ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

CONTROVERSY PREDICTION IN TWITTER

Διπλωματική Εργασία

της

Αγγελικής Κωνσταντινίδου

Θεσσαλονίκη, Ιούνιος 2018

ΠΡΟΒΛΕΨΗ ΑΜΦΙΛΕΓΟΜΕΝΩΝ ΘΕΜΑΤΩΝ ΣΤΟ TWITTER

Αγγελική Κωνσταντινίδου

Πτυχίο Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας, 2013

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπουσα Καθηγήτρια
Γεωργία Κολωνιάρη

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25/06/2018

Γεωργία Κολωνιάρη	Ευαγγελίδης Γεώργιος	Σαμαράς Νικόλαος
.....	.....	.....

Αγγελική Κωνσταντινίδου

.....

Περίληψη

Η διαφωνία απόψεων είναι ένα σύνθετο θέμα και έχει προσελκύσει έντονα το ενδιαφέρον της ερευνητικής κοινότητας. Στα μέσα κοινωνικής δικτύωσης, η ανίχνευση αμφιλεγόμενων θεμάτων αποτελεί μεγάλη πρόκληση λόγω της πληθώρας των πληροφοριών που εκφράζονται από μεγάλο κοινό συμπεριλαμβανοντας απόψεις για την επικαιρότητα, δρώμενα και κάθε είδους ερεθίσματα. Η παρούσα εργασία εστιάζει στα αμφιλεγόμενα θέματα στο Twitter χρησιμοποιώντας μια μέθοδο βασισμένη σε ερωτήματα για ανάκτηση δεδομένων και προτείνει ένα μοντέλο πρόβλεψης που υπολογίζει το ενδεχόμενο ένα θέμα να αποτελέσει αντικείμενο διαμάχης στο μέλλον. Θεωρούμε το πρόβλημα της πρόβλεψης διαφωνίας ως ένα πρόβλημα δυαδικής ταξινόμησης και προτείνουμε ένα λογιστικό μοντέλο παλινδρόμησης για να προβλέψουμε αν ένα θέμα θα προκαλέσει διαφωνία μελλοντικά ή όχι. Μετά από ένα στάδιο προ-επεξεργασίας των tweets που έχουν συλλεχθεί, τα ταξινομούμε αρχικά στο πλαίσιο της ανάλυσης συναισθημάτων. Στο επόμενο στάδιο, μια μεγάλη γκάμα χαρακτηριστικών που εκφράζουν διαφορετικά γνωρίσματα των tweets, όπως γλωσσικές και χρονικές πληροφορίες, εξάγονται για να καλύψουν τους σκοπούς της παρούσας εργασίας. Προτείνουμε τη συγκέντρωση συνόλων tweets, αντί να εξετάζουμε κάθε tweet ξεχωριστά, και την εξαγωγή χαρακτηριστικών που είναι σημασιολογικά πιο πλούσια. Χρησιμοποιώντας τη λογιστική παλινδρόμηση, επιλέγονται τα στατιστικά σημαντικά χαρακτηριστικά και στη συνέχεια χρησιμοποιούνται για την δυαδική ταξινόμηση. Τα αποτελέσματα των πειραμάτων δείχνουν ότι το μοντέλο μπορεί να πετύχει 77% ακρίβεια και επιπρόσθετα ότι τα 5 επιλεγμένα στατιστικώς σημαντικά χαρακτηριστικά ενισχύουν την προσέγγιση μας.

Abstract

Controversy is a complex subject that has attracted the attention of research work in different fields. In social media, the detection of controversy is a big challenge due to the huge amount of information that is expressed by large audiences, containing opinions for news, events and any kind of stimulation. The current work focuses on controversy in Twitter using a query-based approach for data retrieval and proposes a prediction model which estimates the possibility for a topic to raise controversy in the future. We consider the problem of controversy prediction as a binary classification problem, and propose a logistic regression model to predict whether a topic is to become controversial or not. After pre-processing the collected tweets, they are classified in the context of sentiment analysis. Next, a variety of features expressing different characteristics of the tweets, such as linguistic and temporal information, are extracted for the purposes of our work. We propose aggregating sets of tweets, instead of considering each tweet separately, and extracting aggregated features that are semantically richer. Using logistic regression the statistically significant features are selected and used for the classification. Our experimental results show that the model can achieve 77% accuracy and that statistically significant features express different characteristics strengthening our approach.

Keywords: Controversy, prediction, Twitter, logistic regression, feature selection

Acknowledgments

I would like to thank my supervisor for her support and guidance during the elaboration of this work and also express my sincere gratitude to my family for being here for me.

Table of contents

List of Tables

List of Images

1 Introduction

1.1 Problem Statement

Nowadays, social media provide an impressive amount of data about users and their societal interactions, thereby offering computer and social scientists, economists, and statisticians, new opportunities for research exploration (Harald Schoen, 2013). Arguably, one of the most interesting lines of work is predictive analysis of future events and developments based on social media data, as recently have been seen in the areas of politics, finance, entertainment, market demands, health, etc. Predictive analysis on social media enables understanding and predicting the sentiment change of public opinions in the aforementioned areas and even more.

Controversial events are among the topics that attract the attention of researchers, in terms of provoking public discussion in which audience members express opposing opinions, surprise or disbelief (Ana-Maria Popescu, 2010). Due to the widespread adoption of social media, and the fact that much of the activity they host is publicly available, they offer a unique opportunity to study social phenomena such as peer influence, bias, and controversy (Kiran Garimella G. D., 2016). It could be said that almost any subject can be a source of controversy as it is impossible for everyone to agree on any subject. Moreover, people tend to argue about opinions for different entities at given periods of time, search for truth or common ground.

Several research works have been conducted regarding detecting controversy in Wikipedia and the Web. More specifically, a classification based method for automatic controversy detection for articles and categories in Wikipedia has been proposed in (Kazimierz Zielinski, 2017). A method for characterizing conflict in Wikipedia at global, article and user level was investigated in (Aniket Kittur, 2007). Different controversy models in Wikipedia were examined for their discriminative power, cost of learning and monotonicity condition in (Hoda Sperhri Rad, 2012). Other research works refer to detecting controversy in news articles (Yoonjung Choi Y. J.-H., 2010) and several works have examined detecting controversy in Twitter (Ana-Maria Popescu, 2010, Kiran Garimella G. D., 2016, Marco Pennacchiotti, 2010).

The continuous investigation that has been done in the area of detecting controversy causes the challenge of finding a way to predict controversy. The current methodology focuses on controversy in Twitter using query-based data and proposes a

prediction model which given a set of tweet features for a specific topic, it can provide the probability for raising controversy in the future. Our work is based on previous work done by (Ana-Maria Popescu, 2010) and (M. Pennacchiotti, 2010) in extracting features that could describe controversy and goes one step beyond the state-of-the-art by proposing an approach for predicting controversy, using a logistic regression model, thus considering the prediction problem as a classification problem.

1.2 Objectives

This thesis proposes a method for predicting controversy using query-based data derived from Twitter. The problem of controversy prediction is examined as a binary classification problem using a logistic regression model to predict whether will raise controversy or not. As a first step, we pre-process the collected tweets and then classify them in the context of sentiment analysis. A definition for controversy is introduced and used in order to characterize the collected subsets of tweets as controversial or non-controversial by taking into account the class of sentiment that was extracted for each tweet. Next, a variety of features expressing different characteristics of the tweets, such as linguistic and temporal information are extracted. We propose aggregating sets of tweets and extracting aggregated features that are semantically richer, instead of take each tweet separately. This way enables us take into account the sense of time and try to examine how long back in history we should go to make a prediction for a topic. Finally, by applying the logistic regression model, we select the most statistically significant features that are used for classification.

1.3 Structure of the Thesis

The rest of the thesis is structured as follows. A literature overview is presented in Section 2 giving the status of the research areas that the current thesis touches. The aforementioned areas have Twitter as their common denominator, as all of them use Twitter as their data source and cover the following topics: a) controversy and how it is detected 2.1 the concept of time in Twitter data 2.2 and the regression analysis as a method of prediction for Twitter data 2.3 3 presents the methods and techniques that have been used in the current thesis and covers the following topics: Twitter and the communication with it in Section 3.1 and 3.2 a state of the art in techniques of text pre-

processing in Section 3.3 an overview of sentiment analysis is presented in 3.4 emphasizing on machine learning classifiers. Feature extraction for controversy detection is examined in section 3.5 and finally, the logistic regression model which is the method used for prediction is described in Section 3.6 Section 4 presents the proposed methodology starting from the initial step of the communication with the Twitter API 4.1 moving to the creation of the dataset 4.2 and finally reaching the description of the proposed prediction model in Section 4.3 A list of experiments along with their results is presented in Section 5 covering the different tests that were conducted to verify our thesis. Finally, Section0contains the conclusions of the current thesis and some future plans, regarding improving the proposed approach.

2 Literature Overview

The explosion of social media has allowed researchers unprecedented access to data about the opinions of large audiences regarding political developments, popular culture events, etc. (Ana-Maria Popescu, 2010). Among topics discussed on social media, some of them spark more heated debates than others i.e. elections or healthcare than for example a music event (Kiran Garimella G. D., 2016). Exploring the topics of discussion on Twitter and understanding which ones are controversial is useful for a variety of purposes, such as for journalists to understand what issues divide the public, or for social scientists to understand how controversy is manifested in social interactions (Kiran Garimella G. D., 2016). Another purpose to study controversy is predicting real world outcomes and finding the correlations between the features of the social media data with the discussed controversial or non-controversial topics. Moreover, several attempts have been conducted to explore the wealth of social media data not only as reactive analytics tools but also as predictive tools (Le T. Nguyen, 2012) by taking into account the sense of time.

These three concepts of controversy, time and prediction are the keys which the current work is based on. Towards this direction, this section presents an overall overview of the research work that has been done focusing on Twitter data in the area of a) detecting controversy, b) the time which is the umbrella under which the controversy is examined and c) prediction with regression analysis, which is the method examined for modelling the proposed approach in this work.

2.1 Controversy in Twitter

An approximate definition for controversy is a strong disagreement among large groups of people for specific topics. Like the definition of relevance, it is possible that controversy should be defined operationally: whatever people perceive as controversial, is controversial (Dori-Hacohen, 2017). Another definition for controversy derives from a piece of text that lends itself to a query for a search engine and invokes conflicting sentiment or views (Yoonjung Choi, 2010).

Millions of bloggers participate in blogs by posting entries as well as writing comments expressing their opinions on various subjects, such as reviews on consumer products and movies, news, politics, etc. on online social media (i.e. Twitter, Facebook etc.), essentially providing a real-time view of opinions, intentions, and activities of

individuals and groups across the globe (Peter Gloor, 2009). The result of the continuing growth of information on online social media is the creation of a great amount of opinionated text, generated every day, which stimulates the research area of opinion mining. There is much progress in opinion mining techniques in recent years, and finding out contrastive arguments that are for or against a controversial issue is also a challenging task that has motivated many research works in detecting and predicting controversy (Jinlong Guo, 2015). The following paragraphs discuss the state of the art in detecting controversy in Twitter data.

Popescu et al. proposed a method for detecting controversial events using Twitter data (Ana-Maria Popescu, 2010). By assigning a controversy score to sets of tweets and ranking them according to this score, the authors managed to extract statistically significant performance in controversy detection using regression machine learning models. Similarly, Colleto et al. focused on detecting controversy in social media by exploiting network motifs (Mauro Coletto, 2017). The proposed approach catches antagonism in conversations, and allows dynamical discovery of potential controversial sub-discussions that may be present within otherwise non-controversial topics. It finally proved using a benchmark Twitter dataset, that the aforementioned motifs are more powerful in predicting controversy than other baseline used graph properties.

Pennacchiotti et al. worked on detecting controversies involving popular entities using Twitter data (Marco Pennacchiotti, 2010). The proposed method assigns a controversy score by combining a timely controversy and a historical controversy score. The former estimates the controversy of an entity by analyzing the discussion among Twitter's users at a given time period and the latter the overall controversy level of an entity in Web data, independently of time. The controversy in Twitter was also investigated by Garimella et al., who proposed a system that processes the daily trending topics discussed on Twitter and assigns to each topic a controversy score, which is calculated based on the user interactions (Kiran Garimella, 2016). Garimella also visualized the user interactions and allowed users to explore the sequence of tweets for each topic.

More specific investigation focusing on politic polarization on Twitter was performed in (Conover Michael, 2011). The study proved that the retweet network, in which users are connected iff one has re-broadcast content produced by another, exhibits a highly modular structure in contrast with the mention network, where users are

connected if one has mentioned another in a post, including the case of users replying to each other. Morales et al. proposed a methodology that can detect different degrees of polarization depending on the structure of the network, which was applied on Twitter data (Morales A., 2015). The study proposed a model to estimate opinions in which a minority of influential individuals propagate their opinions resulting in an opinion probability density function and an index to quantify the polarization of the distribution.

A systematic study of controversy detection on Twitter data was performed by K.Garimella et al.. The authors focused on the content and the network structure of the social media by a) building a conversation graph about a topic, b) partitioning the conversation graph to identify potential sides of controversy and c) measuring the amount of controversy characteristics of the graph (Kiran Garimella G. D., 2016). The study resulted in identifying that the proposed random-walk-based feature outperforms existing ones in detecting controversy and the content features are less helpful in this task. Smith et al. investigated the role of social media in discussing and debating controversial topics (Laura Smith, 2013). After applying sentiment analysis to classify the position expressed in a tweet about a controversial topic, they used the results in studying the user behaviour, resulting in that Twitter is primarily used as a means for rebroadcasting information and secondly as a means of communication with other users.

2.2 Time in Twitter Data

Time series is an ordered sequence of values of a variable at equally spaced time intervals (Introduction to Time Series Analysis). They are valuable for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for. The main usages of time series are:

- obtain an understanding of the underlying forces and structure that produced the observed data, the so-called "time series analysis",

- fit a model and proceed to forecasting, monitoring etc., the so-called "time series forecasting".

More specifically, time series analysis can be useful to see how a given asset, security or economic variable changes over time (Investopedia). It involves developing models that best capture or describe an observed time series in order to understand the

underlying causes. This field of study seeks the "why" behind a time series dataset. This often involves making assumptions about the form of the data and decomposing the time series into constitution components. The quality of a descriptive model is determined by how well it describes all available data and the interpretation it provides to better inform the problem domain.

Additionally, the prediction uses information regarding historical values and associated patterns to predict future activity (Investopedia). Most often, this relates to trend analysis, cyclical fluctuation analysis and issues of seasonality. Making predictions about the future is called extrapolation in the classical statistical handling of time series data. More modern fields focus on the topic and refer to it as time series forecasting. Forecasting involves taking models fit on historical data and using them to predict future observations. An important distinction in forecasting is that the future is completely unavailable and can only be estimated from what has already happened.

Several research works have been conducted in the area of Twitter by using time series for prediction purposes. The following paragraphs present the most notable methods proposed in the areas of stock market and public mood using Twitter data.

Si et al. proposed a stock prediction framework that is based on the characteristics of Twitter topics in the recent past (Jianfeng Si, 2013). The proposed model is a vector auto-regression model, a model that operates under the premise that past values have an effect on current values, for two time series, which are a) a sentiment time series and b) an S&P 100[1] stock market index time series, which are calculated from the prices of specific stocks. The results of the method are interesting for short periods and provide more training and testing points, as the model is trained by using a training and prediction process under sliding windows, instead of training in a specific period and predicting over another. Bollen et al. examined if public sentiment, derived from Twitter posts, can be used to predict the stock market (Bollen Johan, 2011). The public mood was extracted from tweets using the OpinionFinder[2] (OP) and the Google Profile of Mood States (GPOMS), which both analyze the text context of a tweet. The former provides positive and negative time series and the latter generates a six-dimensional daily time series of public mood (calm, alert, sure, vital, kind, and happy). The results showed that there was high correlation between specific public mood indicators and the stock

[1] https://us.spindices.com/indices/equity/sp-100

[2] http://www.cs.pitt.edu/mpqa/opinionfinderrelease/

market. The study in (Le T. Nguyen, 2012) presented a method of building statistical models from social media dynamics, derived from Twitter data, to predict the sentiment change toward particular products or brands at certain time in the future. It focuses on predicting the aggregated population sentiment ratio and its transformation through time by introducing a history window size, a prediction bandwidth and response time indicators, and examines how these parameters are related with the sentiment prediction.

An investigation to identify how public mood patterns, extracted from sentiment analysis of Twitter posts, are related with social, economic and other events was performed in (Johan Bollen, 2011). The study introduced the so-called POMS-ex, an extended version of the Profile Mood States (POMS), which is a psychometric instrument that measures six dimensions of mood (tension, depression, anger, vigour, fatigue, and confusion). By measuring the sentiment of each tweet using POMS-ex and using time series to express daily mood vectors, Bollen et al. compared the results to popular events of the same period and found that social, political and economic events are highly correlated with the public mood. B. O'Connor et al. tried to link text sentiment derived from Twitter posts to public opinion time series, which was derived from polls (Brendan O'Connor, 2010). In particular, measurements of aggregated textual sentiment using time series were compared to polling data, finding that in many cases there was high correlation between them. Moreover, the results highlighted the potential use of text streams as a substitute and supplement for traditional polling.

2.3 Prediction with Regression Analysis in Twitter

The current section discusses two methods of regression analysis, a) linear regression and b) logistic regression, applied in Twitter data for prediction purposes and mentions indicative examples of their application in the state-of-the-art.

2.3.1 Prediction with Linear Regression

Linear regression is a common method for describing the relation between predictors and outcome and in mathematical words it could be the method of approximating a mapping function (f) from input variables (X) to output variables (y). It is a very popular method mainly because of the interpretability of the parameters and has been widely used as a method for predicting the future behaviour of several areas using

Twitter data. These areas cover many topics such as TV series and Box-Office, public mood and elections, stock market, crime and health.

2.3.1.1 Prediction of TV Series and Box-Office

Several works have been conducted in the prediction of TV series and Box-Office. An indicative example is an application for forecasting TV ratings using Twitter data that was developed by Molteni et al. The authors collected 2.5 million tweets for USA TV series, classified them according to the sentiment using decision trees classifiers and clustered them based on the average audience (Molteni Luca, 2016). After applying linear regression, the method resulted in that there is an important correlation between audience size and tweets volume.

Similarly, Asur et al. investigated how Twitter data can be used to forecast box-office revenues for movies (Sitaram Asur, 2010). The research showed that social media feeds can be indicators for real-world performance. In particular, the rate at which movie tweets are generated was used to build a linear regression model for predicting movie box-office revenue, by using seven variables each corresponding to the tweet-rate for a particular day. Thus, the model provided an accurate prediction of movie performances.

2.3.1.2 Prediction of Public Mood and Elections

Linear regression was also used to predict people's opinions and trends by analyzing tweets in (Lee Hooyeon, 2011). The proposed method used as feature variables word frequencies and as target variables specific topics and concluded that the model benefits from larger datasets and a larger set of features. The method performs better prediction results than aggregated models with randomly picked words and it indicated that it is easier to perform near future prediction using Twitter data, which confirms that Twitter data is volatile.

Bermingham et al. attempted to model political sentiment in order to capture the voting intentions for the upcoming elections (Adam Bermingham, 2011). Their approach combined sentiment analysis using supervised learning in Twitter data and volume-based measures. By fitting a regression model, the study resulted in that the best method for predicting the result of the first preference votes in the elections is the share of volume of tweets that a given party received in total over the examined period.

2.3.1.3 Prediction of Stock Market

A study focused on finding the relation between micro blogging data for forecasting stock market variables was performed at (Nuvo Oliveira, 2013). The dataset that was used was collected from StockTwits[3], a social network service targeted in market communications. N. Oliveira et al. examined several regression models by indicating sentiment indicators and posting volume and found interesting results in models using larger datasets, concluding that predicting stock market is a very complex task.

Another attempt to identify relationships between Twitter based sentiment analysis of a company and its short-term market performance using Twitter dataset was performed at (Tushar Rao, 2012). More specifically, the results showed that negative and positive dimensions of public mood are significantly correlated with price movements of stocks. Using linear regression models in specific time windows, the research resulted in that monthly predictions have higher accuracy in predicting anomalies in the returns.

Moreover, an investigation regarding whether public sentiment as expressed in daily Twitter posts, can be used to predict the stock market was conducted in (Bollen Johan, 2011), as previously mentioned in Section 2.2 The results indicated that prediction accuracy is increased when certain mood dimensions are included in the linear regression model.

2.3.1.4 Prediction of Crime

Wang et al. presented a preliminary investigation of Twitter-based criminal incident prediction (Xiaofeng Wang, 2012). By applying automatic semantic analysis and understanding of natural language on Twitter data, they created a dataset. Dimensionality reduction was applied to the dataset, and the resulting data were fed to a generalized linear regression model. The results show that the model can forecast hit-and-run incidents uniformly across all days.

2.3.1.5 Prediction for spread of illness

The following references are derived from investigation that has been conducted in the area of predicting the spread of illness using Twitter data. Linear regression models were used to measure the correlation between patterns in Twitter messages and national health statistics (Cullota Aron, 2010). In particular, several models have been investigated to analyze Twitter posts in order to predict rates of influenza-like illness in a

[3] https://stocktwits.com/

population. The results presented that aggregating keywords frequencies using separate predictor variables outperforms aggregating keyword frequencies into a single predictor variable and it is more efficient to select keywords based on the residual sum of squares.

A similar investigation for achieving real time detection and prediction of spread of influenza epidemic was conducted in (Harshavardhan Achrekar, 2012). The authors applied text classification on the flu related tweets and correlated them with influenza-like illness (ILI) rates from Centres for Disease Control and Prevention[4] (CDC), which calculates them from data collected from sentinel medical practices. The results show that there is high correlation between them. The prediction model that was used was an auto-regression model with exogenous input, where the ILI rates were the realistic metrics of flu and the Twitter data were the real time assessment of the current condition.

2.3.2 Prediction with Logistic Regression

The second regression method that is examined is logistic regression. Logistic regression is a generalized linear regression method for learning a mapping from any number of numeric variables to a binary or probabilistic variable (David W. Hosmer, 2005). Mathematically, it could be the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y), thus finding a class or category for a given observation. Several works have been conducted applying logistic regression on Twitter data to predict. The aforementioned works refer to predicting popularity by means of retweet etc., crime and health.

2.3.2.1 Prediction for popularity

Naveed et. al proposed a logistic regression model to forecast for a given tweet its probability of being retweeted based on its contents (Nasir Naveed, 2011). Towards this direction, the authors analyzed a set of high-level and low-level content-based features on several collections of Twitter messages and resulted that a tweet is likely to be retweeted when it discusses a general, public topic instead of a narrow, personal topic.

A similar work that examined Twitter hashtag popularity was performed in (Zongyang Ma, 2013). The authors proposed methods to predict the popularity of new hashtags on Twitter by formulating the problem as a classification task. Among the classifiers that were used, the logistic regression model performed best and the final results show that the contextual features are more effective than content features.

[4] https://www.cdc.gov/

2.3.2.2 Prediction for crime

A method for predicting crime was examined in (Gerber Matthew, 2014). More specifically, a logistic regression model was implemented in the context of this work to predict the likelihood of a crime of specific type to occur. Their results showed that Twitter-derived features improved prediction performance for certain types of crime and did not for certain surveillance ranges.

2.3.2.3 Prediction for health

The potential of using Twitter to detect and diagnose major depressive disorder in individuals was investigated in (Munmun De Cloudhury, 2013). The authors used crowd sourcing to collect gold standard labels on a cohort's depression and proposed a set of social media measures to characterize depression. In the process of detecting the most useful features, logistic regression was used. However, the final classifier was an SVM classifier as it performed higher accuracy among others.

3 Background of Proposed Methodology

In this Section, some basic concepts are presented on the base of which the current work is developed. First of all, some information regarding Twitter and its streaming API is presented in Section 3.1 and 3.2 respectively. Next, in Section 3.3 3.3 , an overview of text pre-processing methods is presented along with the existing algorithms used to perform it. After pre-processing, a state-of-the-art in sentiment analysis of Twitter data is presented in Section 3.4 3.4 , emphasizing in machine learning classifiers. A feature extraction literature overview follows in Section 3.5 which focuses on existing research work in features used in controversy detection in Twitter data. Finally, the principles and the concept of the logistic regression model are presented in Section 3.6

3.1 Twitter

Twitter is a microblogging service, which is growing rapidly and used to spread recent happenings (Sitaram Asur, 2010). It can be considered a directed social network, where each user has a set of subscribers known as “followers”. Each user submits periodic status updates, the so-called “tweets” that consist of short messages of maximum size 140 characters. The aforementioned updates usually consist of personal information about the users, news or links to content such as images, video and articles and are displayed on the user’s profile page, as well as to his followers. It is also possible to send a direct message to another user, proceeding it by @userId indicating its destination. Moreover, posts that are made by one user that are forwarded by another user are called “retweets” empowering users to spread information of their choice beyond the reach of the original tweet’s followers (Haewoon Kwak, 2010).

3.2 Twitter Streaming API

The Twitter Application Programming Interface (API) (https://developer.twitter.com, 2018) currently provides a Streaming API and two discrete REST APIs. Through the Streaming API users can obtain real-time access to tweets in sampled and filtered form. The API is HTTP based, and GET, POST, and DELETE requests can be used to access the data.

Based on the Streaming API users can access subsets of public status descriptions in almost real time, including replies and mentions created by public accounts. An

interesting property of the streaming API is that it can filter status descriptions using quality metrics, which are influenced by frequent and repetitious status updates, etc. The API uses basic HTTP authentication and requires a valid Twitter account.

All Twitter APIs that return Tweets provide that data encoded using JavaScript Object Notation (JSON). JSON is based on key-value pairs, with named attributes and associated values. These attributes and their state are used to describe objects, such as tweets and users (Introduction to Tweet JSON, 2018). More specifically, each Tweet has: an author, a message, a unique ID, a timestamp of when it was posted, and sometimes geo metadata shared by the user. Each User has a Twitter name, an ID, a number of followers, and most often an account bio. With each Tweet "entity" objects are also generated, which are arrays of common Tweet contents such as hashtags, mentions, media, and links. If there are links, the JSON payload can also provide metadata such as the fully unwound URL and the webpage's title and description. So, in addition to the text content itself, a Tweet can have over 150 associated attributes (Introduction to Tweet JSON, 2018).

3.3 Text Pre-processing

Text mining is a technique which is used for extracting useful information from text data and finding patterns (Vijayarani Mohan, 2015). Text mining techniques are used in various types of research domains like natural language processing, information retrieval, text classification and text clustering. In this section, we focus on text pre-processing techniques of natural language text processing, a research area of Natural Language Processing (NLP), which explores how computers can be used to understand and manipulate natural language text (Jusoh Shaidach, 2007).

The idea of text pre-processing is to do some form of analysis or processing to natural language texts, so as the machine can understand, at least to some level, what the text means, says, or implies (PythonProgramming.net). This is an obviously massive challenge, but there are many research steps done towards this direction. The main idea is that computers simply do not understand words directly and there is a need to find a way to get as close to that as possible. The process of converting data to information that a computer can understand is referred as "pre-processing". In the next sections, some basic techniques of pre-processing such as tokenization, stop words removal and stemming, are presented.

3.3.1 Tokenization

The initial step of the analysis is breaking the text down into words, the so-called "tokenization". Tokenization is the process of breaking a stream of text into words, phrases, symbols, or other meaningful elements called "tokens" (Kannan S., 2014). The aim of the tokenization is the exploration of the words in a sentence. The list of tokens becomes input for further processing such as parsing or text mining.

3.3.2 Stop Word Removal

Stop word removal is the process of removing the words that do not give meaning to text, the so-called "stop words" (Vijayarani Mohan, 2015). In general, stop words are a division of natural language and should be removed from text during processing as they make the text look heavier and less important for analysts. The most common words in texts are considered as stop words such as articles, prepositions and pronouns etc. (i.e. the, in, a, an, with). The aforementioned words do not add any value to the context of a document and as a result they are not measured as keywords in text mining applications (Porter Martin, 1980).

There are four methods for stop word removal that are used to remove stop words from files (Porter Martin, 1980). The first method is the "classic method" which obtains stop words from pre-compiled lists and excludes them from the text (Jivani Anjali, 2011). The second method combines the "classic method" and Zipf's Law, according to which the frequency of any word is inversely proportional to its rank in the frequency table (Jivani Anjali, 2011, Sharma Deepika, 2012). More specifically, the method includes: removing most frequent words, words that occur once and words with low inverse document frequency. The third stop word removal method is the so-called "mutual information method", which is a supervised method that works by computing the mutual information between a given term and a document class, providing a suggestion of how much information the term can tell about a given class (Jivani Anjali, 2011, Sharma Deepika, 2012). The last method iterates over separate chunks of data, that are randomly selected and ranks the terms based on their format values using the Kullback-Leibler divergence measure, which measures how a probability distribution diverges from a another one (Jivani Anjali, 2011). The final stop word list is constructed by taking the least informative terms, removing all possible duplicates.

3.3.3 Stemming

The purpose of stemming is to remove various suffixes, reduce the number of words, have accurately matching stems, save time and memory space (Vijayarani Mohan, 2015). Stemming is based on two important points: firstly, words that do not have the same meaning should be kept separate and secondly, morphological forms of a word assumed to have the same base meaning should be mapped to the same stem. Stemming algorithms are grouped based on the approach they follow to: truncating, statistical and mixed methods.

Truncating methods remove suffixes and prefixes of a word. The most basic stemmer of this category is the Truncate stemmer which truncates a word at the n^{th} symbol (Vijayarani Mohan, 2015). Another algorithm is S-stemmer proposed by Donna Harman, that removes suffixes in plurals so as to convert them to singular forms (Harman Donna, 1991). Other known stemmers are the Lovins stemmer (Mladenic Dunja, 2002), which removes the longest suffix from a word, and the Porter stemmer (Porter Martin, 1980, 2001), which is the most popular stemming algorithm. The Porter stemming algorithm is based on the idea that the suffixes in English language are mostly made up of grouping smaller and simpler suffixes and it has five steps of rules, after the acceptance of each one of them, the suffix is removed and the final fifth step returns the final stem. Furthermore, the Paice/Husk stemmer is an iterative algorithm with one table containing about 120 rules indexed by the last letter of a suffix (Paice Chris, 1990). Another truncating stemming algorithm is the Dwason stemmer which is an extension of the Lovins approach (Sharma Deepika, 2012).

The stemmers, which are based on statistical methods, remove the affixes after implementing some statistical procedures. The mostly known statistical stemmers are N-Gram, HMM and YASS. The N-Gram stemmer is a language independent stemmer that uses a string-similarity approach to convert word inflation to its stem (Jivani Anjali, 2011, Sharma Deepika, 2012). The HMM stemmer is based on the concept of Hidden Markov Models (HMMs) which are finite-state automata where transitions between states are ruled by probability functions (Massimo Melucci, 2003). The last statistical stemming approach is YASS, whose name is an acronym for Yet another Suffix Striper and is based on clustering lexicons (Mladenic Dunja, 2004).

The mixed stemming methods are categorized to the following groups: a) inflectional and derivational, b) corpus based and c) context sensitive stemmers

(Vijayarani Mohan, 2015). More specifically, inflectional and derivational stemmers involve both inflectional and derivational morphology analysis and require large corpus in order to be developed. The inflectional analysis relates the word variants to the language specific syntactic variations such as gender, plural etc., whereas in derivational analysis the word variants are related to the part of speech of a sentence where the word occurs. An example of an inflectional analysis stemmer is Krovetz (Krovetz Robert, 1993) and an example that combines both techniques is the Xerox Inflectional and Derivational Analyzer (Vijayarani Mohan, 2015). The next mixed stemming category, the so-called "corpus based stemmers", refers to automatic modification of words that have resulted in a common stem, to suit the characteristics of a given text corpus using statistical methods (Sharma Deepika, 2012). Finally, context sensitive stemmers perform context sensitive analysis using statistical modelling on the query side unlike the usual method where stemming is performed before indexing a document and was initially proposed in (Funchun Peng, 2007).

3.4 Sentiment Analysis

Sentiment analysis aims to identify and extract opinions and attitudes from a given piece of text towards a specific subject (Sunny Kumar, 2016). This sentiment analysis process uses systematic ways to identify, extract and study affective states and subjective information. There has been much progress on sentiment analysis of conventional text which is usually found in open forums, blogs and the typical review channels (Bharat Naiknaware, 2017). The following paragraphs focus on the methods used for sentiment analysis, which are categorized in three groups based on the approach they follow: a) lexicon-based, b) machine-learning and c) hybrid methods. Emphasis is given to the machine learning based approaches, as they are used in the current work.

3.4.1 Lexicon Based Classifiers

Lexicon-based approaches determine the sentiment or polarity of opinion via some function of opinion words in the document or the sentence and they can vary according to the context in which they are created (Sarlan Aliza, 2014). Ding et al. proposed a lexicon-based approach by exploiting external evidence and linguistic conventions of natural language expressions in order to handle opinion words that are context dependent (Ding Xiaowen, 2008). A lexicon-based method to determine whether the opinion expressed in a product review is positive or negative was proposed in (Hu

Minqing, 2004) and a very similar approach in (Kim Soo-Min, 2004). The Semantic Orientation Calculator was introduced in (Taboada Maite, 2010), which uses dictionaries of words annotated with their semantic orientation and is used in the polarity classification task to capture the text's opinion towards its main subject matter.

3.4.2 Machine Learning Classifiers

Machine learning-based approaches typically rely on classification approaches where sentiment detection is considered as a binary class (positive, negative) (Sarlan Aliza, 2014). Most techniques use some form of supervised learning by applying different learning techniques that need manual labelling of training examples for each application domain. The most commonly used supervised learning techniques are Naive Bayes, Maximum Entropy and Support Vector Machines.

3.4.2.1 Naive Bayes

The Naive Bayes classifier is based on Bayes' theorem (Russel Stuart 2003) and assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature, which is the so-called the Naïve Bayes assumption or independence assumption. Under this assumption, the classifier chooses the most likely label for an input. More specifically, it finds the probability of a given set of inputs for all possible values of a class variable and selects the output with maximum probability (Bo Pang, 2002).

In sentiment analysis, the approach is to assign to a document *d* the class *c,* which has the values of positive and negative. The Naïve Bayes classifier is derived by observing the Bayes' rule:

$$P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)},$$

where *P(d)* plays no role in selecting *c*. To estimate the term *P(d|c)*, Naive Bayes decomposes it, as follows:

$$P_{\mathrm{NB}}(c \mid d) := \frac{P(c)\left(\prod_{i=1}^{m} P(f_i \mid c)^{n_i(d)}\right)}{P(d)},$$

by assuming that the f_i's features are conditionally independent given d's class, where $\{f_1,\ldots,f_m\}$ is a predefined set of m features, $n_i(d)$ is the number of times f_i occurs in

document *d* and *d* is represented by the document vector $d=(n_1(d), n_2(d),...n_m(d))$. The Naïve Bayes classifier has been used in many notable research works regarding sentiment analysis (Hao Wang, 2012, Alexander Pak, 2010, Alec Go, 2009, Hassan Saif, 2012).

3.4.2.2 Maximum Entropy

The Maximum Entropy classifier is a probabilistic classifier which belongs to the class of exponential models, whose probability distribution follow a Poisson process (Bo Pang, 2002). Unlike the Naive Bayes classifier, Maximum Entropy does not assume that the features are conditionally independent of each other. It is based on the Principle of Maximum Entropy and from all the models that fit the training data, selects the one which has the largest entropy. Its estimate of $P(c|d)$ takes the following exponential form:

$$P_{\mathrm{ME}}(c \mid d) := \frac{1}{Z(d)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c)\right)$$

where $Z(d)$ is a normalization function, $F_{i,c}$ is a feature/class function for feature f_i, $\lambda_{i,c}$'s are feature-weight parameters and class c, defined as follows:

$$F_{i,c}(d,c') := \begin{cases} 1, & n_i(d) > 0 \text{ and } c' = c \\ 0 & \text{otherwise} \end{cases}$$

The maximum entropy classifier has been used in sentiment analysis indicatively in (Alec Go, 2009) and (Adam Berger, 1996).

3.4.2.3 Support Vector Machines (SVM)

Support Vector Machine performs classification tasks by constructing hyperplanes, that are subspaces whose dimension is one less than that off its ambient space, in a multidimensional space that separates cases of different class labels (Bo Pang, 2002). The basic idea for SVM in sentiment analysis is a) to find a hyperplane through the training procedure, represented by vector w, that not only separates the document vectors in one class from those in the other, and b) find for which separation or margin, this hyperplane is as large as possible. This search corresponds to a constraint

optimization problem, letting c_j that belongs to {1,-1} (corresponding to positive and negative) be the correct class of document d_j, the model can be written as:

$$\vec{w} := \sum_j \alpha_j c_j \vec{d}_j, \quad \alpha_j \geq 0,$$

where the a_j's are obtained by solving a dual optimization problem. Those d_j such that a_j is greater than zero are called support vectors, since they are the only document vectors contributing to vector *w*. Classification of the test instances consists simply of determining which side of *w*'s hyperplane they fall on. Support vector machines have been used for sentiment analysis in (Alec Go, 2009) and (Dr.Balasaravanan.K, 2018).

3.4.3 Hybrid Classifiers

There are also some approaches that utilize both the opinion words/lexicon and the machine learning approach, the so-called "hybrid" approaches. For example, Wiebe et al. (Wiebe Janyce, 2005) used a subjectivity lexicon to identify training data for supervised learning for subjectivity classification. L. Zhang et al. proposed a hybrid method for sentiment analysis in Twitter data (Lei Zhang, 2015). The method first adopted a lexicon-based approach to perform entity level sentiment analysis, then added more tweets that are likely to be opinionated, and finally, a classifier was used to assign polarities to the entities of the newly identified tweets. A hybrid scheme for sentiment classification was also proposed in (Songbo Tan, 2008). The authors first used a lexicon-based technique to label a portion of examples and then a supervised classifier trained on the labeled ones and applied this classifier to the task.

3.5 Feature Selection for Controversy Detection

A literature overview is presented in this section regarding existing approaches in feature selection towards detecting controversy in Twitter. A method for detecting controversial events in social media was proposed in (Ana-Maria Popescu, 2010). Two sets of features were used in the attempt to detect controversy: a) twitter-based features including linguistic, structural, business, sentiment and controversy characteristics and b) external features including news buzz and web-news controversy. It also worth mentioning, the creation and use of a controversy lexicon which contained 750

controversial words derived from Wikipedia controversial topic list and a bad words lexicon of 687 English bad words.

Colleto et al. detect potential controversy in social media, by examining if network motifs and other baseline features such as structural features in user-interaction, propagation-based and temporal features can predict controversy (Mauro Coletto, 2017). An attempt to detect controversies involving popular entities was performed in (Marco Pennacchiotti, 2010). The research investigated Twitter snapshots by computing a controversy score, which was calculated by combining a timely controversy and a historical controversy score as mentioned in section 2.1 The method used a) a sentiment lexicon composed by assigning polarity to OpinionFinder terms and b) a controversy lexicon, which was built by collecting possible controversial words from Wikipedia pages of Wikipedia controversial topic list.

Garimella et al. proposed a method for detecting controversy, as mentioned in section 2.1 applying the following steps: a) creation of the retweet graph, b) portioning of the graph and finally c) measuring controversy by computing the value of a random-walk-based controversy measure (Kiran Garimella G. D., 2016). Mejova et al. demonstrated that controversial issues in news can be characterized by the use of fewer positive words and a greater presence of negative words (Yelena Mejova, 2014). This finding was verified in different media sources and confirmed with four different sentiment lexicons: a) Affective Norms for English Words[5] (ANEW), b) General Inquirer[6,] c) MicroWNOp[7] a list of WordNet synsets (Cerini S., 2007) and d) SentiWordNet[8] (Baccianella Stefano, 2010). Moreover, a bias-specific lexicon was used the so-called Bias Lexicon[9], containing a list of 654 bias-related lemmas.

3.6 Logistic Regression Model

Logistic regression is a case of Generalized Linear Models, which is a framework for modelling response variables that are bounded or discrete (David W. Hosmer, 2005). As previously mentioned in section 2.3.2, logistic regression is a statistical method for

[5] http://csea.phhp.ufl.edu/media/anewmessage.html

[6] http://www.wjh.harvard.edu/~inquirer/homecat.htm

[7] http://www-3.unipv.it/wnop/

[8] http://sentiwordnet.isti.cnr.it/

[9] http://www.mpi-sws.org/~cristian/Biased_language.html

analyzing a dataset in which there are one or more independent variables that determine the dependent variable which is discrete and can have two or more categorical levels.

For categorical variables it is less appropriate to use linear regression because the response values are not measured on a ratio scale and the error terms are not normally distributed (Czepiel Scott, 2002). Moreover, the linear regression model can generate as predicted values any real number ranging from negative to positive infinity, whereas a categorical variable can take on a limited number of discrete values with specific range. For these reasons, specific modelling techniques have been developed that can be used for categorical variables, in a way analogous to that in which linear regression is used for continuous variables.

Linear regression models equate the expected value of the dependent variable to a linear combination of independent variables and their corresponding parameters. In contrast to that, generalized linear models equate the linear component to some function of probability of a given outcome of the dependent variable. In logistic regression, the logit transform is the function that is used, which is the logarithm of the odds that an event will occur. Mathematically, logistic regression with one independent variable can be expressed as follows:

$$\text{logit}(p(x)) = \log(p(x)/1-p(x)) = \beta_0 + \beta_1 x \,, \quad (6)$$

where, $p(x)$ is the probability that the dependent variable equals a case, given some linear combination of predictors. The formula for $p(x)$ illustrates that the probability of the dependent variable equalling a case is equal to the value of the logistic function of the linear regression expression. This shows that the value of the linear regression expression can vary from negative to positive infinity and after the transformation, the resulting expression for the $p(x)$ ranges between 0 and 1. Moreover, β_0 is the intercept from the linear regression equation, which is the value of the criterion when the predictor is equal to zero and $\beta_1 x$ is the regression coefficient multiplied by some value of the predictor.

The parameters in linear regression are estimated using the method of least squares by minimizing the sum of squared deviations of the predicted values from observed values. In the case of logistic regression, the coefficients are calculated with the Maximum-likelihood estimation (Czepiel Scott, 2002) as least squares cannot produce minimum variance unbiased estimators for actual parameters. The intuition for

maximum-likelihood for logistic regression is that a search procedure seeks values for the coefficients that minimize the error in the probabilities predicted by the model to those in the data.

4 Methodology

The proposed methodology aims at deriving a prediction model, which given a specific set of tweet features for a topic, it can provide the topic's probability for raising controversy in the future. This section aims to give an overall overview of the followed approach and describe the final proposed model and the features that seem to be statistically significant for predicting controversy.

4.1 Query-based Data Collection from Twitter

Our main goal was to collect tweets that contained specific keywords or hashtags referring to different topics, from which we could extract features after certain pre-processing. In order to do so, we need to have access to Twitter data, thus an app was created that interacts with the Twitter API. The first prerequisite to create this app was creating a Twitter account, on behalf of which the app is registered. After creating and successfully logging in to the account, the app is registered at http://apps.twitter.com providing a name and description. Then, a consumer key and a consumer secret are received, which are confidential and need to be kept private and provide read-only permissions.

A Python-based client, the so-called "Tweepy"[10], was installed to establish communication with the Twitter API. After installing Python 3.5, the following command is executed in order to install Tweepy:

pip install tweepy==3.3.0.

Moreover, the OAuth interface needs to be used in order to authorize the app to access Twitter on behalf of the created account. Table 1 shows the snippet of code that ensures access to Twitter using the corresponding credentials.

Table 1 Tweepy Python client

```
import tweepy
from tweepy import OAuthHandler

consumer_key = 'CONSUMER-KEY'
consumer_secret = 'CONSUMER-SECRET'
access_token = 'ACCESS-TOKEN'
access_secret = 'ACCESS-SECRET'
```

[10] https://github.com/tweepy/tweepy

```
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)
```

The Streaming API of Tweepy is used in order to gather all the upcoming tweets about particular events. In order to be able to customize the way we process the incoming data, the StreamListener() is extended. Depending on the search term, a lot of tweets can be gathered within a few minutes. The following table (Table 2) shows the sample of code that enables streaming using Tweepy and gathers tweets for the search term #trump and stores its object in a JSON file.

Table 2 Streaming using Tweepy

```
from tweepy import Stream
from tweepy.streaming import StreamListener

class listener(StreamListener):

    def on_data(self, data):
        try:
            with open('python.json', 'a') as f:
                f.write(data)
                return True
        except BaseException as e:
            print("Error on_data: %s" % str(e))
        return True

    def on_error(self, status):
        print(status)
        return True

twitter_stream = Stream(auth, MyListener())
twitter_stream.filter(track=['#trump])
```

4.2 Data Preparation

After establishing the communication with the Twitter API, hundreds of tweets are gathered for a variety of topics forming initial dataset. The current section describes in detail the sequence of steps performed in order to finalize the dataset which trained the

prediction model. These steps in brief are: a) pre-processing the collected sets of tweets, b) performing sentiment analysis per topic and c) extracting the features that will be fed to the prediction model.

4.2.1 Text Pre-processing

The pre-processing of the dataset is implemented using the NLTK[11] library, which is a powerful tool for working in computational linguistics using Python. After installing Python the NLTK library is installed using the following command:

pip install nltk.

The first step of pre-processing is to tokenize the collected data using sent_tokenize from the nltk.tokenize module of NLTK. The next step is removing the stop words, by using the list of stop words from the module nltk.corpus of NLTK. Finally, stemming is applied using one of the most popular stemming algorithms, Porter stemmer, which is embedded in NLTK by importing PorterStemmer from nltk.stem.

4.2.2 Sentiment Analysis

NLTK is also used for classifying the collected tweets and extracting their polarity, as it contains a list of available modules for classifiers which are the ConditionalExponentialClassifier, DecisionTreeClassifier, MaxentClassifier, NaiveBayesClassifier and the WekaClassifier. For the specific purposes of this work, the Naive Bayes classifier is used for characterising the collected tweets as positive or negative. It should be mentioned that in the current work, it is assumed that all tweets are opinionated, and neutral tweets are not taken into account. For the training of the classifier, a dataset of Amazon reviews is used, as it proved to be very difficult to find up-to-date labelled Twitter datasets, that could be used for the purposes of this work.

In the context of this work, we introduce a definition for controversy that enables characterizing a subset of tweets as controversial or non-controversial. A factor δ=0.05 is defined indicating a threshold for the maximum percentage of difference between positive and negative tweets, under which a topic is defined as controversial. More specifically, if the absolute difference between positive and negative tweets, divided by the total number of tweets, is less than the δ factor, then the corresponding topic is considered controversial. In the context of this work, the subsets whose tweets are completely controversial or non-controversial were taken into account, given the fact that

[11] https://www.nltk.org/

the public opinion agrees with the result that was extracted by the classification and the hypothesis of the δ factor.

4.2.3 Feature Extraction

A list of features are selected to create the initial dataset based on the work done in (Ana-Maria Popescu, 2010), as previously referred in Section 3.5 In several cases, we encountered similarities among the research works that have been conducted till now, so the method proposed by Popescu et al. was the most complete and became the one on which the current work is based and gave impulses for further improvements. Going one step beyond the state-of-the-art, the features were aggregated in sets of tweets making the dataset richer semantically than examining each tweet separately.

The following table (Table 3) illustrates the features along with their description and category. The source categories which features derived from are structural, linguistic, sentiment, twitter-based and time. Structural features refer to characteristics that are related with: a) parts of the speech, i.e. average of nouns, average of verbs, average of personal pronouns, b) punctuation i.e. average of emphasis punctuation and hashtags, i.e. average of hashtags. Twitter-based features derive from the information we get from the Twitter API i.e. average of replies, average of retweets, average of retweets count and average of replies count. Linguistic features are related with the meaning that each word carries, i.e. average of bad words and average of controversial words. Towards this direction, a bad words lexicon of 460 words was downloaded from the Web[12] and a *controversial words lexicon* was created with 3240 words, derived from Wikipedia pages of people mentioned in the Wikipedia *controversial topic list,* similarly with (Ana-Maria Popescu, 2010). Moreover, sentiment features are depicting the polarity of the tweets i.e. average of polarity and last but not least, the time features represent the relation of the tweets with time. In our case, average of time difference shows the difference in the time of creation between the collected tweets

.

Table 3 Features for Controversy Prediction

	Name	Description	Category
1	Average of nouns (an)	$an_n = (\sum_{i=1}^{n} nt_i/tt_i)/n$, where:	Structural

[12] https://en.wiktionary.org/wiki/Category:English_swear_words , http://www.noswearing.com/dictionary/c , https://www.ofcom.org.uk/__data/assets/pdf_file/0023/91625/OfcomQRG-AOC.pdf

		nt_i : the number of noun tokens of tweet *i* tt_i : the number of total tokens of tweet *i*	
2	Average of verbs (av)	$av_n = (\sum^n_{i=1} vt_i/tt_i)/n$, where: vt_i : the number of verb tokens of tweet *i* tt_i is the number of total tokens of tweet *i*	Structural
3	Average of personal pronouns (app)	$app_n = (\sum^n_{i=1} ppt_i/tt_i)/n$, where: ppt_i : the number of personal pronouns tokens of tweet *i* tt_i is the number of total tokens of tweet *i*	Structural
4	Average of emphasis punctuation (aep)	$aep_n = (\sum^n_{i=1} ept_i/tt_i)/n$, where: ept_i : the number of emphasis punctuations tokens (?,!) of tweet *i* tpt_i is the number of total punctuation tokens of tweet *i*	Structural
5	Average of hashtags (ah)	$ah_n = (\sum^n_{i=1} ht_i/tt_i)/n$, where: ht_i : the number of hashtag tokens of tweet *i* tt_i is the number of total tokens of tweet *i*	Structural
6	Average of Retweet counts (arc)	$arc_n = (\sum^n_{i=1} rc_i)/n$, where: rc_i : the retweet count of tweet *i*	Twitter
7	Average of Favourite counts (afc)	$afc_n = (\sum^n_{i=1} fc_i)/n$, where: rc_i : the favorite count of tweet *i*	Twitter
8	Average of retweets (ar)	$ar_n = (\sum^n_{i=1} r_i)/n$, where: r_i : the boolean (1,0) measure which indicates if tweet *i* is a	Twitter

		retweet	
9	Average of replies (arep)	$arep_n = (\sum^n_{i=1} rep_i)/n$, where: rep_i : the boolean (1,0) measure which indicates if tweet *i* is a reply tweet	Twitter
10	Average of polarity (apol)	$apol_n = (\sum^n_{i=1} pol_i)/n$, where: pol_i : the boolean (1,0) measure which indicates if tweet *i* has negative polarity	Sentiment
11	Average of bad words (abw)	$abw_n = (\sum^n_{i=1} bwt_i/tt_i)/n$, where: bwt_i : the number of bad words tokens of tweet *i* (extracted using the bad words lexicon) tt_i is the number of total tokens of tweet *i*	Linguistic
12	Average of controversial words (acw)	$acw_n = (\sum^n_{i=1} cwt_i/tt_i)/n$, where: cwt_i : the number of controversial word tokens of tweet *i* (extracted using the controversial words lexicon) tt_i is the number of total tokens of tweet *i*	Controversy/ Linguistic
13	Average of Time Difference (atd)	$ph_n = (\sum^n_{i=1} td_i - td_{i-1})/n$, where: $td_i - td_{i-1}$: the time difference between tweet *i* and twee *i-1*	Time

The extracted features were stored in a csv file where each line contained the features of each tweet. Each line started with the sample's classification regarding the class of controversy (controversial or non-controversial) of the topic, followed by the extracted features separated by comma. The aforementioned file was used to train and test the classifier.

4.3 Controversy Prediction Model

Although logistic regression has not been widely used on Twitter data, the nature of the problem of predicting a binary and not a continuous variable indicates the use of logistic regression, thus formulating the problem as a classification problem. Therefore, logistic regression is chosen as the prediction model for classifying a tweet topic as controversial or not. Our goal is to examine multiple features of the tweet dataset so as to determine the most significant features that yield the best classification results, and therefore derive the best prediction model using these selected features.

To this end, the prepared dataset is used to fit a logistic regression model using as parameters for the model all the features extracted for the collected dataset, as described in the previous section. The logistic regression model can be expressed as follows:

$$\text{logit}(p(X)) = \log(\, p(X) / 1 - p(X)) = \beta_0 + \sum_i^n \beta_i X_i \ ,$$

where *logit(p(X))* denotes the controversy to be predicted and $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}$ and X_{13} represent the features of Table 3, respectively. The β values correspond to the regression coefficients.

A list of libraries in python are installed and used in order to train and test the prediction model. More specifically, *Pandas*[13] is installed in order to read the csv file using the following command:

pip install pandas

The *Skikit-learn*[14] package, a tool for data mining and data analysis, is installed in order to create the logistic regression model using the following command:

pip install skikit-learn.

[13] https://pandas.pydata.org/

[14] http://scikit-learn.org/stable/index.html

5 Experimental Results

This section contains all the experiments performed that lead to the proposed model. Several combinations and features are examined in order to verify the actual results. This section starts with an overview of the metrics that are used in the following sections for validating the results. Next, a list of experiments follows, examining the sense of time in the current dataset using the proposed model. Finally, a list of experiments is presented that combines features from different categories together with the corresponding results.

The total subsets of tweets resulted in 2 controversial topics (champions' league, Trump's travel ban) and 4 non-controversial topics (earth day, mother's day, new year's eve, Russian plane crash). The following table shows the number of collected tweets per topic and its characterization based on our definition for controversy.

Table 4 Dataset Description

Topic	Number of Tweets	Controversy
Trump's travel ban	535	Controversial
Champion league match	439	Controversial
Earth Day	353	Non-controversial
Mother's Day	229	Non-controversial
2018 New Year's Eve	308	Non-controversial
Russian plane crash	340	Non-controversial

5.1 Validation of Predicted Values

The metrics used to evaluate the output of the experiments are a) precision, recall, f1-score and support, which are included in a classification report and represented also in a diagram and b) the ROC curve diagram, which is a common metric of a classifier's predictive quality.

5.1.1 Cross Validation Classification Report

The classification report[15] is provided by the *Skikit-learn* library and includes metrics that are used to evaluate the accuracy of the model in classification problems. The aforementioned metrics are the precision, recall, f1 score and queue rate or support.

[15] http://scikit-learn.org/stable/modules/model_evaluation.html

More specifically, precision denotes the proportion of true positives that are correctly real positives. The recall or sensitivity refers to the true positive rate, which is the number of instances the positive (first) class that actually were predicted correctly. The harmonic mean of precision and recall calculates the f1 score, which is a useful metric for comparing classifiers. Finally, the support shows the number of occurrences of each class in the predicted values.

Additionally, a visualization of these measures with respect to the discrimination threshold of the classifier is presented, which is the probability at which the positive class is chosen over the negative class. If the probability (score) is greater than some discrimination threshold then the positive class is selected, otherwise, the negative class is selected. The discrimination threshold is generally set to 50% but the threshold can be adjusted to increase or decrease the sensitivity to false positives or to other application factors.

5.1.2 ROC Curve

The Receiver Operating Characteristic (ROC) curve[16] is a measure of a classifier's predictive quality that compares and visualizes the tradeoff between the models' sensitivity and specificity. The term sensitivity refers recall, as previously referred. Specificity refers to the true negative rate, which is the number of instances from the negative (second) class that were predicted correctly.

In the following section, the diagrams that are used to present the ROC curve, contain also information regarding the area under the curve (AUC), which is the computation of the relationship between false positives and true positives.

5.2 Group Tweets Sorted by Time

In the current section, we demonstrate a list of experiments in an attempt to examine the effect of time in the proposed model and present the corresponding results. In order to include time in the model, the initial dataset is sorted by time and the features are calculated by grouping tweets together. All the models that are examined in this section use all the calculated features and the results of each iteration are reported in the form of a classification report, a corresponding diagram and a ROC curve diagram.

[16] http://scikit-learn.org/stable/modules/model_evaluation.html

5.2.1 Test Initial Dataset

The fist experiment was performed with the dataset in its initial form without taking into account the concept of time, using the rest of the extracted features from all sets of tweets in a csv file (n=1). Table 5 shows the results of the first experiment, where we can see that precision is equal to recall in average. Moreover, the accuracy of the model is very close to the accuracy extracted from 10-fold cross validation experiment with difference almost 0.0.2 and the support has total of 440 tweets-

Table 5 Experimental Results – n=1

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.64	0.75	0.69	239
True (controversial)	0.63	0.50	0.56	201
Avg/total	0.64	0.64	0.63	440
Summary	**Accuracy of logistic regression classifier on test set**: 0.64 **Mean Absolute Error**: 0.36 **10-fold cross validation average accuracy**: 0.663			

Figure 3 is the visualization of precision, recall, f1 score, and queue rate with respect to the discrimination threshold of the classifier. In our case, the discrimination threshold is calculated to 0.40.

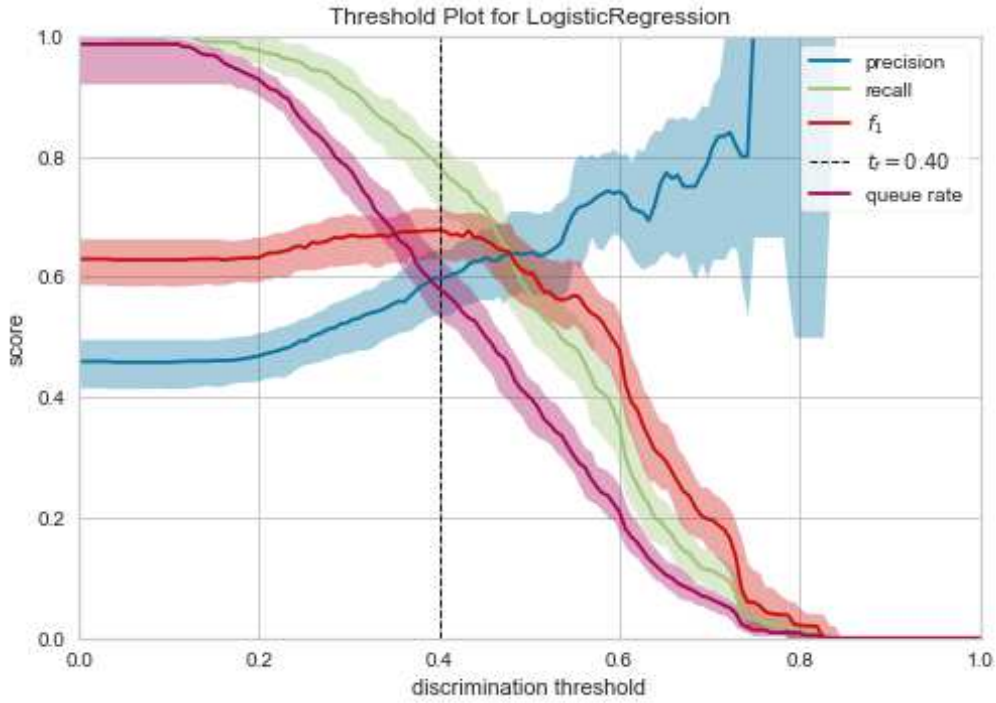

Figure 1 Threshold Plot for Logistic Regression 5.2.1

In the following image (Figure 2), the ROC curve displays the true positive rate on the Y axis and the false positive rate on the X axis on average basis. The ideal point is therefore the top-left corner of the plot: false positives are zero and true positives are one. Moreover, AUC reaches 0.63. The diagonal red line represents a random classifier as a baseline for comparison, so points above the diagonal represent good classification results (better than random), points below the line represent poor results (worse than random).

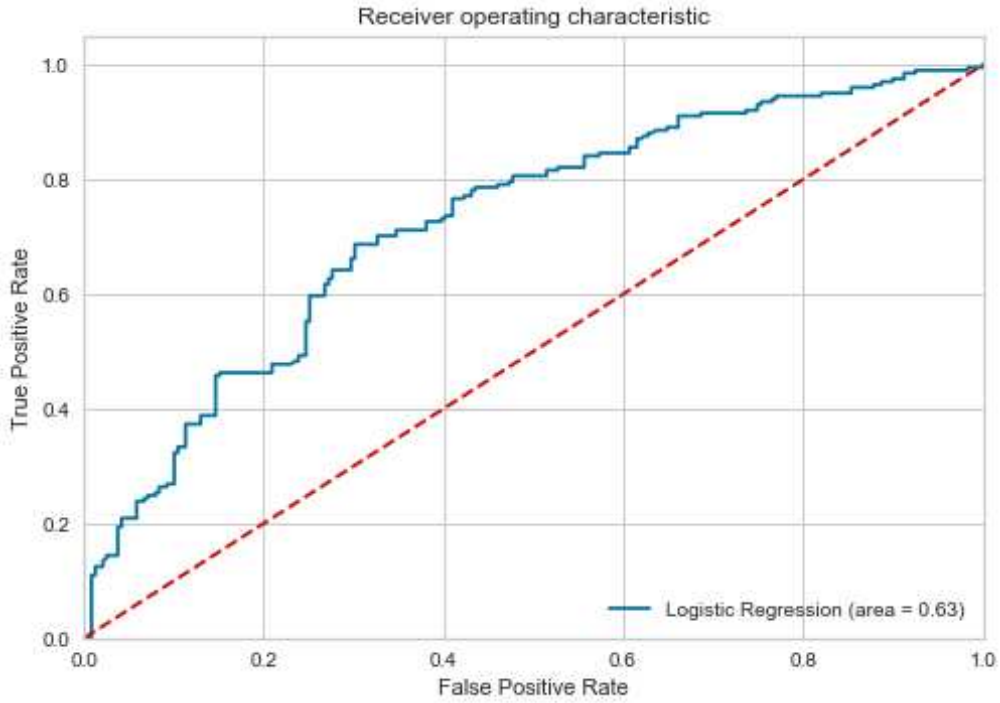

Figure 2 ROC Curve for Logistic Regression model 5.2.1

5.2.2 Test Aggregated Information in Groups of 2 Tweets

In the current experiment, time is taken into account, trying to compare the collected tweets in pairs of tweets. All the features are included in the model and they are calculated for n=2. In this experiment, the results seem to have been improved and reached the highest accuracy of all the examined tests. More specifically, the precision is almost equal to recall, reaching the percentage of 0.78 and 0.77, correspondingly. The support has 146 tweets and the accuracy of logistic regression reaches the 77% which is higher than the accuracy of the previous experiment which was 0.64.

Table 6 Experimental results – n=2

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.83	0.64	0.73	70
True (controversial)	0.73	0.88	0.80	76
Avg/total	0.78	0.77	0.76	146
Summary	**Accuracy of logistic regression classifier on test set**: 0.77 **Mean Absolute Error**: 0.23 **10-fold cross validation average accuracy**: 0.717			

All the aforementioned metrics are represented in the following image (Figure 3). The discrimination threshold in this case is calculated at 0.44 and it is higher than the threshold of the experiment with the initial dataset.

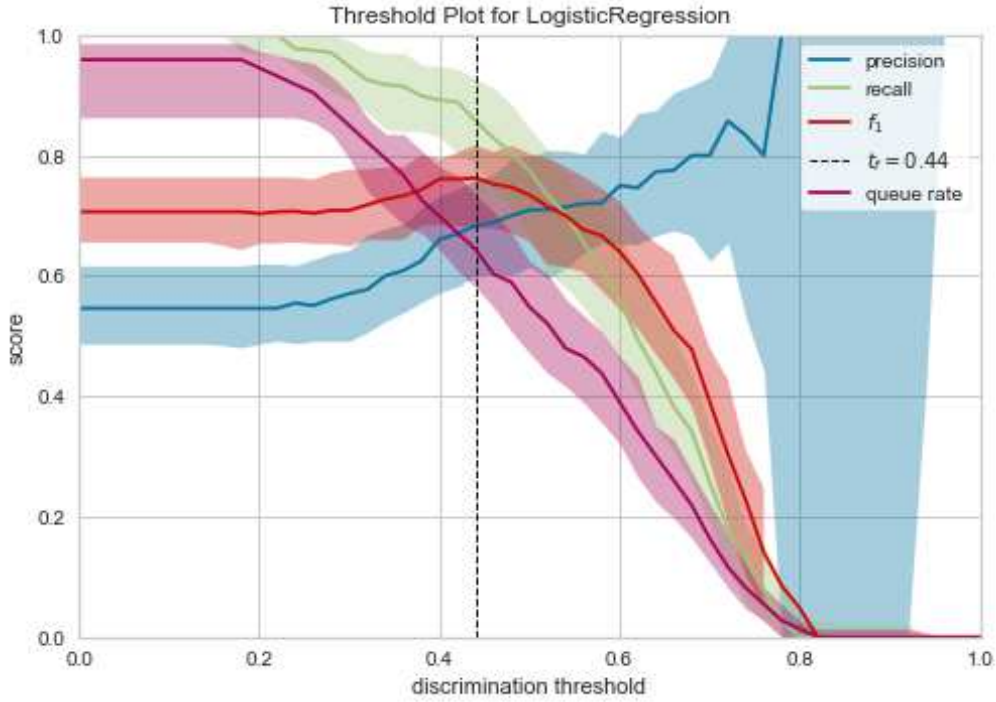

Figure 3 Threshold Plot for Logistic Regression 5.2.2

In the following image (Figure 4), the ROC curve of the experiment is displayed. The area under the curve (AUC) reaches 0.76, which is higher than the AUC calculated in the experiment of the initial dataset.

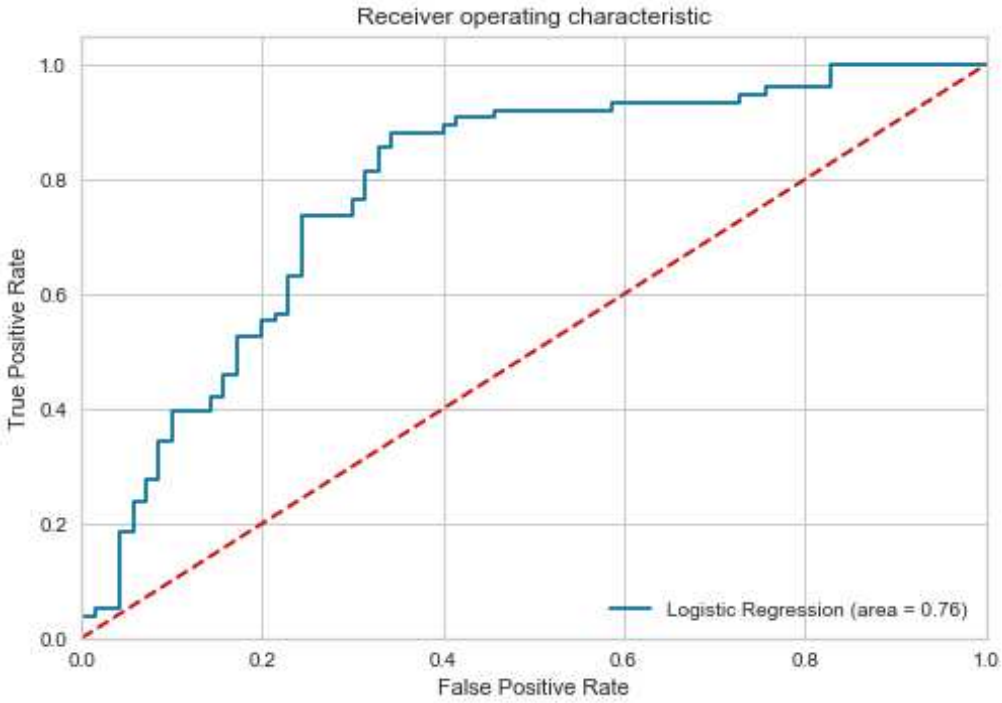

Figure 4 ROC Curve of Logistic Regression Model 5.2.2

5.2.3 Test Aggregated Information in Groups of 5 Tweets

In case 3 experiment, the dataset was grouped in teams of 5 tweets, as the grouping in 2 tweets provided better results than not modifying the dataset at all. The features were recalculated as averages of 5 tweets and the time was taken into account, in grouping tweets that were in series. The following table (Table 7) illustrates the results, which were poorer in contrast to the previous experiment, with accuracy 0.64, whereas the experiment of aggregating tweets in sets of 2 tweets reaches 0.77. Moreover, the precision and the recall are lower too and have almost the same average around 0.65. The support in this experiment is 73 tweets. We could say that the poorer performance in this experiment derives from the reduction of the dataset.

Table 7 Experimental Results – n=5

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.53	0.57	0.55	28
True (controversial)	0.72	0.69	0.70	45
Avg/total	0.64	0.65	0.64	73
Summary	**Accuracy of logistic regression classifier on test set**: 0.64 **Mean Absolute Error**: 0.36			

	10-fold cross validation average accuracy: 0.69

Figure 5 is the visualization of precision, recall, f1 score, and queue rate with respect to the discrimination threshold of the classifier. In the current case, the discrimination threshold is calculated to 0.40 which is lower comparing it with the threshold of the previous example, which was 0.44 as presented in Figure 3.

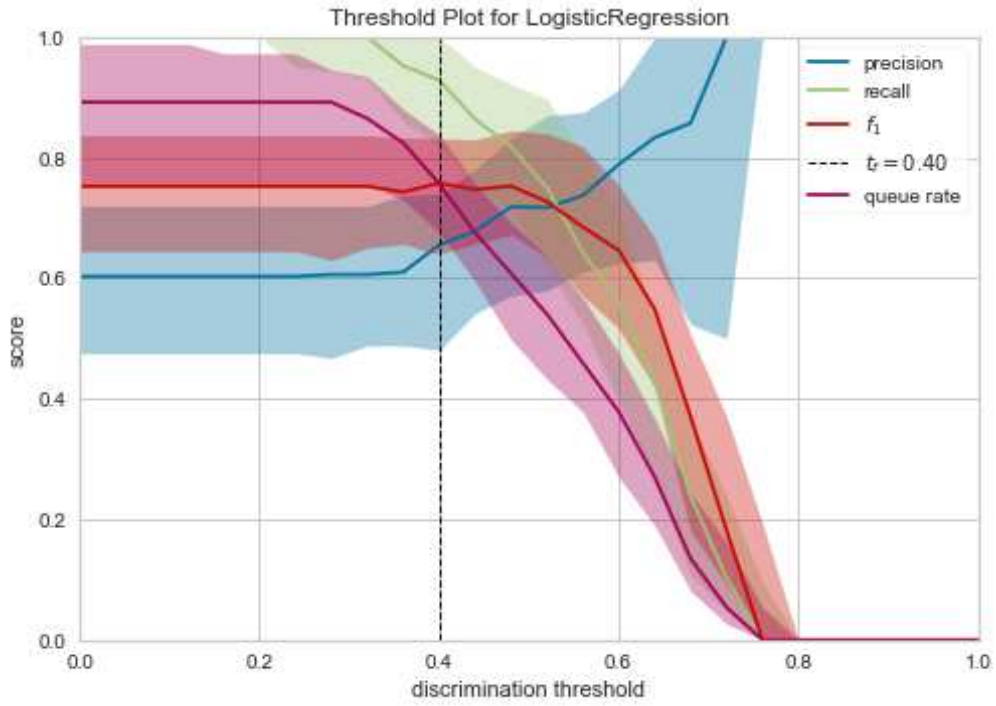

Figure 5 Threshold Plot of Logistic Regression 5.2.3

The ROC curve of the current experiment is presented in Figure 6. Moreover, the area under the curve (AUC) is presented and reaches the percentage of 0.63 which is lower than the experiment conducted aggregating the information in sets of 2 tweets, which was 0.76.

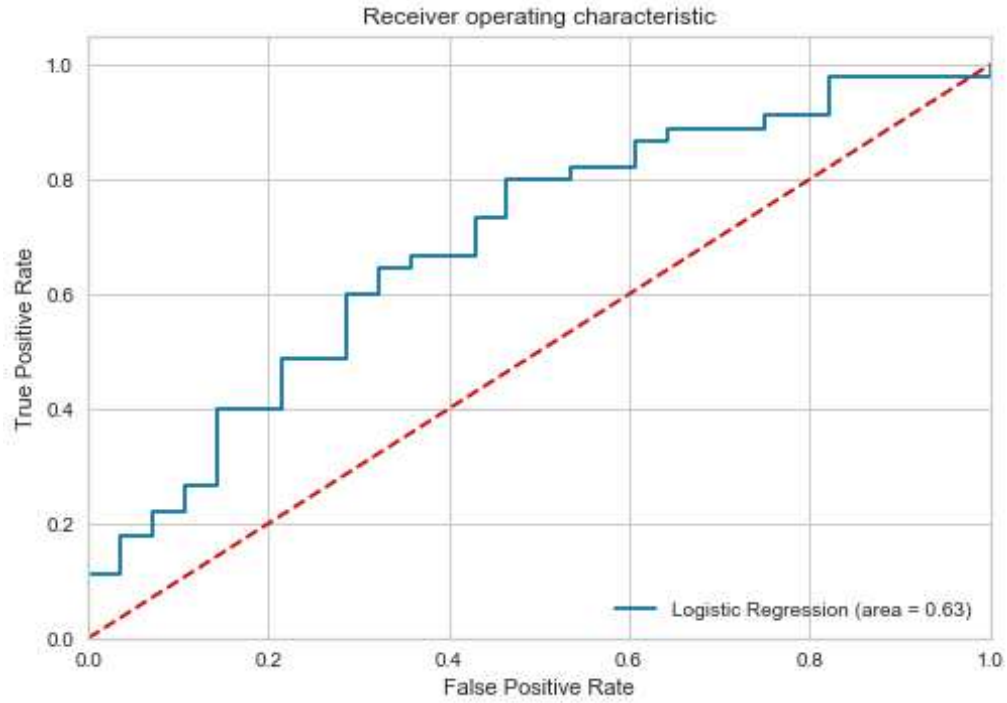

Figure 6 ROC Curve of Logistic Regression Model 5.2.3

5.2.4 Test Aggregated Information in Groups of 10 Tweets

Moving on with increasing the number of tweets that are grouped to examine the performance, the case 4 illustrates the results of grouping the dataset to team of 10 tweets and taking the time into account as the tweets were grouped after being sorted. The features were calculated as average metrics. The results of this experiment are presented in Table 8. We notice that precision and recall are quite similar and the accuracy reaches 0.71, which is higher than the previous experiment with groups of 5 tweets, but lower than the second experiment where the accuracy is 77%. The support is very low at 41 tweets. Similarly with the previous experiment, the results of this experiment are probably poor because the dataset was even more reduced.

Table 8 Experimental Results – n=10

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.79	0.68	0.73	22
True (controversial)	0.68	0.79	0.73	19
Avg/total	0.74	0.73	0.73	41
Summary	**Accuracy of logistic regression classifier on test set**: 0.71			

	Mean Absolute Error: 0.29 **10-fold cross validation average accuracy**: 0.68

Figure 7 is the visualization of precision, recall, f1 score, and queue rate with respect to the discrimination threshold of the classifier. The discrimination threshold is 0.38 and it is lower than the discrimination threshold of the previous experiment in Figure 5.

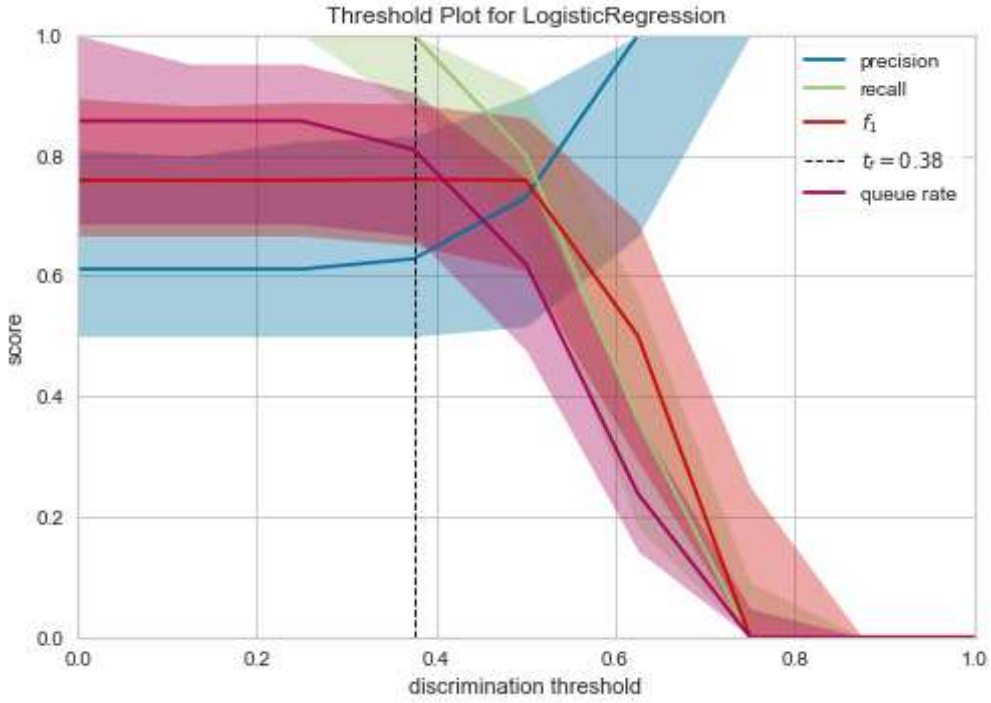

Figure 7 Threshold Plot of Logistic Regression 5.2.4

Figure 8 displays the ROC curve of the experiment, where the area under the curve (AUC) reaches 0.70, which is higher than the previous experiment where AUC reaches 0.64 but lower than the second experiment where AUC reaches 0.76.

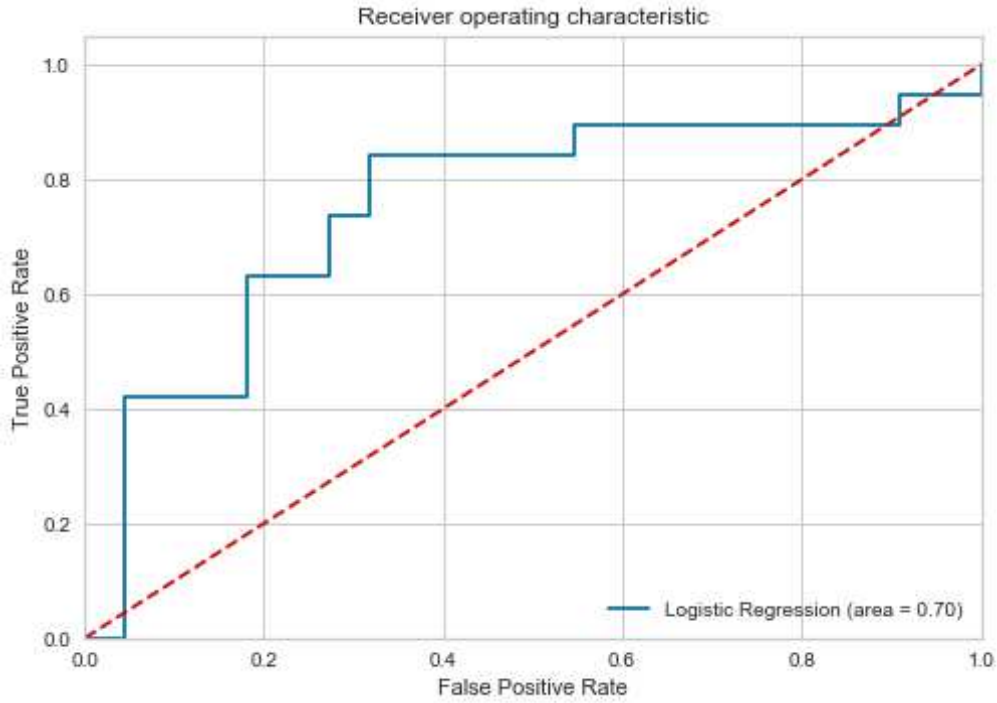

Figure 8 ROC Curve of Logistic Regression Model

5.2.5 Summary

From the cases presented previously, it is obvious that the best performance is achieved while grouping per 2 tweets the sorted dataset (Section 5.2.2 so this is the proposed model and the model from which the most statistically significant features derive. In this experiment, the accuracy reaches ~0,77 and the most statistically important features are the average of bad words, the average of replies, the average of retweets, the average of polarity and the average of personal pronouns, having the lowest p values ($p < 0.05$).

The final logistic regression results that achieved the highest accuracy with n=2 are included in Table 9. The first column shows the coefficients and the second column is the standard error. The z value in the third column indicates the coefficient divided by the standard error and the fourth column is the p-value for each term, which tests the null hypothesis that the coefficient is equal to zero (no effect). A low p-value (< 0.05) indicates that the null hypothesis can be rejected, thus the predictor is a meaningful addition to the model because changes in the predictor's value are related to changes in the response variable. The results of the training indicated that the following features: average of bad words, average of personal pronouns, average of replies, average of retweets and average of polarity are significantly correlated with controversy as they exhibit the lowest p values. The features: average of favorite count, average of retweet

count, are excluded from training the model as they cause the problem of "singular matrix", which is a common problem that means that the determinant of the matrix is zero and is caused due to dependencies among the variables. In our case, this probably occurred due to the values of these features in the dataset, that are almost zero.

Table 9 Logistic Regression Results of Proposed Model

	Coeff.	**Std.err**	**z**	**P> |z|**
averageOfNouns	-1.3943	0.809	-1.724	0.085
averageOfVerbs	-0.1704	2.573	-0.066	0.947
averageOfEmphasisPunctuation	68.0825	45.136	1.420	0.156
averageOfHashtags	1.2826	2.116	0.606	0.544
averageOfPersonalPronouns	-12.7391	1.369	-9.305	**0.000**
averageOfReplies	2.8234	0.590	4.785	**0.000**
averageOfRetweets	0.9350	0.137	6.832	**0.000**
averageOfBadWords	-10.1498	3.858	-2.631	**0.009**
averageOfControversialWords	-1.5094	1.652	-0.914	0.361
averageOfPolarity	0.9767	0.241	4.053	**0.000**
averageOfTimeDifference	-0.1667	0.199	-0.839	0.402
Summary	**Accuracy of logistic regression classifier**: 0.77 **Mean absolute error**: 0.232			

The final model is formulated as follows, taking into account only the statistically significant variables:

$$\text{logit}(p) = -1.086 - 12.7391*X_3 + 0.935*X_8 + 2.8234*X_9 + 0.9767*X_{10} - 10.1498*X_{11},$$

where X_3, X_8, X_9, X_{10}, X_{11} are the significant features: average of pronouns, average of retweets, average of replies, average of polarity and average of bad words, correspondingly.

The results of the 10-fold cross validation of the regression model are also verified by training the regression model using Weka[17](Hall M., 2009), a machine learning software written in Java developed at the University of Waikato, New Zealand.

[17] https://www.cs.waikato.ac.nz/ml/weka/

5.3 Combination of Features

The experiments of this section are based on combination of features of different sources on the dataset of groups of 2 tweets, where the highest accuracy is achieved, to verify that the most significant features have been selected for the current prediction model. The logistic regression analysis report is presented together with the classification report for each experiment for analysing them. The source of the features has been presented in Table 3 of section 4.2.3 In the following experiments, the sentiment and time features participate in all the examined cases.

5.3.1 Combination of Structural, Sentiment and Time Features

The first test of this category combines structural, sentiment and time features. The following table (Table 10) shows the result of this experiment. The statistically significant features are the average of verbs, average of hashtags, average of personal pronouns and average of polarity which have p-values lower than 0.05 as presented in the table below. We notice that from the most significant features in this test, only average of personal pronouns and average of polarity are included in the final proposed model.

Table 10 Logistic Regression Results 5.3.1

	Coeff.	**Std.err**	**z**	**P> |z|**
averageOfNouns	0.7936	0.737	1.077	0.281
averageOfVerbs	7.4282	2.352	3.159	**0.002**
averageOfEmphasisPunctuation	62.9946	41.934	1.502	0.133
averageOfHashtags	5.6149	1.847	3.039	**0.003**
averageOfPersonalPronouns	-10.0446	1.187	-8.464	**0.000**
averageOfPolarity	1.1689	0.223	5.239	**0.000**
averageOfTimeDifference	-0.2858	0.187	-1.527	0.127

Table 11 shows the results of the 10-fold cross validation in the examined case where the accuracy reaches the percentage of 0.68. We can see also that precision and recall reach the same percentage at 0.66 for both metrics.

Table 11 Results of 10-fold Cross Validation of 5.3.1

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.69	0.51	0.59	70

True (controversial)	0.64	0.79	0.71	76
Avg/total	0.66	0.66	0.65	146
Summary	**Accuracy of logistic regression classifier on test set**: 0.66 **Mean Absolute Error**: 0.34 **10-fold cross validation average accuracy**: 0.68			

5.3.2 Combination of Linguistic, Sentiment and Time Features

The next experiment combines the linguistic features that have been extracted together with the sentiment and time features. As presented in Table 12, the significant feature seems to be the average of bad words and the average of polarity. Both these features are included in the final proposed model.

Table 12 Logistic Regression Results 5.3.2

	Coeff.	**Std.err**	**z**	**P> |z|**
averageOfBadWords	-17.2622	3.396	-5.083	**0.000**
averageOfControversialWords	0.7545	0.158	4.765	0.893
averageOfPolarity	0.2061	1.535	0.134	**0.000**
averageOfTimeDifference	-0.3357	0.173	-1.936	0.053

In the following table, the results of the 10-fold cross validation seem to be poorer in contrast with the previous tests, with the accuracy reaching the percentage of 0.517. The precision and recall metrics reach the 0.48 and 0.49 correspondingly, which are very poor, too. These results are expected as we few features and we have already noticed that the 2 of them (average of bad words and average of polarity) have been proved to be significant in previous tests.

Table 13 Results of 10-fold Cross Validation of 5.3.2

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.45	0.24	0.31	70
True (controversial)	0.51	0.72	0.60	76
Avg/total	0.48	0.49	0.46	146
Summary	**Accuracy of logistic regression classifier on test set**: 0.49 **Mean Absolute Error**: 0.51 **10-fold cross validation average accuracy**: 0.517			

5.3.3 Combination of Twitter based, Sentiment and Time Features

The combination of twitter based, sentiment and time features is examined in this experiment. This combination performs better than the previous combination (5.3.2). The significant features are the average of replies, the average of retweets and the average of time difference.

Table 14 Logistic Regression Results 5.3.3

	Coeff.	**Std.err**	**z**	**P> |z|**
averageOfReplies	1.5096	0.492	3.069	**0.002**
averageOfRetweets	0.3409	0.084	4.077	**0.000**
averageOfPolarity	-0.2348	0.186	-1.260	0.208
averageOftimeDifference	-0.5347	0.176	-3.036	**0.002**

The accuracy in this experiment reaches the percentage of 0.645 with the final results presented in the table that follows. In this case, we can see that precision and recall perform better, reaching almost the 70% in average, performing better than the previous experiment where precision and recall nearly reached 50%.

Table 15 Results of 10-fold Cross Validation of 5.3.3

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.73	0.57	0.64	70
True (controversial)	0.67	0.80	0.73	76
Avg/total	0.70	0.69	0.69	146
Summary	**Accuracy of logistic regression classifier on test set**: 0.69 **Mean Absolute Error**: 0.308 **10-fold cross validation average accuracy**: 0.645			

5.3.4 Combination of Structural, Twitter based, Sentiment and Time Features

The current experiment combines structural, twitter based, sentiment features and time features. The results (Table 16) show that the significant features are the average of nouns, the average of personal pronouns, the average of replies, the average of retweet

and the average of polarity. It is expected that this experiment outperformed the rest in this category as it combines more features from different categories.

Table 16 Logistic Regression Results 5.3.4

	Coeff.	**Std.err**	**z**	**P> |z|**
averageOfNouns	-1.5788	0.798	-1.979	**0.048**
averageOfVerbs	-0.2044	2.569	-0.080	-0.937
averageOfEmphasisPunctuation	49.900	42.100	1.185	0.236
averageOfHashtags	0.7447	2.101	0.354	0.723
averageOfPersonalPronouns	-13.6271	1.345	-10.135	**0.000**
averageOfReplies	2.8342	0.594	4.773	**0.000**
averageOfRetweets	0.9554	0.135	7.060	**0.000**
averageOfPolarity	0.8967	0.238	3.762	**0.000**
averageOfTimeDifference	-0.1856	0.196	-0.947	0.344

The following table shows the results of 10-fold cross validation of the current experiment, where the accuracy reaches 0.71, percentage that is very close to the accuracy that achieves the proposed model, as it contains 4 out of 5 of the features that are selected in the final model. We can see that precision and recall reach almost 0.78.

Table 17 Results of 10-fold Cross Validation of 5.3.4

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.83	0.64	0.73	70
True (controversial)	0.73	0.88	0.80	76
Avg/total	0.78	0.77	0.76	146
Summary	**Accuracy of logistic regression classifier on test set**: 0.76 **Mean Absolute Error**: 0.238 **10-fold cross validation average accuracy**: 0.71			

5.3.5 Combination of Structural, Linguistic, Sentiment and Time Features

An experiment that combines structural, linguistic, sentiment and time features is presented in this section. As presented in the following table, the significant features are the average of verbs, the average of hashtags, the average of personal pronouns, the average of bad words and the average of polarity. Only 3 of these features are included in the list of features of the proposed model, which are average of personal pronouns, average of bad words and average of polarity.

Table 18 Logistic Regression Results 5.3.5

	Coeff.	**Std.err**	**z**	**P> |z|**
averageOfNouns	1.0313	0.752	1.372	0.170
averageOfVerbs	7.2120	2.374	3.037	**0.002**
averageOfEmphasisPunctuation	751036	44.890	1.673	0.094
averageOfHashtags	5.8930	1.869	3.153	**0.002**
averageOfPersonalPronouns	-9.0.35	1.205	-7.495	**0.000**
averageOfBadWords	-12.0964	3.665	-3.301	**0.001**
averageOfControversialWords	-0.6031	1.677	-0.360	0.719
averageOfPolarity	1.2487	0.225	5.540	**0.000**
averageOfTimeDifference	-0.2599	0.190	-1.369	0.171

The aforementioned combination reaches the percentage of 0.678 in 10-fold cross validation, as presented in the table below. The precision and recall reach the 0.70, performing poorer that then previous experiment, where precision and recall reached almost 78%.

Table 19 Results of 10-fold Cross Validation of 5.3.5

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.74	0.56	0.63	70
True (controversial)	0.67	0.82	0.73	76
Avg/total	0.70	0.69	0.69	146
Summary	**Accuracy of logistic regression classifier on test set**: 0.69 **Mean Absolute Error**: 0.309 **10-fold cross validation average accuracy**: 0.678			

5.3.6 Combination of Twitter based, Linguistic, Sentiment and Time Features

The final experiment combines Twitter based, linguistic, sentiment and time features. The results show that the significant features are the average of bad words, the average of replies, the average of retweets and the average of polarity, as presented in the table that follows. All these features are included in the list of the statistically significant features of the proposed model.

Table 20 Results of 10-fold Cross Validation of 5.3.6

	Coeff.	**Std.err**	**z**	**P> |z|**

averageOfReplies	2.7349	0.564	4.850	**0.000**
averageOfRetweets	0.8719	0.106	8.227	**0.000**
averageOfBadWords	-10.1594	9.787	-2.683	**0.007**
averageOfControversialWords	-1.0551	1.577	-0.669	0.503
averageOfPolarity	0.8867	0.235	3.779	**0.000**
averageOfTimeDifference	-0.2039	0.197	-1.035	0.301

The following table shows the results of 10-fold cross validation of the current experiment, where the accuracy reaches 0.71, percentage that is very close to the accuracy that achieves the proposed model. The precision and recall reached almost 78%, percentage which is the same with the results of experiment in Section 5.3.4

Table 21 Results of 10-fold cross validation of 5.3.5

	Precision	**Recall**	**F1-score**	**Support**
False (non-controversial)	0.84	0.66	0.74	70
True (controversial)	0.74	0.88	0.80	76
Avg/total	0.78	0.77	0.77	146
Summary	**Accuracy of logistic regression classifier on test set**: 0.76 **Mean Absolute Error**: 0.226 **10-fold cross validation average accuracy**: 0.71			

5.3.7 Summary

Concluding with the results of all the experiments performed in the current section, it worth mentioning that higher accuracy is reached when combining 4 out of the 5 categories of features that we propose (structural, linguistic, sentiment, twitter based and time), i.e. experiments of Sections 5.3.4 5.3.5 5.3.6 where 10-fold cross validation reaches accuracy with values higher of 69%. We get poorer results in the cases where 3 categories of features are combined as happens in the experiments of Sections 5.3.1 5.3.2 5.3.3 where accuracy of 10-fold cross validation is lower than 69% in all cases. The results of the aforementioned experiments verify that higher accuracy is achieved when combining features of different categories in the prediction model. The proposed model of this thesis achieves the highest accuracy of all the combinations presented with accuracy 77%, as it combines features from all categories that contain the most statistically significant features, thus verifying our hypothesis.

6 Conclusion and Further Research

In this thesis, aspects of the proposed method in predicting controversy in Twitter have been discussed. In the process of our research, a dataset was collected from Twitter, in order to be used in this research for training, testing and further data analysis. Using the aforementioned dataset, an approach that can predict the controversy in Twitter data was proposed. The performance of the proposed approach was evaluated with respect to its prediction accuracy. Furthermore, a variety of features describing the dataset was studied and our results report on how efficient these features are in predicting controversies. Features from different works have been examined and adapted to the current needs.

In this last chapter of this thesis, we shortly summarize our research and its main findings, and point out possible improvements and avenues for future research.

6.1 Conclusions and Results

The current work proposes a method for predicting controversy in Twitter using a query-based approach for data retrieval. As an initial step, a dataset was created by parsing tweets that contained specific keywords or hashtags referring to different topics. The dataset was pre-processed and then classified using the Naïve Bayes classifier to extract their polarity. An innovative definition of controversy is proposed introducing a factor δ, which is the threshold of difference between positive and negative tweets of a subset. A topic is characterized as controversial when the aforementioned difference is lower than the threshold. After classifying the topics, a list of features were extracted covering different aspects of the data, such as structural, linguistic, twitter-based, sentimental and temporal characteristics. We aggregated the features using set of tweets making our dataset richer semantically than examining each tweet separately. The final dataset was used to train a logistic regression model and extract the statistically significant features that are used in classification. The results showed that the most statistically significant features are the following: average of bad words, average of replies, average of retweets, average of polarity and average of personal pronouns. Moreover, we ended up that combining features from different categories provides better results and strengthens our proposal.

Concluding, the proposed methodology is an initial attempt in predicting controversy in Twitter using query-based data based on work done previously for

detecting controversy. It goes one step beyond the state-of-the art by proposing a list of features that are statistically significant for predicting controversy and examines which time window could be enough to predict controversy.

6.2 Future Research

A first approach for further research could be training the proposed model using richer datasets, to examine if our intuition of aggregating the features in group of tweets is verified in large datasets, too. Moreover, a lexicon based classifier with labeled tweets can be used for the classification of the collected tweets to enrich the sentiment analysis part of the proposed methodology. A set of features derived from external resources i.e. articles could be used to enhance the dataset similarly with Popescu et al.

Further research work could emphasize in predicting controversy using real-time streams of data, as the current work uses query-based data that were parsed by using specific keywords for trending topics. Moreover, prediction at this point could be combined with topic detection as in streaming data, the discussed entities should be first recognized. Moreover, a controversy level could be defined for different topics indicated by how popular the topic is and other semantic features. This controversy level could be taken into account in the prediction of the future. Another important aspect could be the duration of the controversy i.e. lasting or short-lived controversy in the future.

7 References

2018. https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json.

Adam Berger, Stephen Della Pietra, Vincent Della Pietra. "A maximum entropy approach to natural language processing." *Computational Linguistics*, 1996: 39-71.

Adam Bermingham, Alan F. Smeaton. "On Using Twitter to Monitor Political Sentiment and Predict Election Results." *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP)*, 2011.

AJ Morales, J Borondo, JC Losada,RM Benito. "Measuring political polarization: Twitter shows the two sides of Venezuela." 2015.

Alec Go, Richa Bhayani, Lei Huang. "Twitter Sentiment Classification using Distant Supervision." *Processing*, 2009: 1-6.

Alexander Pak, Patrick Paroubek. "Twitter as a Corpus of Sentiment Analysis and Opinion Mining." *European Language Resources Association*, 2010.

Ana-Maria Popescu, Marco Pennacchiotti. "Detecting controversial events from twitter." *CIKM '10 Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010: 1873-1876 .

Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, Ed H. Chi. "He Says, She Says: Conflict and Coordination in Wikipedia." *In Proc. CHI*, 2007: 453-462.

Aron, Cullota. "Towards detecting influenza epidemics by analyzing Twitter messages." *Proceedings on the first workshop on Social Media Analytics*, 2010: 115-122.

Baccianella Stefano, Esuli Andrea, Sebastiani Fabrizio. "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining." *LREC*, 2010: 2200-2204.

Bharat Naiknaware, Bindesh Kushwaha, Seema Kawathekar. "Social Media Sentiment Analysis using Machine Learning Classifiers." *International Journal of Computer Science and Mobile Computing*, 2017: 465-472.

Bo Pang, Lillian Lee, Shivakumar Vaithyanathan. "Thumbs up? Sentiment Classification using Machine Learning techniques." *Proceedings of the ACL-02 conference on Empirical methos in natural language processing-Volume 10, Association for Computational Linguistics*, 2002: 79-86.

Bollen Johan, Huina Mao, Xiao-Jun Zeng. "Twitter mood predicts the Stock Market." *Journal of Computational Science*, 2011.

C. Radhakrishna Rao, Helge Toutenburg. "The Linear Regression Model ." In *Linear Models: Least Squares and Alternatives, Second Edition*, by Helge Toutenburg C. Radhakrishna Rao, 23-33. Springer, 1999.

Cerini S., Compagnoni V., Demontis A., Formentelli M, Gandini G. "Micro-WNOp: A gold standard for the evaluation of automatically compiled lexican resources for opinion mining." 2007.

Conover Michael, Ratkiewicz, J., Francisco, M., Gonalves, B., Flammini, A.,Menczer, F. "Political polarization on Twitter." *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011: 89-96.

David W. Hosmer, Stanley Lemeshow. *Applied Logistic Regression.* John Wiley & Sons, Inc., 2005.

Ding X., Liu B., Yu P. "A holistic Lexicon-based approach to Opinion Mining." *WSDM 2008*, 2008.

Dori-Hacohen, Shiri. "Controversy Analysis and Detection." 2017.

Dr.Balasaravanan.K, Bharathi Bhaskaran R, Prabhakaran R,Saravanan S ,Vinoth M. "TWITTER SENTIMENT ANALYSIS." *International Journal of Pure and Applied Mathematics*, 2018.

Dunja, Mladenic. "Automatic word lemmatization." *Proceedings B of the 5th International Multi-Conference Information Society*, 2002: 153-159.

Funchun Peng, Nawwaz Ahmed, Xin Li, Yumao Lu. "Context sensitive stemming for web search." *Proceedings of the 30th annual international ACM SIGIR conference of Research and development in information retrieval*, 2007: 639-646.

Gloor, P. A., Krauss, J., Nann, S., Fischbach, K., & Schoder, D. "Web Science 2.0: Identifying Trends through Semantic Social Network Analysis." *In Computational Science and Engineering*, 2009: 215-222.

Haewoon Kwak, Changhyun Lee, Hosung Park, Sue Moon. "What is Twitter, a Social Network or News Media?" *In Proceedings of the 19th International World Wide Web Conference*, 2010.

Hall M., Frank E., Holmes G., Pfahringer B., Reautemann P, Witten I. "The WEKA data mining software: an update ." *SIGKDD Explor*, 2009: 10-18.

Hao Wang, Dogan Can, Abe Kazemzadeh. "A system for real time twitter sentiment analysis of US Presidential Election Cycle." *In Proceedings of ACL conference*, 2012.

Harald Schoen, Daniel Gayo-Avello, Panagiotis Takis Metaxas, Eni Mustafaraj, Markus Strohmaier, Peter Gloor. "The power of prediction with social media." *Internet Reasearch, Vol. 23, Issue 5*, 2013: 528-543.

Harman, Donna. "How effective is suffixing?" *Journal of the American Society of Information Science*, 1991: 7-15.

Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, Benyuan Liu. "Twitter improves seasonal influenza prediction." *Proceeding in Health Informatics (HEALTHINF)*, 2012: 61-70.

Hassan Saif, Yulan He, Harith Alani. "Alleviating Data Sparsity for Twitter Sentiment Analysis." *2nd Workshop on Making Sense of Microposts (#MSM 2012): Big Things Come in Small Packages: in Conjunction with WWW*, 2012.

Hoda Sperhri Rad, Denilson Barbosa. "Identifying controversial articles in Wikipedia: A comparative study." *In Proc. WikiSym*, 2012.

https://developer.twitter.com. 2018.

Hu Minqing, Liu Bing. "Mining and summarizing customer reviews." *KDD'04*, 2004.

Introduction to Time Series Analysis. https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm (accessed May 2018).

Investopedia. *Time Series.* https://www.investopedia.com/terms/t/timeseries.asp (accessed May 2018).

Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, Xiaotie Deng. "Exploiting topic based twitter sentiment for stock prediction." *Proceedings of ACL*, 2013: 23-29.

Jinlong Guo, Yujie Lu, Tatsunori Mori, Catherine Blake. "Expert-Guided Contrastive Opinion Summarization for Controversial Issues." *In Proceedings of the 24th International Conference on World Wide Web*, 2015: 1105-1110.

Jivani, Anjali. "A Comparative Study of Stemming Algorithms." *Int.J.Comp.Tech.Appl, Vol 2 (6)*, 2011: 1930-1938.

Johan Bollen, Huina Mao, Alberto Pepe. "Modeling Public Mood and Emotion:Twitter Sentiment and Socio-Economic Phenomena." *ICWSM11*, 2011.

Jusoh Shaidach, Hejab Al-Fawareh. "Natural language interface for online sales." *Proceedings of the International Conference of Intelligent and Advanced Sstem (ICIAS2007)*, 2007: 224-228.

Kannan S., Gurusamy Vairaprakash. "Preprocessing Techniques for Text Mining." 2014.

Kazimierz Zielinski, Radoslaw Nielek, Adam Wierzbicki, Adam Jatowt. "Computing controversy: Formal model and algorithms for detecting controversy on Wikipedia and in search queries." *Information Processing and Management*, 2017: 14-36.

Kim Soo-MIn, Hovy Eduard. "Determining the Sentiment of Opinions." *COLING'04*, 2004.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, Michael Mathioudaki. "Quantifying Controversy In Social Media." *WSDM*, 2016: 32-42.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, Michael Mathioudakis. "Exploring Controversy in twitter." *CSCW [demo]*, 2016: 33-36.

Krovetz, Robert. "Viewing morphology as an inference process." *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, 1993: 191-202.

Le T. Nguyen. Pang Wu, William Chan, Wei Peng. "Predicting Collective Sentiment Dynamics from Time-series Social Media." *Proceedings of 18th ACM SIGKDD*, 2012.

Lee, Hooyeon. "Using Twitter to Estimate and Predict the Trends and Opinions." *Social and Information Network Analysis*, 2011.

Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, Bing Liu. "Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis." *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)*, 2015.

M. Pennacchiotti, A. M. Popescu. "Detecting controversies in Twitter: A first study." *In Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, WSA '10*, 2010: 31-32.

Massimo Melucci, Nicola Orio. "A novel method for stemmer generation based on hidden Markov models." *Proceedings of the 12th international conference on Information and knowledge management*, 2003: 131-138.

Matthew, Gerber. "Predicting Crime Using Twitter and Kernel Density Estimation." *Decision Support Systems*, 2014: 115-125.

Mauro Coletto, Kiran Garimella, Aristides Gionis, and Claudio Lucchese. " A motif-based approach for identifying controversy." *In Proceedings of the 10th International on Conference on Web and Social Media*, 2017.

Mladenic Dunja, Plisson Joel, Lavrac Nada. "A rule based apprach to word lemmatization." *Proceedings C of the 7th International Mult-Conference Information Society*, 2004.

Moletni Luca, J. Ponce Deleon. "Forecasting with Twitter Data: An Application to USA TV Series Audience." *International Journal of Design & Nature and Ecodynamics*, 2016: 220-229.

Munmun De Cloudhury, Michael Gamon, Scott Counts, Eric Horvitz. "Predicting Depression via Social Media." *ICWSM 13*, 2013: 1-10.

Nasir Naveed, Jerome Kunegis, Thomas Gottron, Arifah Che Alhadi. "Bad News Travel Fast: A Content-based Analysis of Interestingness on Twitter." *Proc. Web Science Conf*, 2011.

Nuvo Oliveira, Paulo Cortez, Nelson Areal. "On the Predictabilty of Stock Market Behavior using StockTwits Sentiment and Posting Volume." *Progress in Artificial Intelligence*, 2013: 355-365.

O'Connor Brendan, Balasubramanyan Ramnath, Routledge Bryan R., Smith Noah A. "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series." *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.

Paice, Chris. "Another stemmer." *ACM SIGIR Forum, Volume 24*, 1990: 131-138.

PythonProgramming.net. https://pythonprogramming.net/ (accessed May 2018).

Scott, Czepiel. "Maximum Likelihood Estimation of Logistic Regression Models: Theory and Implementation." *Available at czep. net/stat/mlelr. pdf.*, 2002.

Sharma, Deepika. "Stemming algorithms: a comparative study and their analysis." *International Journal of Applied Information Systems*, 2012: 7-12.

Smith Laura, L. Zhu, K. Lerman, Z. Kozareva. "The role of social media in the discussion of controversial topics." *SocialCom/PASSAT Conference*, 2013.

Appendix A

In this section the bad words list and the controversial words list are attached in the following files: a) badWordsList.txt and b) controversialWordsList.txt, correspondingly.


badWordsList.txt


controversialWordsList.txt