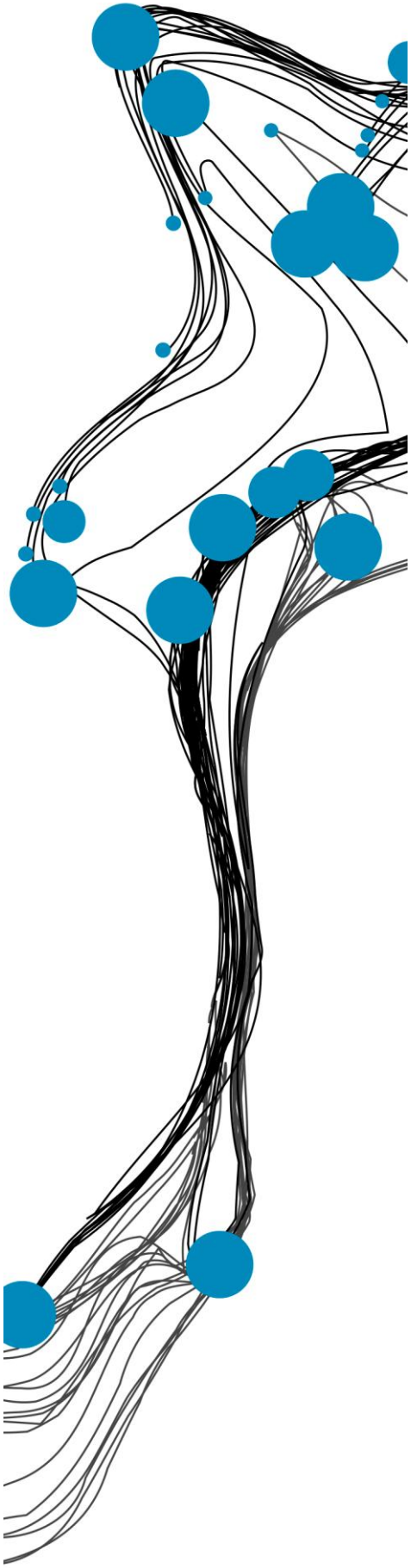

MASTER THESIS

Sentiment Analysis on Social Media using Machine Learning-Based Approach

Try Agustini
S1574728

Examination Committee
prof. dr. M. E. Iacob
dr. F.A. Bukhsh

June 2021

Faculty of Electrical Engineering, Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Abstract

In the modern society, social media has become more prevalent as its usage and popularity continues to rise within the advancement of connectivity and technology. Internet users across the world no longer use social media sites only to exchange personal information or to communicate with friends, colleagues, or relatives, but also to express thoughts and opinions on various topics. These topics are widely ranging from products, people, events, trends, and social issues. Opinions are useful for companies to find out what the customers think about their goods or services and their feedback is valuable to help decision making within the business, government, and other public institutions.

This research investigates the opinions and thoughts written by social media users about a particular product then determines the relationship between its underlying sentiments and a financial indicator of the said products. The two main topics selected as case study are Bitcoin crypto currency and a ridesharing application service named Uber, obtained from Twitter posts with time period of year 2019. The approach being applied is Machine learning-based sentiment analysis, in which a dataset of labeled texts will be used to train a classifier model and subsequently the resulting model is used to categorize the sentiments of the unlabeled texts. This way the trend of sentiment changes can be observed and this will help the writer to study the correlation between online sentiments and the stock price of Uber and Bitcoin respectively.

Essentially, the goal is to answer three main research questions. The first one is to identify which machine learning algorithm has the best performance in classification task of sentiments of Twitter data. The second is to find out to what extent the sentiments on social media can affect the financial performance of a product. Lastly we will observe how the outcomes of this correlation based on machine learning method compares to the lexicon-based approach. This study shows that boosting algorithm has performed the best among the other machine learning techniques. It is also evident that positive sentiments have more significant relationship on Bitcoin price, whereas negative sentiments have more influence on Uber price. Finally these findings are consistent with the experiment result of lexicon-based approach, which means the study proposed with machine learning approach is proven to be reliable.

Table of Contents

1 Introduction

The use of social media has become more prevalent in the recent years as its popularity continues to rise in the modern society. Internet users around the world no longer use social media sites only to exchange personal information or to communicate with friends, colleagues, or relatives, but also to express thoughts and opinions on various topics. These topics are widely ranging from products, people, events, trends, and social issues. Opinions are useful for companies to find out what the customers think about their goods or services and their feedback is valuable to help improving their business. In a similar manner, government, institutions and public figures would want to get insights and understand the impression of the general public towards them. From customers' perspective, opinions and sentiments about products are also beneficial as a reference in the decision making before purchasing those particular products.

There are numerous studies to analyze opinions and sentiment on social network platforms and they are conducted on different domains, finance is being one of them, specifically on share markets. The share markets are very volatile and a companies' share price is often very susceptible to rumors or news on social media. In the paper by Campbell & Hentschel (1992), some of the fluctuations in the share market are theorized. Essentially, they argue that the share markets are susceptible to certain market shocks. A market shock can be a news report by a popular news outlet or speculation, or changes in market regulations. Campbell & Hentschel explain these impacts by the so-called volatility feedback (1992). The volatility feedback shows that large negative share returns are more likely than large positive share returns. While the volatility feedback provides some explanation of the impact of market fluctuations, the crucial importance is how these news stories are perceived.

This study will aim to combine the natural language processing, machine learning and financial modeling to analyze the different impact of sentiments on social media towards a company's share prices. This can be very useful for businesses and can also shed light on how news or rumors circling on the internet might benefit or disadvantage the business on the short and long-run. It can also help understand what kind of factors and events that affect the trust and opinion of people towards products sold by a business.

1.1 Research Goals

The goal of this research is to answer two main research questions. The first one is to determine which one from the various machine learning classifiers that has the best performance in classification of sentiments of Twitter data. Once classification is carried out, the trend of sentiment changes can be observed. This led to the second research question, which is to find out the extent of which the sentiments on social media affect the financial performance of a product. Lastly, certain factors or events that influenced the share price changes of Bitcoin and Uber will be examined.

RQ1: Which machine learning technique has the best performance in sentiments classification of Twitter data?

RQ2: To what extent are the sentiments on social media can affect the financial performance of a product?

RQ3: Is there a difference between how the price of Bitcoin and Uber respectively are influenced by the sentiments on social media?

In accordance to these research questions, we assume the null hypothesis as follows

H_0: *None of the predictor variables will have a significant impact on the share price.*

Predictor variables are average positive sentiments and average negative sentiments daily for Bitcoin and Uber respectively. Whereas adjusted closing prices for Bitcoin and Uber serve as the outcome variables. In addition, based on the research questions that have been formulated, the following hypotheses will be assumed.

H1: Sentiments on social media will impact Bitcoin price.
At least one of the predictors will have an impact on Bitcoin price.

H2: Sentiments on social media will impact Uber share price.
At least one of the predictors will have an impact Uber share price.

H3: The result between machine learning-based sentiment analysis and lexicon-based analysis are consistent.

2 Literature Review

2.1 Systematic Literature Review

In this phase, the research method being used is SLR (systematic literature review) to find the most recent studies and experiments in this area.

2.1.1 Planning

Searching Process

The SLR process is done primarily by running a query and searching in scientific databases. The following databases were selected for the searching process:

- Google Scholar (https://scholar.google.com/)
- SCOPUS (http://www.scopus.com/)
- IEEE (http://ieeexplore.ieee.org/)
- ScienceDirect (http://www.sciencedirect.com/)

During the searching process, several search terms are defined with the following query: “Machine learning” AND “sentiment analysis” AND (“social media" OR “social network”)

2.1.2 Selection

Inclusion and exclusion criteria

In the searching process, not all articles and journals are taken into account. They are checked and then filtered based on the following criteria:

Inclusion criteria:

1. The study report is written in English
2. The study is published between the year 2000 until the present
3. It answered at least one of the research questions specified in this study
4. It is relevant to the search terms that have been defined

Exclusion criteria:

1. The study does not satisfy the inclusion criteria
2. The study is not related to any of the research questions
3. Duplicated studies
4. Studies that do not include the necessary details
5. Studies that do not seem reliable or give questionable results

In general, all studies that are not written in English are taken out. Several articles are also removed from candidate articles since they do not present important details such as the algorithms being used and the performance measures. Once the literature is collected, the writer goes through them quickly to decide whether they are relevant to the study. The steps followed are depicted in Figure 1.1.

The writer used several databases to search for the relevant works in the past. At the end of the process, a total of 33 articles are selected and included as the final reference for this study. The most contribution is from IEEE with a total of 595 articles and the second one is from Scopus with 577 articles. The complete list of the articles used is listed in the Appendix.

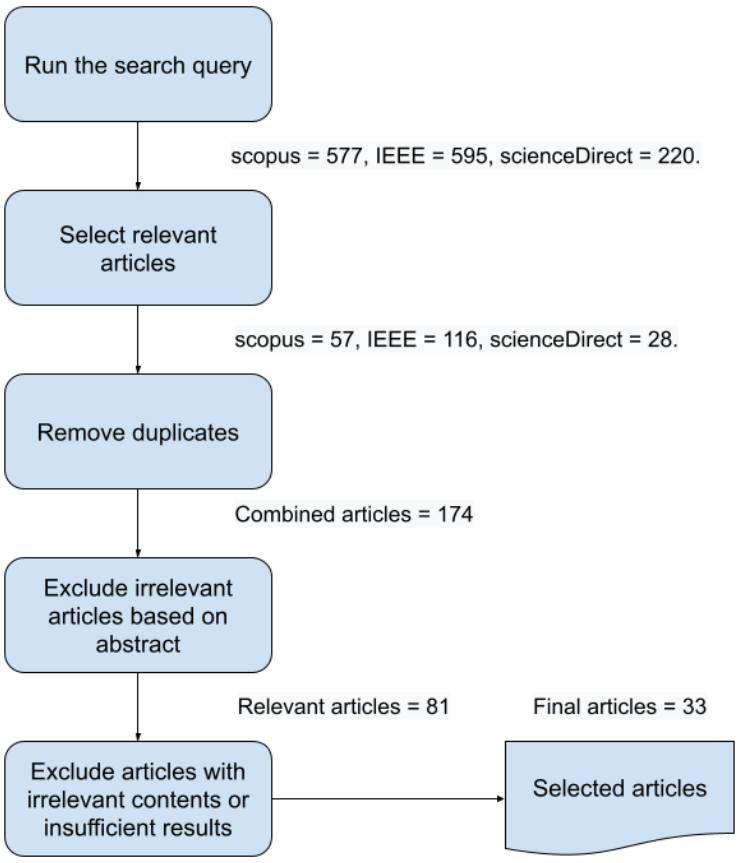

Figure 2.1 SLR selection process

2.2 Sentiment Analysis

The term “sentiment analysis” originated from as early as 2001 in a research attempting to discover and predict market sentiment based on an evaluative text (Das and Chen, 2001). By late 2003, more studies were published using this same phrase which contributed to its popularity. The term was used to describe a task of classifying reviews into certain sentiment polarity, which is either positive or negative. However it is often used interchangeably with “opinion mining” as nowadays the term may be used broadly, such as in the computational application of finding opinion and subjectivity in a text (Pang & Lee, 2008).

Opinion mining or sentiment analysis can be done in several ways in terms of granularity, namely document level, sentence level, and feature level. Document level seeks for the general opinion of the author of a text, for example a product summary or a movie review. The opinion could be negative or positive. Likewise, sentence level analysis returns the polarity or sentiment of one single sentence based on the words forming it. The problem with document and sentence level is that they do not address specifically the entity on which the author expresses his or her sentiment, while the author may discuss different entities and talk about different topics on each sentence. This is why a fine-grained analysis is much needed.

The increasing popularity of web 2.0 encourages the surge of user-generated content and this has opened various opportunities for practitioners, business, and academia to instigate new research methods to solve and answer different problems. Nonetheless this is still a challenging task because the varying forms of textual representations can be included in a social media post, such as slang words, punctuation, emoticons, and URLs. Another challenge is the existence of irony, sarcasm, or ambiguity in the way people express their thoughts and feelings. For example, an exclamation mark can either indicate that the writer is angry or excited, likewise, a crying emoticon may be interpreted as sadness or happiness.

There have been numerous studies done in the sentiment analysis area that are focusing on social media, due to the rapid growth of social networking websites in the recent years. Practitioners and researchers have made continuous effort to explore and analyze this massive data, taking advantage of the ever-growing user-generated content and conversations exchanged daily through these sites, to help improving business or to solve various real-world problems. Ducange and Fazzolari (2017) used reviews and ratings data from Amazon and TripAdvisor as well as the opinions posted on Facebook and Twitter, to train their classifiers to recognize three types of

sentiments, namely positive, negative, and neutral. The results show that the system is able to classify restaurant reviews from TripAdvisor with accuracy of 91.95%, and the online shop reviews from Amazon with 93.01% accuracy. They also build a traffic detection system on Italian road networks based on the tweets posted by Italian users that contain certain keywords, such as *queue*, *crash*, and *accident*. It was revealed that this system is able to monitor real-time traffic events even earlier than online news websites (Ducange & Fazzolari, 2017).

Sentiment analysis could also be used to find out citizens' opinions towards government policies, new bills or laws, political views, and national events. Fatyanosa and Bachtiar (2017) attempted to classify the polarity of Indonesian Twitter users regarding the election of Jakarta's governor. Such application of sentiment analysis has attracted high interest from people nowadays, although the opinion and attitude of social media users do not necessarily represent those of the whole nation. In one research a method to predict the outcome of the US presidential election in 2016 have been proposed. Even though the outcome has predicted that Hillary Clinton will win the voting, it turned out that Donald Trump won the election (Wicaksono, 2016). Obviously much work is still needed to improve the existing method but this shows the unprecedented possibilities that could be discovered beyond the big data available on the internet.

Companies would also be interested to qualitatively measure the key values essential for their business, such as brand awareness, reputation, and customer engagement. Saragih and Girsang (2017) investigated the opinion on online transport mobile applications by extracting users' comments on the respective apps from Facebook and Twitter. They categorized the sentiment of each comment using TF-IDF score and classified its context, such as the quality of service and feedback on the system. The study revealed that the comments posted by the users are mostly complaints, which is quite understandable as it is not common for customers to praise a product or service on social network sites intentionally, although they will likely give their fair and true judgments on review and rating websites. Further, according to Rathan et al. (2017) sentiment analysis methods primarily can be divided into three types:

- Lexicon-based approaches
- Machine learning-based approaches
- Hybrid approaches

Lexicon based sentiment analysis techniques typically make us of a bag of words, in which the words are mapped with its predefined sentiment value. To calculate a sentiment of a sentence or a

post, the sentiments from all of words are aggregated. On the other hand, machine learning approaches take advantage of a dataset of which the sentiments are already known and use it to determine the sentiments of new unlabeled dataset. Finally, hybrid methods employ the combination of these two approaches. The complete classification of textual sentiment analysis is presented in Figure 1.2 (Rathan et al., 2017).

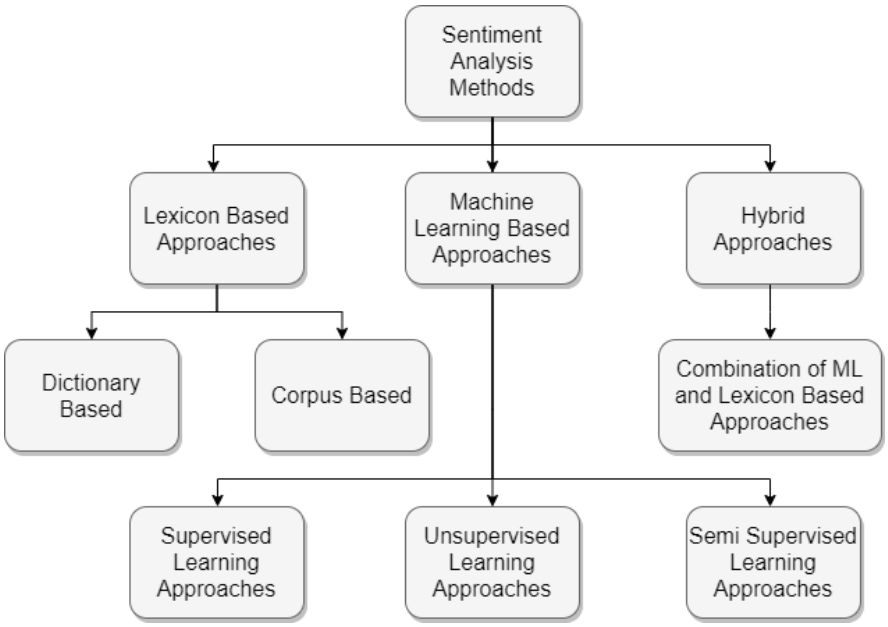

Figure 1.2 Classification of sentiment analysis methods

2.3 Machine Learning

Machine learning is a process to optimize the performance of a system by means of programming, using past data and experiences (Alpaydin, 2010). Machine learning is needed to perform data analysis in the absence of human expertise and it could be done in a short time over a large amount of data. For example, any person would normally be able to discern men's voices from those of women in one listen. However in case there are one hundred of voice recordings to recognize, it may take some time for a person to complete this task. By designing and running machine learning algorithms that are run on a computer, this can be done automatically and therefore is beneficial to save time and resources compared to doing manual work carried out by humans.

Nowadays the application of machine learning can be observed everywhere in various domains. Practitioners and the research community have been implementing this paradigm to help solving

problems in the real world, for instance in predicting customer's purchasing patterns in supermarket chains, credit application assessment, face and speech recognition, medical diagnosis, network optimization, and so on. Occasionally, the machine learning tasks make use of the availability of datasets that have been labeled. This is often called as a classification problem, where the designated algorithm needs to decide where the new data should be placed amongst the available classes or categories.

As an illustration, a set of pictures from a photo hosting service such as Flickr or Instagram would typically have been tagged manually by their users with multiple words that describe the objects captured in the images, e.g. "man", "car", "building" or "mountain". This labeled data is useful to train the machine learning algorithm for it to understand the features and characteristics of different photos corresponding with the associated tags. Such mechanism is called supervised learning where the data instances are labeled with the correct output. In contrast, in unsupervised learning the correct label is unknown (Kotsiantis et al., 2007). In this case, the aim of machine learning is to find structures or patterns in the data based on the similarities or difference between data points. An example of unsupervised learning is clustering, where the analysis tries to establish hidden groups, categories, or classes in the data.

Aside from supervised and unsupervised machine learning, there is the third type of machine learning called reinforcement learning. Unlike the other two types in which the learner should be able to decide the best action or output, reinforcement learning returns a sequence of outputs that will yield to the most optimum reward (Sutton, 1992). This is applicable mostly in gaming situations, where overall actions are more important than merely a single action. For example, when playing a game of chess, one certain move is not that important as it will not give immediate reward, rather the next sequence of moves matters more as it will determine the end result of the game. In this study the writer will narrow down the machine learning topics and focus only on a variety of supervised machine learning methods. The following subsections will discuss the several classifier algorithms that are commonly used in text classification.

2.3.1 Linear Classifier

Naive Bayes

Naive Bayes is one of the most popular classifiers due to its simplicity and reasonably well performance. In naive bayes learning, independence between attributes or features is assumed. This is not always true in most real-world problems, therefore in some cases it is not preferred despite the ability to still produce a good result with high reliability. Surprisingly, naive bayes classifier works with both the case where the features are completely independent and where the features are functionally dependent (Rish, 2001).

Bayes classifier can be denoted by the following function:

$$p(C = i|\, X = x) \; = \frac{p(X = x \mid C = i\,)\, p(C = i\,)}{p(X = x)}$$

With $p(C = i|\, X = x)$ represents the probability of instance x belongs to class i, $p(X = x \mid C = i\,)$ represents the probability of generating instance x given the class is i, $p(C = i\,)$ is the probability of class i, and $p(X = x)$ is the probability of instance x to occur.

In text classification there are two different approaches that use Naive-bayes assumption. One which describes a document in a vector binary attribute that indicates whether a word occurs or not in the corresponding document is called multivariate Bernoulli model (McCallum & Nigam, 1998). In the second model, a document is represented by a set of words that appear without keeping the order in which they occurred. This second approach is referred to as multinomial Naive-Bayes. In several studies multinomial Naive-Bayes has been tested and proven to outperform other classifiers in some cases (Mahalaksmi & Sivasankar, 2015; Saad, 2014).

Logistic Regression

Logistic regression is a statistical model that aims to calculate the probability of a parameter having a certain value. The parameter can have two possible values (binary) or more than two (categorical). The model itself is inherently not a classifier, but it can be used as a classification method by choosing threshold values that will determine whether that parameter belongs to one class or another.

Samal et al. (2017) analyzed a dataset of movie reviews and categorized them as positive and negative using several classifiers. Logistic Regression came out as the second highest in performance with accuracy of 99.46% for 85,600 feedback collected from users. In another study, Zhang et al. (2016) predicted the orientation of consumer opinions on Chinese social media called Sina Weibo. They compared two collective classification methods that are logistic regression and naive bayes. The experiment result shows that logistic regression has better accuracy than naive bayes algorithm.

2.3.2 Support Vector Machine

Support Vector Machine (SVM) is one of the latest supervised machine learning techniques (Kotsiantis et al., 2007). The model learns from a set of data samples that has been labeled as one of two categories, then based on the learning algorithm, attempts to assign new data into the right category. Therefore the model is called a binary linear classifier. Also referred to as *kernel machine*, it is a maximum margin method that can also be represented by a sum of the influence of a subset of the training instances (Alpaydin, 2010). Suppose some data points that are defined in n-dimensional vector space, the algorithm tries to establish an optimal-separating hyperplane that divides the data instances into two sides with the maximum margin. Consequently, this hyperplane will be defined in (n-1) dimensional vector space.

Nair et al. (2015) conducted an experiment of Sentiment analysis on Malayalam film reviews using hybrid approach comprising machine learning and rule-based approach. In this study they used a support vector machine and Conditional Random Field (CRF) method that each is combined with a rule-based approach, and found out that SVM outperformed CRF with accuracy of 91% at the highest. Many other past researches have shown that SVM is a great solution for social media sentiment analysis on social media, especially for Twitter (Pang et al., 2002; Salvetti et al., 2006).

2.3.3 Decision Trees

Decision tree is a hierarchical decomposition of data space. It is one of supervised learning models, in which the local region is identified by splitting recursively in sequence in a smaller number of steps (Alpaydin, 2010). Using this method, the data is hierarchically

divided by certain conditions on the features or attributes. Given a document feature vector space, it will be iteratively partitioned to form the decision tree. At each node, some conditions are applied to decide which branch is taken. This process is done repeatedly starting from the root until the leaf node is reached. The problem with conventional trees is the partitioning only done to the coordinate axes. With the growth of a tree, any subtle patterns that can be recognized from the input space may keep being partitioned into tiny segments, hence this could lead to overfitting.

Jain and Dandannavar (2016) presented a step-by-step process to conduct sentiment analysis on Twitter using machine learning. They applied the proposed framework on twitter dataset related to 3 different domains; IT Industry (Apple), Bank (ICICI), and Telecom (BSNL). Apache Spark is used as the tool, meanwhile Naive Bayes and Decision trees are used as the algorithms for the proposed framework. Interestingly, the result shows that DT performs extremely well with Accuracy, precision, recall and F1-Score up to 100%.

2.3.4 Maximum Entropy

Maximum entropy (MaxEnt) is another classifier to estimate general probability distributions from data. The main principle is that when nothing is known, the distribution should be as uniform as possible, or in other words it has maximal entropy (Kotsiantis et al., 2007). The conditional distribution is defined as follows:

$$P_\lambda(y|X) = \frac{1}{Z}(X)exp\left\{\sum_i \lambda_i\, f_i\,(X,y)\right\}$$

Where 'X' denotes the feature vector and 'y' is the class label. λ_i represents the weight coefficient and Z(X) is the normalization factor. Meanwhile $f_i\,(X,y)$ is the feature function which can be defined as:

1, if $X = x_i$ and $y\ = y_i$

or 0, otherwise

Unlike Naive Bayes, it doesn't assume independence between features and there is no assumption made regarding the relationship between them, so it can perform better in standard text classification tasks. In natural language processing, the features are seldom

independent hence each feature in Maximum Entropy classifier is an indicator function of some document properties.

Ashok et al. (2016) proposed a framework for a recommender system that is using machine learning-based sentiment analysis on social media posts. They used mainly 3 approaches; rule-based sentiment analysis, sentiment analysis using machine learning techniques which includes Naive Bayes, SVM, Random Forest, and Maximum Entropy, and lastly aspect based sentiment analysis (ABSA). In their study Maximum Entropy has the best accuracy value amongst other techniques.

2.3.5 Artificial Neural Networks

Neural network classifier or often referred to as Artificial Neural Networks (ANN) is a statistical model that is based on biological neural systems like human brains. An ANN consists of a set of nodes or neurons that connect to each other and hold certain values in its edges. The simplest form of neural network that uses a single layer is called *perceptron* and it works well for linear problems, such as text classification (Liu & Zhang, 2012). The more complicated form of ANN may use multiple layers of neurons and this can be generalized to solve non-linear separation.

In Natural Language Processing, neural networks have been used in a wide variety of tasks. Collobert (2011) used a convolutional neural network for syntactic parsing and also for semantic role labeling to avoid excessive task-specific feature engineering. In his works he applied convolutional layers to extract sentence-level features. Similar to this study, Dos Santos and Gatti (2014) added another layer to propose a new deep convolutional neural network to perform sentiment analysis of short texts that utilize information from character to sentence-level. Arulmurugan et al. (2019) attempted to integrate artificial neural networks with naive bayes and support vector machines to build a cloud machine learning system. The proposed method proved to have a better performance than the existing tools.

2.3.6 Random Forest

Random forest classifier is defined as a function that builds multiple decision trees and yields a class which is basically the mode of all classes. Decision trees that grow deep often

try to learn irregular patterns, in other words, over-fit the data. Random forests is a way to average multiple deep decision trees, which is done by training different subsets of the training set at a time, with the aim of increasing the variance.

In a study of examining the impacts of reviews towards product sales, Ghose and Ipeirotis (2010) conducted prior trials with SVM and Random Forest to decide which machine learning algorithm they will use in the final experiments. It was evident that SVM consistently performed worse than Random Forest, besides its running time that is significantly higher. Therefore they have selected Random Forest as it is more robust for their tasks.

2.3.7 Nearest Neighbor

The k-nearest neighbor (KNN) is a supervised machine learning classifier that relies on the class labels from the training documents to determine which class is the most likely the test document belongs to. It takes the labels from an existing training set as input and the output will be class membership. This is based on the principle that the instances within the dataset will generally exist in close proximity to the other instances with similar properties (Cover & Hart, 1967). Therefore it does not build an explicit declarative model for a class. Given a test document d, KNN finds the k-nearest neighbors among training documents. The similarity score of each nearest neighbor document to the test document is used as the weight of the classes of the neighbor document. An instance is then classified by a similarity-based vote of its neighbors to a class c that is the most common amongst its k nearest neighbors. K is a positive number and an odd number is typically chosen to avoid tied score between classes. If k = 1, then the instance is simply assigned to the class of its nearest neighbor.

Unlike the majority of machine learning algorithms, KNN is based on instances as opposed to data distributions; hence it is regarded as non-parametric and lazy learning algorithm. Baydogan and Alatas (2018) have conducted a sentiment analysis study on a dataset of 10,000 English tweets using KNN and decision tree algorithms. The experiment shows that the best sentiment classification result is obtained using k = 3 and k = 5 values. One of the tests generated an accuracy of 0.752, precision 0.758, and a solid recall value of 0.987. In respect to recall statistics, KNN algorithm steadily outperformed those of the decision trees.

2.3.8 Ensemble Classifiers

Ensemble method allows multiple models to be trained using the same learning algorithm. This method tries to produce the improved reliability and stability by combining multiple classifiers rather than a single one. In any machine learning technique, the differences between predicted and actual values may be caused by bias, variance, and noise. With ensemble method bias and variance can be reduced effectively and it also helps to increase the robustness of the model. Examples of ensembles classifiers are bagging, boosting, and bootstrapping.

Perikos and Hatzilygeroudis (2017) analyzed the user reviews for hotels and rooms by collecting a dataset from booking.com using a combination of several classifiers. It was found that ensemble classifiers work better than individual-based classifiers. Therefore it is recommended to employ the ensemble learning approach for sentiment classification to boost the result. The majority voting has also been observed in the same study and it was revealed to outperform best individual classifiers.

2.4 Sentiment Analysis on Social Media and Stock Value

In the recent rise popularity of social media, large volumes of opinions are expressed by users online which are freely accessible but unstructured in nature. Many academic researches have been done to predict the stocks movement by means of correlating the public sentiment to the behavior of stock prices, considering all variables that might affect the stock performance and values. The advanced technology of natural language processing which provides helpful feature extraction techniques in combination with machine learning approaches offer powerful classification algorithms with more accurate results.

Deepa et al. (2020) in his study has performed an analysis on various feature extraction techniques to detect the polarity of words from the stocktwits then classify the opinion using Logistic Regression classifier. Evaluation is done with feature engineering techniques like CountVectorizer, TF-IDF, Word2Vec and Glove by using machine learning. The proposed system is combining data preprocessing, feature extraction: TF-IDF, and machine learning classifier. They used StockTwits website to gather a dataset of tweets from investors,

traders, analysts, and others about the stock market. The authors attempted to find out if there is any correlation between the posters' feelings and the future stock movement. In short, forecasting. The result shows that feature representation Word2Vec resulted in the best performance. The experiment also shows that the model can correctly classify the polarity according to human psychology.

Both social media data and stock market information may also be used to predict future sales. Pai et al. (2018) in his experiments presented a framework that combine sentiment scores with the current stock market values to forecast monthly total vehicle sales in USA. The results indicated that forecasting vehicle sales by hybrid multivariate regression data with de-seasonalizing procedures obtained more accurate results compared to other forecasting models. The use of hybrid data containing sentiment analysis of social media and stock market values can also improve the forecasting accuracy.

Nivetha and Chaya (2017) performed various prediction algorithms in their research to build a stable prediction model. The prediction model is based on monthly prediction and daily prediction to forecast the next day market price. The model estimates the open value of the next day in the market. Sentiment Analysis is required to identify and extract sentiments from each individual in the social media. The correlation between the sentiments and the stock value is to be determined. A comparative study is conducted using of three algorithms, namely Multiple Linear Regression (MLR), Support Vector Machine (SVM) and Artificial Neural Network (ANN) are done. The stock price is then predicted by sentiment analysis with the best forecasting algorithm. The result shows that the deep learning algorithm performs better than the MLR and SVM. In deep learning algorithm the hidden layer neuron learns in every prediction. Hence the output layer neuron produces the best outcome. Artificial Neural Network is the best predicting algorithm.

Apart from financial indicators, there are possibility to add more attributes and supporting variables to improve the prediction accuracy. Bujari et al. (2017) considered economic and psychological factors as well in their study. They do not focus on a generic stock market index or on the sole sentiment analysis. They investigated whether tweet messages can be used to predict the future trend, e.g., positive, negative or neutral, of the stocks of specific companies listed in the Dow Jones stock market. In particular, they focus on companies belonging to three different economic sectors; namely technology, service and health-care.

They also included the trend of 5 different metrics for each stock (e.g., highest, lowest, opening price, etc.) and the trend of 13 different variables of the tweets (e.g., volume, sentiment, tweets with links, etc.). Through an evaluation that employed more than 800,000 tweets, the experiment showed that some of the proposed ad-hoc prediction methods can well predict the next day trend of the stock values of specific companies with up to 82% of success.

In another research, Attigeri et al. (2015) explored two types of analysis possible for prediction, technical and fundamental. In their study both technical and fundamental analysis are evaluated. Technical analysis is done using historical data of stock prices by applying machine learning and fundamental analysis is done using social media data by applying sentiment analysis. Social media data has high impact today than ever, it can aide in predicting the trend of the stock market. The method involves collecting news and social media data and extracting sentiments expressed by individual. Then the correlation between the sentiments and the stock values is analyzed. The learned model can then be used to make future predictions about stock values. It can be shown that this method is able to predict the sentiment and the stock performance and its recent news and social data are also closely correlated.

Stock prices may also be driven by a different other factors, such as industry performance, company news and performance, investor confidence, micro and macro economic factors like employment rates and wage rates. Stock pricing trends can be gauged from the factors that drive it as well as from the stock's historical performance. As fluctuations in stock prices become more volatile and unpredictable, forecasting models help reduce some of the randomness involved in investing and financial decision making. Users on social media platforms like Twitter, StockTwits, and eToro discuss issues related to the stock market. Coelho et al. (2019) investigated whether the analysis of posts on StockTwits add value to the existing features of stock price predicting models. Initial results indicate that the addition of sentiment analysis of the text referenced by the URL does not improve the performance of the model when all twits for a given day are taken into account since the model only identifies the direction of change and not the degree of change. The stock prediction model achieves 65% accuracy compared to the base case accuracy of 44% and augmenting the model with sentiment analysis did not change the accuracy.

3 Methodology

As mentioned in the research goals section, the following research questions will be answered.

> RQ1: Which machine learning technique has the best performance in sentiments classification of Twitter data?
>
> RQ2: To what extent are the sentiments on social media can affect the financial performance of a product?
>
> RQ3: Is there a difference between how the price of Bitcoin and Uber respectively are influenced by the sentiments on social media?

To answer the first research question, machine learning experiments will be performed with R programming tools. This activity will be divided into training and classification step. The result will show which algorithm has the best performance for classification of sentiment task. Afterwards, we can answer the second research question by carrying out the regression analysis, to see to what extent any of the predictors may affect Bitcoin and Uber share price. As such the hypotheses H1 and H2 can be tested.

H1: Sentiments on social media will impact Bitcoin price.
At least one of the predictors will have an impact on Bitcoin price.

H2: Sentiments on social media will impact Uber share price.
At least one of the predictors will have an impact Uber share price.

H3: The result between machine learning-based sentiment analysis and lexicon-based analysis are consistent.

To test hypothesis H3, the writer will also conduct sentiment analysis with a different approach than machine learning, namely lexicon-based sentiment analysis. The analysis is utilizing the “bag of words” with predetermined value of sentiments score. The tool being used for this method is Vader (Valence Aware Dictionary for sEntiment Reasoning). In Vader, words are associated with valence score that indicates the intensity of positive, neutral, and negative sentiments (Hutto & Gilbert, 2014).

In general, the methodology can be divided into three main parts. The first is the preparation of dataset, including collection and preprocessing. The second is the automatic sentiment analysis

classification using machine learning algorithms implemented with R programming language, and lastly the regression analysis on SPSS. The workflow and processes followed to conduct this study is depicted in Figure 3.1.

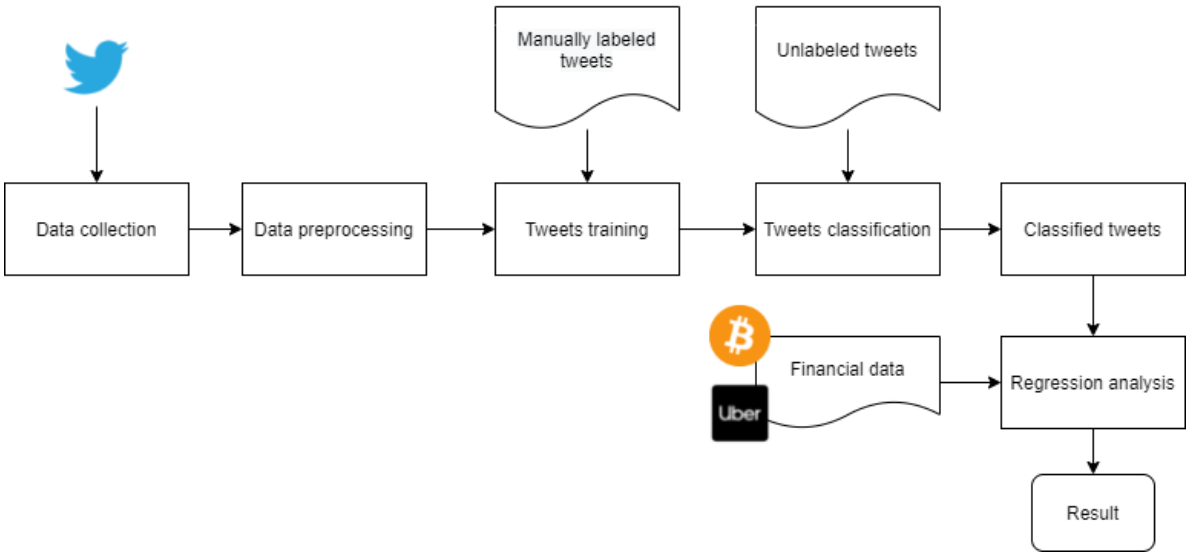

Figure 3.1 Workflow of sentiments classification on Twitter data

The first step is to collect data from Twitter using a scraper, followed by the cleaning of the data to remove unwanted and unnecessary text that can infer with the analysis. Once data is ready the machine learning is carried out using R programming tools. This step consists of two parts, namely training and classification. The classified tweets then can be utilized for the regression analysis to observe whether there is a relationship between sentiments financial indicators. These steps individually will be explained further in the following subsections.

3.1 Data Collection and Data Pre-Processing

Twitter data set is gathered using a Chrome browser add-on called Web Scraper. Using this tool the writer is able to collect tweets data containing the keyword “Bitcoin” from 1st of January 2019 until 31st of December 2019. For tweets containing “Uber” keyword, the collected dataset starts from 1st of June 2019 until 31st December 2019. This is because the initial public offering (IPO) of Uber was on May 9th 2019.

The following attributes are gathered for each tweet.

1. Twitter handle or username
2. Display name
3. Content
4. Number of replies, retweets, and favorites
5. Unix timestamp
6. Published date
7. URL to individual tweet

Once the dataset are sorted and merged for each month, a number of tweets are selected randomly and labeled with the corresponding sentiments:

- 0 represents negative sentiment
- 1 represents neutral sentiment
- 2 presents positive sentiment

Month	Number of tweets	
	Uber	BTC
January	-	9,610
February	-	8,637
March	-	8,261
April	-	9,287
May	-	10,826
June	6,476	9,623
July	6,018	9,833
August	6,051	9,064
September	5,828	7,784
October	6,119	8,646
November	7,016	7,938
December	6,837	7,579
Total	44,345	107,088

In total, the number of tweets collected for Bitcoin is 107,088, whereas the total number of tweets collected for Uber is 44.345. The breakdown of monthly tweets gathered is depicted in Table 3.1. The number of tweets labeled for Bitcoin and Uber are 1153 and respectively 1061. To collect the necessary daily price for Bitcoin and Uber, the financial data is downloaded from Yahoo finance website.

Dataset preprocessing is carried out to omit several unnecessary tokens that are considered noise and not useful to the classifications. The following steps are applied:

- All punctuations are removed to reduce each comment to purely words.
- All numbers are removed since it is irrelevant with the sentiments
- Stop words e.g. “and", “but”, “the” are removed this helps the algorithm because these are considered as noise words.
- Whitespace is removed to ensure the data is cleaner
- All words were converted to lowercase so avoid the classifier having duplicated words and treating them as different terms, for example “Service” and “service”

3.2 Sentiments Data Training and Classifications

The tweets datasets that has been cleaned and preprocessed are converted into a .csv file which will be the input for the training and test step. The following algorithms are tested to find out which one has the best performance:

1. Naïve Bayes
2. Support vector machine (SVM)
3. Maximum entropy
4. Decision tree (CART)
5. Random forest
6. Bagging
7. Boosting

Class	Tweet
1	Uber ridesharing proposal rejected by Wellington council
0	Service sucks, I had booked a cab just few mins ago and though driver cancelled the trip I am charged the cancellation amount, when I am trying to get help for that I am not able to connect agent & app is of no help Unprofessional
1	Do I ask my Uber driver to play the new Katy Perry song right now?
1	Free cabs in Los Angeles? The 'CLARIBELT1' promo code on Uber beats any Lyft code for free credit. Good deal.
2	Bitcoin is the blockchain King. Data mining is the new gold rush.
0	Maybe because of that? bitcoin is horrible for payment system that is why they are building a technology in top of it to make it feasible
0	Uber needs to sort this problem out. Whatever happens currently is seriously

	failing drivers who still don't know the rules around assistance dogs & passengers. Must do induction/training of their drivers (assess effectiveness) & be innovative. Be strategic.
1	Can you schedule an order like a ride? Can I get breakfast to my doorstep by the time I have to leave for work? @UberEats @Uber
2	Everything is good for Bitcoin. But since MMT accelerates the demise of the fiat regime, it is especially good for Bitcoin.

Table 3.2 Sample of classification output of the sentiment analysis

Once the algorithm has been determined than it will be used to train and classify the final dataset that are not labeled yet. All of the programs are written in R programming language. Example output of the program is pictured in Table 3.2.

3.2.1 Machine Learning Experiment

To teach the classifier models, a manually labeled dataset of 1153 tweets for Bitcoin and 1061 tweets for Twitter were prepared in advance. First, the dataset was read into R as a csv file. The order of the data was randomized to ensure that the training and test sets had a random selection of negative, neutral, and positive and comments. This dataset was saved and used in all the algorithms onwards. A corpus of words was then created using the Corpus function from the tm package. After creating the corpus, the data was preprocessed and the following pre-processing was applied, as it was described in the section 3.1.

Once the corpus was created, it was converted into a Document Term Matrix. This is a matrix which counts the frequency of each word in a comment. In addition, the data was split into a training set and a test set using k-folds cross validation. This was done to have a better estimation of how each classifier model is performing. The result is shown in Table 4.1.

In order to improve the classification, there are multiple methods incorporated into the program. One of these methods is to remove words from the corpus that do not appear in at least X documents. This can be done by using the findFreqTerms function from the tm package. To get a range of results the writer experimented with X ranging from 5 to 10. The second method is the use of N-Grams. N-Grams are defined as a set of N words within a document, when computing N-Grams usually the list of N-Grams start one word after each

other. A unigram is each word in a sentence. Bi-grams are when N = 2, for example the sentence “They live in San Francisco” would be split into “They live", “live in", “San Francisco", and so on. Generally, it is know that bi-grams are the more informative N-Gram combination (Analytics Vidya Contents Team, 2016). In this study, it is found that trigrams yield to the best accuracy and it has improved the result is relatively better than not using an N-grams at all. It should also be noted that the Naiıve Bayes algorithm is generated in a different package to the other algorithms, and applying N-Grams was a lot more complicated than for the others. Because of this reason and the fact that Naïve Bayes was performing poorly compared to the other algorithms, Naïve Bayes was not included in the experiments for N-Grams.

3.3 Regression Analysis

Since the dependent and independent variables are continuous variables, a multiple regression analysis was conducted. In the regression we assume the null hypothesis as: none of the predictor variables will have a significant impact on the Bitcoin price or Uber share price. For this research, the writer will conduct a simple regression analysis. It is a statistical tool that is used in the quantification of the relationship between a single independent variable and a single dependent variable based on observations that have been carried out in the past. This means a simple linear regression analysis can be utilized in the demonstration of how a change in the product's or company's sentiment analysis will consequently result in a change of the company's financial indicators.

The tools being used for the analysis is IBM SPSS Statistics. The function used is the standard linear regression and ‘Enter’ is chosen as variable selection method. For each of predictor variables, namely the average negative sentiments and average positive sentiments, the regression will be carried out against the dependent or outcome variables, namely adjusted closing price for Bitcoin and Uber. For each run of linear regression, SPSS will show the result of the following values; coefficients of standard error, T-value, and statistical significance or p-value, and Standardized coefficient ß.

The author will specifically focus on coefficient ß and the statistical significance of the experiment result indicated by the p-value. The threshold being used is $\alpha = 0.05$, therefore hypothesis will be accepted if p-value is below 0.05. Similarly, the same experiments will be conducted with sentiment analysis data produced by the lexicon-based approach. This way author may compare the result between the two methods and deduce on the reliability of machine learning approach.

4 Results

4.1 Sentiment Analysis Results

Firstly, based on the machine learning experiments that is done in the previous step to compare 7 different techniques, only one technique is chosen based on the performance result. Table 4.1 reveals the comparison of accuracy between each algorithms that were applied on Uber and Bitcoin dataset respectively.

Algorithm	Accuracy
Naïve Bayes	0.6579
SVM	0.7271
MaxEnt	0.5363
Decision Tree	0.7623
Random Forest	0.7713
Bagging	0.7771
Boosting	0.8102

Table 4.1 The classification accuracy of each algorithm on Bitcoin dataset

Algorithm	Accuracy
Naïve Bayes	0.4559
SVM	0.6473
MaxEnt	0.3971
Decision Tree	0.6276
Random Forest	0.6652
Bagging	0.6424
Boosting	0.8268

Table 4.2 The classification accuracy of each algorithm on Uber dataset

After experimenting with various methods of improving the accuracy of the classification algorithms, it can be concluded that Boosting algorithm has produced the best result. When running the boosting algorithm there is variations in the accuracy. To ensure that the 0.813 was reached, a run until function was used which meant the boosting algorithm would run over and over until the accuracy >= 0.80. Once the best accuracy was achieved, the final dataset for each month was read into R and the boosting algorithm classified the unlabeled tweets. This was done for each month in the entire year.

Date	Tweets	Negative	Neutral	Positive	Neg %	Neut %	Pos %
01/12/2019	203	15	180	8	7,39	88,67	3,94
02/12/2019	211	19	181	11	9,00	85,78	5,21
03/12/2019	236	25	193	18	10,59	81,78	7,63
04/12/2019	261	21	216	24	8,05	82,76	9,20
05/12/2019	217	19	182	16	8,76	83,87	7,37
06/12/2019	250	28	211	11	11,20	84,40	4,40
07/12/2019	213	23	178	12	10,80	83,57	5,63
08/12/2019	191	17	169	5	8,90	88,48	2,62
09/12/2019	264	21	223	20	7,95	84,47	7,58
10/12/2019	260	23	222	15	8,85	85,38	5,77
11/12/2019	210	18	180	12	8,57	85,71	5,71
12/12/2019	302	45	234	23	14,90	77,48	7,62
13/12/2019	237	29	198	10	12,24	83,54	4,22
14/12/2019	214	20	181	13	9,35	84,58	6,07
15/12/2019	277	39	224	14	14,08	80,87	5,05
16/12/2019	282	25	238	19	8,87	84,40	6,74
17/12/2019	267	23	232	12	8,61	86,89	4,49
18/12/2019	297	42	236	19	14,14	79,46	6,40
19/12/2019	276	26	239	11	9,42	86,59	3,99
20/12/2019	226	20	191	15	8,85	84,51	6,64
21/12/2019	264	36	209	19	13,64	79,17	7,20
22/12/2019	210	20	168	22	9,52	80,00	10,48
23/12/2019	256	28	206	22	10,94	80,47	8,59
24/12/2019	266	25	223	18	9,40	83,83	6,77
25/12/2019	209	15	186	8	7,18	89,00	3,83
26/12/2019	184	22	146	16	11,96	79,35	8,70
27/12/2019	284	39	229	16	13,73	80,63	5,63
28/12/2019	226	24	188	14	10,62	83,19	6,19
29/12/2019	280	36	224	20	12,86	80,00	7,14
30/12/2019	236	14	203	19	5,93	86,02	8,05
31/12/2019	270	18	213	39	6,67	78,89	14,44

Table 4.3 Example result of sentiment classifications

Table 4.3 summarizes the result of sentiment classification for the month of December. For example, on the 3rd of December there is a total number of 236 tweets talking about Bitcoin. Among them 25 tweets are classified as negative, 193 as neutral, and 18 as positive tweets. Then the percentage or ratio of each sentiment compared to the total number of tweets each day. This will be used as the strength measure and is used in the regression analysis.

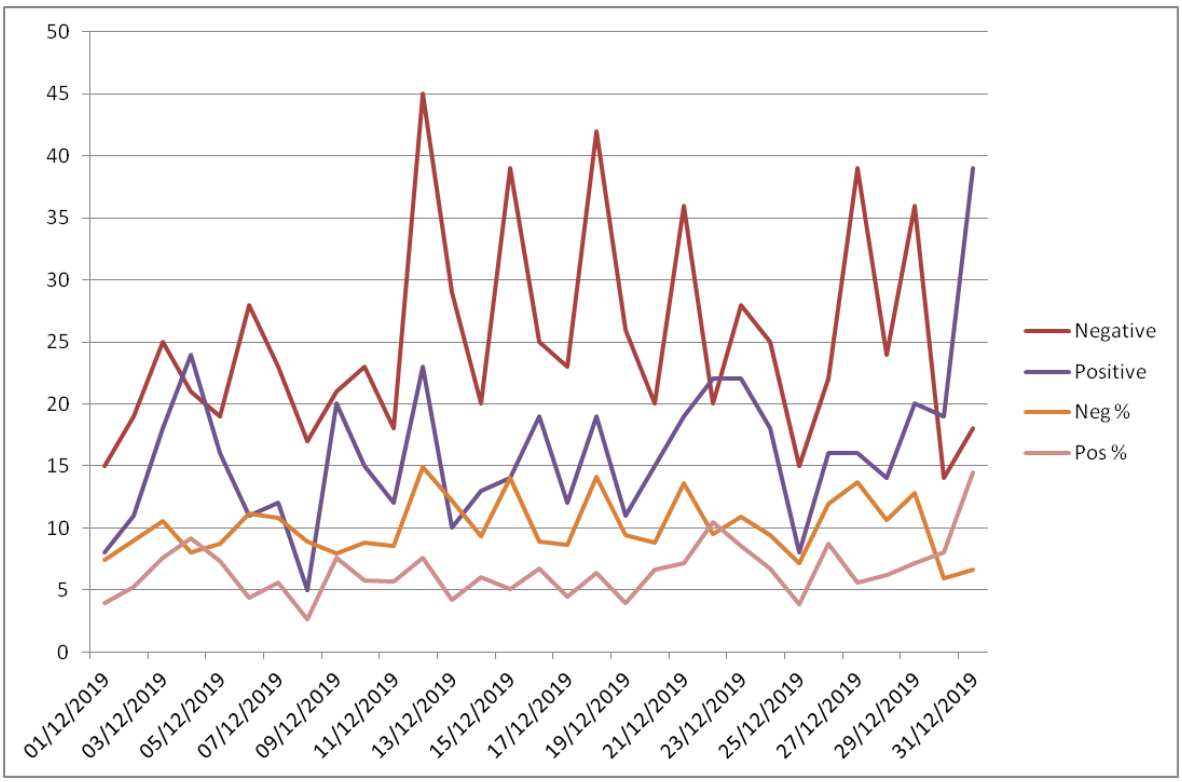

Figure 4.1 The sentiments fluctuations of Bitcoin in the month of December

4.2 Regression Analysis

In the regression we assume the null hypothesis as follows

H_0: *None of the predictor variables will have a significant impact on the share price.*

As mentioned previously, independent or predictor variables are average positive sentiments and average negative sentiments daily for Bitcoin and Uber respectively. Whereas adjusted closing prices for Bitcoin and Uber serve as the outcome or dependent variables.

In addition, based on the research questions that have been formulated, the following hypotheses will be assumed.

H1: Sentiments on social media will impact Bitcoin price.
At least one of the predictors will have an impact on Bitcoin price.

H2: Sentiments on social media will impact Uber share price.
At least one of the predictors will have an impact Uber share price.

H3: The result between machine learning-based sentiment analysis and lexicon-based analysis are consistent

4.2.1 BITCOIN – Machine learning-based sentiment results

Dependent variable: Adjusted Closing Price

Independent variable: Average negative sentiments (-)

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	7695.062	709.867		10.840	.000
	NegAvg	-2796.337	6494.032	-.023	-.431	.667

a. Dependent Variable: Adj Close

Negative sentiments has standardized coefficient ß = -0.023 and p-value of 0.667 > 0.05 therefore the result is not statistically significant.

This implies that there is not enough statistical evidence to conclude that the predictor variable has any impact to the outcome variable. We cannot reject the null hypothesis H_0 and evidently there is no significant relationship between negative sentiments of Bitcoin on twitter and Bitcoin price.

Dependent variable: Adjusted Closing Price

Independent variable: Average positive sentiments (+)

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	5829.095	656.586		8.878	.000
	PosAvg	11921.955	4887.800	.127	2.439	.015

a. Dependent Variable: Adj Close

Positive sentiments has standardized coefficient ß = 0.127 and p-value of 0.015 < 0.05 therefore the result is statistically significant.

Therefore we can reject the null hypothesis H_0 and this implies there is positive correlation between positive sentiments of bitcoin on twitter and bitcoin price. We can interpret this as follows: The increase of positive sentiments on social media will also increase the stock prices.

4.2.2 UBER - Machine learning-based sentiment result

Dependent variable: Adjusted Closing Price

Independent variable: Average negative sentiments (-)

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	50.302	3.630		13.856	.000
	NegAvg	-40.696	9.571	-.332	-4.252	.000

a. Dependent Variable: Adj Close

In this case we found that negative sentiments has standardized coefficient ß = -0.332 and p-value of 0.00 < 0.05 therefore the result is statistically significant.

We can reject the null hypothesis H_0 and this implies there is a negative correlation between negative sentiments of Uber on twitter and Uber share price.

Dependent variable: Adjusted Closing Price

Independent variable: Average positive sentiments (+)

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	33.022	1.174		28.134	.000
	PosAvg	71.451	38.213	.153	1.870	.064

a. Dependent Variable: Adj Close

In this case we found that positive sentiment has standardized coefficient ß = 0.153 and p-value of 0.064 > 0.05 therefore the result is not statistically significant.

We cannot reject the null hypothesis H_0 and this implies there is no relation between negative sentiments of Uber on twitter and Uber share price.

4.2.3 BITCOIN – Lexicon-based sentiment results

Dependent variable: Adjusted Closing Price

Independent variable: Negative sentiments (-)

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	6180.754	662.010		9.336	.000
	NegAvg	24128.633	12864.922	.098	1.876	.062

a. Dependent Variable: Adj Close

Average of negative sentiments has standardized coefficient ß = 0.098 and p-value of 0.062 > 0.05 therefore the result is not statistically significant.

We cannot reject the null hypothesis H_0 and this implies there is no relation between the amount of negative sentiments of Bitcoin on twitter and Bitcoin price.

Dependent variable: Adjusted Closing Price

Independent variable: Positive sentiment average

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	2696.153	1114.249		2.420	.016
	PosAvg	42951.085	10109.552	.218	4.249	.000

a. Dependent Variable: Adj Close

In this case we found that positive sentiments has standardized coefficient ß = 0,218 and p-value of 0.000 < 0.05 therefore the result is statistically significant.

We can reject the null hypothesis H_0 and this implies there is a relation between positive sentiments of Bitcoin on twitter and Bitcoin share price.

4.2.4 UBER – Lexicon-based sentiment results

Dependent variable: Adjusted Closing Price

Independent variable: Negative sentiment (-)

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	46.902	3.679		12.749	.000
	NegAvg	-158.284	48.492	-.261	-3.264	.001

a. Dependent Variable: Adj Close

Average of negative sentiments has standardized coefficient ß = -0.261 and p-value of 0.01 < 0.05 therefore the result is statistically significant.

We can reject the null hypothesis H_0 and this implies there is a relation between negative sentiments of Uber on twitter and Uber share price.

Dependent variable: Adjusted Closing Price

Independent variable: Positive sentiment (+)

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	34.019	5.012		6.788	.000
	PosAvg	9.515	48.204	.016	.197	.844

a. Dependent Variable: Adj Close

Average of positive sentiments has standardized coefficient ß = 0.016 and p-value of 0.844 > 0.05 therefore the result is not statistically significant.

We cannot reject the null hypothesis H_0 and this implies there is no relation between positive sentiments of Uber on twitter and Uber share price.

4.2.5 Comparison

Table 4.4 summarizes the comparison between the results of machine learning-based and lexicon based sentiments classification on Bitcoin and Uber dataset. The significance level being used in this regression analysis is α = 0,05.

Machine Learning-based Sentiments				
Predictor variable	**var name**	**Standardized coeff. ß**	**p-value**	**statistic. Sig.**
BTC				
Negative sentiments average	NegAvg	-0,023	0,667	No
Positive sentiments average	PosAvg	0,127	0,015	Yes
UBER				
Negative sentiments average	NegAvg	-0,332	0,000	Yes
Positive sentiments average	PosAvg	0,153	0,064	No
Lexicon-based Sentiments (Vader)				
Predictor variable	**var name**	**Standardized coeff. ß**	**p-value**	**statistic. Sig.**
BTC				
Negative sentiments average	NegAvg	0,098	0,062	No
Positive sentiments average	PosAvg	0,218	0,000	Yes
UBER				
Negative sentiments average	NegAvg	-0,261	0,001	Yes
Positive sentiments average	PosAvg	0,016	0,844	No

Table 4.4 Comparison between machine learning and lexicon-based approach

5 Discussion

5.1 RQ1: Which machine learning technique has the best performance to classify the sentiment for sentiment on Twitter data?

As discussed in chapter 4, Table 4.1 has revealed the comparison of accuracy between each algorithm that were applied on Uber and Bitcoin dataset respectively. The result for Uber classification performance is not very high, the average accuracy is around 50% to 60% accuracy. However Boosting algorithm has outperformed the other algorithms with accuracy of 0.82.

Similarly, we are also able to get an accuracy of 81.02% using Machine Learning methods in R to classify sentiments for Bitcoin dataset. This result was achieved using the Boosting algorithm. Due to the nature of boosting, it is easy to understand why this is the best result, as the algorithm uses multiple different algorithms to find the best prediction. As it has been mentioned previously, there is no guarantee that an ensemble algorithm will work better than a single algorithm, but in the case of this research the boosting algorithm performed much better than the single algorithms.

The downside to the Boosting algorithm was that it took a lot longer to run than the other algorithms. In the case where the boosting algorithm was not the best, the boosting algorithm often gave the best result. Quinlan (2006) found that Boosting performs better than Bagging. Therefore it has been proven that the Boosting algorithm still performed better than the non-ensemble algorithms.

The Naive Bayes algorithm had the worst results, with accuracy achieved is only 0.6579 for Bitcoin dataset and 0.4559 for Uber respectively. This contradicts the recurrent outcomes from other text classification research mentioned in Chapter 2. This could be because the algorithm assumes independence between words, when in practice this is often not the case. Words may be more likely to appear in a negative comment than a non-negative one, and vice versa.

5.2 RQ2: To what extent are the sentiments on social media can affect the financial performance of a product?

From the results of the analysis we can now answer the relevant research question and the hypotheses. Firstly, we can see that in the Bitcoin case we found significant findings for positive

sentiments. The result showed that an increase in positivity will positively affect the price return of stock. Furthermore, the standardized coefficient beta 0.127 proved to be significant at p-value of 0.015. This implies that we can reject the null hypothesis since at least one of the predictor variables has a statistically significant impact on the adjusted close price. Based on this result, the H1 can be validated, sentiments on social media will have impact on Bitcoin price.

H1: Sentiments on social media will impact Bitcoin price.
At least one of the predictors will have an impact on Bitcoin price.

In our case, the sample only includes data about Bitcoin and can only be generalized to Bitcoin or companies with similar volatility levels. Secondly, in the case of Uber, we can also reject the null hypothesis since at least one of the predictor variables, which is negative sentiment, has a statistically significant impact on returns. Based on this result, the H2 can be validated, sentiments on social media will have impact on Uber price.

H2: Sentiments on social media will impact Uber share price.
At least one of the predictors will have an impact Uber share price.

Again, it has to be noted that these results are conclusive for Uber and can only give us information about Uber or companies with similar volatilities. While we cannot say a lot more about the causality of this difference from the current data, it is still an important finding to our research. Furthermore, there were clear differences in the impact of predictor variables. While positivity was more significant in case of Bitcoin share price, we found more negativity was crucial in case of Uber share price.

5.3 RQ3: Is there a difference between how the prices of Bitcoin and Uber respectively are influenced by the sentiments on social media?

After the experiments are done for both machine learning and lexicon-based sentiment analysis, it is apparent that the results are consistent. As observed in Table 4.1, in machine learning approach, positive sentiments led to significant results for BTC dataset. This implies that there is positive correlation between positive sentiments of Twitter users and Bitcoin price. Standard coefficient beta of 0,127 means that for every increase unit increase of positive sentiments will account for 0,127 unit increase of the bitcoin share price. Similarly, the results of Vader sentiment analysis which is based on lexicon-based approach also showed the same behavior. For Bitcoin dataset,

positive sentiments has shown significant result with standardized coefficient ß = 0,218. This can be interpreted as the increase of every one unit of positive sentiments of Bitcoin on Twitter will lead to approximately 0,219 unit increase of Bitcoin price.

Evidently the result of machine learning approach and lexicon-based approach for Uber dataset also display consistent results. As shown in Table 4.1, positive sentiments as predictor variable did not show significant result for both approaches. This means positive sentiments on Twitter do not influence the fluctuation of Uber stock price. On the opposite, negative sentiments as predictor has led to significant result with p-value = 0,000 for machine learning, and p-value = 0,001 for lexicon-based method respectively. This can be understood as the existence of relationship between Twitter users' negative sentiments and Uber financial performance. For example, for machine learning method, the standardized coefficient Beta is -0,332. The minus sign indicates that the relationship is negative. Every one unit of increase for negative sentiments will cause a decrease of 0,332 unit of Uber share price. This has demonstrated the contradictory behavior between Bitcoin and Uber share price. Despite the opposite nature of sentiments between the two, these outcomes are consistent across the two approaches of machine learning and lexicon-based approach. Therefore we can verify the third hypothesis as correct.

H3: The result between machine learning-based sentiment analysis and lexicon-based analysis are consistent.

One of the factors that may explain these contrasting outcomes between the two dataset is the difference between Uber and Bitcoin in terms of industries and customer behaviors. Bitcoin traders generally post about any kind of news related to cryptocurrency and discuss about the climate of the current trading affairs. This is not the case with Uber users, as they often only tweet when there is a problem with the services or if they want to complain about the customer experiences. It is quite infrequently that a uber user share their positive experience on Twitter because such happenings are expected and not considered valuable to discuss or broadcast to the internet. This justifies how negative sentiments have stronger impacts on Uber share price instead of the positive sentiments.

6 Conclusion

6.1 Conclusion

Based on the obtained results and discussion several conclusions can be drawn. First of all, boosting algorithm has shown satisfactory result with accuracy above 0.8 for both dataset. This number is considerably higher than the accuracy of other techniques that are approximately above 0.6. Due to the nature of boosting, it is understandable how this result is achieved, as the algorithm uses multiple different algorithms to find the best prediction. As it has been mentioned previously, there is no guarantee that an ensemble algorithm will work better than a single algorithm, but in the case of this research the boosting algorithm performed much better than the other single algorithms.

The other main point to acknowledge is that the sharing of opinions on the internet has become very vital on social media and it may impact the sales and valuation of a company. There are a lot of different sources and different networks that people use to receive their information. As such, we can observe a wide range of opinions, sentiments and comments online. As we can see, these can have real-world implications as they can affect the opinions of people and even impact changes in financial markets. This research concludes that there are differences in types of opinions and how they affect different types of companies.

We have seen that more positive sentiments on social media had an impact on Bitcoin price and we have seen that negative sentiments have had an impact on the decrease of Uber share price. These companies had large differences in terms of industry and volatility. As such, we can see that positive and negative sentiments create different impacts on the real-world, in this case particularly on financial performance. Further research in these fields should focus on extending the database of opinions and beliefs on social media. Moreover, the external validity of the study can also be extended with more data on different companies.

Lastly, we have seen the usage of two different methods in this study, namely machine learning method and lexicon-based method. It is apparent that the results of both are consistent. Despite the contradictory nature of how sentiments polairty between the Uber and Bitcoin affect the respective share price, there are coherent outcomes across the two approaches of machine learning and lexicon-based approach. Therefore we can conclude that the experiments proposed in this study can be perceived with acceptable confidence.

6.2 Limitations and Further Work

It is important to understand the limitations of the research. The scope of this study primarily focuses primarily on the stock prices of Bitocoin and Uber. The behavioral effect of positive and negative sentiment is limited to these companies. Further research in this field can provide more detailed research into the impact of different kinds of business and industry types. Higher number of samples can provide a lot more information into the field of sentiment analysis on the social networks. Moreover, another approach that is yet included in this research is the hybrid approach, which combines machine learning and lexicon-based approaches. If the dataset and the methods used in the study can be expanded, the outcome and confidence of this research may be improved greatly as well.

Bitcoin and Uber are renowned examples for their respective industries but there are other types business that are still not investigated and may yield interesting and different results. If the data source and types of corporations are diversified we may be able to show causality relationship beyond the ranges of crypto currency and ridesharing service applications. Likewise, the time span selected for this research is only one year (2019). If this can be prolonged to not only one year but multiple years, the result might be better as well and the outcome could show more significant and convincing results.

If one were interested in looking more broadly location-wise, other mediums and platforms to find opinionated comments would be Facebook for Uber dataset, and Reddit for Bitcoin data, or other specific sites used for crypto currency trading. Reddit is an American discussion website in which users can comment on threads and subthreads. This would allow the analyst to view more in-depth discussions about Bitcoin, compared to the comments used in this research. Although Reddit is an American site, it has millions of users worldwide so the contribution to the world opinion of Bitcoin could be large.

7 References

Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press.

Arulmurugan, R., Sabarmathi, K., & Anandakumar, H. (2019). Classification of sentence level sentiment analysis using cloud machine learning techniques. *Cluster Computing*, *22*(1), 1199-1209.

Ashok, M., Rajanna, S., Joshi, P. V., & Kamath, S. (2016). A personalized recommender system using machine learning based sentiment analysis over social data. 2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS),

Baydogan, C., & Alatas, B. (2018). Sentiment analysis using Konstanz Information Miner in social networks. 2018 6th International Symposium on Digital Forensic and Security (ISDFS),

Collobert, R. (2011). Deep learning for efficient discriminative parsing. Proceedings of the fourteenth international conference on artificial intelligence and statistics,

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.

Dos Santos, C., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers,

Ducange, P., & Fazzolari, M. (2017). Social sensing and sentiment analysis: Using social media as useful information source. 2017 International Conference on Smart Systems and Technologies (SST),

Fatyanosa, T. N., & Bachtiar, F. A. (2017). Classification method comparison on Indonesian social media sentiment analysis. 2017 International Conference on Sustainable Information Engineering and Technology (SIET),

Ghose, A., & Ipeirotis, P. G. (2010). Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE transactions on knowledge and data engineering*, *23*(10), 1498-1512.

Hutto, C., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. Proceedings of the International AAAI Conference on Web and Social Media,

Jain, A. P., & Dandannavar, P. (2016). Application of machine learning techniques to sentiment analysis. 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT),

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3-24.

Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data* (pp. 415-463). Springer.

Mahalakshmi, S., & Sivasankar, E. (2015). Cross domain sentiment analysis using different machine learning techniques. Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO-2015),

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. AAAI-98 workshop on learning for text categorization,

Nair, D. S., Jayan, J. P., Rajeev, R., & Sherly, E. (2015). Sentiment Analysis of Malayalam film review using machine learning techniques. 2015 international conference on advances in computing, communications and informatics (ICACCI),

[Record #48 is using a reference type undefined in this output style.]

Perikos, I., & Hatzilygeroudis, I. (2017). Aspect based sentiment analysis in social media with classifier ensembles. 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS),

Rathan, M., Hulipalled, V. R., Murugeshwari, P., & Sushmitha, H. (2017). Every post matters: a survey on applications of sentiment analysis in social media. 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon),

Rish, I. (2001). An empirical study of the naive Bayes classifier. IJCAI 2001 workshop on empirical methods in artificial intelligence,

Saad, F. (2014). Baseline evaluation: an empirical study of the performance of machine learning algorithms in short snippet sentiment analysis. Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business,

Salvetti, F., Reichenbach, C., & Lewis, S. (2006). Opinion polarity identification of movie reviews. In *Computing attitude and affect in text: Theory and applications* (pp. 303-316). Springer.

Samal, B., Behera, A. K., & Panda, M. (2017). Performance analysis of supervised machine learning techniques for sentiment analysis. 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS),

Saragih, M. H., & Girsang, A. S. (2017). Sentiment analysis of customer engagement on social media in transport online. 2017 International Conference on Sustainable Information Engineering and Technology (SIET),

Sutton, R. S. (1992). Introduction: The challenge of reinforcement learning. In *Reinforcement Learning* (pp. 1-3). Springer.

Wicaksono, A. J. (2016). A proposed method for predicting US presidential election by analyzing sentiment in social media. 2016 2nd international conference on science in information technology (ICSITech),

Zhang, L., Yuan, H., & Lau, R. Y. (2016). Predicting and visualizing consumer sentiments in online social media. 2016 IEEE 13th International Conference on e-Business Engineering (ICEBE),

8 Appendix

Appendix 1: R Code for learning

Bitcoin Dataset

```
library(stringr)
library(tm)
library(RTextTools)
library(e1071)
library(caret)
library(quanteda)

Sys.setlocale("LC_ALL",'C')

# Read raw data and assign column names
rawdata <- read.csv("C:/Users/Try/OneDrive/BIT
Courses/Thesis/R/btc/jan/unclassified.csv", header = TRUE,
              sep = ";", stringsAsFactors = FALSE, encoding = "UTF-8")

names(rawdata) <- c("class", "tweet")

# Convert class into factor
rawdata$class <- as.factor(rawdata$class)

# remove hyperlinks
rawdata$tweet <- str_replace_all(rawdata$tweet,"'","'")
rawdata$tweet <- str_replace_all(rawdata$tweet,"\\n","")
rawdata$tweet <- str_replace_all(rawdata$tweet,"http\\S+\\s*"," ")

# Create corpus from tweets
vecsource <- VectorSource(rawdata$tweet[1:1104])
corpus <- Corpus(vecsource)
corpus

corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removeWords, stopwords('english'))

###################################
# Classification with Naive Bayes #
###################################

# Create document-feature matrix from unigrams and bigrams and apply
to data
corpus1 <- quanteda::corpus(corpus)

# Applying trigrams
```

```
dfm1 <- tokens(corpus1) %>%
  tokens_ngrams(1:3) %>%
  dfm()

dfm2 <- dfm_trim(dfm1, min_docfreq = 3)
dfm2 <- dfm_tfidf(dfm2)

# Convert the dtm into matrices
# mtr <- as.matrix(dfm2[1:883,])
mtr <- as.matrix(dfm2)

modelNB1 <- naiveBayes(mtr[1:800], rawdata$class[1:800], laplace = 0)
predictNB1 <- predict(modelNB1, mtr[801:1104])

# Create confusion matrix
conf_mtr <- confusionMatrix(predictNB1, rawdata$class[801:1104])
conf_mtr

#################################################################
# Classifications with SVM, ME, DT, RF, Bagging and Boosting #
#################################################################

# Create container from the matrix
container <- create_container(mtr, rawdata$class, trainSize = 1:800,
testSize = 801:1104, virgin = FALSE)

# Build the SVM classifier model
modelSVM <- cross_validate(container, 10, 'SVM')
# Calculate the mean accuracy of classified data
modelSVM$meanAccuracy

# Build the MaxEnt classifier model
modelMaxEnt <- cross_validate(container, 10, 'MAXENT')
# Calculate the mean accuracy of classified data
modelMaxEnt$meanAccuracy

# Build the CART (decision tree) classifier model
modelTree <- cross_validate(container, 10, 'TREE')
# Calculate the mean accuracy of classified data
modelTree$meanAccuracy

# Build the Random Forest classifier model
modelRF <- cross_validate(container, 10, 'RF')
# Calculate the mean accuracy of classified data
modelRF$meanAccuracy

# Build the Bagging classifier model
modelBagging <- cross_validate(container, 10, 'BAGGING')
# Calculate the mean accuracy of classified data
modelBagging$meanAccuracy
```

```
# Build the Boosting classifier model
modelBoosting <- cross_validate(container, 10, 'BOOSTING')
# Calculate the mean accuracy of classified data
modelBoosting$meanAccuracy
```

Uber Dataset

```
library(stringr)
library(tm)
library(RTextTools)
library(e1071)
library(caret)
library(quanteda)

Sys.setlocale("LC_ALL",'C')

# Read raw data and assign column names
rawdata <- read.csv("C:/Users/Try/OneDrive/BIT
Courses/Thesis/R/uber/jun/unclassifiednew1.csv", header = TRUE,
          sep = ";", stringsAsFactors = FALSE, encoding = "UTF-8-BOM")

names(rawdata) <- c("class", "tweet")
nrow(rawdata)

# Convert class into factor
rawdata$class <- as.factor(rawdata$class)

levels(rawdata$class)
head(rawdata)

# remove hyperlinks
rawdata$tweet <- str_replace_all(rawdata$tweet,"'","'")
rawdata$tweet <- str_replace_all(rawdata$tweet,"\\n","")
rawdata$tweet <- str_replace_all(rawdata$tweet,"http\\S+\\s*"," ")

# Create corpus from tweets
vecsource <- VectorSource(rawdata$tweet[1:1061])
corpus <- Corpus(vecsource)
corpus

corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removeWords, stopwords('english'))

###################################
# Classification with Naive Bayes #
###################################
```

```
# Create document-feature matrix from unigrams and bigrams and apply
to data
corpus1 <- quanteda::corpus(corpus)

# Apply trigrams
dfm1 <- tokens(corpus1) %>%
  tokens_ngrams(1:3) %>%
  dfm()

# Taking only tokens with min frequency = 3
dfm2 <- dfm_trim(dfm1, min_docfreq = 3)
dfm2 <- dfm_tfidf(dfm2)

# Convert the dtm into matrices
mtr <- as.matrix(dfm2)

modelNB1 <- naiveBayes(mtr[1:800], rawdata$class[1:800], laplace = 0)
predictNB1 <- predict(modelNB1, mtr[801:1061])

# Create confusion matrix
conf_mtr <- confusionMatrix(predictNB1, rawdata$class[801:1061])
conf_mtr

#################################################################
# Classifications with SVM, ME, DT, RF, Bagging and Boosting #
#################################################################

# Create container from the matrix
container <- create_container(mtr, rawdata$class, trainSize = 1:800,
testSize = 801:1061, virgin = FALSE)

# Build the SVM classifier model
modelSVM <- cross_validate(container, 10, 'SVM')
# Calculate the mean accuracy of classified data
modelSVM$meanAccuracy

# Build the MaxEnt classifier model
modelMaxEnt <- cross_validate(container, 10, 'MAXENT')
# Calculate the mean accuracy of classified data
modelMaxEnt$meanAccuracy

# Build the CART (decision tree) classifier model
modelTree <- cross_validate(container, 10, 'TREE')
# Calculate the mean accuracy of classified data
modelTree$meanAccuracy

# Build the Random Forest classifier model
modelRF <- cross_validate(container, 10, 'RF')
# Calculate the mean accuracy of classified data
modelRF$meanAccuracy
```

```
# Build the Bagging classifier model
modelBagging <- cross_validate(container, 10, 'BAGGING')
# Calculate the mean accuracy of classified data
modelBagging$meanAccuracy

# Build the Boosting classifier model
modelBoosting <- cross_validate(container, 10, 'BOOSTING')
# Calculate the mean accuracy of classified data
modelBoosting$meanAccuracy

# Build the Bagging classifier model
modelBagging <- train_model(container, 'BAGGING', kernel = 'linear')
# Make prediction based on the trained model
resultBagging <- classify_model(container, modelBagging)
# Calculate the recall accuracy of classified data
recall_accuracy(rawdata$class[801:1061], resultBagging$BAGGING_LABEL)
```

Appendix 2: R Code for predicting Bitcoin Dataset

```
library(stringr)
library(tm)
library(RTextTools)
library(e1071)
library(caret)
library(quanteda)

Sys.setlocale("LC_ALL",'C')

# Read raw data and assign column names
rawdata <- read.csv("C:/Users/Try/OneDrive/BIT
Courses/Thesis/R/btc/dec/unlabeled12.csv", header = TRUE,
              sep = ";", stringsAsFactors = FALSE, encoding = "UTF-8")

names(rawdata) <- c("class", "tweet")

# Convert class into factor
rawdata$class <- as.factor(rawdata$class)

# remove hyperlinks
rawdata$tweet <- str_replace_all(rawdata$tweet,"'","'")
rawdata$tweet <- str_replace_all(rawdata$tweet,"\\n","")
rawdata$tweet <- str_replace_all(rawdata$tweet,"http\\S+\\s*"," ")

##########################################
# Classification of new unlabeled dataset #
##########################################
```

```
# Create corpus from tweets
vecsource <- VectorSource(rawdata$tweet[1:8683])
newcorpus <- Corpus(vecsource)
newcorpus

newcorpus <- tm_map(newcorpus, content_transformer(tolower))
newcorpus <- tm_map(newcorpus, removeNumbers)
newcorpus <- tm_map(newcorpus, removePunctuation)
newcorpus <- tm_map(newcorpus, stripWhitespace)
newcorpus <- tm_map(newcorpus, removeWords, stopwords('english'))

newcorpus <- quanteda::corpus(newcorpus)
dfm1 <- tokens(newcorpus) %>%
  tokens_ngrams(1:3) %>%
  dfm()

dfm2 <- dfm_trim(dfm1, min_docfreq = 3)
dfm2 <- dfm_tfidf(dfm2)

# Convert the dtm into matrices
mtr_new <- as.matrix(dfm2)

# Create container from the matrix
container_new <- create_container(mtr_new, rawdata$class, trainSize =
1:1104, testSize = 1105:8683, virgin = TRUE)

modelBoosting <- train_model(container_new, 'BOOSTING', kernel =
'linear')

# Apply the classifier
resultNewData <- classify_model(container_new, modelBoosting)

# Append the classification result to unlabeled dataset
finaldata <-  data.frame(class = resultNewData$LOGITBOOST_LABEL, tweet
= rawdata$tweet[1105:8683])

# Write everything to file
write.csv(finaldata, "C:/Users/Try/OneDrive/BIT
Courses/Thesis/R/btc/dec/labeled12.csv")

timeseries <- read.csv("C:/Users/Try/OneDrive/BIT
Courses/Thesis/R/btc/dec/labeled12.csv", header = TRUE,
                       sep = ",", stringsAsFactors = FALSE, encoding =
"UTF-8-BOM")
```

Uber Dataset

```
library(stringr)
library(tm)
```

```
library(RTextTools)
library(e1071)
library(caret)
library(quanteda)

Sys.setlocale("LC_ALL",'C')

# Read raw data and assign column names
rawdata <- read.csv("C:/Users/Try/OneDrive/BIT
Courses/Thesis/R/uber/dec/unlabeled12.csv", header = TRUE,
                    sep = ";", stringsAsFactors = FALSE, encoding =
"UTF-8-BOM")

# Convert class into factor
rawdata$class <- as.factor(rawdata$class)

# remove hyperlinks
rawdata$tweet <- str_replace_all(rawdata$tweet,"'","'")
rawdata$tweet <- str_replace_all(rawdata$tweet,"\\n","")
rawdata$tweet <- str_replace_all(rawdata$tweet,"http\\S+\\s*"," ")

##########################################
# Classification of new unlabeled dataset #
##########################################

# Create corpus from tweets
vecsource <- VectorSource(rawdata$tweet[1:7898])
newcorpus <- Corpus(vecsource)
newcorpus

newcorpus <- tm_map(newcorpus, content_transformer(tolower))
newcorpus <- tm_map(newcorpus, removeNumbers)
newcorpus <- tm_map(newcorpus, removePunctuation)
newcorpus <- tm_map(newcorpus, stripWhitespace)
newcorpus <- tm_map(newcorpus, removeWords, stopwords('english'))

# Apply trigrams
newcorpus <- quanteda::corpus(newcorpus)
dfm1 <- tokens(newcorpus) %>%
  tokens_ngrams(1:3) %>%
  dfm()

# Taking only tokens with min frequency = 3
dfm2 <- dfm_trim(dfm1, min_docfreq = 3)

# Convert the dtm into matrices
mtr_new <- as.matrix(dfm2)
dfm2 <- dfm_tfidf(dfm2)

# Create container from the matrix
```

```
container_new <- create_container(mtr_new, rawdata$class, trainSize =
1:1061, testSize = 1062:7898, virgin = TRUE)

modelBoosting <- train_model(container_new, 'BOOSTING', kernel =
'linear')

# Apply the classifier
resultNewData <- classify_model(container_new, modelBoosting)

# Append the classification result to unlabeled dataset
finaldata <-  data.frame(class = resultNewData$LOGITBOOST_LABEL, tweet
= rawdata$tweet[1062:7898])

# Write everything to file
write.csv(finaldata, "C:/Users/Try/OneDrive/BIT
Courses/Thesis/R/uber/dec/labeled12.csv")

timeseries <- read.csv("C:/Users/Try/OneDrive/BIT
Courses/Thesis/R/uber/dec/labeled12.csv", header = TRUE,
          sep = ",", stringsAsFactors = FALSE, encoding = "UTF-8-BOM")
```