4.3.6 Modeling the Method for the Soccer Team Ratings

Even if the use of the MAUT/MAVT method finds wide applicability in different sectors of daily life, in the rating of soccer teams slight research activity has been observed until now. MADM approaches applied to rank soccer teams mainly utilize economic criteria. (Sinuany-Stern, 1988) used the AHP method to rank 16 soccer teams of the Israeli National League. The six criteria were (1) the team's facilities, (2) the coach of the team, (3) the level of players, (4) the team's fans, (5) and (6) the performance of the team for previous and current season respectively. Kiani et al. (Kiani Mavi, Kiani Mavi, & Kiani, 2012) in their study have ranked football teams based on economic criteria such as total revenue, players' wages, etc. The weight importance of attributes had been assigned by AHP, and then they applied the TOPSIS method.

The primary purpose here is to rate soccer teams by taking into account multiple attributes (statistics of teams). Furthermore, we aim to keep the modeling simple and as objective as possible. This signifies that we have to select the reward functions objectively and at the same time consider them the simplest possible. In our attempt to define a model that meets those goals as closely as possible, we have been inspired by the GPA (Grade Point Average) used to measure students' performance. All the details of the modeling are explained in the rest of this subsection.

In this model, for simplicity and fairness comparison, we have selected the same soccer statistics (attributes in the current case) that were used for the GeM rating system in the example from section 3.2. Also, the procedure explained below is followed in the same way for all attributes.

The logic behind the assessment of points and classes is performed empirically based on rewarding the teams for their high performance. For this reason, exponential functions are preferred since the variations at low attribute performance are considered

less significant than variations at the higher part of the scale. We have kept a common reward function for all the attributes for more objectivity. Especially, the reward functions of attributes are comprised of five classes and we allocate 100 reward points by giving 20 points to each class. For each attribute, the first class with range [0-0.7) represents 70% of total performance and then for the other four classes follow the ranges [0.7-0.77), [0.77-0.85), [0.85-0.93), and [0.93-1.0]. As already noted, this idea is based on the grading system of GPA (Grade Point Average) that is commonly used in most universities in the United States. The target of this particular assessment of points and reward functions is to rate each team based on its overall performance and strength. Indirect objectives like high accuracy on predictions or profitable betting investments based on such ratings may not be met perfectly, since separate statistical analysis on attributes is required. Also, the contribution of sports analysts, coaches, and experts in the development of the basic model is very important.

For each team, the value of each attribute is computed as an average per game by taking into account the total games played by that team before the itinerary match. Then we convert them as percentages by defining the most and least preferred value for each attribute. Here, a local scale is used which means that those values are defined by reference to the best and worst performing team in the particular attribute. In our case, each attribute belongs to Type-1 and the normalization formula to convert it between 0 and 1 is

$$x_i = \frac{\bar{t}_i - \bar{t}_i^{+}}{\bar{t}_i^{+} - \bar{t}_i^{-}}$$

where i is the attribute, t_i is the total score of all games, $\bar{t}_i$ is the average score (per game), the notation $^{+}$ denotes the most preferred value and $^{-}$ the least preferred value, and x_i is the scaled value.

Due to the use of a local scale, this means that the most and least preferred values are defined by the performance of teams. In contrast, the use of a global scale usually requires an analysis of the dataset or the experience of an expert in the field to define those reference values. Therefore, two reasons led us to choose the local scale. The first reason was to avoid any statistical analysis of the soccer dataset or any subjective assignments. The second reason was to ensure a fair comparison with other rating systems since it would be wrong to involve any information from the dataset during

modeling due to the fact that ratings will be utilized later for foresight predictions in the experimental part of Chapter 6.

Subsection 4.3.7 explains in detail the computation of attributes in our illustrative example. Below we show the second step of the method in order to apply the modeling of the method for the case of the soccer team ratings we have described.

A. Points of interest are k=6, $b_{TW} = b_{TG} = b_{TST} = b_{TS}$ ={0, 0.7, 0.77, 0.85, 0.93, 1}. The resulting ranges are:
   $C_{TW} = C_{TG} = C_{TST} = C_{TS}$ = {[0,0.7),[0.7,0.77),[0.77,0.85),[0.85,0.93),[0.93,1]}.
B. Reward points are $p_{TW} = p_{TG} = p_{TST} = p_{TS}$ = {0, 20, 20, 20, 20, 20}.
C. Cumulative reward points are $s_{TW} = s_{TG} = s_{TST} = s_{TS}$ = {0, 20, 40, 60, 80, 100}
D. We have selected the linear reward scheme to distribute points for all ranges. The piecewise function is the same for all attributes and we only show the v_{TW} which can be written as follows:

$$v_{TW}(x_{TW}) = \begin{cases} v_{TW,1}(x_{TW}), & if\ x_{TW} \in C_{TW,1} \\ v_{TW,2}(x_{TW}), & if\ x_{TW} \in C_{TW,2} \\ v_{TW,3}(x_{TW}), & if\ x_{TW} \in C_{TW,3} \\ v_{TW,4}(x_{TW}), & if\ x_{TW} \in C_{TW,4} \\ v_{TW,5}(x_{TW}), & if\ x_{TW} \in C_{TW,5} \end{cases}$$

Then, we estimate $v_{TW,1}$, $v_{TW,2}$, $v_{TW,3}$, $v_{TW,4}$, $v_{TW,5}$ based on a linear reward scheme and the relations

$$\{(b_{TW,1}, S_{TW,1}), (b_{TW,2}, S_{TW,2})\} = \{(0,0), (0.7,20)\},$$
$$\{(b_{TW,2}, S_{TW,2}), (b_{TW,3}, S_{TW,3})\} = \{(0.7,20), (0.77, 40)\},$$
$$\{(b_{TW,3}, S_{TW,3}), (b_{TW,4}, S_{TW,4})\} = \{(0.77,40), (0.85,60)\},$$
$$\{(b_{TW,4}, S_{TW,4}), (b_{TW,5}, S_{TW,5})\} = \{(0.85, 60), (0.93,80)\},$$
$$\{(b_{TW,5}, S_{TW,5}), (b_{TW,6}, S_{TW,6})\} = \{(0.93, 80), (1,100)\} \text{ respectively.}$$

Finally, the v_{TW} is written below:

$$v_{TW}(x_{TW}) = \begin{cases} 28.57 \cdot x_{TW}, & if\ x_{TW} \in [0,0.7) \\ 285.71 \cdot x_{TW} - 180, & if\ x_{TW} \in [0.7,0.77) \\ 250 \cdot x_{TW} - 152.5, & if\ x_{TW} \in [0.77,0.85) \\ 250 \cdot x_{TW} - 152.5, & if\ x_{TW} \in [0.85,0.93) \\ 285.71 \cdot x_{TW} - 185.71, & if\ x_{TW} \in [0.93,1] \end{cases}$$

All the other attributes have the same reward function. The graph of the reward function is depicted in Figure 4-9 and as we can see the reward function is similar to the

exponential function curve. Table 4-13 summarizes the attribute ranges, points, and functions.

Table 4-13: Reward points details for an attribute of the soccer team

RR	RP	CRR	RFS
[0, 0.7)	20	[0, 20)	linear
[0.7, 0.77)	20	[20, 40)	linear
[0.77, 0.85)	20	[40, 60)	linear
[0.85, 0.93)	20	[60, 80)	linear
[0.93, 1]	20	[80, 100]	linear

RR: reward ranges;

RP: reward points;

CRR: cumulative reward ranges;

RFS: reward function scheme

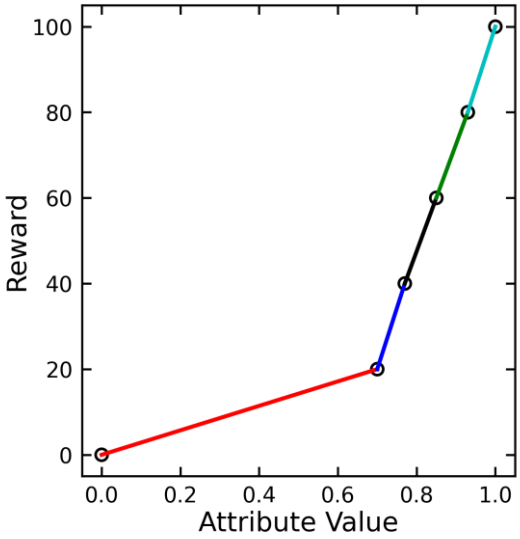


Figure 4-9: Reward function for an attribute of soccer team

Next, the weights elicitation is performed with the direct rating method and the aim is to approximate the importance of each attribute on a common basis with the total strength of the teams.

- h_{TW} = 10 (“Extremely Important”): The highest value of the scale is given to the total wins since the most important motivation for each team is to win.
- h_{TG} = 8 (“Very Important”): The number of total goals scored by each team is considered that it cannot be placed in the highest value because, in the games between a strong and a weak team, the strong team can improve its ranking position.
- h_{TST} = 6 (“Important”): The number of total shots on target shows the offensive characteristics of the team and for this reason is considered important.
- h_{TS} = 4 (“Moderately Important”): The lower value of the scale is given for the number of total shots because includes all types of shots (goals, shots in and off target).

Then, we convert them into weights by applying the (4.8) equation and we have:

w_{TW} = 0.357, w_{TG} = 0.286, w_{TST} = 0.214, w_{TS} = 0.143.

It is important to mention that the weighting scheme shown above is subjective and for this reason, we avoid applying it in the main application of this thesis. Instead,

the weights in the main application are optimized for the final objective. Finally, we highlight that the selection of attributes plays an essential role in the final ranking. For instance, imagine that there are two different fan profiles. The first one is concerned about its team's performance while the second one places more emphasis on team ethics (fair play). The attributes we have chosen and described in this section are more suitable for the first fan, while for the second the candidate attributes could be the total fouls, the total yellow cards, or the total red cards.

4.3.7 Illustrative Example

In this subsection, we apply the PointRATE method to rate and rank soccer teams by recalling the example stated in section 3.2. The details of classes, points, and reward functions are presented in subsection 4.3.6. For a better understanding, the steps to calculate Arsenal's rating are shown below:

- Step 1:

  Our alternatives are O = {Arsenal, Bournemouth, …, West Ham} and m=20.

  The attributes utilized are A = {*TW, TG, TST, TS*}, n=4 with domains:

  D_{TW} = [0,1], D_{TG} = [0,1], D_{TST} = [0,1], D_{TS} = [0,1] ⇒ D = {[0,1], [0,1], [0,1], [0,1]}.
- Step 2 and Step 3 are represented in subsection 4.3.6.
- Step 4:

  First, from Table 3-1 we summarize the results of Arsenal:

  1st match week: Arsenal – Man City, Final Outcome: 0-2, $t^1_{TST,Arsenal}$ = 3, $t^1_{TS,Arsenal}$ = 9

  2nd match week: Chelsea – Arsenal, Final Outcome: 3-2, $t^2_{TST,Arsenal}$ = 6, $t^2_{TS,Arsenal}$ = 15

  Next, as mentioned in subsection 4.3.6., the attribute values are converted into percentages by taking into account their most preferred values and the total number of games. For each attribute, the most preferred value is defined by the score of the team(s) that achieved the highest average score per game which is computed as the sum of scores in all match weeks divided by the total number of games played by the team. Similarly, the least preferred score is defined. In our example, the highest average number of total goals per game is 4 from the team Manchester City (2 in the 1st match week and 6 in the 2nd match week and then divided by 2 games played). Table 4-14 shows the most and least preferred value (best score per game column) for each attribute.

Table 4-14: The best/worst value of attributes

Attribute	Best				Worst			
	Average score per game	Match week scores		Reference Team(s)	Average score per game	Match week scores		Reference Team(s)
		1st	2nd			1st	2nd	
TW	1 (2/2)	1	1	*	0	0	0	**
TG	4 (8/2)	2	6	Man City	0 (0/2)	0	0	Cardiff
TST	11 (22/2)	8	14	Man City	1 (2/2)	1	1	Cardiff
TS	24.5 (49/2)	17	32	Man City	5.5 (11/2)	6	5	Huddersfield

*: Bournemouth, Chelsea, Liverpool, Man City, Tottenham, Watford;

**: Arsenal, Burnley, Cardiff, Fulham, Huddersfield, Newcastle, Southampton, West Ham, Wolves.

After finding the most and least preferred value for each attribute the computation of *x* for Arsenal is shown below:

$$\bar{t}_{Arsenal,TW} = \frac{t^1{}_{Arsenal,TW} + t^2{}_{Arsenal,TW}}{gn_{Arsenal}} = \frac{0+0}{2} = 0, \qquad \bar{t}_{TW}{}^{+} = 1, \qquad \bar{t}_{TW}{}^{-} = 0,$$

$$\Rightarrow x_{Arsenal,TW} = \frac{\bar{t}_{Arsenal,TW} - \bar{t}_{TW}{}^{-}}{\bar{t}_{TW}{}^{+} - \bar{t}_{TW}{}^{-}} = 0$$

$$\bar{t}_{Arsenal,TG} = \frac{t^1{}_{Arsenal,TG} + t^2{}_{Arsenal,TG}}{gn_{Arsenal}} = \frac{0+2}{2} = 1, \qquad \bar{t}_{TG}{}^{+} = 4, \qquad \bar{t}_{TG}{}^{-} = 0,$$

$$\Rightarrow x_{Arsenal,TG} = \frac{\bar{t}_{Arsenal,TG} - \bar{t}_{TG}{}^{-}}{\bar{t}_{TG}{}^{+} - \bar{t}_{TG}{}^{-}} = \frac{1}{4} = 0.25$$

$$\bar{t}_{Arsenal,TST} = \frac{t^1{}_{Arsenal,TST} + t^2{}_{Arsenal,TST}}{gn_{Arsenal}} = \frac{3+6}{2} = 4.5, \quad \bar{t}_{TST}{}^{+} = 11, \quad \bar{t}_{TST}{}^{-} = 1,$$

$$\Rightarrow x_{Arsenal,TST} = \frac{\bar{t}_{Arsenal,TST} - \bar{t}_{TST}{}^{-}}{\bar{t}_{TST}{}^{+} - \bar{t}_{TST}{}^{-}} = \frac{3.5}{10} = 0.35$$

$$\bar{t}_{Arsenal,TS} = \frac{t^1{}_{Arsenal,TS} + t^2{}_{Arsenal,TS}}{gn_{Arsenal}} = \frac{9+15}{2} = 12, \qquad TS^{+} = 24.5, \quad TS^{-} = 5.5,$$

$$\Rightarrow x_{Arsenal,TS} = \frac{\bar{t}_{Arsenal,TS} - \bar{t}_{TS}{}^{-}}{\bar{t}_{TS}{}^{+} - \bar{t}_{TS}{}^{-}} = 0.342$$

where i = team, gn_i = games number of team i, mw = match week number

$t_{i,TW}^{mw}$= win in mw, $\bar{t}_{i,TW}$= average win (per game),

$t_{i,TG}^{mw}$= total goals in mw, $\bar{t}_{i,TG}$= average goals (per game),

$t_{i,TST}^{mw}$= total shots on target in mw, $\bar{t}_{i,TST}$= average shots on target (per game),

$t_{i,TS}^{mw}$= total shots in mw, and $\bar{t}_{i,TS}$= average shots (per game).

The notation $^{+}$ denotes the most preferred value and $^{-}$ the least preferred value. Finally, we compute ratings based on (4.10) as follows:

$$r_{Arsenal} = w_{TW} \cdot v_{TW}(x_{Arsenal,TW}) + w_{TG} \cdot v_{TG}(x_{Arsenal,TG}) + w_{TST} \cdot v_{TST}(x_{Arsenal,TST}) + w_{TS} \cdot v_{TG}(x_{Arsenal,TS})$$

$$= 0.357 \cdot v_{TW}(0) + 0.286 \cdot v_{TG}(0.25) + 0.214 \cdot v_{TG}(0.35) + 0.143 \cdot v_{TG}(0.342)$$

$$= 0 + 0.286 \cdot 28.57 \cdot 0.25 + 0.214 \cdot 28.57 \cdot 0.35 + 0.143 \cdot 28.57 \cdot 0.342 = 5.58$$

Similarly, we calculate the ratings for the other teams, and the final rating and ranking results generated by PointRATE for example of section 3.2 appear in the following table:

Table 4-15: PointRATE rating and ranking results

Team	Rating	Rank	Team	Rating	Rank
Arsenal	5.5800	12	Leicester	10.2470	10
Bournemouth	43.3351	6	Liverpool	51.6541	3
Brighton	8.5768	11	Man City	100.0000	1
Burnley	4.5596	17	Man United	11.9710	8
Cardiff	1.1815	19	Newcastle	4.5757	16
Chelsea	52.2825	2	Southampton	4.9141	13
Crystal Palace	10.6498	9	Tottenham	50.5371	4
Everton	13.1042	7	Watford	45.3974	5
Fulham	4.6670	15	West Ham	3.0881	18

Table 4-16 shows the results of ranking correlation with the other systems. As we can see there is a strong relationship with the Win-Loss system due to the high weight given to the *TW* attribute. Also, a high correlation with AccuRATE can be explained by the inclusion of the attributes *TST* and *TS* in the modeling of PointRATE for the soccer team ratings. The reason for low *tau* values in the comparisons with Massey and ODM is possibly due to the insufficient number of games. The GeM also has a great sensitivity at the beginning of the season and this may affect the results. Additionally, the weights applied in the GeM are different from the weights applied in PointRATE. Specifically, as pointed out in subsection 3.3.7, the GeM ratings are generated by applying equal weights. Further research for their ranking correlation has been conducted in section 4.4.

Table 4-16: Kendall's tau and p-values for PointRATE

	WL	Colley	Massey	Elo_{win}	Elo_{point}	Keener	ODM	GeM	Accu RATE
tau	0.824	0.621	0.211	0.686	0.758	0.8	0.253	0.295	0.812
p-value	***	***	0.209	***	***	***	0.128	0.074	***

*** p-values of each pair < 0.001

The rating and ranking results produced by PointRATE for the illustrative example can be applied to hindsight and foresight predictions. The results are depicted in Table 4-17 and as was expected the hindsight accuracy is higher than foresight. However, the number of games is very small and no comparison or clear conclusion can be drawn.

Table 4-17: Hindsight and Foresight prediction accuracy of PointRATE

Rating Method	Hindsight Accuracy		Hindsight Correct Games		Foresight Accuracy		Foresight Correct Games	
	RANK	MLE	RANK	MLE	RANK	MLE	RANK	MLE
PointRATE	0.80	0.85	16	17	0.7	0.6	7	6

4.3.8 Sensitivity Analysis

The sensitivity analysis aims to test how the weight of attributes affects the final ranking list. The dataset used is the EPL for the seasons from 2005/06 to 2017/18. By changing the importance of one attribute each time which also affects the final weights of the other attributes, we examined how sensitive the method is to changes in the weight of attributes. In each change of attribute importance, the resulting ranking list is compared to the initial ranking list with Kendall's tau rank correlation coefficient.

The sensitivity analysis is conducted on each season separately in three different periods similar to the sensitivity test performed in the AccuRATE method. Those periods are: at the beginning of the season (1^{st} - 3^{rd} match weeks are included), at the first half of the season (1^{st} - 19^{th} match weeks), and for the entire season (all match weeks are included). The results of the sensitivity analysis are depicted in Table 4-18, Table 4-19, and Table 4-20 for the beginning, the first half, and the entire season respectively.

In order to make the results clearer we explain the result (*TW*,7)=0.97 in Table 4-20. The *TW* refers to the attribute and the value of 7 means that the importance of the attribute has a value of 7 on the direct rating scale. Especially, the h_{TW} importance value is changed to 7 while all the other attributes' importance remains the same (h_{TG}=8, h_{TST}=6, h_{TS}=4). Thus, we compare the modified ranking list (h_{TW}=7, h_{TG}=8, h_{TST}=6, h_{TS}=4) with the initial (h_{TW}=10, h_{TG}=8, h_{TST}=6, h_{TS}=4) at the end of each season. The comparison results of each season are measured by Kendall's tau metric and then the average *tau* is computed from all the seasons which in our case is 0.97.

Table 4-18: Changes in the weights at the beginning of the season

Attribute	Average tau									
Scale:	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
TW	-	0.99	0.98	0.97	0.95	0.94	0.91	0.89	0.87	0.83
TG	0.98	0.99	-	0.99	0.97	0.96	0.94	0.93	0.91	0.89
TST	0.95	0.96	0.98	0.98	-	0.98	0.97	0.95	0.93	0.9
TS	0.91	0.92	0.93	0.94	0.96	0.98	-	0.98	0.96	0.92

Table 4-19: Changes in the weights in the first half of the season

Attribute	Average tau									
Scale:	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
TW	-	0.99	0.98	0.97	0.96	0.94	0.93	0.92	0.9	0.89
TG	0.99	0.99	-	0.99	0.98	0.97	0.96	0.95	0.93	0.91
TST	0.96	0.97	0.98	-	0.99	0.99	0.98	0.96	0.94	0.93
TS	0.94	0.95	0.96	0.96	0.97	0.98	-	0.99	0.97	0.95

Table 4-20: Changes in the weights at the end of the season

Attribute	Average tau									
Scale:	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
TW	-	0.99	0.98	0.97	0.96	0.95	0.94	0.92	0.91	0.89
TG	0.99	0.99	-	0.99	0.99	0.98	0.97	0.96	0.95	0.94
TST	0.97	0.98	0.99	0.99	-	0.99	0.98	0.96	0.95	0.93
TS	0.95	0.96	0.97	0.97	0.98	0.99	-	0.99	0.97	0.95

From the above tables, we conclude that a large change in the weights can lead to greater sensitivity which is reasonable. An example of a large change is when the *TW* importance changed from 10 (Extremely Important) to 1 (Slightly Important). As we can observe for this change the average *tau* values are lower compared to smaller modifications and in particular, the values are 0.83, 0.89, and 0.89 for the beginning, the first half, and at the end of each season respectively. Furthermore, we conclude that small modifications affect the ranking results to a small degree. Another important observation

is that more games offer more stability to the ranking vectors concerning weight changes in attributes. This can be seen from the mean values, the associated standard deviation, and the minimum average *tau* that are depicted in Table 4-21. Note that minimum, maximum, average, and standard deviation values are computed from all average *tau* values. For example, the average value = 0.97 at (PointRATE, Full period) in Table 4-21 is computed as an average of the values of Table 4-20.

The most sensitive attribute in large changes seems to be the *TW*. This can be interpreted due to the maximum number of wins which is defined by the number of games and as a result, the rate of wins has different behavior over the games for each team compared with the other attributes. In contrast, the other attributes can be improved even if a team lost a game. This observation can be explained by considering a hypothetical example of the following two matches:

- TeamA, TeamB: Final Outcome: 2-0
- TeamC, TeamD: Final Outcome: 3-2

Although TeamA was the winner and TeamD was the loser, they scored the same number of goals in two different games. Thus, this gives the possibility for the weaker team in a match to improve its rating after scoring a satisfactory number of goals, shots, and shots on target. However, the importance of the *TW* attribute plays the main role in the ranking formation.

The next step is to compare the results with other methods. Specifically, the sensitivity analysis is repeated in the same way, under the same weighting scheme and modifications in weights for two other methods. The first method selected is the WSM while the second method is the GeM. The first method was selected with the aim of simulating its behavior as a baseline to our method since PointRATE is an alteration of WSM. The GeM is selected because it takes into account multiple statistics of teams and weight is assigned to each one of them to generate rankings. Especially for GeM, the voting schemes and damping factor are those applied in our illustrative example in 3.3.7. The results of those methods are presented in Table 4-21.

The first comparison is between PointRATE and WSM, where we observe that the differences are very small. Nevertheless, PointRATE seems to have less sensitivity at the beginning of the season in large modifications in weights if the minimum values are compared (0.83>0.8). The second comparison is between PointRATE and GeM. As we notice the GeM has greater sensitivity in all periods of the season where the average and

minimum values are lower. Notably, the GeM has the most sensitivity at the beginning of the season. Also, the standard deviation values of GeM are higher compared to those of PointRATE.

Table 4-21: Comparison results with Weighted Sum Method and GeM

Method	Period	Min	Max	Average	Std
PointRATE	Begin	0.83	0.99	0.94	0.04
	First Half	0.89	0.99	0.96	0.03
	Full	0.89	0.99	0.97	0.03
Weighted Sum Method	Begin	0.8	0.99	0.94	0.04
	First Half	0.89	0.99	0.96	0.03
	Full	0.89	0.99	0.96	0.03
GeM	Begin	0.69	0.99	0.92	0.07
	First Half	0.83	0.99	0.95	0.04
	Full	0.85	0.99	0.96	0.03

4.3.9 Conclusions

In this section, we outline PointRATE, a rating system that is based on MAUT/MAVT and uses points to model user's preferences. Initially, the detailed steps are presented. Next, we examine the possibility of applying the method to rate and rank soccer teams. The modeling of the method for the soccer team ratings is applied in our illustrative example from section 3.2. Then the rankings generated are compared to the rankings from the other methods and they are also utilized for hindsight and foresight predictions. The method's performance in terms of predictive ability and investment potential is examined in-depth in Chapter 6 where a comparison is conducted with other methods. Also, the method is applied in the context of financial management and optimization in section 7.3.

Next, the sensitivity analysis conducted exhibited stable results in the case of EPL soccer games. Based on the findings, it can be inferred that the most sensitive attribute is the Total Wins (*TW*). Also, the results are similarly stable compared to those of the WSM and seem stable enough when they are compared with those of the GeM in the three periods of the sports season we tested.

One limitation is the lack of consideration for real-world user utility in the modeling of the method in soccer team ratings. Another weak point is that all attributes

are modeled using the same function. However, this was done to simplify modeling and keep it as unbiased as possible. One improvement would be to incorporate experts, sports analysts, and bettors in order to model attributes effectively. An additional consideration could be to take advantage of other capabilities of the MAUT/MAVT approach to achieve a more comprehensive solution.

Our main conclusion is that even closely related items, (e.g., soccer teams in terms of their strength and soccer teams from an economic perspective) require completely different attributes, reward functions, and weights for attributes, in order to rate them effectively. Thus, it is necessary to develop different models for different item types and objectives. Moreover, when the ratings are intended to be utilized as a part of another process, it is possible to define ranges for the inputs in order to optimize ratings according to the final objective. Although this enables the support of more complex decision processes, it should be utilized when such processes are mandatory, in order to eliminate further complexity in the design and implementation of the method.

4.4 Comparison with Other Rating Systems

The purpose of this section is to compare the ranking lists of our proposed methods with those of other established methods in order to examine if we are moving in the same direction, i.e., to find the top-performing team. The similarity of ranking lists is evaluated in pairs with Kendall's tau distance. The comparison results can be seen as a first estimation of the diversity between ranking lists. This also affects the predictions that will be made in the accuracy and profit-oriented approaches, which are explained in Chapter 6. In other words, the pairs that do not have a strong correlation in their ranking lists, probably also produce different predictions.

The parameters selected for the systems are described in subsection 6.7.6 of our main application. In particular, are almost the same as those used in our illustrative example. It is important to highlight that in the rating list comparisons, for simplicity and fairness comparison, we have used the same weighting scheme in the statistics of GeM and attributes of PointRATE. The weighting scheme is represented in subsection 4.3.6. Also, for the Elo, the choice of parameters is those suggested by World Football Elo Ratings (EloRatings, 2023), K=40, ξ=400 but without taking into account the home-field advantage (HA=0).

The values under the main diagonal show the average *tau* value calculated from the comparison between ranking lists for the seasons 2005 to 2018. We have to note that all p-values for each comparison pair are less than 0.001 (p-value < .001) for all sports seasons. Therefore, this indicates that we reject the null hypothesis (*tau*=0) for each pair per season. Additionally, as we observe in the table below, average *tau* values show a strong correlation between ranking lists. The most correlated ranking lists are between:

1. (AccuRATE, Massey), *tau* = 0.918
2. (AccuRATE, Colley) or (Offense-Defense, Massey), *tau* = 0.895
3. (Colley, Win-Loss), *tau* = 0.881
4. (Elo-Point, Elo-Win), *tau* = 0.873
5. (AccuRATE, Keener) or (AccuRATE, Offense-Defense) *tau* = 0.871

Table 4-22: Average tau values of ranking lists for the EPL 2005-2018 seasons

	WL	**Colley**	**Massey**	**Elo$_w$**	**Elo$_p$**	**Keener**	**ODM**	**GeM**	**Accu RATE**	**Point RATE**
WL	1.000	***	***	***	***	***	***	***	***	***
Colley	0.881	1.000	***	***	***	***	***	***	***	***
Massey	0.817	0.841	1.000	***	***	***	***	***	***	***
Elo$_{win}$	0.778	0.819	0.748	1.000	***	***	***	***	***	***
Elo$_{point}$	0.758	0.783	0.782	0.873	1.000	***	***	***	***	***
Keener	0.821	0.857	0.856	0.772	0.778	1.000	***	***	***	***
ODM	0.803	0.827	0.895	0.748	0.781	0.861	1.000	***	***	***
GeM	0.767	0.78	0.761	0.728	0.74	0.777	0.785	1.000	***	***
AccuRATE	0.848	0.895	0.918	0.783	0.794	0.871	0.871	0.78	1.000	***
PointRATE	0.756	0.729	0.718	0.67	0.665	0.702	0.715	0.794	0.727	1.000

*** All p-values of each pair per season < 0.001

The analysis of Kendall's tau distance metric revealed that all pairs have a significant correlation. Thus, the conclusion is that there are small differences in the final ranking lists produced by the methods.

With regard to the AccuRATE method, the ranking results seem to have a strong correlation with the results of the other methods. The strongest is with the Massey method due to the fact that both methods use the margin of victory. This can be explained because the *k* ratio (*TST/TS*) plays a role in the rating only if the margin of victory is more than one goal. As we will show later in section 6.2, there is a high

percentage of games that ended with a total of one goal, i.e., 1-0 or 0-1. This percentage is the third possible outcome with 17.7% which means that the difference of one goal is higher than this percentage because if two teams scored in total more than one goal then it is also possible to have one goal difference (e.g., 3-2, 2-1).

The results related to the PointRATE method indicate that there is a correlation with the rankings generated by the other methods. In addition, the correlation with GeM is stronger because both methods use the same team statistics and weighting scheme.

After the ranking correlation results, our final step is oriented to examine the differences if we apply a different weighting scheme in PointRATE and GeM methods. Especially, our intention here is to make comparisons with equal weights. Two comparisons are conducted. First, we compare the PointRATE method by applying the weighting scheme in subsection 4.3.6 with equal weights. Second, we compare the GeM in the same way. The results are depicted in Table 4-23 and Table 4-24.

Table 4-23: Comparison PointRATE - EW

	PointRATE	**PointRATE-EW**	
PointRATE	1	***	
PointRATE-EW	0.933	1	

Table 4-24: Comparison GeM - EW

	GeM	**GeM-EW**	
GeM	1	***	
GeM-EW	0.94	1	

EW: Equal Weights

*** All p-values of each pair per season < 0.001

In the comparison with equal weights, Also, based on their p-values (<0.001) we reject the null hypothesis (*tau*=0) and we conclude that there are small differences in the final ranking lists of pairs, which indicates that methods are relatively stable.

Finally, we can conclude that the rankings generated by our proposed methods have similarities with the rankings of the other rating systems tested. This is encouraging and shows that we are moving in the right direction - to find the best-performing team - compared with the rankings of the other methods.

4.5 Conclusions

This chapter describes our proposed rating systems. The AccuRATE is developed for rating soccer teams and the PointRATE is an alteration to the WSM and MAUT/MAVT methods. The main characteristics of AccuRATE are:

- It is oriented to rate and rank soccer teams.
- It takes into account game outcomes, margin of victory, and shooting accuracy.

- It is an adjustive rating system according to the categorization of (Stefani R. T., 1999; Stefani & Pollard, 2007) which means that the system modifies a team's rating by either increasing or decreasing it, based on the game's final outcome, and the goals and shots scored in a game.

As for PointRATE, the main characteristics are:

- It is capable of incorporating multiple attributes as inputs to generate ratings.
- It considers the utility/value function of the user.
- It is a generalized approach that rates items.

In order to evaluate the stability of ranking results of our proposed methods, we have employed data from EPL games and conducted a sensitivity analysis by modifying inputs *k*, *d* for AccuRATE and weights for PointRATE. From the results obtained, we can conclude that only large-scale modifications affect the rankings. Furthermore, as we observed, the rankings can be affected more when there is a small number of games and this happens at the beginning of the season. In particular, the sensitivity results for AccuRATE suggest that *d* is more sensitive than *k*. It is also concluded that each team in order to maintain its ranking position will have to achieve good shooting accuracy in addition to goals scored. As for the PointRATE is concluded that significant changes in the weights can lead to greater sensitivity.

We have also examined the methods' ability to rank the EPL teams by comparing their ranking results with established rating systems. The similarities we have identified among the results showed a positive indication that the proposed methods perform well in team rankings based on their performance. However, the fact that a method may rate/rank satisfactorily does not guarantee that it could be utilized effectively for predictions or betting purposes. This motivates us to evaluate them for their efficiency in predictions and profitability. Therefore, we have tested them in our main application (Chapter 6) where they were also evaluated against the other popular methods discussed.

Overall, the proposed methods have the potential to be applied in other similar contexts where the ratings and rankings are important. Researchers, decision-makers, data scientists, sports analysts, coaches, bettors, and other similar groups can find proposed methods to be useful tools. Assumptions and weaknesses for those methods as well as possible improvements that could be made are discussed in their conclusion sections (4.2.7, 4.3.9).

5 - Theoretical Background of Machine Learning Techniques

5.1 Introduction

This chapter presents several techniques and methods of machine learning. All the topics discussed here will be utilized in the subsequent chapters of this dissertation. The chapter is structured in the following manner. In the first place, the classification algorithms of machine learning, evaluation metrics, and hyperparameter tuning are studied. Then, the cost-sensitive learning is introduced and following this, we will cover the "Binary vs Multi-class" classification concept. Next, we discussed how machine learning techniques and rating methods can be combined. Finally, this chapter closes with some conclusions.

5.2 Machine Learning and Classification

Machine learning (ML) is a subfield of Artificial Intelligence that deals with the development of algorithms and techniques that allow machines to learn from data, make predictions, and improve their performance over time (Sen, Hajra, & Ghosh, 2020). Machine learning techniques can be broadly categorized into distinct types based on the nature of the training data: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. In this thesis, we focus on supervised learning where an algorithm is used to "learn" relations between input data and output labels. The machine learning algorithm can then generalize the relations and predict outputs based on inputs not present in the training set, with the highest possible accuracy (Kyriakides, Talattinis, & Stephanides, 2017). In supervised learning, classification is the approach when the output labels are categorical and regression is when output labels are continuous values (Sen, Hajra, & Ghosh, 2020).

Conventional classification procedures are fundamental to data mining. They have been used for decades in research areas such as machine learning and statistics. The functions of a classifier are to be trained from a set of unknown objects for which class labels have been defined and to be used in making predictions of these classes to a new set of objects. In such cases, the results are assessed by analyzing various metrics such as the accuracy (i.e., the ratio of correct to total predictions). There are various well-known

and well-understood techniques and algorithms that are extensively applied for predictive modeling. In this section, we provide a brief description of 7 basic classifiers that are utilized in the next chapters of this thesis.

5.2.1 Naive Bayes

Naive Bayes (NB) classifier is a fairly simple process and easy to implement algorithm (Hand & Yu, 2001). It is based on two types of probabilities that can be calculated directly from the training data: (1) the probability of each class, and (2) the conditional probability for each class. The probability model can be used to make predictions for new data using Bayes Theorem. A general formula of Bayes Theorem is

$$P(y \mid X) = \frac{P(X|y)P(y)}{P(X)},$$

where *P(y|X)* is the posterior probability of *y* given *X*, *P(X|y)* is the likelihood of *X* given *y*, *P(y)* is the prior probability of *y*, *P(X)* is the prior probability of *X*. In the context of Naive Bayes, *y* is the class variable and $X=\{x_1, x_2,\ldots, x_n\}$ is the vector of input *n* features. Assuming that features are independent we have the following:

$$P(y \mid X) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

Then, the prediction is based on the Maximum a Posteriori (MAP) estimation, where the class with maximum posterior probability over all classes is chosen:

$$\hat{y} = \arg\max_{k \in y} P(y_k) \prod_{i=1}^{n} P(x_i \mid y_k),$$

where $\hat{y}$ is the prediction.

Furthermore, Gaussian Naive Bayes is a variant of Naive Bayes, that assumes the likelihood of the features to be Gaussian (normally distributed). The feature probability for feature x_i is given as follows:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi{\sigma_y}^2}} e^{-\left(\frac{x_i-\mu_y}{{\sigma_y}^2}\right)},$$

where μ_y is the mean of feature *i* for class *y*, and σ_y is the standard deviation of feature *i* for class *y*. Those two parameters are estimated by Maximum Likelihood Estimation.

Naive Bayes can be considered a special type of Bayesian network, relying on the assumption that the attributes are independent and that no other attributes influence the predicted class. It is also considered a computationally fast and surprisingly powerful

technique that performs well in most cases, especially in a large range of complex problems. Another advantage of the classifier is the small amount of training data that is needed in comparison to other algorithms. On the other hand, the assumption of independent attributes may not accurately reflect real-world data. Despite its independence assumption, there are studies (Domingos & Pazzani, 1997; Friedman, Geiger, & Goldszmidt, 1997) that show that Naive Bayes performs well in real-world problems. It is remarkable that the research (Rish, 2001) conducted by Rish demonstrates that the level of feature dependencies is not directly related to the accuracy of Naive Bayes.

5.2.2 Logistic Regression

Logistic Regression (LR) is a popular technique that models the probability of an event. In particular, the output of a linear equation is converted into a probability using the logistic function. If our linear equation of n independent variables X_i, and Y is the dependent variable that is either 0 or 1, then, the logistic function used is the following:

$$p(X) = \frac{1}{1 + e^{-z}}, \qquad z = a + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n,$$

where $p(X) = p(Y = 1 \mid X_1, X_2, \ldots, X_n)$ is the predicted probability for the binary outcome and z is the linear combination of independent variables X_i.

The formula for the logistic function can also be derived from the log-odds:

$$odds = \frac{p(X)}{1-p(X)} \text{ and } \ln\left(\frac{p(X)}{1-p(X)}\right) = a + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n.$$

The logistic function maps any input value to output between 0 and 1 and the coefficients $\beta_1, \beta_2, \ldots, \beta_n$ are estimated using Maximum Likelihood Estimation.

Some advantages of Logistic Regression include that it is simple and effective. It provides probabilities and the classification of a new data instance is based on the set of independent variables which can be both categorical and continuous. Moreover, its coefficient can be directly interpreted. Despite that the method has several advantages, it is considered less accurate and less effective in a small amount of data. A further limitation of Logistic Regression that must be acknowledged is the linearity assumption between independent variables and log-odds, which may not hold in all real-world problems.

5.2.3 Decision Trees

A Decision Tree (DT) is a type of machine learning algorithm that is primarily utilized for regression and classification. It is a tree-like graph that is used to describe all possible decisions based on certain conditions, leading to a prediction or outcome. Algorithms usually construct Decision Trees top-down, by choosing the variable that best splits the data into appropriate sets. The structure includes a root node, internal nodes that represent test conditions on attributes, branches that represent the outcome of test conditions, and leaf nodes that represent an outcome/class (categorical or continuous value).

Popular Decision Tree algorithms are the ID3, C4.5, and CART. The ID3 (Iterative Dichotomiser 3) generates a tree by dividing the data into subsets based on the attribute with the highest information gain, using a recursive approach (Quinlan, 1986). The C4.5 is an extension of ID3 and is more advanced (Quinlan, 1993). The CART (Classification and Regression Trees) algorithm can be applied to classification or regression problems and builds a binary tree by recursively dividing the data into two groups according to the best split.

Decision Trees can be interpreted and visualized, which can aid in the comprehension of the model. Additionally, their interpretability makes them useful for explaining how the model reaches its predictions. Another advantage is that Decision Trees are nonparametric methods and they can also handle both categorical and numerical data (Rokach & Maimon, 2005). However, the extraction of the optimal tree is hard to calculate. Another limitation of Decision Trees to be considered is the overfitting problem where the model lacks generalization to new data. This issue is encountered by decision-tree learners that are prone to creating overly complex models that may lead to poor generalization on unseen data. To address this challenge, it is essential to implement techniques such as pruning, setting a minimum lower bound on the number of samples per leaf node, or specifying the maximum depth for the tree.

5.2.4 Random Forest

Random Forest (RF) is a specific type of non-parametric ensemble machine learning approach. The basic idea behind Random Forest is the creation of multiple Decision Trees (tree predictors) and combining their predictions in order to achieve more accurate and stable results. In greater detail, Random Forest is an approach that involves

a process of selecting a random subset of features, subdividing the dataset into smaller subsets, and assigning them to individual trees. Each tree is constructed using the values of an independently sampled random vector with the same distribution for all the trees in the forest (Breiman, 2001). Contrary to traditional bagging where a random vector of instances is sampled from the dataset, in Random Forest the data is sampled as well as the features. This means that Random Forest introduces additional randomness to the feature selection for each Decision Tree in Random Forest while in bagging all available features are used for each Decision Tree. The final predictions are obtained by aggregating all the individual trees' outputs in the forest to arrive at the final prediction.

Random Forest is considered a robust method with respect to noise and helps to reduce overfitting (Breiman, 2001). However, Random Forest is less interpretable compared to Decision Trees. In other words, when dealing with a large number of trees it is difficult to interpret and understand the model. Also, large datasets or a high number of trees can lead Random Forest to be computationally expensive and require significant memory usage.

5.2.5 Neural Networks

Neural Networks (NNs) are inspired by the way that human brains work where the aim is to learn and make predictions. They are networks of interconnected nodes called neurons where the nodes accept a set of inputs and produce output results after processing.

Neural Networks are usually organized in layers. Multilayer Perceptron (MLP) is a Neural Network architecture that consists of multiple layers of neurons and is commonly used in supervised learning classification or regression. A layer is a group of nodes that are not connected with any node of the same group. The input data is presented to the input layer, one or more "hidden" layers process the data via a system of weighted connections, and finally, the presentation layer outputs the results. Each node typically performs the following main functions. The node receives inputs from the previous layer or input layer and sums all the inputs multiplied by their respective weights. Then, the weighted sum and a bias term are passed through an activation function that transforms the value. Some widely used functions include the logistic function: $f(z) = \frac{1}{1+e^{-z}}$, the rectified linear unit (ReLU) function: $f(z) = \max(0, z)$, the hyperbolic tangent function (tahn): $f(z) = tanh(z)$, and the linear function $f(z) = z$.

The output of the activation function is ultimately the output of the node which is either passed on to the next layer of nodes or to the output layer.

Backpropagation is a significant learning algorithm used in Neural Network training and propagates the error from the output layer to the input layer (layer by layer) and updates weights in MLP. This process is performed iteratively until the error is minimized. A typical Neural Network starts with random weights and tries to adjust its neurons' weights until has reached a certain level of performance and any further change makes it less accurate or does not increase its accuracy any further. During the training of a Neural Network, the optimizer is responsible for adjusting the weights of the connections between neurons to minimize the error between predicted and actual outputs. Among the several optimization algorithms (optimizers), Stochastic Gradient Descent (SGD), and Adaptive moment estimation (Adam) are widely used.

Despite their ability to easily build models and their adaptability, the NN-based methods have several weaknesses that can limit their performance. The first weakness is that can easily overfit to the training data (e.g., if they are too complex), which can have the effect of poor performance to unseen data. The existence of hidden layers makes it difficult to monitor the training phase and thus, we cannot have a clear view during the process. As a result, they are often viewed as "black boxes". Also, large quantities of training data are often necessary for Neural Networks in order to achieve good performance, which can be a limitation for some applications. Furthermore, their computational complexity makes them computationally expensive, especially for large datasets or complex architectures.

5.2.6 Support Vector Machine

Support Vector Machine (SVM) is a pattern recognition method that can be employed for both classification and regression problems and it is based on statistical learning theory. SVM was developed by Cortes and Vapnik (Cortes & Vapnik, 1995) and is based on the idea of finding the best separation boundary between the classes. In particular, the objective of the Support Vector Machine algorithm is to find an optimal hyperplane in an N-dimensional space, where *N* is the number of features, that distinctly classifies the data points. In order to select the optimal hyperplane, SVMs utilize a subset of training data points that lie on the edge of each class called support vectors. Following, they select the hyperplane that has the maximum possible margin from the

support vectors. Also, SVM can map the data into a higher dimensional feature space by utilizing a kernel function that can allow more effective separation of the classes.

A major advantage of SVM is that it can handle high dimensional data which means that it can be applied to datasets with a large number of features. Moreover, the fact that SVM uses only a subset of training data points in the decision function, makes it memory efficient. Also, it is a versatile method as different kernel functions can be used for the decision function. On the other hand, SVM has the disadvantage that is difficult to select the kernel function and its hyperparameters, which implies that the performance of the SVM is sensitive to the choice of the kernel function.

5.2.7 K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a nonparametric machine learning algorithm that can be used for classification or regression. KNN is a lazy learning algorithm as it does not require any training phase. Instead, it stores all the training data, thus the model representation for KNN is the entire training dataset. A new data point is predicted by searching through the training set for the *K* most similar instances (the neighbors) and summarizing the output variable for those *K* instances. In classification, this is usually achieved through majority voting, i.e., selecting the class that is most prevalent amongst the *K* nearest neighbors. For example, if *K*=5 then the algorithm takes into account the 5 closest points to classify based on the majority of their values. In regression, the output of a new data point is based on the average of the output values of its K-neighbors. In order to determine the nearest neighbors of each data point, KNN uses a distance metric such as the Euclidean distance, Manhattan distance, Minkowski distance, etc.

KNN is very simple, effective, and easy to implement. However, KNN can be computationally expensive in large datasets. Also, another limitation is that is sensitive to irrelevant features because it considers all features of equal importance in the computation of distance. Furthermore, the performance of KNN is sensitive to the number of neighbors and the selection of distance.

5.3 Evaluation Metrics

The evaluation metrics are used to measure the performance of the classifier. In the first place, we introduce the Confusion Matrix in order to explain the metrics of Accuracy, Precision, Recall, and F1-score. Then, the Ranked Probability Score (RPS) used for probabilistic forecasts in multi-class problems is explained.

- Confusion Matrix: contains information about actual and predicted classifications (Kohavi & Provost, 1998). The strength of a confusion matrix lies in the fact that it identifies the nature of the classification errors as well as their quantities. It is commonly used to assess the performance of a classification approach. The confusion matrix of a binary problem of 2 classes: Negative and Positive is shown in Table 5-1.

Table 5-1: Confusion Matrix

	Predicted Negative Class	**Predicted Positive Class**
Actual Negative Class	*TN*	*FP*
Actual Positive Class	*FN*	*TP*

TN: True Negative is the number of instances that were correctly classified as Negative.

FN: False Negative is the number of instances that were incorrectly classified as Negative.

TP: True Positive is the number of instances that were correctly classified as Positive.

FP: False Positive is the number of instances that were incorrectly classified as Positive.

The sum of the cells of the confusion matrix is the total number of instances evaluated by the classifier.

- Accuracy: represents the overall performance of the model and is computed by the ratio of the number of correct predictions to the number of total predictions. It ranges from 0 to 1 and a higher score indicates more accurate predictions. Accuracy is more useful when the classes are balanced. It is computed as follows:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} = \frac{TN + TP}{TN + FN + TP + FP}$$

- Precision: represents the proportion of the number of correct positives to the predicted total positives. Precision ranges from 0 to 1 and a higher score indicates that the model is accurately predicting positive instances. It is computed as follows:

$$Precision = \frac{TP}{TP + FP}$$

- Recall: represents the proportion of the number of correct positives to the number of actual positives. Precision ranges from 0 to 1 and a higher score indicates that the model is accurately identifying positive instances. It is computed as follows:

$$Recall = \frac{TP}{TP + FN}$$

- F1-score: represents the balance between precision and recall and is computed as a harmonic mean of precision and recall. F1-score ranges from 0 to 1, where the highest score of 1 indicates perfect precision and recall. It is computed as follows:

$$F1 - score = 2 \times \frac{(Precision \times Recall)}{Precision + Recall}$$

- Ranked Probability Score (RPS): The ranked probability score is used to assess the performance of probabilistic predictions (Epstein, 1969; Murphy, 1969; Murphy, 1971). It is computed as follows:

$$RPS = \frac{1}{k-1}\sum_{i=1}^{k-1}\left(\sum_{j=1}^{i}(p_j - o_j)\right)^2,$$

where k is the total number of outcomes, p_j is the predicted probability of j-th game outcome and o_j is the observed outcome of j-th game.

5.4 Hyperparameter Tuning

Hyperparameter tuning in machine learning is an essential and crucial task as it can significantly impact the performance and accuracy of the model. It is an optimization problem where the aim is to select the best set of hyperparameters that maximize the performance of the model. The model's performance is evaluated on validation data for a specific evaluation metric. Proper hyperparameter tuning can improve the model's generalization ability, making it better to predict unseen data.

In this section, two commonly used methods for the tuning of hyperparameters are briefly explained. Those methods are not intended only for the hyperparameter tuning of machine learning classification algorithms but also can be used in other optimization problems. The first is the Grid Search while the second is the Genetic Algorithm approach. Both methods were used in Chapter 6, while the genetic algorithms were also used in the application of section 7.3.

5.4.1 Grid Search

One of the most well-known and simple techniques used to tune hyperparameters for a given machine learning model is the grid search method. It is an exhaustive search for the selection of the best hyperparameters where a grid is created from each possible combination of all the discrete values of hyperparameters while the continuous variables

are discretized. Then each model is trained and evaluated and the hyperparameters of the best combination are selected as the optimal.

Grid search is a common and widely used method for hyperparameter tuning in machine learning. The method can detect the best hyperparameter values, however, its complexity grows exponentially at a rate of $O(n^k)$ where n is the number of distinct values and k is the number of hyperparameters (Yang & Shami, 2020). Overall, the method is computationally expensive for high-dimensional hyperparameter spaces.

5.4.2 Genetic Algorithm

Genetic Algorithm (GA) is a heuristic search technique that is utilized to find solutions to optimization problems. GA was first introduced by Holland (Holland, 1975). They exploit information to optimize the exploration of the search space with the goal of attaining improved performance. Their strategy for solving problems entails the selection of the most fit individuals from one generation to the next through the principle of natural selection. Each individual of a population within a search space is a particular chromosome that represents a possible solution and it is assigned a fitness value. It is assumed that chromosomes correspond to a possible solution to a given problem, whereas variables are analogous to genes. In hyperparameter tuning, each gene in the chromosome represents a different hyperparameter. GAs, as well as generally evolutionary computational methods, pursue a common execution procedure involving the following steps:

(1) Initialization of Population: Typically, random values are assigned to the individuals of a population in order to establish diversity across the solution space.
(2) Evaluation: The fitness function is employed to evaluate the adequacy of a solution, as it pertains to an individual of the population under examination. In this step, based on the fitness function each chromosome in the population is evaluated.
(3) Selection: A pair of chromosomes are selected as parents for the next generation. The selection of chromosomes is based on the principle that the chromosomes with better fitness values have a higher chance of being selected as a parent.
(4) Crossover: The selected pair of chromosomes are combined to form two offspring.
(5) Mutation: The two offspring are subject to mutation where a subset of genes (variables) is randomly selected and their values are changed. The primary goal of the mutation is to introduce new genetic material into a population, which allows it to

escape from suboptimal solutions. This process is performed with a small probability in order to ensure that important properties of the individuals are maintained.

(6) New population: Based on a replacement strategy, the individuals selected for the new population will be included in the next generation. Steps 3 to 6 are repeated in order to create a new population of chromosomes.

(7) Termination conditions: The procedure continues to iterate through steps 2 to 6 until a predefined condition is verified (e.g., number of iterations, time limitations, fitness level, etc.).

5.5 Cost-Sensitive Learning

The aim of this section is to provide a detailed background for Cost-Sensitive (CS) learning which we have employed in soccer outcome prediction in order to improve the performance of our proposed betting models in the next chapter.

A major task of most ML techniques and methods including the above-mentioned ones is the process of classification. What is not considered though by these methods is the cost of misclassification of the different classes. Cost-sensitive learning, as noted by (Elkan, 2001), surveys each type of emerging cost and replaces that amount with the average cost per prediction. The ultimate aim is to minimize the average cost per object.

The cost in each case is given by a specific entry in the confusion matrix (Sheng & Ling, 2009). The behavior of a classifier can be further interpreted by using a cost matrix that corresponds to the confusion matrix and provides the costs for each of the outcomes shown in the confusion matrix (McCarthy, Zabar, & Weiss, 2005). In a classification problem with K classes, the misclassification costs can be represented by a K x K cost matrix. The rows of the matrix represent the classes, whereas the columns represent the predicted classes. The on-diagonals depict the costs of correctly classified instances, while the off-diagonals depict the misclassification costs (Ting, 1998). If the positive and negative classes are labeled 1 and 0 respectively, the cost matrix of a two-class case would be configured as in Table 5-2.

Table 5-2: Cost matrix

	Predicted Negative class	**Predicted Positive Class**
Actual Negative Class	*C*(0,0) or *TN*	C(1,0) or *FP*
Actual Positive Class	*C*(0,1) or *FN*	C(1,1) or *TP*

Suppose that an entry of a matrix *C* is depicted by *(i, j)*, where *i* and *j* represent the cost of the predicting class *i* when the actual class is *j*. We can produce a general rule to define the minimum expected cost of an example *x* if *x* belongs to class *i*. Much of what follows is taken from (Michie, Spiegelhalter, & Taylor, 1994; Sheng & Ling, 2009):

$$R(i|x) = \sum_{j} P(j|x)C(i,j)\,, \tag{5.1}$$

where $R(i|x)$ in the expected cost of *x* to be classified into class *i*, and $P(j|x)$ is the probability of each class *j* to be the true class of *x*. Taking into account (5.1), a classifier will identify an example *x* as belonging to a positive class if

$$P(0|x)C(1,0) + P(1|x)C(1,1) \leq P(0|x)C(0,0) + P(1|x)C(0,1). \tag{5.2}$$

Note that a positive class is more difficult to predict than a negative one. Thus, our efforts will focus primarily on the recognition of positive instances, as their misclassification values are higher than those of negative instances.

Moreover, if we simplify and rewrite equation (5.2) to

$$P(0|x)\big(C(1,0) - C(0,0)\big) \leq P(1|x)\big(C(0,1) - C(1,1)\big), \tag{5.3}$$

then our initial cost matrix can be converted to the cost matrix shown in the table below.

Table 5-3: Converted cost matrix

	Predicted Negative class	**Predicted Positive Class**
Actual Negative Class	0	*C*(1,0) – *C*(0,0)
Actual Positive Class	*C*(0,1) – *C*(1,1)	0

As we notice the cost matrix above, has zero cost for the correct predictions. Then, by considering that $P(0|x) = 1 - P(1|x)$ and from (5.3) we have

$$P(0|x)C(1,0) \leq P(1|x)C(0,1) \Rightarrow P(1|x) \geq \frac{C(1,0)}{C(1,0) + C(0,1)}. \tag{5.4}$$

From (5.4), the threshold p^* is

$$p^* = \frac{C(1,0)}{C(1,0) + C(0,1)}, \tag{5.5}$$

where the example *x* can be classified as positive if the posterior probability of the classifier $P(1|x) \geq p^*$.

Another approach to understanding the performance is by using Receiver Operating Characteristic (ROC) curves. A ROC graph is a plot in which the axis *X*

represents the false-positive rate of the confusion matrix and the *Y* axis depicts the positive well-classified samples. Noting that, by definition, the true-positive rate represents the sensitivity of a system, while the false-positive rate indicates the probability that an error (false) is occurring, the AUC (Area Under the Curve) of the ROC curve measures the performance of the classifier. Specifically, the ROC graph encapsulates the information contained in the confusion matrix and forms an efficient tool with which to evaluate both the classifier's ability to correctly identify positive cases and its ability to determine the number of negative misclassification cases. ROC curves are generally used for binary classification problems; however, we can use them to analyze pairwise comparisons, which allows us to interpret 3-class problems such as in soccer outcome prediction.

There are two commonly accepted cost-sensitive methods: the direct learning method and the meta-learning process (Sheng & Ling, 2009). The former concerns the implementation of algorithms that are cost-sensitive oriented, while the latter concerns the generic approach with which to evaluate methods that work as a "wrapper," i.e., the methods that convert a cost-insensitive classification method into a cost-sensitive one (Sheng & Ling, 2009).

5.5.1 MetaCost Classifier

Aiming to transform a cost-insensitive classification problem into a cost-sensitive one, the MetaCost classifier (Domingos, 1999) combines the predictive ability of bagging with an accessible model for cost-sensitive prediction. Bagging is a powerful method because of its ability to produce very accurate probability estimates. The MetaCost function focuses on assigning new labels to the training data examples (i.e., relabeled data examples) with their classes that minimize the estimated misclassification cost. The new labels are defined based on the probability estimates of bagging that are used as an ensemble classifier. Next, the function discards these labels and learns a new classifier from the relabeled data. As the costs have been incorporated into the class labels, the newly generated model is able to make cost-sensitive predictions. Thus, the MetaCost classifier is unique in that it produces a single cost-sensitive classifier of the base learner which results in interpretable output. This cost utilizes all bagging iterations when reclassifying the training data, as discussed by (Domingos, 1999).

5.5.2 Cost-Sensitive Classifier

In addition to the MetaCost classification technique, the Cost-Sensitive Classifier (CSC) is an alternative cost-sensitive learning method (Witten & Frank, 2005). There are two different approaches to this method. The first approach consists of changing the proportion of each class in the training data to reflect the cost matrix. The second approach concerns the prediction of the class with the minimum expected misclassification cost. This second process involves both a learning and a testing phase where CSC generates probability estimations $P(j|x)$ (from (5.1) equation) for each test instance by using a cost-insensitive classifier. Then, CSC uses (5.5) in order to make predictions.

The fundamental structure of both the MetaCost and the Cost-sensitive classifiers is concerned with making its base learner cost-sensitive.

5.5.3 Cost-Sensitive Learning Applied to Soccer Outcome Prediction

Several publications have demonstrated various attempts to develop more accurate prediction models in sports analytics, including computational and statistical methods. Our investigations into such studies have shown that most of them try to either minimize the error rate (number of incorrect predictions) or maximize the accuracy ratio (proportion of the correct predictions). Several studies are focused on the accuracy part of the models and for this reason disregard the differences between types of misclassification errors, giving them equal weight in their analysis because they do not play a role. Nevertheless, when the predictions are utilized for betting purposes, we have to take into account the misclassification errors.

Moreover, in other real-world problems, the mispredictions are not equally costly. In fact, such an assumption could have a considerable impact on one's decision-making process. Ignoring the differences in these costs could lead to a useless model because only the most frequent types of mispredictions would be considered in the analysis, even though the less frequent types of mispredictions could also result in a substantial cost. For example, in the medical diagnosis of cancer, misdiagnosing a health patient as a cancer sufferer has less impact on the patient than diagnosing as cancer-free when the patient is ill because the latter error could result in loss of life due to a delay in treatment.

To understand this issue, various approaches have been reported with a bias toward minimizing the total cost of misclassification. Cost-sensitive classification is the

general method that is applied to minimize the expected cost. However, even though it finds wide applicability in different sectors of daily life, in the sports field slight research activity has been observed until now and one of the improvement steps of the main application of this thesis of Chapter 6 is focused on filling this need.

The cost-sensitive learning in sports outcome prediction lies in the idea that a prediction model considers costs (or benefits) and can also predict rare outcomes, rather than ordinary ones. For example, assume a soccer game with average odds (1.2, 12, 7.5) for (Home-Win, Away-Win, and Draw) respectively, and a stake of 100 monetary units. If we bet on Home-Win correctly, we win 20 monetary units but in case of a loss we need 5 times of correct predictions to balance our initial capital. On the other hand, suppose that we bet on Away-Win or Draw, our net profit will be 1100 monetary units and 650 monetary units respectively giving us room for more unsuccessful predictions. Thus, the cost or the risk of misprediction of outcomes for low-valued odds is higher. Considering that models are biased in predicting outcomes with low odds (e.g., favorite team, home team, or top-ranked team, etc.) as they tend to improve the number of correct predictions, we apply cost-sensitive methodologies in order to direct our model to also focus on high-valued odds (e.g., draw outcome, outsider team, etc.). In this way, the model is trained to be reluctant to predict low-valued odd outcomes for fear of loss, emphasizing predictions of high-valued ones that allow unsuccessful predictions in a larger amount. We acknowledge the difficulty in this endeavor, however, this is a matter of profitability.

For this purpose, we propose an example scheme of the cost matrix structure. The reasoning that led to the proposed scheme is that we consider that the potential profit from betting has an effect on our final model and therefore it must be included in the calculation of cost values. We start by introducing the following notation required for the example scheme

$$\begin{aligned} profit_{home} &= (H_{odds} - 1) \times bs_{home}, \\ profit_{away} &= (A_{odds} - 1) \times bs_{away}, \\ profit_{draw} &= (D_{odds} - 1) \times bs_{draw}, \end{aligned} \tag{5.6}$$

where bs_{home}, bs_{away}, and bs_{draw} are the betting sizes placed for the Home-win, Away-win, and Draw respectively, and H_{odds}, A_{odds}, and D_{odds} are the Home-win, Away-win, and Draw betting odds respectively. Note that *profit* and *bs* variables are expressed as monetary units.

Our proposed scheme can be interpreted as for each misclassified instance, the cost can be expected to the monetary units we have bet (*bs*). If we consider the example above, the betting size is constant for all outcomes and therefore the cost is 100 monetary units. As we can see from the table below, there are negative costs at the main diagonal which are interpreted as benefits. The benefits represent the profit we gain after betting in the case of correct prediction. This potential profit depends on the betting size and the winning odds of the outcome.

Table 5-4: Cost matrix for the case of soccer game outcome

	Predicted: Home	**Predicted: Away**	**Predicted: Draw**
Actual: Home	$-\ profit_{home}$	bs_{away}	bs_{draw}
Actual: Away	bs_{home}	$-\ profit_{away}$	bs_{draw}
Actual: Draw	bs_{home}	bs_{away}	$-\ profit_{draw}$

By applying the same procedure employed to convert the cost matrix of Table 5-2, the cost matrix of Table 5-4 will be simplified similarly to Table 5-3 which has zero cost values at the main diagonal. Due to the negative values, the conversion is made by subtracting the benefit value of each diagonal element from the other elements of the same row. Table 5-5 shows the converted cost matrix.

Table 5-5: Converted cost matrix for the case of soccer game outcome

	Predicted: Home	**Predicted: Away**	**Predicted: Draw**
Actual: Home	0	$bs_{away} + profit_{home}$	$bs_{draw} + profit_{home}$
Actual: Away	$bs_{home} + profit_{away}$	0	$bs_{draw} + profit_{away}$
Actual: Draw	$bs_{home} + profit_{draw}$	$bs_{away} + profit_{draw}$	0

For further explanation, consider the example we mentioned before:

- The betting odds are $H_{odds} = 1.2$, $A_{odds} = 12$ and $D_{odds} = 7.5$
- The profits are $profit_{home} = 20$, $profit_{away} = 1100$, and $profit_{draw} = 650$
- The $bs_{home} = bs_{away} = bs_{draw} = 100$

The resulting cost matrix for this example is depicted in Table 5-6 and demonstrates the costs according to the types of misclassification errors.

Table 5-6: Cost matrix example for soccer game outcome

	Predicted: Home	**Predicted: Away**	**Predicted: Draw**
Actual: Home	0	120	120
Actual: Away	1200	0	1200
Actual: Draw	750	750	0

5.6 Binary Classification vs Multi-class Classification

The processes discussed above were approached from the perspective of extending the analysis of cost-sensitive learning. However, these methods may also be examined as two-class or multi-class problems.

Binary classification examines problems entirely defined by two classes. On the other hand, multi-class classification is a process that uses more than two classes and aims to assign instances to one of the possible discrete classes. Many classifiers handle multi-class problems directly, such as Naive Bayes, Decision Trees, Random Forests, Neural Networks, and KNN. Two well-known approaches in the literature are suggested for the extension of the binary to the multi-class case. The first entails the training of a classifier over a particular class, taking into account that the samples of that class are assumed to be the positives, while all the remaining samples are categorized as negatives (One-vs-All or One-vs-Rest classification). The second, defined as a K-class problem, considers $K(K-1)/2$ binary classifiers and assigns samples of two classes to each of them. The goal of the K-class problem is to learn to recognize this pair of classes (All-vs-All or One-vs-One classification) (Bishop, 2006).

When assessing the binary problem from a cost-sensitive view, one sees that there are two types of costs that need to be addressed: the cost of misclassifying the first class as the second and the cost of misclassifying the second class as the first. The illustration of a two-class problem using a cost matrix is already presented in Table 5-2, whereas the three-class problem of soccer outcome prediction by considering the betting odds is discussed in subsection 5.5.3.

5.7 Combining Rating Methods and Machine Learning Techniques

The combination of rating methods and machine learning techniques is gaining traction across multiple applications. The primary objective is to achieve more efficient

and enhanced results in accordance with the intended purpose of the application. For example, if we focus on the problem of predicting the outcome of a soccer match, this can be made through rating systems and machine learning classification. In this particular case, we will replace the RANK and MLE methods with a machine learning model in order to perform the predictions. This section demonstrates how rating systems and machine learning can be combined by highlighting relevant studies and discussing potential applications with examples.

❖ Feature Engineering

This approach incorporates rating systems in the feature engineering procedure of machine learning.

- Feature Extraction

  Rating systems serve as feature extraction tools, allowing us to incorporate more information into the machine learning model. A simple way to do this is to compute ratings first and then transform them into features that can be utilized by machine learning algorithms. In other words, rating values are used as input features to machine learning algorithms. We followed this approach in our study (Kyriakides, Talattinis, & Stephanides, 2017), where in our proposed models we have used rating values as machine learning features in a model that predicts the soccer outcome of the EPL. Also, Berrar et al. (Berrar, Lopes, & Dubitzky, 2019) in their study predicted soccer outcomes with machine learning algorithms where they used rating features that are calculated by considering the performance of each team. Herbinet (Herbinet, 2018) utilized Elo ratings and other statistical attributes in machine learning methods to forecast soccer outcomes.

  The feature extraction process is depicted schematically below:

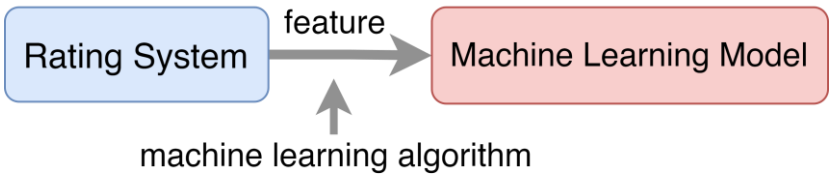


Figure 5-1: Feature extraction

- Feature selection

  Rating systems can be used to calculate the significance of machine learning features. Henni et al. (Henni, Mezghani, & Gouin-Vallerand, 2018) propose a method for unsupervised feature selection. The authors in their study determine

the importance of features and employ Google's PageRank algorithm as a centrality measure.

The feature selection process is depicted schematically below:

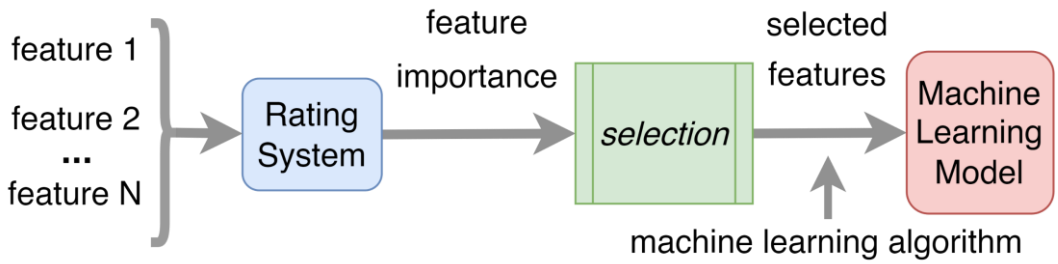


Figure 5-2: Feature selection

❖ Ensemble Model

In this approach rating and machine learning models are trained independently and then their predictions are combined. Therefore, they are used as base models in an ensemble model. For example, the sports outcome predictions of machine learning models can be combined with Rank-based predictions (i.e., based on rankings generated by rating systems).

The ensemble model is depicted schematically below:

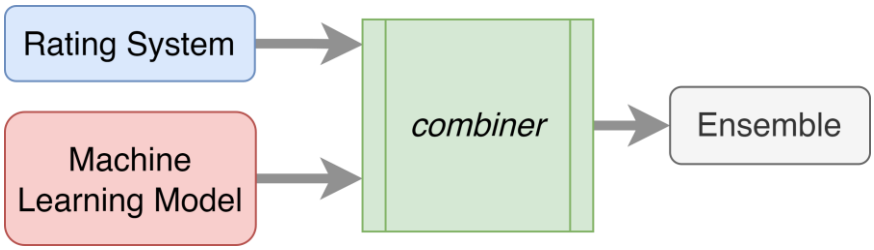


Figure 5-3: Ensemble model

❖ Target Variables (in supervised learning)

The results of rating/ranking systems (rating scores or rankings) can be used as the target output of a supervised learning task. The limitations of some rating systems render them unsuitable for use directly in specific applications, especially during real-time deployment. There are several reasons why it may not be feasible to calculate ratings directly and instead, it may be necessary to predict them. A few reasons are mentioned below:

- Complexity: Rating systems with high computational complexity are inappropriate for real-time applications as they require extensive computational resources.

- Scalability: Some rating methods are not scalable in large problems.
- Data Issues: In some cases, specific data required by the rating systems are not available. Usually, this happens in real-time applications. For example, due to data privacy, some data required are private or sensitive making the task of access to them difficult or impossible. Also, noisy or missing data is another issue in some applications because makes the rankings sensitive.

Incorporating machine learning techniques can help to address some of these challenges leading to more accurate results.

The general process of the target variable is depicted schematically below:

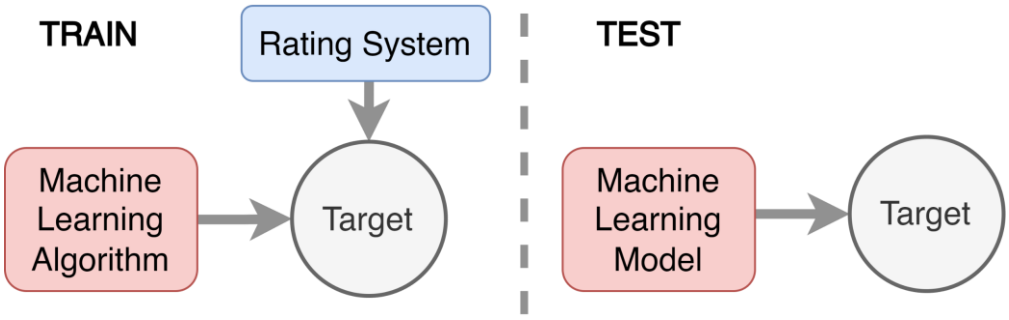


Figure 5-4: Target variables

❖ Hybrid Model

The hybrid model leverages the strength of multiple approaches to solving a problem. In this approach rating systems can be integrated into the machine learning process as a component. Examples of hybrid models are the two-stage models presented below:

- Pre-processing step

  The rating system can be utilized as a way to preprocess data or as a filter. For example, in movie recommendations, we can rate movies first in order to find their popularity, and then their rankings can act as a filter for the recommendation system. This case is explained in the application of section 7.4.

  The pre-processing step procedure is depicted schematically below:

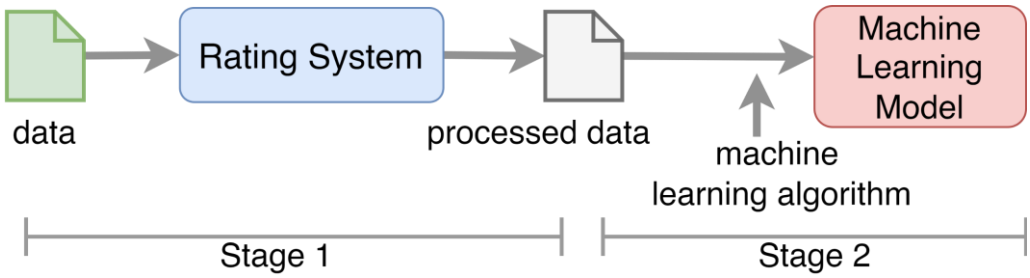


Figure 5-5: Pre-processing step

- Model Postprocessing/Model Improvement

  The rating systems can be used as a postprocessing step. One case is to re-rank the predicted rankings to improve the overall quality. For example, in product recommendation, relevant products can be found first, and then based on their popularity the final suggestion can be changed. Another case is to provide future rankings. As an example of this case, consider the predictions of the outcome or margin of victory in soccer games that can be performed in the first step, and then predictions can be utilized as input to rating systems to generate future rankings.

  The post-processing model procedure is depicted schematically below:

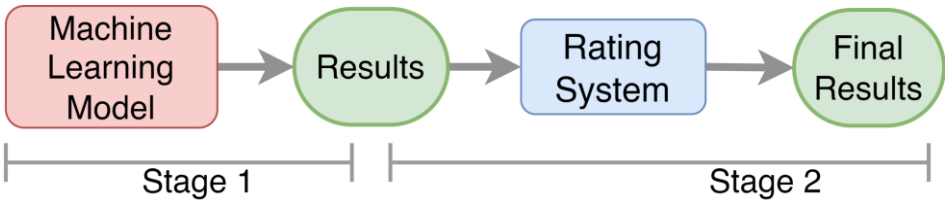


Figure 5-6: Model postprocessing/improvement

The approaches presented above can be extended more, by considering studies from other fields. Bikmukhametov et al. (Bikmukhametov & Jäschke, 2020) in their noteworthy study proposed combinations of first principles models of process engineering systems and machine learning models.

5.8 Conclusions

The focus of this chapter was on supervised learning, with a particular emphasis on the classification methods that will be utilized in the next chapter. Seven well-known classifiers and two techniques for hyperparameter tuning are briefly examined.

The significance of cost-sensitive techniques was extensively discussed and emphasized since we have employed them in the soccer outcome prediction application of this thesis to gain an advantage in betting odds. Specifically, we outlined two classifiers that take into consideration the cost of misprediction class and we have also proposed a cost matrix scheme for the soccer outcome prediction that takes into account the betting odds as benefits.

The last section demonstrated various combinations of rating systems and machine learning approaches. More emphasis was given when the ratings were utilized as input features of classifiers where we intend to make predictions and calculate the probabilities of the predicted class.

6 - Sports Outcome Prediction by Utilizing Rating Methods and Machine Learning Techniques – The EPL Case

6.1 Introduction

In the last few years, considerable attention has been paid to the forecasting of sporting event results. One of the reasons that triggered the interest is the potential profits from utilizing the predictions in sports betting. As already mentioned in section 1.1, this can be explained if we take into consideration the reports of the European Gaming & Betting Association - EGBA (European Gaming & Betting Association, 2022) in 2022 which "sports/other types of betting" is the second most popular product generating 35% (€13.6 billion) of the gross gaming revenue. Particularly, in 2021, sports betting is the first product that generates 46% (€5.3 billion) of the total online gross gaming revenue. There is no need to question why many scholars are fascinated by and try to analyze sports outcomes. Both the inherent difficulty of the task, as well as the potential reward (both psychological due to the difficulty, and financial), are irresistible forces to many personalities.

Sports betting attracts the interests of casual investors, researchers, and academics whose efforts have been devoted to proving that sports outcomes can be predictable. The problem with this approach is the fact that competitive sports such as soccer are inherently unpredictable. It is difficult to predict the outcome of any game that is played between two equally poised teams, particularly in a soccer match that is a low-scoring game. However, soccer is one of the more popular sports to bet on in Europe, and despite all the opposition and extensive research by numerous authors, there is still room for improvement in terms of developing more accurate forecast models.

In this chapter, we utilize the rating systems, statistical, and machine learning methods presented in previous chapters in order to forecast the soccer outcome. The prediction models generated are only based on basic input team statistics (wins, goals, and shots) in the EPL. The EPL was selected due to the fact that is one of the most

watched and competitive competitions worldwide (Elliott, 2017). This is also one of the main reasons many studies have been focused on this league.

The research methodology consists of the following steps:

(1) Problem definition
(2) Related work and relation to our previous work
(3) Dataset preparation
(4) Methods of Approach and Prediction Techniques
(5) Experimental Design and Procedure
(6) Experimental Results
(7) Evaluation

The experimental part is designed to cover two different approaches. The first regards the accuracy of the models while the second evaluates their profitability in betting. For those two approaches, the empirical study aims to measure and compare the performance of the prediction models generated by three different categories. Those categories which we also call prediction techniques are the predictions based on: (1) team rankings (Rank-based), (2) statistical methods, and (3) machine learning algorithms. The third category refers to a hybrid technique where a rating system is combined with machine learning algorithms to make predictions. It is important to mention that our focus is not primarily directed toward proposing a high-performing model in terms of accuracy or profitability. Instead, we also aim to show which category of prediction models performs better in each approach. However, for evaluation purposes, several prediction baseline models are taken into consideration and compared with our top-performing models.

The structure of this chapter is as follows:

- Section 6.2 provides a brief analysis of the problem definition.
- Section 6.3 discusses the related work on soccer outcome prediction.
- Section 6.4 includes the relation of this application to our previous work.
- Section 6.5 explains the data preparation process.
- Section 6.6 introduces the methods of approach and prediction techniques.
- Section 6.7 describes in detail the experimental design and procedure.
- Section 6.8 presents the experimental results.
- Section 6.9 deals with the evaluation of the experimental results.
- Section 6.10 draws the conclusions of the chapter.

6.2 Problem Definition

Soccer is quite an interesting sport. As one of the most popular sports in the world, soccer match outcomes can be difficult to predict. Additionally, not all matches are made equal. Some matches are more important than others. Predicting a match between the best and the worst team in the league is hardly a challenge, but predicting a match between two teams close in strength is a highly valuable tool in the arsenal of someone interested in making a profit.

Suppose that the outcome signifies the strength of one team over another and the two teams are close to each other. In this case, in a low-scoring sport such as soccer, a Draw is a likely outcome. There are also a limited number of times when the Draw is not an acceptable outcome, and a series of penalties are shot. This happens mainly during final games, in order to declare a winner for the cup, or in knockout games (such as the World Cup knockout stage).

The EPL is a highly competitive and valuable league worldwide (Elliott, 2017). The analysis of EPL games for the seasons 2005/2006-2017/2018 that follows aims to provide a clearer understanding of the nature of the prediction problem. The games are analyzed from three different perspectives: (1) outcomes, (2) goals, and (3) betting odds.

❖ Outcomes Analysis

Besides the fact that fans can get passionate about their favorite team, many times the outcome is a draw. Actually, in EPL, matches end up in a Draw almost a quarter of the time. Figure 6-1 (A) shows the percentage of times the matches end in a Home team win (46.6%), an Away team win (25.1%), and a Draw (28.3%). The fact that a quarter of the matches end up in a draw, can partly be attributed to the fact that the goals scored in soccer are significantly less than the goals scored in other sports. The Draw percentage is a lot higher than in most other team sports. For example, in the NFL (National Football League), for the season 2020, only one draw (tie) occurred. Also, the NBA (National Basketball League) avoids ties, by giving extra time in order to declare a winner.

The outcomes can be transformed into Favorite, Outsider, and Draw based on the average odds given by the three bookmaker companies mentioned next in Odds Analysis. From three possible outcomes, the Favorite represents the lowest odd, the Outsider is the highest odd, and the Draw is in the middle. Figure 6-1 (B) demonstrates the percentages of times the matches end in Favorite, Draw, and Outsider outcomes. It can be observed

that the Favorite outcome has a win rate of 54.6% among total games. This implies that a simple prediction model that always predicts the Favorite class can achieve an accuracy equal to 54.6% which is better than predicting only the Home-win which is 46.6%.

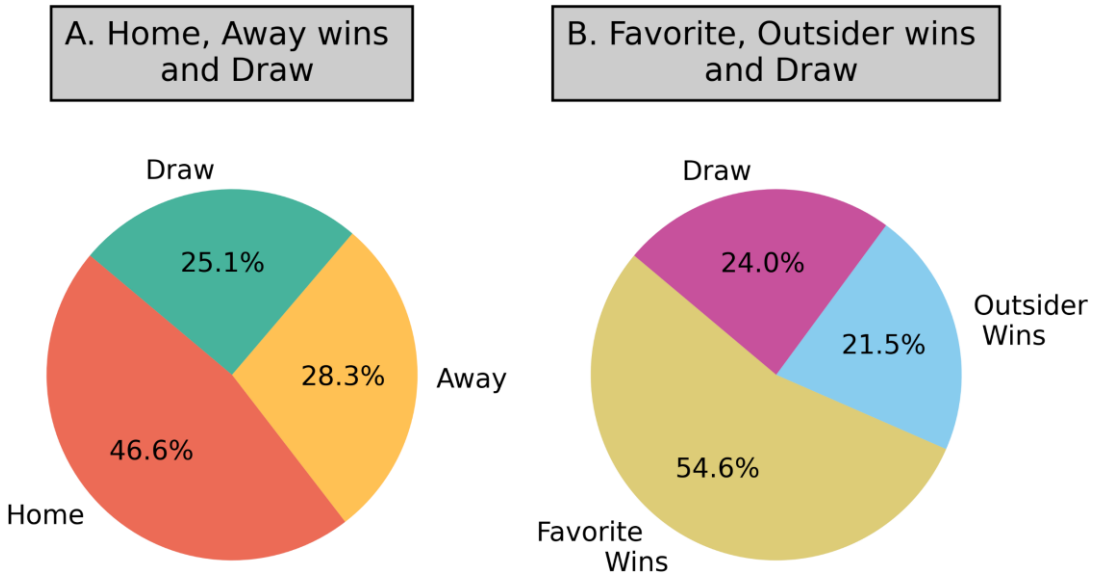


Figure 6-1: Home-Away-Draw and Favorite-Outsider-Draw results

❖ Goals Analysis

The frequencies of the total goals scored by both teams per game in the EPL for the seasons 2005/2006 to 2017/2018 are shown in Figure 6-2.

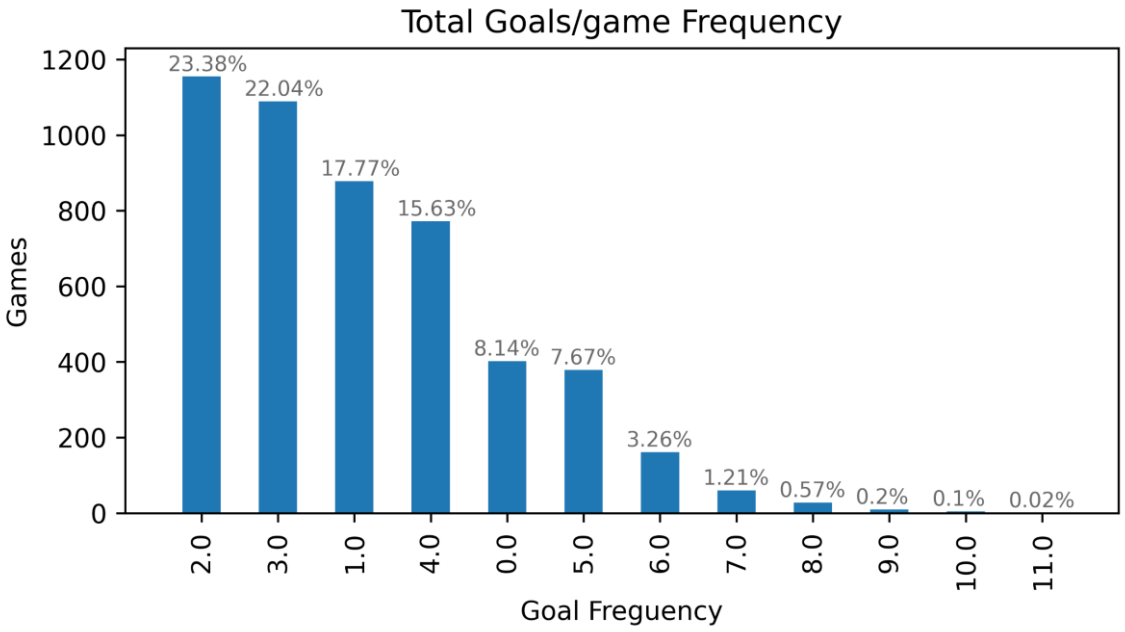


Figure 6-2: Total goals scored per game frequencies

It is evident that the most frequent result is the 2 goals (23.38%), and then follows the 1 goal and 4 goals, with 17.77% and 15.63% respectively. This suggests that the occurrence of a tie is common as an outcome, due to low scores.

❖ Odds Analysis

In soccer, as it is known there are three possible outcomes where the bookmakers offer their betting odds. An interesting point is to study those betting odds. Thus, as a pre-analysis step, descriptive statistics were used for the final outcome odds, and the results are depicted in Table 6-1. Note that for each outcome the maximum odds that are offered by three well-known online bookmaker companies: Bet365, Bet & Win and Interwetten are taken into account. Also, each row of Table 6-1 pertains to all offered odds except the last row which represents the average winning odds per outcome, i.e., the average from the subset of betting odds associated with successful outcomes.

Table 6-1: Descriptive statistics of betting odds

Descriptive Statistics	Home-win Odds	Away-win Odds	Draw Odds
total matches	4940	4940	4940
mean	2.77	4.99	3.97
standard deviation	1.85	4.09	1.08
minimum	1.1	1.2	3
5% percentile	1.25	1.55	3.2
25% percentile	1.7	2.5	3.3
50% percentile (median)	2.2	3.6	3.5
75% percentile	2.9	5.5	4.11
95% percentile	6.5	15	6.5
maximum	17	34	13
winning mean odds	2.16	3.14	3.75

Table 6-1 demonstrates that for the sports seasons 2005-2018, the average odds of Away-win outcome (4.99) is considerably higher than the other outcomes. Then, it follows the Draw outcome (3.97) and the Home-win outcome (2.77). From their standard deviations' values, it can be concluded that the odds of the Draw outcome are more stable, whereas the Away-win odds are the most volatile. The general conclusion that can be drawn by considering minimum, maximum, and percentile values is that the Away-

win class gathers more "outsiders" compared to the other outcomes. Finally, the medians for Away-win and Draw are very close, while the 95% percentiles are the same for Home-win and Draw. Additionally, examining their histograms, and overlay histograms presented in Figure 6-3 and Figure 6-4 respectively, can provide a clearer explanation.

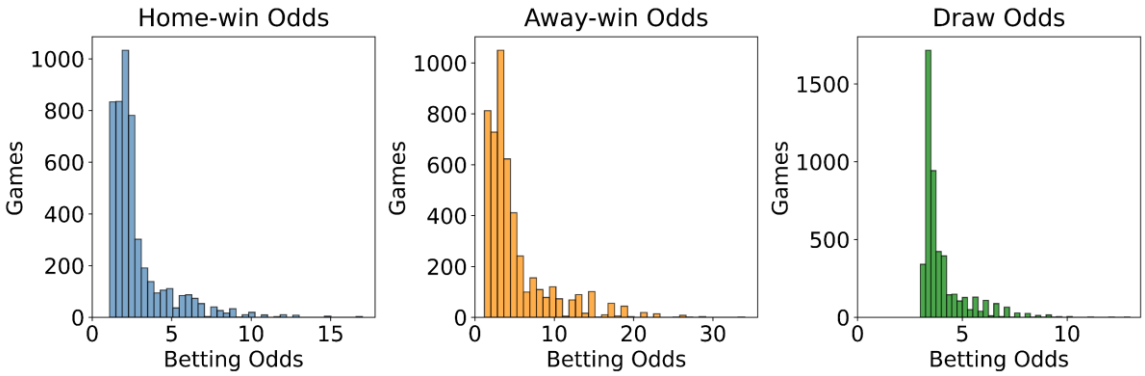


Figure 6-3: Histograms of betting odds per outcome

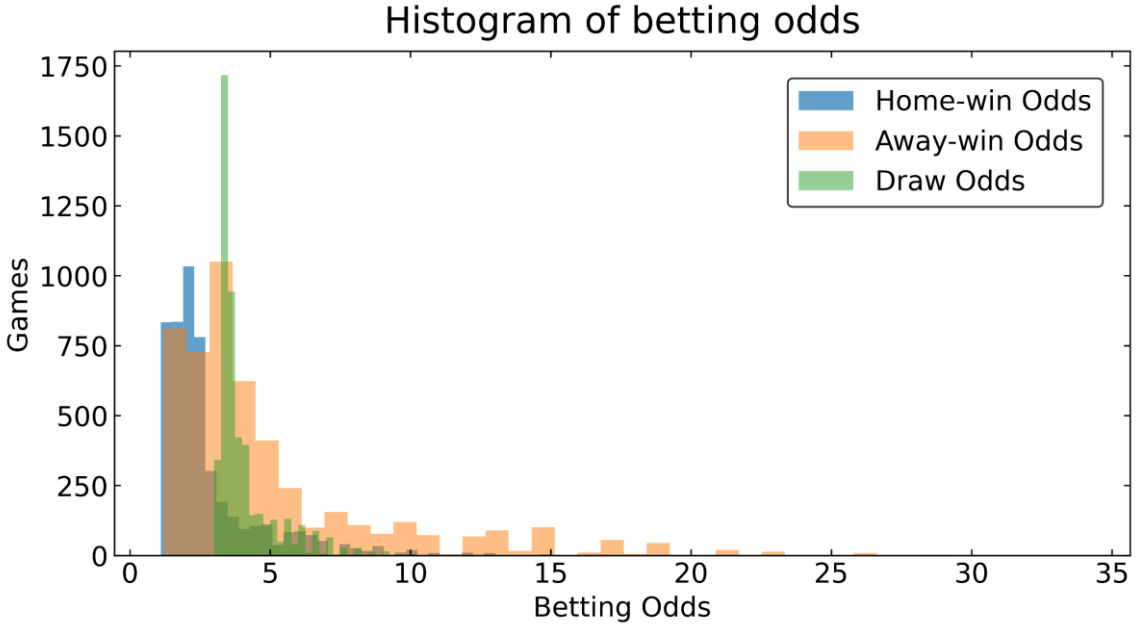


Figure 6-4: Overlay histogram of betting odds (Home-win, Away-win, and Draw)

From the comparison of individual histograms per outcome, as well as the overlay histograms, it is clear that Away-win odds have a greater proportion of values towards the right of the x-axis. This suggests that Away-win odds distribution has more outsiders. In contrast, the Home-win odds gathered more favorites and lower odds. The Draw odds are very stable and are placed around the 4 value.

After analyzing the odds per outcome offered by bookmakers, it is interesting to focus on the winning outcomes' odds that are shown to their averages in the last row of

Table 6-1. Two observations can be derived from the winning odds average values. The first is that for all outcomes their average winning odds are lower than their mean odds. The second is that although the Draw average odd (3.97) is lower than the Away-win average odd (4.99), for winning odds the opposite holds, i.e., the Draw (3.75) is higher than the Away-win (3.14).

The expected betting investment profits per outcome can be computed by examining the winning odds. By taking into account the profit formula (5.6), the expected profit of each outcome is

$$E(Profit_c) = (\overline{wo}_c - 1) \cdot p(c) \cdot bs \ - \big(1 - p(c)\big) \cdot bs,$$

where *c={home, away, draw}*, $\overline{wo}_c$ is the average winning odds for outcome *c*, *p(c)* is the probability of *c* to win (in this case is the winning percentage of *c*), and *bs* is the betting size.

Thus, the expected profits of each outcome, when the *bs* is equal to 1 monetary unit are

$$\mathrm{E}(Profit_{home}) = (2.16 - 1) \cdot 0.466 \ - 0.534 = 0.0065,$$
$$\mathrm{E}\big(Profit_{away}\big) = (3.14 - 1) \cdot 0.283 \ - 0.717 = -0.1114,$$
$$\mathrm{E}(Profit_{draw}) = (3.75 - 1) \cdot 0.251 \ - 0.749 = -0.0588.$$

From the expected profits, it is evident that $\mathrm{E}(Profit_{home})$ is slightly positive whereas the other two are negative. The $\mathrm{E}\big(Profit_{away}\big)$ is particularly poor, which can be attributed to the high number of outsiders.

Overall, by analyzing the outcomes, goals, and betting odds the general finding is that the prediction of the EPL game outcomes is a very challenging task and this difficulty is magnified when the predictions are oriented to be utilized as betting decisions.

6.3 Soccer Outcome Prediction Related Work

After a thorough review of the literature, we found several scientific publications that focus on how someone can predict soccer results in terms of investment returns. These studies can be categorized based on the kind of data they use such as structured data (e.g., game/player data, odds data, etc.) or unstructured data (e.g., tweets, etc.), the type of outcome they are about to predict (e.g., number of goals scored or the direct outcome "win-draw-loss") or the techniques they use for prediction (i.e., statistical models, machine learning, etc.).

As for the employed techniques and methods, there are various ways in which the quality of a forecast model can be assessed. We identify two types of approaches (1) accuracy-oriented: studies that seek to enhance the prediction ability of the suggested models (i.e., making the lowest number of incorrect predictions); and (2) profit-oriented: studies that emphasize generating profitable outcomes by utilizing predictions in the betting market. Also, Constantinou et al. in their study included a reference to this categorization (Constantinou, Fenton, & Neil, 2012).

6.3.1 Accuracy-Oriented

A number of authors have attempted to model efficient processes to determine the outcome of soccer matches. The earlier focus was on developing statistical and probabilistic models.

Maher (Maher, 1982) proposed a statistical model for soccer match modeling based on the assumption that the number of goals scored by a team follows the Poisson distribution. Also, an improvement is applied to the basic model using the bivariate Poisson model in order to consider the dependence between scores.

Karlis and Ntzoufras (Karlis & Ntzoufras, 2003) model sports data by employing the bivariate Poisson distribution, which enables correlation between the scores of the opposing teams. To further enhance the modeling characteristics, diagonal inflated models are also proposed by them.

Goddard (Goddard, 2005) compares the goal-based and result-based approaches by evaluating their performance in predictions for soccer game outcomes. The author employed the bivariate Poisson regression and the ordered probit regression for the goal-based and result-based estimations respectively. Four models are analyzed, covering all possible combinations of goal-based and result-based dependent variables as well as lagged performance covariates. The author suggests that a hybrid specification that combines a dependent variable based on results with lagged performance covariates based on goals has the highest forecasting performance.

Karlis and Ntzoufras (Karlis & Ntzoufras, 2008) propose a novel method for modeling soccer data based on the margin of victory. The suggested model has a simple Poisson latent variable interpretation without relying on assumptions about the distributions of the actual goals scored by each team. Although the authors applied their

suggested model in the EPL, the model has the potential to be utilized with data from a variety of soccer leagues, and compared to the bivariate Poisson model, its parameter estimation is simpler and its parameters are easier to interpret.

Newer contributors adopted more computational methodologies. Also, machine learning techniques such as Naive Bayes, Bayesian Networks, Support Vector Machines, Neural Networks, and combinations of various machine learning algorithms are utilized by several researchers. For example, Buursma (Buursma, 2011) applied machine-learning techniques utilizing several features for the Dutch soccer competition. Their models exhibit accuracy rates that do not exceed 55%. Odachowski and Grekow (Odachowski & Grekow, 2013) took their research a step further using predictive machine learning algorithms over a novel binary classification approach that focuses on the prediction of each possible outcome individually.

Lasek et al. (Lasek, Szlávik, & Bhulai, 2013) evaluate the predictive ability of several rating systems in soccer. Their findings highlight that the top-performing rating system is the Elo while the other rating systems tested also outperform the official rating system of the Fédération Internationale de Football Association (FIFA) that was used as a benchmark.

Haaren and Broeck (Haaren & den Broeck, 2014) applied a relational learning approach for the prediction of goal difference in EPL games. Haaren and Davis (Haaren & Davis, 2015) utilized rating systems to predict several final league tables of several European Soccer Leagues.

Herbinet (Herbinet, 2018) examined the prediction of soccer match scores and outcomes in 5 European Leagues by utilizing machine learning techniques. The study explores the combination of expected goals and Elo ratings as features in classification and regression tasks. In terms of the outcome prediction, the accuracy rate achieved was 51.1% which was lower than the bookmaker's odds model.

Baboota and Kaur (Baboota & Kaur, 2019) focused on the outcome prediction of the EPL and compared the results to the bookmaker's odds model. They utilize feature extraction and exploratory data analysis to identify the feature set that contains the most significant criteria for predicting match outcomes. The training data spanned 11 seasons while testing data was 2 seasons. In terms of accuracy, the Gradient Boosting approach (56.7%) performed the best, followed by the Random Forest (56.4%), SVM models

(RBF 54.5%, linear 54.2%), and the Gaussian Naive Bayes method (52.6%) performed the worst. In terms of RPS, Gradient Boosting was their top-performing model that achieved 0.2156 while the bookmaker's odds model outperformed with 0.2012.

6.3.2 Profit-Oriented

This subsection provides several research works that explore the profitability part (either primarily or secondarily) when predictions are used for betting purposes. The availability of multiple forecasting methods raised questions about their effective use and the potential for systematic profitability for investors in sports markets. Among the works that tried to come up with this challenge, there are several interesting approaches. Some of these cases showed abnormal positive returns from betting strategies.

Dixon and Coles (Dixon & Coles, 1997) proposed a parametric statistical model that uses a bivariate Poisson distribution to estimate the number of goals scored by each team in a match. The estimation of goals allows the calculation of goal probabilities which can be converted into outcome probabilities and into a prediction of the final outcome. Their approach includes the computation of the offensive (attack) and defensive ratings for teams. The authors used historical data from the English League and Cup matches to fit their model which gives more weight to recent matches and takes into account the home-field advantage. Following that, they applied a betting strategy in validation sample data to examine the performance of their model, resulting in positive returns.

Rue and Salvesen (Rue & Salvesen, 2001) proposed a statistical model for predicting soccer match outcomes employing a Bayesian dynamic generalized linear model. Their model is time-dependent and every match week estimates the offensive and defensive ratings of teams (that reflect their updated relative strengths). Based on the past results of the EPL and Division 1 they predicted match outcomes and examined the performance of their model in terms of betting, yielding promising results.

Dixon and Pope (Dixon & Pope, 2004) explored the UK football (soccer) association betting market's efficiency and they suggest that the market is inefficient. The authors were based on the historical outcomes and betting odds from the EPL over several seasons. They provide an in-depth analysis in terms of the statistical forecast efficiency and demonstrate the possibility of generating positive returns in the long run.

Their study suggests that bookmakers' odds are not fully efficient and there are opportunities where the bettors can generate profits by utilizing statistical models.

Goddard and Asimakopoulos (Goddard & Asimakopoulos, 2004) utilized an ordered probit regression model for the forecasting of the English League soccer match outcomes. The authors used their forecasting model to investigate the weak-form efficiency of betting markets, and their analysis suggests that the bookmakers' odds are weak-form inefficient. Also, employing a betting strategy based on their model would have resulted in a positive gross return of 8% in the end-season games.

Forrest et al. (Forrest, Goddard, & Simmons, 2005) focused their interest on English soccer games and analyzed the performance of predictions based on publicly available odds in comparison to predictions generated by a benchmark statistical model. Their findings suggest that the published odds predictive strength improved over the studied period. In terms of profitability, subjective predictions by experts that are based on odds can outperform statistical model predictions when the financial stakes are sufficiently high.

Spann and Skiera (Spann & Skiera, 2009) forecasted the results of the German Premier Soccer League, examining the predictive accuracy and profitability capacity of three different methods: prediction markets, betting odds, and tipsters. Their findings suggest that the methods of prediction markets and betting odds exhibit similar levels of forecasting accuracy, outperforming tipsters significantly. In spite of this, none of the predictions result in consistent financial gains in the betting market due to high fees.

Hvattum and Arntzen (Hvattum & Arntzen, 2010) utilized the Elo rating differences as covariates in ordered logit regression models to predict the soccer results of the top four divisions of the English League. The results from their study highlighted the advantage of utilizing Elo ratings over the goal-based approach by Goddard (Goddard, 2005) when applied to relatively smaller datasets.

Constantinou et al. (Constantinou, Fenton, & Neil, 2012) proposed the pi-football model to forecast the outcomes of soccer games in the EPL. Their proposed model is a Bayesian network model that takes into account various objective and subjective information. Also, the time-dependent data are weighted through the use of different levels of uncertainty. The authors applied their model in the EPL and according to their findings, the profit (%) gained from betting ranges from 2.87% to 9.48%.

Godin et al. (Godin, Zuallaert, Vandersmissen, De Neve, & Van de Walle, 2014) investigated four different techniques to predict the outcome of a soccer match for the EPL. The first type of technique used statistical information, while the other three used Twitter microposts. By extracting and aggregating data from over 50 million Twitter microposts, the collective knowledge-based prediction techniques utilized were based on Twitter volume, sentiment analysis, and user prediction analysis. The study results demonstrated that these techniques outperformed forecasts made by experts and bookmakers. Their approach yields a profit of 30%.

Koopman and Lit (Koopman & Lit, 2015) introduced a statistical model for soccer match outcomes forecasting. The model is based on the assumption that match outcomes follow a bivariate Poisson distribution where the intensity coefficients change in a stochastic way. Their methodology is applied in the EPL and demonstrates noteworthy effectiveness for generating favorable returns in a betting strategy, outperforming bookmakers' odds.

Boshnakov et al. (Boshnakov, Kharrat, & McHale, 2017) introduce a forecasting model designed to predict the distribution of scores in soccer matches. The model is based on the assumption that the inter-arrival time for goals follows a Weibull distribution. The application of their approach in betting for the EPL data yields positive results, with the highest achieved ROI reaching 21.2%.

Constantinou (Constantinou, 2019) designed a model to predict match outcomes of various soccer leagues across the globe using a mixture of dynamic ratings and Hybrid Bayesian Networks. The ROI metric was considered as a profit evaluator. In the case of the EPL, the model performed remarkably managing to reach ROI with values ranging from 6.4% to 38% per season.

6.4 Contribution and Relation to Our Previous Work

In this section, we discuss our contribution based on our prior work in the field of sports outcome prediction and its relation with the present application. The application presented in this chapter is developed by the combination of some ideas and approaches from our publications discussed below. The core of this application is based on our prior works (Talattinis, Kyriakides, Kapantai, & Stephanides, 2019; Talattinis & Stephanides,

2022). In fact, the current chapter considers the performance of both machine learning and rating systems on the prediction of the final outcomes of upcoming soccer games.

In the paper (Kyriakides, Talattinis, & Stephanides, 2014) we have worked on the EPL data to predict final outcomes and identify profitable methods. The performance of Offense-Defense, Colley, and Massey systems to machine learning classification algorithms (Neural Networks, Decision Trees, Random Forests) are compared. Concluding that accurate models are not always profitable, we tried to improve them by following another approach in the publication (Kyriakides, Talattinis, & Stephanides, 2015) where risk management is applied as a filter. Moreover, in our publication (Kyriakides, Talattinis, & Stephanides, 2017) we suggested a hybrid method, combining Colley, Offense-Defense, and novel AccuRATE rating (discussed in section 4.2) systems with machine learning methods (Artificial Neural Networks, Decision Tables, Naive Bayes, Logistic Model Trees, Bagging, Stacking). Our results indicate that their combination performed better and allowed greater flexibility in terms of the desired goal (accuracy, profit).

Also, in the paper (Efstathiou, Diamanti, Talattinis, & Stephanides, 2015), the adaptability of the LRMC rating system (Kvam & Sokol, 2006) on other sporting events than NCAA was tested. The method was compared with variations, and is also, tested for profitability after the utilization of predictions in betting. Firstly, the method was examined for its applicability in other leagues than the NCAA but in the same sport that was proposed. Particularly, the method was applied to predict team rankings in the Spanish professional First Division of basketball and the findings have shown that it can be applied successfully. Then, we tried to apply the method in other sports such as soccer and handball in the competitions of the Spanish professional First Division (Primera División) and Handball-Bundesliga respectively. The results suggest that the LRMC method generates positive results and can effectively predict the rankings in other sports after the analogous modifications. In addition, we attempted to use other methods than logistic regression in order to compare the results produced by them and explore the possibility of using them as an improved model. Our findings indicate that in some cases other methods perform better than logistic regression. Finally, when the method is applied to real case scenarios in order to test its profitability in betting, the findings are useful for future research.

Furthermore, in our research paper (Talattinis, Kyriakides, Kapantai, & Stephanides, 2019) we employed a genetic algorithm to generate better features for machine learning algorithms. The generated models were used to virtually bet on soccer outcomes and the Sharpe ratios of their equity curves were calculated. We used this approach in order to perform binary comparisons of various methods of generating betting strategies from predictions. We compared a fixed stake size with a variable stake size approach (Kelly criterion), a ternary versus binary classification scheme, a cost-insensitive versus cost-sensitive approach as well as a comparison of various learning algorithms. Cost-sensitive approaches seem to work better in this domain since the betting odds are taken into account and play a significant role in the building of profit-oriented models. Also, we have concluded that the Kelly criterion as money management and binary classification, focusing on the win or loss of the home team, combined with a cost-sensitive classification, seemed to be the best approach. Further experiments suggested that the use of Naive Bayes or Random Forest as a base classifier performed best in the betting prediction task, evaluated by the Sharpe ratio of the produced equity curves.

Finally, our software paper (Talattinis & Stephanides, 2022) includes an illustrative example that demonstrates the overall procedure of outcome prediction in the EPL. In this example, the predictive performance of various rating systems is assessed with three different prediction techniques when the target class is the final outcome of the EPL (2009–2018 seasons) matches. The prediction techniques are the Rank-based, MLE, and machine learning classification where the Naive Bayes is examined.

6.5 Dataset Preparation

This section deals with the data preparation process. The highlights of the process involve the data used, the selection of data attributes, the conversion of rating values to machine learning features, and the data preprocessing.

6.5.1 Data used

To ensure accurate results, we selected only those features that were either publicly available or easily calculated. The set of input data was collected from the online available database at http://www.football-data.co.uk (Football-data, 2023). We used 13 years of soccer games spanning the years 2005 - 2018 in the EPL. The historical data

used consists of the wins, goals, and shots (which are also used in our illustrative example 3.2). Although there are several useful features to predict the game outcome, our focus is only on the basic soccer teams' statistics and attributes, which are often featured in other research studies. Also, the importance of *TST* and *TS* statistics is highlighted in the introduction of the AccuRATE (4.2.1). The betting odds are taken as an average and as a maximum from three well-known online bookmaker companies (Bet365, Bet & Win, and Interwetten).

6.5.2 Data Attributes

Each data instance represents an independent game that is described by a set of different attributes. Those attributes are also described in section 3.2. However, for the PointRATE system, the data attributes are pre-calculated and transformed as averages to the number of total games played by each team. Then during rating computation, the most and least preferred value is used. The computation details of attributes can be found in subsection 4.3.6. Additionally, the maximum betting odds are used in the betting procedure while the average odds are used in cost-sensitive learning, and in some other prediction models (Favorite, Outsider). Thus, for each soccer game, the calculated attributes are:

- *TW*: team's average total wins per game prior to the itinerary match;
- *TG*: team's average total goals scored prior to the itinerary match;
- *TST*: team's average total number of shots-on-target prior to the itinerary match;
- *TS*: team's average total number of shots prior to the itinerary match;
- *AVGODD*: the average odd for each outcome;
- *MAXODD*: the maximum odd for each outcome;
- *FO*: the final outcome of the match (1 for Home-win, 2 for Away-win, and 3 for Draw).

An example of a data instance for some of the above attributes can be found in Table 6-2. The initial letter 'H' indicates the Home team or Home-win result, 'A' the Away team or Away-win result, and 'D' the Draw result.

Table 6-2: Example of data instance

HTW	ATW	HTG	ATG	…	HMAXAODD	AMAXODD	DMAXODDS	FO
0.75	0.25	3	0.5	…	1.36	10	5.3	1

Note that all teams' statistics are calculated by considering all game data prior to the itinerary match, i.e., the target match. The purpose of this step was to exclude any information about the itinerary match outcome in the training and testing phase since such information, i.e., goals scored, shots scored, etc. of the target match will also not be available in practical settings. For example, if the match in Table 6-2 regards the 5th match week this indicates all the attributes are calculated from the 1st to 4th match weeks. For this example, the only column in Table 6-2 that applies to the 5th match week is the FO. For better understanding, the HTW, HTG, HMAXODDS, and FO values of the example instance are explained in the following way:

- HTW with a value of 0.75 indicates that the home team has 75% of total wins without including the 5th match week, i.e., 3 of 4 games are won by the home team of that match from the 1st until the 4th match week.
- HTG with a value of 0.5 indicates that the home team (without including the 5th match week) scored 50% of average goals compared to the best team in the league for those 4 match weeks.
- HMAXODDS: from the odds given by the three bookmakers mentioned in subsection 6.5.1, the highest odd for the Home-win outcome is 1.36
- FO: the outcome of the 5th match week for this dataset instance game is 1, which means that the home team has won.

In a similar way, the other values can be explained.

6.5.3 Rating Values as ML Features

The ratings of teams are computed on a weekly basis based on the data from the previous match weeks and then the weekly values are normalized on a scale of 0 to 1 by applying the formula

$$r_{norm} = \frac{r - min\,(r)}{max(r) - min\,(r)},$$

where *r* is the rating vector, r_{norm} is the normalized rating vector, *min(r)*, and *max(r)* are the minimum value and maximum value of the vector *r* respectively.

The approach of rating values generation is called anchored walk-forward and we will discuss it in detail in subsection 6.7.2. The parameters of rating systems are optimized to previous sports seasons data, and the optimization procedure is explained in subsection 6.7.6. The process of rating is performed separately for each rating system

(RS) and restarts at the beginning of every season. For better understanding, the process is depicted schematically in Figure 6-5.

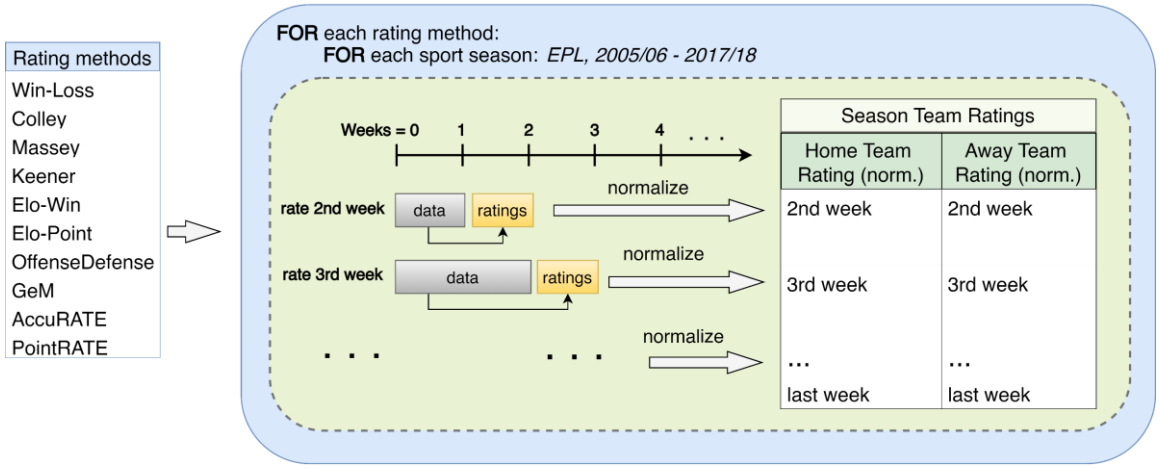


Figure 6-5: Rating values as ML features

As we can observe from the figure above, there are two features: the Home Team Rating and the Away Team Rating, which are calculated based on teams' statistics prior to the itinerary match. Finally, the normalized rating values are utilized as ML features for the classifiers.

In this application, the ML classifiers utilized two feature sets. In the first, only the normalized rating values of teams are included while in the second, the average betting odds are also used as features. For the first and second feature sets, we assign the acronyms ML[rs] (i.e., rs: rating system scores), and ML[rs+odds] (i.e., rs+odds: rating system scores and odds) respectively. Also, examples of data instances with the target class are shown in Table 6-3 and Table 6-4 for the ML[rs] and ML[rs+odds] respectively.

Table 6-3: Rating scores as ML features and target class - ML[rs]

Features		**Class**
Home Team Rating (norm)	**Away Team Rating (norm)**	**FO**
0.75	0.25	1

Table 6-4: Rating scores as ML features, avg odds, and target class - ML[rs+odds]

Features					**Class**
Home Team Rating (norm)	**Away Team Rating (norm)**	**HAVGODD**	**AVGODD**	**DAVGODD**	**FO**
0.75	0.25	1.36	9.33	5.18	1

6.5.4 Preprocessing

After converting rating values to machine learning features, we apply a preprocessing procedure. The preprocessing procedure aims to filter out games between non-rated teams that appear in the first week of every sports season. Note that some rating systems require more match weeks to produce ratings. For example, the Massey method requires enough games to make the games graph connected and sometimes there are non-rated teams for a period longer than one week.

The importance of this step can lead to enhancing the performance of machine learning algorithms since the training dataset will be improved by removing non-rated teams.

6.6 Methods of Approach and Prediction Techniques

- Methods of Approach

  Although in most surveys on machine learning the determination of the best model is based on the proportion of accurate predictions as discussed in related work, this method is not always cost-effective when the objective is to invest in the betting market. Additionally, the risk should be taken into consideration. Therefore, our approach divides our experimental part into two major categories: (1) accuracy-oriented models, and (2) profit-oriented models, which are explained below:

  - Accuracy-Oriented

    The accuracy-oriented approach focuses on making predictions for future outcomes of games with the least possible error. The overall objective was to reveal the best-performing model in terms of prediction ability. Various evaluation metrics are used such as Accuracy, Weighted Average F1-score (i.e., takes into account the number of samples in each class), and Average RPS for the performance measurement.

  - Profit-Oriented

    As already mentioned in section 6.3 the profit-oriented approach focuses on the exploitation of predictions of game outcomes as betting decisions. However, our approach is different than most studies mentioned in 6.3.2. In the present application, betting activity is viewed as a form of investment, where the betting portfolio is evaluated. Similar to a financial portfolio, the betting portfolio cannot

be considered successful solely based on its profitability; it might also be adjusted for risk. Thus, the aim is the construction of a risk-adjusted betting portfolio based on the predictions made by the prediction model. Although rating systems can be a valuable tool in any sports bettor's arsenal, they can never fully model the complexity of other factors involved in betting. For this reason, cost-sensitive learning and money management techniques are involved in order to optimize models in this orientation. The Sharpe ratio (Sharpe, 1994) is used as a performance evaluation metric, similar, to financial portfolios. Our overall objective in this approach is oriented to identify the right direction that has better prospects for discovering better risk-adjusted models.

❖ Prediction Techniques

To verify the validity of the methods and techniques described in previous chapters, we carried out several experiments to develop our best-performing models. The prediction techniques are split into three categories explained below. The initial objective is to reveal the best-performing category/subcategory/model in terms of prediction ability or profit regularity, continuity, and risk-adjusted performance.

- Rank-based: The logic of prediction is based on team rankings. The "RANK" approach is explained in section 3.5 and also applied in the illustrative example of section 3.2.
- Statistical-based: The logic of prediction is based on statistical methods. In this category, the "MLE" method is used. It is explained in section 3.5 and also applied in the illustrative example of section 3.2.
- Machine Learning Classification (ML-based): The prediction can be made by utilizing the ratings of teams (Home and Away teams) of one rating system (RS) every time as machine learning features for the classifiers.

Additionally, the following other types of prediction techniques that do not belong to the above categories and are taken into consideration for comparison purposes:

- Odds-Oriented: The prediction is based exclusively on bookmaker odds. Favorite and outsider prediction strategies belong to this category.
- Random: Two types are considered: (1) Uniform predictions (i.e., each outcome has an equal probability of being selected), and (2) Stratified predictions (i.e., predictions are made in proportion to the class distribution).

- Constant: The prediction is constant for all games. There are three possible prediction strategies: Home-Only, Away-Only, and Draw-Only.

Further explanations of Odds-oriented, Random, Constant, and other types can be found in section 6.9.

When dealing with Rank-based or MLE, 10 different prediction models for each technique are generated with each rating system contributing to one. In ML classification, a separate model is generated for each rating system resulting in 10 different prediction models for each classifier, which are a total of 70 models since 7 classifiers are examined. Note that for each prediction technique, separate models are generated for accuracy and profit-oriented approaches. Especially in profit-oriented models, in order to measure the efficiency of different money management methods or cost-sensitive learning, the models are generated separately as well.

6.7 Experimental Design and Procedure

This section deals with the experimental design and procedure which concern the accuracy and profit-oriented approaches.

6.7.1 Backtesting and Simulation

The terms “backtester” and “backtesting” originate from the field of Finance and they refer to the testing of trading strategies by simulating historical data. Similarly in this application, a sports backtester was developed in order to address the simulation of historical soccer matches between the teams and simultaneously to test the performance of accuracy and profit-oriented models. The backtesting/simulation runs for a range of sports seasons. Particularly, it starts from the 4^{th} match week of each season to simulate the soccer matches and apply the predictions made by the models. Every match week starts on Thursday and ends on Wednesday. However, even though there are 20 teams in EPL, due to postponed or rescheduled matches, in some cases it is possible to count either more or less than 10 matches in a match week.

6.7.2 Walk-Forward Analysis

The Walk-Forward Analysis (WFA) is a common method that can be applied to time-series data with the goal of splitting them into several training and testing sets by

preserving their order. Since the soccer matches and team ratings follow a chronological order, walk-forward is used to preserve the order of data during the prediction process.

WFA can be a valuable technique for identifying weaknesses in a prediction model and can also help to avoid overfitting to historical data. Overfitting is a common issue where a model performs well on the training data but cannot generalize on unseen data. We have also referred to overfitting in section 5.2.

Two types of walk-forward analysis are employed for our experiments' purposes. The first one is called Rolling WFA while the second is the Anchored WFA (Jaekle & Tomasini, 2019). In both types, the dataset is divided into multiple "in-sample" and "out-of-sample" periods that follow a chronological order. Then, a model iteratively is trained on an in-sample period and validated on an out-of-sample period, until all periods are covered. On one hand, the Rolling WFA utilizes a rolling window where the in-sample and out-of-sample data are shifted forward over time. In each (next) step forward, the tested out-of-sample data of the previous point is included in the in-sample data, and a part (or whole) of old in-sample data is discarded. On the other hand, in the Anchored WFA, the in-sample periods are non-overlapping and the beginning point remains stable for each subsequent out-of-sample period and does not change over time. Subsequently, the tested out-of-sample data is added to the next set of in-sample data and this procedure continues until all data is tested. Figure 6-6 and Figure 6-7 demonstrate the Rolling WFA (RWF) and Anchored WFA (AWF) respectively.

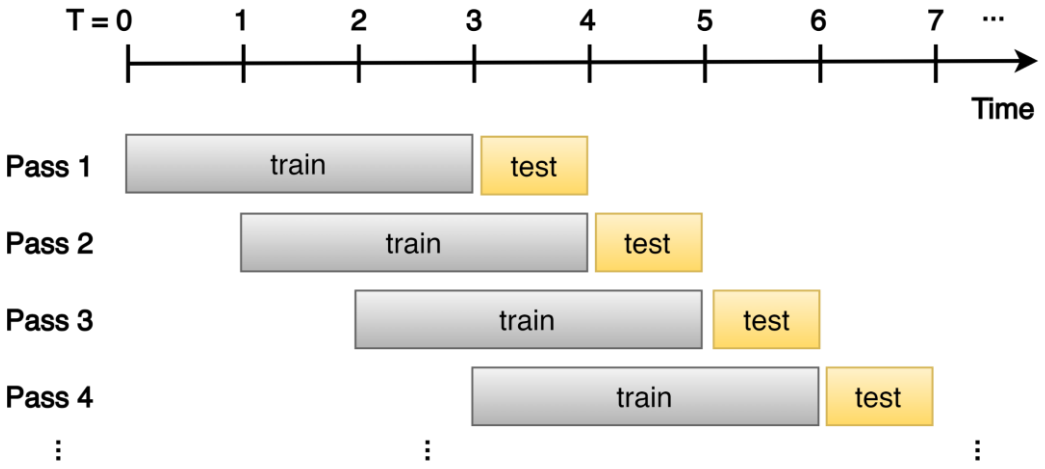


Figure 6-6: Rolling Walk-Forward Analysis (RWF)

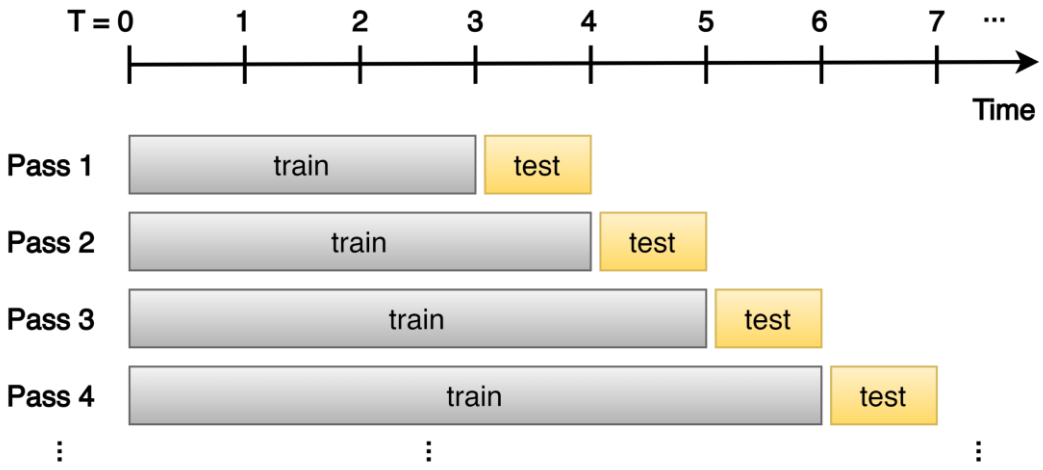


Figure 6-7: Anchored Walk-Forward Analysis (AWF)

From the diagrams, it can be observed that in the Anchored WFA there are more available training data after the first pass which can lead to a better-performed and more generalized model. On the other hand, this means that is computationally intensive because the training dataset is expanded over the periods. The Rolling WFA has the advantage that the model is trained to more recent data. Also, when the model is affected by noise from old data the utilization of Rolling WFA is preferred.

In this application, the Rolling WFA (RWF) is applied for the hyperparameter tuning of rating systems and machine learning models while the Anchored WFA (AWF) is applied during the computation of ratings and backtesting/simulation process. The ratings of teams are computed on a weekly basis using an AWF approach (i.e., based on the data from the previous match weeks of the same season), and they are reset at the beginning of every season in order to reflect teams' current strengths. The same, during backtesting/simulation, the walk-forward process restarts at the beginning of every season in each model. The purpose of adopting this methodology was to ensure fairness in comparison between Rank-based models with statistical or ML models. The predictions of each model are constructed by aggregating them across multiple seasons, and the model's overall performance is measured based on that.

6.7.3 Evaluation Metrics

Different evaluation metrics have been utilized for accuracy and profit-oriented approaches. For the accuracy-oriented approach, the Accuracy is used as a primary metric as it is very simple, and also a common metric in many related works. Also, we

have considered the Weighted Average F1-score which takes into account the number of samples in each class. For simplicity reasons, we refer to it as F1-score. Finally, the Average RPS metric (i.e., computed as an average RPS of all predictions) is utilized to evaluate the probabilistic predictions. For simplicity, we refer to it as RPS. Evaluation metrics for this accuracy-oriented approach are explained in detail in section 5.3.

To assess the profit-oriented models' performance, the Sharpe ratio (SR) (Sharpe, 1994) was used as a risk-adjusted measure. The Sharpe ratio was also utilized in our study (Talattinis, Kyriakides, Kapantai, & Stephanides, 2019). Financial investors widely tend to utilize the Sharpe ratio, in order to understand the return of an investment compared to its risk. In this thesis, we investigate the feasibility of such methods in sports betting where a betting portfolio with a higher Sharpe ratio or a class of portfolios with a higher mean and lower standard deviation, indicate a better performance. The Sharpe ratio quantifies an investment strategy's performance by combining its strategy's average returns with its standard deviation of returns. In our case, it gave the amount i.e., the average returns of the betting portfolio, that a bettor is compensated for with respect to the risk taken. The formula is given below:

$$Sharpe\ Ratio = \sqrt{K}\frac{\bar{R}_x - R_f}{S_x}, \tag{6.1}$$

where K is the average number of trades (bets) per sport season, x are the betting trades' returns, $\bar{R}_x$ is the average return, R_f is the risk-free rate, and S_x is the standard deviation of returns.

We integrate the square root of K in (6.1), in order to evaluate portfolio returns over an annualized basis. This eliminates cases where a high but statistically insignificant Sharpe ratio is generated, due to a low number of observations (trades/bets). Furthermore, $\sqrt{K}$ regulates the impact that negative Sharpe ratios have on the selection of models during the optimization process of hyperparameter tuning. As the Sharpe ratio score also acts as a rating metric, betting portfolios with negative Sharpe ratios and a high number of betting trades (and thus consistently negative returns) will have a lower score than betting portfolios with the same Sharpe ratio but lower number of trades (and thus more probable to be random).

Usually, in financial applications, a risk-free rate, i.e., theoretical rate of return on safe investment with no risk, is subtracted from the expected return. In the present

application, we will omit it as there is no risk-free investment in terms of betting and because the purpose is to make comparisons among our models. Additionally, by assuming a zero risk-free instrument, as any other value will interact with $\sqrt{K}$ and affect the relative rankings of our models, by reducing its effect on the final rating.

Other popular metrics are the ROI and the Annualized Rate of Return (ARR). ROI which is also referred to in section 2.4 is a measure of the profitability of an investment and the result can be also expressed as a percentage. ROI is calculated by dividing the total profit or loss by the initial bankroll while ARR measures the average annual growth rate of the betting portfolio by taking into account the effect of compounding and it is expressed as an annual percentage rate. ARR is computed as

$$ARR = \left(\frac{ending\ bankroll}{initial\ bankroll}\right)^{\frac{1}{n}} - 1,$$

where *n* is the number of sports seasons that the betting portfolio was held (i.e., placed bets).

In the experimental results, we also report the R-square (R^2) and slope value calculated from linear regression of the returns of each betting portfolio. The linear regression is also discussed at the beginning of subsection 3.3.3. R-square (R^2) is a statistical measure and is also known as a coefficient of determination. It has range from 0 to 1. When R-square is close to 1 this indicates that a betting portfolio is more stable with lower variance. However, at the same time, it is important to examine the slope that represents the growth rate of the betting portfolio over time, where a positive slope signifies an increasing trend line in the betting portfolio. Evaluating both the R-square and a positive slope is essential for analyzing the portfolio trend.

6.7.4 Money Management for Profit-Oriented Approach

There are several money management strategies to define the betting size. The efficient (indicative) stake (betting size) was defined in two different methods, in this application. The first method involved fixed placing bets, while in the second method, the stake was determined by the Kelly criterion (Kelly, 1956) method.

❖ Fixed Amount Strategy

In this method, all bets are placed under the same risk by setting them to have the same betting size. This signifies that for the calculation of the betting size a fixed

percentage is multiplied by the initial betting bankroll. For example, if the initial capital of the betting portfolio is 1000 monetary units and the betting percentage size is 2.5% then the betting size for each bet is 25 monetary units. It is important to highlight that during the backtest we kept the betting size constant since the latter is related to the initial betting capital. Another variation could be to calculate the betting size based on the available capital before betting.

Furthermore, it is worth mentioning that several studies utilize another version of this money management method which places a variable betting size to achieve a fixed amount of profit per bet. In other words, each bet wins the same amount.

❖ Kelly Criterion

Many studies recommend the use of the Kelly criterion, as it is assumed to hold a distinct advantage over other similar methods because of its lower level of risk. This method calculates the amount with which to make a bet based on an outcome with a probability of success that is higher than the given odds. A successful outcome of the method involves the long-term growth of the fund, i.e., capital for betting.

The Kelly criterion formula is

$$f^* = p - \frac{1-p}{b}, \tag{6.2}$$

where f^* is the bankroll proportion per bet, b is the net betting odds, (i.e., H_{odds} - 1 for Home-win, A_{odds} - 1 for Away-win, or D_{odds} - 1 for Draw), p is the probability of success, and (1-p) is the probability of failure.

Formula (6.2) suggests that the optimal betting size is determined by the difference between the winning probability and the proportional probability of loss over the net betting odds. A negative value of f^* suggests to not place a bet. Many times, the Kelly formula is applied to a portion (fraction) of the bankroll for every bet in order to reduce the risk of money lost. This approach is commonly referred to as the "Fractional Kelly" or "Fractional Kelly Criterion" strategy.

It is worth mentioning that Kelly's strategy relies on accurate probability estimates in order to define the optimal betting size. Therefore, when the prediction method being used does not produce probabilities or the probabilities produced are inaccurate, it can be difficult to apply the Kelly Criterion effectively. For this reason, the Kelly formula is not applied to the Rank-based ("RANK") prediction technique.

By applying the Kelly Criterion in this application, the aim is two-fold. The first is to improve betting portfolio performance and the second is to evaluate the quality and reliability of the probabilities produced by the prediction method. The latter can be done if we compare two betting portfolios originating from the same prediction technique, where the first portfolio utilizes the Fixed Amount strategy and the second utilizes the Fractional Kelly. To conduct a fair comparison between two portfolios, the Fractional Kelly strategy should place bets with a fraction that achieves an equivalent average betting size with the Fixed Amount strategy. If the Fractional Kelly portfolio performed better than the Fixed Amount this suggests that the probabilities derived from the prediction method are reliable and consistent since the first portfolio utilized probabilities when placing bets while the second did not.

❖ Betting Portfolio Definition and Betting Size

Each experiment represents a separate independent portfolio of bets that starts with 1000 monetary units bankroll (initial capital). The betting process is connected with the backtester and our purpose is to run the simulation of betting events, as close as possible to the real betting market conditions. The details are explained below:

- A successful bet was defined as a successful prediction of the outcome and was rewarded as the maximum odds from the three online bookmakers we have chosen.
- The minimum betting size is set to 2 monetary units as usually, bookmakers do not accept less.
- Before placing a bet, the backtester checks if there is available capital for betting.
- The "no bet" decision is only made from the Kelly Criterion method when the bankroll proportion of a bet is negative.
- Money management:

  In the Fractional Kelly Criterion, a portion of the bankroll represents the betting percentage. For a fair comparison between Fixed stake and Kelly's strategy, the *bs* is computed on the initial capital with 2.5% for Fixed and 6.25% or 15.625% for Kelly. Practically, we have selected a higher percentage in Kelly's strategy because if we set the same betting percentage the latter will place smaller bets than the Fixed Amount method. As a consequence, a smaller betting size could lead the portfolio to withstand the risk of loss for a longer period as there

will be available capital to place bets. As mentioned before, those percentages are calculated on the initial capital for each bet and the current capital of the portfolio (before placing the bet) is not considered.

The rate of 2.5% is selected on the basis that each match week has 10 games and consequently the initial bankroll will be sufficient for at least 40 wrongly predicted games that are equal to 4 match weeks. In addition, setting a low percentage offers a lower portfolio variance. The rate for the Kelly criterion is selected after experimentation to be equal to 6.25% (2.5 times higher) for ML models and 15.625% (6.25 times higher) for MLE models in order to achieve the same average betting size as the Fixed Amount method.

6.7.5 Cost-Sensitive Learning Parameters

The main aim of cost-sensitive learning in this application is to improve profit-oriented models. Talattinis et al. compared two cost-sensitive algorithms (Talattinis, Kyriakides, Kapantai, & Stephanides, 2019). CSC and MetaCost are described in section 5.5. However, in this application, we have only employed the CSC (Witten & Frank, 2005) as MetaCost is more computationally expensive.

As noted, the cost-sensitive meta-classifiers described earlier consider the benefit of correct prediction as well as the cost of the incorrect prediction (misclassification cost), in order to achieve as possible, the most ideal prediction. Regarding the profit-oriented models in soccer, the ideal predictions depend on the cost matrix selection that should reflect model performance based on the Sharpe ratio of a hypothetical portfolio. However, some difficulties are posed when trying to define fixed cost weight values for the cost matrix scheme presented in subsection 5.5.3. As we can observe from Table 5-4 and Table 5-5, the benefits depend on the betting size and profit which are usually computed after the training of the classifier. To explain that, suppose that the Kelly criterion is applied to define betting size, thus the latter depends on the probability calculated from the prediction model. Additionally, for each matchup, the upcoming outcomes have different betting odds and many times the money management may consider the final decision as “no bet”. Example-depend cost-sensitive learning approach could be more advantageous because incorporates example-specific costs into the learning process.

Finally, the cost matrix used is calculated according to the values of the training set and the cost function which is based on the scheme presented in subsection 5.5.3. Since the training set differs every week due to AWF, the cost matrix is also different every week. The betting odds for each class are calculated based on the average winning odds of the training set and we set the betting stake size equal to 1 which means that all bets have the same size in money units. Fixed Amount money management follows this principle since the placed bets have a fixed and the same amount of money units. Also, this rule can be held for the Kelly criterion if we consider that the average size of bets from Fractional Kelly is almost equal on average to those of Fixed stake size. The cost matrix and converted cost matrix are shown in Table 6-5, and Table 6-6 respectively.

Table 6-5: Cost matrix based on average win max odds of the training set

	Predicted: Home	**Predicted: Away**	**Predicted: Draw**
Actual: Home	$-(\overline{wo}_{home}-1)$	1	1
Actual: Away	1	$-(\overline{wo}_{away}-1)$	1
Actual: Draw	1	1	$-(\overline{wo}_{draw}-1)$

Table 6-6: Converted cost matrix based on average win max odds of the training set

	Predicted: Home	**Predicted: Away**	**Predicted: Draw**
Actual: Home	0	$\overline{wo}_{home}$	$\overline{wo}_{home}$
Actual: Away	$\overline{wo}_{away}$	0	$\overline{wo}_{away}$
Actual: Draw	$\overline{wo}_{draw}$	$\overline{wo}_{draw}$	0

From Table 6-6 the cost matrix is formed as

$$cost\ matrix = \begin{cases} 0 & if\ predicted = actual \\ \overline{wo}_{outcome} & if\ predicted \neq actual \end{cases},$$

where $\overline{wo}_{outcome}$ is the average from maximum (based on 3 bookmakers) winning odds of all instances in training set for the given outcome.

Summarizing the cost matrix scheme applied in the experimental part is oriented to achieve higher payouts from betting and is not optimized directly to the Sharpe ratio.

6.7.6 Hyperparameters and Optimization

Tuning of hyperparameters is a crucial step of the overall experimental procedure. The main aim of tuning is to discard hyperparameter values that generate poor results.

The two tuning methodologies described in section 5.4, i.e., grid search and genetic algorithm (GA), were employed. As stated in section 5.4 the grid search exhaustively searches for the best-performing combination related to the evaluation metric while GA is an evolutionary approach that can search large and complex hyperparameter spaces efficiently. The choice between the two methods depends on the size of the hyperparameters space and the computational cost of the particular model under examination. Grid search is used when dealing with models that have a small hyperparameter space and a small discrete grid space. However, when the hyperparameter space of the model under examination is large and, in the cases, when continuous hyperparameters are involved the GA search is preferred. For example, when KNN utilizes Colley ratings as features, then is easier to apply grid search. However, when KNN utilized GeM ratings as features where the weights of GeM have a continuous range then the GA search is preferred.

At this point, we have to mention that the terms "hyperparameters" and "hyperparameter tuning" are usually used in the field of Machine Learning. In the sports rating systems, the relevant terms that are used frequently are "parameters" and "parameters calibration". However, for simplicity reasons, we refer to them as "hyperparameters" and "hyperparameter tuning" for both cases.

In this application, we have used fixed and variable hyperparameters. The first is held constant over the different seasons while the second is adjusted based on the best-performing combination of the past 4 sports seasons. For example, if we perform the AWF for the sports season 2009/10 then we will use the optimal values of hyperparameters that are tuned for the 2005/06 to 2008/09 seasons. To avoid overfitting during tuning we have employed the AWF on every candidate model from search (grid search or GA). Figure 6-8 illustrates the above example.

Each candidate model starts predictions from the 4th match week of each season and the AWF is restarted at the beginning of every new sports season. This implies that every model undergoes training and testing around 140 times, which is equivalent to 35 match weeks (38-3=35 match weeks because AWF started from the 4th match week) multiplied by 4 sports seasons. Since AWF restarts every season, the predictions are aggregated over the 4 seasons. The selection of the 4 sports seasons sliding window has relied on the fact that using more recent data is correlated with upcoming games and

consequently can help the optimization process to obtain more efficient and consistent results. Moreover, a larger window size leads to more computational time and cost.

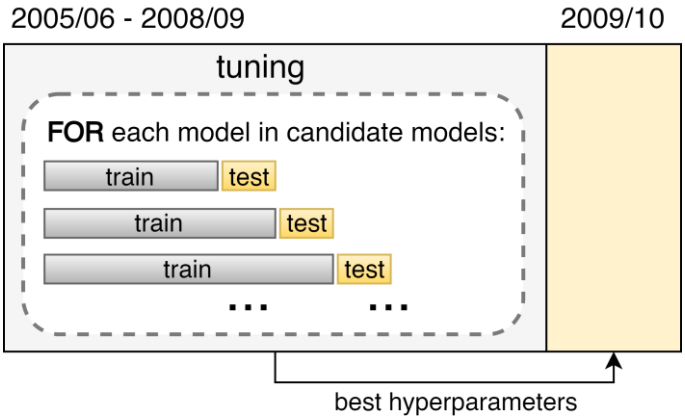


Figure 6-8: Tuning hyperparameters example

To address the selection of hyperparameters two batches of optimization for the same hyperparameters were performed: one for accuracy-oriented and one for profit-oriented approach. The optimization of hyperparameters both for rating systems and machine learning classifiers is done simultaneously to achieve the best possible performance and avoid overfitting. The criteria used to choose the optimal values of hyperparameters depend on the purpose of the experiment, i.e., the Accuracy metric for accuracy-oriented approaches and the Sharpe ratio for profit-oriented approaches.

When employing the GA the settings established include a pool of 10 phenotypes and a total of 50 generations in order to complete the search and return the best solution found. The fitness function is determined by the Accuracy metric for accuracy-oriented models and the Sharpe ratio metric for profit-oriented models.

The overall procedure of tuning essentially utilizes the rolling walk-forward (RWF) and it is described algorithmically in Figure 6-9.

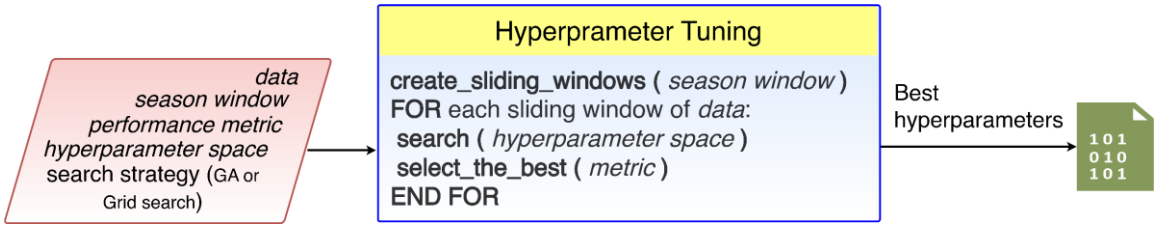


Figure 6-9: Tuning procedure

The inputs consist of the data, the number of seasons for the sliding window, the performance evaluation metric, the hyperparameter space, and the search strategy (i.e., GA, grid search). The procedure initially creates the sliding windows and then for each

sliding window searches the hyperparameter space and finally selects the best-performing hyperparameter values based on the total performance of evaluation metrics. Finally, the output is the best-combined hyperparameters for each model.

Figure 6-10 demonstrates how the procedure of RWF works for tuning for the period 2005/06 to 2017/18. Since our dataset starts from 2005/06 the computed ratings can be used for the predictions of upcoming matches from 2009/10.

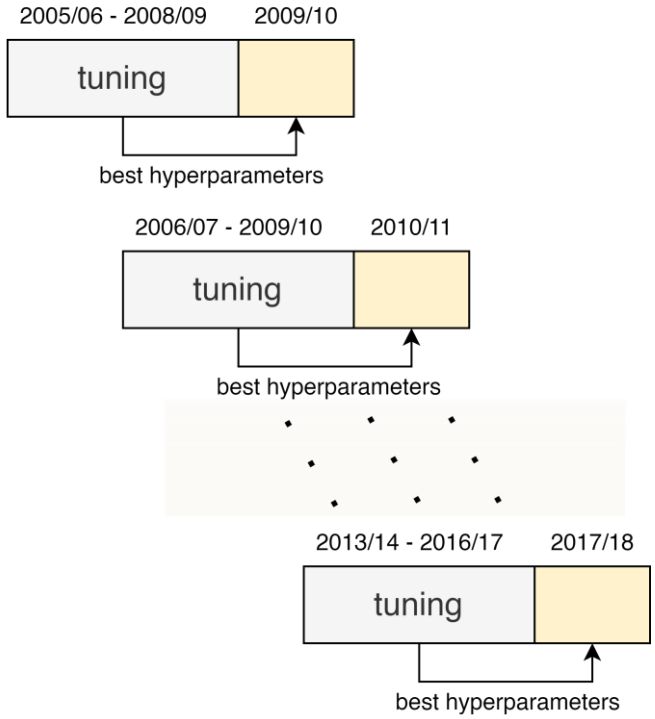


Figure 6-10: Tuning hyperparameters - RWF

The remainder of this subsection provides information about the fixed and variable hyperparameters. Notably, for the fixed hyperparameters, their constant values are shown while for the variable hyperparameters, their ranging values (hyperparameter space) are presented. It is also worth noting that sometimes a part of calculations can be avoided. An example where the calculations can be shortened is during the weights tuning of GeM and PointRATE. In each case, below is explained in more detail the final rating computations for each weight combination without additional calculations. Subsequently, the fixed and variable hyperparameters of each rating system are presented first. Then we continue with the hyperparameter space of machine learning classifiers and finally, we shall refer to some details about tuning.

❖ Rating systems - Hyperparameters:

The hyperparameters below are the same for accuracy and profit-oriented approaches. In each rating system, the hyperparameters that are held constant are presented first.

- Win-Loss and Keener

  Normalization based on the games played by each team, is employed to produce reliable ratings since the teams may have a different number of games played (due to postponed or rescheduled matches).

- Colley and AccuRATE: -

- Massey

  A limit of more than 20 games has been set to start the rating of teams. This number has been selected to provide enough games, and it ensures that the games graph is connected.

- Offense - Defense (ODM)

  After experimenting with the tolerance level, 0.0001 was selected.

- Elo-Win and Elo-Point

  The values of *k, ξ, HA* (Home-field Advantage) are determined from tuning based on the previous 4 seasons by searching the following continuous ranges of values:

  · *k*: [1, 100], *ξ*: [100, 500], *HA*: [0, 200].

- GeM

  The game statistics utilized are *TW*, *TG*, *TST*, and *TS* and their voting matrices have been formed in the same manner as stated in our example in section 3.2. The damping factor *b* was set to 0.85 and its choice was made after experimentation with the rate of convergence speed. As for their associated weights, are selected after tuning based on the previous 4 sports seasons' data. The weights related to game statistics were optimized according to the evaluation metric. As indicated above, the recalculation of stochastic matrices' elements can be avoided. Particularly, if we apply equation (3.18) of GeM for the 4 game statistics then

  $$G = a_{TW}S_{TW} + a_{TG}S_{TG} + a_{TST}S_{TST} + a_{TS}S_{TS},$$

  where $0 \leq a_i \leq 1,\ i \in \{TW, TG, TST, TS\}$ and $a_{TW} + a_{TG} + a_{TST} + a_{TS} = 1$

  As we can see, regardless of the weight values (a_i) the stochastic matrices S_{TW}, S_{TG}, S_{TST}, and S_{TS} remain the same and hence can be calculated at once. The final stochastic matrix can be computed later by the equation (3.19).

- PointRATE

  The modeling of soccer team ratings described in section 4.3.6 was employed. The weights of attributes are determined after tuning in a similar manner, as described in the GeM. The rating of a team could be calculated without additional computations for each different combination of weights. More specifically, the reward (value) function can be calculated once for each attribute of the team, and then for each weight combination, we apply the weights' values.

  Below we show the computation of the rating for team *i* based on (4.10):

$$r_i = w_{TW} \cdot v_{TW}(x_{i,TW}) + w_{TG} \cdot v_{TG}(x_{i,TG}) + w_{TST} \cdot v_{TST}(x_{i,TST}) + w_{TS} \cdot v_{TS}(x_{i,TS}),$$

  where $0 \leq w_j \leq 1, j \in \{TW, TG, TST, TS\}$, and

$$w_{TW} + w_{TG} + w_{TST} + w_{TS} = 1$$

  The values of *v* are computed at once and then stored. For each combination, the final rating r_i is produced if the weights are applied.

❖ Machine learning classifiers - hyperparameters:

The hyperparameter space is identical for both accuracy and profit-oriented models and is listed below for each classifier. In the case of fixed hyperparameters, those are presented first.

- Naive Bayes: -
- Logistic Regression

  The number of maximum iterations used for solver convergence was set to 1000 and the L2 penalty was fixed for all solvers.

  - solver: lbfgs, newton-cg, sag
  - inverse regularization parameter: 0.001, 0.01, 0.1, 1, 10, 100
  - multi-class strategy one-vs-rest, multinomial

- Decision Trees and Random Forests

  The selection of hyperparameters was made in a common base for Decision Trees and Random Forests. The maximum features parameter was set to *m*.

  - number of estimators: 10, 15, 25 (for random forest only)
  - split criterion: gini, entropy
  - maximum depth: 2, 3, 5, 10

- minimum number of samples required to be a leaf node: 1, 5, 8
- minimum number of samples required to split an internal node: 2, 3, 5

▪ Neural Network

Maximum iterations were set to 500 without shuffling the data.

- solver for weight optimization: sgd (stochastic gradient descent), adam
- learning rate: 0.001, 0.01, 0.1, 1
- activation function: logistic (sigmoid function), tahn (hyperbolic tan function), relu (rectified linear unit function)
- number of hidden units in layer: 5, 10, 15, 20

▪ Support Vector Machines

The shape of the decision function was set to a one-vs-rest (or one-vs-all) approach. Moreover, the kernel coefficient is used only for rbf and sigmoid kernel types.

- regularization parameter (C): 0.001, 0.01, 0.1, 1, 10, 100
- kernel type: linear, rbf (radial basis function), sigmoid
- kernel coefficient (gamma) for rbf and sigmoid: $1 / [m \times Var(tds)]$, $1 / m$

▪ K-Nearest Neighbors

- number of neighbors: 3, 5, 7, 9
- distance metric: Euclidean, Manhattan
- weight function in prediction: uniform weights, inverse distance

where:

m = number of features (m=2 for the ML[rs] feature set: home and away team ratings and m=5 for the ML[rs+odds] feature set: home, away team ratings and average odds)

tds = training data set without final outcomes (sample size differs each week because of the walk-forward procedure).

The hyperparameters listed above are fully documented in the scikit-learn (Pedregosa, et al., 2011) manual.

As mentioned before GA is used when at least one hyperparameter has a continuous range. Since WL, Colley, Massey, Keener, ODM, and AccuRATE have only fixed hyperparameters or implemented without hyperparameters, then in RANK and MLE methods the tuning is not required. Finally, as presented above all ML classifiers have discrete hyperparameters. Therefore, if the rating system's hyperparameters are discrete or tuning is not required grid search is employed; whereas, if they are continuous, i.e., hyperparameters of Elo-Win, Elo-Point, GeM, and PointRATE, the GA is used. Table 6-7 provides information about which tuning method was employed for each prediction technique and how many models were optimized.

Table 6-7: Tuning method used per prediction technique

Prediction Technique	Grid Search		Genetic Algorithm	
	Rating System	**Models**	**Rating System**	**Models**
RANK	-	-	Elo_{win}, Elo_{point}, GeM, PointRATE	4
MLE	-	-	Elo_{win}, Elo_{point}, GeM, PointRATE	4
ML	WL, Colley, Massey, Keener, ODM, AccuRATE	36*	Elo_{win}, Elo_{point}, GeM, PointRATE	28**

* ML has 36 models in Grid Search because 6 classifiers are combined with 6 rating systems that have only fixed hyperparameters. Naive Bayes is not included because rating systems are not required to be tuned.

** ML has another 28 models when the GA is used that resulted from the 4 rating systems combined with 7 classifiers. Naive Bayes is included because the ratings are computed after tuning.

6.7.7 Implementation Details

❖ General

All the experimental part was implemented in Python. Some details are also discussed in (Talattinis & Stephanides, 2022) and presented in Chapter 8. The essential components of this application are shown and described below:

- Data preparation: This component makes the preprocessing and then prepares the dataset by calculating the ratings and statistics of teams for every sports season. The datasets are stored in files to avoid recalculations every time we run new

experiments. The Pandas library (pandas development team, 2020; McKinney, 2010) was chosen for handling data as it is open-source and provides several useful functions.

- Experiment: All the details of the simulation and the performance evaluation metrics (i.e., Accuracy, F1-score, RPS, Sharpe Ratio, ROI, ARR, APB, etc.) of the prediction model and betting portfolio are stored in this data structure. Each experiment can be considered as a prediction model or a betting portfolio when belongs to the accuracy-oriented or profit-oriented approach respectively.
- Experimental container: A special data structure is implemented and allows us to conduct a large set of experiments.
- Tuning hyperparameters procedure: The tuning of hyperparameters for rating systems and machine learning algorithms is performed in this component. After tuning, the optimal hyperparameters are stored in a serializable file.
- Prediction procedure: The prediction techniques can be performed in 3 different ways as described in section 6.6. Each prediction technique is represented by a prediction procedure that makes predictions for upcoming matches. Additionally, predictions for evaluation purposes have a different prediction procedure.
- Sport Backtester: The backtester simulates the soccer matches weekly and calls the prediction and betting procedures. At the beginning of each sports season, the backtester is responsible for organizing the data by creating the walk-forward splits that are intended for the prediction procedure. The implementation of the backtester has been made with the aim of serving several experiments with different objectives simultaneously. Particularly, the experiments can be either accuracy-oriented or profit-oriented, with different evaluation metrics. Furthermore, the backtester can perform the experiments in a parallel way.
- Report Analyzer: Initially, when the backtesting is done, this component analyzes the results of experiments. Finally, it generates a report containing all statistics, metrics, and comparisons among experiments. This final report can be either written to a file or printed to the console.

❖ Rating systems

All rating systems described in this thesis have been implemented by exploiting several functions of NumPy (Harris, et al., 2020) and SciPy (Virtanen, et al., 2020)

libraries in Python that are intended for algebraic and scientific computations. Particularly, NumPy was used:

(1) matrices and vectors handling
(2) linear systems solving
(3) finding eigenvalues and eigenvectors
(4) other problems of linear algebra required for the implementation of rating methods

For the estimation of parameters used in the computation of the MLE, we have used the function "minimize" from the "optimize" module of the SciPy library. As for the statistical tests, such as Kendall's tau for the correlation of ranking lists, SciPy was used as well.

- Machine Learning procedure

  The classifiers of scikit-learn (Pedregosa, et al., 2011) were used for our machine learning predictions. Scikit-learn is an open-source library in Python that offers machine learning functions and several data science tasks.

- Cost-sensitive implementation

  The cost-sensitive classifier was implemented in Python and it was based on the implementation offered by the WEKA open-source software (Hall, et al., 2009). Note, that if the classifier in scikit-learn supports sample weights then only the sample weights vector is passed during the training phase.

- Computational Environment and Time

  The tuning procedure and the experiments of this application were run on a computer with the following features: Intel Core - i7-8550U CPU, 16GB RAM of DDR4, and 128 GB SSD. For accuracy and profit-oriented approaches, the total time required for hyperparameter tuning was 190 hours and for the final experiments was 37 minutes.

6.7.8 Experimental Procedure

In this subsection, we will explain the experimental procedure that was used to do so. Initially, we collect the data from football-data.co.uk for the sports seasons 2005/06 to 2017/18, then the experimental procedure is divided into three steps explained below. Figure 6-11 illustrates a brief diagram of the experimental procedure. The dotted line border indicates that the tuning procedure is repeated for each 4-season sliding window.

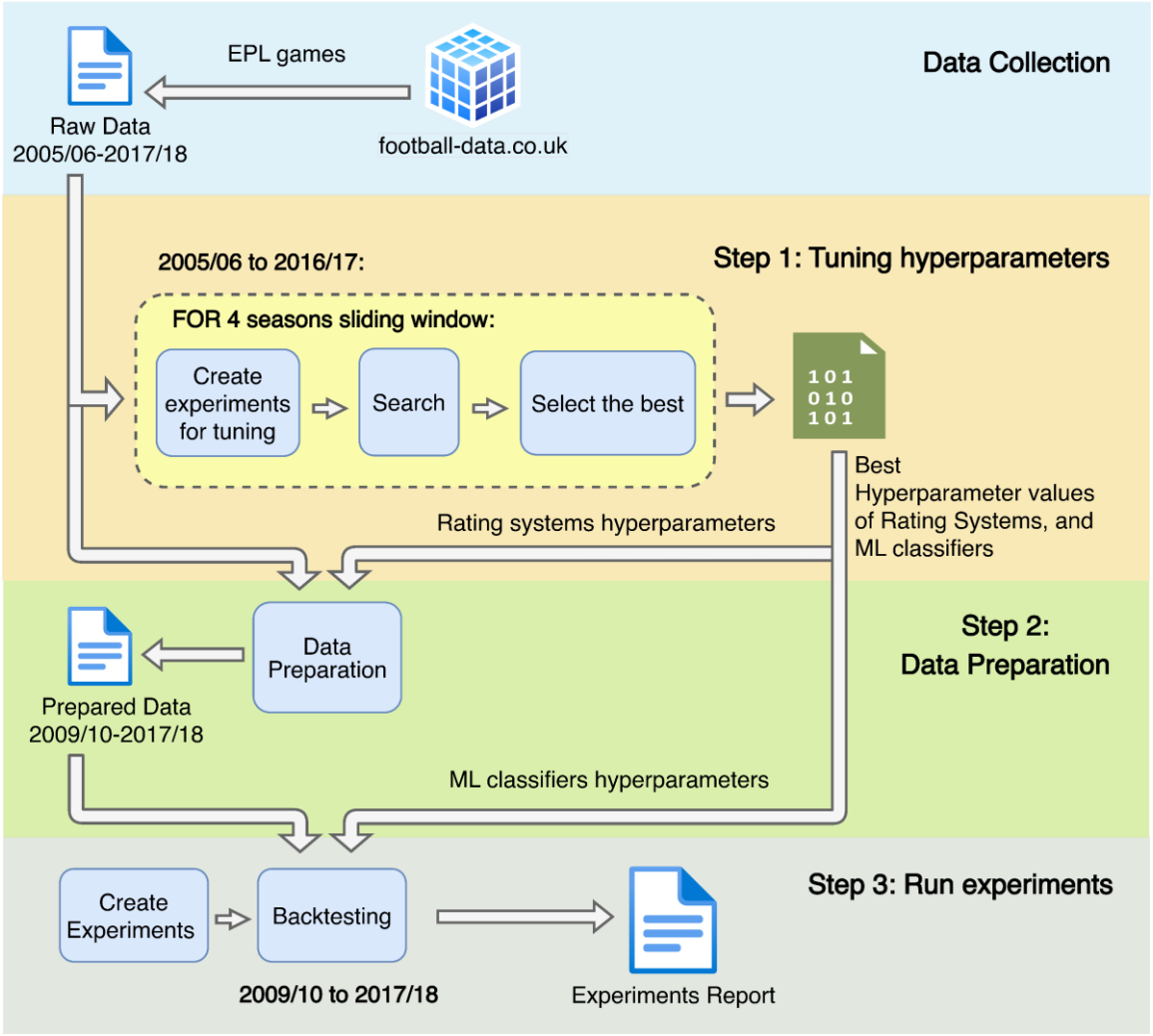


Figure 6-11: Experimental procedure steps

- Step 1: Tuning hyperparameters of rating systems and machine learning classifiers

  In the first step, we tune the hyperparameters of rating systems and machine learning classifiers to discover the best values according to the purpose of each experiment. For a sliding window of 4 seasons, the best hyperparameters are selected and stored for use in the next step. Particularly, the tuning procedure starts from 2005/06 to 2008/09 then follows 2006/07 to 2009/10, and continues in the same way until 2013/14 to 2016/17 seasons.

- Step 2: Data Preparation

  In the second step, the data preparation has as inputs the raw data of soccer matches and the best hyperparameters that have been selected in the first step. For each sports season, the statistics and ratings of the teams are computed and stored. The stored hyperparameter values of rating systems of the first step are obtained and applied in

the rating computations. Note that this step starts from the 2009/10 sports season since the previous step started from the 2005/06 season and consequently 2009/10 is the first sports season in which tuned hyperparameters are available. For better understanding, in the 2009/10 season, we used the values of the hyperparameters that performed best (according to the evaluation metric) from 2005/06 to 2008/09.

- Step 3: Run experiments

  In the final step, we create the experiments, and then the backtesting (simulation) is performed. The training of classifiers uses the stored hyperparameters of the first step. The backtesting starts from the 2009/10 sports season since it is the first season where the best hyperparameters of classifiers are available and it ends in the 2017/18 sports season. As noted in subsection 6.7.2, the AWF process restarts at the beginning of every season in each model and the predictions from the different seasons are combined. The performance evaluation metrics and comparisons among experiments are generated as a report after the backtesting.

6.7.9 Experimental Design and Comparison Steps

Our experimental design exhibits a step-wise behavior. The experiments of each approach (accuracy or profit-oriented) produced by the experimental procedure are further analyzed in terms of their average performance based on the prediction technique used. The overall experimental design is based on comparison steps and entails:

(1) Selection of prediction technique

Comparison of the performance of each prediction technique.

(2) Selection of Money Management

The efficiency between Fixed Amount stakes and the Kelly criterion is examined.

(3) Improvement

The focus in this step is on the employment of cost-sensitive techniques to minimize the average cost per prediction (or to maximize the benefit per prediction). Since for cost-sensitive learning the odds are used to form the cost matrix, it can also be included as machine learning features for classifiers.

(4) Selection of best models/algorithms

The selection of best-performing models is based on evaluation metrics and overall performance.

(5) Evaluation

The final step is the evaluation phase where our top-performing models are compared to baseline models.

For accuracy-oriented models, we omit the 2nd and 3rd steps. The steps until the improvement stage are tackled in the next section, while the steps of selection of top models and evaluation are presented in section 6.9.

6.8 Experimental Results

In this section, the experimental results are categorized according to their respective approach. Therefore, two subsections are included, 6.8.1 for accuracy-oriented experiments and 6.8.2 for profited-oriented experiments. In both approaches, the analysis of experimental results involves the utilization of statistical tests to compare the performance of prediction techniques. The appropriate statistical test is chosen depending on the characteristics of comparison group results, whether they are paired or unpaired, and their distribution. All statistical tests are two-tailed and each one focuses on the average performance score for a specific evaluation metric of two different prediction techniques. The null and alternative hypotheses for all tests are:

- H_o: There is no significant difference between the mean performance of the two prediction techniques at a significance level of a=0.05.
- H_1: The mean performances of the two prediction techniques are significantly different in a significance level of a=0.05.

When the results of comparison groups are obtained from paired observations in which their differences are normally distributed (e.g., RANK vs MLE), then a paired t-test is employed to evaluate the statistical significance. Conversely, if the results do not meet the assumption of normality the Wilcoxon signed-rank test is performed. For the comparison between independent groups, such as when the samples have unequal size (e.g., RANK vs ML or MLE vs ML) then if the two samples exhibit a normal distribution an independent t-test or a Welch's t-test is performed when they have equal or unequal variances respectively. Alternatively, if the results of independent groups do not follow the normal distribution the Mann-Whitney U rank test is conducted.

6.8.1 Accuracy-Oriented

Starting with Table 6-8, the results of each predictive technique per rating system are demonstrated. The RPS metric was not computed for Rank-based predictions, as the prediction technique does not provide probabilities. As for machine learning predictions, the metrics are computed by averaging the results across all classifiers within their respective rating system (ML features). Following that, the mean performances of prediction techniques are evaluated through statistical tests, enabling a comparison between them and finding the best predictive technique according to the evaluation metric. The statistical analysis results are depicted in Table 6-9.

Table 6-8: Accuracy-oriented results per prediction technique

Rating	RANK		MLE			ML[rs]		
System	Accur.	F1-score	Accur.	F1-score	RPS	Accur.[1]	F1-score[1]	RPS[1]
WL	0.4879	**0.4671**	0.5067	0.4169	0.2078	0.4837	0.4340	0.2164
Colley	0.4793	0.4178	0.5105	0.4241	0.2069	0.4847	0.4342	0.2161
Massey	0.4886	0.4258	0.5179	0.4323	0.2049	0.4857	0.4425	0.2176
Elo$_{win}$	0.4828	0.4194	0.5111	0.4255	0.2069	0.4770	0.4297	0.2186
Elo$_{point}$	0.4924	0.4258	0.5130	0.4281	0.2065	0.4787	0.4351	0.2169
Keener	0.4854	0.4221	0.5153	0.4285	0.2058	0.4881	0.4419	0.2164
ODM	0.4860	0.4235	0.5048	0.4086	0.2088	0.4763	0.4283	0.2227
GeM	0.4971	0.4317	0.5188	**0.4336**	**0.2025**	0.4918	0.4492	**0.2128**
AccuRATE	0.4898	0.4260	0.5143	0.4312	0.2056	0.4877	0.4462	0.2147
PointRATE	**0.5038**	0.4376	**0.5197**	0.4264	0.2040	**0.4932**	**0.4527**	0.2138
Average	0.4893	0.4297	**0.5132**	0.4255	**0.2060**	0.4847	**0.4394**	0.2164

1: The results per rating system regard the average performance of all classifiers tested

Before proceeding to the comparisons of prediction techniques, we can notice from Table 6-8 that our proposed system PointRATE achieved the highest Accuracy in RANK and MLE techniques and the highest average Accuracy in the ML technique.

The results of Table 6-8 suggest that the MLE consistently outperforms Rank-based models in terms of the Accuracy metric. Also, the MLE achieves higher Accuracy than ML models. However, the ML models perform better in terms of the F1-score over Rank-based and MLE. This implies that ML models achieved a balance between

precision and recall. Also, the probabilistic predictions made by the MLE models are more successful than those of the classification models, in terms of the RPS metric. In order to ensure the validity of the conclusions of this paragraph, statistical tests were conducted with the aim of determining whether the observed differences between prediction techniques are statistically significant or simply due to random variability.

Based on the p-values that are lower than 0.01 in Table 6-9 it can be concluded that there are significant differences in the mean performances which are consistent with the above conclusions. However, in the following cases: RANK vs MLE in terms of F1-score and RANK vs ML in terms of Accuracy, we can't reject the null hypothesis and therefore no conclusion can be drawn. This can be explained because RANK and MLE techniques focus only on home and away win classes, thus they have similar performance in the F1-score. Also, the ML models have lower Accuracy than MLE because they provide predictions in all outcomes and their Accuracy is similar to Rank-based models.

Table 6-9: Comparison tests for the accuracy-oriented approach

Comparison Groups		**Test Type**	**Metric**	**t-stat.**	**p-value**
Group 1	**Group 2**	**Two-tailed**			
RANK	**MLE**	paired t-test	Accuracy	-13.8007	2.3E-07***
		Wilcoxon signed-rank test	F1-score	27.0000	1.0^{ns}
RANK	**ML**	Mann Whitney U test	Accuracy	401.0000	0.4625^{ns}
		Mann Whitney U test	F1-score	162.5000	0.0065**
MLE	**ML**	Mann Whitney U test	Accuracy	695.0000	5.4E-07***
		Welch's t-test	F1-score	-5.0023	1.4E-04***
		Mann Whitney U test	RPS	19.5000	1.6E-06***

ns: not significant; ** p-value<0.01; *** p-value < 0.001

After the statistical tests, the results derived from the ML approach are further analyzed based on their average performance per rating system (ML features) and then per classifier (for the 10 rating systems) which are depicted in Table 6-10 and Table 6-11 respectively. Also, the analytical results of machine learning models' Accuracy, F1-score, and RPS are demonstrated in Figure 6-12 where the horizontal (x) axis represents the rating system utilized as ML features while the vertical (y) axis represents the classifier. Consequently, the value of each cell indicates the performance of the model for the respective metric (e.g., KNN with WL ratings as features has an Accuracy of 0.4947).

Table 6-10: ML performance per rating system in accuracy-oriented approach

M.	S.	WL	Colley	Massey	Elo_w	Elo_p	Keener	ODM	GeM	Accu RATE	Point RATE
Accuracy	avg	0.484	0.485	0.486	0.477	0.479	0.488	0.476	0.492	0.488	**0.493**
	min	0.465	0.465	0.461	0.450	0.451	0.468	0.452	0.476	0.465	0.469
	max	0.496	0.503	0.501	0.496	0.494	0.501	0.485	0.505	0.502	**0.511**
	std	0.010	0.013	0.015	0.014	0.017	0.011	0.012	0.010	0.014	0.015
F1-score	avg	0.434	0.434	0.443	0.430	0.435	0.442	0.428	0.449	0.446	**0.453**
	min	0.426	0.422	0.425	0.417	0.425	0.428	0.413	0.435	0.434	0.435
	max	0.440	0.448	0.455	0.441	0.451	0.456	0.440	0.458	0.457	**0.465**
	std	0.005	0.009	0.010	0.009	0.009	0.010	0.009	0.007	0.009	0.010
RPS	avg	0.216	0.216	0.218	0.219	0.217	0.216	0.223	**0.213**	0.215	0.214
	min	0.210	0.209	0.207	0.210	0.210	0.209	0.212	**0.205**	0.208	0.206
	max	0.228	0.227	0.234	0.232	0.227	0.226	0.234	0.224	0.224	0.225
	std	0.006	0.007	0.010	0.008	0.007	0.006	0.009	0.007	0.006	0.007

M.: Metrics; **S.**: Statistics

Table 6-11: ML performance per classifier in accuracy-oriented approach

Metrics	Statistics	NB	LR	DT	RF	NN	SVM	KNN
Accuracy	avg	0.4886	**0.4987**	0.4752	0.4859	0.4914	0.4896	0.4636
	min	0.4692	0.4841	0.4612	0.4755	0.4812	0.4768	0.4495
	max	0.5010	**0.5114**	0.4959	0.5022	0.5019	0.4984	0.4771
	std	0.0098	0.0076	0.0100	0.0076	0.0068	0.0065	0.0099
F1-score	avg	**0.4503**	0.4410	0.4376	0.4438	0.4340	0.4320	0.4369
	min	0.4335	0.4134	0.4253	0.4312	0.4172	0.4208	0.4241
	max	**0.4646**	0.4519	0.4584	0.4614	0.4505	0.4523	0.4530
	std	0.0096	0.0115	0.0103	0.0100	0.0106	0.0101	0.0100
RPS	avg	0.2135	**0.2084**	0.2242	0.2126	0.2132	0.2168	0.2273
	min	0.2071	**0.2051**	0.2167	0.2080	0.2090	0.2145	0.2239
	max	0.2308	0.2115	0.2337	0.2151	0.2179	0.2205	0.2341
	std	0.0067	0.0019	0.0050	0.0025	0.0029	0.0017	0.0034

NB: Naive Bayes; **LR**: Logistic Regression; **DT**: Decision Tree; **RF**: Random Forest; **NNs**: Neural Networks; **SVM**: Support Vector Machines; **KNN**: K-Nearest Neighbors

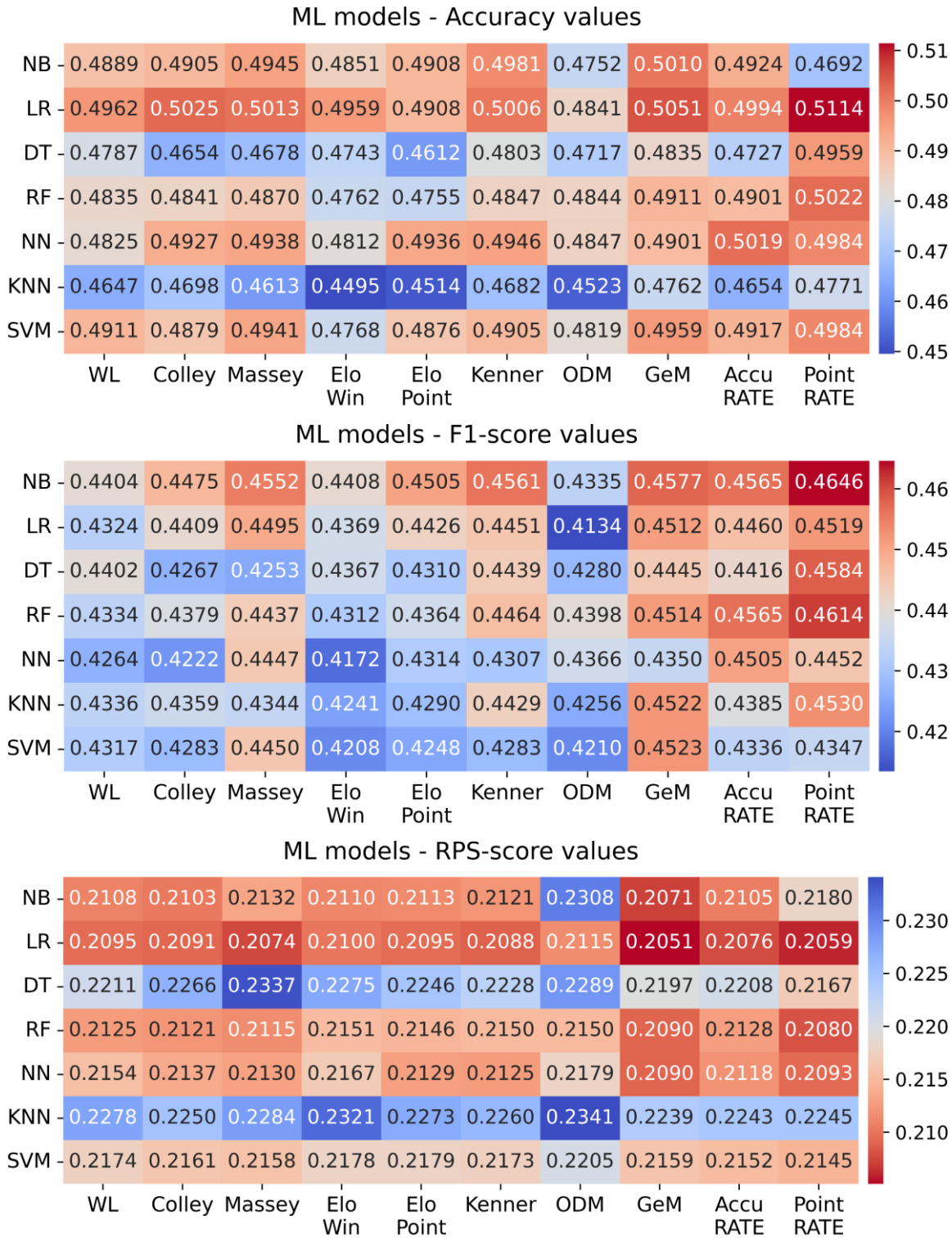


Figure 6-12: Accuracy, F1-score, and RPS per ML (accuracy-oriented) model

According to the results depicted in Table 6-10 the most effective rating system when utilized by any classifiers in terms of Accuracy rate and F1-score is the PointRATE, while in terms of the RPS is the GeM. The Offense-Defense ratings exhibit the poorest performance in all metrics as ML features. From the results of Table 6-11, on

average the top-performing classifier across all feature sets (rating scores) in terms of Accuracy rate and RPS score is the Logistic Regression, while in terms of F1-score is the Naive Bayes. Also, the worst-performing classifiers were determined to be the KNN, in terms of Accuracy rate and RPS score while in terms of F1-score is the SVM.

Moreover, Massey seems to have a better RPS score when its ratings are utilized by a statistical-based prediction technique. This conclusion is reached by focusing on the performance of the MLE models on probabilistic predictions (from Table 6-8) and specifically after comparing only the rating systems that utilize fewer attributes to generate ratings i.e., those that do not use *TST* and *TS* (all except GeM, PointRATE, and AccuRATE). Finally, from Table 6-8 and Figure 6-12, among all models, the ones with the top performance are PointRATE (MLE) with the highest Accuracy, Win-Loss (RANK) with the highest F1-score, and GeM (MLE) with the lowest RPS score.

6.8.2 Profit-Oriented

As already mentioned in subsection 6.7.9, the experimental design for profit-oriented experiments involves another two steps, the 2nd and the 3rd. More precisely, the 1st and the 2nd steps are merged because every prediction technique in the 1st step requires money management. As a result, we perform binary comparisons and proceed to every next step by taking into consideration the results of previous steps. In order to conclude which of the two variations outperforms every time, the statistical test between them from Table 6-12 and their mean performances from Table 6-13 are taken into account.

Table 6-12: Comparison tests for the profit-oriented approach

Comparison Groups		Test Type	Metric	t-stat.	p-value
Group 1	Group 2	Two-tailed			
RANK-Fixed	MLE-Fixed	paired t-test	Sharpe	-3.3714	0.0082**
MLE-Fixed	MLE-Kelly	paired t-test	Sharpe	-2.9579	0.016*
ML[rs]-Fixed	ML[rs]-Kelly	WSR test	Sharpe	696.0	0.0014**
MLE-Kelly	ML[rs]-Kelly	MWU test	Sharpe	207.0	0.0382*
ML[rs]-Kelly	ML[rs+odds]-Kelly	WSR test	Sharpe	970.0	0.1108ns
ML[rs]-Kelly	ML[rs+odds]-Kelly	WSR test	ARR	873.0	0.0306*
ML[rs]-Kelly	ML[rs+odds]-Kelly	WSR test	ROI	721.5	0.0023**
ML[rs+odds]-Kelly	CS[rs+odds]-Kelly	WSR test	Sharpe	846.0	0.0203*

ns: not significant; * p-value < 0.05; ** p-value < 0.01; **WSR** test: Wilcoxon signed-rank test; **MWU** test: Mann Whitney U test

Table 6-13: Profit-oriented results per prediction technique

Metric	Stat	RANK	MLE		ML [rs]		ML [rs+odds]	CS [rs+odds]
		Fixed	Fixed	Kelly	Fixed	Kelly	Kelly	Kelly
SR	avg	-1.7039	-0.8027	-0.4682	-0.4294	-0.2236	-0.1437	**0.0597**
	min	-3.0051	-1.4084	-1.1655	-3.2342	-2.2297	-2.4593	-1.3369
	max	-0.4103	-0.1158	0.0343	0.6459	0.4818	0.6437	**0.7414**
	std	0.7069	0.3969	0.3764	0.5825	0.5702	0.6394	0.4896
ARR	avg	-0.9632	-0.7629	-0.5406	-0.6094	-0.3694	-0.2888	**-0.2391**
	min	-0.9938	-0.8740	-0.7775	-1.0000	-1.0000	-0.9937	-0.9576
	max	-0.8613	-0.6216	-0.0007	0.1520	0.1416	0.1945	**0.2032**
	std	0.0517	0.0820	0.2548	0.2949	0.3833	0.3826	0.3936
ROI	avg	-0.9865	-0.9860	-0.8654	-0.7294	-0.4637	-0.1038	**0.1906**
	min	-0.9938	-0.9980	-0.9955	-1.0000	-1.0000	-0.9962	-0.9977
	max	-0.9800	-0.9768	-0.0060	2.5745	2.2938	3.9497	**4.2841**
	std	0.0048	0.0080	0.3079	0.7478	0.6814	1.0748	1.2736
ABS		0.0250	0.0250	0.0237	0.0250	0.0208	0.0236	0.0242
PBP%		0.00%	0.00%	0.00%	10.00%	27.14%	44.29%	**54.29%**
NBP%		0.00%	0.00%	20.00%	2.86%	20.00%	8.57%	5.71%
FBP%		100.00%	100.00%	80.00%	87.14%	52.86%	47.14%	**40.00%**

SR: Sharpe Ratio (annualized); **ARR**: Annualized Rate of Return; **ROI**: Return on Investment; **ABS**: Average Bet Size; **PBP:** Positive Betting Portfolio% (e.g., 50%: half of the models have ROI≥0); **NBP**: Negative Betting Portfolio% (e.g., 50%: half of the models have ROI<0 and sufficient capital to place bets); **FBP**: Failed Betting Portfolios% (e.g., 50%: half of the models failed and lost their initial capital)

Initially, the Rank-based is compared with the MLE method by applying Fixed stake size as money management in both techniques. Note that the Fractional Kelly Criterion cannot be applied to Rank-based predictions because the prediction technique does not provide probabilities. The mean performances of the two techniques are significantly different (p-value=0.0082<0.01) and according to their observed average performances, the MLE models are superior to Rank-based betting portfolios in terms of Sharpe ratio. However, in both techniques according to FBP, all predictions proved to be ineffective since all betting portfolios resulted in the loss of their initial capital. By considering that MLE seemed better than Rank-based, we next compare the MLE models

to their effectiveness in different money management strategies, where based on p-value=0.016<0.05 and their average performances, the MLE with Fractional Kelly outperforms the MLE with Fixed Amount. Before comparing the performances between MLE-Kelly and ML techniques, we first examined which money management performed better in ML models. The comparison between betting portfolios of ML models reveals significant differences (p-value=0.0014<0.01) in performance with the Fractional Kelly exhibiting an enhanced average Sharpe ratio. Following this, the average performance of betting portfolios of MLE and ML when both employed Fractional Kelly are compared and the results suggest that there is a significant difference (p-value=0.0382<0.05), and from the average Sharpe ratio, the ML models seemed to perform better.

After identifying the advantage of using the Fractional Kelly criterion to determine the betting stake size, we proceeded to the improvement step (3rd step) of experimental design where we tested the feasibility of using cost-sensitive learning to enhance the models. Given that the cost-sensitive approach utilized the betting odds to construct the cost matrix this implies that there is additional information. One way to ensure a fair comparison is to include this information, i.e., betting odds, in the cost-insensitive models as ML features. We referred to this feature set as the ML[rs+odds] which is also described in Table 6-4. However, as a preliminary step, we examined whether the incorporation of betting odds as features could enhance the average performance of cost-insensitive machine learning models. When the variation ML[rs+odds] is compared to ML[rs] in terms of the average Sharpe ratio, we can't reject the null hypothesis. For this reason, we proceeded to two additional comparisons based on ARR and ROI to make an informed decision on the superior variant. Based on the two statistical tests, there is a significant difference between the means and the inclusion of odds improves ARR (p-value=0.0306<0.05) and ROI (p-value=0.0023<0.01), thus for the final comparison, the odds will be included as features in both variants. Finally, we compared the betting portfolios generated by the models of cost-insensitive (ML[rs+odds]) with those of cost-sensitive (CS[rs+odds]). It can be concluded that there is a statistical difference between them (p-value=0.0203<0.05) and according to observed average performances of Sharpe ratio the cost-sensitive outperforms the cost-insensitive.

Figure 6-13 provides a graphical description of the overall results and shows that we proceed to every next step by considering the results of previous steps.

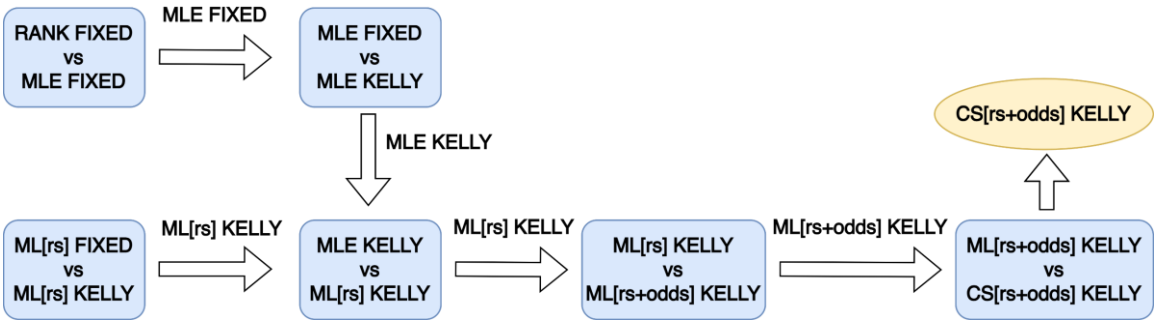


Figure 6-13: Comparisons in profit-oriented approach

Through a comparative analysis of the mean performances of different groups, cost-sensitive seemed to produce the best models. Also, FBP and PBP have the lowest and highest percentages respectively among the other techniques which implies that cost-sensitive learning improves the models. However, despite the overall improvements with Fractional Kelly and cost-sensitive learning the average Sharpe ratio is slightly positive but still low and ARR continues to show a negative average value. After the statistical tests, the results of the ML models from 3 variations are further analyzed and depicted in Table 6-14 based on their average performance per rating system (for the 7 classifiers).

Table 6-14: ML average performance per rating system in profit-oriented approach

Rating	**ML[rs]-Kelly**				**ML[rs+odds]-Kelly**				**CS[rs+odds]-Kelly**			
System	**SR[1]**	**ARR[1]**	**ROI[1]**	**PBP**	**SR[1]**	**ARR[1]**	**ROI[1]**	**PBP**	**SR[1]**	**ARR[1]**	**ROI[1]**	**PBP**
WL	-0.42	-0.46	-0.59	1/7	-0.01	-0.16	0.20	4/7	0.20	**-0.10**	0.58	4/7
Colley	0.04	-0.21	-0.27	3/7	-0.01	-0.21	0.05	4/7	0.08	-0.17	0.35	**5/7**
Massey	-0.01	-0.23	-0.42	1/7	-0.27	-0.35	-0.27	2/7	0.13	-0.21	0.42	4/7
Elo_{win}	-0.41	-0.53	-0.68	2/7	0.00	-0.25	-0.24	3/7	-0.02	-0.25	-0.02	4/7
Elo_{point}	-0.61	-0.50	-0.76	0/7	-0.21	-0.38	-0.33	3/7	**0.29**	-0.13	**0.85**	**5/7**
Keneer	-0.18	-0.40	-0.61	2/7	0.06	-0.27	-0.33	3/7	0.09	-0.28	-0.09	3/7
ODM	-0.27	-0.38	-0.11	3/7	-0.51	-0.49	-0.57	2/7	-0.31	-0.55	-0.50	2/7
GeM	-0.10	-0.24	-0.12	4/7	-0.46	-0.42	0.30	3/7	-0.07	-0.37	-0.08	3/7
AccuR.	-0.06	-0.39	-0.55	1/7	0.02	-0.13	0.07	4/7	0.15	-0.16	0.10	4/7
PointR.	-0.21	-0.35	-0.53	2/7	-0.04	-0.23	0.07	3/7	0.06	-0.18	0.32	4/7
Avg	-0.22	-0.37	-0.46	0.27	-0.14	-0.29	-0.10	0.44	**0.06**	**-0.24**	**0.19**	**0.54**

PBP: The ratio of Positive Betting Portfolio yielding ROI ≥ 0 out of the 7 classifiers tested; **1**: The results per rating system regard the average performance of all classifiers tested.

Evaluating the transition from ML[rs] to ML[rs+odds] and then to CS[rs+odds], the majority of betting portfolios per rating system demonstrate improvements, with the most significant observed in CS[rs+odds]. Focusing on cost-sensitive learning as the most advantageous selection, the Elo-Point is the most effective rating system in terms of the average Sharpe ratio when its ratings are utilized by any of the classifiers as ML features. Then, follow the Win-Loss and AccuRATE with the 2nd and 3rd highest average Sharpe ratios respectively. Also, when Elo-Point or Colley ratings are utilized as ML features, 5 out of 7 classifiers generate models that lead to betting portfolios with positive ROI. Finally, the rating system with the poorest performance is the Offense-Defense.

In Table 6-15 the 3 variations are analyzed based on their average performance per classifier (for the 10 rating systems) where it is evident that there has been a notable improvement in CS[rs+odds] per classifier. In cost-sensitive learning, on average the top-performing classifier across all feature sets (rating scores) in terms of Sharpe ratio and ARR is the SVM followed by the LR. In addition, all the models generated by SVM lead to positive betting portfolios (PBP=10/10). Finally, RF and KNN were identified as the worst-performing classifiers in all categories evidenced by their negative values across all metrics and from their small PBP ranging from 0 to 2 out of 10.

Table 6-15: ML average performance per classifier in profit-oriented approach

CLF	ML[rs]-Kelly				ML[rs+odds]-Kelly				CS[rs+odds]-Kelly			
	SR[1]	ARR[1]	ROI[1]	PBP	SR[1]	ARR[1]	ROI[1]	PBP	SR[1]	ARR[1]	ROI[1]	PBP
NB	0.01	-0.18	-0.31	3/10	-0.21	-0.48	0.22	4/10	0.14	-0.20	0.77	7/10
LR	0.17	0.004	0.09	7/10	0.15	0.01	0.14	9/10	0.18	0.02	0.18	8/10
DT	-0.41	-0.50	-0.43	2/10	-0.59	-0.56	-0.63	1/10	0.10	-0.27	**0.93**	5/10
RF	-0.46	-0.63	-0.90	0/10	-0.44	-0.47	-0.64	2/10	-0.36	-0.48	-0.57	2/10
NNs	-0.27	-0.59	-0.78	1/10	0.09	-0.14	0.38	6/10	0.13	-0.13	0.35	6/10
SVM	0.15	0.004	0.07	6/10	0.27	0.03	0.49	7/10	**0.43**	**0.05**	0.64	**10/10**
KNN	-0.74	-0.68	-0.98	0/10	-0.27	-0.42	-0.69	2/10	-0.21	-0.67	-0.97	0/10
Avg	-0.22	-0.37	-0.46	0.27	-0.14	-0.29	-0.10	0.44	**0.06**	**-0.24**	**0.19**	**0.54**

CLF: Classifiers; **PBP:** The ratio of Positive Betting Portfolios yielding ROI ≥ 0 out of the 10 feature sets (ratings) tested; **1**: The results per classifier regard the average performance across all feature sets (ratings).

The Sharpe ratios of betting portfolios derived from the three categories are demonstrated in Figure 6-14 where the horizontal (x) axis represents the rating system that its scores are used as ML features while the vertical (y) axis represents the classifier.

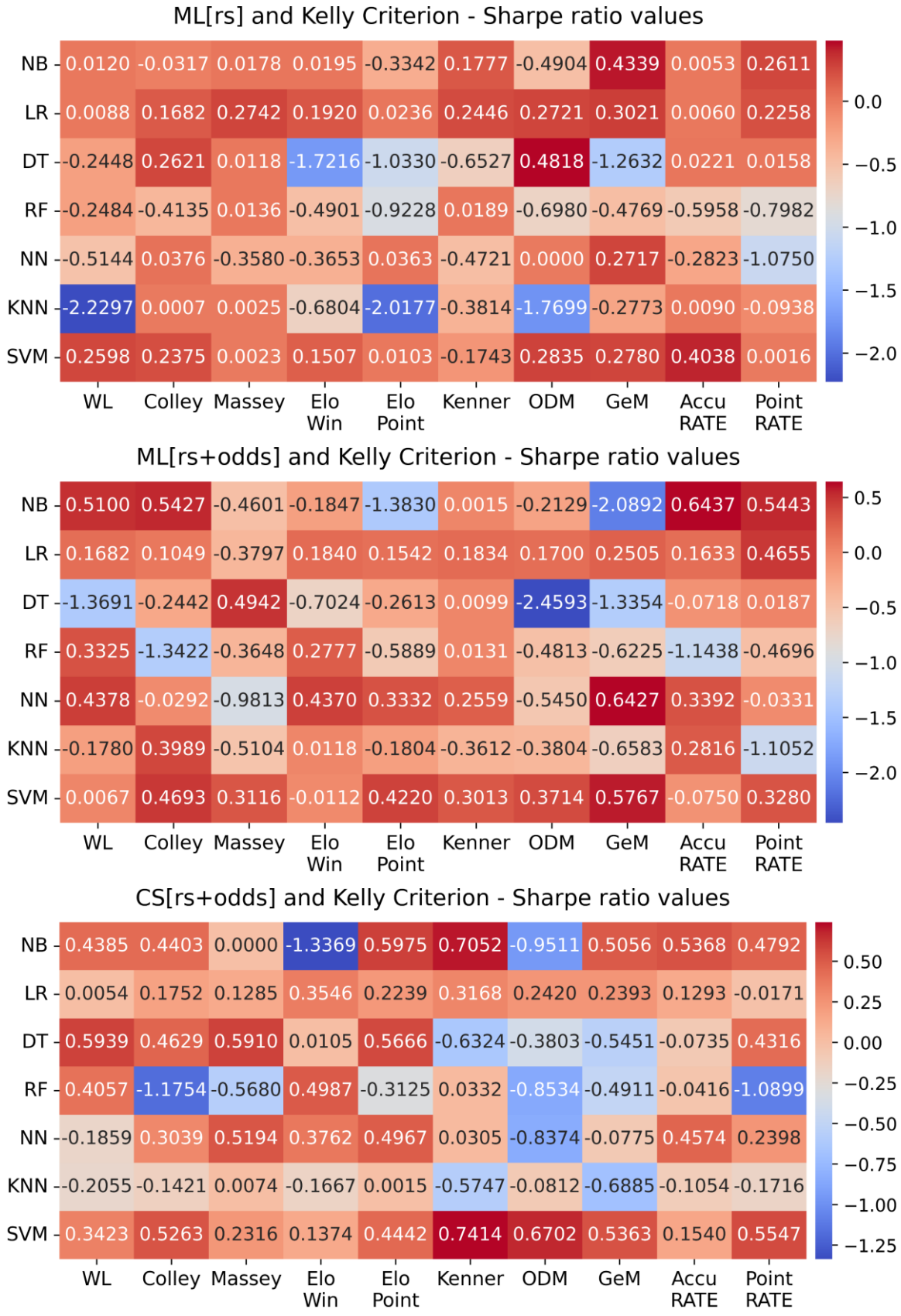


Figure 6-14: Sharpe ratio per portfolio by ML[rs], ML[rs+odds], and CS[rs+odds]

From Figure 6-14, the top-performing betting portfolio from all categories is derived from the combination of SVM with Keener in the CS[rs+odds] category with a Sharpe ratio of 0.7414. Also, our proposed system AccuRATE combined with NB in ML[rs+odds] is placed in the top 5 betting portfolios from all categories with a Sharpe ratio of 0.6437. In all categories, the Kelly criterion was used for money management.

6.9 Evaluation

To assess the effectiveness of our top-performing models we compare their performance with several baseline models and discuss the relative merits and drawbacks. Many of the baselines have been selected based on their use as baselines in previous studies (Hvattum & Arntzen, 2010; Herbinet, 2018; Lasek, Szlávik, & Bhulai, 2013).

- Models from Studies as baselines
  - Dixon and Coles model (Dixon & Coles, 1997): The prediction of the outcome is based on the probabilities given by applying the model. Some games at the beginning of each season are excluded from prediction when the model does not converge. Based on the authors' study where they utilized half-week data, the optimal value of parameter ξ in the weighting function was 0.0065. We adjusted it by dividing it by 3.5 to reflect our daily data, thus we used $\xi = 0.001857$. For the Factional Kelly, the maximum bet size percentage was set to 6.25% on initial capital, for a fair comparison with our models.
- Bookmaker odds baselines
  - Favorite: The prediction is based on the outcome with the lowest average odd. As a side note, the computation of the RPS score is based on the implied probabilities. Calculating the implied probability involves taking the inverse of the decimal odds (1/odd) associated with a particular outcome. Since bookmaker companies incorporate their profit margin in their offered odds, the sum of derived implied probabilities for the outcome of a soccer match will tend to be slightly higher than 1. Therefore, after the computation of implied probabilities for particular match outcomes, then they are normalized, to sum up to 1.
  - Outsider: The prediction is based on the outcome with the highest average odd.
- Naive baselines
  - Home-Only: The prediction in every match is always the Home-win.

- Away-Only: The prediction in every match is always the Away-win.
- Draw-Only: The prediction in every match is always the Draw.
- Random: The average performance metrics of 50 different random models are aggregated. Each model generates random predictions for the outcome of games. Also, the top-performing random models which exhibit the highest performance out of 50 random models are included for comparison in each approach. Specifically, two random models are selected in accuracy-oriented approach, each focusing on different performance metrics. The first model exhibits the highest Accuracy, while the second model has the highest F1-score. In the profit-oriented approach, the random model with the highest Sharpe ratio is selected. The random predictions can be established as a baseline performance measure model whereas any other model that performs better implies that it is learning something from data. On the other hand, if any model is worse than the random prediction model this indicates that the model needs to be improved. Two ways of random predictions are included: the uniform and the stratified. In uniform random predictions, each outcome has an equal chance of being selected while in a stratified way, the predictions are made in proportion to the class distribution. Note that for each method 50 different random models are generated.

6.9.1 Accuracy-Oriented - Comparison with Baselines

In this subsection, we compare our top-performing accuracy-oriented models with baseline models. Three top-performing models are chosen among all accuracy-oriented models based on their performance in relation to the specific metric, i.e., Accuracy, or F1-score, or RPS. Those are the models mentioned in subsection 6.8.1. The results are depicted in Table 6-16 and as we can observe from the confusion matrices, RANK, and MLE techniques could be considered to lack the ability to accurately predict draws, classifying them as home or away wins all of the time. Our top models' performances are noteworthy compared to the Dixon-Coles model. Specifically, PointRATE-(MLE), GeM-(MLE), and Win-Loss-(RANK) demonstrate better performance than Dixon-Coles in Accuracy, RPS, and F1-score respectively. However, considering the Accuracy and RPS metrics, our models do not outperform the Favorite model that represents the bookmakers' odds. On the other hand, Win-Loss-(RANK) our top-performing model in

F1-score achieved better results than bookmakers. Finally, our models were better than the Home-Only, Random Uniform, and Random Stratified models.

Table 6-16: Comparison of top accuracy-oriented models with baselines

	Model Name	Accur.	CG	WG	Confusion Matrix	F1-score	RPS
	PointRATE (MLE)*ACC.	0.5197	1635	1511	a\p H A D H 1317 150 0 A 565 318 0 D 639 157 0	0.4264	0.2040
	Win-Loss (RANK)F1**	0.4879	1535	1611	a\p H A D H 850 492 125 A 221 575 87 D 306 380 110	**0.4671**	-
	GeM (MLE)*RPS**	0.5188	1632	1514	a\p H A D H 1245 222 0 A 496 387 0 D 599 197 0	0.4336	0.2025
	Dixon-Coles	0.5018	1542	1531	a\p H A D H 1018 293 120 A 329 452 84 D 433 272 72	0.4651	0.2096
	Favorite	**0.5451**	1715	1431	a\p H A D H 1244 223 0 A 412 471 0 D 549 247 0	0.4609	**0.193**
	Home-Only	0.4663	1467	1679	a\p H A D H 1467 0 0 A 883 0 0 D 796 0 0	0.2966	-
UNIFORM	**Random1AVG**	0.3316	1043.1	2102.9	-	0.3377	-
UNIFORM	**Random*ACC.**	0.3449	1085	2061	-	0.3497	-
UNIFORM	**Random**F1**	0.3449	1085	2061	-	0.3497	-
STRATIFIED	**Random1AVG**	0.3588	1128.6	2017.4	-	0.3603	-
STRATIFIED	**Random*ACC.**	0.3694	1162	1984	-	0.3710	-
STRATIFIED	**Random**F1**	0.3687	1160	1986	-	0.3717	-

CG: Correct Games; **WG**: Wrong Games; * The top model in terms of Accuracy; ** The top model in terms of F1-score; *** The top model in terms of RPS; **1**: average values of 50 random models.

6.9.2 Profit-Oriented - Comparison with Baselines

In this subsection, we compare our top-performing betting portfolio which represents a profit-oriented model with the results from baselines. The top-performing betting portfolio is based on its highest performance in Sharpe ratio (SR) among all betting portfolios that are generated by the predictions of profit-oriented models. Table 6-17 demonstrates the performances of betting portfolios (TOP^1 and baselines) in SR, ARR, and ROI metrics. The R-square (R^2), and the slope values calculated from linear regression of the returns of each betting portfolio are also reported.

Table 6-17: Comparison of top betting portfolios with baselines

Betting Portfolio	MMT	SR	ARR	ROI	BN	FBP	R^2	Slope
TOP^1	Kelly	**0.7414**	0.0964	1.2896	1556	NO	0.917	0.72
Dixon-Coles	Fixed	-1.2885	-0.9935	-0.9935	217	YES	0.953	-4.11
Dixon-Coles	Kelly	-1.5555	-0.8988	-0.9898	393	YES	0.793	-1.50
Favorite	Fixed	-0.2653	-0.8483	-0.9770	539	YES	0.878	-1.73
Outsider	Fixed	-0.4514	-0.9800	-0.9800	181	YES	0.172	-1.63
Home-Only	Fixed	0.4600	0.1147	1.6578	3146	NO	0.023	0.07
Draw-Only	Fixed	-0.4520	-0.5942	-0.9890	1688	YES	0.083	-0.22
Away-Only	Fixed	-2.5158	-0.9768	-0.9768	83	YES	0.956	-11.90
$Random_{Uniform}^2$	Fixed	-0.8768	-0.7901	-0.9324	861.7	96%	-	-
$Random_{Uniform}^3$	Fixed	0.3760	0.0738	0.8988	3146	NO	0.003	0.04
$Random_{Stratified}^2$	Fixed	-0.7352	-0.7278	-0.9229	1080.7	94%	-	-
$Random_{Stratified}^3$	Fixed	0.3785	0.0035	0.0315	3146	NO	0.148	-0.32

1: Top-1 (based on SR) Cost-Sensitive - SVM with Keener ratings and average odds as features, and Kelly criterion as money management; **2**: Average values of the total 50 random models; **3**: The random model with the highest SR; **MMT**: Money Management Type; **BN**: Bets Number placed; **FBP**: Failed (Yes) or Not (No) or a percentage of Failed Betting Portfolio% out of the total 50 random models

Our top-performing betting portfolio (TOP^1) has shown remarkable performance and outperforms the baselines. Particularly, it exhibits a SR of 0.7414 and a satisfactory ARR while its high R^2 value of 0.91 implies its stability and the slope value of 0.72 is also high. Also, the TOP^1 outperforms the Dixon-Coles model concerning the SR and all the other metrics. Surprisingly the betting portfolio of the Home-Only achieved a positive ARR which can be attributed to the small betting size set. However, its

reliability remains uncertain due to low SR, R^2, and slope. Although the best betting portfolios from random predictions have positive profits, they exhibit very low values in the remaining metrics, and is important to note that the majority of random models in uniform and stratified methods failed at 96% (48/50) and 94% (47/50) respectively. Finally, all the other baselines have insufficient capital indicating their failure.

6.10 Conclusions

The empirical study and evaluation procedure of the work is based on EPL soccer games, simple historical data, and well-known bookmakers' market odds. In contrast to other studies, our focus was only on the simple data attribute and teams' statistics that can be utilized effectively by the rating systems. Also, to obtain a more comprehensive understanding of the performance, we employed numerous sports seasons of the EPL. Seeking to improve the performance of the models, the tuning of the hyperparameters was conducted by applying the grid search or genetic algorithm search. The ability to fine-tune the models, either for accuracy or for profitability, is a clear advantage over models constructed exclusively for either purpose.

The performance of our models appeared to be quite promising in relation to the small number of data attributes utilized. However, the models can be extended and use more data attributes. One possible improvement would be to examine the regression models to predict the margin of victory in games. Additionally, more complex algorithms such as boosting, bagging, and deep learning approaches could potentially improve the robustness of our prediction models. Also, the efficiency of binary predictions can be tested over the ternary. Furthermore, the study could benefit from the inclusion of sentiment analysis data related to sports teams or players which could help to generate more informed predictions. Finally, the involvement of data from betting exchanges may contribute to the development of better models.

The application and the experimental study of this chapter have provided valuable insights and have highlighted the importance of rating systems and their combination with statistical methods or machine learning algorithms in the prediction procedure. Despite some limitations and areas for improvement, the overall performance of the best models was promising. The conclusions, limitations, and possible future work are discussed for each approach separately below.

❖ Accuracy-Oriented

Three different evaluation metrics are examined for the accuracy-oriented models. The MLE method seems to work better for probabilistic predictions and provides the highest performance in the average RPS and Accuracy metrics. The statistical results of these experiments favor the machine learning classification models that performed better in terms of balanced predictions according to the weighted average F1-score. Given the utilization of a reduced number of data attributes (wins, goals, and shots), we can determine that the models have a satisfactory performance in relation to other studies that utilized numerous data attributes and teams' statistics. Also, our proposed system PointRATE with MLE achieved the highest Accuracy among all the models. Additionally, it has the highest Accuracy in every category (RANK, MLE, and ML).

While the derived models exhibit acceptable findings, none of them can outperform the bookmakers' odds (Favorite) in terms of Accuracy or RPS score. However, in F1-score, Rank-based predictions of Win-Loss and several ML models perform better than bookmaker odds (Favorite). The top-performing prediction techniques, rating systems, classifiers, and models per metric are shown in Table 6-18.

Table 6-18: Accuracy-oriented results - summary

Type	Metric: Accuracy	Metric: F1-score	Metric: RPS
Top Prediction Technique	MLE	ML	MLE
Top Model	PointRATE (MLE)	Win-Loss (RANK)	GeM (MLE)
Top RS as ML feature	PointRATE	PointRATE	GeM
Top classifier in ML	LR	NB	LR

One limitation to consider is that only basic classifiers have been used. This limitation could be addressed through the use of more advanced classification methods such as deep learning or ensemble approach, which can unlock new possibilities for improved accuracy and predictive power. Moreover, the combination of statistical methods with machine learning techniques can improve the models' overall performance. Also, to enhance the predictive ability of our models, one possible improvement would be to explore a broader hyperparameter space.

❖ Profit-Oriented

Although the measurement of prediction accuracy is a very important part of the validation of each model, its economic significance is studied in this approach. In terms

of profit-oriented models, this work aims to provide a unique insight into the very challenging world of sports betting, examining the predictions of game results from an economic perspective and by taking into account the risk.

The team rankings in a league do not guarantee an effective or profitable prediction of future matches. The sports betting market needs more than just a fundamental analysis of the product being traded (match outcomes). Correct prediction techniques, as well as a sound staking plan, are needed in order to be profitable in these markets. In this application, we tried to prove these statements by comparing raw models (Rank-based) against the statistical models and then against ML models. The generated models were used to virtually bet on soccer outcomes (Home-win/Away-win/Draw) and then a distinct hypothetical betting portfolio was constructed for each model. The Sharpe ratio metric is calculated and analyzed as a performance metric for the returns of each betting portfolio. The top models exhibit superior performance compared to baseline models. Particularly, the utilization of ratings from our proposed system AccuRATE and average odds as features by Naive Bayes and Kelly criterion as money management is placed in the top 5 best-performing betting portfolios in terms of risk-adjusted performance. Moreover, among the ratings used as features by ML classifiers in cost-sensitive learning, AccuRATE ratings hold the 3rd position in terms of their effectiveness in the average Sharpe ratio of betting portfolios. Also, the advantage of using the Kelly criterion implies that the probabilities from ML and MLE models are reliable. Table 6-19 shows the top prediction technique, rating system, classifier, and model/betting portfolio.

Table 6-19: Profit-oriented results - summary

Type	**Metric: Sharpe**
Top Prediction Technique	CS[rs+odds]
Top Model/Betting Portfolio	CS-SVM with features: Keener ratings and average odds
Top RS as ML feature	Elo-Point
Top classifier in ML	SVM

Realizing the significant effect that misprediction has on many real-world problems, our approach was focused on the way these costs could affect a hypothetical betting portfolio in terms of soccer outcome predictions. In our experimental analysis, we consider the potential influence of a cost-sensitive approach rather than traditional machine-learning methods. Our research confirms that it is worthwhile to employ cost-

sensitive methods for the successful predictions of soccer results and better investment opportunities. Also, this can be explained by the inherent imbalance in the EPL soccer games dataset on which the home team wins almost 50% of all matches. This means that the Home-win class dominates the other two classes, having 100% more examples on average. This allows cost-sensitive classification to provide an informed class rebalance.

Our conclusion is that the cost-sensitive approach allows greater errors in forecasts on the premise that riskier matches yield higher profits. However, our main goal is to achieve better models in terms of profit related to the risk taken. Several studies such as (Zhang, Tan, & Ren, 2016) have demonstrated that the selection of the right cost values of the cost matrix for real-case problems usually is a complex procedure. In our case, the optimal cost matrix selection is a quite difficult task because the objective is indirect. Basically, we search for a cost matrix that could improve the final Sharpe ratio of the portfolio if the predictions are utilized for betting. The cost assignment plays an important role since the Sharpe ratio metric depends on the total risk and takes into account the variance of betting portfolio returns. Especially if the model focuses only on the high payout odds that usually appear in the Away-win outcome, probably this tactic will increase the total risk and provide a lower Sharpe ratio, since high odds represent the outsider team with low chances to win. On the other hand, the low odds outcomes have an increased probability of winning but we have to consider that they give a small profit margin relative to the risk of loss.

Since our proposed cost matrix scheme is oriented to profit, one important consideration when looking for potential improvement would be to consider in our search space various strategies for cost matrix construction that could compensate for the predictions in terms of betting odds and risk reduction. In this potential improvement, the main goal is to emphasize those bets that may not increase the variance of the betting portfolio. One way to reduce the variance of the portfolio is to avoid high drawdowns by restricting some bets. Additionally, other algorithms such as the MetaCost (Domingos, 1999) can be tested for potential enhancements. Another potential improvement would be to apply other money management methods in betting portfolios. Also, the prediction probabilities from the models could be utilized to place opposite bets or other bet types.

Despite the room for improvement, we feel that the utilization of rating systems captures a different perspective in soccer forecasting and sports betting.

7 - The Applications of Rating Systems in Real-world Problems

7.1 Introduction

This chapter presents three applications of rating systems in real-world problems. With these applications, we intend to show the use of rating systems in other fields rather than sports. The first application deals with the ranking of Internet domain names. For this purpose, Colley, Massey, and GeM methods are used. The second application deals with the selection of trading strategies in financial markets. In this application, we use a genetic algorithm to produce different strategies according to the investors' preferences where the PointRATE is used as a fitness function. The third application deals with the ranking of movies when the ratings from users are available. For the purpose of comparison, the rating system of the Internet Movie Database (IMDb) is selected as a baseline method. Also, in this application, we have tested the involvement of rating systems in the recommendation process of movies.

The motivation for selecting these applications is to demonstrate the capabilities of rating systems in three different dimensions for real-world cases. The first dimension is oriented to apply the rating/ranking systems in other fields than sports, where we have to rank multiple items from best to worst. The second dimension is the utilization of a rating system as a part of the optimization procedure of a problem. In particular, the rating system is utilized as a fitness function that defines the objective of a genetic algorithm. The third dimension examines the possibility of building ratings and recommendations based on user preference ratings combined with other popular methods and techniques. Figure 7-1 illustrates the structure schematically.

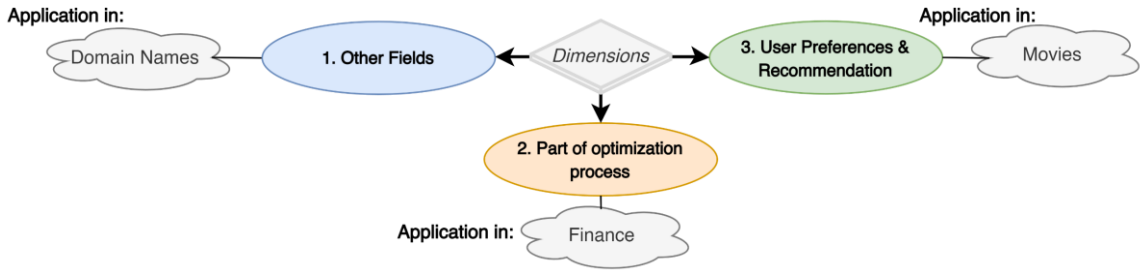


Figure 7-1: Dimensions and Applications

The remainder of this chapter is divided into four sections, with the first three sections covering one application each and the final section providing conclusions.

7.2 Domain Name Market

7.2.1 Introduction

In this application, we present the utilization of the rating methods in Internet domain name rankings. Particularly, it is based on our paper (Talattinis, Zervopoulou, & Stephanides, 2014) with updated data in experimental results and some modifications. This kind of ranking is considered important because it is associated with the formation of the price at which domain names can be sold. Specifically, these two figures are proportional amounts, i.e., the higher the rank of a domain name, the higher its selling price will be. The main contribution of this application lies in the fact that rating and ranking can assist people involved in the domain name market in terms of buying or selling domain names.

It is a fact that the growth of the Internet has resulted in the appearance of multiple sources of profitability. Thus, the concept of domain name emerged, which mainly refers to the names of websites and their extensions. It has also been proved that the ownership of domain names can be particularly lucrative for their owners. Specifically, each domain name can create value for its owner, through revenues from an active website or even without its existence (Jindra, 2005). Nowadays, due to the rapid development of e-commerce on a global level, the domain name market has already grown into a robust and profitable industry, where millions of customers search time after time for high-quality domain names in order to promote their businesses. Currently, the actual value of a domain name is difficult to be accurately determined. However, there are several objective factors involved in determining the final selling price. The ownership of a domain name grants its owner two types of rights: (1) Managerial flexibility, and (2) Legal protection of trademarks. Therefore, value can be created from a domain name in two ways: either by the expected profits or by options for action, such as the creation of an active website.

Creating an active site is not so easy, because the development of its content requires hard work, thoroughness, and imagination, contrary to domain name fortification, which is achieved by a few "clicks" at the website of the pertinent regulatory authority. Acquiring a domain name has always been speculative. Aspiring investors taking advantage of new profit opportunities offered by the Internet, register a domain name and place a simple graphic like "page under construction". Then, they only

have to wait for someone who has an exploitation plan for the domain name, but has however not acquired the appropriate website.

Another profitable and efficient strategy is to use a synonym for a domain name in conjunction with an intensive advertising campaign. The phenomenon of "cybersquatting" has become apparent. Particularly, it concerns the creation of a website with a name, closely related to the name of an already popular website in order to exploit its reputation.

Trading of domain names is usually made via an auction. Investors often need to know which domain trading is the most profitable. A domain name can be considered an investment similar to the real estate market (Tajirian, 2010). However, it is not clear how to estimate domain names value, because this market is relatively new. Consequently, some domain name sellers set selling prices arbitrarily without taking into account the actual value of the domain name. Domain name ranking can help investors to choose which domain to negotiate. Ranking refers only to the domain name and not to the active website.

After this short introduction to the subject, follows subsection 7.2.2, where we discuss some factors that determine domain names' rating and ranking. From these factors, we chose the most frequently used by the majority of people involved in the domain name market. Subsection 7.2.3 provides an illustrative example of the application of methods to rate and rank certain domain names. Then, subsection 7.2.4 and 7.2.5 presents the application aim and the data used respectively. Next, subsection 7.2.6 illustrates the experimental results generated by these methods and a comparison of them. Indicatively, the top 10 domain names are presented as a partial list of the full aggregated list from all the domains tested with these methods. Finally, in the last subsection, conclusions are drawn.

7.2.2 Determinant Factors

In order to rank a group of domain names, we must first clarify which are the factors that affect their importance, their value, and consequently, their rank. It is worth mentioning that there is a small amount of literature referring to the selection criteria of these factors and no other direct approaches for domain name ranking have yet been proposed.

Though there are many factors that determine domain names' rankings, we indicatively mention these that are usually used by the majority of people (domain traders) involved in the domain name market. Also, some of these factors have been discussed by (Tajirian, 2005; Tajirian, 2010; Jindra, 2005). Below we present 5 factors that can be easily computed:

(1) Keyword popularity: The number of search results on Google for a keyword is a good indicator of how efficient is the keyword.

(2) Search volume of the keyword: The comparison of keyword popularity over a specific period of time. Google Trends is one of the most popular and free tools used to accomplish this task. In Google Trends, up to five keywords can be queried simultaneously.

(3) Traffic: Refers to a website's number of visitors and the amount of data exchanged to and from it. There are many ways to measure traffic such as the number of unique visitors, number of page views, duration of visits, etc. It is an important metric for evaluating the popularity of a website and also affects the ranking of the domain name.

(4) Domain name extension: The extension of a domain name, in other words, the top-level domain name can affect the value and the rank of the domain name. The most dominant extension is .com. Below .com, follow .net, .org, and domestic extensions.

(5) The size of the domain name word: Names with many characters are usually hard to memorize so those with the least possible characters are more preferred.

Some other factors that also affect domain name rank but are difficult enough to be expressed quantitatively are industry popularity and brandability. Industry popularity relates to the market volume to which a specific domain name can be applied, while brandability refers to the case that someone comes up with such an interesting new word that can become a trademark (Trent, 2008).

For this first approach to the subject, we believe that keyword popularity and search volume of the keyword will have strong importance in the ranking process. Motivated by several studies that have employed Google Trends in their research, we decided to adopt Google Trends from the determinant factors. These studies demonstrate the potential of using search volume data to forecast economic indicators. A study by Choi and Varian (Choi & Varian, 2009) utilized Google Trends to predict the

unemployment rate. In their later study (Choi & Varian, 2012) showed how Google Trends can be used to predict the values of economic indicators. Also, Preis et al. (Preis, Moat, & Stanley, 2013) have employed Google Trends for financial search terms and they found patterns related to stock market movements.

Google Trends provides relative numbers. In fact, it analyzes a portion of searches done in Google in order to compute how many of them have been done for the terms entered, compared to the total number of searches done on Google over time. Google does not reveal absolute numbers for competitive reasons, but also because those numbers would not be exact. The fact that Google Trends provides relative numbers implies that there may have been more searches for term A than for term B, but these searches may be fewer than those for term C. For example, assuming that term A is Gauss and term B is Markov, the winner is Gauss with 54 Google trends points average against Markov's 27 points average. However, if term C is Shannon, Gauss becomes the underdog with 9 points average, while Shannon is given 54. The high degree of similarity between Google Trends and points in a game is another reason that led us to employ Google Trends as a determinant factor for the ranking methods applied.

7.2.3 Illustrative Example

The systems utilized to rank domain names are Colley, Massey, and GeM. In this example, there are five domain names that have been sold in early 2014, which are jean.com, desirous.com, authorization.com, true.com, and finally, peaked.com. We will attempt to rank these domains based on the average search query volume by Google Trends (GT) during the period 2013.

The question is how can the search volume average be related to the points that a team succeeded against another? There are many ways to define the notion of a game for domain names. For example, considering the popularity numbers given by Google Trends when two domain names i and j are the input keywords, then we can say that domain i beats domain j if $d_i > d_j$, where d_i and d_j are the Google Trends measures for these domains. Therefore, $d_i - d_j$ represents the difference in trends' value between domains i and j.

Table 7-1 shows Google Trends (GT) data for the five domains of our example. Then follows the rating process steps by each method.

Table 7-1: Google Trends of domain names example

Domain *i*	**Domain *j***	**GT *i, j***	**Domain *i***	**Domain *j***	**GT *i, j***
jean.com	desirous.com	88, 0	desirous.com	true.com	0, 73
jean.com	authorization.com	88, 4	desirous.com	peaked.com	20, 80
jean.com	true.com	76, 73	authorization.com	true.com	3, 73
jean.com	peaked.com	88, 0	authorization.com	peaked.com	93, 5
desirous.com	authorization.com	1, 93	true.com	peaked.com	73, 0

GT: Google Trends

❖ Colley

Using the Colley rating method for domain names, we compute the coefficient matrix *C* and the right-hand side vector *b* after applying equations (3.3) and (3.4) respectively. The final rating result can be obtained after applying (3.5).

$$C = \begin{array}{c} \\ \text{authorization} \\ \text{desirous} \\ \text{jean} \\ \text{peaked} \\ \text{true} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 6 & -1 & -1 & -1 & -1 \\ -1 & 6 & -1 & -1 & -1 \\ -1 & -1 & 6 & -1 & -1 \\ -1 & -1 & -1 & 6 & -1 \\ -1 & -1 & -1 & -1 & 6 \end{pmatrix} \end{array} \qquad b^{\mathrm{T}} = \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 1 & -1 & 3 & 0 & 2 \end{pmatrix} \end{array}$$

❖ Massey

Adjusting the Massey rating method for domain names, we start with the same idealized function (3.6). Then Massey matrix *M* and point differential vector *d* are computed. Next After replacing the last row of *M* with *e* (vectors of 1's) and the last element of *d* with 0 we name them as $\bar{M}$ and $\bar{d}$ respectively and they are shown below:

$$\bar{M} = \begin{array}{c} \\ \text{authorization} \\ \text{desirous} \\ \text{jean} \\ \text{peaked} \\ \text{true} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \qquad \bar{d}^{\mathrm{T}} = \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 26 & -313 & 263 & -189 & 0 \end{pmatrix} \end{array}$$

Then, the Massey domain ranking method proceeds as usual according to equation (3.10).

❖ GeM

We continue with the GeM rating method where we utilized two determinant factors. The formation of the voting matrix for Google Trends data is based on the third voting scheme (Winners and Losers vote with points) mentioned in subsection 3.3.7. The logic is that both domain names should be allowed to cast votes equal to the number of points given up in the hypothetical game. Below, the voting and stochastic matrices are shown:

$$V_{GT} = \begin{array}{c} \\ \text{authorization} \\ \text{desirous} \\ \text{jean} \\ \text{peaked} \\ \text{true} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 0 & 1 & 88 & 5 & 73 \\ 93 & 0 & 88 & 80 & 73 \\ 4 & 0 & 0 & 0 & 73 \\ 93 & 20 & 88 & 0 & 73 \\ 3 & 0 & 76 & 0 & 0 \end{pmatrix} \end{array} \quad S_{GT} = \begin{array}{c} \\ \text{authorization} \\ \text{desirous} \\ \text{jean} \\ \text{peaked} \\ \text{true} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 0 & 0.01 & 0.53 & 0.03 & 0.44 \\ 0.28 & 0 & 0.26 & 0.24 & 0.22 \\ 0.05 & 0 & 0 & 0 & 0.95 \\ 0.34 & 0.07 & 0.32 & 0 & 0.27 \\ 0.04 & 0 & 0.96 & 0 & 0 \end{pmatrix} \end{array}$$

In the next step, the second determinant factor is involved. For this purpose, the Google Results of the keyword of each domain name are depicted in Table 7-2. The numbers are represented in thousands (K).

Table 7-2: Keyword popularity of domain names example

	authorization	**desirous**	**jean**	**peaked**	**true**
Google Results (GR)	72,700K	1,660K	358,000K	5,780K	676,000K

To form the voting matrix, we consider Google Results as game scores, namely, a winning domain name gets as many votes by a weaker opponent as the margin of victory in the hypothetical game between them. This is the second voting scheme (Losers Vote with point differential) mentioned in subsection 3.3.7. Below the voting and stochastic matrices are presented:

$$V_{GR} = \begin{array}{c} \\ \text{authorization} \\ \text{desirous} \\ \text{jean} \\ \text{peaked} \\ \text{true} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 0 & 0 & 285300 & 0 & 603300 \\ 71040 & 0 & 356340 & 4120 & 674340 \\ 0 & 0 & 0 & 0 & 318000 \\ 66920 & 0 & 352220 & 0 & 670220 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad S_{GR} = \begin{array}{c} \\ \text{authorization} \\ \text{desirous} \\ \text{jean} \\ \text{peaked} \\ \text{true} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 0 & 0 & 0.32 & 0 & 0.68 \\ 0.06 & 0 & 0.32 & 0 & 0.61 \\ 0 & 0 & 0 & 0 & 1 \\ 0.06 & 0 & 0.32 & 0 & 0.62 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{pmatrix} \end{array}$$

To compute the G matrix based on equation (3.18), we have set equal weights (a_{GT}=a_{GR}=0.5). Finally, the $\bar{G}$ is computed after applying equation (3.19) where we have set the damping factor b to 0.85. Below are shown G and $\bar{G}$ matrices:

$$G = \begin{array}{c} \\ \text{authorization} \\ \text{desirous} \\ \text{jean} \\ \text{peaked} \\ \text{true} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 0 & 0 & 0.42 & 0.01 & 0.56 \\ 0.17 & 0 & 0.29 & 0.12 & 0.41 \\ 0.03 & 0 & 0 & 0 & 0.97 \\ 0.2 & 0.04 & 0.32 & 0 & 0.44 \\ 0.12 & 0.1 & 0.58 & 0.1 & 0.1 \end{pmatrix} \end{array} \quad \bar{G} = \begin{array}{c} \\ \text{authorization} \\ \text{desirous} \\ \text{jean} \\ \text{peaked} \\ \text{true} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{authorization} & \text{desirous} & \text{jean} & \text{peaked} & \text{true} \end{array} \\ \begin{pmatrix} 0.03 & 0.03 & 0.39 & 0.04 & 0.5 \\ 0.18 & 0.03 & 0.28 & 0.13 & 0.38 \\ 0.05 & 0.03 & 0.03 & 0.03 & 0.86 \\ 0.2 & 0.06 & 0.3 & 0.03 & 0.4 \\ 0.13 & 0.12 & 0.52 & 0.12 & 0.12 \end{pmatrix} \end{array}$$

The final rating and ranking lists for Colley, Massey, and GeM are depicted in the table below:

Table 7-3: Rating and Ranking results of domain names example

Domain	**Colley**		**Massey**		**GeM**		**Price**
	Rating	**Rank**	**Rating**	**Rank**	**Rating**	**Rank**	**(USD)**
authorization	0.5000	3	5.20	3	0.1037	3	35,100
desirous	0.2143	5	-62.60	5	0.0694	5	2,600
jean	0.7857	1	52.60	1	0.319	2	50,000
peaked	0.3571	4	-37.80	4	0.0753	4	4,000
true	0.6429	2	42.60	2	0.4327	1	350,000

As we conclude from the above table, in Massey and Colley the domain jean.com has beaten all the other four domain names and, thus, it has been ranked first. Contrary to jean.com, the domain desirous.com has been defeated by all others, therefore, it has the lowest rating of all and so, it takes the last position in the ranking.

As we can notice from the results of the GeM, due to the use of two determinant factors, rank positions between the first two domains have interchanged and this agrees with their selling prices ranking. Therefore, for this example, the GeM method can be characterized as more representative than the others.

7.2.4 Application Aim

This application examines the possibility of applying rating methods that mainly originate from the sports field, in the domain name market. Thus, this application aims to rank domain names by applying various rating methods, and then the results of ranking lists are compared in order to draw some conclusions. In addition, with the goal of achieving accurate rankings for the domain names, we employed a real dataset with actual transactions from the domain name market. Also, we have used real data for keyword popularity and search volume determinant factors.

7.2.5 Data

Our data was gathered from a variety of public Internet sources. The data does not contain private transactions that occur frequently. In any case, data gathering and parsing had to be automated. Currently, the database consists of more than 100,000 transaction prices that occurred during the period between 1999 to 2022. We have conducted thorough research and have already implemented some techniques for parallelization of collecting data, in order to keep our database updated in time. For more details about crawling, its parallelization, and parsing processes we refer the reader to (Talattinis, Sidiropoulou, Chalkias, & Stephanides, 2010).

7.2.6 Experimental Results

This section presents the empirical results generated by Colley, Massey, and GeM in the same way that applied in the illustrative example of subsection 7.2.3. The numerical computations of the ratings were done using Python. In order to make the data preparation more manageable we chose to focus on a subset of transactions. For the experiment purposes, several domain names were selected and a key factor in the selection was the selling price which had to be the top above 2,000,000 USD. Also, we have chosen transactions of domain names from all years.

We begin by demonstrating the comparison of the different methods using Kendall's tau correlation coefficient. The lower diagonal elements represent Kendall's tau values of each pair, while the upper diagonal elements are the p-values of each pair from the two-sided hypothesis test, whose null hypothesis is an absence of association.

Table 7-4: Kendall's tau comparison for domain name ranking lists

	Colley	**Massey**	**GeM**
Colley	1.000	1.26E-21	2.61E-19
Massey	0.933	1.000	1.28E-20
GeM	0.877	0.909	1.000

From Table 7-4, when we compared the ranking lists according to the Kendall-Tau correlation method, we conclude that Massey and Colley are in close agreement, while GeM shows a slight variation due to the use of two determinant factors. All p-values are less than 0.001 therefore we reject the null hypothesis (*tau*=0). This indicates that the ranking lists across all methods share many similarities.

The top 10 domain names as a partial list are presented in Table 7-5 after their aggregation by the Perron method discussed in subsection 3.4.2. Table 7-5 has been constructed in the following format: the first column is the domain name itself, the second and third column represents the rating and ranking of each domain name respectively, and the fourth column is the price at which the domain name was sold and finally, the last column consists of the date on which each domain was sold. Though we refer to the selling price, it cannot be a reliable measure of comparison, due to its dependence on the time that happened.

Table 7-5: Top 10 domain names

Domain Name	Aggregated (Perron)		Selling Price (USD)	Selling Date
	Rating	Rank		
it.com	0.5875	1	3,800,000	6/19/2022
we.com	0.1963	2	8,000,000	6/19/2015
mi.com	0.0818	3	3,600,000	1/4/2014
z.com	0.0485	4	6,784,000	1/11/2014
shop.com	0.0354	5	3,500,000	1/11/2003
pizza.com	0.0151	6	2,605,000	1/4/2008
social.com	0.0151	7	2,600,000	1/7/2011
fb.com	0.0039	8	8,500,000	1/9/2010
express.com	0.0034	9	2,000,000	1/3/2000
Connect.com	0.0032	10	10,000,000	4/15/2022

As we can observe from the above table, the aggregated list of the top 10 includes very popular domain names and for this reason, their ranking can be justified as a positive indication that the rating systems can be applied in other fields outside of sports. However, the ranking list of domain names should not be compared with their selling prices because the amount sold reflects past factors, while our ranking is based on current factors and also without including economic factors such as future earnings/cash flows, etc.

7.2.7 Conclusions

In this section, we show how we can rank domain names with three different methods that are mainly used in the sports field. For generating our empirical results, in

Massey and Colley methods we used Google Trends as a determinant factor, while in the GeM method, we used Google Trends and Google search results. The determinant factors are simple and were chosen with consideration of what people in the domain name market searching for and focusing on. From Kendall's tau correlation results, we conclude that raking lists derived from Massey and Colley methods have a strong correlation. The correlation with GeM ranking results has a lower *tau* due to the use of more than one determinant factor. In our empirical results, we cannot use the selling price as a criterion to check if ranking lists generated via different ranking methods match. This is due to the fact that selling prices were formed based on past factors or data, while our ranking is based on current factors or data.

While the present application has contributed new insights and understanding in the field, it is important to acknowledge the limitations that may have influenced the results. One limitation is the small number of determinant factors used may not perfectly capture the ranking position of domain names. Another limitation is the difficulty of having a large number of search volume comparison pairs. Last, as we explained previously, rankings do not reflect selling price history and this makes it difficult to draw conclusions about rankings.

In order to further improve the present application, several potential directions could be explored. One possible improvement is to extend the present application to work with more determinant factors. The inclusion of additional data sources may also enhance the ranking results. One more option for improvement would be to apply well-known techniques from the real estate market field since as previously noted there are similarities between these markets. Another potential improvement would be to extend to other rating systems that have their roots in decision theory. This could involve methods such as the MAUT/MAVT we presented in Chapter 4 where we can utilize multiple attributes and we can take into account the preference of the user.

In conclusion, rating methods presented in this section may be used by many groups of people, such as domain traders, portfolio managers, and investors. Concerning to decision-making process, i.e., if someone decides to buy a domain name according to its rank, the methods presented in this section can be a utility tool, but not the only one. Also, our approach to utilizing those methods can be adapted to new alternative markets such as the NFT (Non-fungible Token) market or the virtual properties market in Metaverse, as long as data and relevant factors are available for analysis.

7.3 Financial Management and Optimization

7.3.1 Introduction

The goal of this section is three-fold. First, we aim to show the use of rating methods in financial management with illustrative examples. The second aim is to deploy a real-world application in financial management where we examine if a rating system can be utilized as part of an optimization process. In particular, the application is oriented to test the optimization and selection of trading strategies in financial markets where investors have different profiles and preferences. Third, we aim to utilize the application as a part of the validation and as an extension of sensitivity analysis for the PointRATE method presented in section 4.3.

In this section, we present the related work, then we introduce two hypothetical examples from the domain of investment and finance in order to demonstrate the applicability of rating methods in those fields. Next, we explain the application and experimental part and after presenting the experimental results, finally we draw some conclusions.

7.3.2 Related Work

The purpose of this section is to outline the previous work that addresses similar topics to the present application. The studies were selected for their relevance to the following areas of interest: investment selection, portfolio rankings, stock rankings, trading strategies evaluation and optimization, and genetic algorithms in financial applications.

The first study we will consider was conducted by Martel et al. (Martel, Khoury, & Bergeron, 1988) who made portfolio comparisons, by the use of Electre methods. They proposed an alternative approach that emphasizes a multicriteria analysis, based on the principle that risk has a multidimensional nature. This approach also takes into account the dimensions related to returns and other factors that influence the decision-making process. In addition, in their study, they chose to test 10 combinations of weights to reflect the decision-maker profile.

A noteworthy study by Berutich et al. (Berutich, López, Luna, & Quintana, 2016) used a genetic algorithm to discover profitable strategies based on technical rules. The authors by using the random sampling method, solutions show improved performance when tested on unseen data, and overfitting is reduced. Furthermore, this study explores

the use of a combination of both traditional technical indicators and novel financial metrics including those calculated over different periods.

Mendonça et al. (Mendonça, Ferreira, Cardoso, & Martins, 2020) in their paper propose a new multi-objective financial portfolio optimization model, which is considered an improvement over the model proposed by Ferreira et al. (Ferreira, Hanaoka, Paiva, & Cardoso, 2018). Specifically, the authors suggest an evolutionary algorithm and two new decision-making methods that are guided by investors' preferences. They conduct computational simulations that consider monthly maximum drawdown and cumulative return using assets from the Brazilian stock exchange.

In another study by Rajabioun et al. (Rajabioun & Rahimi-Kian, 2008), a precise predictive model is applied to four companies in the Boston stock market. The authors use genetic programming to produce the mathematical model, which was the most successful among the other methods (artificial neural networks and neuro-fuzzy networks) in their simulations to predict future trends.

Lee and Sabbaghi (Lee & Sabbaghi, 2020) in their research focus on using genetic algorithms and multi-objective optimization to create algorithmic trading rules for foreign exchange markets. The authors compare two approaches: multi-objective optimization and spontaneous optimization of design variables. They present an algorithm for identifying the best indicator and operator values for algorithmic trading, using a multi-objective optimization approach that considers trade-offs. They also examine the use of multi-objective optimization in trading and investing, presenting a framework for using this approach with an evolutionary algorithm. This approach differs from typical trading methods and focuses on finding trade-off relationships among multiple objective functions.

Dacorogna et al. (Dacorogna, Gençay, Müller, & Pictet, 2001) introduce two performance measures that consider an investor's risk aversion. The maximization of these measures is equivalent to maximizing the expected utility of an investment for a risk-averse investor. The empirical results of this study are compared with traditional performance measures, such as the Sharpe ratio and maximum drawdown. The introduced measures demonstrate robustness against the clustering of losses and provide a comprehensive characterization of the dynamic behavior of investment strategies.

Gadallah et al. (Gadallah, Fors, & Moneim, 2015) propose a combination of multiple decision-making models to address the challenges investors face during the

investment decision process. The introduced model is formulated as a multi-criteria optimization problem, with the objectives of maximizing profit and minimizing the maximum drawdown. The proposed strategy, which combines various trading decision models, appears to be effective.

Brandouy et al. (Brandouy, Mathieu, & Veryzhenko, 2013) introduce an agent-based model for risk-adjusted performance evaluation and compare the performance of various risk-aware investment strategies. They demonstrate that only investors with a moderate level of risk aversion are able to sustain profitability in the long term, as opposed to those who are either risk-seeking or absolutely risk-averse.

Quah (Quah, 2008) proposes a systematic approach for selecting equities based on the Receiver Operating Characteristic (ROC) curve and soft-computing models that incorporate fundamental analysis. They compare the performance of three different soft-computing models and study their computational time complexity, using several metrics to evaluate their performance. The results show that higher predicted values correspond to higher probabilities of positive appreciation.

Sevastjanov and Dymova (Sevastjanov & Dymova, 2009) propose a new method for stock ranking that is based on multiple criteria decision making and optimization. This approach enables stock selection by using two general criteria, one based on financial indicators and the other based on stock prices. In addition, the authors also compare the two different approaches to stock selection.

7.3.3 Example 1: Investment Selection

Our first hypothetical example deals with investment selection. Assume that we have to select among three available investments. In this simple hypothetical example, we take into account only two attributes the Return on Investment (simple ROI) and the Payback Period (PP). We referred to ROI in section 2.4. Although ROI has a range between $[-1,+\infty)$, in this example, we are interested only in investments with nonnegative ROI and we suppose that the max value of ROI is 1 (100%). The main disadvantage of simple ROI is that it does not use information about the holding period of an investment. Therefore, the Payback period metric as an attribute will be useful as it measures the time taken to recover the initial investment. We consider that for this example the range of PP is between 1 to 4 years. The shortest PP is considered more acceptable with lower risk. Table 7-6 represents each of the three investments' attribute values.

Table 7-6: Performance of investments

Attribute / Investment (I)	Investment 1 (I1)	Investment 2 (I2)	Investment 3 (I3)
ROI	0.8	0.4	0.1
PP	4.0	1.5	1.0

In this example, we rate and rank investments in three ways: (1) PointRATE, (2) rank aggregation methods, and (3) rating aggregation methods. It is worth noting that in aggregation methods the scoring of each attribute represents the rating value.

❖ PointRATE:

- Step 1: Our alternatives/items are O = {Investment$_1$, Investment$_2$, Investment$_3$} and m=3. The attributes utilized are A = {*ROI*, *PP*}, n=2 with domains D_{ROI} = [0,1], D_{PP} = [1,4] ⇒ D = {[0, 1], [1, 4]}.
- Step 2:

  - ROI attribute (Type-1):

    A. Splits of interest are k=2, b_{ROI} = {0, 1}.
    The resulting set of classes is C_{ROI} = {[0,1]}.
    B. Reward points are p_{ROI} = {100}.
    C. Cumulative reward points are s_{ROI} = {[0,100]}.
    D. In our example we have selected the linear reward scheme for C_{ROI}:

$$v_{ROI}(x_{ROI}) = v_{ROI,1}(x_{ROI}), \qquad if\ x_{ROI} \in C_{ROI,1}$$

    We have only to estimate $v_{ROI,1}$ based on the linear reward scheme and the relation $\{(b_{ROI,1}, S_{ROI,1}), (b_{ROI,2}, S_{ROI,2})\} = \{(0,0), (1,100)\}$.
    Therefore, the v_{ROI} is written

$$v_{ROI}(x_{ROI}) = 100 \cdot x_{ROI}, \qquad if\ x_{ROI} \in [0,100],$$

    and its graph is plotted in Figure 7-2.

  - PP attribute (Type-2):

    A. Points of interest are k=3, b_{PP} = {1,2,4}.
    The resulting ranges are C_{PP} = {[1,2),[2,4]}.
    B. Reward points are p_{PP} = {100,-80,-20}.
    C. Cumulative reward points are s_{PP} = {100,20,0}
    D. We have selected the linear reward scheme for $C_{PP,1}$= [1,2) and $C_{PP,2}$ = [2,4] to distribute points.
    The piecewise function v_{pp} can be written as follows:

$$v_{PP}(x_{PP}) = \begin{cases} v_{PP,1}(x_{PP}), & if\ x_{PP} \in C_{PP,1} \\ v_{PP,2}(x_{PP}), & if\ x_{PP} \in C_{PP,2} \end{cases}$$

Firstly, we estimate $v_{ROI,1}$ based on the linear reward scheme and the relation $\{(b_{PP,1}, S_{PP,1}), (b_{PP,2}, S_{PP,2})\} = \{(1,100), (2,20)\}$.

Secondly, we estimate $v_{ROI,2}$ based on the linear reward scheme and the relation $\{(b_{PP,2}, S_{PP,2}), (b_{PP,3}, S_{PP,3})\} = \{(2,20), (4,0)\}$.

Finally, the v_{pp} is written below and its graph is depicted in Figure 7-2.

$$v_{PP}(x_{PP}) = \begin{cases} -80 \cdot x_{PP} + 180, & if\ x_{PP} \in [1,2) \\ -10 \cdot x_{PP} + 40, & if\ x_{PP} \in [2,4] \end{cases}$$

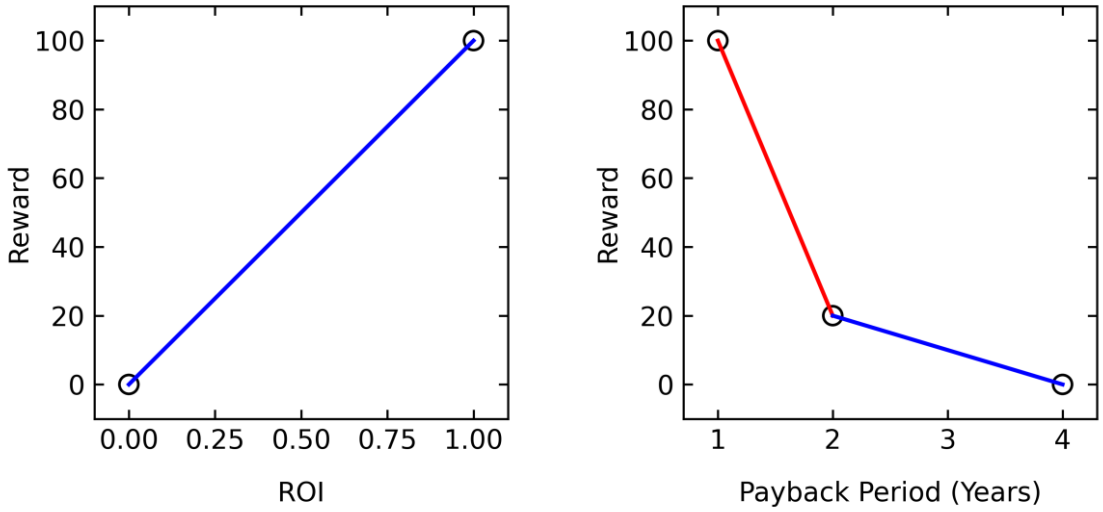


Figure 7-2: Reward functions of attributes

- Step 3: All attributes will be evaluated with the same level of importance (equal weights scheme), thus we can set h_{ROI} = 10 (very important) and h_{PP} = 10 (very important). After normalization, the final weights are w_{ROI} = 1/2, w_{TR} = 1/2.
- Step 4: Finally, we compute ratings based on (4.10).

$$r_1 = w_{ROI} \cdot v_{ROI}(x_{1,ROI}) + w_{PP} \cdot v_{PP}(x_{1,PP}) = \frac{1}{2} \cdot v_{ROI}(0.8) + \frac{1}{2} \cdot v_{PP}(4) = 40.0$$

$$r_2 = w_{ROI} \cdot v_{ROI}(x_{2,ROI}) + w_{PP} \cdot v_{PP}(x_{2,PP}) = \frac{1}{2} \cdot v_{ROI}(0.4) + \frac{1}{2} \cdot v_{PP}(1.5) = 50.0$$

$$r_3 = w_{ROI} \cdot v_{ROI}(x_{3,ROI}) + w_{PP} \cdot v_{PP}(x_{3,PP}) = \frac{1}{2} \cdot v_{ROI}(0.1) + \frac{1}{2} \cdot v_{PP}(1) = 55.0$$

The final rating and rank of each investment are represented in Table 7-7. It is clear from the final ratings that Investment 3 is preferred, as it offers the smallest payback period, regardless of its low ROI value. This happened due to the fact that our modeling was oriented to reward the investments with low PP.

❖ Rank Aggregation:

If we rank the investments based on ROI and PP then we can apply a simple rank aggregation based on the Borda Count and Average Rank presented in subsection 3.4.1. Table 7-7 presents the rank aggregation results where no conclusion can be drawn from the results since all methods rank investments equally.

❖ Rating Aggregation:

The rating aggregation methods (1) Perron, (2) ODM, and (3) Markov from subsection 3.4.2 are applied. For the Markov method, the damping factor *b* was set to 0.9. By performing the rating aggregation, a single rating list is generated where we consider that ROI and PP represent 2 different rating lists for the investments. First, we form the matrices of rating differences and then we normalize them, based on (3.20) and (3.21) equations respectively. As we can see below the formation of the PayBack period matrix is different since the numerical scoring runs in the opposite direction.

$$R_{ROI} = \begin{pmatrix} 0 & 0.4 & 0.7 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0 \end{pmatrix} \quad \bar{R}_{ROI} = \begin{pmatrix} 0 & 0.29 & 0.5 \\ 0 & 0 & 0.21 \\ 0 & 0 & 0 \end{pmatrix}$$

$$R_{PP} = \begin{pmatrix} 0 & 0 & 0 \\ 2.5 & 0 & 0 \\ 3 & 0.5 & 0 \end{pmatrix} \quad \bar{R}_{PP} = \begin{pmatrix} 0 & 0 & 0 \\ 0.42 & 0 & 0 \\ 0.5 & 0.08 & 0 \end{pmatrix}$$

Next $\bar{R}_{ave}$ is computed by equation (3.22) by applying equal weights and finally, the aggregation method can be applied. The final results are depicted in Table 7-7 and indicate that investments 1 or 2 are ranked first due to high ROI and low PP respectively.

Table 7-7: Investment rating aggregation

I	**PointRATE**		**Borda Count**		**Avg Rank**		**Aggr. Perron**		**Aggr. ODM**		**Aggr. Markov**	
	Rating	**#**	**Count**	**#**	**Rating (Avg)**	**#**	**Rating**	**#**	**Rating**	**#**	**Rating**	**#**
I1	40.00	3	2	1	1.0	1	0.3660	1	0.8571	2	0.4186	1
I2	50.00	2	2	1	1.0	1	0.3237	2	1.7097	1	0.2847	3
I3	55.00	1	2	1	1.0	1	0.3103	3	0.8167	3	0.2967	2

I: Investment; **#**: Rank

From this example, we notice the rankings from rating aggregation methods differ compared to PointRATE results. It can be concluded that PointRATE offers the advantage to model preferences and this plays a significant role in determining the final rankings.

7.3.4 Example 2: Portfolio Selection Based on Rankings

Our second hypothetical example comes from the field of portfolio rankings. Assume that there are three different investors. First, there is an independent engineer, trading during their free time and seeking some quick profit. The next one is a start-up investment firm, in need of some profits, but also cautious of losses. Finally, a well-established investment fund is concerned with the customer's financial safety. We could say that the engineer is a risk-seeking investor, the start-up is neutral, and the investment fund is a risk-averse investor. In our example, there are five portfolios available for each investor to choose from. The data for each portfolio's performance, for the past three years, are available, in order to assess their characteristics. For simplicity, we assume that the initial capital is 100 monetary units and each investor has to take into account three variables (attributes) that describe each portfolio:

- Total Return (TR) is a useful metric that evaluates the performance of an investment over a specific period by summing the returns. In this hypothetical example, we examine total returns between 0 to 0.5.
- Maximum Drawdown (MaxDD) is a widely used metric for quantifying the largest percentage loss of the portfolio (or investment), i.e., from the highest value to the lowest value, during a specific time period. It is calculated as a percentage difference between two points: the historical peak (highest value of the portfolio) and the lowest trough (lowest value of the portfolio) prior to a new peak being achieved. Maximum Drawdown can range from 0 (perfect) to -1 (worst). Here we examine values from -0.5 to 0.

Figure 7-3 shows the graph of portfolios' growth while Table 7-8 represents portfolios' performance, i.e., values of their attributes.

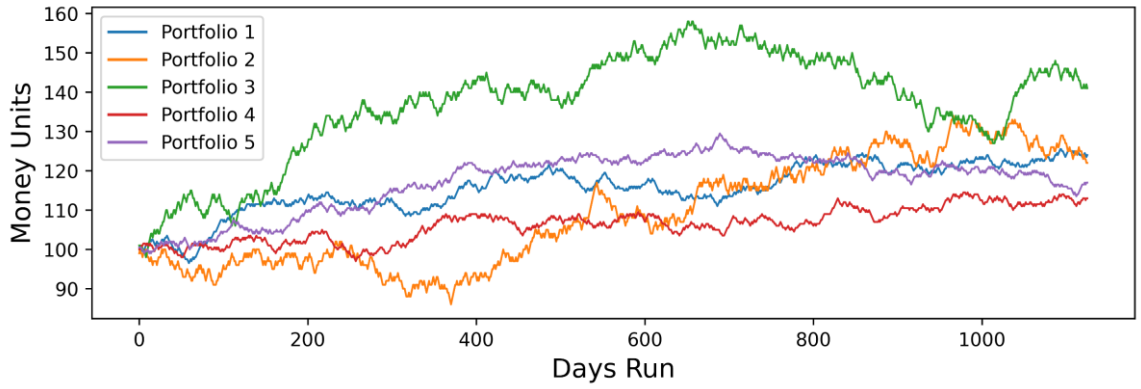


Figure 7-3: Portfolios' growth

Table 7-8: Performance of portfolios

Attribute / Portfolio (P)	P1	P2	P3	P4	P5
Total Returns (TR)	0.222	0.232	0.365	0.130	0.164
Maximum Drawdown (MaxDD)	-0.086	-0.157	-0.196	-0.076	-0.124

The rating of portfolios is done with PointRATE and rating aggregation methods.

- PointRATE:
  - Step 1: $X = \{TR, Max\ DD\}$, $D = \{[0, 0.5], [-0.5, 0]\}$.

    Step 2: Each investor defines the ranges and reward points. The overall reward function for the TR attribute is similar to the exponential function curve for risk-seeking, linear for neutral, and logarithmic for risk-averse. Their selection was made in accordance with the theoretical basis that relates to the utility functions of each investor type. In Table 7-9 the hypothetical investors' preferences for each attribute are demonstrated.

Table 7-9: Preferences for each investor

Investor	Attributes	Reward Ranges	Reward Points	Cumulative Reward Points	Reward scheme
-RISK SEEKING- ENGINNER	TR	[0, 0.25)	30	30	linear
		[0.25, 0.5]	70	100	linear
			p={0,30,70}	*s*={0,30,100}	
	Max DD	[-0.5, 0)	100	100	linear
			p={0,100}	s={0,100}	
-NEUTRAL - START–UP	TR	[0, 0.5]	100	100	linear
			p={0,100}	*s*={0,100}	
	Max DD	[-1, 0]	100	100	linear
			p={0,100}	*s*={0,100}	
-RISK AVERSE- FUND	TR	[0, 0.25)	70	70	linear
		[0.25, 0.5]	30	100	linear
			p={0,70,30}	*s*={0,70,100}	
	Max DD	[-0.5, 0)	100	100	linear
			p={0,100}	s={0,100}	

- Step 3: The logic of weighting is explained as follows. The engineer places more emphasis on the total returns, generating 90.9% of the total rating from the portfolio's profitability. The start-up is unbiased towards any of the attributes. The investment fund places more importance on the stability of the portfolios, generating only 9.1% of the total value from the portfolio's profitability. The weights are shown in Table 7-10.

Table 7-10: Attributes weights

Investor	h_{TR}	h_{MaxDD}	w_{TR}	w_{MaxDD}
Risk-Seeking - Engineer	1	10	0.0909	0.9091
Neutral - Start-up	10	10	0.5000	0.5000
Risk-Averse - Fund	10	1	0.9091	0.0909

- Step 4: For each investor, we can calculate the rating points for each portfolio, by utilizing the weights, and the reward functions. By sorting ratings, we can rank the portfolios, and finally, conclude which portfolio each investor will choose.

The final rating and ranking are listed in Table 7-11 for each investor. The results can be interpreted as follows. It is noticeable that the engineer will prefer Portfolio-3 over the other four, as it is more profitable. Additionally, Portfolio-1 is the second in rank, as it has similar profitability to Portfolio-2 but better stability. The start-up will prefer Portfolio-3 over the others, as it offers the best combination of performance, while Portfolio-1 is the next one in ranking and is a considerably stable alternative with similar returns. Finally, the investment fund will prefer Portfolio-1, while it is slightly less stable than Portfolio-4 its profitability compensates for that loss in stability. Still, their next choice is Portfolio-4, as it exhibits great stability with relatively low drawdowns.

❖ Rating Aggregation Methods:

Three rating aggregation methods presented in subsection 3.4.2 are applied for this example. Firstly, we form the matrices of rating distances for each attribute by applying equation (3.20), secondly, we normalize each matrix by applying equation (3.21), and finally, we use the same weights shown in Table 7-10 for equation (3.22).

$$R_{TR} = \begin{pmatrix} 0 & 0 & 0 & 0.09 & 0.06 \\ 0.01 & 0 & 0 & 0.1 & 0.07 \\ 0.14 & 0.13 & 0 & 0.23 & 0.2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.03 & 0 \end{pmatrix} \quad \bar{R}_{TR} = \begin{pmatrix} 0 & 0 & 0 & 0.09 & 0.05 \\ 0.01 & 0 & 0 & 0.09 & 0.06 \\ 0.13 & 0.12 & 0 & 0.22 & 0.19 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.03 & 0 \end{pmatrix}$$

$$R_{mDD} = \begin{pmatrix} 0 & 0.07 & 0.11 & 0 & 0.04 \\ 0 & 0 & 0.04 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.01 & 0.08 & 0.12 & 0 & 0.05 \\ 0 & 0.03 & 0.07 & 0 & 0 \end{pmatrix} \quad \bar{R}_{mDD} = \begin{pmatrix} 0 & 0.11 & 0.18 & 0 & 0.06 \\ 0 & 0 & 0.06 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.02 & 0.13 & 0.19 & 0 & 0.08 \\ 0 & 0.05 & 0.12 & 0 & 0 \end{pmatrix}$$

The rating aggregation methods applied are (1) Perron, (2) ODM, and (3) Markov. For the Markov method, the damping factor *b* was set to 0.9. The final results are depicted in Table 7-11.

Table 7-11: Portfolios rating and ranking results

Investor	***P***	**PointRATE**		**Aggr. Perron**		**Aggr. ODM**		**Aggr. Markov**	
		Rating	**Rank**	**Rating**	**Rank**	**Rating**	**Rank**	**Rating**	**Rank**
-RISK SEEKING- ENGINEER	P1	31.75	2	0.1695	2	1.2143	2	0.1943	2
	P2	31.55	3	0.1336	3	1.1318	3	0.1253	4
	P3	62.07	1	0.5317	1	12.0699	1	0.4060	1
	P4	21.89	5	0.0936	4	0.0968	5	0.1626	3
	P5	24.73	4	0.0717	5	0.1526	4	0.1118	5
-NEUTRAL- START–UP	P1	63.60	2	0.2470	2	3.1054	1	0.2137	2
	P2	57.50	3	0.1226	4	0.5462	4	0.1218	4
	P3	66.90	1	0.2950	1	1.2070	2	0.3491	1
	P4	55.40	4	0.2161	3	0.9677	3	0.2018	3
	P5	54.00	5	0.1193	5	0.4532	5	0.1135	5
-RISK AVERSE- FUND	P1	80.92	1	0.3077	2	12.0816	1	0.2239	3
	P2	68.27	4	0.0853	5	0.2564	4	0.1151	4
	P3	62.89	5	0.1135	4	0.1207	5	0.2584	2
	P4	80.40	2	0.3559	1	9.6770	2	0.2962	1
	P5	72.54	3	0.1376	3	1.0197	3	0.1064	5

P: Portfolio

From Table 7-11 variations are observed in the rankings generated by each method. For example, according to Perron and Markov, the start-up will opt for Portfolio-3, whereas ODM will lead them to select Portfolio-1. In addition, the Fund will make its decision on Portfolio-4 based on the result of Perron or Markov, and Portfolio-1 based on ODM. Also, an important note is that the Markov method ranks Portfolio-3 in 2nd place for the Fund, while the other methods placed it either 4th or 5th. The engineer

prefers high returns and is willing to take more risk by selecting Portfolio-3 (in all methods). Contrary, the Investment Fund is biased toward the stability of the portfolios and prefers lower-risk portfolios that are not in danger of losing clients' money. Moreover, PointRATE and ODM generated the same ranking lists for the engineer and the start-up. Finally, by comparing all methods we can conclude that Portfolio-1 is a good all-around choice, which some individuals may not choose at first, because of personal bias, but is a viable alternative to them in case their preferred portfolio is unavailable for some reason.

In summary, the ranking lists generated by four different methods share many similarities. However, for a more thorough examination and in order to meet the user's preferences MAUT/MAVT methodology is more capable.

7.3.5 Application Aim

This experimental application aims to examine the ability of a rating method to guide correctly the optimization process of a trading strategy, by utilizing a genetic algorithm. The trading strategies are optimized according to the investment profile of the user. The performance of strategies is evaluated by applying a rating method as a fitness function to rate them. The PointRATE was selected as a fitness function because is capable of incorporating multiple attributes. Also, unlike rank and rating aggregation methods we presented in previous examples, it does not require all alternatives to be paired together to make comparisons. In addition, it has a specific rating scale from 0 to 100 which makes it capable of being used as a fitness function. Last, it can be easy to apply depending on the specific problem, and as previously discussed in the related work section, analogous methods have been applied to similar topics.

The development of the present application is based on the studies discussed in the related work (subsection 7.3.2). The logic of trading strategy that is intended to be optimized is very simple and has its basis in Technical Analysis. Using two different Simple Moving Average (SMA) indices the strategy assumes positions based on the corresponding signal. A short position is assumed on downward crossovers, while a long position is assumed on upward crossovers. The maximum number of simultaneously opened positions is set to one. In the present application, the concept of genetic algorithm is used to find the SMA periods that optimize the user's utility. The pseudocode of the SMA strategy is given below and is triggered by every new price bar.

- Pseudocode of the SMA strategy:

```
Input: prices_history, periods_short, periods_long, opened_position_num
Output: position
position = None
sma_short = sma(prices_history,periods_short)
sma_long = sma(prices_history,periods_long)
if opened_position_num<=1:
        if crossover(sma_short,sma_long):
                position = open_long_position()
                close_any_short_position()
        else if crossover(sma_long, sma_short):
                position = open_short_position()
                close_any_long_position()
```

7.3.6 Data

Historical prices of 10 stock market companies selected from various sectors were used. The company's name, the symbol name, and the sector that the company operates are shown in the following table.

Table 7-12: List of stocks

Company Name	Symbol	Sector
Microsoft Corporation	MSFT	Technology
Apple Inc.	AAPL	Technology
ASML Holding N.V.	ASML	Technology
Visa Inc.	V	Financial Services
Mastercard Incorporated	MA	Financial Services
Alibaba Group Holding Limited	BABA	Consumer Cyclical
Starbucks Corporation	SBUX	Consumer Cyclical
UnitedHealth Group Incorporated	UNH	Healthcare
Thermo Fisher Scientific Inc.	TMO	Healthcare
CVS Health Corporation	CVS	Healthcare

The data consist of daily prices from January 1, 2014, to December 31, 2019. For the purpose of the experiment, we tested several financial instruments and their final selection was made after considering the applicability, the total number of trades placed, and the positive profitability of the SMA crossover strategy. The time series data were

obtained from Yahoo Finance (Yahoo-Finance, 2023) and they comprised daily price bars including Open, High, Low, and Close (OHLC) prices.

7.3.7 Investor Profiles and Metrics

We assume that we have two basic types of investors, risk-averse and risk-seeking. The risk-averse investor (R-A) is concerned with minimizing losses and avoiding risk even if the investment has lower potential returns. On the other hand, the risk-seeking (R-S) investor is willing to take on a higher level of risk.

Each trading strategy can be characterized by several metrics. The criteria that are commonly used are associated with return-based and risk-based metrics. Although there are more ways to characterize a strategy, in this application we utilize two attributes for the return-based and one for the risk-based metric. It is important to note that in order to focus on specific attributes we avoided using metrics that combine multiple measures such as risk-adjusted metrics. The Sharpe ratio used in the application of Chapter 6 belongs to risk-adjusted metrics and as concluded by (Berutich, López, Luna, & Quintana, 2016) is not an ideal fitness function for a genetic algorithm. In this application, we have isolated the mean returns from the Sharpe ratio and the Maximum Drawn Down (MaxDD) from the Calmar ratio (Young, 1991). Notably, we focused on some of the popular metrics outlined in (Pardo, 2008). The return-based metrics used are the ROI (presented in the investment selection example) and the average returns, while for the risk-based metrics, we have considered the Maximum Drawdown (presented in the portfolio rankings example). The Mean Return is the Total Returns (presented in the portfolio rankings example) divided by the number of returns.

The modeling of profiles is oriented to make the method suitable as a fitness function that measures how well the trading strategy is achieving its objective. To maintain simplicity and objectivity in the modeling process, we have chosen the linear function for MaxDD for both types of investors. Especially for the R-S type, we have chosen the exponential function for the “ROI” and “Mean Return” attributes while for the R-A type, we have used the logarithmic function. The selection of those function schemes was made in accordance with the theoretical basis that relates the utility functions for risk-averse and risk-seeking types.

For the experiment purposes, we examine values of ROI from 0 to 0.5, Mean Return from 0 to 0.1, and MaxDD from -0.5 to 0. The selection of those values was made

after considering that investors behave rationally and are not interested in strategies with negative ROI or negative Mean Returns. Also, the upper limits are determined by experimenting with the SMA crossover strategy in each attribute separately. Additionally, a significant percentage of our capital is used when opening positions (the details of backtesting are presented in the next subsection), and the performance of strategy with values of MaxDD lower than -0.5 is considered unacceptable. This signifies that the solution from GA is rejected in the cases where the strategy evaluation produces negative values in ROI, Mean Return, and MaxDD lower than -0.5.

Table 7-13: Investor types

Investor Profile	Attributes	Reward Ranges	Reward Points	Cumulative Reward Points	Reward Scheme
-RISK AVERSE-	ROI	[0, 0.5)	100 p={0,100}	100 s={0,100}	logarithmic
	Mean Return	[0, 0.1]	100 p={100, -100}	100 s={100, 0}	logarithmic
	Max DD	[-0.5, 0)	100 p={0,100}	100 s={0,100}	linear
-RISK SEEKING-	ROI	[0, 0.5)	100 p={0,100}	100 s={0,100}	exponential
	Mean Return	[0, 0.1]	100 p={100, -100}	100 s={100, 0}	exponential
	Max DD	[-0.5, 0)	100 p={0,100}	100 s={0,100}	linear

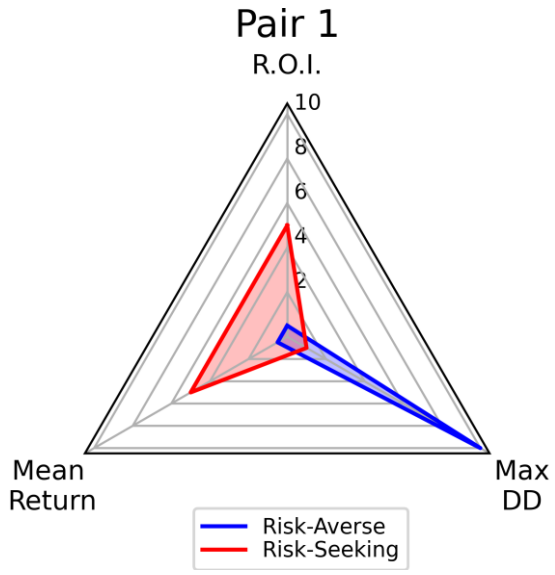
Next, is the step of attributes importance. Our logic that concerned weighting is similar to our example in subsection 7.3.4. The R-A investor places more emphasis on the MaxDD which represents the risk-based metric, while the R-S investor on the ROI and Mean Return which represent the return-based metrics. Following the procedure set by (Martel, Khoury, & Bergeron, 1988), we have developed multiple weighting schemes to compare. For our purposes, we developed a total of 8 different weighting schemes in which we have created 4 pairs of investors. The direct rating score values were chosen to represent quite different types of investor profiles. The two investors in each pair exhibit

distinctive weights and are specifically intended for a different objective. To simplify the weighting process in the direct rating method we employed two abstract attributes: G_{Return} and G_{Risk}. For example, in the first pair, R-A gives the highest score (h_{Risk}=10) to the risk attribute while the risk-seeking gives the lowest (h_{Risk}=1). In every next pair score values of h_{Return} and h_{Risk} are incremented by 1 for R-A and R-S investors respectively. Also, in all pairs the h_{Return} and h_{Risk} have the highest value of 10 for R-S and R-A investors respectively. It should be noted that the scores are balanced equally between h_{ROI} and $h_{mean\text{-}returns}$ that belong to G_{Return}. Next by applying equation (4.9), we can calculate the final weights. Table 7-14 demonstrates the pairs, scores, and final attributes' weight.

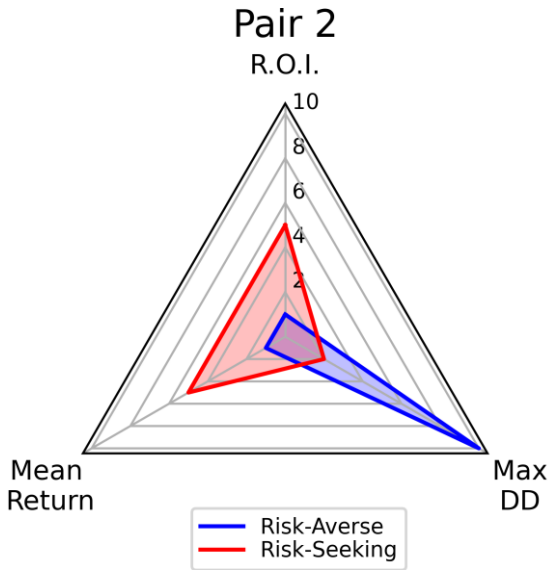
Table 7-14: Investor pairs and weight of attributes

Pair	**Risk-averse (R-A)**					**Risk-seeking (R-S)**				
	h_{Return}	h_{Risk}	w_{ROI}	$w_{mean\text{-}ret}$	w_{MaxDD}	h_{Return}	h_{Risk}	w_{ROI}	$w_{mean\text{-}ret}$	w_{MaxDD}
1	1	10	0.045	0.045	0.909	10	1	0.455	0.455	0.091
2	2	10	0.083	0.083	0.833	10	2	0.417	0.417	0.167
3	3	10	0.115	0.115	0.769	10	3	0.385	0.385	0.231
4	4	10	0.143	0.143	0.714	10	4	0.357	0.357	0.286

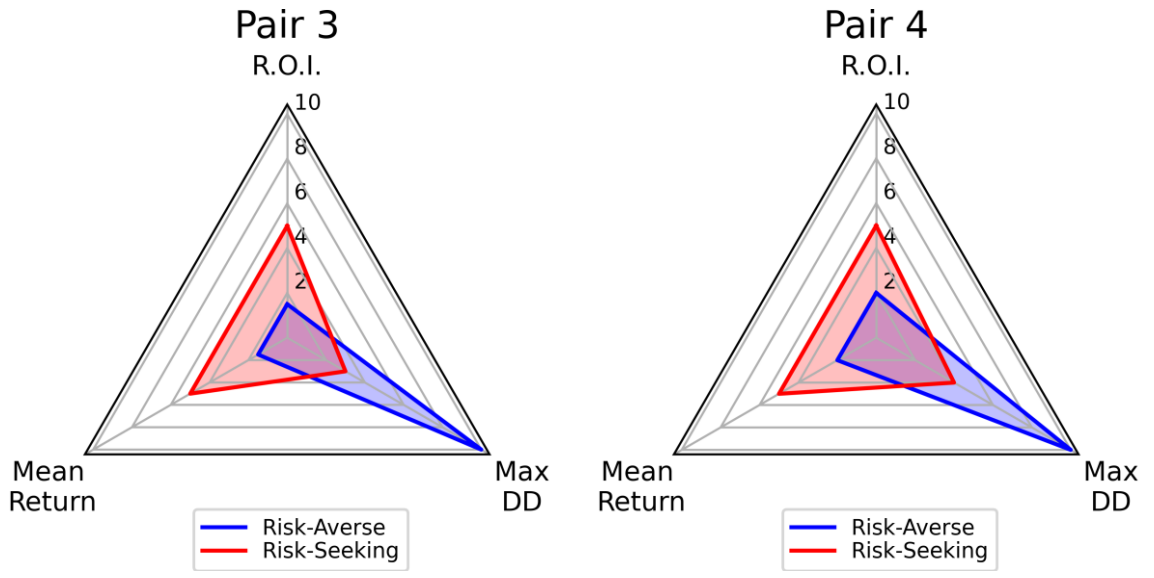
As shown in the table above the first pair (Pair 1) indicates the greatest contrast between two investors, while each subsequent one has a smaller contrast. Radar diagrams in Figure 7-4 (a, b, c, d) illustrate the significance of the difference between the investors in a clearer way.



(a)



(b)



(c) **(d)**

Figure 7-4: Investor pairs radar diagrams

7.3.8 Experimental Design and Procedure

A genetic algorithm implemented in Python was used to assist in the selection of the trading strategy parameters by utilizing PointRATE as a fitness function. Each phenotype, by using the evolved coefficients of the SMA index assumes long or short positions, depending on the index's signals, resulting in a time series of returns from trading. In order to optimize the SMA index, the periods of Slow-Moving Average (longer periods) and Fast-Moving Average (shorter periods) were evolved. As already mentioned, there are 8 different investor profiles I_p that have been tested for each financial instrument F_i. The genetic algorithm evolved a pool of 10 phenotypes for 50 generations, and the best solution was chosen which in our context refers to the two SMA periods. The process is repeated 100 times for each instrument, generating 100 best solutions $S_{i,j}^{p}$ for each investor, where S is the set of best trading strategy parameters, i is the financial instrument, j denotes the run number ($1 \leq j \leq 100$) and p is the investor profile.

A backtester was implemented in Python and it simulates the behavior of trading strategy according to the given set of historical prices. The initial capital of each strategy was 100,000 money units and the broker commission was set at 1% per trade. The maximum number of simultaneously opened positions is limited to one. The position size was set fixed at 50% of the initial capital. The positions are opened and closed based on the daily Close price of the time series of the financial instrument.

The overall experimental procedure is demonstrated in Figure 7-5.

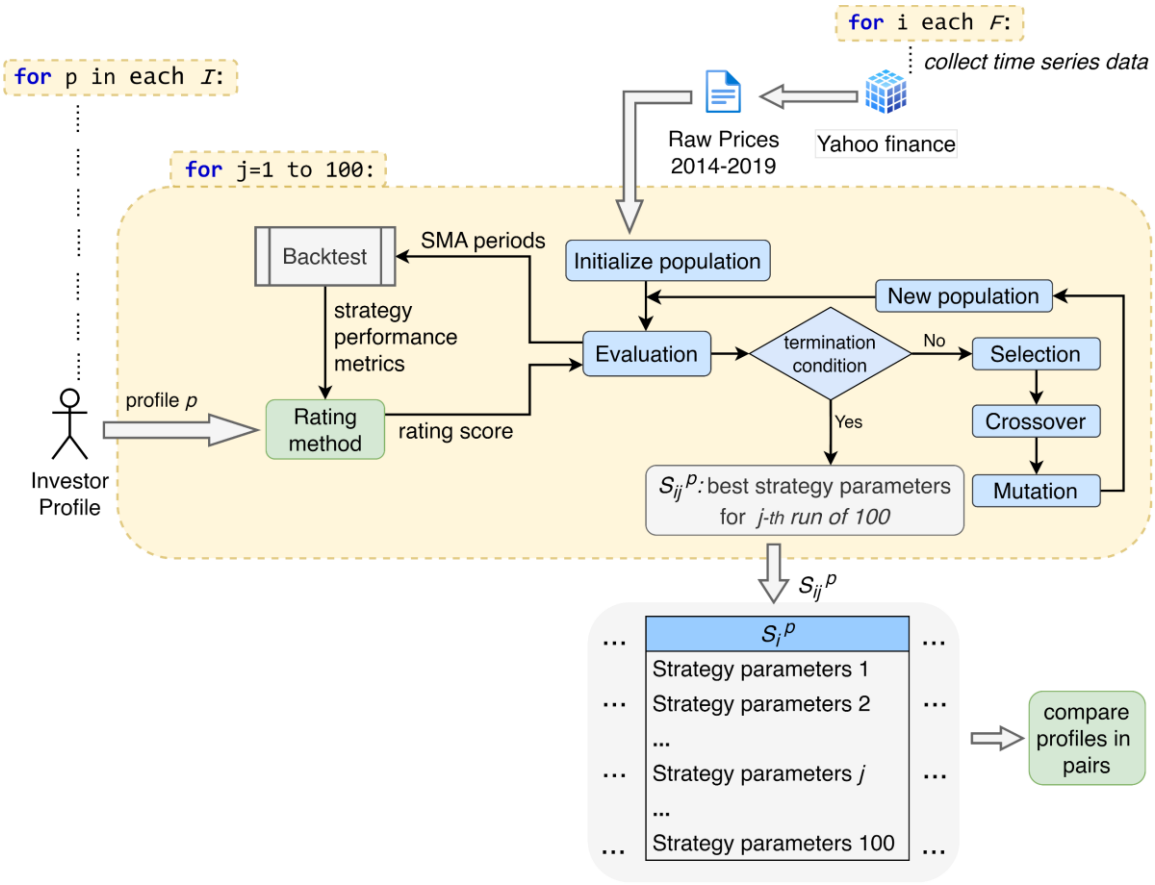


Figure 7-5: Experimental procedure of application

The comparison is made in two ways for each instrument and each pair separately:

1. First, using a Kolmogorov-Smirnov two-sample test, we compare the distribution of attribute levels between two investor profile sets' in each pair. In this way, we examine the possibility of two investor sets (from the same pair) containing solutions originating from the same distribution.
2. Second, we compare for each investor two groups of rating scores in each financial instrument. The first refers to the ratings after applying the solutions generated by the experimental procedure for the given investor and financial instrument. The second is computed after applying the solutions produced for one profile to the other (from the same pair). In this comparison, we expect a higher average rating score in the first group than in the second. To determine whether there is a significant difference between the two means, the two-tailed paired t-test is conducted.

7.3.9 Experimental Results

Table 7-15 presents the first comparison results. Specifically, it lists the Kolmogorov-Smirnov p-values of the attributes' values, for each financial instrument and each pair of investors, evaluating the probability that the null hypothesis is true. The null and alternative hypotheses are given below:

- H_o: The two samples originate from the same distribution at a significance level of a=0.05.
- H_1: The two samples do not originate from the same distribution at a significance level of a=0.05.

From Table 7-15, it is evident that in most comparisons, attributes' distributions are different between the investors belonging to the same pair. We reach this conclusion because the majority of observed p-values are below 0.05. Thus, in most cases, the two sets of solutions do not contain attribute values from the same distribution. However, this is a preliminary assessment of the results and it does not imply that the fitness function can satisfy user preferences. Therefore, we conduct the second comparison.

In the second comparison, there are two groups of ratings for each investor profile that belongs to a particular pair. The first group is the initial ratings that computed based on strategy parameters S_i^p produced by the experimental procedure for the particular investor profile *p* and *i* financial instrument. The second group of ratings is obtained after applying the solutions generated from the experimental procedure for the other investor's profile in the same pair and financial instrument. For the first group, the mean rating score is the $\bar{r}_p$ and for the second group is $\bar{r}'_p$. The null and alternative hypotheses are:

- H_o: There is no significant difference between $\bar{r}_p$ and $\bar{r}'_p$ at a significance level of a=0.05.
- H_1: The $\bar{r}_p$ and $\bar{r}'_p$ are significantly different at a significance level of a=0.05.

The results are depicted in Table 7-16 and Table 7-17 which contain the following columns: the pair of investors, $\bar{r}_p$, $\bar{r}'_p$, the difference $\bar{r}_p - \bar{r}'_p$, and the p-values from two-tailed paired t-tests when the mean rating scores of two groups of the pair are compared. In Table 7-16 the best strategy parameters of R-S are applied to R-A whereas in Table 7-17 the opposite (R-A to R-S).

Table 7-15: Kolmogorov Smirnoff test results

pair	F_i	ROI	MR	MaxDD	F_i	ROI	MR	MaxDD
1	MSFT	3.7E-36	3.6E-32	6.7E-27	AAPL	7.9E-06	0.0364	7.4E-04
2		3.6E-32	2.5E-30	5.0E-23		3.2E-05	0.0156	0.0156
3		3.9E-33	3.1E-31	8.8E-15		1.8E-06	0.0539^{ns}	0.0022
4		6.7E-27	1.6E-24	2.7E-25		8.0E-07	2.2E-04	1.2E-04
1	ASML	6.3E-05	4.4E-09	0.0539^{ns}	V	1.2E-04	0.0061	4.4E-09
2		2.2E-04	1.7E-09	0.3682^{ns}		6.3E-05	0.0364	2.9E-11
3		4.1E-04	6.4E-10	0.1112^{ns}		0.0156	0.0156	3.6E-07
4		0.0037	3.6E-07	0.2112^{ns}		4.1E-04	0.0156	1.1E-08
1	MA	2.7E-08	1.9E-29	2.5E-30	BABA	3.3E-12	2.0E-39	1.3E-21
2		4.4E-09	1.4E-28	1.0E-27		2.7E-08	4.1E-34	1.3E-21
3		1.6E-07	5.0E-23	1.0E-27		2.7E-08	3.2E-37	1.3E-21
4		2.4E-10	1.6E-24	2.6E-22		4.4E-09	1.9E-29	1.1E-17
1	SBUX	1.6E-07	0.0782^{ns}	6.3E-05	UNH	0.0099	0.0782^{ns}	0.0099
2		0.0013	0.0099	7.9E-06		0.0364	0.0241	0.0156
3		2.2E-04	0.2112^{ns}	0.0061		0.0061	0.0241	0.0241
4		1.2E-04	0.3682^{ns}	0.0013		0.0782^{ns}	0.1548^{ns}	0.0241
1	TMO	1.6E-05	3.6E-07	7.4E-04	CVS	0.0099	0.2819^{ns}	0.0061
2		2.2E-04	7.9E-06	0.0539		0.0364	0.0364	0.0364
3		0.7021^{ns}	0.0013	3.8E-06		0.0156	0.0539^{ns}	0.0241
4		0.583^{ns}	0.0037	0.3682^{ns}		0.0061	0.0037	7.9E-06

ns: not significant

From the tables below (Table 7-16 and Table 7-17), the results imply that the fitness function seems to be performing well and indeed directs the genetic algorithm in the right direction. This conclusion is reached after considering the positive difference ($\bar{r}_p - \bar{r}'_p$) in all cases and the observed p-values. Specifically, in rare instances, the p-value is considered not significant. The positive difference indicates that $\bar{r}_p > \bar{r}'_p$. This suggests that the solutions generated by GA for the given investor profile are more

suitable and satisfy the investor to a higher degree than those generated for the other profile that belongs to the same pair. Additionally, it was noticed as expected that the difference ($\bar{r}_p - \bar{r}'_p$) in most cases is decreased in every subsequent pair where the investor profiles are closer (i.e., smaller contrast). One example of the latter is the MSFT differences in Table 7-17, which are 63.38, 57.48, 52.63, and 45.3 for Pair 1, Pair 2, Pair 3, and Pair 4 respectively.

Table 7-16: Mean comparison: R-S solutions applied to R-A

pair	F_i	$\bar{r}_{R-A}$	$\bar{r}'_{R-A}$	diff.	p-value	F_i	$\bar{r}_{R-A}$	$\bar{r}'_{R-A}$	diff.	p-value
1	MSFT	90.39	85.81	4.58	6.2E-21	AAPL	95.2	93.79	1.4	2.0E-05
2		90.9	86.92	3.98	5.7E-22		95.54	94.79	0.75	0.0036
3		91.21	88.13	3.08	1.4E-15		95.88	94.76	1.12	3.7E-05
4		91.99	88.74	3.25	1.5E-17		96.08	95.14	0.93	2.5E-05
1	ASML	97.29	96.51	0.78	0.0033	V	97.14	95.97	1.17	3.5E-07
2		97.62	97.25	0.38	0.0326		97.44	96.02	1.42	1.0E-10
3		97.77	97.11	0.66	4.6E-04		97.46	96.52	0.93	3.5E-06
4		97.86	97.52	0.34	0.0424		97.5	96.6	0.9	6.2E-07
1	MA	94.31	90.73	3.58	7.2E-24	BABA	91.36	88.32	3.03	7.0E-22
2		94.68	91.4	3.28	1.4E-23		91.79	89.45	2.34	5.6E-17
3		95.23	92.26	2.98	9.3E-21		92.46	90.16	2.31	1.0E-14
4		95.52	93.15	2.37	1.5E-17		92.85	91.15	1.7	5.5E-15
1	SBUX	94.53	92.77	1.76	5.9E-07	UNH	95.37	95.0	0.37	0.044
2		94.71	93.66	1.05	1.9E-04		95.85	95.37	0.48	0.0221
3		94.86	94.09	0.77	0.003		96.13	95.72	0.41	0.0069
4		95.27	94.71	0.56	0.0084		96.5	96.09	0.41	0.0057
1	TMO	90.88	90.38	0.5	0.026	CVS	91.98	90.11	1.87	2.5E-06
2		91.5	90.95	0.55	0.0086		92.29	90.68	1.61	8.4E-05
3		92.01	91.66	0.35	0.0424		92.21	90.53	1.68	8.7E-06
4		92.39	92.19	0.21	0.2309[ns]		92.57	90.24	2.33	5.1E-07

ns: not significant

Table 7-17: Mean comparison: R-A solutions applied to R-S

pair	F_i	$\bar{r}_{R-S}$	$\bar{r}'_{R-S}$	diff.	p-value	F_i	$\bar{r}_{R-S}$	$\bar{r}'_{R-S}$	diff.	p-value
1	MSFT	89.95	26.57	63.38	2.6E-40	AAPL	46.59	37.06	9.53	2.0E-04
2		89.36	31.89	57.48	7.7E-37		52.1	42.66	9.44	2.5E-06
3		88.33	35.7	52.63	2.6E-36		56.85	48.32	8.53	4.3E-06
4		85.0	39.7	45.3	2.1E-28		57.22	49.26	7.96	1.3E-04
1	ASML	98.72	81.01	17.71	3.1E-12	V	52.1	36.78	15.33	5.3E-06
2		98.46	82.77	15.68	6.9E-12		59.45	43.81	15.64	4.9E-07
3		98.99	83.22	15.77	3.6E-14		54.88	47.24	7.63	0.0061
4		97.66	85.8	11.86	1.4E-09		59.02	47.96	11.05	3.4E-05
1	MA	84.14	55.42	28.72	2.2E-29	BABA	94.71	47.21	47.5	1.6E-35
2		84.14	58.62	25.52	1.6E-27		92.37	48.32	44.05	1.3E-32
3		83.42	61.6	21.83	9.4E-24		93.44	53.39	40.05	2.5E-29
4		82.06	63.31	18.75	8.9E-22		91.96	57.43	34.53	7.2E-28
1	SBUX	15.45	9.24	6.22	1.4E-05	UNH	45.01	36.99	8.02	0.0059
2		20.07	16.22	3.85	4.1E-04		47.26	42.78	4.47	0.0721^{ns}
3		26.4	22.77	3.64	0.0011		50.81	45.45	5.36	0.0234
4		30.64	27.52	3.12	1.9E-04		56.51	52.49	4.02	0.0621^{ns}
1	TMO	15.41	11.1	4.31	0.0046	CVS	12.95	8.35	4.6	2.9E-04
2		19.93	17.77	2.16	0.0918^{ns}		18.8	15.32	3.47	5.3E-04
3		26.48	23.59	2.89	0.014		25.23	21.15	4.08	1.3E-04
4		29.61	27.77	1.84	0.0632^{ns}		30.69	26.26	4.44	8.1E-05

ns: not significant

We conclude that our experimental results indicate that it is possible to model this direction, in other words, a user's preference, which is inherently a qualitative evaluation. Moreover, it does so in a strictly quantitative manner, as it is indeed able to use this modeling as a fitness function for the genetic algorithm.

7.3.10 Conclusions

In this section, we provide examples, in order to exhibit the rating methods' potential as decision processes in financial problems. The examples are concerned with investment selection, and portfolio rankings by utilizing rank and rating aggregation methods. Next, in the application part, a genetic algorithm was used for the optimization of a trading strategy and a rating system acts as a fitness function. Here, we examined if a rating method can be adapted to play the role of a fitness function in order to rate/rank strategies and therefore the genetic algorithm produces results capable of satisfying different users' preferences. Since the modeling of investors' preferences plays a crucial role in the optimization process, the PointRATE was utilized by taking into account well-known evaluation metrics as attributes. Additionally, because PointRATE acts as a fitness function, the modeling process followed general principles that are commonly used in fitness function development. Some of the key considerations were a clear purpose, simplicity, objectivity, an appropriate scale, and computational efficiency of the fitness function.

Regarding the three objectives we outlined in the introduction of this section, we have reached the following conclusions:

- For the first objective, we conclude that it is feasible to apply the rating methods discussed in previous chapters for financial management purposes. In the examples illustrated we have utilized PointRATE, rank and rating aggregation methods. The selection of these methods was based on their capability to measure multiple metrics without extensive modifications. The successful adoption of these methods in the financial sector does not necessarily imply superiority over other established methods.
- As far as the second objective is concerned, we found that PointRATE was able to guide a genetic algorithm in optimizing trading strategies in the financial stock market when different investor profiles are given as input. Specifically, the statistical tests were significant indicating that the method produces results capable of satisfying different users' preferences. Moreover, from the obtained experimental results it was noticed as expected that different users' preferences produce diverse results distributions.
- In terms of the last objective, the modeling used in the PointRATE method seems to produce stable results. This conclusion is made because, in all financial

instruments, the statistical tests showed a significant difference between the two groups of investors tested in each pair. It is also important to consider the attributes' weights carefully. Here we conclude that another implication of PointRATE is that it can be used as a fitness function in a genetic algorithm.

At this point, we will discuss some assumptions and weaknesses of the present application. One weakness of using technical analysis rules to generate trading strategies is that the strategies generated may be overfitted. This implies that an optimized strategy performs well on historical data but may not perform to the same degree on new unseen data or different market conditions. Nonetheless in this application, we do not aim to propose a profitable strategy. In contrast, our aim is oriented to test if a rating method can be useful in the optimization procedure. Also, another weakness is that the investors' profile is very generalized. This was made to minimize the subjectivity of the assessment of utility functions due to the fact that we have hypothetical investors. Certainly, the same experiment could be repeated with real investors by modeling their preferences and by considering more attributes related to the risk, stability, and profitability of investments. Subsequently, we can leverage additional benefits and related methods of MAUT/MAVT and MADM to enhance our application. Furthermore, the same experiment could be run by employing more advanced prediction techniques such as machine learning classification or regression. For example, in the context of machine learning model, the goal would be to optimize the hyperparameters of the model. Despite this, we opted for a solution that is simple and computationally efficient such as the SMA crossover strategy that does not require a training process.

7.4 User Preference Ratings and Recommendations: Application in Movies

7.4.1 Introduction

Big companies such as Netflix, Amazon, and eBay collect various data from users and combine them to generate ratings for products or services. The most common way for a person to make a review on a product or service is to assign a rating (or vote) on a sequential scale, using usually stars (from 1-5) or other indicators. Those ratings are called user preference ratings and the challenge in this application is to turn them into a single rating value for each item rated by the users. There are two common types of

ratings: the first is called explicit, and the second is implicit. The first refers to the ratings where the user has the role of evaluator and is asked directly to rate an item. On the other hand, the second type does not require a user to rate the items directly. Instead, each time a user interacts with the system, it can contribute to implicit rating computation (Claypool, Le, Wased, & Brown, 2001). In this application, we only deal with explicit ratings.

Another aspect of the exploitation of user-preference ratings is the development of recommender systems that could discover products or services that are relevant to the users' needs by exploiting users' past behaviors. In the case of movies, the purpose of those systems is to predict user preferences and suggest relevant movies to users.

In this section, we first utilize rating systems to rank movies and then we evaluate the results by performing comparisons. Next, we present possible applications. After this introduction, a short overview of theoretical aspects and related work for the user preference ratings and recommendations follows. Then, an illustrative example is provided to explain the rating process regarding the movies. The next subsections deal with the experimental aim, the data used, and the experimental process. Finally, we present the experimental results and this section ends with some conclusions.

7.4.2 Background and Related Work

Movie ranking is the process of ordering movies based on factors such as popularity, quality, or user preference ratings. User preference ratings are a valuable source of information for movie rankings and recommendation systems. In the context of movie rankings, user preferences are utilized to determine the relative importance or popularity of movies.

A popular rating system that ranks movies is developed by Internet Movie Database (IMDb) and it is based on a weighted average of the ratings given by users. The IMDb provides the top-rated list which is widely used as a reference for the popularity of movies. The procedure operates in the following steps. Each user can rate a movie on a 10-point scale. Then the rating is weighted considering the number of ratings the user has given in the past. Finally, to determine a rating of a movie the weighted average of all ratings is calculated. Overall, a movie's rating is a good indicator of the popularity of a movie among the general audience. According to IMDb (IMDb, 2023) their rating system implements the formula (7.1) that provides a true "Bayesian estimate" and

considers the number of votes each movie has garnered, the threshold number of votes needed for a movie to appear in the list, and the mean vote across all movies. The formula is computed as follows:

$$WR = \frac{v}{v+m}R + \frac{m}{v+m}C, \quad (7.1)$$

where *WR* is the weighted rating, *v* is the number of votes that a movie received, *m* is the minimum votes required, *R* is the average rating of a movie, and *C* is the mean vote across all movies.

Another rating system is based on centroid ratings which can be characterized as a way to determine the central tendency of user preference ratings (Langville & Meyer, 2012). The most common way to calculate centroid ratings is by taking the averages of all ratings. In movies, this method can be used as a measure of the overall popularity of a movie from users.

Also, the rating systems could be utilized directly or indirectly for recommendations. The direct utilization of the generated ranking list as a recommendation refers to a popularity-based recommender system where all the recommendations are the same for all users. The indirect refers to the combination of one or more types of well-known recommender systems. The purpose of a recommender system is to provide personalized recommendations to users based on their preferences. In addition, some of the main tasks that a movie recommender system performs are:

1. to find similar movies to a given movie,
2. to predict the rating or preference of a user for a movie,
3. to identify items that are most probable to be of interest to a user,
4. to recommend items to a user based on their preferences.

Especially, if we use a recommender for predictions, after training the model, the system is tested with a set of ratings from other users who have not been used in the training process. Then the performance of the model is usually assessed with RMSE and MAE metrics that compare the predicted ratings to the actual ratings. RMSE stands for Root Mean Square Error and is calculated by taking the square root of the average of the squared differences between predicted and actual values. The smaller the RMSE, the better the model. MAE stands for Mean Absolute Error and is determined by computing the average of the absolute differences between the predicted and actual values. Like the

RMSE, the smaller the MAE value, the better the model or algorithm is at making predictions. The formulas for calculating RMSE and MAE are as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i-\alpha_i)^2}{n}}, \qquad MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - \alpha_i|,$$

where n is the number of instances, p_i is the predicted values, and a_i is the actual values.

Generally, recommender techniques can be categorized into the following major types:

1. Collaborative Filtering methods recommend items that are popular with other users with similar preferences (Adomavicius & Tuzhilin, 2005). This approach is based on the assumption that people with similar preferences in the past, will also have similar preferences in the future. It has been suggested by experts in the field that collaborative filtering algorithms can be classified into two primary categories: (1) user-based algorithms, and (2) item-based algorithms. The user-based utilizes techniques that take into account the similarity between users in order to make recommendations while the item-based focuses on the similarity between items.
2. Content-based recommend items to the user that are similar to the same user's past preferences (Adomavicius & Tuzhilin, 2005). This category analyzes the associated features of an item to make recommendations (Burke, 2002). The idea behind content-based filtering is that users tend to prefer items that are similar to items they have previously expressed interest in.
3. Demographic recommender systems use demographic information (i.e., age, gender, education, etc.) of the user and this information is utilized, with the aim of identifying demographically similar types of users (Burke, 2002). These systems are based on the assumption that the users who belong to a certain demographic group, are users with similar preferences. One advantage of this category is that it may not necessitate user rating history as opposed to collaborative and content-based techniques (Burke, 2002). Also, this category can be useful in the case of cold-start problems which occur when there is not enough information about a new user or new movie (Chikhaoui, Chiazzaro, & Wang, 2011).
4. Utility-based recommender systems category bases its suggestions on the utility of each object for the user (Burke, 2002). Many utility recommendation systems have been developed, and most of them are based on the MAUT approach (Huang, 2011).

5. Popularity-based recommender systems make recommendations based on the popularity of items (i.e., top-ranked items) that have not been rated by the user (Javari & Jalili, 2014). These systems do not take into account the preferences of an individual user. This category can be used in cold-start scenarios.
6. Hybrid recommender systems consist of mixed approaches that aim to take advantage of the benefits that come from the combination of methods and eliminate the disadvantages of each approach (Burke, 2002). This category combines two or more methods with content-based and collaborative filtering being a popular combination (Adomavicius & Tuzhilin, 2005).

The rating methods discussed in this dissertation can be useful in many of the above techniques. For instance, once the top-rated movies have been identified, general suggestions can be made, or a collaborative filtering approach can be applied using the top-ranked films as a reference. Also, several research works have employed them and have reported positive outcomes.

Chartier et al. (Chartier, Langville, & Simov, 2010) in their study used the Colley method to rank movies in the Netflix dataset. They also exposed a limitation in the average rating algorithm and highlighted the significant role of a rating system in determining the final movie ranking position.

Xu et al. (Xu, Yao, Tong, Tao, & Lu, 2017) inspired by the Elo rating system, propose a novel rating comparison strategy called RAPARE, a generic method that can be integrated into existing methods in recommender systems. Their proposed strategy aims to learn the latent profiles of cold-start users/items by examining the differences between cold-start and existing users/items. The results of the experiments on five real datasets demonstrate that the RAPARE strategy is more efficient than existing methods in cold-start scenarios.

A noteworthy study by Gori and Pucci (Gori & Pucci, 2007) proposes ItemRank, a random-walk based rating method that can be used to rank products according to expected user preferences. ItemRank is similar to the PageRank algorithm. The main idea behind ItemRank is to use relationships between the items in order to calculate their similarity. The authors applied their method to the field of movie recommendations and found it to be more effective than the other techniques examined.

Furthermore, methods based on the MAUT approach were utilized in the study of Huang (Huang, 2011) for two recommendation contexts, movies, and notebooks. The

author examined if the utility-based techniques are more effective than the traditional content-based ones. Their results indicate that the effectiveness of utility-based techniques methods is contingent on the recommendation context.

Finally, Javari and Jalili (Javari & Jalili, 2014) proposed an algorithm for personalized recommendations to users by analyzing the popularity of items over time and predicting their future trends using wavelet transform. Particularly, the authors introduced two filtering methods which are based on the information obtained by analyzing the popularity trends of items over time. Their proposed algorithm was found to significantly enhance the accuracy and novelty of recommendations provided by classic methods as demonstrated in their results.

To summarize, rating systems can contribute to various methods and are especially effective in hybrid techniques.

7.4.3 Illustrative Example

In this simple hypothetical example, we aim to rate and rank movies based on their popularity. Particularly there are 3 movies and 5 users. Firstly, we present how to convert the user-movie matrix to a movie-movie matrix. The conversion is based on the fact that a hypothetical matchup between two movies exists if a user rates both movies. For example, in Table 7-18 User-1 rated Movie-1 and Movie-2 with 1 and 5 stars respectively. We considered this as a match between Movie-1 and Movie-2 with a final score of 1-5. The logic behind this conversion is based on (Chartier, Langville, & Simov, 2010).

Table 7-18: User-Movies matrix

User	Movie 1	Movie 2	Movie 3
User 1	1	5	0
User 2	4	3	5
User 3	2	4	0
User 4	1	0	4
User 5	0	2	3

Table 7-19: Movie-Movie matrix

Movie *i*	Movie *j*	Rating *i*	Rating *j*
Movie 1	Movie 2	1	5
Movie 1	Movie 2	4	3
Movie 1	Movie 2	2	4
Movie 1	Movie 3	4	5
Movie 1	Movie 3	1	4
Movie 2	Movie 3	3	5
Movie 2	Movie 3	2	3

Then we rate and rank movies based on pairwise comparison data from Table 7-19 and using the rating systems: Colley, Massey, Offense-Defense, and Centroid (or mean ratings). The Colley, Massey, and Offense-Defense ratings are computed in a similar manner as presented in subsections 3.3.2, 3.3.3, and 3.3.6 respectively. The analytical calculation of Centroid ratings mentioned in subsection 7.2.3 is shown below: First, the score matrix S is formed where S_{ij} is the average number of points (which in our case are the ratings given by users) scored by movie i against movie j. Then, K is the skew-symmetric matrix of the score differences matrix where each element $K_{ij} = S_{ij} - S_{ji}$ for the movies i, j.

$$S = \begin{array}{c} \\ \text{Movie 1} \\ \text{Movie 2} \\ \text{Movie 3} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Movie 1} & \text{Movie 2} & \text{Movie 3} \end{array} \\ \begin{pmatrix} 0 & 2.33 & 2.5 \\ 4 & 0 & 2.5 \\ 4.5 & 4 & 0 \end{pmatrix} \end{array} \quad K = \begin{array}{c} \\ \text{Movie 1} \\ \text{Movie 2} \\ \text{Movie 3} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Movie 1} & \text{Movie 2} & \text{Movie 3} \end{array} \\ \begin{pmatrix} 0 & -1.67 & -2 \\ 1.67 & 0 & -1.5 \\ 2 & 1.5 & 0 \end{pmatrix} \end{array}$$

Then rating vector r can be computed by Ke/n where e is a vector of 1's and n is the number of movies.

Table 7-20 demonstrates the results of the final rating and ranking lists for the illustrative example. As we observe the ranking lists generated across all the methods are identical.

Table 7-20: Movies example ratings and rankings

Rating System		Movie 1	Movie 2	Movie 3
Colley	Rating	0.325	0.425	0.75
	Ranking	3	2	1
Massey	Rating	-1.2708	0.1042	1.1667
	Ranking	3	2	1
Offense-Defense	Rating	10.6187	15.8213	21.9989
	Ranking	3	2	1
Centroid	Rating	-1.2222	0.0556	1.1667
	Ranking	3	2	1

7.4.4 Application Aim

The application aim is twofold:

(1) To generate ranking lists by different methods and then compare them with a baseline method.

(2) To utilize rating systems as a part of the recommendation process where our purpose is to improve recommendations for movies and predictions for user ratings.

In particular, for the second part, the development of this application drew inspiration from the study of (Javari & Jalili, 2014) where their proposed methodology is composed of two-steps. Particularly, in the first step, a filtering algorithm selects a subset of items, and then in the second step, a personalized list of items is recommended to the target user from the selected subset of items using any recommendation algorithm. Similarly, in our application, we have employed a two step-process. In the first phase, we use the rating systems to generate rankings and we create a subset of popular movies based on rankings. Then in the second step, we use item-based collaborative filtering with KNN to recommend movies that are chosen from the selected subset.

7.4.5 Data

The dataset used for the experiments of this application is the MovieLens (Harper & Konstan, 2015) dataset. The dataset, which was generated on September 26, 2018, includes 100,836 ratings on 5-star scale given in half-star increments (0.5 stars to 5.0 stars) by 610 users for a total of 9742 movies. The ratings were collected between the dates of March 29, 1996, and September 24, 2018. Only ratings from users who have provided ratings for a minimum of 20 movies were included as a criterion for selection.

7.4.6 Experimental Design and Procedure

As already mentioned, a hypothetical matchup between two movies exists if at least one user has rated both movies. In this application, we consider only the movies that have been rated from the 75% percentile of the total average number of ratings. This experiment is split into four parts (A, B, C, D):

A. The top 10 movies

Firstly, we rate and rank movies with the following systems: Colley, Massey, Keener, Offense-Defense, and Centroid. Then, the results of rating lists are aggregated into a single rating list with the Perron method described in 3.4.2. The final top 10 movies

are presented here as a partial list of the full aggregated list. The results of ratings can be used as a simple recommender system where all users have the same recommendation based on the movie's popularity. The top movies list will also be helpful for new users where it exists the problem of cold-start.

B. Ranking lists comparisons

In this part, we compare the rankings results of the rating systems tested in part A. Moreover, the IMDb rating system (IMDb, 2023) was added to the ranking comparisons as a baseline, since it is one of the most popular rating systems for movies. Our goal is to test how close the ranking results are from the mentioned rating systems to IMDb.

C. Recommendation example based on KNN

In this part, we use the KNN algorithm for the recommendation of movies. The type of recommendation is an item-based collaborative filtering recommender with KNN. Following that, in order to improve the recommendations, we first applied the top-ranked movies from the aggregated list as a filter to remove the less-known movies, and then we applied the KNN algorithm for movie recommendation.

D. Predictions of ratings and evaluation

Since part C is a simple example of a movie recommendation, in this part, we aim to evaluate the recommendations. In particular, we make predictions for user ratings based on item-based collaborative filtering with KNN and then evaluate the predictions according to RMSE and MAE metrics. The predictions in this experiment have been implemented by exploiting several functions of the SurPRICE (Simple Python Recommendation System Engine) library (Hug, 2020) in Python that is intended for building and analyzing recommender systems by exploiting users' explicit data ratings.

7.4.7 Experimental Results

A. The top 10 movies

Below we present our top 10 movies as a partial list with the highest ratings. This is our aggregated list after applying the Perron method on the lists from Colley, Massey, Offense-Defense, and Centroid. For our top 10 list, we have chosen the 75% percentile of the average total number of ratings of users.

Table 7-21: Top 10 movies

Movie Name	Year	Rating	Rank
The Shawshank Redemption	1994	0.1470	1
The Godfather	1972	0.1081	2
Fight Club	1999	0.0754	3
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb	1964	0.0743	4
Amelie (Fabuleux destin d'Amélie Poulain, Le)	2001	0.0595	5
Memento	2000	0.0430	6
Star Wars: Episode V - The Empire Strikes Back	1980	0.0427	7
Pulp Fiction	1994	0.0413	8
Casablanca	1942	0.0294	9
Spirited Away (Sen to Chihiro no kamikakushi)	2001	0.0292	10

B. Ranking lists comparisons

As previously mentioned, we have also considered the IMDb rating system in the comparisons. Specifically, for the IMDb rating system we have applied equation (7.1) where a minimum threshold for the number of votes *m* was defined using the same number used in the other methods (75% percentile of the average total number of ratings of users). The results of the Kendall-Tau correlation analysis are presented in Table 7-22 which follows the same format and logic as the previous tables of comparisons with Kendall's tau. In particular, the lower diagonal elements represent Kendall's tau values of each pair, while the upper diagonal elements are the p-values of each pair from the two-sided hypothesis test, whose null hypothesis is an absence of association.

Table 7-22: Kendall's tau comparison for movie ranking lists

	Colley	Massey	ODM	Centroid	Perron	IMDb
Colley	1.0000	2.49E-84	4.36E-84	1.76E-84	2.15E-87	3.19E-67
Massey	0.9400	1.0000	4.46E-93	1.10E-91	1.21E-91	4.93E-71
ODM	0.9380	0.9880	1.0000	3.27E-92	2.17E-91	2.91E-70
Centroid	0.9410	0.9810	0.9840	1.0000	1.73E-91	2.94E-69
Agg. Perron	0.9570	0.9810	0.9790	0.9800	1.0000	1.34E-69
IMDb	0.8370	0.8610	0.8560	0.8500	0.8520	1.0000

As we conclude from the results of Table 7-22, all p-values are less than 0.001, therefore we reject the null hypothesis (*tau*=0). This implies that the final ranking lists across all methods share many similarities. However, the comparisons with the IMDb have a slightly lower *tau* than the other pairs, which highlights small differences in the IMDb ranking list compared to the other lists.

C. Recommendation example based on KNN

In this part, we will focus on making recommendations for the movie named "Batman (1989)" based on the KNN algorithm with 5 nearest neighbors and the distance used is the cosine. The top 5 recommendations are depicted in Table 7-23.

Table 7-23: Top 5 recommended movies

Recommendation without filtering			**Recommendation based on the top 95%**		
Movies	**Year**	**Distance**	**Movies**	**Year**	**Distance**
Jurassic Park	1993	0.3605	Dances with Wolves	1990	0.4000
The Fugitive	1993	0.3602	Jurassic Park	1993	0.3605
Terminator 2: Judgment Day	1991	0.3544	The Fugitive	1993	0.3602
True Lies	1994	0.3032	Terminator 2: Judgment Day	1991	0.3544
Batman Forever	1995	0.2944	True Lies	1994	0.3032

As demonstrated, the recommendation system is capable of identifying a movie's genre. The example of "Batman (1989)" was used and it was identified as a Batman film, resulting in the recommendation of other Batman films. However, the limitation of the current system is that it does not take into account important characteristics such as the popularity and the overall rating of movies.

To obtain better recommendations, a filtering process is used first to narrow down the selection of movies to those that are in the top 95% of the full aggregated list results of part A from the full ranking list. In this way, ranking results are utilized to filter out the movies having a lower level of recognition or popularity. Then, KNN with 5 neighbors and cosine distance is performed again to make recommendations. By limiting the recommendations to the top-performing movies as determined by the full aggregated list, we have observed variations in the recommended movies, with the movie "Batman Forever" not being included. Given that this is only one recommendation sample, it is

difficult at this stage to draw a general conclusion. For this reason, in the next stage, we will try to predict users' ratings and evaluate the results.

D. Predictions of ratings and evaluation

At this stage, we utilize the KNN algorithm for making predictions in users' ratings. As a first step, we employed a 10-fold cross-validation strategy and ranked the movies in each of the resulting partitions separately with Colley, Massey, Keener, Offense-Defense, Centroid, Aggregation (Perron), and IMDb. In a similar manner to the previous step, we compare the results with and without the filtering process when we are using the KNN algorithm in conjunction with the 10-fold cross-validation technique. According to the top-ranked movies of each method, we conducted tests using various threshold values for filtering, by only considering those that ranked within the top 95%, 90%, 85%, and 80% of the training set in the respective fold. For evaluation purposes, we have used the two metrics described in 7.4.2 namely the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE). The results are depicted in Table 7-24 where the row labeled "No filter" represents the results without implementing any filtering process and the row labeled "Random" pertains to the results obtained from randomly filtering the data at a certain level.

Table 7-24: RMSE and MAE values for prediction results of movies

Filter by:	**RMSE**				**MAE**			
No filter	0.9530				0.7518			
Filter level:	**95%**	**90%**	**85%**	**80%**	**95%**	**90%**	**85%**	**80%**
Colley	0.9353	0.9218	0.9123	0.9061	0.7292	0.7116	0.6966	0.6838
Massey	0.9349	0.9182	**0.9105**	**0.9041**	0.7281	0.7090	0.6957	0.6818
ODM	0.9347	**0.9181**	0.9113	0.9042	0.7281	0.7089	0.6961	0.6816
Centroid	0.9355	0.9197	0.9108	0.9046	0.7286	0.7098	0.6959	0.6825
Aggregation	0.9355	0.9200	0.9110	0.9048	0.7286	0.7100	0.6958	0.6827
IMDb	**0.9301**	0.9182	0.9115	0.9049	**0.7245**	**0.7082**	**0.6943**	**0.6806**
Random	0.9531	0.9523	0.9524	0.9560	0.7521	0.7497	0.7518	0.7538

From the obtained results, we can conclude that the filtering we applied improves the predictions, since "no-filter" and "random filtering" yield poorer results with higher RMSE and MAE. This serves as an indication that the sports rating systems functioning well in this field and generating remarkable ranking results. The utilization of IMDb as a

filter has the lowest MAE value in all filter levels in comparison to the sports rating methods. In terms of the RMSE metric, IMDb is better at a 95% filter level while Offesne-Defense performs better at 90%, and Massey at 85% and 80%. As it is evident the utilization Colley method has the highest MAE and RMSE across nearly all filter levels in comparison with the other sports rating methods and IMDb method. This may be attributed to the fact that all examined methods consider points except the Colley system which takes into account only wins and losses. The terms "win" or "loss" in this case simply mean that the user prefers movie-*i* over movie-*j* or the opposite.

7.4.8 Conclusions

In this section, we demonstrated how to rate and rank movies by utilizing various rating systems studied in this dissertation. In reference to the twofold aim, we outlined in subsection 7.4.4 the following conclusions have been drawn:

- By using a real dataset from MovieLens we ranked movies based on their popularity with various rating systems and the results are aggregated into a single rating list. The ranking results derived from the methods applied are compared with a baseline method as a reference point. There is a strong correlation between the ranking list generated by IMDb (which serves as a baseline method) with those produced by the most rating systems. This suggests that the information is being effectively used to derive the ratings and rankings. Also, this implies that the generated rankings can be utilized as a popularity-based recommender.
- We tried to integrate ranking results into the recommendation process in order to enhance recommendations results. By utilizing a two step-process, we first use the rating systems to generate rankings, and then several certain percentages were used to select the top-ranked movies. In the second step, the selected movies from each percentage level are used in item-based collaborative filtering with KNN to predict user ratings and recommend movies. The main findings from the experimental results are positive and two general conclusions can be drawn from them. The first is that the predictions have been improved, as evidenced by the decrease in RMSE and MAE values. The second is that the rating methods applied seem to rate and rank the movies effectively.

One limitation is that the system only deals with a subset of movies in order to increase the likelihood that users will like the recommendations. This suggests that is

difficult to recommend unpopular movies to users as they lack the popularity to attract users. However, in other applications, to minimize the risk of recommending an item that a user may not like, it could be a safe strategy to limit recommendations to a subset of items. One possible improvement for this limitation is to make predictions for the popularity of the items in a similar way that is proposed by (Javari & Jalili, 2014). Thus, we will recommend unpopular items that are predicted to be popular in the future.

Finally, this approach we followed could be more effective for filtering and providing relevant information to the user if we incorporate demographic filters or limit the evaluation to specific movie genres. For example, if the user input is specific to a movie genre, a filter can be applied based on the popularity of movies in that genre before making recommendations.

7.5 Conclusions

In this chapter, we gave more emphasis to the application part of rating systems for real-world cases in other fields than sports. We started by utilizing them to rank the domain names through examples and afterward by utilizing real data. Then, we demonstrate their use in the financial management field with examples. Following that, we involve a rating system in the optimization procedure for the selection of trading strategy parameters. Finally, we applied rating systems in the field of user preference ratings where we focused on ranking and improving recommendations for movies.

Through this chapter, we conclude that rating systems can serve both as the rating/ranking of items and as part of larger processes with multiple objectives. Our findings are highly promising and positive. Overall, our experimental results suggest that the rating systems can be a valuable addition to the fields we examined.

However, our purpose was not to show or prove that the presented rating systems perform better than other well-known methods that have been used to solve those problems. Instead, we aimed to show that the rating systems mainly originating from the sports field can also be utilized in other areas satisfactorily. Assumptions and weaknesses for those applications and experiments as well as possible improvements that could be made are discussed in their conclusion subsections (7.2.7, 7.3.10, and 7.4.8). Therefore, we conclude that the research can be expanded and more comparisons can be made between the existing techniques and rating systems concerning the specific problems.

8 - RatingsLib: A Python Library for Rating Methods with Applications

8.1 Introduction

This chapter introduces RatingsLib, a library in Python that implements a plethora of methods with applications and examples in various fields. This work is published (Talattinis & Stephanides, 2022) as a software paper that includes all the details about the repository, code reproducibility, and the manual (documentation).

8.2 Library Overview

RatingsLib is a free Python library offered as open-source software under the MIT license. It requires Python 3.8 or newer, and the following Python libraries: numpy, scipy, pandas, scikit-learn, and joblib. As a Python package, it can be easily installed with the command "pip install". Additionally, RatingsLib provides high-quality and comprehensive documentation for use and API package. Also, it contains detailed step-by-step examples and several use cases.

8.3 Architecture

The architecture of the software consists of three basic components: datasets, ratings, and applications. An overview is presented in Figure 8-1. The "datasets" component is responsible for data functions i.e., parsing, data preparation, pre-processing, etc. For this component, the pandas library (pandas development team, 2020; McKinney, 2010) was used. The "applications" component embeds the necessary modules for each application and examples from various fields.

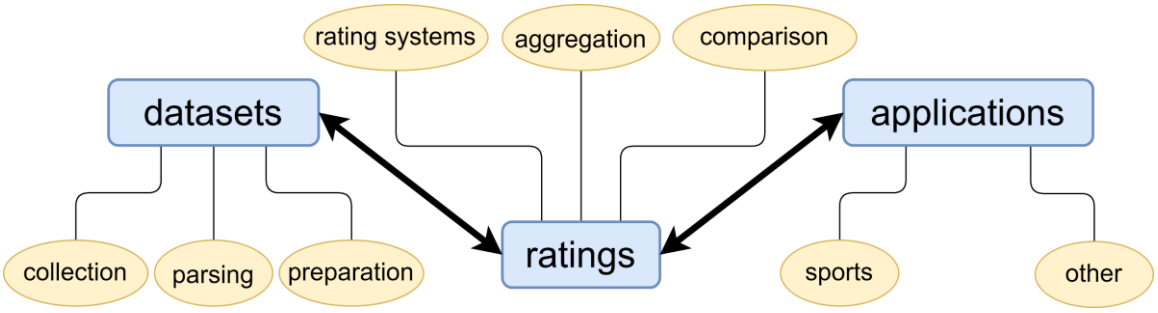


Figure 8-1: Software architecture

Rating is the key component of the library and interacts with the other two components. Basically, it comprises the implementation of rating systems, aggregation methods, and comparison measures. The class diagram of the ratings package is depicted in Figure 8-2. The rating procedure is split into three phases: preparation, computation, and rate. As we can observe, the *RatingSystem* is an abstract class and each rating system implements the abstract methods: `preparation_phase()`, `computation_phase()`, and `rate()`. We can perceive that the extension and implementation of new systems and methods can be easily made.

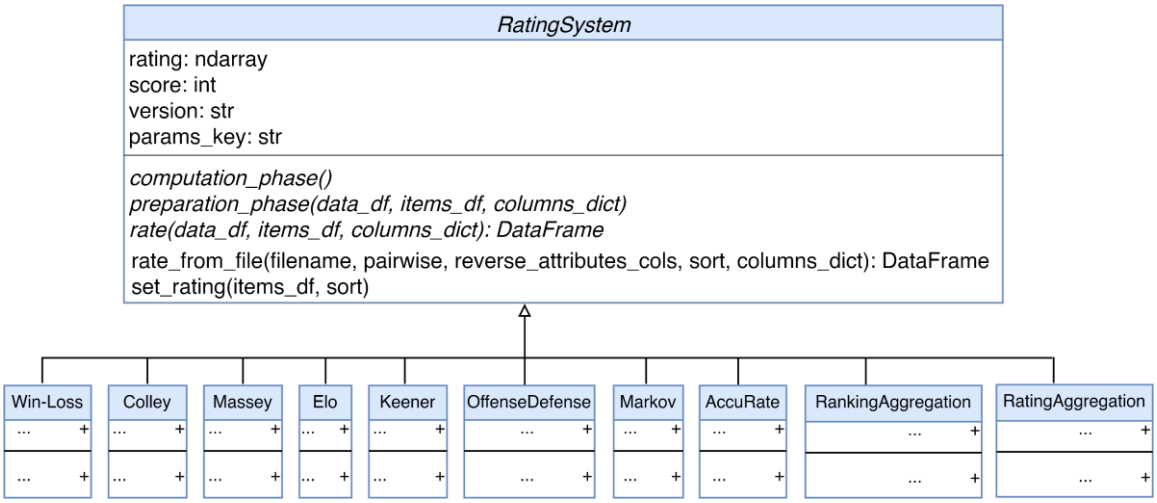


Figure 8-2: Class diagram of rating system

8.4 Functionalities

The most important features of the library will be presented here. More details on the functionalities and the use of software can be found in the documentation. Also, examples and small tutorials are available as Jupyter notebooks.

❖ Data collection and preparation related functions:

- Raw data collection: This function is available for soccer data where the user can collect and store online data from the football-data.co.uk repository (Football-data, 2023).
- Pre-process data: Basic pre-process procedures are implemented. For example, to remove unrated or NaN values.
- Calculation of statistics: Calculates statistics of the dataset. For example, soccer teams' statistics (e.g., number of total wins, total goals, etc.).

- Data preparation: Prepares the data in a suitable form required by the method or application. Since the library deals with pairwise comparisons, the data for rating and ranking must have the form of pairs.

❖ Rating/Ranking systems, aggregation methods, and comparisons measures

In this work, we have implemented the following systems and methods:

- Rating systems: Win-Loss, Colley (Colley, 2002), Massey (Massey, 1997), Keener (Keener, 1993), Elo(win and point versions) (Elo, 1978), Offense-Defense (Govan, Langville, & Meyer, 2009; Govan A. Y., 2008), GeM (Brin & Page, 1998; Govan A. Y., 2008; Ingram, 2007), AccuRATE (Kyriakides, Talattinis, & Stephanides, 2017).
- Rating aggregation methods: Markov (Govan A. Y., 2008; Ingram, 2007; Langville & Meyer, 2012), Perron (Keener, 1993; Langville & Meyer, 2012), Offense-Defense (Govan, Langville, & Meyer, 2009; Langville & Meyer, 2012; Govan A. Y., 2008).
- Rank aggregation methods: Borda count (Borda, 1784), Average Rank (Langville & Meyer, 2012).
- Comparison Measures: Kendall's tau (Kendall, 1938).

❖ Soccer outcome prediction

The prediction techniques can be performed in the following ways:

- Rank-based: The logic of prediction is that a higher rating is preferred over a lower rating or by considering the ranking of items.
- MLE: This method is explained in section 3.5. We note that for their computation the function minimize from the numpy (Harris, et al., 2020) library is used.
- Machine Learning Classification: The prediction can be made by utilizing the ratings of teams as features for the machine learning classifiers. Also, it is possible to combine other features (e.g., game statistics). The available classifiers from scikit-learn library (Pedregosa, et al., 2011) can be passed in RatingsLib's functions.

There are two types of predictions that are offered by RatingsLib: hindsight and foresight which are explained in subsection 3.6.2. For the foresight predictions, we have included the following options to split the train and test sets:

1. percentage split (e.g., 70% train, 30% test).
2. match week number split (e.g., 1-25 week train, 26-38 week test).
3. anchored walk-forward (AWF) (Jaekle & Tomasini, 2019) with a window size.

Additionally, in order to test multiple rating methods and classifiers simultaneously, the multiprocessing library joblib (Joblib Development Team, 2020) has been employed to improve the overall computational time.

Finally, the prediction procedure provides a classification and accuracy report.

- Parameters Tuning: The rating system parameters can be tuned to improve a specific metric, e.g., Accuracy, F1-score, etc. The tuning is performed after defining the search space of each parameter. An example of the tuning of Elo system parameters is provided in the manual.

❖ Applications and Examples in Other Fields

- Domain names ranking: An illustrative example of domain names ranking based on (Talattinis, Zervopoulou, & Stephanides, 2014), is provided.
- Finance: Examples are provided from investment selection and portfolios ranking by utilizing rank and rating aggregation methods.
- Rating and ranking of movies from users' ratings: This application is tested on the MovieLens dataset (Harper & Konstan, 2015).

❖ Testing – Validations

RatingsLib provides unit tests for most of the mentioned functionalities.

8.5 Usage

Below we show an example of using RatingsLib to rate soccer teams of the EPL in the 2018-2019 season by the Win-Loss method. Also, examples from soccer and the NFL are included in the manual.

```
from ratingslib.datasets.filenames import get_season_footballdata_online
from ratingslib.ratings.winloss import Winloss
filename = get_season_footballdata_online(2018, 'EPL')
Winloss().rate_from_file(filename)
```

As we observe from the code above, the computation of ratings is performed by calling `rate()` or `rate_from_file()` methods of the rating system.

8.6 Impact Overview

RatingsLib is an open-source library in Python that is not only limited to the implementation of rating systems but comprises applications as well. Moreover, several examples are provided with the aim of making clearer the use of implemented methods in other areas and fields. The fact that the code is written in Python makes the tool more accessible since Python is very popular.

The objective of RatingsLib is twofold: (1) to rate, rank, and compare a set of objects/items, and (2) to apply rating methods in real-world applications. In this way, the software can be helpful for researchers, data scientists, academics, students, and professionals. For example, this library may significantly help sports analysts, bettors, and coaches to rate and rank teams or players, and make predictions for the outcome of future sports games. Furthermore, decision-makers, investors, and portfolio managers can adapt code to solve ranking problems in a simple way. Also, it can help scientists combine a plethora of well-known rating methods with other approaches and techniques such as machine learning. This software can also be very useful for educational purposes. Since most of the ranking systems provided in this library use linear algebra and computational methods, this could help students understand the applications and extensions of linear algebra, optimization models, and computational methods.

8.7 Conclusions

Obviously, rating and ranking comprise a scientific field that attracts not only the interest of academia but also of the business community. For this reason, unlike other available tools, our work is more oriented to applications of rating methods. RatingsLib is the first open-source library that integrates rating methods with machine learning in the field of sports outcome prediction. Also, applications in other fields such as movie rankings are tested with a benchmark dataset. In addition, the software embeds several methods such as rating aggregation methods and calculation of soccer outcome probabilities. Furthermore, comparison measures and data functions are provided to make the library more comprehensive.

9 - Conclusions

This chapter summarizes the thesis and provides its significance in the broader context of the research field. Initially, in section 9.1 conclusions are drawn related to objectives and contributions. Next, in section 9.2 limitations of the research are discussed, and finally, in section 9.3 the possible future directions are considered. The conclusions, limitations, and future work are also discussed in more detail in the "conclusions" sections and subsections of previous chapters.

9.1 Summary and Conclusions

The present dissertation is focused on 4 objectives mentioned in the Introduction Chapter (section 1.3). In the first place, we review and summarize those objectives.

❖ 1st Objective

The first objective was oriented to propose novel rating systems or altered approaches to existing ones. Regarding this objective two rating systems are proposed. First, the rating method entitled "AccuRATE" is introduced to provide ratings for soccer teams based on the efficiency of each team in offensive opportunities. Second, "PointRATE" is introduced as an altered version based on WSM and MAUT/MAVT approaches. While the first is oriented on soccer teams based on game outcomes, margin of victory, and shooting accuracy, the second is a generalized approach that considers the utility/value of the user and can also be applied in other fields to rate items. To further evaluate the reliability of their results, a sensitivity analysis is conducted on the EPL (2005-2018) dataset to investigate their performance under different scenarios. Also, their ranking results are compared to those of established rating systems for validation.

❖ 2nd Objective

Regarding the second objective, the main application of this thesis focuses on predicting the outcome of upcoming games in the EPL by utilizing various established rating systems, machine learning techniques, and statistical methods. Specifically, the ratings generated by the rating methods are integrated into the feature engineering process of machine learning techniques. The main application is introduced gradually from the previous chapters, starting with an illustrative example that is applied to all rating systems, followed by a demonstration of their predictive performance by applying Rank-based and statistical prediction techniques.

By categorizing the prediction models into accuracy-oriented and profit-oriented, three different prediction techniques have been applied and tested (i.e., Rank-based, statistical-based, and ML-based) in the main application. In addition, for profit-oriented models, money management and cost-sensitive approaches are taken into consideration in order to improve the overall performance of betting portfolios. The overall results of each prediction technique are compared to the other techniques in order to assess their performance. After identifying the best-performing models of each category they are evaluated and compared to the results derived from other baseline prediction models.

❖ 3rd Objective

In the context of the third objective, three real-world applications are provided where our proposed methods and other popular ranking systems are applied. These cases fall outside the sports field and comprise the following fields:

A. The case of the Domain Name Market: Rating systems are utilized to rank the domain names through examples and afterward by utilizing real datasets.
B. Financial Management and Optimization: Started with an illustration of rating methods in the financial management field with examples, and following that, our proposed rating system PointRATE is involved as a fitness function in the optimization procedure for the selection of trading strategy parameters.
C. Users' preference ratings and recommendations: Rating systems are applied in the field of user preference ratings where the primary focus is to generate accurate rankings for movies and enhance movie recommendations.

❖ 4th Objective

Concerning the fourth objective, RatingsLib has been published as open-source software that is not only limited to the implementation of rating systems but comprises applications as well.

Our findings confirm the contributions mentioned in the Introduction Chapter (section 1.4). In the second place, we will review each contribution and highlight our findings and conclusions.

❖ 1st Contribution

In the context of the first contribution, two rating systems are proposed and have been applied successfully to rate soccer teams. They produced quite stable rankings and exhibited promising results in comparison to established rating systems for both ranking

results and their performance in the soccer forecasting application. PointRATE achieved the highest Accuracy in every category (RANK, MLE, and ML), and in statistical-based predictions (MLE) has the highest accuracy among all the accuracy-oriented models. Also, in cost-sensitive learning the utilization of AccuRATE (ratings as features) has the 3rd position in terms of effectiveness in the average Sharpe ratio of betting portfolios. In addition, the utilization of AccuRATE ratings and average odds as features by Naive Bayes and Kelly criterion as money management is placed in the top 5 best-performing betting portfolios in terms of risk-adjusted performance. Moreover, PointRATE has been utilized as a fitness function for a genetic algorithm where it was able to guide a genetic algorithm in optimizing trading strategies in the financial stock market when different investor profiles are given as input. Overall, the proposed rating methods can be adapted for use in other related contexts where the ratings and rankings are important.

❖ 2nd Contribution

Our second contribution is the comprehensive analysis we provided for the soccer outcome prediction in the EPL (2009-2018) by applying three prediction techniques where several rating methods and machine learning algorithms are compared for their predictive ability. The empirical study and evaluation procedure are primarily based on simple historical data and well-known bookmakers' odds. Our findings are of practical significance and depending on the purpose different conclusions may be reached.

- Accuracy-Oriented: For the accuracy-oriented purpose, findings indicate that the statistical-based prediction technique (MLE) achieves better probabilistic predictions with lower RPS than machine learning classification techniques. However, machine learning classifiers are better suited for outcome predictions since they consistently achieve a higher F1-score than other methods. Also, the statistical-based prediction technique (MLE) provides on average the highest performance in terms of Accuracy compared to other prediction techniques. Moreover, our analysis reveals that Naive Bayes exhibits the best performance in F1-score across all feature sets (rating scores).
- Profit-Oriented: For the profit-oriented models, the main idea of our approach is that the primary emphasis should not be placed on generating more accurate models, neither only profitable models, but profitable risk-adjusted models, i.e., models that can lead to profitable bets with low risk. The methodology used shares similarities with techniques utilized in the selection of investments, portfolios, and financial trading strategies. Our first investigation is that the Rank-based and statistical-based

prediction (MLE) techniques are limited to generating profitable risk-adjusted models. Also, traditional machine learning classifiers that utilized rating scores as features (from one rating system each time), improve the overall models but may not lead to reliable and generalized ones. The results suggest that the most effective models are those generated by cost-sensitive learning when integrating the ratings of teams and betting odds as input features to machine learning classifiers and when the Fractional Kelly criterion is employed as money management. This confirms our initial hypothesis that the employment of cost-sensitive techniques can minimize the average cost per prediction (or maximize the average benefit). Also, the best average Sharpe ratio was obtained by the utilization of Elo-Point ratings as features from any classifiers. Moreover, the classifier that demonstrated the highest performance across all feature sets (rating scores) was the SVM in terms of the average Sharpe ratio.

Overall, the category of models utilized cost-sensitive learning with rating scores of teams and betting odds as features, it could consist of the inception of a new investigation sequence in sports betting.

❖ 3rd Contribution

The third contribution of this thesis deals with the demonstration of three applications of rating systems in fields outside the realm of sports.

A. The case of the Domain Name Market: Massey, Colley, and GeM rating systems are shown to be applicable for the case of domain name rankings according to our empirical results. The approach we introduced can serve as a complementary tool to the value estimation of domain names and their ranking can be used by many groups of people, such as domain traders, and investors.
B. Financial Management and Optimization: It can be concluded that it is feasible to apply PointRATE, rank and rating aggregation methods for financial management purposes. Also, the statistical tests performed were significant indicating that PointRATE can be utilized successfully as a fitness function during the optimization of stock trading strategy parameters, as produces results capable of satisfying different users' preferences.
C. Users' preference ratings and recommendations: The generated rankings from rating systems (Massey, Colley, Keener, Offense-Defense, and Aggregated-Perron) for the movies from MovieLens (Harper & Konstan, 2015) dataset seem to be reliable as they were compared to the IMDb rating system as a baseline.

Therefore, they can be utilized as a popularity-based recommender for movies. Next, by employing a 10-fold cross-validation strategy, the rankings of movies generated by rating systems in the training phase are utilized as a popularity filter in item-based collaborative filtering with KNN to predict new user ratings and recommend movies. The conclusion reached is that the integration of rating systems in the recommendation process leads to enhanced user rating predictions, as evidenced by the decrease in RMSE and MAE values.

- 4th Contribution

The contribution of the RatingsLib lies in the idea that is an open-source software suitable for use in a range of settings including academic and professional environments. The software offers a lot of functionalities around the rating systems and their applications and it can be beneficial for different groups of people including researchers, data scientists, academics, students, and professionals. For example, sports analysts, bettors, and coaches can use this library to rate and rank teams or players and make future predictions for the outcome of sports games. Moreover, the code can easily be adapted by a variety of individuals and groups such as decision-makers, investors, and portfolio managers to solve ranking problems in a simple way. Also, it can facilitate researchers in integrating a plethora of well-known rating methods with other approaches and techniques such as machine learning. This software can be useful for educational purposes. Since most of the ranking systems provided in this library are based on linear algebra and computational methods, this could help students understand the applications and extensions of linear algebra, optimization models, and computational methods.

In conclusion, after reviewing the objectives and contributions and drawing conclusions, we believe that the present dissertation can help the field progress and provide new insights into the field of Rating and Ranking, and its applications. Overall, the applications, proposed methods, and findings of this work have the potential to be useful for a variety of individuals and groups.

9.2 Limitations of the Research

It is generally accepted that there is not a single best method to rate and rank items. Although many have been proposed, it is the application's constraints that dictate which one should be used. Different methods exhibit different levels of efficiency under different conditions. Concerning our proposed rating methods, some limitations need to

be considered. AccuRATE is primarily designed for soccer team ratings by considering the game outcomes, margin of victory, and shooting accuracy, and this adds an extra degree of difficulty in its applicability to other fields. Essentially, the difficult part which is the key to using it in other fields is to determine the term “shooting accuracy” for the items to be ranked. In other sports, this may be effortless, but in completely different contexts, it can be more challenging. Also, the soccer team modeling in PointRATE faces some limitations. One limitation is the lack of consideration for real-world user utility while another weak point is that all attributes are modeled using the same utility function. However, this was done to simplify modeling and keep it as unbiased as possible.

In terms of our main application, the machine learning techniques used were centered on classification methods while regression models have not been explored. As a result, the main application does not provide any insight into how regression models performed in predicting the margin of victory in games and then deciding on the game outcomes. Moreover, only basic data attributes from teams and simple machine learning classifiers have been considered in the experimental part. Also, the proposed cost matrix scheme is oriented more toward profitability than to risk-adjusted performance.

As for the applications of Chapter 7, more emphasis was given to the application part of rating systems for real-world cases in other fields than sports. However, the successful adoption of these methods in various sectors does not necessarily imply superiority over other established methods in each field. Therefore, we conclude that the research can be expanded and more comparisons can be conducted with other commonly used techniques concerning the specific problems.

9.3 Future Directions

While the insights derived from this dissertation are important, there are several opportunities for future work. To expand upon our proposed rating systems, a possible future direction would be to further investigate their effectiveness in other sports, fields, and applications. Concerning the modeling of PointRATE for soccer teams rating, can be further optimized by incorporating experts, sports analysts, and bettors for the effective modeling of attributes. Also, more capabilities of the MAUT/MAVT and additional benefits from related methods of MADM could be leveraged to enhance our applications.

With respect to the soccer forecasting application, although it was centered on the EPL one of the most famous leagues worldwide, it would be interesting to see how well

it performs in other leagues. Also, while our experimental analysis produced promising results using only basic data attributes, those can be extended to more advanced ones where more teams' statistics can be involved. Moreover, the dataset can be expanded to incorporate data from betting exchanges. Additionally, the study could benefit from the inclusion of sentiment analysis data related to sports teams or players which could help to generate more informed predictions.

Another potential avenue for future work in the soccer forecasting application would be to investigate the use of regression to predict the margin of victory in games. Furthermore, it would be interesting to explore more complex algorithms such as boosting, bagging, and deep learning approaches that could potentially improve the robustness of our prediction models. Also, the proposed cost matrix scheme can be improved to consider the cost (or benefits) in a not only profitable way but also a more risk-adjusted way.

The applications of Chapter 7 can be extended to examine more advanced cases and more comparisons with other established methods. For example, in the application of domain name rankings, the Discounted Cash Flow method (DCF) can be used to evaluate and rank domain names after estimating their future cash flows. Also, our approach can be adapted to new alternative markets such as the NFT (Non-fungible Token) market or the virtual properties market in Metaverse, as long as data and relevant factors are available for analysis. A possible direction for future work for the Financial Management and Optimization application could be the replacement of the SMA strategy of technical analysis with a machine learning prediction model in order to provide signals for the trades in the stock market. This implies that the genetic algorithm will use the rating system as a fitness function to optimize the hyperparameters of the machine learning model in reference to investor preferences. Finally, in terms of movie application, it is important to balance popularity with relevance as focusing more on popular items may not provide personalized recommendations that meet the specific needs and preferences of the user. Therefore, possible future work is to make predictions for the popularity of the items in a similar way that is proposed by (Javari & Jalili, 2014). Thus, unpopular items will also be recommended if they are predicted to be popular in the future.

As the field of Rating and Ranking continues to evolve, there are several avenues for future research. Overall, this dissertation provides valuable insights and a foundation for future research in the field.

Bibliography

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734–749. doi:10.1109/TKDE.2005.99

Angelini, G., & Angelis, L. D. (2019). Efficiency of online football betting markets. *International Journal of Forecasting, 35*(2), 712–721. doi:https://doi.org/10.1016/j.ijforecast.2018.07.008

Arrow, K. J. (1951). *Social Choice and Individual Values.* Wiley.

Baboota, R., & Kaur, H. (2019). Predictive analysis and modelling football results using machine learning approach for English Premier League. *International Journal of Forecasting, 35*(2), 741–755. doi:https://doi.org/10.1016/j.ijforecast.2018.01.003

Barros, C. P., & Leach, S. (2006). Performance evaluation of the English Premier Football League with data envelopment analysis. *Applied Economics, 38*(12), 1449–1458. doi:10.1080/00036840500396574

Belton, V., & Stewart, T. (2002). *Multiple criteria decision analysis: an integrated approach* (3 ed.). Springer Science & Business Media.

Bennett, J., & Lanning, S. (2007). The Netflix Prize. *Proceedings of KDD cup and workshop*, (pp. 3–6).

Berrar, D., Lopes, P., & Dubitzky, W. (2019, January 01). Incorporating domain knowledge in machine learning for soccer outcome prediction. *Machine Learning, 108*(1), 97–126. doi:10.1007/s10994-018-5747-8

Berutich, J. M., López, F., Luna, F., & Quintana, D. (2016). Robust technical trading strategies using GP for algorithmic portfolio selection. *Expert Systems with Applications, 46*, 307–315. doi:https://doi.org/10.1016/j.eswa.2015.10.040

Bikmukhametov, T., & Jäschke, J. (2020). Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. *Computers & Chemical Engineering, 138*, 106834. doi:https://doi.org/10.1016/j.compchemeng.2020.106834

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* New York: Springer-Verlag.

Bodin, L., & Epstein, E. (2000). Who's on first–with probability 0.4. *Computers & Operations Research, 27*(3), 205–215. doi:https://doi.org/10.1016/S0305-0548(99)00002-7

Boldi, P., Santini, M., & Vigna, S. (2005). PageRank as a Function of the Damping Factor. *Proceedings of the 14th International Conference on World Wide Web*

(pp. 557-–566). New York, NY, USA: Association for Computing Machinery. doi:10.1145/1060745.1060827

Borda, J. d. (1784). Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)*.

Boshnakov, G., Kharrat, T., & McHale, I. G. (2017). A bivariate Weibull count model for forecasting association football scores. *International Journal of Forecasting, 33*(2), 458–466. doi:https://doi.org/10.1016/j.ijforecast.2016.11.006

Bottomley, P. A., Doyle, J. R., & Green, R. H. (2000). Testing the Reliability of Weight Elicitation Methods: Direct Rating versus Point Allocation. *Journal of Marketing Research, 37*(4), 508–513.

Bradley, R. A., & Terry, M. E. (1952). Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika, 39*(3/4), 324–345.

Brandouy, O., Mathieu, P., & Veryzhenko, I. (2013). Risk Aversion Impact on Investment Strategy Performance: A Multi Agent-Based Analysis. In A. Teglio, S. Alfarano, E. Camacho-Cuena, & M. Ginés-Vilar (Eds.), *Managing Market Complexity: The Approach of Artificial Economics* (pp. 91–102). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-31301-1_8

Brans, J. P. (1982). L'ingénierie de la décision: l'élaboration d'instruments d'aide a la décision. Methodé PROMETHEE. In R. Nadeau, & M. Landry (Eds.), *L'aide à la décision: Nature, Instruments et Perspectives d'Avenir* (pp. 183–213). Quebec, Canada: Presses de l'Université Laval.

Breiman, L. (2001, October). Random Forests. *Machine Learning, 45*(1), 5–32.

Brier, G. W. (1950). Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review, 78*(1), 1–3.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems, 30*(1), 107–117. doi:https://doi.org/10.1016/S0169-7552(98)00110-X

Burke, R. (2002, November 01). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331–370. doi:10.1023/A:1021240730564

Buursma, D. (2011). Predicting sports events from past results Towards effective betting on football matches. *In 14th Twente Student Conference on IT. 21*. Twente, Holland.

Callaghan, T., Mucha, P. J., & Porter, M. A. (2007). Random Walker Ranking for NCAA Division I-A Football. *The American Mathematical Monthly, 114*(9), 761–777. doi:10.1080/00029890.2007.11920469

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to Rank: From Pairwise Approach to Listwise Approach. *Proceedings of the 24th International*

Conference on Machine Learning (pp. 129-–136). New York, NY, USA: Association for Computing Machinery. doi:10.1145/1273496.1273513

Castellano, J., Casamichana, D., & Lago, C. (2012). The Use of Match Statistics that Discriminate Between Successful and Unsuccessful Soccer Teams. *Journal of Human Kinetics*, *31*(2012), 137–147. doi:10.2478/v10078-012-0015-7

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, *2*(6), 429–444.

Chartier, T. P., Kreutzer, E., Langville, A. N., & Pedings, K. E. (2011). Sensitivity and Stability of Ranking Vectors. *SIAM Journal on Scientific Computing*, *33*(3), 1077–1102. doi:10.1137/090772745

Chartier, T., Langville, A., & Simov, P. (2010). March Madness to Movies. *Math Horizons*, *17*(4), 16–19.

Chikhaoui, B., Chiazzaro, M., & Wang, S. (2011). An Improved Hybrid Recommender System by Combining Predictions. *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, (pp. 644–649). doi:10.1109/WAINA.2011.12

Choi, H., & Varian, H. (2009). Predicting initial claims for unemployment benefits. *Google Inc*, *1*(2009), 1–5.

Choi, H., & Varian, H. (2012). Predicting the Present with Google Trends. *Economic Record*, *88*(s1), 2–9. doi:https://doi.org/10.1111/j.1475-4932.2012.00809.x

Claypool, M., Le, P., Wased, M., & Brown, D. (2001). Implicit Interest Indicators. *Proceedings of the 6th International Conference on Intelligent User Interfaces* (pp. 33–40). New York, NY, USA: Association for Computing Machinery. doi:10.1145/359784.359836

Colley, W. (2002). Colley's bias free college football ranking method: The Colley Matrix Explained.

Constantinou, A. C. (2019). Dolores: a model that predicts football match outcomes from all over the world. *Machine Learning*, *108*(1), 49–75. doi:10.1007/s10994-018-5703-7

Constantinou, A. C., Fenton, N. E., & Neil, M. (2012). pi-football: A Bayesian network model for forecasting Association Football match outcomes. *Knowledge-Based Systems*, *36*, 322–339.

Copeland, A. H. (1951). A reasonable social welfare function. Mimeo, University of Michigan USA.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Coulom, R. (2007). Computing “elo ratings” of move patterns in the game of go. *ICGA Journal, 30*, 198–208. doi:10.3233/ICG-2007-30403

Dacorogna, M. M., Gençay, R., Müller, U. A., & Pictet, O. V. (2001). Effective return, risk aversion and drawdowns. *Physica A: Statistical Mechanics and its Applications, 289*(1), 229–248. doi:https://doi.org/10.1016/S0378-4371(00)00462-3

Das Sarma, A., Das Sarma, A., Gollapudi, S., & Panigrahy, R. (2010). Ranking Mechanisms in Twitter-like Forums. *Proceedings of the Third ACM International Conference on Web Search and Data Mining* (pp. 21–30). New York, NY, USA: Association for Computing Machinery. doi:10.1145/1718487.1718491

Davidson, R. R. (1970). On Extending the Bradley-Terry Model to Accommodate Ties in Paired Comparison Experiments. *Journal of the American Statistical Association, 65*(329), 317–328.

Deng, H., Yeh, C.-H., & Willis, R. J. (2000). Inter-company comparison using modified TOPSIS with objective weights. *Computers & Operations Research, 27*(10), 963–973. doi:https://doi.org/10.1016/S0305-0548(99)00069-6

Diakoulaki, D., Mavrotas, G., & Papayannakis, L. (1995). Determining objective weights in multiple criteria problems: The critic method. *Computers & Operations Research, 22*(7), 763–770. doi:https://doi.org/10.1016/0305-0548(94)00059-H

Dixon, M. J., & Coles, S. G. (1997). Modelling Association Football Scores and Inefficiencies in the Football Betting Market. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 46*(2), 265–280.

Dixon, M. J., & Pope, P. F. (2004). The value of statistical forecasts in the UK association football betting market. *International Journal of Forecasting, 20*(4), 697–711.

Domingos, P. (1999). MetaCost: A General Method for Making Classifiers Cost-Sensitive. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 155–-164). New York, NY, USA: Association for Computing Machinery. doi:10.1145/312129.312220

Domingos, P., & Pazzani, M. (1997, November). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning, 29*(2), 103–130.

Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001a). Rank Aggregation Methods for the Web. *Proceedings of the 10th International Conference on World Wide Web* (pp. 613––622). New York, NY, USA: Association for Computing Machinery. doi:10.1145/371920.372165

Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001b). Rank aggregation revisited. Manuscript.

Efstathiou, A., Diamanti, Talattinis, K., & Stephanides, G. (2015). Extensions of LRMC model for ranking teams with interest on profit. *Proceedings of 4th International Symposium & 26th National Conference on Operational Research.* Chania.

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2* (pp. 973–-978). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Elliott, R. (Ed.). (2017). *The English Premier League: A Socio-Cultural Analysis* (1st ed.). London: Routledge.

Elo, A. E. (1978). *The rating of chessplayers, past and present.* Arco Pub.

EloRatings. (2023). *About World Football Elo Ratings.* World Football Elo Ratings. Retrieved May 2023, from https://www.eloratings.net/about

Epstein, E. S. (1969). A Scoring System for Probability Forecasts of Ranked Categories. *Journal of Applied Meteorology, 8*(6), 985–987.

European Gaming & Betting Association. (2022, December). European Online Gambling Key Figures 2022. Retrieved May 2023, from https://www.egba.eu/uploads/2022/12/221222-European-Online-Gambling-Key-Figures-2022.pdf

Farrell, M. J. (1957). The Measurement of Productive Efficiency. *Journal of the Royal Statistical Society. Series A (General), 120*(3), 253–290.

Favardin, P., Lepelley, D., & Serais, J. (2002, September 01). Borda rule, Copeland method and strategic manipulation. *Review of Economic Design, 7*(2), 213–228.

Fechner, G. T. (1860). *Elemente der psychophysik* (Vol. 2). Breitkopf u. Härtel.

Fechner, G. T. (1966). *Elements of Psychophysics: Transl. by Helmut E. Adler* (Vol. I). Holt, Rinehart and Winston.

Ferreira, F. G., Hanaoka, G. P., Paiva, F. D., & Cardoso, R. T. (2018). Parallel MOEAs for Combinatorial Multiobjective Optimization Model of Financial Portfolio Selection. *2018 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 1–8). doi:10.1109/CEC.2018.8477688

Fishburn, P. C. (1982). *The Foundations of Expected Utility.* Springer Netherlands.

Football-data. (2023). *Historical Football Results and Betting Odds Data.* Football Results, Statistic & Soccer Betting Odds Data. Retrieved May 2023, from https://www.football-data.co.uk/data.php

Forrest, D., Goddard, J., & Simmons, R. (2005). Odds-setters as forecasters: The case of English football. *International Journal of Forecasting, 21*(3), 551–564. doi:https://doi.org/10.1016/j.ijforecast.2005.03.003

Fowlkes, E. B., & Mallows, C. L. (1983). A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association, 78*(383), 553–569.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning, 29*(2), 131–163.

Gadallah, T. M., Fors, M. N., & Moneim, A. F. (2015). A new approach for combining multi-criteria trading decision models. *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, (pp. 1–6). doi:10.1109/IEOM.2015.7093802

Glickman, M. E. (1999). Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 48*(3), 377–394.

Goddard, J. (2005). Regression models for forecasting goals and match results in association football. *International Journal of Forecasting, 21*(2), 331–340. doi:https://doi.org/10.1016/j.ijforecast.2004.08.002

Goddard, J., & Asimakopoulos, I. (2004). Forecasting football results and the efficiency of fixed-odds betting. *Journal of Forecasting, 23*(1), 51–66. doi:https://doi.org/10.1002/for.877

Godin, F., Zuallaert, J., Vandersmissen, B., De Neve, W., & Van de Walle, R. (2014). Beating the Bookmakers: Leveraging Statistics and Twitter Microposts for Predicting Soccer Results. In *KDD Workshop on large-scale sports analytics.*

Gori, M., & Pucci, A. (2007). ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines. *Proceedings of the 20th International Joint Conference on Artifical Intelligence* (pp. 2766–2771). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Govan, A. Y. (2008). *Ranking Theory with Application to Popular Sports.* Ph.D. dissertation, North Carolina State University.

Govan, A. Y., Langville, A. N., & Meyer, C. D. (2009). Offense-defense approach to ranking team sports. *Journal of Quantitative Analysis in Sports, 5*(1).

Haaren, J. V., & Davis, J. (2015). Predicting the Final League Tables of Domestic Football Leagues. *Proceedings of the 5th International Conference on Mathematics in Sport*, (pp. 202–207). Loughborough.

Haaren, J. V., & den Broeck, G. V. (2014). Relational Learning for Football-Related Predictions. In *Latest Advances in Inductive Logic Programming* (pp. 237–244). doi:10.1142/9781783265091_0025

Haas, D. J. (2003). Technical Efficiency in the Major League Soccer. *Journal of Sports Economics, 4*(3), 203–215. doi:10.1177/1527002503252144

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl., 11*(1), 10-–18. doi:10.1145/1656274.1656278

Hand, D. J., & Yu, K. (2001). Idiot's Bayes—Not So Stupid After All? *International Statistical Review, 69*(3), 385–398. doi:https://doi.org/10.1111/j.1751-5823.2001.tb00465.x

Harper, F. M., & Konstan, J. A. (2015, December). The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst., 5*(4). doi:10.1145/2827872

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., . . . Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature, 585*(7825), 357–362. doi:10.1038/s41586-020-2649-2

Hastie, T., & Tibshirani, R. (1997). Classification by Pairwise Coupling. In M. Jordan, M. Kearns, & S. Solla (Ed.), *Advances in Neural Information Processing Systems. 10*. MIT Press.

Heidemann, J., Klier, M., & Probst, F. (2010). Identifying Key Users in Online Social Networks: A PageRank Based Approach. *ICIS 2010 Proceedings.*

Henni, K., Mezghani, N., & Gouin-Vallerand, C. (2018). Unsupervised graph-based feature selection via subspace and pagerank centrality. *Expert Systems with Applications, 114*, 46–53. doi:https://doi.org/10.1016/j.eswa.2018.07.029

Herbinet, C. (2018). Predicting football results using machine learning techniques. *Predicting football results using machine learning techniques*.

Herbrich, R., Minka, T., & Graepel, T. (2007, January). TrueSkill(TM): A Bayesian Skill Rating System. *Advances in Neural Information Processing Systems 20* (pp. 569–576). MIT Press. Retrieved from https://www.microsoft.com/en-us/research/publication/trueskilltm-a-bayesian-skill-rating-system/

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* University of Michigan Press.

Huang, S.-l. (2011). Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods. *Electronic Commerce Research and Applications, 10*(4), 398–407. doi:https://doi.org/10.1016/j.elerap.2010.11.003

Hug, N. (2020). Surprise: A Python library for recommender systems. *Journal of Open Source Software, 5*(52), 2174. doi:10.21105/joss.02174

Hvattum, L. M., & Arntzen, H. (2010). Using ELO ratings for match result prediction in association football. *International Journal of Forecasting, 26*(3), 460–470. doi:https://doi.org/10.1016/j.ijforecast.2009.10.002

Hwang, & Yoon, K. (1981). *Multiple Attribute Decision Making: Methods and Applications.* Heidelberg: Springer-Verlag.

IMDb. (2023). *How do you calculate the rank of movies and TV shows on the Top 250 Movies and Top 250 TV shows?* IMDb | Help. Retrieved May 2023, from https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV#calculatetop

Ingram, L. C. (2007). Ranking NCAA sports teams with Linear algebra. *Ranking NCAA sports teams with Linear algebra*. Charleston.

Jaekle, U., & Tomasini, E. (2019). *Trading Systems 2nd Edition: A new approach to system development and portfolio optimisation.* Harriman House.

Jahan, A., Mustapha, F., Sapuan, S. M., Ismail, M. Y., & Bahraminasab, M. (2012, January 01). A framework for weighting of criteria in ranking stage of material selection process. *The International Journal of Advanced Manufacturing Technology, 58*(1), 411–420. doi:10.1007/s00170-011-3366-7

Javari, A., & Jalili, M. (2014, December). Accurate and Novel Recommendations: An Algorithm Based on Popularity Forecasting. *ACM Trans. Intell. Syst. Technol., 5*(4). doi:10.1145/2668107

Jindra, M. (2005). The market for Internet domain names. *Proc. 16th ITS Regional Conf.*

Joblib Development Team. (2023). Joblib: running Python functions as pipeline jobs. Retrieved from https://joblib.readthedocs.io/

Jorion, P. (2006). *Value at Risk* (3 ed.). McGraw-Hill.

Karlis, D., & Ntzoufras, I. (2003). Analysis of sports data by using bivariate Poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician), 52*(3), 381–393.

Karlis, D., & Ntzoufras, I. (2008). Bayesian modelling of football outcomes: using the Skellam's distribution for the goal difference. *IMA Journal of Management Mathematics, 20*(2), 133–145. doi:10.1093/imaman/dpn026

Keener, J. P. (1993). The Perron-Frobenius theorem and the ranking of football teams. *SIAM Review, 35*(1), 80–93.

Keeney, R. L. (1971). Utility Independence and Preferences for Multiattributed Consequences. *Operations Research, 19*(4), 875–893.

Keeney, R. L. (1996). *Value-Focused Thinking: A Path to Creative Decisionmaking.* Harvard University Press.

Keeney, R., & Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Trade-Offs.* New York: Wiley.

Kelly, J. L. (1956). A new interpretation of information rate. *The Bell System Technical Journal, 35*(4), 917–926.

Kendall, M. G. (1938). A New Measure Of Rank Correlation. *Biometrika, 30*(1/2).

Kendall, M. G. (1945). The Treatment of Ties in Ranking Problems. *Biometrika, 33*(3), 239–251.

Kiani Mavi, R., Kiani Mavi, N., & Kiani, L. (2012). Ranking football teams with AHP and TOPSIS methods. *International Journal of Decision Sciences, Risk and Management, 4*(1-2), 108–126.

Kleinberg, J. M. (1999, September). Authoritative Sources in a Hyperlinked Environment. *J. ACM, 46*(5), 604-–632. doi:10.1145/324133.324140

Kohavi, R., & Provost, F. (1998). Glossary of Terms. *Special Issue on Applications of Machine Learning and the Knowledge Discovery Process, 30*(2-3), 271–274.

Koopman, S. J., & Lit, R. (2015). A dynamic bivariate Poisson model for analysing and forecasting match results in the English Premier League. *Journal of the Royal Statistical Society. Series A (Statistics in Society), 178*(1), 167–186.

Kvam, P., & Sokol, J. S. (2006). A logistic regression/Markov chain model for NCAA basketball. *Naval Research Logistics (NrL), 53*(8), 788–803.

Kyriakides, G., Talattinis, K., & Stephanides, G. (2014). Rating Systems Vs Machine Learning on the Context of Sports. *Proceedings of the 18th Panhellenic Conference on Informatics.* Athens: Association for Computing Machinery. doi:10.1145/2645791.2645846

Kyriakides, G., Talattinis, K., & Stephanides, G. (2015). Raw Rating Systems and Strategy Approaches to Sports Betting. *Proceedings of the 5th International Conference on Mathematics in Sport*, (pp. 97–102). Loughborough.

Kyriakides, G., Talattinis, K., & Stephanides, G. (2017). A Hybrid Approach to Predicting Sports Results and an AccuRATE Rating System. *International Journal of Applied and Computational Mathematics, 3*(1), 239–254. doi:10.1007/s40819-015-0103-1

Lago-Peñas, C., Lago-Ballesteros, J., Dellal, A., & Gómez, M. (2010). Game-related statistics that discriminated winning, drawing and losing teams from the Spanish soccer league. *Journal of sports science & medicine, 9*(2), 288–293.

Langville, A. N., & Meyer, C. D. (2012). *Who's# 1?: the science of rating and ranking.* Princeton University Press.

Lasek, J., Szlávik, Z., & Bhulai, S. (2013). The predictive power of ranking systems in association football. *International Journal of Applied Pattern Recognition, 1*(1), 27–46.

Lee, J., & Sabbaghi, N. (2020). Multi-objective optimization case study for algorithmic trading strategies in foreign exchange markets. *Digital Finance, 2*(1), 15–37.

Li, H. (2011). A short introduction to learning to rank. *IEICE TRANSACTIONS on Information and Systems, 94*(10), 1854–1862.

Liu, H., Hopkins, W. G., & Gómez, M.-A. (2016). Modelling relationships between match events and match outcome in elite football. *European Journal of Sport Science, 16*(5), 516–525. doi:10.1080/17461391.2015.1042527

Liu, T.-Y. (2009). Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval, 3*(3), 225–331. doi:10.1561/1500000016

London, A., & Csendes, T. (2013). HITS based network algorithm for evaluating the professional skills of wine tasters. *2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, (pp. 197–200). doi:10.1109/SACI.2013.6608966

Ma, J., Fan, Z.-P., & Huang, L.-H. (1999). A subjective and objective integrated approach to determine attribute weights. *European Journal of Operational Research, 112*(2), 397–404. doi:https://doi.org/10.1016/S0377-2217(98)00141-6

Maher, M. J. (1982). Modelling association football scores. *Statistica Neerlandica, 36*(3), 109–118. doi:https://doi.org/10.1111/j.1467-9574.1982.tb00782.x

Martel, J.-M., Khoury, N. T., & Bergeron, M. (1988). An Application of a Multicriteria Approach to Portfolio Comparisons. *Journal of the Operational Research Society, 39*(7), 617–628. doi:10.1057/jors.1988.107

Martin, P. G., & McCann, B. B. (1989). *The Investor's Guide to Fidelity Funds.* Wiley.

Massey, K. (1997). Statistical models applied to the rating of sports teams. *Statistical models applied to the rating of sports teams*.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure, 405*(2), 442–451.

McCarthy, K., Zabar, B., & Weiss, G. (2005). Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes? (pp. 69-–77). New York, NY, USA: Association for Computing Machinery. doi:10.1145/1089827.1089836

McHale, I. G., Scarf, P. A., & Folker, D. E. (2012). On the Development of a Soccer Player Performance Rating System for the English Premier League. *Interfaces, 42*(4), 339–351.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. (S. van der Walt, & J. Millman, Eds.) 56–61. doi:10.25080/Majora-92bf1922-00a

Mendonça, G. H., Ferreira, F. G., Cardoso, R. T., & Martins, F. V. (2020). Multi-attribute decision making applied to financial portfolio optimization problem. *Expert Systems with Applications, 158*, 113527. doi:https://doi.org/10.1016/j.eswa.2020.113527

Meyer, C. D. (2000). *Matrix analysis and Applied Linear Algebra.* Philadelphia: Siam.

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (Eds.). (1994). *Machine Learning, Neural and Statistical Classification.* Ellis Horwood.

Murphy, A. H. (1969). On the "Ranked Probability Score". *Journal of Applied Meteorology and Climatology, 8*(6), 988–989.

Murphy, A. H. (1971). A Note on the Ranked Probability Score. *Journal of Applied Meteorology and Climatology, 10*(1), 155–156.

Neumann, V. J., & Morgenstern, O. (1947). *Theory of games and economic behavior.* Princeton university press.

Nitzan, S. (1985, January 01). The vulnerability of point-voting schemes to preference variation and strategic manipulation. *Public Choice, 47*(2), 349–370.

Odachowski, K., & Grekow, J. (2013). Using Bookmaker Odds to Predict the Final Result of Football Matches. In M. Graña, C. Toro, R. J. Howlett, & L. C. Jain (Ed.), *Knowledge Engineering, Machine Learning and Lattice Computing with Applications* (pp. 196–205). Berlin: Springer Berlin Heidelberg.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999, November). *The PageRank Citation Ranking: Bringing Order to the Web.* Stanford InfoLab. Stanford InfoLab. Retrieved from http://ilpubs.stanford.edu:8090/422/

pandas development team, T. (2020, February). pandas-dev/pandas: Pandas. doi:10.5281/zenodo.3509134

Paradowski, B., Shekhovtsov, A., Bączkiewicz, A., Kizielewicz, B., & Sałabun, W. (2021). Similarity Analysis of Methods for Objective Determination of Weights in Multi-Criteria Decision Support Systems. *Symmetry, 13*(10). doi:10.3390/sym13101874

Pardo, R. (2008). *The evaluation and optimization of trading strategies* (2nd ed.). John Wiley & Sons.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Pelánek, R. (2016). Applications of the Elo rating system in adaptive educational systems. *Computers & Education, 98*, 169–179. doi:https://doi.org/10.1016/j.compedu.2016.03.017

Pieters, W., van der Ven, S. H., & Probst, C. W. (2012). A Move in the Security Measurement Stalemate: Elo-Style Ratings to Quantify Vulnerability. *Proceedings of the 2012 New Security Paradigms Workshop* (pp. 1-–14). New York, NY, USA: Association for Computing Machinery. doi:10.1145/2413296.2413298

Preis, T., Moat, H. S., & Stanley, H. E. (2013, April 25). Quantifying Trading Behavior in Financial Markets Using Google Trends. *Scientific Reports, 3*(1), 1684. doi:10.1038/srep01684

Quah, T.-S. (2008). DJIA stock selection assisted by neural network. *Expert Systems with Applications, 35*(1), 50–58. doi:https://doi.org/10.1016/j.eswa.2007.06.039

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Quinlan, J. R. (1993). *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc.

Rajabioun, R., & Rahimi-Kian, A. (2008, July). A genetic programming based stock price predictor together with mean-variance based sell/buy actions. *Proceedings of the World Congress on Engineering, 2*, pp. 1136–1141. London, U.K.

Ramanathan, R. (2004). Multicriteria Analysis of Energy. In C. J. Cleveland (Ed.), *Encyclopedia of Energy* (pp. 77–88). New York: Elsevier. doi:https://doi.org/10.1016/B0-12-176480-X/00240-0

Rampinini, E., Impellizzeri, F. M., Castagna, C., Coutts, A. J., & Wisløff, U. (2009). Technical performance during soccer matches of the Italian Serie A league: Effect of fatigue and competitive level. *Journal of Science and Medicine in Sport, 12*(1), 227–233. doi:https://doi.org/10.1016/j.jsams.2007.10.002

Rao, P. V., & Kupper, L. L. (1967). Ties in Paired-Comparison Experiments: A Generalization of the Bradley-Terry Model. *Journal of the American Statistical Association, 62*(317), 194–204.

Redmond, C. (2003). A Natural Generalization of the Win-Loss Rating System. *Mathematics Magazine, 76*(2), 119–126. doi:10.1080/0025570X.2003.11953163

Riabacke, M., Danielson, M., & Ekenberg, L. (2012, December). State-of-the-Art Prescriptive Criteria Weight Elicitation. *Advances in Decision Sciences, 2012*, 1–24. doi:10.1155/2012/276584

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence, 3*, pp. 41–46.

Rokach, L., & Maimon, O. (2005). Decision Trees. In O. Maimon, & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (pp. 165–192). Boston, MA: Springer US.

Ross, S. (2010). *A First Course in Probability* (8 ed.). Pearson Prentice Hall.

Roy, B. (1968). Classement et choix en présence de points de vue multiples. *Revue française d'informatique et de recherche opérationnelle. Série verte, 2*, 57–75.

Rue, H., & Salvesen, Ø. (2001). Focus on Sport: Prediction and Retrospective Analysis of Soccer Matches in a League. *Journal of the Royal Statistical Society Series D: The Statistician, 49*(3), 399–418. doi:10.1111/1467-9884.00243

Sen, P. C., Hajra, M., & Ghosh, M. (2020). Supervised Classification Algorithms in Machine Learning: A Survey and Review. In J. K. Mandal, & D. Bhattacharya (Ed.), *Emerging Technology in Modelling and Graphics* (pp. 99–111). Singapore: Springer Singapore.

Sevastjanov, P., & Dymova, L. (2009). Stock screening with use of multiple criteria decision making and optimization. *Omega, 37*(3), 659–671. doi:https://doi.org/10.1016/j.omega.2008.04.002

Sharpe, W. F. (1994). The Sharpe Ratio. *The Journal of Portfolio Management, 21*(1), 49–58.

Sheng, V. S., & Ling, C. X. (2009). Cost-sensitive learning. In J. Wang (Ed.), *Encyclopedia of Data Warehousing and Mining* (Second ed., pp. 339–345). IGI Global.

Sinuany-Stern, Z. (1988, July 01). Ranking of Sports Teams via the AHP. *Journal of the Operational Research Society, 39*(7), 661–667.

Sortino, F. A., & Price, L. N. (1994). Performance Measurement in a Downside Risk Framework. *The Journal of Investing, 3*(3), 59–64.

Spann, M., & Skiera, B. (2009). Sports forecasting: a comparison of the forecast accuracy of prediction markets, betting odds and tipsters. *Journal of Forecasting, 28*(1), 55–72. doi:https://doi.org/10.1002/for.1091

Stefani, R. T. (1999). A taxonomy of sports rating systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 29*(1), 116–120. doi:10.1109/3468.736367

Stefani, R., & Pollard, R. (2007). Football Rating Systems for Top-Level Competition: A Critical Survey. *Journal of Quantitative Analysis in Sports, 3*(3). doi:doi:10.2202/1559-0410.1071

Szczecinski, L., & Djebbi, A. (2020). Understanding draws in Elo rating algorithm. *Journal of Quantitative Analysis in Sports, 16*(3), 211–220.

Szwarc, A. (2004). Effectiveness of Brazilian and German teams and the teams defeated by them during the 17th Fifa World Cup. *Kinesiology, 36*(1), 83–89.

Tajirian, A. (2005, April). Valuing Domain Names: Methodology. Retrieved May 2023, from http://domainmart.com/news/methodology.htm

Tajirian, A. (2010, June). *Statistical Models for Market Approach to Domain Name Valuation.* Market Approach to Domain Name Valuation. Retrieved May 2023, from http://www.domainmart.com/news/Statistical_Models_for_Market_Approach_to_Domain_Name_Value.pdf

Talattinis, K., & Stephanides, G. (2022). RatingsLib: A python library for rating methods with applications. *Software Impacts, 14*, 100416. doi:https://doi.org/10.1016/j.simpa.2022.100416

Talattinis, K., Kyriakides, G., Kapantai, E., & Stephanides, G. (2019). Forecasting Soccer Outcome Using Cost-Sensitive Models Oriented to Investment Opportunities. *International Journal of Computer Science in Sport, 18*(1), 93–114. doi:10.2478/ijcss-2019-0006

Talattinis, K., Sidiropoulou, A., Chalkias, K., & Stephanides, G. (2010). Parallel Collection of Live Data Using Hadoop. *2010 14th Panhellenic Conference on Informatics*, (pp. 66–71). doi:10.1109/PCI.2010.47

Talattinis, K., Zervopoulou, C., & Stephanides, G. (2014, June). Ranking Domain Names Using Various Rating Methods. *Proceedings of the Ninth International Multi-Conference on Computing in the Global Information Technology* (pp. 107–114). Seville: IARIA.

Thurstone, L. L. (1927a). A law of comparative judgment. *Psychological Review, 34*(4), 273–286.

Thurstone, L. L. (1927b). Psychophysical Analysis. *The American Journal of Psychology, 38*(3), 368–389.

Thurstone, L. L. (1927c). The method of paired comparisons for social values. *The Journal of Abnormal and Social Psychology, 21*(4), 384–400.

Ting, K. M. (1998). Inducing cost-sensitive trees via instance weighting. In J. M. Żytkow, & M. Quafafou (Ed.), *Principles of Data Mining and Knowledge Discovery* (pp. 139–147). Berlin: Springer Berlin Heidelberg.

Trent, J. (2008, August). Domain Appraisal Guide - 20 Factors That Decide the Selling Price. Retrieved May 2023, from https://ezinearticles.com/?Domain-Appraisal-Guide—20-Factors-That-Decide-the-Selling-Price&id=1436181

UNDP. (1990). *A human development index.* New York: Oxford University Press.

Vaziri, B. (2016). *Markov-based ranking methods.* Ph.D. dissertation, Purdue University.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., . . . Contributors, S. 1. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods, 17*, 261–272. doi:10.1038/s41592-019-0686-2

von Winterfeldt, D., & Edwards, W. (1986). *Decision Analysis and Behavioral Research.* Cambridge University Press.

Wang, Y.-M., & Parkan, C. (2006). A general multiple attribute decision-making approach for integrating subjective preferences and objective information. *Fuzzy Sets and Systems, 157*(10), 1333–1345. doi:https://doi.org/10.1016/j.fss.2005.11.017

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Xu, J., Yao, Y., Tong, H., Tao, X., & Lu, J. (2017). RaPare: A Generic Strategy for Cold-Start Rating Prediction Problem. *IEEE Transactions on Knowledge and Data Engineering, 29*(6), 1296–1309. doi:10.1109/TKDE.2016.2615039

Yahoo-Finance. (2023). *Yahoo Finance–stock market live, quotes, business & finance news.* Yahoo! Finance. Retrieved May 2023, from https://finance.yahoo.com

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing, 415*, 295–316. doi:https://doi.org/10.1016/j.neucom.2020.07.061

Young, T. W. (1991). Calmar ratio: A Smoother Tool. *Futures, 20*(1), 40.

Zhang, C., Tan, K. C., & Ren, R. (2016). Training cost-sensitive Deep Belief Networks on imbalance data problems. *2016 International Joint Conference on Neural Networks (IJCNN)*, (pp. 4362–4367). doi:10.1109/IJCNN.2016.7727769