
Πρόγραμμα Μεταπτυχιακών Σπουδών

στην Αναλυτική των Επιχειρήσεων και Επιστήμη των Δεδομένων

Τμήμα Οργάνωσης και Διοίκησης Επιχειρήσεων

Διπλωματική Εργασία

# AI-Enabled Stock Prediction with Social Sensing, Technical Analysis and Forecasting Techniques

του

ΤΑΓΚΟΥΛΗ ΔΗΜΗΤΡΙΟΥ

Υποβλήθηκε ως απαιτούμενο για την απόκτηση του μεταπτυχιακού διπλώματος στην Αναλυτική των Επιχειρήσεων και Επιστήμη των Δεδομένων

Αύγουστος 2023

Acknowledgements,

I am deeply grateful to Professor Konstantinos Tarabanis and Assistant Professor Evangelos Kalampokis for their outstanding mentorship and unwavering support during my thesis project. Their extensive knowledge and expertise were invaluable to me and played a major role in the successful completion of my thesis. I am truly thankful for their guidance and help in overcoming various challenges. Their significant contributions to my work have inspired and motivated me.

## Abstract

The application of both Machine Learning (ML) and sentiment analysis from microblogging services has become a common approach for stock market prediction. In this thesis, we analyzed the stock movements of three companies, namely Amazon, Microsoft, Apple and Tesla using both historical and sentiment big data. Specifically, we collected 19,790,818 tweets from Twitter covering the period from 31-11-2018 to 31-12-2021. These tweets were collected with queries regarding either the company ticker or the company CEO. We also mined historical data from the Yahoo Finance website for the same period. The sentiment analysis of social media data was conducted using two specialized pre-trained models from Hugging Face: Twitter XLM-roBERTa and an alternative roBERTa model fine-tuned with data taken from Stocktwits. Also, multiple technical analysis indicators were created from historical data to aid with the final prediction. Finally, we used multiple forecasting algorithms to identify the best model to forecast the final prediction of price movement. We implemented multiple ML models, including KNN, SVM, Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and MLP. Our results indicate that when using tweets from Twitter with both sentiment models as the sentiment analysis tools, LGBM is the ML algorithm that gives the highest f-score of 62 % and an Area Under Curve (AUC) of 62%.

## Table of Contents

# Contents

## List of Tables



## List of Figures

# 1. Introduction

The stock market has become a fundamental part of a nation's economy because it provides a means for making investments and generating high capital (Billah et al., 2016a). It is a network of economic transactions that facilitates the purchasing and selling of stocks. An equity or share market reflects ownership claims on businesses, such as shares from public stock exchanges or individual transactions, such as selling private company shares to investors. The trading of stocks entails the transfer of funds from small individual investors to large merchants such as banks and corporations. Nevertheless, investing in the stock market is a high-risk endeavor due to its unpredictability (Gurjar et al., 2018). Successful stock market forecasting can be crucial for investors. Accurate forecasts can aid investors in making informed decisions about purchasing or selling shares. Numerous methods for predicting the stock market have been developed over the years, but they can be grouped into four categories: fundamental analysis based on published financial statements; technical analysis utilizing historical data and prices; ML methods applied to large amounts of data from multiple sources; and sentiment analysis utilizing published news, articles, or blogs (Huang et al., 2021a). The combination of the last two categories is much more recent than the others, and studies and research indicate that it has significant potential.

This thesis aims to predict stock movements for Microsoft, Apple, Amazon and Tesla by combining Technical Analysis, Time Series Forecasting, ML techniques, and Sentiment Analysis. Twitter data is used to conduct sentiment analysis, which provides insight into the emotions of individuals. According to the prevalent theory (Valle-Cruz et al., 2022), the public sentiment toward a company is favorable, its stock prices tend to rise, and vice versa. However, this theory is not always true when other economic factors are considered. Between November 30, 2018, and December 31, 2021, we deployed multiple ML techniques to 19,790,818 tweets extracted from Twitter. Using the open and close prices of the equities, we also mined historical data from Yahoo Finance and performed technical analysis and time series forecasting to enhance prediction performance.

# 2. Literature Review

## 2.1. Theoretical Background

There are a number of hypotheses for predicting stock market prices, but only two are well-known (Falinouss, 2007). First, there is the Efficient Market Hypothesis (EMH), followed by the Random Walk Theory (Burton G. Malkiel, 1973).

### 2.1.1. Efficient Market Hypothesis (EMH)

According to the Efficient Market Hypothesis (EMH), share prices should reflect all available information and market expectations. In other words, new information is promptly reflected in stock prices (Teknologi MARA et al., 2017). Thus, the current market price of a company is the most accurate indicator of its intrinsic value (Attigeri et al., 2015). When the market incorporates new information, the system immediately becomes unbalanced, and the new prices nullify the anticipated correct change. This information may be fundamental or non-fundamental (Fakhry, 2016). On the sovereign debt market, fundamental information includes yields or macroeconomic variables, whereas non-fundamental information originates from the news (Fakhry, 2016). The EMH theory is divided into three forms: Weak, Semi-Strong, and Strong (Ajekwe et al., 2017).

Feeble: In the feeble form of EMH, only historical data, such as prior prices, are factored into the current price.

Semi-Strong: The semi-strong form comprises all historical and current data, as well as all public information, such as profit and sales forecasts.

Strong: The strong form contains all public and private information, including intimate information about the stock price. Numerous research investigations frequently employ the feeble form (Ajekwe et al., 2017). According to this research experiments of weak form efficiency were conducted in Nigeria using monthly data from 1981 to 1992 on 59 arbitrarily selected stocks. It has been determined that the Nigerian market conforms to the feeble form when there are ten delays in return data. Concluded that the Nigerian stock market exhibits inefficient form efficiency.

### 2.1.2. Random Walk Theory

Random walk theory is the second prevalent theory for stock market prediction. According to this theory, variations in stock prices are independent over time, have the same distribution, and can be described by a random process such as flipping a coin (Ajekwe et al., 2017). Stock prices fluctuate when new information is released, and since information is released sporadically, stock prices fluctuate unpredictably. It is believed that it is impossible to predict stock prices if they are determined randomly (Falinouss, 2007). In other terms, prices fluctuate arbitrarily. This suggests to investors that the only way to outperform the market is by assuming additional risks. A market is considered efficient if prices respond rapidly and impartially to new information. It is important to note that the random walk theory and the semi-strong form of efficiency share the same foundation because all public information is accessible to everyone. According to Ajekwe, correlation experiments were conducted between July 1977 and July 1979 on the weekly prices of 21 selected Nigerian companies. It was determined that fluctuations in stock prices were not correlated and followed a random walk. Additional research analyzed the price behavior of 30 equities from 1977 to 1980, using Monday closing prices adjusted for cash dividends and script issues, to determine their price behavior. Stock prices were inferred to have followed a random walk (Ajekwe et al., 2017).

## 2.2. Related Work

According to the Dow Jones theory, market price fluctuations develop in trends (Picasso et al., 2019). As a result, researchers have introduced techniques for forecasting market trends and evaluating stocks, leading to the creation of two major types of stock market prediction methods: technical analysis and fundamental analysis (Picasso et al., 2019).

### 2.2.1. Technical Stock Prediction Analysis

The primary objectives of technical analysis are to evaluate investments, anticipate the thoughts of stakeholders, and identify purchasing or selling opportunities based on

historical price and volume data (Huang et al., 2021b). In other words, technical analysts attempt to identify patterns and predict future stock prices using stock prices and various mathematical indicators derived from historical stock prices and volume (Picasso et al., 2019). This data was extracted from graphs. Nonetheless, a significant drawback of infographics and time series data is that they only display the event and not its cause (Falinouss, 2007). Notably, technical analysis is more appropriate for short-term forecasting (Huang et al., 2021b). Technical analysis is founded on three fundamental tenets: market action diminishes everything, prices move in trends, and history repeats itself (Bohn & Ling, 2017).

Everything that can affect the price fundamentally, politically, or psychologically is discounted by market action, according to (Bohn & Ling, 2017). This means that historical values reflect all fundamental information influencing the price. Technicians concur that price fluctuations should influence demand and supply. For instance, if market prices are increasing, demand should outweigh supply, and fundamentals should be favorable (Bohn & Ling, 2017). The second hypothesis is that prices follow trends. This implies that price movements following a trend are more likely to continue in the same direction than to reverse (Bohn & Ling, 2017). The third premise is that the past is repeated. This premise asserts that the future is a repetition of the past and that price fluctuations are influenced by human emotions such as dread and optimism. Technical analysis uses chart patterns to analyze human emotions and comprehend stock market movements, enabling analysts to determine whether the market is bullish or adverse (Bohn & Ling, 2017).

Noting that the majority of existing studies on stock market prediction are founded on technical analysis is crucial. According to (Huang et al., 2021b), a study utilizing a feed-forward neural network for stock market prediction was conducted in 1990. As inputs for their predictive model, technical indicators and macroeconomic indices such as interest rates and foreign exchange rates were utilized. From January 1987 to September 1989, the model was tested for the existence of purchasing or selling signals for the TOPIX index. The study found that the neural network model's buy-and-hold strategy yields superior returns (Huang et al., 2021b). Using an ANFIS model and a Recurrent Neural Network (RNN), (Huang et al., 2021b) forecasted the trend of the NASDAQ and NIKKEI indices for the following day. Both models utilized the previous day's closing

price as a predictor for the following day's closing price. The training dataset contained information from 1971 to 1998, whereas the test dataset contained information from 1998 to 2002. The study concluded that the ANFIS model had a higher rate of return than the RNN model and the buy-and-hold strategy for both indices.

Noting that the majority of existing studies on stock market prediction are founded on technical analysis is crucial. According to (Huang et al., 2021b), a study utilizing a feed-forward neural network for stock market prediction was conducted in 1990. As inputs for their predictive model, technical indicators and macroeconomic indices such as interest rates and foreign exchange rates were utilized. From January 1987 to September 1989, the model was tested for the existence of purchasing or selling signals for the TOPIX index. The study found that the neural network model's buy-and-hold strategy yields superior returns(Huang et al., 2021b). Using an ANFIS model and a Recurrent Neural Network (RNN), forecasted the trend of the NASDAQ and NIKKEI indices for the following day. Both models utilized the previous day's closing price as a predictor for the following day's closing price. The training dataset contained information from 1971 to 1998, whereas the test dataset contained information from 1998 to 2002. The study concluded that the ANFIS model had a higher rate of return than the RNN model and the buy-and-hold strategy for both indices.

### 2.2.2. Fundamental Stock Prediction Analysis

Fundamental analysis uses financial data that companies are required to publish on a regular basis, such as financial status, annual reports, balance sheets, and income statements, to predict whether future stock prices will increase or decrease (Nti et al., 2019). This strategy seeks to investigate economic factors that may influence stock prices and establish a company's genuine value (Bohn & Ling, 2017). In particular, fundamental analysts analyze economic factors and stock price movements through the lens of three dimensions: the economy, the industry, and the company.

When the time horizon is a quarter, a year, or longer, fundamental analysis is more optimal for mid- and long-term stock market forecasting (Nti et al., 2019). Financial ratios are useful for comparing businesses of various sizes within the same industry. When evaluating the performance of a company, it is crucial to disregard its scale, as

real profit is a function of percentage price change rather than absolute price change (Bohn & Ling, 2017). Profitability ratios, liquidity ratios, debt ratios, asset utilization ratios, and market value ratios are the most significant financial ratios utilized in fundamental analysis (Bohn & Ling, 2017). Profitability ratios measure a company's ability to generate profit; liquidity ratios evaluate a company's ability to pay off its immediate debt obligations; debt ratios assess a firm's ability to pay off its debt liabilities over time; asset utilization ratios calculate the efficiency with which a company uses its assets; and market value ratios reflect the market value of a company's shares and the company as a whole.

Noting that just because a stock has superior ratios than a company does not necessarily mean that it should be purchased is essential. This is due to the fact that the market, industry, and sector may underperform even if the stock price fluctuations of a company are superior to those of comparable companies. The objective of fundamental analysis is to evaluate the price of a stock using publicly available financial ratios. (Huang et al., 2021b) devised a feed-forward neural network model using seven input attributes and financial ratios. Included were historical and projected PE ratios, market capitalization, EPS uncertainty, return on equity, cash flow yield, and a factor based on the weighted average of estimated historical values. The study collected data on 25 equities from the first quarter of 1993 to the fourth quarter of 1996, using the first ten observations for training and the remaining six for testing. The model was able to select portfolios that generated higher returns in 10 of 13 quarters. Due to insufficient data, the experiment could not be completed.

(Namdari & Li, 2018) conducted a study using a Multi-Layer Perceptron (MLP) neural network model and a hybrid model to predict stock market fluctuations using historical data and financial ratios. They chose 12 financial ratios from 578 NASDAQ-listed technology companies and gathered data from June 2012 to February 2017. Then, they created a second MLP model based on a technical analysis of historical data from the same companies. The objective was to evaluate the two models and select the one with the highest predictive accuracy for future stock movements. The MLP model based on fundamental analysis had a higher predictive accuracy (64.38%) than the MLP model based on technical analysis (62.82%), indicating that fundamental analysis is the superior technique for predicting future stock movements.

### 2.2.3. Machine Learning Applications in Stock Prediction

Various machine learning and data mining techniques have been utilized in recent years to forecast stock market movements. The sections that follow examine related work.

Various machine learning algorithms for stock market forecasting have been investigated and discovered that the Artificial Neural Network (ANN) was the most accurate and efficient algorithm (Deepak et al., 2017). The Support Vector Machine (SVM) algorithm was also implemented using the Radial Basis Function (RBF) kernel. The SVM algorithm was selected because it is believed to be the most suitable for predicting time series, including forecasting share prices. SVM is a supervised learning algorithm that uses a hyperplane to divide data into two classes. RBF is a form of feed-forward neural network that employs non-linear supervised learning based on the radial distance from a point.

The study group gathered data from the Yahoo Finance website between 2014 and 2016. They determined the input parameters using historical stock data and selected the open high, close high, and moving average values. Using SVM, they combined the feature profiles of four companies listed on the Bombay Stock Exchange (BSE) and calculated the accuracy, attaining up to 89%. In conclusion, the SVM algorithm played a significant role in the generation of custom features, and the RBF kernel contributed to a more precise outcome.

Overall, the study demonstrates the potential for machine learning algorithms like SVM and RBF to predict stock market movements accurately. By selecting input parameters with care and integrating multiple feature lists, it is possible to achieve impressive results in predicting future stock movements.

Another study focused on using ANN, SVM, and KNN to forecast stock prices one, five, and ten days in advance (Rasel et al., 2016). Their objective was to forecast the closing price of a stock using historical data from NYSE-listed Wal-Mart Stores Inc. from 2010 to 2015. Date, open price, close price, high price, and low price were the five

attributes included in the data set. The data were separated into training and evaluation sets. A windowing operator was utilized to transform time series data into generic data, and MAPE and RMSE were used to compute the error rate for the three models. According to the results, the 1-day-ahead model was the most accurate predictor, while the ANN model had the lowest error rate.

ANN's with backpropagation have also been used to forecast the stock prices of Indian companies (Gurjar et al., 2018). To achieve greater accuracy, they trained the model with historical stock data and extracted features such as the foreign exchange rate, NSE index, moving averages, and Relative Strength Index (RSI). In the model, they also incorporated variables such as moving averages, stochastic oscillators, standard deviation, and on-balance volume. Moving averages eliminate data noise and are based on prior values. For 1 day, 7 days, and 15 days, simple moving averages were used. The stochastic oscillator computes the difference between a stock's closing price and its price range over a period of time. Standard deviation measures the distance of a data set from its mean, whereas on-balance volume utilizes volume flow to predict stock price changes. The authors concluded that ANN outperformed linear regression and that the results could have been enhanced by incorporating additional NSE stocks.

(Choudhry & Garg, 2008) used GA to select the most important indicators as input features for the SVM, selecting a total of 35 features. They also correlated the stock prices of various companies to predict the price of a stock and found that TCS stocks were highly correlated with stocks of similar industries, such as Infosys. The prediction performance was estimated using the hit ratio, which is the percentage of times the prediction system was correct. The results showed that the GA-SVM hybrid model outperformed the standalone SVM model. For example, for Infosys, the hit ratio of GA-SVM was 60.3%, while the hit ratio of SVM was 56.7%. The use of correlation and GA were two important factors in improving the performance of the SVM model.

Another study was conducted in which they compared a system based on the Genetic Algorithm (GA) and SVM to a system based on SVM alone (Choudhry & Garg, 2008). The objective was to forecast the stock prices of Tata Consultancy Services, Infosys, and Reliance Industries Limited (RIL). 1386 trading days of data were obtained from the Yahoo Finance website between August 12, 2002 and January 18, 2008. The

characteristics utilized were the opening, greatest, lowest, and closing stock prices. Training (60%), validation (20%), and test (20%) data sets were created.

They selected a total of 35 input features for the SVM using GA to determine the 35 most significant indicators. In order to predict the price of a stock, they also correlated the stock prices of various companies and discovered that TCS stocks were highly correlated with stocks of similar industries, such as Infosys. The efficacy of the prediction system was measured using the strike ratio or the proportion of times the system was accurate. The results demonstrated that the GA-SVM hybrid model performed better than the SVM model on its own. For instance, the hit ratio of GA-SVM for Infosys was 60.3%, while the hit ratio of SVM was 56.6%. Correlation and genetic algorithms were significant variables in enhancing the performance of the SVM model.

Another approach proposed was using an enhanced Levenberg Marquardt (LM) algorithm of ANN to predict the closing prices of the stock market (Billah et al., 2016b). They gathered information from the Dhaka Stock Exchange (DSE) between January 2013 and April 2015. The historical data features selected included the daily opening price, closing price, highest price, lowest price, and total number of stocks traded. The data were preprocessed to eliminate disturbance and sanitize them.

They also implemented the Adaptive Neuro Fuzzy Inference System (ANFIS) and the conventional LM algorithm to compare the efficacy of their stock prediction with the improved LM algorithm. The accuracy and efficacy of the models were evaluated using Root Mean Square Error (RMSE) and the coefficient of multiple determinations (R2). RMSE values close to 0 indicate less error, whereas R2 values close to 1 indicate a stronger correlation. The results demonstrated that the enhanced LM algorithm was more effective than both RMSE and the conventional LM algorithm, with the highest R2 value and the lowest RMSE value, which was 53% less than the other methods. Time and memory requirements were reduced by 54% and 30%, respectively, compared to the traditional LM algorithm and 59% and 47%, respectively, compared to ANFIS. In terms of stock prediction accuracy, memory computation, and computing time, the enhanced LM algorithm outperformed both the traditional LM algorithm and ANFIS.

A mix of Neural Network, SVM, and Hidden Markov Model (HMM) to predict stock market prices was also employed in another study (Somani et al., 2014). They utilized a model to forecast the share prices of ICICI, SBI, and IDBI. The data was trained on the HMM model with the Baum-Welch algorithm, which employs Expectation-Maximization (EM) to optimize the HMM model's parameters. Afterwards, the data was evaluated using the Maximum a Posteriori (MAP) method. Mean Absolute Percentage Error (MAPE) is the average absolute error between actual and predicted stock prices. It was used to evaluate the performance of the model. ICICI, SBI, and IDBI had respective MAPE values of 2.1, 1.7, and 2.3, indicating excellent performance. However, it was discovered that increasing the amount of training data diminished efficacy.

(S. Liu et al., 2018) extracted feature values, analyzed stock data and predicted stock prices using a Long Short-Term Memory (LSTM) neural network model. LSTM is a type of recurrent neural network (RNN) that is ideally adapted for processing and predicting significant events in time series data with lengthy intervals and delays. Using historical transaction data from the JoinQuant platform for the CSI 300 Index from 2014-05-18 to 2017-01-29, the authors of their paper predicted short-term stock price changes using the LSTM algorithm. They selected features such as open, close, low, high, volume, money, limit_up, and limit_down values and calculated additional indexes such as Moving Average (MA), Exponential Moving Average (EMA), and various ratios including oc = (close – open) / open, oh = (high – open) / open, ol = (low – open) / open, ch = (high – close) / close, cl = (low – close) / close and lh = (high – low) / low.

The previously specified characteristics were used as training samples between 2014-05-18 and 2016-12-25, while the test samples were the closing data of CSI 300 between 2016-12-26 and 2017-01-29. The authors employed a stacked LSTM model with no more than three layers because more than five layers would necessitate additional computational resources. The experimental findings revealed that the accuracy of a single-layer LSTM model was 0.66, whereas the accuracy of a three-layer LSTM model exceeded 0.78. This suggests that the addition of layers improves prediction accuracy but at the expense of increased computational resources. (S. Liu et al., 2018) conclude that prediction performance could be enhanced by extracting additional feature values

for training the LSTM model. Using a three-layer LSTM model resulted in higher prediction performance than using a single-layer model, but adding more layers would require additional computational resources.

Two neural network models, LSTM and Deep Neural Network (DNN) were introduced in another study, for predicting daily and weekly stock price fluctuations of the Indian BSE Sensex index (Shah et al., 2018). As their dataset, they utilized historical Tech Mahindra stock data from 1997 to 2017. Both models were used to predict only the daily closing pricing of the stock, and their performance was compared using two metrics: RMSE and forecast bias. RMSE is an appropriate metric for analyzing predictions of time series, with an ideal value of zero. Forecast bias assesses the model's bias in comparison to the actual values, with a large positive value indicating that the model overestimates the actual data. DNN produced a lower RMSE value than LSTM, whereas LSTM generated a lower forecast bias value. In predicting stock prices, both models demonstrated a high level of accuracy.

In their study, (Shah et al., 2018) contrasted the efficacy of the LSTM and DNN models using a metric known as Directional Accuracy (DA). DA measures the correlation between the direction of each prediction and the direction of the actual data for a given period of time. The results demonstrated that the LSTM model performed better than the DNN model, which struggled to recognize rapid changes in time series data. To avoid overfitting, the authors ceased training the models when the training and validation loss function values stabilized. This assisted in generalizing the data and preventing overfitting. In addition to price data, the study could be enhanced by incorporating daily volume, volatility, and fundamental ratios.

(Patel et al., 2015) compared four models for predicting stock market movements and stock price indices using historical data from Indian stock markets: ANN, SVM, random forest, and Nave-Bayes. From January 2003 to December 2012, the data included CNX Nifty, S&P Bombay Stock Exchange (BSE) Sensex, Infosys Ltd., and Reliance Industries. The authors predicted using two distinct methods. Using stock trading data such as open, high, low, and close prices, the first method involved calculating ten technical parameters. These ten technical parameters were displayed as continuous values representing the actual time series and were utilized as predictor model inputs. The ten indicators included Simple Moving Average (SMA) and

Weighted Moving Average (WMA), stochastic oscillators such as STCK%, STCD%, and William R%, Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Commodity Channel Index (CCI), Accumulation/Distribution oscillator (A/D), and Momentum.

In their second method, (Patel et al., 2015) normalized the ten technical indicators to have values between +1 and -1, where +1 indicates an upward price movement and -1 indicates a downward price movement. These trend-based deterministic data served as the new input for the four models. The performance of the prediction was estimated by computing the precision and F-measure. The outcomes demonstrated that the models performed well when trained on continuous-valued inputs, but their performance improved even more when trained on trend-deterministic data. When trained on continuous-valued data, the Nave Bayes model performed the worst with an accuracy of 73.3%, while the Random Forest model performed the best with an accuracy of 83.56%. When trained on trend-deterministic data, the ANN model performed the worst with an accuracy of 86.6%, while the Nave-Bayes model performed the best with an accuracy of 90.1%. Future work proposed by the authors includes using macroeconomic variables such as inflation and interest rate and attaining long-term stock prediction.

Using historical data from January 2003 to December 2012, (Patel et al., 2015) predicted the future values of two stock market indices, CNX Nifty and S&P BSE Sensex. They utilized the same ten technical indicators as predictor model inputs as in their previous study and implemented two methodologies. The initial strategy consisted of a single-stage application of three models: ANN, Support Vector Regression (SVR), and Random Forest. The second strategy was a two-stage fusion strategy that utilized SVR-ANN, SVR-SVR, and SVR-RF. Experiments involving 1-10, 15, and 30-day advance forecasting were conducted for both approaches. Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), relative Root Mean Squared Error (rRMSE), and Mean Squared Error (MSE) were used to evaluate the efficacy of the two models. The results demonstrated that as the number of days ahead increased, so did the error values. The two-stage fusion strategy outperformed the single-stage strategy on nearly all prediction days, with the SVR-ANN model achieving the highest performance. In future work, the authors suggested incorporating company performance and government policy news into stock prediction models.

### 2.2.4. Sentiment Analysis and Stock Market

The influence of social media, which reflects public opinion on current events, is greater than ever before. Numerous studies have utilized Twitter and StockTwits to implement sentiment analysis within the discipline of trading strategy. This section will discuss past works that have incorporated Machine Learning techniques with sentiment analysis for stock market forecasting.

Another study analyzed the relationship between a company's stock market fluctuations and the sentiment of tweet texts using Twitter sentiment analysis and Machine Learning techniques (Pagolu et al., 2016). Using the Twitter API, they extracted two hundred and fifty thousand tweets from August 31, 2015, to August 25, 2016, that mentioned Microsoft. The authors used $MSFT, #Microsoft, and #Windows to filter the desired content and extract public sentiment regarding the company's stock as well as its products and services. They also obtained the opening and closing prices of Microsoft stock for the same time period from the Yahoo Finance website. Due to the fact that the stock market is closed on weekends and holidays, the authors substituted absent values with the average of opening and closing prices using a technique developed by Goel. The tweets were preprocessed in three steps: tokenization, removal of stop words, and regex matching to remove special characters. During tokenization, tweets were broken down into individual words to create a list of terms for each tweet. During stop word elimination, stop words such as "a", "is", "the", "with", etc. were removed. Special characters such as "#" were removed from identifiers, and URLs and user mentions were substituted with the words "URL" and "USER", respectively.

They extracted features using two textual representations: N-gram and word2vec. In the N-gram representation, tweets were divided into N-grams, and the features represented either a string of 1s or 0s, with 1 indicating the presence of N-grams in the tweets and 0 indicating their absence. In the word2vec representation, each word in the language was mapped to a unique vector, which was then summed to generate a 300-dimensional vector of all words in a tweet. This resultant vector represented the model's characteristics. The word2vec representation was selected for the model because it performs better with vast datasets. Using the characteristics of the word2vec

representation and the Random Forest algorithm, the tweets were categorized as positive, neutral, or negative. 70.2% accuracy was achieved with the word2vec representation, while 70.5% accuracy was achieved with N-grams. Although N-grams were marginally more accurate, word2vec was selected because it performs better with large datasets.

They also labelled Microsoft's stock price data. If the previous day's stock price was higher than the current day's stock price, the current day was labelled with a value of 0; otherwise, it was labelled with a value of 1. The authors used tweets and stock prices to train the algorithm and determine if sentiment and stock prices were correlated. The training set comprised 80% of the entire data, while the assessment set comprised the remaining 20%. Using a Logistic Regression algorithm, 69.01% accuracy was achieved. Using 90% of the data as the training set to train the model with the LibSVM algorithm yielded an accuracy of 71.82 percent. The results demonstrated that a large dataset performed well and that there was a strong correlation between stock market fluctuations and public tweet sentiment. The authors suggested using StockTwits data and a dataset with more than 10,000 tweets for future research.

Using a combination of StockTwits data and market data, (Batra & Daudpota, 2018) predicted Apple's stock price movement. The StockTwits data was extracted using an API from the StockTwits social networking website between 2010 and 2017. These attributes were retrieved: tweet id, user id, time, tweet text, retweets, and user sentiment (bullish or pessimistic) for that tweet. The market data for the same time period was obtained from the Yahoo Finance website and included open price, close price, low and high price volume, and adjusted close. The authors preprocessed the tweets by removing stop words, applying tokenization, and removing symbols, including @, #, URLs, extra spaces, and punctuation with the exception of $, which represents the company's ticker symbol. The market data was also preprocessed by substituting weekend-related absent values with the average of the previous and subsequent day's values. By subtracting today's closing price from yesterday's closing price, the authors also created an attribute that included the stock price decision. If the result was positive, the price would have risen, and the individual could now sell the stock.

In another study the researchers predicted public sentiment by categorizing tweets as either bearish or bullish (Batra & Daudpota, 2018). They used eighty percent of the data

for training and twenty percent for assessment. The accuracy attained for the training set was 91.2%, and for the assessment set it was 63.5%. Using the SVM algorithm, the authors predicted whether a person would purchase or sell a stock based on sentiment and market data. The final model attributes chosen were date, stock price decision, and sentiment. The accuracy of the training model was 75.22%, while the accuracy of the testing model was 76.68%. The results were satisfactory, but they could be enhanced by expanding the dataset.

Using SVM and Nave Bayes algorithms, (Kordonis et al., 2016) investigated the relationship between public sentiment and stock market values for 16 prominent technology companies, including Microsoft, Microsoft, Apple, and Blackberry. They extracted messages from Twitter using an API and selected features such as tweet id, timestamp, and 140-character tweet text. Each day's stock data was extracted from the Yahoo Finance API and consisted of open, close, high, and low values. The tweet data was tokenized, stop words and unnecessary Twitter symbols were removed, and N-grams representation was used for feature extraction. Using Pearson's chi-squared test, the authors evaluated each unigram, bigram, and trigram representation to select the most significant features for training the model. Using 7-fold cross-validation, they utilized SVM and Nave Bayes algorithms to predict the public sentiment of tweet texts, achieving an accuracy of 80.6% with Nave Bayes and 79.3% with SVM. The stock market data was also preprocessed by substituting the average of the previous and subsequent price values for absent values due to weekends and holidays. High-Low Percentage (HLPCT) and Percentage Change (PCT) are two additional metrics that were developed. PCT change is calculated as (Close-Open)/Open and HLPCT is calculated as (High-Low)/Low. Using these metrics, the correlation between tweets and the stock market was determined. The authors incorporated tweets and stock data, such as percentage positive, negative, and neutral sentiment scores, close price, HLPCT, PCT change, and volume.

Another study predicted future stock market trends with an accuracy of 87% using the SVM algorithm (Kordonis et al., 2016). Blackberry had the greatest inaccuracy at 6.29 percent, while all other tech companies had errors under 10 percent. The average prediction error for all companies was 1.668%, with nine of sixteen companies having prediction errors below 1%. The results demonstrated that public sentiment influenced

stock market prices, and the authors suggested utilizing a broader range of data from Twitter and the stock market to enhance precision.

The first to examine the relationship between tweets from Saudi Arabia and the Saudi market index where (AL-Rubaiee et al., 2015). An API was used to extract 3335 tweets for 53 days from the Saudi Arabian website of the Mubasher company. In the Gulf region, Mubasher is a software provider for asset analysis. The closing prices of the TASI index were also extracted from the website of the Mubasher company. "GET statuses/mentions_timeline" returned the twenty most recent mentions for authenticating the user, whereas "GET statutes/user_timeline" returned tweets recently posted by the screen_name or user_id parameters. Preprocessing the texts involved tokenizing them and removing stop words, suffixes, and prefixes.

ML algorithms including Naive Bayes, KNN, and SVM were used to categorize the tweets as positive, neutral, or negative following preprocessing. To evaluate the model, recall and precision values were computed. Using 10-folds cross-validation, Naive Bayes accuracy was 69.86%, SVM accuracy was 96.6%, and KNN accuracy was 96.45%. SVM had the highest recall at 95.71 percent, while KNN had the highest precision at 95.91 percent. The authors constructed a model demonstrating a one-to-one relationship between positive and negative sentiments and the TASI index's closing prices. A chart revealed that negative sentiment increased 24% of the time when the TASI index decreased. Positive sentiment rose 36% of the time when the TASI index increased. 40% of the time, however, a stable fluctuation was observed between positive sentiment, negative sentiment, and the TASI index. The results demonstrated a strong correlation between sentiment and the TASI index, and it is proposed that future research will predict the opening prices for the Saudi stock market (AL-Rubaiee et al., 2015).

(Mittal & Goel, 2012) examined the causal relationship between tweets and the values of the Dow Jones Industrial Average (DJIA). From June 2009 to December 2009, they gathered 476 million tweets posted by over 17 million users and extracted DJIA values from Yahoo Finance for the same period. Timestamp, username, and tweet text were selected as features for tweets, while open, close, high, and low values for a given day were selected as attributes for DJIA values. The stock values were preprocessed by substituting absent values with the average DJIA value for the given day and the

following day and eliminating periods of volatility. The tweets were divided into four categories: tranquil, cheerful, vigilant, and charitable. The authors devised their own analysis code to predict the sentiment of tweets.

The authors created a word list using the Profile of Mood States (POMS) questionnaire, which is a psychometric questionnaire in which respondents rate their current mood on a scale of 1 to 5. The responses correspond to six conventional POMS states, including anxiety, depression, wrath, vigor, and fatigue. They utilized a comparable strategy for N-grams representation. The sentiment-expressing Tweets were filtered, and a word-counting algorithm was applied to estimate the score for each POMS word for a given day. Using correlation criteria, each word's score was transferred to the six POMS moods and then restricted to the four mood states of the authors. For instance, happiness was the sum of vitality and the absence of depression.

The authors realized that they should have compared tweets from different days rather than comparing the value of one emotion to that of others. Granger Causality analysis, a metric indicating the amount of predictive information one attribute has about another for a given time period, was used to determine which mood value could be used to predict future stock movements. A lower p-value indicates greater predictive power. (Mittal & Goel, 2012) concluded that contentment and serenity were the most useful emotions for predicting future DJIA values when using data from the previous three or four days.

The authors trained and evaluated the model using four machine learning (ML) algorithms, namely Linear Regression, Logistic Regression, SVM, and Self Organizing Fuzzy Neural Networks (SOFNNs), after investigating the causality relationship between the previous three days' sentiments and the current day's stock prices. (Mittal & Goel, 2012) conducted six various configurations with the mood values from the previous three days to determine whether or not other mood states were dependent on DJIA.

# 3. Methodology

The collection of the corresponding dataset is a crucial aspect of stock movement prediction. One of the obstacles was gathering and combining the relevant data from multiple sources. It is also crucial to properly arrange the dataset before using it for model training and testing. This chapter describes how Python was used to acquire and prepare the data for measuring model performance. Python was used for data collection and preprocessing to ensure the dataset was appropriately prepared for use in model training and testing.

In this dissertation, we propose a method for predicting the stock market using historical and sentimental data. We trained our predictive model using actual stock data from Apple, Microsoft, and Tesla, three well-known technology companies with prominent CEOs. This thesis' contributions are summarized as follows:

1. As described in Chapter 3.3, we extracted data from Twitter and Yahoo Finance and performed preprocessing.

2. As described in section 3.3.1, we performed sentiment analysis on Twitter data using two pre-trained models from Hugging Face: Twitter XLM-roBERTa (Hugging Face, 2022). XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond (Barbieri et al., 2022) and an alternative roBERTa model refined with Stocktwits data (Y. Liu et al., 2019). Using historical stock data and Technical Analysis, we created new indicators and features.

3. As described in section 4, we trained and evaluated our model using multiple ML algorithms, and in section 4.1.2, we evaluated multiple ML algorithms for stock prediction.

## 3.1. Research Questions (RQs)

This study aims to investigate the state of the art in stock market forecasting using sentiment analysis. To acquire a thorough comprehension of the subject, we concentrated on the following research questions:

(Q1) What are the most prominent stock market prediction theories?

(Q2) What are the conventional methods for stock market forecasting, and in what circumstances have they been utilized?

(Q3) What considerations must be made when designing and constructing a stock market prediction model?

(Q4) What are the best predictors for stock market prediction using sentiment analysis and technical indicators, and what factors contribute to the success of ML applications in this context?

## 3.2. Aim and Objectives

The objective of the first research question is to provide an overview of stock market prediction theories, such as the efficient market hypothesis and the random walk theory. The second query examines traditional approaches, such as technical and fundamental analysis, with an emphasis on case studies in which these approaches were used to predict the stock market. The third question focuses on case studies in which multiple ML techniques have been used to predict stock prices, as well as cases in which both ML techniques and sentiment analysis have been used to predict stock price movements. The final research question concentrates on the evolution of our stock market prediction model, specifically its performance and predictions of the stock movements of Microsoft, Apple, and Tesla.

## 3.3. Data Collection

For forecasting the stock price movements of companies from August 31, 2018, to December 31, 2021, financial data from Yahoo Finance and social media data from Twitter were collected. Using Yahoo Finance, historical stock price information was also compiled for the same time period.

### 3.3.1. Twitter Data Harvesting for Sentiment Analysis

Twitter is a well-known social media platform with over 200 million monthly active users. It has become a valuable source of information for comprehending the opinions of individuals regarding brands, products, and more. We obtained Twitter data for this project in order to perform sentiment analysis on people's opinions of Microsoft and its

products. We created a research account on Twitter and connected to the Twitter API using the appropriate credentials. We were able to gather tweets by authenticating with Twitter using the searchtweets library and establishing the access token and access token secret. The tweets were compiled using two inquiries per company: the ticker symbol ("AAPL", "AMZN", "TSLA", "MSFT") and the CEO's name ("Tim Cook", "Jeff Bezos", "Elon Musk", "Satya Nadella"). Due to the researcher account's enhanced privileges, we were able to extract all tweets for each query for each day. We extracted information such as the keyword, user id, user account, creation date, and tweet text from the English tweets. Additionally we validated that no duplicate tweets were gathered through the query and by manually validating that no duplicate tweet ids exist. Each day's and query's results were saved as pickle files to facilitate high parallelism in the preprocessing phase.

### 3.3.2. Financial and Stock Market Data Procurement

The Yahoo! finance website, which provides a plethora of international market data, news, stock quotations, and portfolio resources, was mined for historical stock data for all companies. We gathered information such as the closing price, the opening price, the low and high price, the volume, and the adjusted price. During the preprocessing phase, these data are utilized to generate numerous technical analysis features.

## 3.4. Data Wrangling

The data preparation process began with Data Wrangling, where Twitter data was analyzed using sentiment models from Hugging Face Hub to gauge opinions on companies and CEOs. In Feature Engineering, datasets were restructured for usability, enriched with stock and time series data, and enhanced with technical indicators. Dimensionality Reduction was then applied using Scikit-learn's SelectFromModel to focus on the top 20% of impactful features. The final phase, Post Feature Engineering Pre-processing, utilized SMOTE for dataset balance and the Yeo-Jonson method to normalize the data distribution, ensuring a well-prepared dataset for subsequent analysis and modeling.

### 3.4.1. Data Pre-Processing

After accumulating the data, we performed sentiment analysis on the Twitter data to identify people's opinions regarding the companies and their CEOs. We utilized two published sentiment analysis models from Hugging Face Hub: Twitter XLM-roBERTa (Barbieri et al., 2022) and an alternative roBERTa (Gitrexx, 2022) model refined with Stock twits' data. After cleansing and assessing each tweet with pre-trained models, we used Python Technical Analysis library's automated feature creation to generate technical analysis features. Then, all features were delayed in time to prevent data loss during training and validation. Finally, time series features such as weekday, monthday, and yearday were created to represent the time series properties of each company. Following this stage, all data were merged to form the final dataset, which would be used for feature selection and modeling.

### 3.4.2. Feature Engineering

Feature engineering was conducted for two separate categories of features, Sentiment Analysis ones and Technical Analysis ones. In the Sentiment Analysis Features phase, two Tweet datasets are loaded and refined by addressing column naming conventions, extracting date elements, creating structured query columns, and organizing the data around daily pivots and company-specific attributes. Advanced statistics functions optimize feature creation, and after ensuring data integrity and merging the datasets, missing values are addressed. For clarity, certain columns are renamed and non-relevant ones dropped. Meanwhile, the Technical Analysis Features segment begins with acquiring stock and time series data. Custom functions create vital columns, such as company names and price changes, and additional technical indicators, like momentum and volatility, are added using the "ta" library. Subsequent steps involve merging this data with Tweet features after renaming and refining the dataset to include only post-January 1, 2019, dates. Date-related features are then extracted for further analysis. Furthermore, all these features undergo a transformation to create time-lagged features, enabling the capturing of autocorrelation, partial autocorrelation, and seasonality effects

within the data while also preventing overfitting (Januschowski et al., 2022; Petropoulos et al., 2022a).

#### 3.4.2.1. Sentiment Analysis Features

The first step involves loading the two Tweet datasets into the system. After loading, the column names containing spaces are modified by replacing the spaces with underscores to ensure data consistency. To make the date column usable, the slice function is applied to extract the dates from it. Similarly, the split function is applied to create query columns (CEO / Ticker) from specific data entries. Next, pivots are generated for each day, providing a summarized view of data for each sentiment scorer, Company and CEO. Additionally, company-specific columns are created to organize the data further. Moving statistics functions are then utilized to identify relevant columns, and these functions are executed to create the desired features. Ensuring data integrity, a check is conducted to confirm that the indexes of both datasets are of the same size. Subsequently, the two datasets are merged together. To handle missing values, appropriate filling methods are applied. Finally, non-query columns are dropped, and the "Created_at" column is renamed to "Date" for clarity and consistency in the dataset.

#### 3.4.2.2. Technical Analysis Features

In the first step, the stock and time series (TS) data is obtained. The custom function "get_stock_data" is then employed to create several essential columns, including "Company" representing the name of the company, "Price Change" indicating the change in the adjusted closing price from the previous day, "Percent Change" denoting the percentage change in the adjusted closing price from the previous day, and "Movement" classifying whether the price went "Up," "Down," or remained approximately the same compared to the previous day.

Additionally, the dataframe is enriched with various technical analysis features calculated using the "add_all_ta_features" function sourced from the "ta" library like Momentum, Volume, Volatility, Trend and Other indicators (Buko Sabino, 2023).

Next, the dataframes are combined, and the index is removed. The dataframe is then renamed to "Date" to facilitate merging with the Tweet features. To refine the data further, only the dates after January 1, 2019, are retained.

Finally, the date features are extracted from the dataframe, potentially encompassing information like day, month, year, weekday, etc., for further analysis and modeling.

## 3.5. Exploratory Data Analysis

In this section of our stock price forecasting paper, we embark on a comprehensive Exploratory Data Analysis (EDA) to gain deeper insights into the underlying dynamics of our dataset. This EDA section is structured into two distinct subsections, each focusing on a crucial facet of our analysis. The first subsection provides a detailed exploration of the key features pertinent to each company, offering a comprehensive overview through descriptive statistics. The second subsection delves into the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots for the Price Change variable, offering a critical examination of temporal dependencies within each company's stock price data. Together, these analyses form the foundation upon which our stock price forecasting models are built, enhancing our understanding and predictive capabilities within the financial domain.

### 3.5.1. Twitter Features Descriptive Statistics

For Amazon (Table 1) the table encapsulates essential insights regarding Price Change, Sentiment, and Signals from a dataset comprising 1095 entries. The "Price Change" column reveals an average price change of 0.031892, with a notable standard deviation of 1.854138, indicating significant variability, spanning from a minimum of -7.92208 to a maximum of 7.92952. The sentiment metrics for stocks, denoted by "PWD Tickers Sentiment Positive" and "PWD Tickers Sentiment Negative," reflect an average of 479.41 positive and 185.97 negative sentiment entries, displaying considerable variability. Correspondingly, the "PWD Tickers Signal Bullish" and "PWD Tickers Signal Bearish" columns illustrate an average of 1414.84 bullish and 430.96 bearish

signals, mirroring sentiment's variability. In CEO-related sentiment, the "PWD Ceos Sentiment Positive" and "PWD Ceos Sentiment Negative" columns exhibit an average of 97.95 positive and 166.61 negative sentiment entries, reflecting standard deviations signifying variability. Similarly, the "PWD Ceos Signal Bullish" and "PWD Ceos Signal Bearish" columns showcase an average of 391.79 bullish and 192.17 bearish signals, emphasizing variability. Overall, this table provides a comprehensive overview of key metrics, underlining the need for a nuanced analysis due to the substantial variability within these data points.

| | Price Change | PWD Tickers Sentiment Positive | PWD Tickers Sentiment Negative | PWD Tickers Signal Bullish | PWD Tickers Signal Bearish | PWD Ceos Sentiment Positive | PWD Ceos Sentiment Negative | PWD Ceos Signal Bullish | PWD Ceos Signal Bearish |
|---|---|---|---|---|---|---|---|---|---|
| count | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 |
| mean | 0,031892 | 479,4064 | 185,968 | 1414,835 | 430,958 | 97,95342 | 166,6082 | 391,7936 | 192,1699 |
| std | 1,854138 | 768,4837 | 220,7291 | 2006,064 | 465,7548 | 147,6141 | 311,9122 | 590,856 | 343,0307 |
| min | -7,92208 | 33 | 3 | 118 | 17 | 9 | 6 | 41 | 13 |
| 25% | -0,88462 | 129 | 43 | 431 | 108,5 | 35 | 39,5 | 115 | 48 |
| 50% | 0,018414 | 217 | 112 | 693 | 261 | 63 | 87 | 252 | 110 |
| 75% | 0,910349 | 373,5 | 251 | 1320,5 | 568 | 106 | 194,5 | 444 | 217 |
| max | 7,92952 | 4375 | 2038 | 9050 | 3071 | 1946 | 4928 | 6767 | 4779 |

*Table 1 AMAZON Tweet Descriptive Statistics*

For APPLE (Table 2) notably, the "Price Change" column demonstrates an average change of 0.181661, marked by a standard deviation of 2.12657, signifying significant variability across the dataset, ranging from -12.8647 to 11.98082. For stock sentiment, as indicated by "PWD Tickers Sentiment Positive" and "PWD Tickers Sentiment Negative," the table reveals an average of 1294.28 positive and 246.96 negative sentiment entries, accompanied by considerable standard deviations. Similarly, "PWD Tickers Signal Bullish" and "PWD Tickers Signal Bearish" columns highlight an average of 3499.09 bullish and 765.45 bearish signals, echoing the dataset's variability. In CEO-related sentiment, "PWD Ceos Sentiment Positive" and "PWD Ceos Sentiment Negative" columns showcase an average of 14.83 positive and 4.32 negative sentiment entries, marked by standard deviations underlining variability. The "PWD Ceos Signal Bullish" and "PWD Ceos Signal Bearish" columns depict an average of 37.76 bullish

and 8.97 bearish signals, emphasizing variability within CEO sentiment metrics. Collectively, this table underscores the importance of a nuanced analysis due to the substantial variability present in these metrics.

| | Price Change | PWD Tickers Sentiment Positive | PWD Tickers Sentiment Negative | PWD Tickers Signal Bullish | PWD Tickers Signal Bearish | PWD Ceos Sentiment Positive | PWD Ceos Sentiment Negative | PWD Ceos Signal Bullish | PWD Ceos Signal Bearish |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 |
| **mean** | 0,181661 | 1294,281 | 246,9553 | 3499,093 | 765,4548 | 14,83014 | 4,322374 | 37,75708 | 8,968037 |
| **std** | 2,12657 | 1419,494 | 225,1771 | 3698,611 | 738,7237 | 23,11545 | 32,30249 | 64,86319 | 32,92696 |
| **min** | -12,8647 | 41 | 2 | 123 | 17 | 0 | 0 | 1 | 0 |
| **25%** | -0,76254 | 175,5 | 39,5 | 503,5 | 107 | 5 | 0 | 13 | 1 |
| **50%** | 0,117487 | 314 | 213 | 980 | 363 | 9 | 1 | 21 | 3 |
| **75%** | 1,198894 | 2987 | 409 | 8351 | 1606,5 | 15 | 2 | 38 | 7 |
| **max** | 11,98082 | 7631 | 1673 | 9622 | 2720 | 310 | 955 | 1328 | 829 |

*Table 2 APPLE Tweet Descriptive Statistics*

Bellow table 3 provides a comprehensive overview of critical metrics pertaining to Microsoft, encompassing Price Change, Sentiment, and Signals, based on a dataset of 1095 entries. Notably, the "Price Change" column indicates an average change of 0.208, marked by a standard deviation of 1.859, signifying considerable variability across the dataset, ranging from -14.739 to 14.217. For sentiment metrics related to Microsoft's stock, represented by "PWD Tickers Sentiment Positive" and "PWD Tickers Sentiment Negative," the table displays an average of 1120.42 positive sentiment entries and 277.82 negative sentiment entries, both exhibiting noteworthy variability with significant standard deviations. Correspondingly, the "PWD Tickers Signal Bullish" and "PWD Tickers Signal Bearish" columns highlight an average of 3140.34 bullish and 781.49 bearish signals, further echoing the dataset's variability.

In the context of CEO-related sentiment associated with Microsoft, the "PWD Ceos Sentiment Positive" and "PWD Ceos Sentiment Negative" columns reflect an average of 388.34 positive sentiment entries and 507.83 negative sentiment entries, accompanied by standard deviations emphasizing variability. The "PWD Ceos Signal Bullish" and "PWD Ceos Signal Bearish" columns reveal an average of 663.83 bullish signals and

1392.33 bearish signals, once again highlighting variability within CEO sentiment metrics.

In summary, this table provides a comprehensive perspective on key metrics associated with Microsoft, emphasizing substantial variability throughout the dataset. The broad range of values underscores the need for a nuanced analysis to interpret the data comprehensively, particularly in the context of Microsoft's stock and CEO-related sentiment and signals.

| | Price Change | PWD Tickers Sentiment Positive | PWD Tickers Sentiment Negative | PWD Tickers Signal Bullish | PWD Tickers Signal Bearish | PWD Ceos Sentiment Positive | PWD Ceos Sentiment Negative | PWD Ceos Signal Bullish | PWD Ceos Signal Bearish |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 |
| **mean** | 0,208 | 1120,416 | 277,8228 | 3140,344 | 781,4886 | 388,3352 | 507,8338 | 1392,332 | 663,8292 |
| **std** | 1,858792 | 1324,388 | 230,1372 | 3394,872 | 656,1363 | 425,2592 | 596,0926 | 1161,531 | 720,3654 |
| **min** | -14,739 | 35 | 3 | 155 | 26 | 73 | 66 | 291 | 98 |
| **25%** | -0,59343 | 158 | 104 | 580 | 236 | 177,5 | 199,5 | 681,5 | 266,5 |
| **50%** | 0,160584 | 338 | 224 | 1237 | 520 | 267 | 327 | 1037 | 426 |
| **75%** | 1,071876 | 2659,5 | 407 | 8204 | 1525 | 450,5 | 560,5 | 1649,5 | 760,5 |
| **max** | 14,2169 | 7797 | 2510 | 9628 | 3706 | 6324 | 5959 | 9231 | 6504 |

*Table 3 MICROSOFT Tweet Descriptive Statistics*

Finally, table 4 offers a comprehensive overview of key metrics encompassing Price Change, Sentiment, and Signals, based on a dataset comprising 1095 entries. The "Price Change" column reveals an average price change of 0.331813731, characterized by a standard deviation of 3.977593233, indicating substantial variability across the dataset, ranging from -21.06282432 to 19.89485938. Regarding sentiment metrics for stocks, denoted by "PWD Tickers Sentiment Positive" and "PWD Tickers Sentiment Negative," the table displays an average of 1155.95 positive sentiment entries and 194.34 negative sentiment entries, both demonstrating significant variability with substantial standard deviations. Correspondingly, the "PWD Tickers Signal Bullish" and "PWD Tickers Signal Bearish" columns highlight an average of 3198.57 bullish and 704.23 bearish signals, further mirroring the dataset's variability.

In the context of CEO-related sentiment associated with this dataset, the "PWD Ceos Sentiment Positive" and "PWD Ceos Sentiment Negative" columns reflect an average of

97.87 positive sentiment entries and 165.46 negative sentiment entries, accompanied by standard deviations emphasizing variability. The "PWD Ceos Signal Bullish" and "PWD Ceos Signal Bearish" columns reveal an average of 389.89 bullish signals and 189.30 bearish signals, once again highlighting variability within CEO sentiment metrics.

In summary, this table provides a comprehensive perspective on key metrics, underlining substantial variability within the dataset. The wide range of values underscores the need for a nuanced analysis when interpreting the data comprehensively, particularly concerning sentiment and signals related to both stocks and CEOs.

| | Price Change | PWD Tickers Sentiment Positive | PWD Tickers Sentiment Negative | PWD Tickers Signal Bullish | PWD Tickers Signal Bearish | PWD Ceos Sentiment Positive | PWD Ceos Sentiment Negative | PWD Ceos Signal Bullish | PWD Ceos Signal Bearish |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 | 1095 |
| **mean** | 0,331814 | 1155,952 | 194,3352 | 3198,571 | 704,2283 | 97,87489 | 165,4612 | 389,8886 | 189,3032 |
| **std** | 3,977593 | 1378,852 | 195,5373 | 3645,338 | 720,6471 | 163,9263 | 284,22 | 564,0194 | 331,3629 |
| **min** | -21,0628 | 39 | 3 | 129 | 20 | 5 | 10 | 36 | 12 |
| **25%** | -1,36499 | 158,5 | 37 | 489 | 94 | 35 | 39,5 | 117 | 48 |
| **50%** | 0,200535 | 261 | 101 | 844 | 287 | 65 | 90 | 248 | 108 |
| **75%** | 2,048563 | 2794 | 347 | 8298,5 | 1572,5 | 103,5 | 197 | 434 | 218 |
| **max** | 19,89486 | 4091 | 1550 | 9008 | 2441 | 3016 | 3936 | 6585 | 5546 |

*Table 4 TESLA Tweet Descriptive Statistics*

## 3.5.2. Price Change ACF and PACF Plots

In this section, we turn our attention to the crucial task of understanding the temporal dependencies within our stock price data. To achieve this, we employ Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots, powerful tools in time series analysis. These plots provide valuable insights into the autocorrelation patterns present in our Price Change variable. By examining the ACF and PACF plots, we aim to uncover the underlying dynamics of stock price changes, paving the way for more accurate and informed stock price forecasting models.

For AMAZON (Fig 1) the plots reveal that stock price changes exhibit significant correlation at lag 1, indicating an Autoregressive (AR) structure of order 1. This means that current price changes are linked to their immediate past values, a valuable insight for predictive modeling. Additionally, the ACF plot highlights a slight negative correlation at lag 1, suggesting a tendency for price changes to reverse direction. The PACF plot shows a mild positive correlation at lag 2, indicating persistence in price changes' direction over two periods.

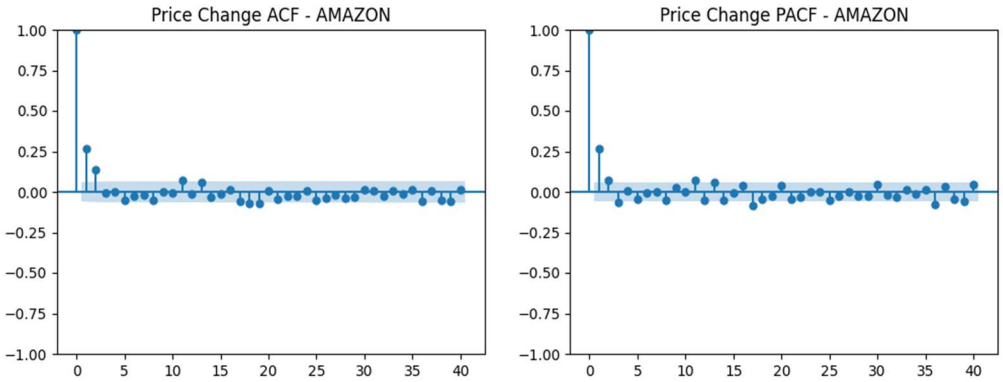



*Figure 1 AMAZON Price Change ACF and PACF Plots*

For APPLE (Fig.2) The ACF plot reveals a significant correlation at lag 1, indicative of an autoregressive (AR) structure of order 1, with no notable correlations at higher lags. Similarly, the PACF plot affirms this AR(1) pattern, showcasing a significant correlation at lag 1 and negligible correlations at subsequent lags.

In summary, the daily return of Apple stock adheres to an AR(1) process, signifying that today's return is correlated with yesterday's return. This valuable insight forms the basis for constructing time series models aimed at forecasting future daily returns of Apple stock. Further examination of the ACF plot reveals a slight negative correlation at lag 1, suggesting a tendency for the daily return to reverse direction from the previous day. Conversely, the PACF plot exhibits a slight positive correlation at lag 2, indicating persistence in the same direction as the daily return two days prior. These nuanced findings offer additional context for refining predictive models and enhancing our understanding of Apple stock's price dynamics.

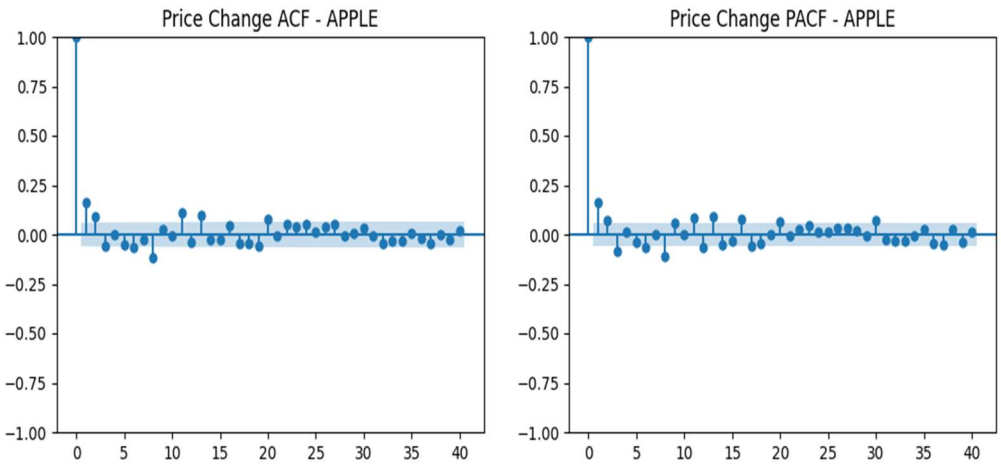



*Figure 2 APPLE Price Change ACF and PACF Plots*

For MICROSOFT (Fig 3) in the ACF plot, a significant correlation is observed at lag 1, indicating an autoregressive (AR) structure of order 1, with no significant correlations at higher lags. Likewise, the PACF plot reaffirms this AR(1) pattern, displaying a significant correlation at lag 1 and negligible correlations at subsequent lags.

Microsoft stock price changes exhibit an AR(1) process, signifying that the current change is correlated with the previous change. This valuable insight forms the foundation for constructing time series models aimed at forecasting future Microsoft stock price changes.

Furthermore, a nuanced examination of the ACF plot reveals a slight negative correlation at lag 1, suggesting a tendency for the Microsoft stock price change to reverse direction from the previous change. Conversely, the PACF plot displays a slight positive correlation at lag 2, indicating persistence in the same direction as the Microsoft stock price change from two periods ago. These additional findings offer valuable context for refining predictive models and gaining a deeper understanding of Microsoft stock's price dynamics.

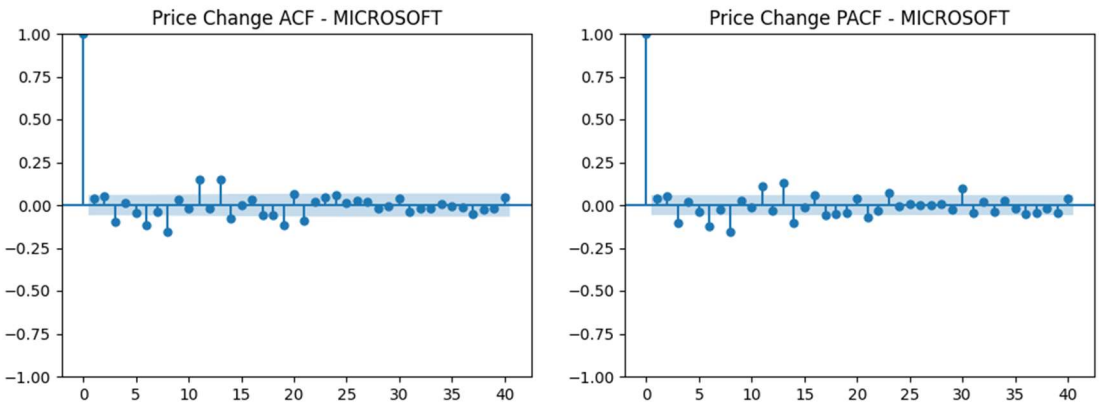



*Figure 3 MICROSOFT Price Change ACF and PACF Plots*

Finally, for TESLA (Fig 4) the ACF plot reveals a significant correlation at lag 1, indicative of an autoregressive (AR) structure of order 1, with no significant correlations at higher lags. Similarly, the PACF plot reaffirms this AR(1) pattern, displaying a significant correlation at lag 1 and negligible correlations at subsequent lags.

In conclusion, the Tesla stock price change follows an AR(1) process, signifying that the current change is correlated with the previous change. This valuable insight provides the basis for constructing time series models for forecasting future Tesla stock price changes.

Furthermore, a nuanced examination of the ACF plot reveals a slight negative correlation at lag 1, suggesting a tendency for the Tesla stock price change to reverse direction from the previous change. Conversely, the PACF plot shows a slight positive correlation at lag 2, indicating persistence in the same direction as the Tesla stock price change from two periods ago. These additional findings offer valuable context for refining predictive models and gaining a deeper understanding of Tesla stock's price dynamics.

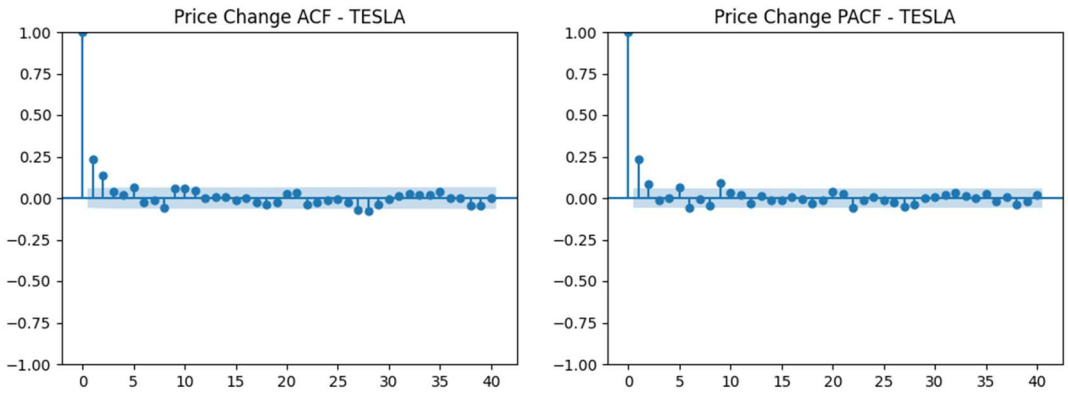



*Figure 4 TESLA Price Change ACF and PACF Plots*

## 3.6. Dimensionality Reduction

The large dimensionality of the combined dataset can potentially be reduced down to a level that is more manageable with the use of the SelectFromModel function that is included right into SkLearn. The SelectFromModel meta-transformer in Scikit-learn is a tool that can be utilized for the purpose of selecting features, and it is a tool that can be employed for the purpose of picking features. It accomplishes this objective by choosing attributes in accordance with the significance weights that a particular estimator (in this example, LGBM) has allocated to each of those attributes. It was decided that a cutoff value of 0.2 would be used for the process of selecting features. This ensures that just the 20% of all of the traits that are considered to be the most useful would be retained.

## 3.7. Post Feature Engineering Pre-processing

In the post-feature engineering pre-processing step, two methods are used: SMOTE to correct the disparity of the dataset and power transformation in all numeric columns to make the data more Gaussian. Similar to the Box-Cox transformation (Sakia, 1992) for the power transformation, the Yeo-Jonson method can manage both positive and negative data. The transformation creates a more uniform dataset by inflating low-variance data and deflating high-variance data (Yeo, 2000). It is valuable for modeling issues associated with heteroscedasticity (variance that is not constant). SMOTE is an acronym that stands for Synthetic Minority Oversampling Technique. It is a technique

used to oversample the minority group in an unbalanced data set. Instead of merely duplicating existing examples, SMOTE generates novel examples for the minority class. This is accomplished by selecting minority class examples that are close in feature space and generating new synthetic examples by linearly combining the selected examples. This can help to enhance the efficacy of machine learning models on unbalanced datasets by balancing the class distribution (Chawla et al., 2002).

# 4. Modeling

Now in the pursuit of identifying optimal models for time series forecasting, an extensive array of classifiers was examined. Each classifier had distinct strengths, allowing us to use their specific attributes in order to achieve optimal predicted performance.

The **Light Gradient Boosting Machine** (LightGBM) is a sophisticated framework for gradient boosting that demonstrates exceptional efficiency in managing big datasets, making it particularly advantageous for time series research. The software utilizes a distinctive methodology based on histograms to perform data binning, resulting in improved speed and efficiency in terms of memory use. The seamless handling of both categorical and numerical variables enables us to derive significant insights from the time series data (Chen & Guestrin, 2016; Friedman, 2001).

The **Extra Trees Classifier** is a type of ensemble learning algorithm that utilizes the construction of numerous decision trees in the training phase and then aggregates their predictions. What distinguishes it is the incorporation of supplementary randomization through the consideration of random divisions for the candidate characteristics. The use of randomization has the potential to enhance the classifier's capacity to effectively process noisy input and mitigate the issue of overfitting (Sharaff & Gupta, 2019).

The **Random Forest Classifier** is a type of ensemble approach that combines predictions from many decision trees, similar to the Extra Trees Classifier. In the forest, every individual tree undergoes training using a randomly selected portion of the available data. This process of diversity serves to improve the model's resilience and its ability to generalize to unseen instances (Breiman, 2001).

**Logistic regression** is a well-established and extensively employed linear classification approach. Despite its inherent simplicity, this approach exhibits a remarkable degree of efficacy across many circumstances, particularly in instances when a distinct linear association exists between the attributes and the target variable. The interpretability of the model renders it highly beneficial in comprehending the influence of many factors on the forecasts of time series (Hosmer & Lemeshow, 2000).

The **Ridge Classifier** is a type of linear classifier that incorporates L2 regularization as a means to address concerns associated with multicollinearity and enhance the model's generalization capabilities. The effectiveness of time series data in classification tasks is enhanced when there is a certain degree of linear separability present, making it a valuable tool to consider among our range of classifiers (Hastie et al., 2009).

The **K Neighbors Classifier** is a non-parametric technique that assigns labels to data by determining the majority class among their k nearest neighbors in the feature space. Although its primary use is in classification jobs, the algorithm may also be modified to do time series forecasting, particularly when the data displays localized patterns and short-term interdependencies (Viswanath & Hitendra Sarma, 2011).

The **Support Vector Machines** (SVM) technique, when combined with the Radial Kernel, is widely used for performing classification and regression problems. The functioning of this method involves the conversion of data into a space with a greater number of dimensions, followed by the identification of an ideal hyperplane that effectively distinguishes between various classes. The aforementioned methodology demonstrates efficacy in comprehensively capturing intricate linkages present within the dataset of time series (Cortes & Vapnik, 1995).

The **MLP Classifier**, also known as the Multi-Layer Perceptron, is an artificial neural network that consists of numerous layers of linked nodes. The method is highly suitable for capturing complex patterns within time series data and is capable of accommodating both linear and non-linear connections. The capacity to autonomously acquire hierarchical representations of features renders it a beneficial instrument for intricate time series forecasting endeavors (Goodfellow et al., 2016).

Through the utilization of a varied array of classifiers, we effectively leveraged the unique capabilities of each technique to construct a complete ensemble of models. This

ensemble facilitated the precise forecasting of future values within our time series data. The use of rigorous methodology eventually facilitated the development of solid and dependable forecasting models, which yielded significant insights and forecasts for our applications within a given area.

## 4.1. Data Modelling

The subsequent step in our research was the use of the sklearn TimeSeries Split technique to partition our datasets, subsequent to the process of Feature Engineering. This was undertaken to facilitate the ongoing training of our models. This methodology enables us to efficiently manage the temporal characteristics of the data and guarantees the preservation of time-dependent patterns and trends throughout the splitting procedure (Petropoulos et al., 2022b).

The data was partitioned into train/test and validation sets using the TimeSeries Split technique, ensuring that the temporal sequence of the records was preserved. Time series analysis is of utmost importance due to its ability to replicate real-world situations, wherein predictions of future events are derived from previous data.

Subsequently, the accessible data were subjected to training, testing, and assessment utilizing distinct sets. This allows for a more precise evaluation of the model's performance by taking into account its capacity to predict future occurrences using past data. The utilization of the TimeSeries Split approach is crucial in maintaining the robustness and dependability of our model's predictions.

### 4.1.1. Model Selection

Retaining the non-training days throughout the testing phase was of utmost importance in order to facilitate the learning process of our models by incorporating the entirety of temporal patterns and obtaining a holistic comprehension of the data's behavior. Incorporating non-training days inside the training set enables the models to effectively capture any temporal patterns such as seasonality, trends, or other time-dependent factors that may exist in the data. This inclusion ultimately enhances the accuracy of the predictions.

Nevertheless, while assessing the performance of the model, it was imperative to employ a more rigorous methodology. In order to enhance the accuracy of the assessment measures, the evaluation method excluded the non-training days and their corresponding forecasts. By employing this approach, we mitigate the risk of artificially inflating the assessment outcomes and guarantee that the model's capacity to make accurate predictions is evaluated only based on data that has not been previously seen.

The aforementioned approach holds significant importance in the field of time series analysis, as it frequently revolves around the objective of predicting future values by leveraging prior data. By eliminating the days without training throughout the assessment process, we create a simulation that closely resembles real-world conditions. This allows us to assess the model's capacity to perform effectively on new and unknown data, which is crucial for determining its practicality and dependability in real-world applications.

Moreover, in our endeavor to attain optimal predictive efficacy, we also investigated the utilization of model ensembles as a methodology to further augment our outcomes. The utilization of model ensembles entails the amalgamation of predictions generated by numerous distinct models, resulting in a final forecast that is more resilient and precise.

Nevertheless, while the potential benefits associated with model ensembles, our specific time series study did not deliver the intended findings. There are several potential factors that may have contributed to this conclusion, including the limited variety across the base models and the challenges associated in integrating several models for time series forecasting.

### 4.1.2. Models Evaluation

In order to effectively train our models and achieve their maximum performance, we employed the initial partition of the data and implemented a rigorous 20-fold cross-validation methodology. The proposed approach entails partitioning the dataset into twenty distinct subsets. Each subset is utilized as the validation set once, while the remaining nineteen subsets collectively provide the training set. By systematically rotating the validation and training sets over all partitions, we were able to acquire a

thorough evaluation of each model's skills and its capacity to generalize across various subsets of the data.

Throughout the process of cross-validation, we assessed the models by using a range of performance measures in order to measure their efficacy across different dimensions. The models were initially classified based on their Area Under the Curve (AUC), a commonly employed metric for assessing the classifier's capacity to differentiate between positive and negative data. A classifier with a larger Area Under the Curve (AUC) value is indicative of superior performance.

Subsequently, the models were evaluated and ranked according to their F-Score, a metric that effectively balances accuracy and recall. The F-Score metric is particularly advantageous in the context of unbalanced datasets, a common occurrence in the domain of time series forecasting.

In conjunction with the evaluation of AUC and F-Score, we conducted a thorough analysis of many other significant metrics to ensure a full evaluation of the model's performance. The metrics included in this analysis are:

**Accuracy** is a metric that measures the ratio of correctly predicted samples to the total number of samples. It is a comprehensive assessment of the model's accuracy (Baldi et al., 2000).

**Area Under the Curve (AUC)** is a metric that is used to evaluate the performance of binary classification models. It is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at different thresholds. The AUC score is the area under this curve, and it ranges from 0 to 1. A perfect model will have an AUC score of 1, while a model that is no better than random guessing will have an AUC score of 0.5 (Bradley, 1997).

**Recall** (also known as the true positive rate or sensitivity) quantifies the ratio of properly detected actual positive samples by the model (Yu et al., 2002).

**Precision** is a metric that evaluates the accuracy of a model by measuring the proportion of properly detected positive samples out of all the samples that were predicted as positive. It provides valuable insights into the model's level of exactness (Davis & Goadrich, 2006).

The **F1** Score is defined as the harmonic mean of accuracy and recall, so offering a well-balanced evaluation of the model's performance (Sokolova et al., 2006).

**Cohen's Kappa coefficient** is a statistical measure that is utilized to assess the level of agreement between a model's predictions and the observed results, taking into account the possibility of random chance (Callender & Osburn, 1977; Cohen, 1960).

**The Matthews Correlation Coefficient (MCC)** is a statistic that is especially valuable in the context of unbalanced datasets. It quantifies the connection between the predicted labels and the real labels (Chicco & Jurman, 2020).

Through a rigorous process of calculating and scrutinizing several indicators, we acquired a thorough understanding of the merits and drawbacks of each model. This enabled us to make educated and judicious choices in identifying the most effective models for our time series forecasting endeavors.

Upon conducting a comprehensive 20-fold cross-validation on the training dataset, a detailed evaluation of the models' performance was carried out. Utilizing the knowledge acquired via this procedure, the models were subsequently utilized to make predictions on the testing dataset, thereby affording us an evaluation of their performance in test predictions. This stage facilitated the evaluation of the models' ability to generalize to unfamiliar data, which is essential for confirming their efficacy in real-world contexts.

After completing the test prediction phase, we proceeded to finalize the models that were selected. The models that exhibited the highest performance, as assessed by their assessment metrics, were selected to undergo additional development. The selected models were subsequently subjected to retraining, wherein the training data was augmented with the inclusion of the testing data. The inclusion of testing data during the retraining process guarantees that the models derive advantages from a broader dataset, therefore catching supplementary patterns and enhancing their capacity for generalization.

After completing the finalization and retraining of the models, we proceeded to utilize them once more for the purpose of making predictions on the validation set. The evaluation score for each model's performance was determined based on the final prediction made on the validation set. The validation set is a separate dataset from both

the training and testing sets, serving as an independent sample that the models have not been exposed to previously. This characteristic makes it a crucial reference point for evaluating the genuine predictive capability of the models.

The identification of the optimal models among the candidate classifiers was facilitated by evaluating their performance on the validation set. The ultimate selection for time series forecasting was made based on the model that achieved the greatest performance score on the validation set. The meticulous procedure guarantees the selection of models that exhibit high performance in both cross-validation and testing, as well as possess robust predicting skills on unobserved data, rendering them highly suitable for practical implementation in real-life situations.

# 5. Results

After collecting and preprocessing the data and devising the methodology, the next crucial stage is to evaluate the accuracy and suitability of our model for predicting the stock price of Microsoft. In this chapter, we will train our model and provide a graphical overview of our data. Then, we will evaluate our model's ability to predict stock prices using a variety of metrics, such as F-score and Area Under the Curve (AUC). We will predict the fluctuation of Microsoft's stock market for each ML model at the conclusion of this chapter.

## 5.1. Models Performance Evaluation

In an endeavor to forecast the future directional movement of a stock, namely whether it will increase or decrease, a heterogeneous ensemble of seven machine learning models was trained and assessed. The models utilized in this study encompass the Light Gradient Boosting Machine (LightGBM), Extra Trees Classifier (ET), Random Forest Classifier (RF), Logistic Regression (LR), Ridge Classifier, K Neighbors Classifier (knn), Support Vector Machine with Radial Kernel (rbfsvm), and the Multilayer Perceptron Classifier (mlp). In order to evaluate the performance and efficacy of these models, a comprehensive range of metrics was utilized. These metrics encompass Accuracy, which serves as an indicator of the overall accuracy of predictions; Area Under the ROC Curve (AUC), a metric that assesses the trade-offs between sensitivity and specificity; and the F1 score, a measure that combines precision and recall to provide insights into the models' capacity to strike a balance between false positives and false negatives. These indicators combined provide a comprehensive perspective on the prediction skills of each model and enable a thorough evaluation of their respective performances.

### 5.1.1. Evaluation on Cross-Validation Procedure

As seen in table 1 below the Light Gradient Boosting Machine (LightGBM), Extra Trees Classifier (ET), and Random Forest Classifier (RF) regularly demonstrate superior performance compared to other models, with accuracy scores over 71%, AUC

values surpassing 0.81, and F1 scores approximately around 0.73. These numbers demonstrate a heightened capacity to accurately forecast the future movement of stocks on the next day while efficiently managing the trade-off between correctly identifying good outcomes and incorrectly identifying positive outcomes. Logistic Regression (LR) and the Ridge Classifier exhibit similar performance, with the Ridge Classifier displaying an AUC of 0, which may suggest a possible concern with either the model or the computation of this measure. The K Neighbors Classifier (KNN), Support Vector Machine with Radial Kernel (RBF-SVM), and Multi-Layer Perceptron Classifier (MLP) demonstrate worse performance in relation to these three criteria. The MLP Classifier exhibits the lowest performance, as indicated by an accuracy rate below 57%, an AUC slightly below 0.66, and an F1 score of 0.5506. These results suggest that more tweaking or tweaks may be necessary to optimize its suitability for this specific task.

| Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC | TT (Sec) |
|---|---|---|---|---|---|---|---|---|
| LGBM | 0,712 | 0,8112 | 0,7523 | 0,7199 | 0,735 | 0,4198 | 0,4216 | 0,8365 |
| ET | 0,7111 | 0,8166 | 0,7516 | 0,7196 | 0,7343 | 0,418 | 0,4198 | 1,054 |
| RF | 0,7107 | 0,809 | 0,748 | 0,722 | 0,7337 | 0,417 | 0,4189 | 1,0715 |
| LR | 0,6411 | 0,6843 | 0,6447 | 0,6701 | 0,6563 | 0,2812 | 0,2822 | 0,8325 |
| RIDGE | 0,6379 | 0 | 0,6474 | 0,6648 | 0,6551 | 0,2741 | 0,275 | 0,787 |
| KNN | 0,6212 | 0,6621 | 0,608 | 0,6562 | 0,6297 | 0,243 | 0,2446 | 0,9005 |
| RBFSVM | 0,6207 | 0,6757 | 0,6466 | 0,6433 | 0,6426 | 0,2381 | 0,2398 | 0,9045 |
| MLP | 0,5689 | 0,6558 | 0,6326 | 0,6478 | 0,5506 | 0,1298 | 0,1735 | 0,8795 |

*Table 5 Evaluation on Cross-Validation Procedure*

### 5.1.2. Evaluation on Testing Procedure

The test set data presents a more varied picture across the models. As seen in table 2 below, the lgbm model maintains its lead, showcasing the best performance with an Accuracy of 0.712, AUC of 0.804, and an F1 score of 0.7353. The Extra Trees Classifier (et) closely follows with similar metrics, reflecting robust performance.

However, the Random Forest Classifier (rf) appears to have slipped in performance compared to the previous evaluations, with a decrease in Accuracy to 0.6859 and an AUC of 0.7952. Although these numbers are still commendable, they signal a relative decline that may warrant investigation.

The middle-tier models such as the Ridge Classifier, K Neighbors Classifier (knn), and SVM with Radial Kernel (rbfsvm) continue to present modest performances with accuracies ranging from approximately 60% to 62%. The slight variations among these models in the AUC and F1 score might indicate differences in how they handle the trade-off between true positives and false positives or their balance between precision and recall.

Logistic Regression (lr) has also experienced a dip in performance, with an Accuracy of 0.5935 and AUC of 0.6542. Its relatively lower recall of 0.5633 and F1 score of 0.5961 may signal difficulties in correctly identifying positive (UP) stock movements.

The MLP Classifier (mlp) persists as the underperformer in this set, with an Accuracy of 0.5696, AUC of 0.6216, and notably, a Recall of only 0.5. This poor recall, along with the lowest Kappa and MCC, underscores the model's limitations in handling this predictive task.

In summary, while the top-tier models like lgbm and et continue to demonstrate strong performance, there have been some shifts among the other models, with notable declines in rf and lr. The continued struggles of the mlp model affirm the need for further tuning or a different modeling approach. The results across these varying models emphasize the importance of understanding the underlying data and the specific requirements of the predictive task when selecting and fine-tuning models.

| Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0,712 | 0,804 | 0,751 | 0,7202 | 0,7353 | 0,4197 | 0,4202 |
| ET | 0,7043 | 0,8002 | 0,7449 | 0,7129 | 0,7285 | 0,4043 | 0,4048 |
| RF | 0,6859 | 0,7952 | 0,7184 | 0,6998 | 0,709 | 0,3679 | 0,368 |
| LR | 0,5935 | 0,6542 | 0,5633 | 0,633 | 0,5961 | 0,1897 | 0,191 |
| RIDGE | 0,6109 | 0,6126 | 0,5857 | 0,6493 | 0,6159 | 0,2237 | 0,2249 |
| KNN | 0,6011 | 0,6535 | 0,5918 | 0,6346 | 0,6125 | 0,2025 | 0,203 |
| RBFSVM | 0,6228 | 0,6657 | 0,6449 | 0,6462 | 0,6456 | 0,2425 | 0,2425 |
| MLP | 0,5696 | 0,6216 | 0,5 | 0,6187 | 0,553 | 0,1469 | 0,15 |

*Table 6 Evaluation on Testing Procedure*

### 5.1.3. Evaluation on Hold Out Dataset

The holdout set results shown in table 3 bellow follow a similar trend to the training and cross-validation results. Tree-based models, namely Light Gradient Boosting Machine (lgbm), Extra Trees Classifier (et), and Random Forest Classifier (rf), have continued to lead in performance, with Accuracy above 74% and AUC values above 0.84. Among them, the Random Forest Classifier slightly outperforms others with an Accuracy of 0.7489 and an F1 score of 0.7734. Their relatively high values across Recall, Precision, Kappa, and MCC affirm their robustness in predicting next-day stock movement.

On the other hand, the Logistic Regression (lr) and Ridge Classifier models present moderate performance with accuracy scores in the range of 64-65% and AUC around 0.65-0.69. The Ridge Classifier's improvement in AUC from 0 in the training set to 0.6491 in the holdout set might indicate a more balanced model when faced with unseen data.

Interestingly, the K Neighbors Classifier (knn) and SVM with Radial Kernel (rbfsvm) showed an improvement in the holdout set compared to the training phase, particularly in the AUC metric, with values of 0.7235 and 0.74, respectively. This could suggest that

these models might have generalization capabilities that were not fully captured in the training phase.

Finally, the MLP Classifier (mlp) continues to struggle, yielding the lowest performance across almost all metrics, including an accuracy of 0.6043 and an AUC of 0.6532. These figures underscore its limitations in handling this prediction task, and it might need substantial tuning or a complete redesign.

Overall, the tree-based models remain the top contenders for this specific task, demonstrating strong predictive capabilities on unseen data. The improvement of some models in the holdout set compared to the training phase can provide valuable insights into their potential generalizability and could guide further model refinement and selection.

| Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0,7466 | 0,846 | 0,7718 | 0,7686 | 0,7702 | 0,4877 | 0,4878 |
| ET | 0,7473 | 0,8422 | 0,7898 | 0,7603 | 0,7748 | 0,4873 | 0,4878 |
| RF | 0,7489 | 0,8408 | 0,7787 | 0,7681 | 0,7734 | 0,4918 | 0,4919 |
| LR | 0,6476 | 0,6883 | 0,6611 | 0,6868 | 0,6737 | 0,2911 | 0,2913 |
| RIDGE | 0,6507 | 0,6491 | 0,6653 | 0,6891 | 0,677 | 0,297 | 0,2972 |
| KNN | 0,6705 | 0,7235 | 0,6957 | 0,7025 | 0,6991 | 0,3349 | 0,335 |
| RBFSVM | 0,6872 | 0,74 | 0,751 | 0,7016 | 0,7255 | 0,363 | 0,3642 |
| MLP | 0,6043 | 0,6532 | 0,5726 | 0,6624 | 0,6142 | 0,2124 | 0,2148 |

*Table 7 Evaluation on Hold Out Dataset*

### 5.1.4. Evaluation on only Training Days

The differentiation between trading and non-trading days has significant importance in the context of predictive modeling for stock market movements. Non-trading days frequently manifest sluggish or repeating price patterns due to the absence of active trading activities. The inclusion of non-trading days in the dataset may result in the inflation of performance measures, as models may readily identify and exploit these

patterns. However, it is important to note that these patterns may not necessarily possess predictive ability for actual trading days.

This phenomenon has the potential to generate a deceptive perception of precision and effectiveness in prediction algorithms. This phenomenon might be likened to a model that is "learning" the noise rather than the fundamental signal. The objective of just considering trading days is to construct a model that accurately captures the underlying dynamics of the market, leading to a more practical and relevant comprehension of stock movements.

Custom weighted measures were utilized in the evaluation of the Light Gradient Boosting Machine (LGBM) model. Weighted measures, such as Weighted Recall, Weighted Precision, and Weighted F1, incorporate the consideration of class imbalance by assigning varying weights to distinct classes during the computation of the metric. This approach can offer a more comprehensive comprehension of the performance of a model across several categories, particularly in situations where specific categories may have little representation. Weighted versions of classification algorithms can provide a more accurate representation of real-world situations, since they account for the varying costs associated with misclassifying various classes.

The choice to remove non-trading days and the utilization of custom weighted criteria are in line with an endeavor to achieve a more realistic and practical assessment of the model, which is geared to address the particular complexities and intricacies associated with forecasting stock market fluctuations.

The lgbm model shows an Accuracy of approximately 62.03% on the given dataset, reflecting its ability to correctly predict next-day stock movement in about three-fifths of the cases. The AUC (Area Under the Curve), AUC Micro, and AUC Weighted metrics are all at around 0.6108, providing a consistent measure of the model's discrimination capabilities between the positive (UP) and negative (DOWN) classes.

The model's Recall, Precision, and F1 score are all equal at around 0.6203. This unusual alignment might indicate a balanced dataset or specific handling of the classification threshold. Kappa and MCC (Matthews Correlation Coefficient), which evaluate the agreement between prediction and observation, are at 0.2219, signifying moderate agreement.

Macro Precision, Macro Recall, and Macro F1 scores, at approximately 0.6111, 0.6108, and 0.6109, respectively, provide insights into the model's ability to balance performance across different classes. The weighted variations of Recall, Precision, and F1 score (0.6203, 0.6199, and 0.6201) provide a slight deviation, perhaps reflecting the weighted contributions of different classes to the metrics.

| Model | lgbm |
|---|---|
| Accuracy | 0,620296 |
| AUC | 0,610827 |
| AUC Micro | 0,610827 |
| AUC Weighted | 0,610827 |
| Recall | 0,620296 |
| Precision | 0,620296 |
| F1 | 0,620296 |
| Kappa | 0,22189 |
| MCC | 0,221896 |
| Macro Precision | 0,611068 |
| Macro Recall | 0,610827 |
| Macro F1 | 0,61094 |
| Weighted Recall | 0,620296 |
| Weighted Precision | 0,619898 |
| Weighted F1 | 0,62009 |

*Table 8 Evaluation on only Training Days Results*

To further understand the predictions of each class, we create a confusion matrix for all predictions, one for each year, and one for each company.

#### 5.1.4.1. Main Confusion Matrix Analysis

The confusion matrix below shows that the model correctly classified 340 as up (true positive) and 1 204 instances as down (true negative). However, the model also misclassified 165 instances as down (false positive) and 168 instances as up (false negative) (DOI: 10.1162/jmlr.2006.7.7.1003).

The accuracy of the model is 0.62, which means that it correctly classified 62% of the instances. The precision of the model is 0.67, which means that 67% of the instances that were classified as up were actually up. The recall of the model is also 0.67, which means that 67% of the actual up instances were correctly classified as up.

The F1 score of the model is 0.67, which is a weighted average of precision and recall. This means that the model did a good job of both correctly classifying ‘’UP’ instances and correctly classifying down instances.

Overall, the results of the confusion matrix show that the model is able to predict the next day movement of a stock price with some accuracy. However, there is still room for improvement, as the model misclassified a significant number of instances.

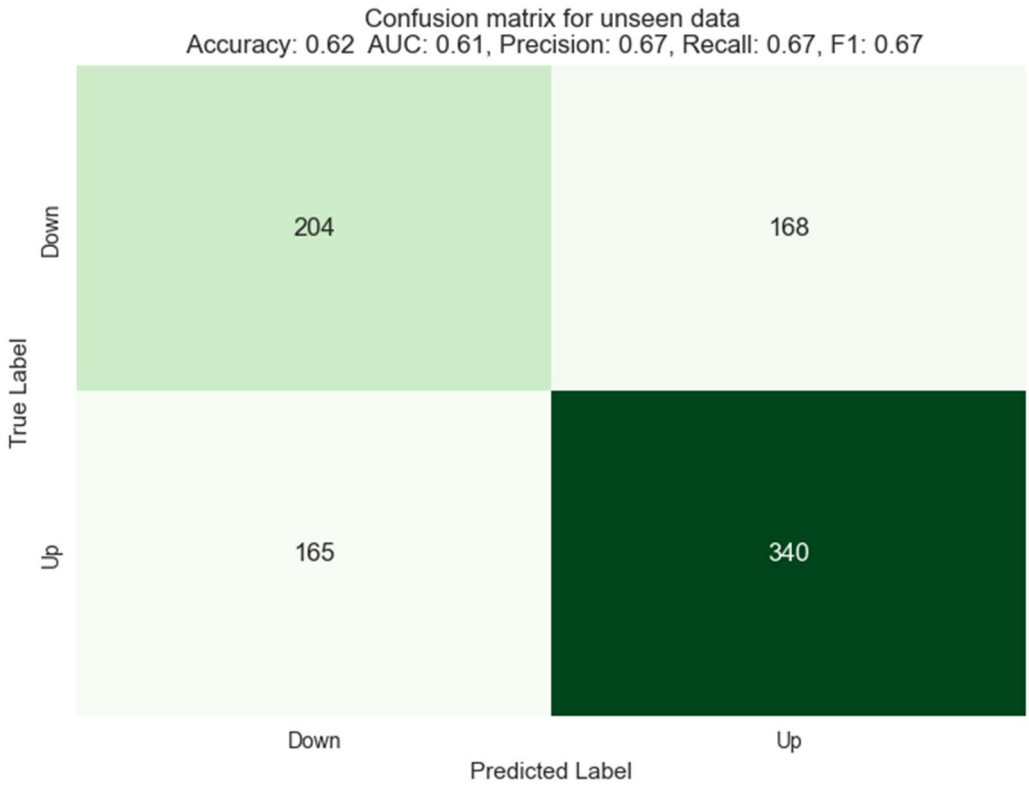



*Figure 5 Holistic Confusion Matrix for Unseen Data*

### 5.1.4.2. Confusion Matrix Per Year Analysis

Through the examination of the confusion matrix on an annual basis, it is possible to detect any potential issues with the model and implement measures to enhance its performance. For instance, in the event that the accuracy of the model exhibits a declining trend over a period of time, it may indicate a potential deterioration in the model's ability to effectively assimilate new data compared to its previous performance. There are several potential causes that may contribute to this phenomenon, including the possibility of the model overfitting to the training data or the potential for the data distribution to undergo temporal changes. Through the examination of the confusion matrix on an annual basis, it becomes possible to discern alterations in the model's efficacy over time, pinpoint particular classes that pose challenges for the model, detect any discernible trends or patterns in the model's performance, ascertain whether the model continues to acquire knowledge from novel data, and identify any potential issues with the model, thereby enabling the implementation of measures to enhance its performance.

| Year | Accuracy | AUC | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| 2019 | 0.60 | 0.59 | 0.67 | 0.64 | 0.66 |
| 2020 | 0.65 | 0.65 | 0.72 | 0.67 | 0.70 |
| 2021 | 0.61 | 0.60 | 0.61 | 0.72 | 0.66 |

*Table 9 Prediction Results Yearly Breakdown*

For the year 2019 (Figure 2), the model accurately identified a total of 71 movements as "Down" and 89 movements as "Up". Nevertheless, the model exhibited erroneous classification by mislabeling 68 movements as "Up" and 59 movements as "Down".

The model's accuracy is 0.60, indicating that it accurately identified 60% of the movements. The area under the receiver operating characteristic curve (AUC) for the model is 0.59, indicating that the model has a 59% probability of accurately discriminating between a "Down" picture and a "Up" image.

The model's precision is 0.67, indicating that 67% of the movements labeled as "Up" by the model were accurately classified as such. The model's recall rate is 0.64, indicating that 64% of the "Down" movements were accurately identified by the model.

The F1 score of the model is 0.66, representing a weighted average of the accuracy and recall metrics.

In general, the model has a satisfactory accuracy level of 0.60. Nevertheless, there is room for improvement in the model by enhancing both precision and recall.

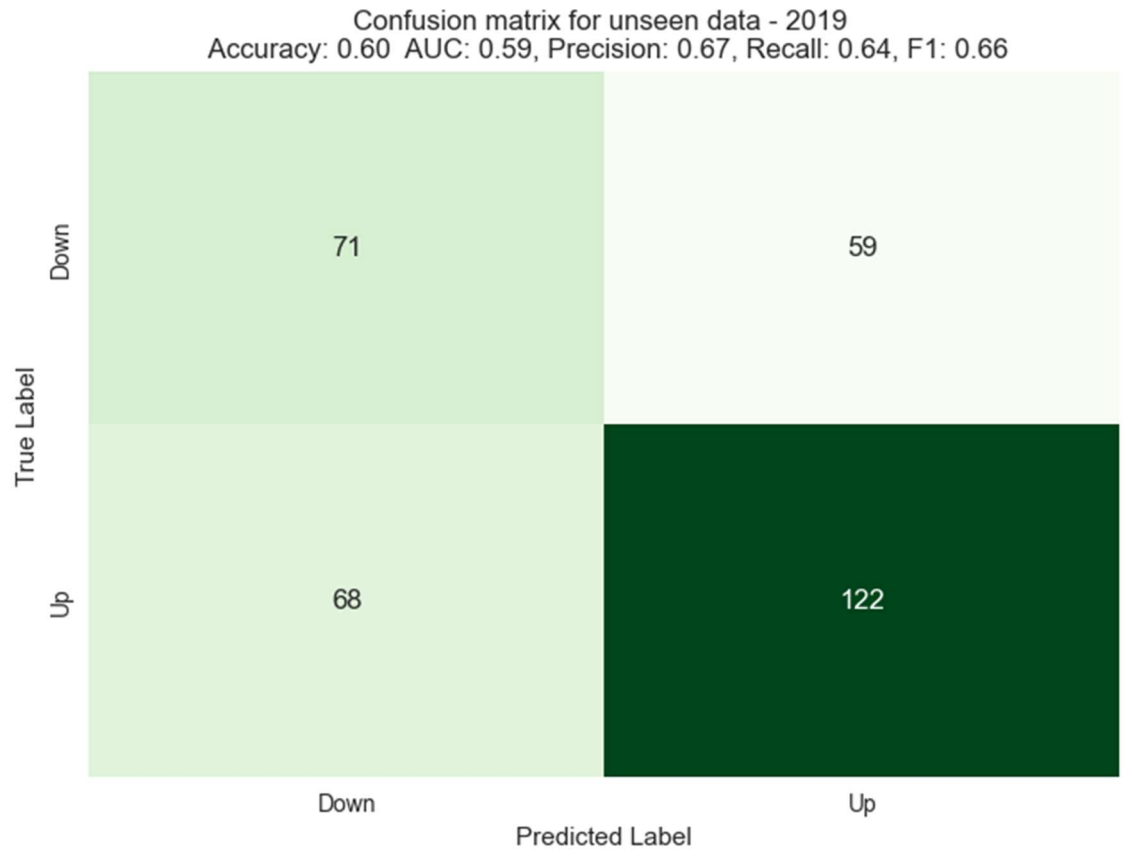



*Figure 6 Confusion Matrix for Year 2019*

The confusion matrix for the year 2020 is shown in figure 3. It shows the performance of a machine learning model that was trained to classify movements as "Down" or "Up".

The table shows that the model correctly classified 73 movements as "Down" and 113 movements as "Up". However, the model incorrectly classified 43 movements as "Up" and 56 movements as "Down".

The accuracy of the model is 0.65, which means that the model correctly classified 65% of the movements. The AUC of the model is 0.65, which means that the model has a 65% chance of correctly distinguishing between a "Down" image and an "Up" image.

The precision of the model is 0.72, which means that 72% of the movements that the model classified as "Down" were actually "Down". The recall of the model is 0.67, which means that 67% of the "Down" movements were correctly classified by the model.

The F1 score of the model is 0.70, which is a weighted average of precision and recall.

Overall, the model has a good accuracy of 0.65. However, the model could be improved for this year also by increasing the precision and recall.

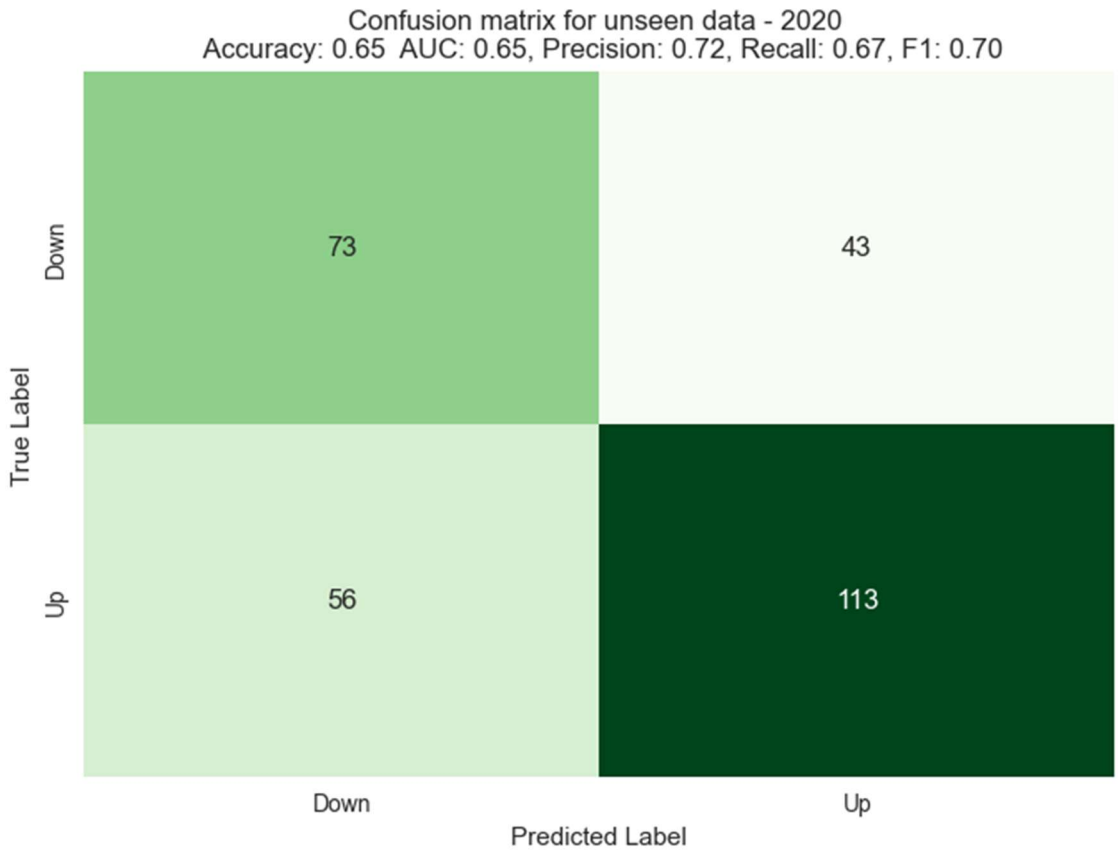



*Figure 7 Confusion Matrix for Year 2020*

The matrix bellow (figure 4) presents the evaluation of the model's performance using previously unobserved data in the year 2021.

The matrix illustrates that the model accurately identified 66 movements as "Down" and 105 movements as "Up". Nevertheless, the model exhibited erroneous classification by labeling 60 movements as "Up" and 41 movements as "Down".

The model's accuracy is 0.61, indicating that it accurately categorized 61% of the movements. The area under the receiver operating characteristic curve (AUC) of the model is 0.60, indicating that the model has a 60% probability of accurately discriminating between a "Down" picture and an "Up" image.

The model's precision is 0.61, indicating that 61% of the "Up" movements were accurately identified by the model.

The model's recall rate is 0.72, indicating that 72% of the "Down" movements were accurately identified by the model.

The F1 score of the model is 0.66, representing a weighted average of the accuracy and recall metrics.

In general, the model has a satisfactory accuracy rate of 0.61. Nevertheless, there is room for improvement in the model by enhancing both precision and recall.

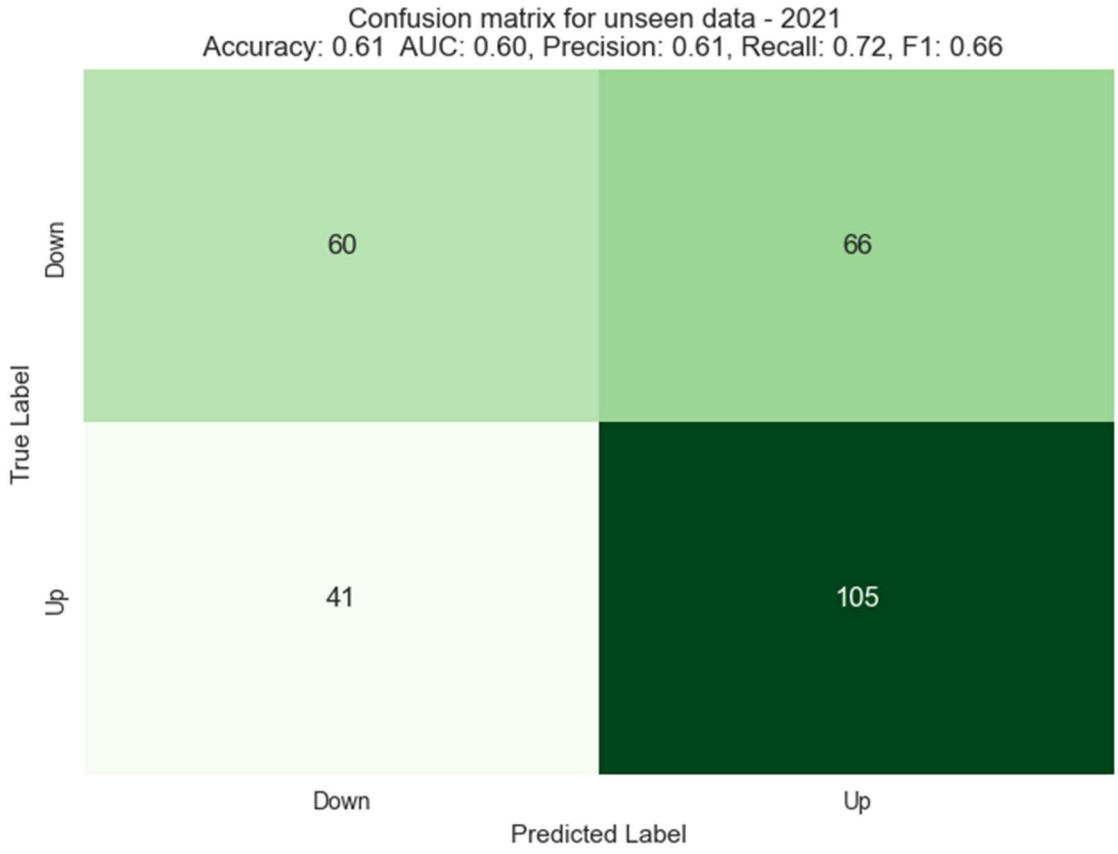



*Figure 8 Confusion Matrix for Year 2021*

### 5.1.4.3. Confusion Matrix Per Company Analysis

The same analysis can be performed to find potential weaknesses of the model for each specific company.

| Company | Accuracy | AUC | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| APPLE | 0.62 | 0.60 | 0.66 | 0.72 | 0.69 |
| MICROSOFT | 0.68 | 0.66 | 0.73 | 0.73 | 0.73 |
| TESLA | 0.61 | 0.61 | 0.68 | 0.62 | 0.65 |
| AMAZON | 0.57 | 0.56 | 0.60 | 0.62 | 0.61 |

*Table 10 Prediction Results Per Company Breakdown*

The model's accuracy for the APPLE dataset is 62%, as shown in the matrix bellow (Figure 5).

The matrix illustrates that the model accurately identified 42 movements as "Down" and 87 movements as "Up". Nevertheless, the model exhibited erroneous classification by labeling 45 movements as "Up" and 42 movements as "Down".

The model's accuracy is 0.62, indicating that it accurately categorized 62% of the movements. The area under the receiver operating characteristic curve (AUC) of the model is 0.60, indicating that the model has a 60% probability of accurately discriminating between a "Down" picture and an "Up" image.

The model's precision is 0.66, indicating that 66% of the "Up" movements were accurately identified by the model.

The model's recall rate is 0.72, indicating that 72% of the "Down" movements were accurately identified by the model.

The F1 score of the model is 0.69, representing a weighted average of the accuracy and recall metrics.

In general, the model has a satisfactory accuracy rate of 0.62. Nevertheless, there is room for improvement in the model by enhancing both precision and recall.

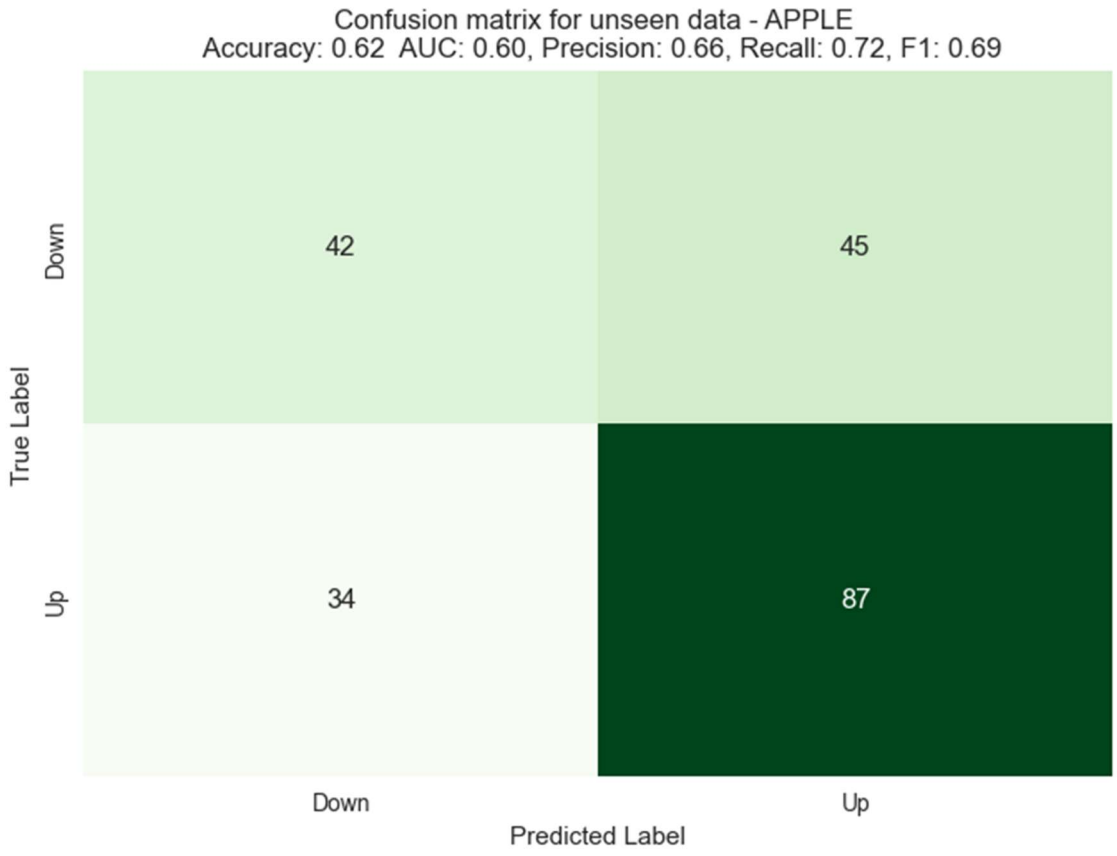



*Figure 9 Confusion Matrix for APPLE*

The model's performance for the MICROSOFT data is significantly better, as shown in the figure 6 bellow.

The matrix illustrates that the model accurately identified 42 movements as "Down" and 87 movements as "Up". Nevertheless, the model exhibited erroneous classification by labeling 45 movements as "Up" and 42 movements as "Down".

The model's accuracy is 0.62, indicating that it accurately categorized 62% of the movements. The area under the receiver operating characteristic curve (AUC) of the model is 0.60, indicating that the model has a 60% probability of accurately discriminating between a "Down" picture and an "Up" image.

The model's precision is 0.73, indicating that 68% of the "Up" movements were accurately identified by the model.

The model's recall rate is 0.72, indicating that 72% of the "Down" movements were accurately identified by the model.

The F1 score of the model is 0.69, representing a weighted average of the accuracy and recall metrics.

In general, the model has a satisfactory accuracy rate of 0.62. Nevertheless, there is room for improvement in the model by enhancing both precision and recall.

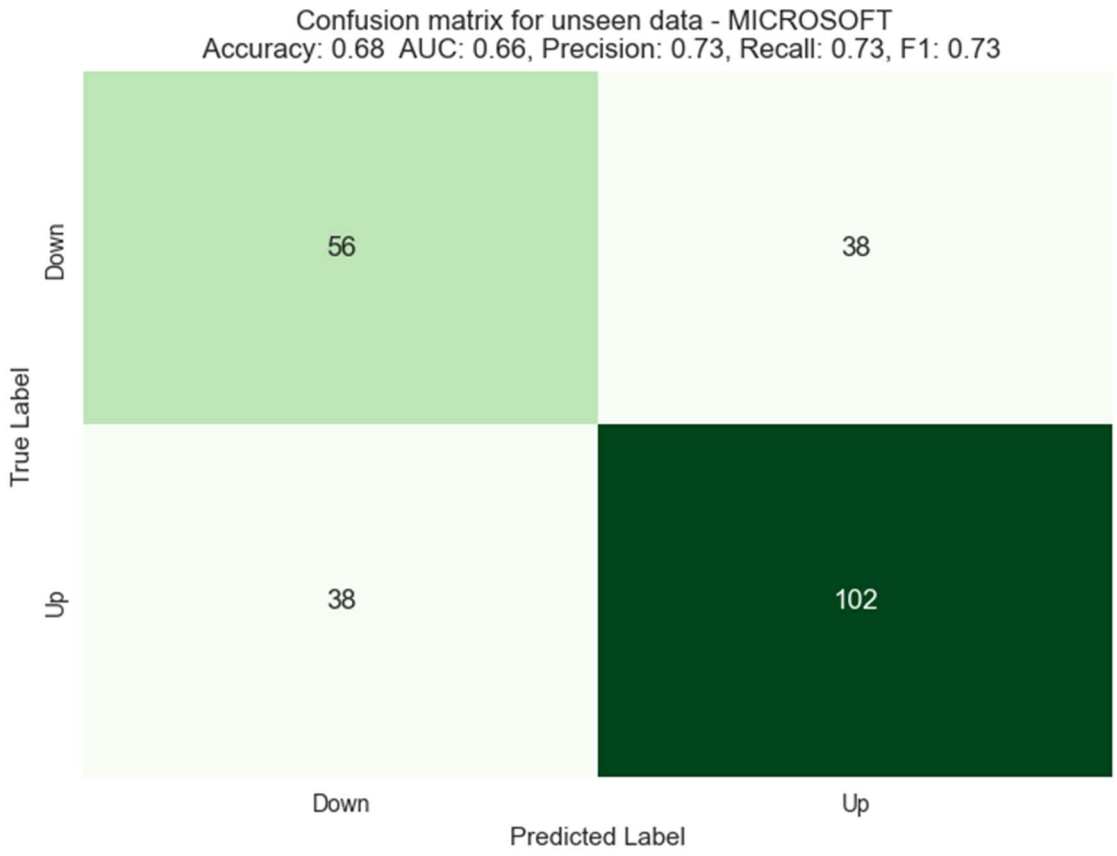



*Figure 10 Confusion Matrix for MICROSOFT*

The model's accuracy for the TESLA dataset is 61%, as shown in the figure 7 bellow.

The matrix illustrates that the model accurately identified 54 movements as "Down" and 76 movements as "Up". Nevertheless, the model exhibited erroneous classification by labeling 35 movements as "Up" and 47 movements as "Down".

The model's accuracy is 0.61, indicating that it accurately categorized 61% of the movements. The area under the receiver operating characteristic curve (AUC) of the model is 0.61, indicating that the model has a 61% probability of accurately discriminating between a "Down" picture and an "Up" image.

The model's precision is 0.68, indicating that 68% of the "Up" movements were accurately identified by the model.

The model's recall rate is 0.62, indicating that 62% of the "Down" movements were accurately identified by the model.

The F1 score of the model is 0.65, representing a weighted average of the accuracy and recall metrics.

In general, the model has a satisfactory accuracy rate of 0.61. Nevertheless, there is room for improvement in the model by enhancing both precision and recall.

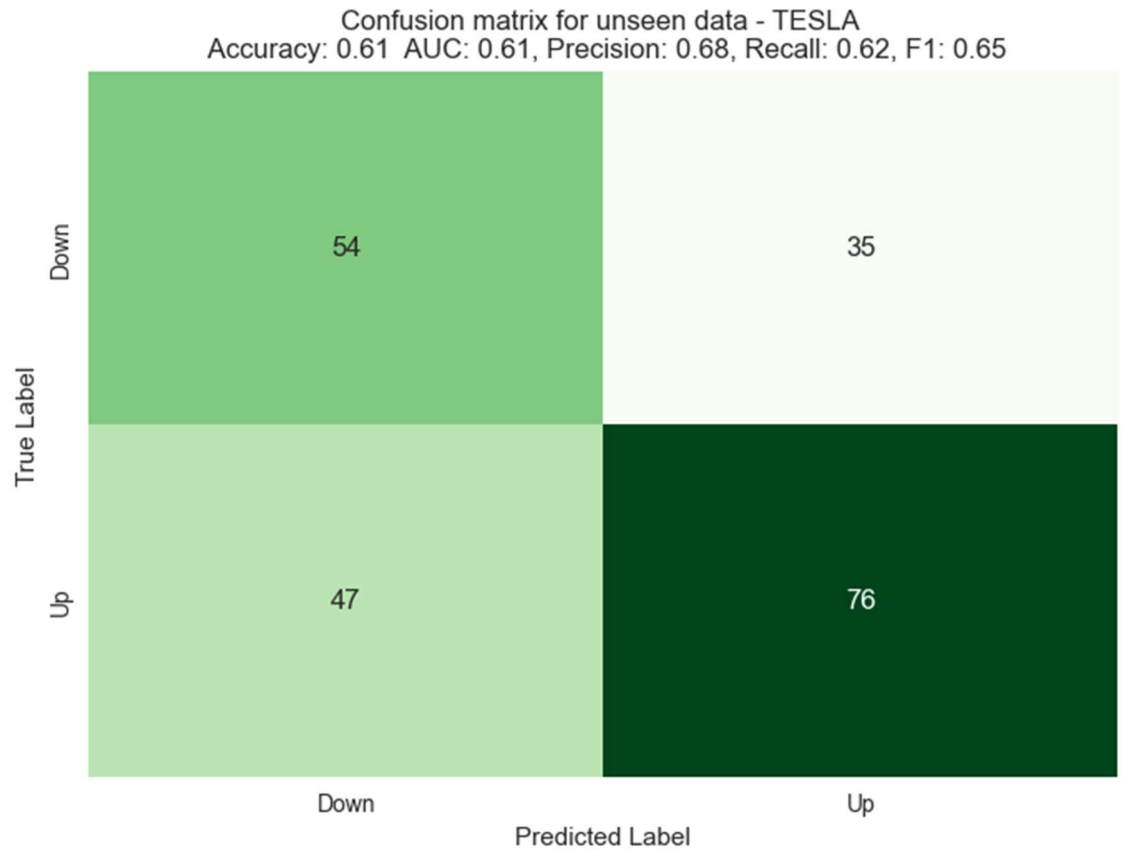



*Figure 11 Confusion Matrix for TESLA*

The model's accuracy for the AMAZON dataset is 57%, the lowest among the companies as shown in the figure 8 bellow.

The matrix illustrates that the model accurately identified 52 movements as "Down" and 75 movements as "Up". Nevertheless, the model exhibited erroneous classification by labeling 50 movements as "Up" and 46 movements as "Down".

The model's accuracy is 0.57, indicating that it accurately categorized 57% of the movements. The area under the receiver operating characteristic curve (AUC) of the model is 0.56, indicating that the model has a 56% probability of accurately discriminating between a "Down" picture and an "Up" image.

The model's precision is 0.60, indicating that 60% of the "Up" movements were accurately identified by the model.

The model's recall rate is 0.62, indicating that 62% of the "Down" movements were accurately identified by the model.

The F1 score of the model is 0.61, representing a weighted average of the accuracy and recall metrics.

In general, the model performs the worst on the AMAZON stocks. There is much room for improvement in the model by enhancing both precision and recall.

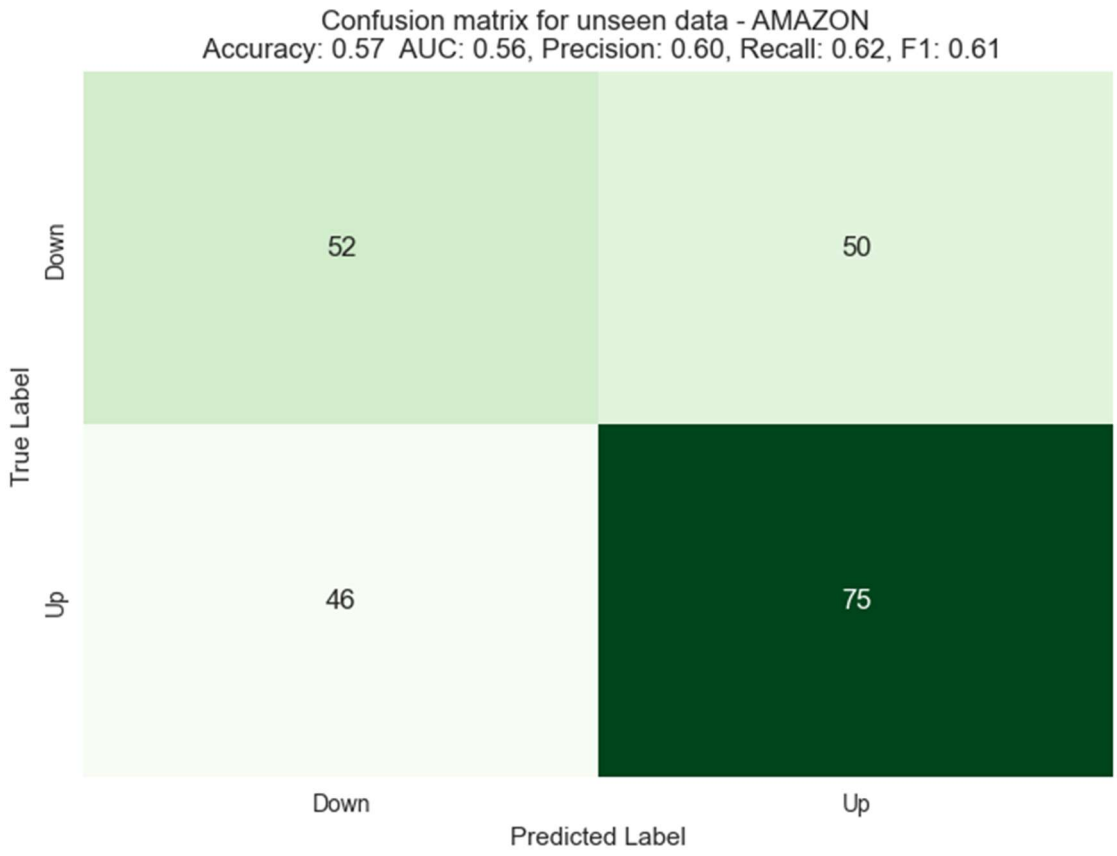



*Figure 12 Confusion Matrix for AMAZON*

### 5.1.1. Feature Importance and Explainability

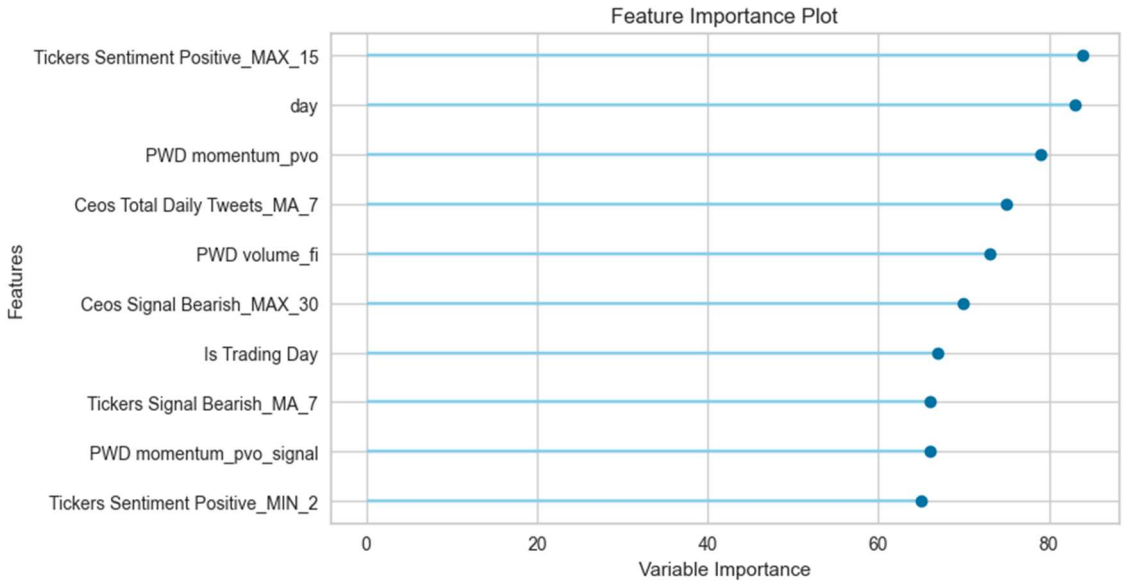
Feature importance scores in SHAP represent a quantification of the significance of each feature in influencing the prediction made by a machine learning model. The SHAP (SHapley Additive exPlanations) algorithm is employed to compute these values. This algorithm is a technique utilized to provide explanations for the predictions made by machine learning models (Nohara et al., 2019).

The SHAP feature significance scores are computed by evaluating the impact of individual features on the model's prediction for a given instance. The determination of the individual impact of each characteristic is computed through the utilization of a game theoretic methodology known as the Shapley value.

The Shapley value is a quantitative metric that assesses the significance of an individual participant inside a cooperative game. Within the realm of machine learning, the entities referred to as "players" may be understood as the distinct characteristics or attributes that constitute the model. Conversely, the term "game" denotes the specific objective of accurately forecasting the target variable.

The computation of the Shapley value for a given feature involves the comprehensive evaluation of all conceivable feature combinations and their respective impacts on the predictive performance of the model. The feature exhibiting the greatest Shapley value holds the utmost significance in determining the model's forecast.

The significance of feature importance scores in SHAP lies in their ability to provide insights into the inner workings of a machine learning model. Additionally, they possess the capability to assist in the identification of significant elements that contribute to the predictions made by the model. This information may be utilized to enhance the model's performance or to facilitate more informed decision-making based on the model's prognostications.




As shown in the table above the Sentiment Analysis features significantly contributed to the final model's predictions as well as some of the technical analysis features like the momentum of the previous working day its signal.

# 6. Discussion

This thesis encompasses a thorough examination aimed at assessing the feasibility and efficacy of utilizing machine learning techniques, namely the Light Gradient Boosting Machine (LightGBM), in the context of predicting stock market behavior. The inquiry involves the integration of Twitter sentiment analysis with financial data. The present chapter provides an analysis of the primary discoveries, their ramifications, constraints, and recommendations for further investigations.

## 6.1. Major Findings and Implications

The utilization of machine learning algorithms in the forecast of stock market trends has garnered significant attention in academic research. The findings of this study further support the efficacy and promise of employing these techniques. The utilization of pre-trained Transformer models, particularly RoBERTa and its fine-tuned iteration, enabled the extraction of sentiment analysis from StockTwits data. These sentiments were subsequently merged with financial data.

Tree-based models, particularly the Random Forest Classifier, outperformed all other models in almost every metric, including Accuracy and AUC. This establishes them as the most reliable algorithms for this specific predictive task. These models achieved Accuracy levels above 74% and AUC values above 0.84, indicating robust capabilities in differentiating between positive (UP) and negative (DOWN) stock movements.

Logistic Regression and Ridge Classifier models delivered moderate performances, with accuracy scores between 64-65% and less impressive AUC values. Interestingly, the Ridge Classifier improved its AUC value when exposed to the holdout set, suggesting it may be more balanced when dealing with unseen data.

K Neighbors Classifier and SVM with Radial Kernel displayed notable improvements in their AUC metrics on the holdout set, which could hint at their underappreciated generalizability during the training phase. On the contrary, the MLP Classifier lagged in performance across all metrics, suggesting a need for substantial tuning or redesign.

The decision to exclude non-trading days from the dataset aimed to prevent models from "learning the noise" rather than genuine market dynamics. This strategy seems to have paid off, as the top-performing models demonstrated strong predictive capabilities.

The use of custom-weighted measures in evaluating the Light Gradient Boosting Machine (LGBM) model was effective in accounting for class imbalance. Though its performance was not at the top, the consistency across metrics like Accuracy, AUC, and the weighted versions of Recall, Precision, and F1 score indicates moderate robustness and offers avenues for further optimization.

The Accuracy measure of 62.03% attained demonstrates a promising ability to predict future stock movements, taking into account the intricate nature and inherent unpredictability of financial markets. Although the predictive capability may not be deemed extremely robust, it nevertheless presents a helpful tool for forecasting that may be utilized in investing strategies.

The model's solid ability in distinguishing between positive and negative stock market movements is underscored by metrics like AUC, Recall, Precision, and F1 score. The model's ability to maintain consistent performance across several classes is seen from the Macro Precision, Macro Recall, and Macro F1 scores.

The analysis of feature importance underscores the significance of integrating sentiment methodologies and technical analysis indicators in enhancing predictive performance. This emphasizes the value of hybrid models that combine traditional financial analytics with social media sentiments.

## 6.2. Limitations

Though the results are promising, several limitations must be acknowledged. The achieved accuracy may still be prone to occasional incorrect predictions. This underscores the inherent complexities of financial markets and the challenges in attaining a fully reliable predictive model. The performance may also vary across different stock indices and time frames, which warrants further investigation.

## 6.3. Contributions

This thesis makes a distinct and valuable contribution to the current body of work on stock market forecasting through several means. The inquiry encompasses the use of seven various machine learning algorithms and the evaluation of their effectiveness across multiple situations, therefore enhancing the comprehension of predictive modeling in financial markets.

Additionally, the integration of pre trained Transformer models like as RoBERTa introduces a state-of-the-art aspect to sentiment analysis, showcasing the potential of sophisticated deep learning methods in extracting delicate subtleties from microblogging data. The use of a RoBERTa model that has been fine-tuned using stockwits data demonstrates the versatility of these models for certain needs.

Moreover, the integration of conventional technical analysis indicators with advanced machine learning algorithms highlights a comprehensive methodology for predicting stock market trends. The proposed combination of historical financial trends with present social media sentiment in this model contributes to its comprehensiveness, boosting the robustness and usefulness of the forecasts.

The aforementioned contributions represent a notable progression in the discipline, creating fresh avenues for scholarly investigation and real-world implementations that might have far-reaching consequences on investing tactics and financial decision-making.

# 7. Conclusions and Future Work

The study found that the combination of machine learning algorithms and sentiment analysis can improve the accuracy of stock market forecasting. The study also found that the use of pre-trained Transformer models can extract significant sentiment insights from microblogging sites. The proposed methodology represents a notable advancement in the field of financial forecasting and has the potential to be used to develop more accurate and reliable stock market forecasting models.

However, the study was limited by the use of a small dataset and the use of a single machine learning algorithm. Future research could use a larger dataset and a wider range of machine learning algorithms to improve the accuracy of the model. Future research could also investigate the use of other data sources, such as news and market movements, to improve the comprehensiveness of the model. Additionally, future research could conduct tests on different markets, sectors, and time periods to better understand the model's capacity to generalize and adapt to evolving market dynamics.

### 7.1.1. Conclusions

This thesis provides an in-depth exploration of the interplay between machine learning algorithms and sentiment analysis, particularly in the context of stock market forecasting. The methodology adopted was both extensive and meticulous, merging sentiment data obtained from Twitter with conventional financial metrics. By utilizing two pre-trained open-source Transformer models, RoBERTa and its fine-tuned variant using stockwits data, we have significantly enriched our sentiment analysis capabilities.

The study employed a well-structured pipeline for data collection, cleansing, and feature engineering, laying the groundwork for effective machine learning applications. Our evaluation involved multiple scenarios and utilized seven different machine learning algorithms, such as Light Gradient Boosting Machine (LGBM), Extra Trees Classifier, and Random Forest Classifier. Metrics like F1 score and AUC provided crucial insights into the models' predictive strengths.

Tree-based models consistently outperformed other algorithms, particularly the Random Forest Classifier, confirming their robustness in predicting stock movements. K Neighbors Classifier and SVM showed promise in their generalizability, while MLP

Classifier lagged behind in almost every metric. Special care was taken to ensure realistic evaluations, for example by excluding non-trading days and using custom-weighted measures.

Feature importance analysis revealed the significant role of sentiment data, along with technical analysis features, in contributing to the predictive power of the models, particularly in the final LGBM model. This resonates well with the study's broader aim to integrate machine learning and sentiment analysis in a meaningful manner for financial forecasting.

This work represents a milestone in financial forecasting research. It offers novel contributions in the form of an integrated approach that combines state-of-the-art machine learning techniques with sentiment data, extracted using advanced Transformer models. This melding of different data streams provides a robust basis for future developments in the field, aligning well with the complexities of modern financial markets.

### 7.1.2. Future Work

The findings of this study set the stage for numerous avenues for future research. Model refinement through fine-tuning or exploration of alternative machine learning algorithms has the potential to enhance predictive accuracy. There is also scope for expanding the feature set by incorporating additional data sources, such as market news or other market indicators, to develop a more holistic predictive model.

Moreover, extending the model evaluation to multiple markets, sectors, and time frames could offer insights into the model's adaptability and generalizability in changing market conditions. This could be particularly useful in understanding how well these models could be applied globally or in different economic climates.

In summary, the thesis presents a comprehensive approach that unites machine learning algorithms and sentiment analysis for stock market prediction. It not only advances the field theoretically but also offers practical insights that could have widespread implications. Future work in this area seems promising, and this study provides a solid foundation upon which to build further research and applications.

# Appendix

## Appendix A. Technical Analysis Indicators Glossary

| Abreviation | Indicator | Category |
|---|---|---|
| MFI | Money Flow Index | Volume |
| ADI | Accumulation/Distribution Index | Volume |
| OBV | On-Balance Volume | Volume |
| CMF | Chaikin Money Flow | Volume |
| FI | Force Index | Volume |
| EoM, EMV | Ease of Movement | Volume |
| VPT | Volume-price Trend | Volume |
| NVI | Negative Volume Index | Volume |
| VWAP | Volume Weighted Average Price | Volume |
| ATR | Average True Range | Volatility |
| BB | Bollinger Bands | Volatility |
| KC | Keltner Channel | Volatility |
| DC | Donchian Channel | Volatility |
| UI | Ulcer Index | Volatility |
| SMA | Simple Moving Average | Trend |
| EMA | Exponential Moving Average | Trend |
| WMA | Weighted Moving Average | Trend |
| MACD | Moving Average Convergence Divergence | Trend |
| ADX | Average Directional Movement Index | Trend |
| VI | Vortex Indicator | Trend |
| TRIX | Trix | Trend |
| MI | Mass Index | Trend |
| CCI | Commodity Channel Index | Trend |
| DPO | Detrended Price Oscillator | Trend |
| KST | KST Oscillator | Trend |
| Ichimoku | Ichimoku Kinkō Hyō | Trend |
| Parabolic SAR | Parabolic Stop And Reverse | Trend |
| STC | Schaff Trend Cycle | Trend |
| RSI | Relative Strength Index | Momentum |
| SRSI | Stochastic RSI | Momentum |
| TSI | True strength index | Momentum |
| UO | Ultimate Oscillator | Momentum |
| SR | Stochastic Oscillator | Momentum |
| WR | Williams %R | Momentum |
| AO | Awesome Oscillator | Momentum |

| | | |
|---|---|---|
| KAMA | Kaufman's Adaptive Moving Average | Momentum |
| ROC | Rate of Change | Momentum |
| PPO | Percentage Price Oscillator | Momentum |
| PVO | Percentage Volume Oscillator | Momentum |
| DR | Daily Return | Others |
| DLR | Daily Log Return | Others |
| CR | Cumulative Return | Others |

# References


Ajekwe, C. C. M., Ibiamke, A., & Haruna, H. A. (2017). Testing the Random Walk Theory in the Nigerian Stock Market. *IRA-International Journal of Management & Social Sciences (ISSN 2455-2267)*, *6*(3), 500. https://doi.org/10.21013/jmss.v6.n3.p15

AL-Rubaiee, H., Qiu, R., & Li, D. (2015). Analysis of the relationship between Saudi twitter posts and the Saudi stock market. *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, 660–665. https://doi.org/10.1109/IntelCIS.2015.7397193

Attigeri, G. V, Manohara Pai M M, Pai, R. M., & Nayak, A. (2015). Stock market prediction: A big data approach. *TENCON 2015 - 2015 IEEE Region 10 Conference*, 1–5. https://doi.org/10.1109/TENCON.2015.7373006

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, *16*(5), 412–424.

Barbieri, F., Espinosa Anke, L., & Camacho-Collados, J. (2022). XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond. *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 258–266. https://aclanthology.org/2022.lrec-1.27

Batra, R., & Daudpota, S. M. (2018). Integrating StockTwits with sentiment analysis for better prediction of stock price movement. *2018 International Conference on Computing, Mathematics and Engineering Technologies (ICoMET)*, 1–5. https://doi.org/10.1109/ICOMET.2018.8346382

Billah, M., Waheed, S., & Hanifa, A. (2016a). Stock market prediction using an improved training algorithm of neural network. *2016 2nd International Conference on Electrical,*

*Computer & Telecommunication Engineering (ICECTE)*, 1–4. https://doi.org/10.1109/ICECTE.2016.7879611

Billah, M., Waheed, S., & Hanifa, A. (2016b). Stock market prediction using an improved training algorithm of neural network. *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, 1–4. https://doi.org/10.1109/ICECTE.2016.7879611

Bohn, T. A., & Ling, C. (2017). *Improving Long Term Stock Market Prediction with Text Analysis Improving Long Term Stock Market Prediction with Text Analysis*. https://ir.lib.uwo.ca/etd://ir.lib.uwo.ca/etd/4497

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145–1159.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Buko Sabino. (2023). *A text analysis library for Python*. https://github.com/bukosabino/ta. (MIT License

Burton G. Malkiel. (1973). *A Random Walk Down Wall Street*.

Callender, J. C., & Osburn, H. G. (1977). A Method for Maximizing Split-Half Reliability Coefficients. *Educational and Psychological Measurement*, *37*(4), 819–825. https://doi.org/10.1177/001316447703700402

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 1–13.

Choudhry, R., & Garg, K. (2008). A Hybrid Machine Learning System for Stock Market Forecasting. *World Academy of Science, Engineering and Technology*, *39*.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, *20*(1), 37–46. https://doi.org/10.1177/001316446002000104

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, 233–240. https://doi.org/10.1145/1143844.1143874

Deepak, R. S., Uday, S. I., & Malathi, D. (2017). Machine learning approach in stock market prediction. *International Journal of Pure and Applied Mathematics*, *115*(8), 71–77.

Fakhry, B. (2016). *A Literature Review of the Efficient Market Hypothesis*. www.kspjournals.org

Falinouss, P. (2007). *MASTER'S THESIS Stock Trend Prediction Using News Articles A Text Mining Approach*.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

Gitrexx. (2022). *PLPPM_Sentiment_Analysis_via_Stocktwits*. https://github.com/Gitrexx/PLPPM_Sentiment_Analysis_via_Stocktwits/tree/main/SentimentEngine

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Gurjar, M., Naik, P., Mujumdar, G., & Vaidya, T. (2018). STOCK MARKET PREDICTION USING ANN. *International Research Journal of Engineering and Technology*. www.irjet.net

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

Hosmer, D. W., & Lemeshow, S. (2000). Applied Logistic Regression 2nd edn John Wiley & Sons. *Inc.: New York, NY, USA*, 160–164.

Huang, Y., Capretz, L. F., & Ho, D. (2021a). Machine Learning for Stock Prediction Based on Fundamental Analysis. *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 01–10. https://doi.org/10.1109/SSCI50451.2021.9660134

Huang, Y., Capretz, L. F., & Ho, D. (2021b). Machine Learning for Stock Prediction Based on Fundamental Analysis. *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 01–10. https://doi.org/10.1109/SSCI50451.2021.9660134

Hugging Face. (2022). *Twitter XLM-roBERTa* . Hugging Face. https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base

Januschowski, T., Wang, Y., Torkkola, K., Erkkilä, T., Hasson, H., & Gasthaus, J. (2022). Forecasting with trees. *International Journal of Forecasting*, *38*(4), 1473–1481. https://doi.org/10.1016/j.ijforecast.2021.10.004

Kordonis, J., Symeonidis, S., & Arampatzis, A. (2016). Stock Price Forecasting via Sentiment Analysis on Twitter. *Proceedings of the 20th Pan-Hellenic Conference on Informatics*, 1–6. https://doi.org/10.1145/3003733.3003787

Liu, S., Liao, G., & Ding, Y. (2018). Stock transaction prediction modeling and analysis based on LSTM. *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2787–2790. https://doi.org/10.1109/ICIEA.2018.8398183

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv Preprint ArXiv:1907.11692*.

Mittal, A., & Goel, A. (2012). *Stock Prediction Using Twitter Sentiment Analysis*.

Namdari, A., & Li, Z. S. (2018). Integrating Fundamental and Technical Analysis of Stock Market through Multi-layer Perceptron. *2018 IEEE Technology and Engineering Management Conference (TEMSCON)*, 1–6. https://doi.org/10.1109/TEMSCON.2018.8488440

Nohara, Y., Matsumoto, K., Soejima, H., & Nakashima, N. (2019). Explanation of machine learning models using improved shapley additive explanation. *Proceedings of the 10th*

*ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 546.

Pagolu, V. S., Reddy, K. N., Panda, G., & Majhi, B. (2016). Sentiment analysis of Twitter data for predicting stock market movements. *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 1345–1350. https://doi.org/10.1109/SCOPES.2016.7955659

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Ben Taieb, S., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Cyrino Oliveira, F. L., De Baets, S., Dokumentov, A., … Ziel, F. (2022a). Forecasting: theory and practice. *International Journal of Forecasting*, *38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Ben Taieb, S., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Cyrino Oliveira, F. L., De Baets, S., Dokumentov, A., … Ziel, F. (2022b). Forecasting: theory and practice. *International Journal of Forecasting*, *38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001

Picasso, A., Merello, S., Ma, Y., Oneto, L., & Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications*, *135*, 60–70. https://doi.org/10.1016/j.eswa.2019.06.014

Rasel, R. I., Sultana, N., & Hasan, N. (2016). Financial instability analysis using ANN and feature selection technique: application to stock market price prediction. *2016 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, 1–4.

Sakia, R. M. (1992). The Box-Cox Transformation Technique: A Review. *The Statistician*, *41*(2), 169. https://doi.org/10.2307/2348250

Shah, D., Campbell, W., & Zulkernine, F. H. (2018). A Comparative Study of LSTM and DNN for Stock Market Forecasting. *2018 IEEE International Conference on Big Data (Big Data)*, 4148–4155. https://doi.org/10.1109/BigData.2018.8622462

Sharaff, A., & Gupta, H. (2019). Extra-tree classifier with metaheuristics approach for email classification. *Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2018*, 189–197.

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. *Australasian Joint Conference on Artificial Intelligence*, 1015–1021.

Somani, P., Talele, S., & Sawant, S. (2014). Stock market prediction using Hidden Markov Model. *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, 89–92. https://doi.org/10.1109/ITAIC.2014.7065011

Teknologi MARA, U., Azri Mohd, M., & Nazrul Mohd Amin, M. (2017). *Testing the weak form of efficient market in cryptocurrency Insurance modeling View project Development of Stop-Loss Time-Discrete Markov Chain Model in Optimising Excess Claims for Medical and Health Insurance View project Saiful Reeza*. https://www.researchgate.net/publication/341536510

Valle-Cruz, D., Fernandez-Cortez, V., López-Chau, A., & Sandoval-Almazán, R. (2022). Does Twitter Affect Stock Market Decisions? Financial Sentiment Analysis During Pandemics: A Comparative Study of the H1N1 and the COVID-19 Periods. *Cognitive Computation*, *14*(1), 372–387. https://doi.org/10.1007/s12559-021-09819-8

Viswanath, P., & Hitendra Sarma, T. (2011). An improvement to k-nearest neighbor classifier. *2011 IEEE Recent Advances in Intelligent Computational Systems*, 227–231. https://doi.org/10.1109/RAICS.2011.6069307

Yeo, I.-K. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, *87*(4), 954–959. https://doi.org/10.1093/biomet/87.4.954

Yu, S., De Backer, S., & Scheunders, P. (2002). Genetic feature selection combined with composite fuzzy nearest neighbor classifiers for hyperspectral satellite imagery. *Pattern Recognition Letters*, *23*(1–3), 183–190. https://doi.org/10.1016/S0167-8655(01)00118-0