THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

PREDICTING BITCOIN PRICE WITH SENTIMENT ANALYSIS OF TWITTER AND NEWS DATA BY INCLUDING INDIVIDUAL PREDICTION RATES

Master's Thesis

AARON NATHAN YAFFE

ISTANBUL, 2019

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

BIG DATA ANALYTICS AND MANAGEMENT

PREDICTING BITCOIN PRICE WITH SENTIMENT ANALYSIS OF TWITTER AND NEWS DATA BY INCLUDING INDIVIDUAL PREDICTION RATES

Master's Thesis

Aaron Nathan Yaffe

Supervisor: ASSIST. PROF. DR. SERKAN AYVAZ

ISTANBUL, 2019

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
BIG DATA ANALYTICS AND MANAGEMENT

Name of the thesis: Predicting Bitcoin Price with Sentiment Analysis of Twitter and News Data by Including Individual Prediction Rates
Name/Last Name of the Student: Aaron Nathan Yaffe
Date of the Defense of Thesis: 05 August 2019

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assist. Prof. Dr. Yücel Batu SALMAN
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Serkan AYVAZ
Program Coordinator

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

<u>Examining Comitee Members</u> <u>Signature</u>

Thesis Supervisor ---------------------------------
Assist. Prof. Dr. Serkan AYVAZ

Member ---------------------------------
Assist. Prof. Dr. Tarkan AYDIN

Member ---------------------------------
Assist. Prof. Dr. Atınç YILMAZ

ABSTRACT

PREDICTING BITCOIN PRICE WITH SENTIMENT ANALYSIS OF TWITTER AND NEWS DATA BY INCLUDING INDIVIDUAL PREDICTION RATES

Aaron Nathan YAFFE

Big Data Analytics and Management

Thesis Supervisor: Assist. Prof. Dr. Serkan AYVAZ

August 2019, 62 pages

This study investigates the correlation between the price change of Bitcoin and the sentiment towards Bitcoin on Twitter and various different news sources, by taking into account the accuracy each user and source has had at predicting the price movement of Bitcoin. Bitcoin is a cryptocurrency, a digital form of currency. You can buy and sell goods using Bitcoin like many types of currency, however due to its current high price and volatility some people view Bitcoin as they would an asset or a stock, an investment opportunity instead of a form of currency, unlike the name cryptocurrency suggests.

There have been numerous studies that have investigated and tried to predict the price of a stock or cryptocurrency with the output of Twitter Sentiment Analysis. Sentiment Analysis is the process of classifying a piece of text to be either positive or negative. This study will investigate the impact of the accuracy measure for each user and source on the accuracy of determining the price change of Bitcoin.

Keywords: User Accuracy, Sentiment Analysis, News article Sentiment, Twitter Users Sentiment

ÖZET

TWITTERDAKI KULLANICI YORUMLARINI VE HABER SITELERINDEKI HABERLERI DUYGU ANALIZI EDEREK, KULLANICILARIN DOGRULUK ORANI KULLANARAK BITCOIN FIYAT TAHMINI

Aaron Nathan YAFFE

Büyük Veri Analitiği Ve Yönetimi

Tez Danışmanı: Dr. Öğr. Üyesi Serkan AYVAZ

Ağustos 2019, 62 sayfa

Bu araştırma Twitter`daki ve farklı haber sitelerindeki Bitcoin hakkında yazılan görüşlerin Bitcoin fiyat değişimine nasıl etki ettiğini incelemek üzere yazılmıştır. Her kullanıcının geçmişteki tahmininde ne kadar doğru olduğu göz önüne alınmıştır. Bitcoin crypto para olarak adlandırıan dijital formu olan bir para birimidir. Diğer para birimlerinde olduğu gibi Bitcoin kullanarak mal alıp satabilirsiniz fakat şu anda çok yüksek fiyatlı olduğu ve fiyat hareketleri çok olduğu için bir para birimi olarak değil de yatırım fırsatı olması açısından varlık ya da hisse senedi olarak gören insanlar var.

Şimdiye kadar Twitter`da yazılan görüşleri inceleyerek bitcoinin ve hisselerin fiyatlarını tahmin etmek için bir çok duyarlılık analizi yapıldı. Bu duyarlılık analizleri bir düşüncenin olumlu ya da olumsuz olduğunu belirtmek üzere yapılır. Bu araştırma her bir kullanıcının ve kaynağın bitcoin fiyat değişimindeki doğruluğunu araştırmaktadır.

Anahtar Kelimeler: Duygu Analizi, Haber Kaynaklarin Duygu Analizi, Twitter Kullanicilarin Duygu Analizi

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

API	:	Application Program Interface
NB	:	Naive Bayes
SVM	:	Support Vector Machine
RSS	:	Rich Site Summary

1. INTRODUCTION

Due to the amount of time society is spending online these days, the amount of digital information we leave online keeps on increasing, creating a digital footprint. There are more than 300 million monthly active users on Twitter alone [43]. Not to mention that Twitter has more than 140 million daily tweets (Gupta et al. 2017). Users post posts about how they feel about things and generally their daily events on platforms like Twitter, Facebook, Instagram, Myspace, etc. Users can also leave "likes" to show that they liked the tweet, one could say therefore, validating or agreeing with a piece of information shared. As more and more people are willing to put their opinions online, companies and entities have started to analyse these comments. Naturally, companies want to know if users that have used their products have had a good experience with their product or negative experience with their product. This is why analysing these comments has become a big process in decision making for many companies. This analysis is not only conducted on company product reviews. There have been studies to show that there is a correlation between Twitter sentiment and the next day stock prices (Pagolu et al. 2016). There have been studies conducted online to determine if Twitter sentiment could predict election outcomes as well (Bovet, Morone, Makse 2018).

This thesis interprets social media data as public opinion, since in fact it can be used as a sample of the opinion of the general public but doesn't necessarily represent factual evidence. News sources on the other hand are a much more factual source of information. Journalists don't write news articles just based on how they feel about a subject, they mostly use facts. These facts may be objective in nature, however may contain sentiment depending on the lexicon we use. For example, let's say that a news article's title is, "The price of Bitcoin increased 80% yesterday", which might be a factual statement which one would then define as an objective statement. However one can interpret this as a positive statement due to the use of the word "increased". If domain specific words are added to a lexicon one could conduct sentiment analysis on news articles themselves. Due to the amount of time society spends on their phone and online in general news sources have started putting news articles online. The information on a news website which is updated frequently contains titles, text, headers; each can be analysed with the same methods used to analyse Twitter data with sentiment analysis. If both Twitter data and news article data was analysed one could see if the news articles affect the sentiment of the public, and if the price of Bitcoin is affected by both.

One source of data representing the public's opinion about a matter and the other source representing actual evidence. As more tools start to develop the way we analyse this part of data has started to develop as well. There have been multiple studies conducted with Twitter and sentiment analysis to determine either an election outcome or the sales of a certain product. During these studies the sentiment polarity extracted from tweets is used as a factor in the analysis. Some studies have used likes and retweets as factors as well. Each user`s “Trust value” was taken into account for each user as well by Vivek Seghal and Charles Song(2007) when the trust value was taken into account it had higher accuracy to predict the movement for certain stock, however it decreased the accuracy to predict other stock movements.

The main focus of this research is to determine if each user’s past accuracy rate should be taken into account while analysing both news sentiment and Twitter sentiment to determine the price of Bitcoin or to value any other asset. To simplify our ideology, imagine all Twitter users claim to be psychic and make psychic “predictions” all the time about stock movement, if one user is more often correct than another then they should be taken more into account than a user that has made more incorrect “predictions”. It is safe to assume an investor would rather take investment advice from users that have been more accurate in the past than users that have been less accurate with their “predictions”. So instead of using sentiment analysis to analyse the current sentiment the population has about a specific subject, we will be using it to determine each user’s prediction and prediction rate. So in conclusion this research aims to see if a user prediction rate should be taken into account and the parts of the theoretical streaming methodology needed to create such a system.

1.1 SENTIMENT ANALYSIS

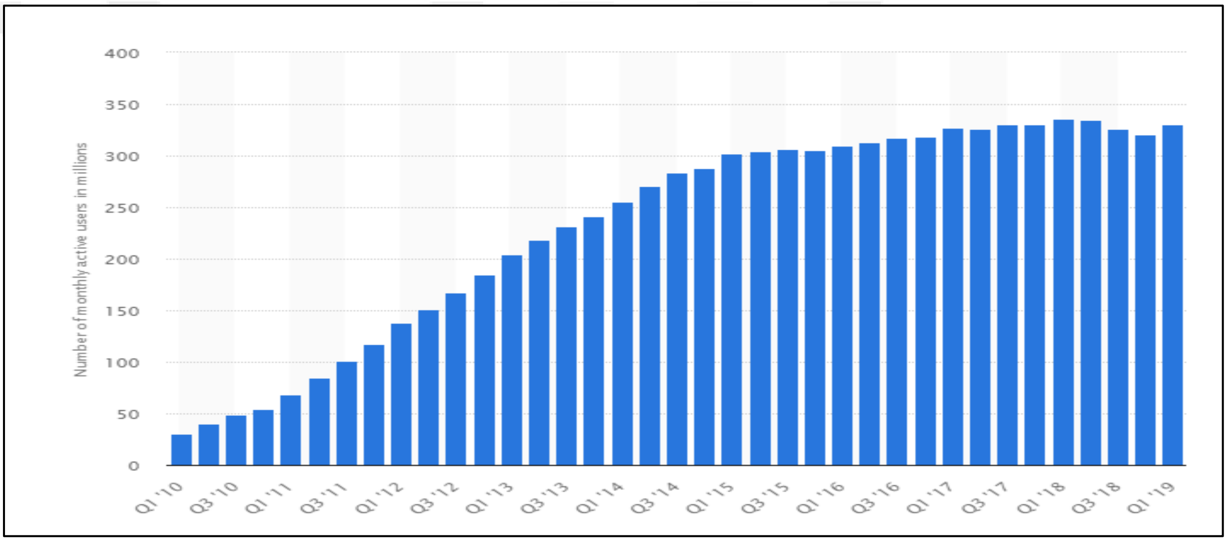
Sentiment analysis can be categorized as both Natural Language Processing and Concept Attraction Task of text analytics (Miner 2012). It refers to the "general method to extract polarity and subjectivity from semantic orientation"(Sarlan, Nadam and Basri 2014). Identifying what others think about a specific topic has always been an important piece of information. With sentiment analysis social media can be used to predict real life outcomes of certain events (Pagolu, et al. 2016). There have been studies conducted that categorize sentiment polarity into two groups, negative and positive, more recently ordinal categorization has been used as well (Saif 2012; Nakov 2016). Ordinal categorization of sentiment for example would be categorizing each tweet into the categories of extreme negative, negative, neutral, positive, extreme positive.

There are different approaches for extracting sentiment automatically. This research will be utilizing both lexicon-based approach and the machine learning approach to create a sentiment classifier. Rule-based, Lexicon based approach is where a list of positive and negative words are created beforehand and this list is used to evaluate a given tweet’s sentiment. For example a list of positive lexicons would be "good, great, increase". For every positive sentiment word that is used the tweet is given +1, for every negative the tweet is given –1 points, this is also called extracting the sentiment polarity of each word. The drawback of this approach is that we must include key industry words to both positive and negative lists, and analysing slang or abbreviations is very hard using this approach. After defining the sentiment polarity for each word in a sentence then the sentiment polarity of the sentence is calculated by adding up each value in the sentence. If sentence polarity is greater than 0 it is given the value +1 for positive sentiment, if sentiment polarity is lower than 0 it is given a value of -1 for negative sentiment.

1.2 BRIEF EXPLANATION OF BITCOIN AND TWITTER

Twitter is a social media platform where a user can send a post, a tweet. A tweet is simply a limited string of characters. Twitter's current character limit is 280 characters, the limit used to be 140 characters. To join Twitter one simply needs to go to their webpage and sign up. As you can see in Figure 1.1 taken from statitica.com, Twitter currently has an average of 330 million monthly users.

Figure 1.1: Monthly Active Twitter Users



As mentioned before Twitter data has been used before to forecast election results, to rate customer satisfaction of products, to forecast the price of a given asset, even to detect who may be suicidal (Pak 2010;Li 2016; Birjali 2017; Bovet 2018).

Bitcoin is a form of cryptocurrency. Cryptocurrencies are a form of digital currency. Bitcoin is arguably one of the most known cryptocurrencies in the world. That being said its price is very volatile and a lot of people perceive Bitcoin as they would an asset or a stock. As an investment opportunity.

1.3 RESEARCH DEFINITION

After the sentiment polarity of the given text is extracted it can be used as the only factor to determine sentiment score of that specific comment to analyse the relation between Twitter, tweets and real live events.

The aims of this this research are to:

a. To determine if each users past sentiment “prediction rate” should be taken as a factor while calculating the sentiment value of a given tweet or news article.

Take any kind of stocks or cryptocurrency, if the past prediction rate of a user is taken into account by saving the current prediction rate and the amount of predictions a user has made, and then multiplying this with the sentiment polarity to determine sentiment score, how would that impact the ability to overall predict the price of a given stock or cryptocurrency. The hypothesis is that if user accuracy rate is taken into account then the overall accuracy of the model will increase, since the sentiment score of a given day would be calculated by giving a higher weight to the users that have a higher accuracy.

b. Create the basis for a theoretical streaming framework that will update user accuracy while making predictions according to the updated data.

For this purpose, this research will analyze the sentiment towards the keywords Bitcoin and BTC for both Twitter users and RSS sources of certain news sources like BBC, Google News, CNN. In summary this research is about determining if there is a correlation between the price of Bitcoin and the sentiment from news sources and Twitter. The way it differs from other Sentiment Analysis research conducted on Twitter is the factors it takes into account while predicting the price of Bitcoin.

2. LITERATURE SUMMARY AND THEORETICAL BACKGROUND

In this chapter I will give a brief explanation of Sentiment Analysis and the different methods used to create a sentiment classifier, review literature that has dealt with Sentiment Analysis of social media posts and news posts to predict the price of stocks or cryptocurrencies, and give a brief explanation about the stock market and financial theories behind stock prediction.

2.1 SUBJECTIVITY AND OBJECTIVITY

Subjectivity is defined by the Online Cambridge Dictionary as “the influence of personal beliefs or feelings, rather than fact”. For example, if an investor has had a bad experience investing in a certain stock market and has lost a lot of money on it, he may make the claim “you can only lose money in the stock market”. Due to his bad experience he is making this comment. However this is a subjective claim since it does not state any objective facts, some investors make millions by investing in stocks, others lose. It may be a statement and opinion however due to his past experience and feelings he is making a subjective claim.

Objectivity on the other hand is defined by the Online Cambridge Dictionary as “the fact of being based on facts and not influenced by personal beliefs or feelings”. For example, if the same investor made the claim, “I lost a lot of money on the stock market,” that would be an objective statement, since even though the investor has had a bad experience with the market he is not making the claim that it's impossible to not make a profit from the stock market. It's important to keep in mind that “Objective sentences can imply opinions or sentiments due to desirable and undesirable facts.”(Liu 2015) even though the investor has tweeted an objective tweet, due to the undesirable fact that this specific investor has lost money investing, one could classify this tweet as negative.

Subjectivity is natural, most claims made by people in general are subjective in nature. Our past experiences influence our decisions today and what we say. It is one reason why racists exist in this world, after having had one bad experience with a certain person that is of a certain race, they feel that all of the people of that race will act the same way and therefore will make subjective claims about that specific race of people. Subjectivity isn't necessarily a bad thing, it is just the expression of personal views and beliefs about a given subject. Comments on Twitter or Facebook or Myspace are mostly subjective ones. They are statements that are made by the influence of one's beliefs. There are some objective tweets, stating facts or statistics about a given matter but mostly tweets are subjective by nature. Alexandre Pak (2010) collected 216 tweets to test a sentiment classifier that was trained using twitter data that was labeled according to the emoticons the specific tweet contained. Out of the 216 tweets she had labeled, 33 were objective while 183 where subjective (Pak 2010). Newspapers on the other hand are supposed to be objective. I use the word supposed to be because as mentioned before an objective statement can still be assigned with positive or negative sentiment.

2.2 SENTIMENT ANALYSIS

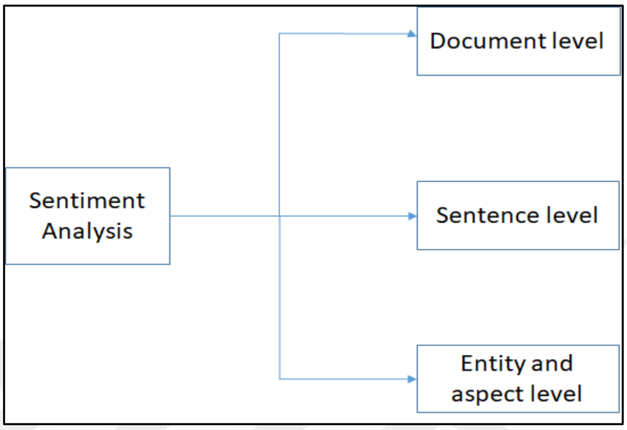
Classifying a piece of text has been a field that has been around for a long period of time. Initially topical categorization was the main focus of this field, “attempting to sort documents according to their subject matter” (Pang, Lee & Vaithyanathan, 2002), “Although linguistics and natural language processing (NLP) have a long history, little research had been done about people’s opinions and sentiments before the year 2000.” (Liu 2015). Both classification of a piece of text according to its subject and classification of a piece of text according to its sentiment could be classified as a text mining task or a NLP (Natural Language Processing) task.

There are various different NLP tasks, the most common used today, without people even realizing it is being utilized, is speech to text and text to speech. When a user tells Siri, the iPhone assistant, to find something on Google, what initially happens is that the words that the user speaks first need to be changed to text, a speech to text task. After Siri finds a result for a query asked by the user then again a piece of text needs to be changed into a speech, text to speech task. There are numerous different NLP tasks. Text mining which I would categories as a subcategory of NLP tasks would include, Document summarization, Document categorization and Sentiment Analysis.

Liu describes Sentiment Analysis as a field which analyzes people’s sentiments, opinions and emotions towards entities such as products and organizations (Liu 2012). Sentiment Analysis is often called Opinion Mining as well, since it is used to derive the opinion of a given person. Lie’s description leads one to believe the idea that Sentiment Analysis can only be used to analyse only subjective comments. However one needs to keep in mind that even objective statements can be classified as positive or negative depending on the way you analyze them. For example if Apple's sales increased 50 % in a year and somebody tweeted or a news article headline was “Apple's sales have increased 50% since last year”, that statement would be objective since it is stating the facts. Even though the statement is objective we could derive that it has a positive sentiment if we include the word “increase” to our positive domain specific lexicon. Why would we do this? For company stocks one would expect that an increase in sales would have a positive impact on the company's stock itself. So a sentiment value can be derived from both objective comments and subjective comments depending on what words in a lexicon are seen as positive and what words are seen as negative. Therefore even though news articles are objective pieces of information by nature they still may provide a sentiment score.

Sentiment Analysis has been used in multiple areas and domains. With more and more comments and information being put freely online the interest of analysing this information to derive knowledge is bound to grow. There has been Sentiment Analysis conducted to predict stock market prices, presidential elections, sales of movie tickets (Pang 2002; Liu 2007; Pagolu 2016; Bovet 2018). There has even been a study that looked at tweets and tried to predict who might be thinking of committing suicide, which I would categorize as a study in psychology (Birjali, Beni-Hssane & Erritali 2017). So the applications of Sentiment Analysis is very vast. One could make the claim that the initial Sentiment Analysis that was conducted would probably have been by marketeers. Even though I don't have any hard factual proof to back this claim up as far as I know some marketeers would have needed to manually go through user reviews and comments to determine if a product or company brand was seen positively by the public. When a company only received 1 or 2 tweets a day a task like this could be done manually but imagine manually going through millions or trillions of tweets and categorizing them to be either positive or negative, remember there are more than 140 million tweets sent out in a day (Gupta et al. 2017). Having a machine classify the tweets for a company with an accuracy rate of 90% is much more cost effective then hiring a whole team to classify these tweets or comments.

Figure 2.1: Different Levels of Sentiment Anlaysis



Sentiment Analysis has been mainly investigated in three different levels, as shown in Figure 2.1 Document level, Sentence level and Entity and Aspect level. The Document level analysis, analyses a document as a whole to determine if the document has positive or negative sentiment. “This level of analysis assumes that each document expresses opinions on a single entity” (Liu 2015).

The Sentence level analysis, analyses each sentence in a given document. It analysis each sentence and determines if it is positive or negative or neutral. Throughout my research I have observed that this analysis is also used to reduce the size of a corpus before Document level analysis is conducted especially when using a machine learning approach to create a Sentiment classifier. Removing the sentences that do not contain sentiment and then turning the document into a Bag of Words increases the Sentiment classifiers accuracy to predict sentiment, since you are removing “noise” sentences and therefore words that do not indicate the actual sentiment of the document itself.

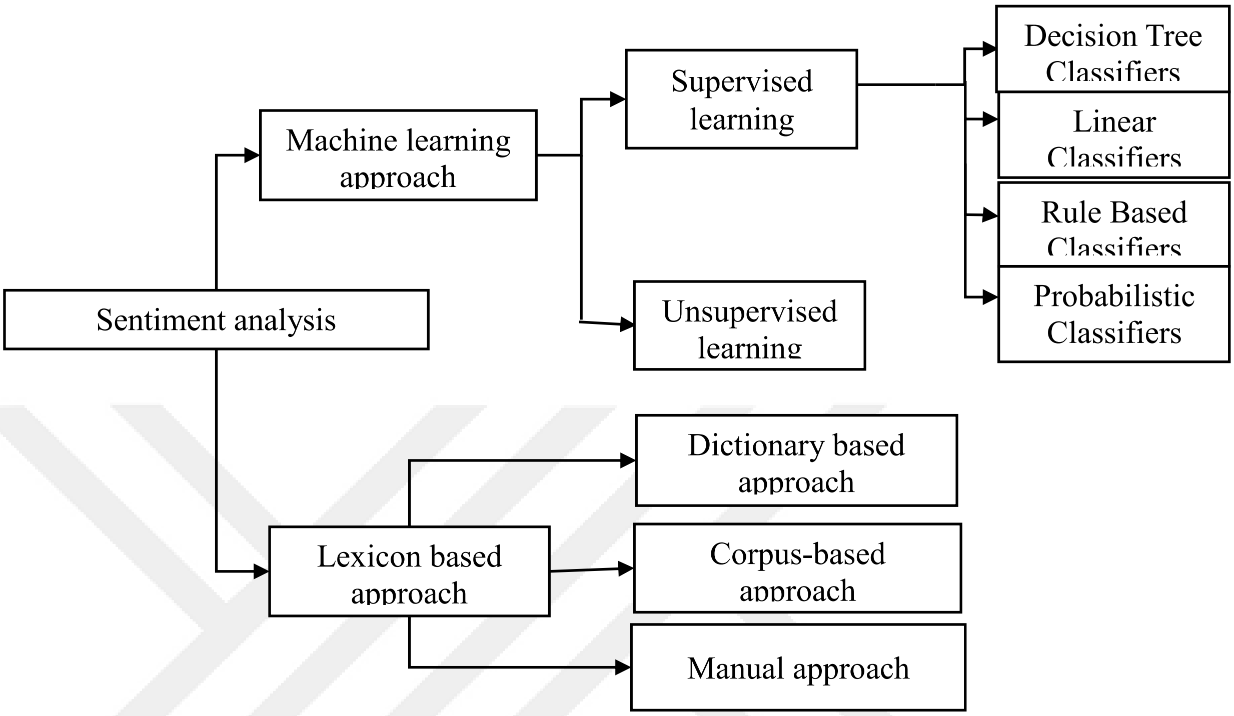
Entity and Aspect level sentiment analysis dives a bit deeper than Sentence level and Document level analysis. “It is based on the idea that an opinion consists of a sentiment (positive or negative) and a target (of opinion).”(Liu 2015). For example take the sentence, “The acting in the movie wasn't that good, however I have to say it was an entertaining movie”. The sentence contains positive sentiment about the movie, “entertaining movie”, however it contains negative sentiment about the acting in the movie. This level of sentiment analysis should theoretically manage to pick up both target matters and both the positive and negative sentiment that this sentence contains. However this level of sentiment analysis is very complex and has a lot of problems that must be dealt with.

The Bag of Words method is commonly utilized to represent a piece of text. “According to the BOW model, the document is represented as a vector of words in Euclidean space where each word is independent from others.” (Kolchyna 2015). The Bag of Words can be thought of as a document that is separated by each word, aka a collection of unigrams. There has been some analysis conducted that use bigrams as well to represent a document, the difference is that in the sentence “the new iPhone is amazing” unigrams would separate this piece of text as “the””new””iPhone””is””amazing” whereas biograms would separate it as “the new””new iPhone””iPhone is””is amazing”.

Noun Phrasing and Named Entities are other methods to represent a piece of text. “Noun Phrasing is accomplished through the use of a syntax where parts of speech (i.e., nouns) are identified through the aid of a lexicon and aggregated using syntactic rules on the surrounding parts of speech, forming noun phrases.”(Schumaker & Chen 2009).

As you can see in Figure 2.2, there are two main methods of Sentiment Analysis, the lexicon approach and the machine learning approach (Taboada 2011;Sarlan 2014 ;Kolchyna 2015 ; Liu 2015). Both these approaches utilize the Bag of Words method to represent a piece of text. “In the machine learning supervised method the classifiers are using the unigrams or their combinations (N-grams) as features. In the lexicon-based method the unigrams which are found in the lexicon are assigned a polarity score, the overall polarity score of the text is then computed as sum of the polarities of the unigrams.”(Kolchyna 2015).

Figure 2.2 : Sentiment Analysis Approach



2.2.1 Machine Learning Approach

Sentiment Analysis when looked at in its simplest form is just a classification task, to classify if a given text is of a positive or negative nature. A text classification task classifies the topic of a given text by using topic related words as a key feature, sometimes classifying them by subject matter. Sentiment analysis simply classifies documents into two topics, positive or negative. Machine learning algorithms can be utilized to determine a piece of text's sentiment. "This approach requires labeled data to train classifiers" (Saif, He & Alani 2012). The machine learning approach is when a given piece of text is labeled, therefore creating training data and by utilizing different algorithms like Naïve Bayes or Neural networks a Sentiment classifier is created. "Naïve Bayesian, represents each article as a weighted vector of keywords" (Salloum, et al.2017). The polarity of text documents is detected according to the classifier that has been trained by the labeled dataset.

“The majority of the statistical text classification research builds Support Vector Machine classifiers, trained on a particular data set using features such as unigrams or bigrams, and with or without part-of-speech labels, although the most successful features seem to be basic unigrams”(Taboada et al. 2011). There has been high accuracy determining sentiment polarity while using the machine learning approach however it also seems to be domain specific. Bo Pang and Lillian Lee (2002) showed that, using Bag of Words of documents as features to classify the sentiment of a given text, using Naïve Bayes or SVM performed well. Besides only using Bag of Words as features to train a Sentiment classifier there have been other features used as well which include, parts of speech, sentiment shifter and syntactic dependency (Liu 2015; Brooke 2019). In this study both Naïve Bayes and SVM where utilized and Naïve Bayes outperformed SVM.

2.2.2 Lexicon Based Approach

“The lexicon-based approach involves calculating orientation for a document from the semantic orientation of words or phrases in the document.” (Taboada 2011). It does such by classifying words beforehand as positive or negative, these lists of words are referred to as lexicons and is the reason why it is referred to as lexicon-based approach. Creation of lexicons can be done manually creating a dictionary, or itself can be a machine learning task where again a collection of data is labeled and according to the sentences that are labeled the machine learning algorithm we use determines the sentiment polarity and sentiment score for each word that it comes across and then gives each word in the lexicon a sentiment score. The sentiment score for a piece of text is then determined by aggregating each words sentiment score within that piece of text.

There are three main approaches to create a lexicon, the dictionary based approach, the corpus based approach and the manual approach. The manual approach is the approach where a lexicon is manually created, this approach can be quite time consuming since you need to create an extensive list of positive and negative terms. A dictionary based approach is where a dictionary and thesaurus is utilized to create a completed lexicon. So a user would manually write down positive and negative terms quite like they would do for the manual approach, in this approach the initial list created by the user is referred to as seed words then by using a dictionary or thesaurus's synonyms and antonyms, a lexicon is generated.

A lexicon can be created by using a specific corpus and pre-labeled data as well. This is called the corpus-based approach. The corpus-based approach turns each labeled document into unigrams using the BOW method and then using machine learning methods words are assigned a specific value and put in a lexicon, this is then used to determine the sentiment score of test data. Dung Nguyen (2013) utilized both the machine learning and dictionary based lexicon approach while analysing Twitter data.

SemaEval-2017 contains a list of available lexicons such as SentiWordNet, SenticNet 4, VADER, Yelp Restaurant Sentiment Lexicon (Cortis et al.2017). Vader is one of the lexicons used in this research to test the accuracy of the lexicon method.

2.2.3 Problems in Sentiment Analysis as Regards to Analysing Tweets

There are a set of problems that Sentiment classifiers face when conducting Sentiment Analysis, especially when conducting Sentiment Analysis on social media posts like Twitter. Take sarcasm for example, sarcasm is defined by the online Oxford dictionary as "a way of using words that are the opposite of what you mean in order to be unpleasant to somebody or to make fun of them". So if a given text that is analysed contains sarcasm a Sentiment classifier would need to be able to understand that the text is sarcastic since the words that are used in this sense mean the exact opposite of what they actually mean. Sarcasm normally is detected by how somebody says a given sentence, in fact the example of sarcasm in the online Oxford dictionary is,

"*'That will be useful,' she snapped with heavy sarcasm* (= she really thought it would not be useful at all)*."*

“That will be useful”, being the sentence that contains sarcasm. Take the same sentence, “That will be useful”, read on a computer screen and I wouldn't be able to tell that it was sarcastic, especially if I was only given that one sentence. Alan Ritter, Sara Rosenthal and Fabrizio Sebastani(2016) turned this into a human intelligent task. By utilizing Annotation with Amazon’s Mechanical Turk, they posted a bunch of jobs for users to manually go through a bunch of tweets and manually chose the sentiment of the tweet, the sentiment it had for the given subject of the tweet and whether or not the sentence contained sarcasm. They then used this labeled data as their training set to create their sentiment classifier.

Another issue that one faces during Sentiment Analysis is spam detection. Spam in the Oxford dictionary is defined as, “Irrelevant or unsolicited messages sent over the Internet, typically to a large number of users, for the purposes of advertising, phishing, spreading malware, etc.”. It is common knowledge that spammers especially on social media share the same message on either one account or multiple different accounts to try to influence people. “Spam is a real threat to usefulness of the web. Spammers mask their content as useful or relevant content and hence is delivered to the user.”(Perveen et al.2016). Therefore if the tweets gathered happen to be spam tweets instead of tweets that are created by actual users then the result of any analysis will not be a true representation of the public, in fact spam tweets could skew the results of any analysis since spammers send the same message over and over again. For example, let's say somebody is analysing the sentiment of Twitter for Bitcoin and a spammer has sent over a thousand messages stating, “Bitcoin is great and the price will increase really soon”, besides the spammer for arguments sake, let’s say 50 users have stated negative comments about Bitcoin and only 10 users have stated positive comments, however the spammer has sent over 2000 positive comments using different accounts. If one simply collects all tweets and analyses if the aggregate sentiment is positive or negative the results will indicate that the aggregate sentiment is positive, however that isn't the case.

Another issue with sentiment classification for social media for the lexicon approach is acronyms. Due to the limited amount of characters a user is allowed for each post a lot of users use abbreviations while posting, for example BTW means “by the way”. PANS means “pretty awesome new stuff”. Again while using a lexicon sentiment classifier that has been manually created or created by using a dictionary and has a list of positive and negative words to determine if a given text is positive or negative if these acronyms are not contained in the lexicon the classifier will not be able to identify the acronyms to be positive or negative. Some abbreviations may cause issues as well, for example a very common abbreviation of increased is inc, again if not added to the lexicon manually for a lexicon classifier that uses the manual or dictionary method it again would be hard to classify these tweets . However this issue would not arise in the machine learning approach if the training data contained acronyms and abbreviations.

2.2.4 Literature Review of Sentiment Analysis and Price Prediction

There has been much research conducted on predicting stock prices. Research on stock theory was originally based on Efficient Market Hypothesis (Fama 1965, Fama 1969, Yadav 2017) and Random Walk Theory (Pagolu 2016). According to these theories the raise in stock prices is independent from past stock prices, rather it is dependent on the information about the company that is available to the public, there have been studies that collect not only news articles but financial statements about companies to make stock predictions. “Efficient market hypothesis states that stock market prices are largely driven by new information and follow a random walk pattern”(Mittal 2011). According to EMH claims financial market information is already reflected in the price. According to this theory there are three types of markets, weak-form of efficiency, semi-form of efficiency and strong-form of efficiency. Strong form efficiency being that the stock prices reflected are always based on the available information about that specific stock.

“The past movement or trend of a stock price cannot be used to predict its future movement. The stock prices are fluctuating and status of financial fields of market can be predicted as random walk” (Pagolu 2016).

So according to financial theory, the price of a given stock isn't dependent on past price but on the information that investors have on that price. As this isn't a finance paper we will not go into depth about finance theory. However with this simple explanation on some basic financial theory, one could defend the position to try to predict a given stock by just news sources and social media tweets since according to these theories the past price of stock does not necessarily reflect the future price of a given stock.

As mentioned in the second chapter Sentiment Analysis has been conducted on various different disciplines and studies. Asur (2010) constructed a model that predicted the sales rate of a given movie, the input of this model was how many times the movie was tweeted. He then used Sentiment Analysis to prove that if sentiment extracted from a tweet is also taken into account the prediction rate of the model increases. In his research he also found correlation with the amounts tweeted and movie sales (Asur & Huberman, 2010). This could be applicable for the stocks as well. After all there is an old saying, “Bad Press is Better Than No Press”, now for stocks it may be less unlikely to happen, however again, if an investor is looking for investing advice and sees a lot of tweets about a given stock, even though there are more negative than positive, he may be more inclined to buy that specific stock rather than a stock that he can't find any advice for on Twitter. However as Asur’s (2010) findings for movies indicates that sentiment extraction improves the prediction rate for the model it is not far-fetched to assume that the prediction of price movement of stock will be more accurate when the sentiment of the given tweets is taken into account as well.

Besides utilizing Twitter, other sources have been scraped to analyse and predict future events. Vivek Sehgal and Charles Song (2007) web scraped http : //f inance.yahoo.com and collected 260,000 messages for 52 different stocks. Their model was able to make accurate predictions about Apple stock where it was able to achieve an accuracy of 81% using both sentiment and the Trust Value that they created. Sehgal & Song (2007) calculated Trust value with the formula:

$$\text{Trust Value} = \frac{\text{P redictionScore}}{\text{NumberOfP reictions}} + \frac{\text{Exact Predictions} + \text{Close Predictions}}{\text{NumberOfP redictions} + \text{ActivityConstant}}$$

(2.1)

For Apple stock they managed to increase the accuracy from 72% to 81% by using this method. However for the stock prediction of EXXONMOBILE the accuracy rate only increased from 61% to 62%. Due to the fact that some finance.yahoo data is labeled by the users that post on the blog, Sehgal & Song (2007) were able to use a Machine learning approach to classify each data point. The sentiment analyzer labeled the data, as “StrongBuy”, “Buy”, “Hold”, “Sell” or “StrongSell” according to the labeled training set the classifier was trained with. “The number of features used by each classifier was in the range of 10,000” (Sehgal & Song 2007). They then used the classifiers Decision Tree, Naive Bayes and Bagging to predict the rise or fall for stock using the previous days Trust value and sentiment as input. For Sehgal & Song(2007), using a machine learning approach to construct a sentiment classifier makes perfect sense due to the fact that yahoo data comes pre-labeled, labeled by the writer themselves. However Twitter data or news articles don't tend to come pre-labeled. One could try to train a sentiment classifier according to gathered yahoo data and apply them to Twitter data however sentiment classifiers tend to be less accurate when trained in different domains. The reason being there is no character limitation on what can be posted in the yahoo.finance blog whereas Twitter has a set number of characters a user can use when posting and the user profiles that use these platforms are different as well. While Sehgal & Song have taken into account a “Trust value” similar to this study’s Accuracy score, they also used how frequent a user post about a given subject as an input in calculating that score, they theorized that users that know more about a given stock would post about that stock more frequently. However anyone on Twitter may post as many posts as they want, it doesn't necessarily mean that they have more knowledge about a subject. That's why in this study we will only be looking at the “Accuracy rate” of each user instead of their “Trust Value”. The sentiment score that we will calculate based on accuracy rate sentiment polarity will also take into account the amount of likes the tweet had as well as the amount of retweets a post had.

Xue Zhang (2011) measured the positive and negative emotions that tweets contained and the correlation between stock movements and daily emotions. They found that, “Among all the emotional words, hope, fear and worry work best in this analysis.” (Zhang, Fuehres, & Gloor 2011). Stating that whenever these terms were frequently used the stock price the next day for certain stock market indices would decline. Bollen

(2011) measured mood in tweets by utilizing two tools, “OpinionFinder which measures positive vs. negative mood from text content, and GPOMS” (Bollen, Mao & Zeng 2011) resulting in 7 mood dimensions they analyzed. GPOMS is Google Profile of Mood States and contains 6 different mood alerts, calm, happy, sure, vital and kind. They found that Opinion Finder`s positive vs negative mood didn't have that much correlation with the stock movements, however GPOMS`s dimension labeled “Calm” had a significant correlation with price movements 3 to 4 days after they were posted. Bollen went on to prove that socioeconomic events have an effect on the GPOMs moods that are detected in Twitter , hence one could argue from both his findings that socioeconomic events that are commonly found in news sources effect tweets and that tweets affect the price of stock (Bollen 2009; Bollen 2011). Therefore somewhat backing the idea that using both news sources sentiment and twitter sentiment for a given subject may increase the predictive outcome.

Anshul Mittel (2011) also measured mood in tweets to analyse stock movements, they created a word list based on a Profile of Mood States (POMS) questionnaire. They used a similar way to extend their word list as Bollen (2011) had done “by considering all commonly occurring synonyms of the base 65 words using SentiWordNet and a standard Thesaurus” (Mittal 2011). Anshul Mittel (2011) utilized a lexicon dictionary approach for creating a sentiment classifier that was explained in section 2.2.2 and found a casualty relation with the past 3 days moods and current stock price and reached a 75.6% accuracy using the happy and calm dimensions. Ray Chen (2011) as well measured mood in tweets to analyse stock movement, he did so by creating a lexicon of 5000 words “along with log probabilities of ‘happy’ or ‘sad’ associated with the respective words.” however this method resulted with uncorrelated data so they swapped their list with a list they generated from utilizing SentiWordNet (Chen & Lazer, 2011). Ray Chen (2011) also concluded that there was a correlation between Twitter data and stock market prices however the Twitter data predates the market data by about 3 days, the same as the findings of Bollen (2011) and Anshul Mital (2011). Hence an influence window of up to 3 days should be investigated in further research (Bollen 2011; Chen 2011 ; Mittal 2011).

Stefan Nann, Jonas Krauss and Detlef Schoder (2013) used a number of sources including Twitter, Yahoo! Finance Boards, Investor’s Hub, Investor Village and Raging Bull, they “used a 30-day simple moving average (SMA30) to calculate sentiment values” (Nann, & Krauss & Schoder 2013), they created their sentiment classifier by manually annotating a few hundred tweets as test data, as well as manually adding a key set of words to their lexicon. So they used a combination of the statistical and the manual approach to create their lexicon. This study also confirmed the predictive power that social media and many blogs have. They saw a 60.38% percent prediction accuracy for all stock predictions for the various stocks that were analysed. For their analysis the influence window was only 1 day.

3. DATA AND METHOD

As shown there has been past research on blogs and social media to predict the stock market price movement, some using machine learning approaches, some lexicon approach (Zhang 2011; Mittal 2011;Chen 2011 ; Dung 2013; Sarlan 2014). Only Seghal Vivek (2007) out of all the articles that were read tried to see if an accuracy or "Trust Value" of each user would increase the ability to predict the price of the stock market. News sources have also been used to predict the price of an asset. However both have very rarely been analyzed together.

The underlying theory is that the news articles represent factual evidence themselves, whereas the users of Twitter`s sentiment represent subjective sentiment. As stated before, sentiment can be derived from objective factual statements. As well as trying to determine if the accuracy rate of each news source and user should be taken in to account while performing the same analysis, this study also aims to determine if one can predict the price movement of Bitcoin using both sentiment from news sources and Twitter.

Seghal Vivek (2007) while using a machine learning algorithm collected pre-labeled data from Yahoos finance blog, this labeled data was labeled by users themselves so one could say that the labels represent the true sentiment behind the piece of text since the users that wrote the comment were labeling it themselves. Seghal Vivek (2007) then used "Trust value" to calculate somewhat the accuracy rate of a blogger.

Tweets and news articles on the other hand do not come prelabeled, therefore a lexicon approach was originally utilized. A lexicon method as previously mentioned in the literature review, is the approach of using a predefined list of words to derive the sentiment out of a corpus. For example if our positive word list contained "good, best" and we were analyzing the sentence “ you are the best,” we would tokenize the sentence and then give each word a positive value or negative value depending if it was located in our positive lexicon or negative lexicon. Using the lexicon each tweet and each news title and news description collected was marked as either positive or negative. However the results of the lexicon based sentiment analysis indicated that the sentiment analyzer wasn't classifying the tweets appropriately. Therefore a machine learning algorithm was used to create a sentiment analyzer. To create a sentiment analyzer we first needed to hand label about 800 news source articles as positive or negative and label 1000 tweets as positive, negative or neutral. We then used these labeled datasets as test data. We will go into this further with more detail. After the sentiment analyzer was trained on the labeled training set data then the analyzer was used to predict the sentiment of each tweet and news article.

For this study the classified piece of text’s sentiment will be taken as a “prediction” in a similar to Seghal Vivek’s Trust method measure. Giving a sentiment score to each tweet and then seeing the sentiment was a good indicator for the stock market. If it had a positive sentiment and the price went up or down for the influence windows observed up to 3 day the tweet or news source would receive a Prediction rate and giving each tweet a sentiment score based on the tweets sentiment and the users prediction rate. To see if there is a correlation of Sentiment between news sources sentiment and Twitter sentiment as factors using Decision Tree and regression to determine if news source can be used to accurately predict the price of Bitcoin. News was collected from API’s and RSS feeds. Twitter tweets were gathered from Twitter.

Figure 3.1: Methodology Steps Taken

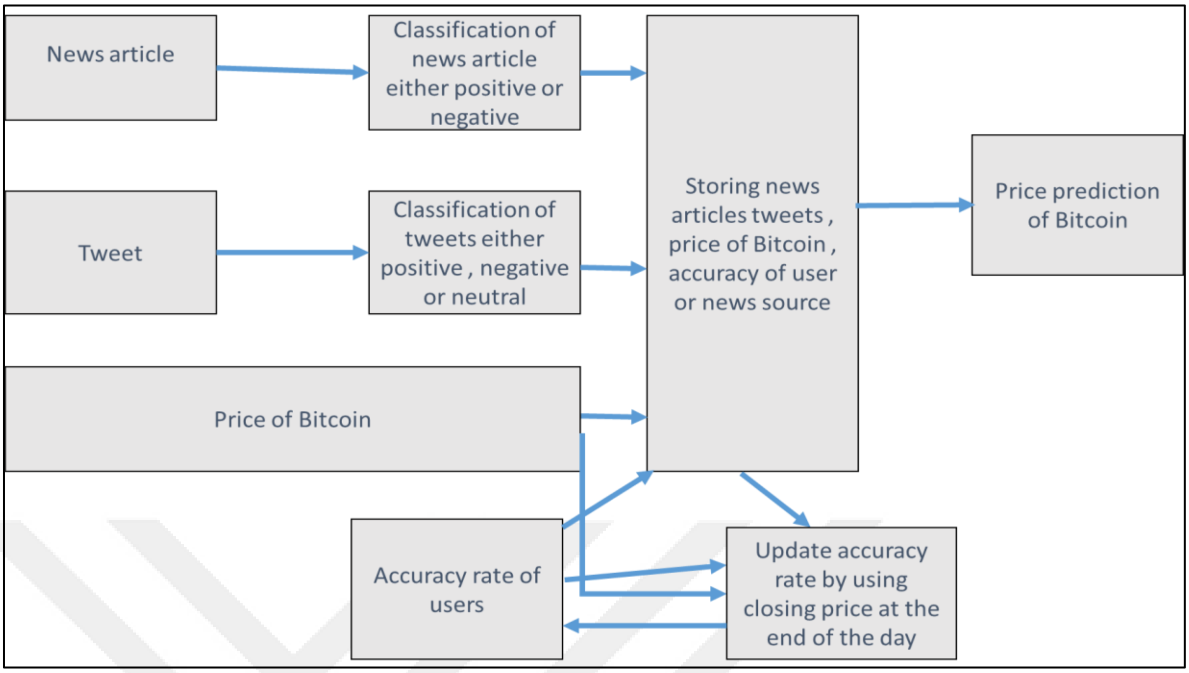

News article
Classification of news article either positive or negative
Tweet
Classification of tweets either positive , negative or neutral
Price of Bitcoin
Storing news articles tweets , price of Bitcoin , accuracy of user or news source
Price prediction of Bitcoin
Accuracy rate of users
Update accuracy rate by using closing price at the end of the day

Figure 3.1 explains the steps taken in order to analyse both Twitter data and news feed titles and descriptions. Originally news article corpus's were going to be analysed as well, however to simplify the process to compare the Twitter data and news sources data only the description and title were utilized in the process.

After collecting news sources and tweets, using the sentiment classifier each article and tweet were classified to be either positive or negative. A lexicon dictionary based approach and a machine learning approach were utilized to create the classifier and each classifier's accuracy rate was compared to determine the best classifier.

The Sentiment classifier that was utilized was designed with a lexicon dictionary method, as mentioned in section 2.2. A sentiment classifier can be created by using a lexicon. A lexicon is a list of words where each word is assigned a sentiment polarity and given the technique used, sentiment score as well. Three different lexicons where utilized. If a sentence in a given tweet contains a word from the positive lexicon +1 was added to the sentence sentiment polarity, or the polarity score given to the word in the designated lexicon. If it contains something from the negative lexicon -1 was added to the sentiment polarity of a sentence, classifying the tweets and news titles using this method into three groups, positive, negative and neutral. Machine learning approach was also utilized, utilizing the Naives Bayes algorithm. By creating a document term matrix and pre labeled dataset a sentiment classifier was trained and then was used to label the unlabeled dataset.

After each item was classified it was stored into a database. For tweet items the user name, sentiment polarity, day of post, and likes the post has gained was extracted. For news sources, the news source, publication date, sentiment polarity of title and description of the cryptocurrency that is mentioned was extracted. After these are extracted, the price of Bitcoin for the day of the posts and the publication date of articles were added as well as their price for the next day, 2 days after posting and 3 days after posting. Then each posts accuracy was determined. If the post had a positive sentiment and the price of the stock increased then it was given the value of 1, if the stock decreased it was given the value of 0. If the post had a negative sentiment and the price increased it was given the value 0 and if the stock decreased it was given the value 1. Then the prediction rate of each user was calculated.

$$\mathrm{PR} = \frac{\sum a}{n}$$

(3.1)

Where n is equal to the number of predictions a user or news source has ever made about any given stocks and ∑a represents the sum of the accuracy value for each prediction a user has made. Hence the prediction rate PR is equal to the amount of times a user has correctly predicted a given outcome divided by the number of predictions he or she has made. Both the accuracy and prediction rate were calculated four times, for the different influence windows chosen.

Even though a tweet may predict the price is going to go up in a month it is still a positive sentiment the author is feeling at this precise moment. With feature extraction it may be possible to extract not only the tweet’s sentiment but, if given, the price prediction and the time frame of the prediction could be extracted and analyzed as well.

However this study aims to see if the correlation with present belief has an effect on current prices. So even though a user says Bitcoin's price will increase in a month another user reading this information may be inclined to buy Bitcoin now due to this information, the same can be said for the news sources analyzed as well, even though the news article may be making a prediction about the near future or the distant future our theory is that the sentiment of the article should affect current perception of the stock and therefore would have an impact on current price as well. Since if an investor reads a BBC article stating that Bitcoin price is predicted to surge next month he probably would be more inclined to buy Bitcoin today rather than when the price has started to increase. So the next question that is raised is how long a "tweet" or new source has an effect on somebody. Due to the fact that things posted online, stay online one could say that the influence window of tweets, or news sources are never ending, for this study however the window of influence is set to 1 day, 2 days and 3 days.

The accuracy rate of each user or news source is calculated in the same way. Even though a user might be making a suggestion for the distant future, for example Bitcoin price will increase 50% by the end of the year, we still only accept that prediction is correct if the price of Bitcoin increases for our influence window which is a period of a day, a period of 2 days and a period of 3 days. For this thesis we want to predict how accurate individual sentiment is rather then how accurate a single prediction is. The tweet the user sends, even though it is actually predicting the price of Bitcoin for the end of the year influences a user that reads this and they are more inclined to act at this given moment. The price increase of the prediction of 50% in this scenario is ignored.

The reason why the influence window was selected to be a day, 2 days and 3 days was that the main study of this study is to determine if individual accuracy rate, when taken into account when calculating the sentiment score of tweets and news feeds, affect the accuracy of a model that will predict the price of Bitcoin. However if the price of Bitcoin was gathered in the streaming data as well one could analyze tweets and news sources aggregate sentiment hourly instead of daily and do the same analysis. After the preprocessing a total sentiment score was derived for the news articles within a given time and tweets sent within a given time. The sentiment was then compared to the price of Bitcoin to see if there was any correlation with the price of Bitcoin and the sentiment derived from news articles and Twitter tweets within a given time period.

In the discussion section the process of how a person would gather the corpus of a news article is explained as well as how someone could use a streaming methodology to stream data via RSS feeds and analyse data in real-time. Due to the fact that it ended up costing a certain amount to extract and store news article information with the methodology utilized, the streaming process was stopped. Neither stream process nor article corpus that was only extracted from the RSS data was analysed, however this could be utilized in future work and therefore a general outline of the process was drawn out while working on this study.

Another reason why an influence window of at least a day was used was due to the method that previous data for Bitcoin price was gathered. Only closing prices of stock was gathered instead of their price at a given moment in a day. Due to the fact that most investors are interested in whether the price of a stock is going to increase or decrease at any given second the streaming methodology that will be analysed in future work may hold great importance. In a steaming methodology a system could be set up to scrape the current price of bitcoin from websites that provide the current price of Bitcoin. Infact one could utilize the streaming methodology utilized by Serkan Ayvaz and Mohammed O. Shiha (2018). Serkan Ayvaz and Mohammed O. Shiha (2018) created a Data Pipeline for Twitter data to analyse the price of Bitcoin. One could integrate the data scraped from the RSS feeds to the data streaming process of their methodology. If done one could analyse the price of Bitcoin with an influence window of an hour or less.

3.1 DATA COLLECTION

3.1.1 News article data

News articles that were used in the analysis were collected utilizing two different methodologies. One methodology used news sources RSS feeds to show the feasibility of streaming news article sources that are being published about Bitcoin. The other methodology used Google news and a web scraper to scrape previous news articles about Bitcoin.

3.1.1.1 RSS News Articles

As mentioned previously some news sources have RSS feeds, these are live feeds that give information about the articles being published on a given news sources website, they are updated as new news articles are published to the site. We utilized, BBC, CNBC, the Guardian, Reuters, Time and Nytimes RSS feeds to gather RSS data. RSS feeds are categorized, as in they are category specific, so Reuters has a different RSS feed for their world news section of their website and has a different RSS feed for the business part of their website. The BBC has a RSS feed for business, technology and world news. These categories are normally the categories one finds on the news sources website itself.

Python and AWS lambda was utilized to create a system where data was parsed and sent to s3 and dynamodb. A Lambda function was written in Python, this Python script goes through the assigned RSS script and extracts any news source article that contains the word Bitcoin, BTC, cryptocurrency in the description or the head of the news article.

Feedparser library, a python library was utilized to sparse the feeds context, Feedparser is a Python library that makes it easier to parse values from RSS feeds. If the news articles title contained one of these keywords related to bitcoin in the news articles on the RSS feed then the description, link, summary, pubdate, and title of the news article was extracted. For most RSS feeds we then needed to scrape the actual content of the news article itself. The Times.com RSS feed contains the content within the feed itself and therefore is the most user friendly of all the RSS feeds that were utilized.

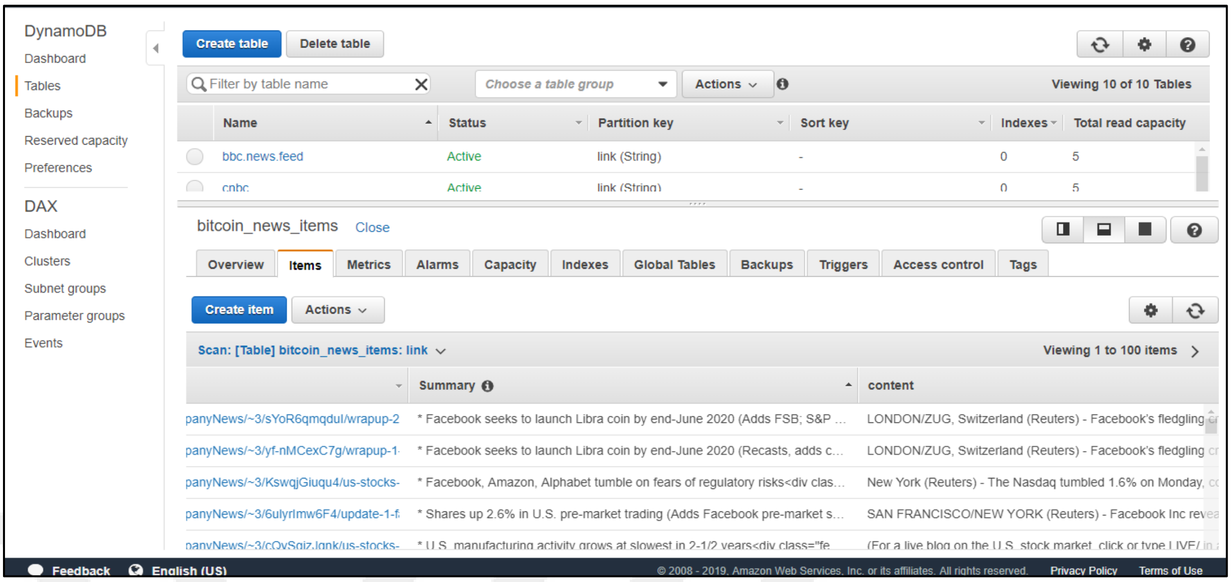
Once the link information was extracted from the RSS feed Python was utilized to scrape the content of the news article itself. Again this was done by AWS`s lambda function. Pythons Beautiful Soup and Request library was utilized to scrape the context of each article. Pythons Beautiful Soup library is a library that helps you parse html text. Python’s Request library is a library that allows you to make requests to web pages. You can scrape the html content of a webpage by utilizing both Python's request library and Pythons Beautiful Soup library.

Due to the fact that RSS feeds are specific to a certain source after learning the html of that specific source we can gather the content of each article, however some news sources like BBC have multiple news article formats. For example, some articles on the BBC are just videos where you have a title, a description and then the video itself, in this situation the description of the article could be saved as the context of the article as well.

Amazon web services, commonly abbreviated as AWS, “provides on-demand cloud computing platforms to individuals, companies and governments, on a metered pay-as-you-go basis”. It has many different products. AWS has storage capability with products like, dynamodb, a service that functions as a database and s3 a service to store items and folders in the cloud; to analytic functions like AWS lambda service which allows you to just run a function on the cloud without having to maintain a server. A lambda function can be written in many different programming languages including Java and Python, the function can be triggered in a variety of ways including on a specific time schedule and objects are put in an s3 bucket.

The lambda functions were created that when triggered went to certain news sources RSS feeds and then stored the data into s3 bucket and a dynamodb table. However, each time you put an object into s3 you are billed. There is a set of free tair that Amazon provides, however this was not sufficient so this methodology ended up costing a certain amount and was stopped.

Table 3.1: Aws dynamodb table



Name	Status	Partition key	Sort key	Indexes	Total read capacity
bbc.news.feed	Active	link (String)	-	0	5
cnbc	Active	link (String)	-	0	5

	Summary	content
panyNews/~3/sYoR6qmqdul/wrapup-2	* Facebook seeks to launch Libra coin by end-June 2020 (Adds FSB; S&P …	LONDON/ZUG, Switzerland (Reuters) - Facebook's fledgling cr
panyNews/~3/yf-nMCexC7g/wrapup-1-	* Facebook seeks to launch Libra coin by end-June 2020 (Recasts, adds c…	LONDON/ZUG, Switzerland (Reuters) - Facebook's fledgling cr
panyNews/~3/KswqjGiuqu4/us-stocks-	* Facebook, Amazon, Alphabet tumble on fears of regulatory risks<div clas…	New York (Reuters) - The Nasdaq tumbled 1.6% on Monday, cc
panyNews/~3/6ulyrlmw6F4/update-1-f	* Shares up 2.6% in U.S. pre-market trading (Adds Facebook pre-market s…	SAN FRANCISCO/NEW YORK (Reuters) - Facebook Inc revea
panyNews/~3/cQvSgiz.Jank/us-stocks-	* U.S. manufacturing activity grows at slowest in 2-1/2 years<div class="fe	(For a live blog on the U.S. stock market, click or type LIVE/ in a

An example of what a dynamodb table looks like is given in Table 3.1. The table contains the fields link, summary, content, publication date, and title. It contains the 210 articles published that were captured by the RSS scraper within the months the scraper was activated.

You can see in table3.1 an example of what the dynomotable looked like and what it contained. The columns the table contains are link, summary, content, pubdate, and title.

Link: the article's link scraped from RSS feed.

Summary: summary description on the RSS feed scraped from the RSS feed.

Content: the article's content was scraped using the link scraped from the RSS feed.

Pubdate: the date the article was published, including the minute and hour the article was published.

3.1.1.2 Google News Articles

Besides setting up the RSS scraper, to show that it is possible to collect a stream of articles from source specific news sources RSS feeds, a Google news scraper was utilized to gather past data from news sources to do the analysis itself. To do so

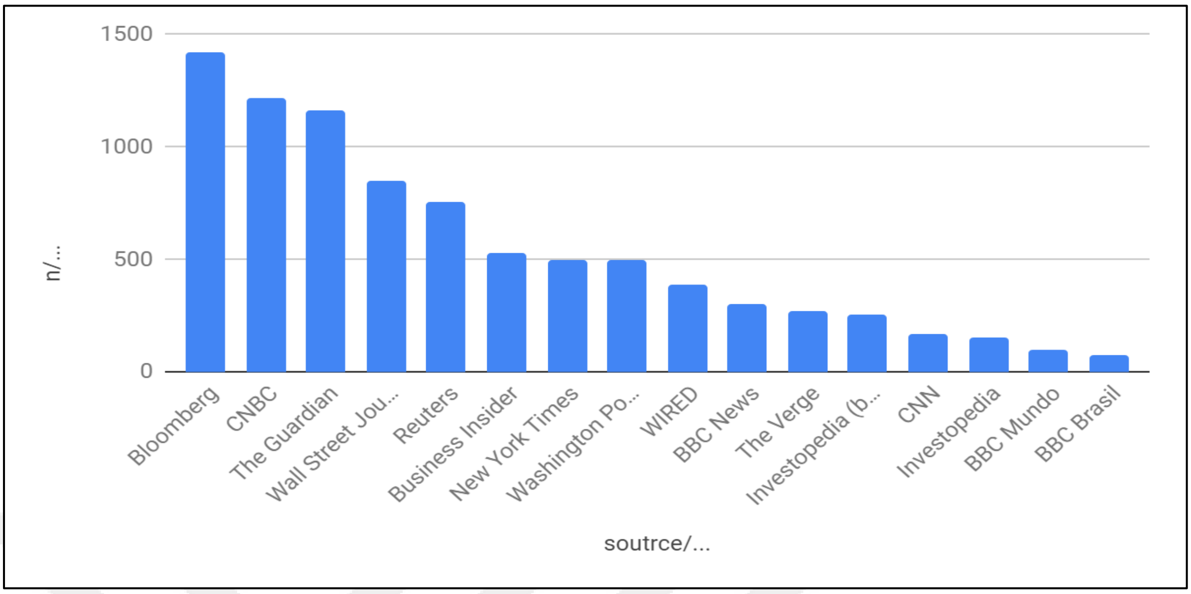
Python’s Selenium library was utilized, Python’s Selenium library allows you to open a web browser through Python, with Selenium utilizing the pages html you then can press on specific elements on the page. So a Python script was written that opened a Firefox web browser and then went to a Google page, went to the news subsection on Google and searched for the term Bitcoin for a given period of time of 2016-09 to 2018-06. Then elements of each article were scraped such as publication date, news article link, news title and the short description that Google provides. After that, using Selenium, the next button on Google was clicked and it repeated the process for each page.

Unlike the data collection with the RSS feeds, where each RSS was source specific, Google news results give us a list of different sources. Since different web sources have different html format we couldn't automatically scrape the articles content as done with our RSS data. RSS data on the other hand is source specific as in for Reuters articles. The Reuters RSS feed was utilized and therefore all the articles that the RSS feed scraper went through were Reuters articles so therefore using the HTML of the page each articles content was scraped. Therefore the corpus of each article was not able to be scraped while using this method since the source wasn't specific.

Another drawback of the Google data is that it doesn’t contain the publication hour of articles, whereas with the RSS feeds the publication date of a news article states the hour and minute. the news articles was published it is given to you in minutes and hours by most RSS feeds, in Google news search results even though the articles publishing date is specified it doesn't specify the hour or minute a specific news article was published. This is a massive drawback since that limits us to only predict daily differences in price since we cannot obtain the specific time a news article was published.

AWith the Google scraper a total of 9100 articles were collected that mentioned the word Bitcoin for the given time period of 2016-09 to 2018-06. Most articles were collected from Bloomberg, CNBC, The Guardian, The Wall Street Journal as demonstrated in Figure 3.2. All these sources have RSS feeds, however Bloomberg's RSS feed wasn't scraped.

Figure 3.2: Amount of Articles from Each Source



3.1.2 Twitter Data

Twitter provides users that sign up for their developers account access to their API. To sign up for a Twitter developer account you need to have a Twitter account and register to get a developers account. After getting a developers account you can get access to their API. You can connect to the API using the consumer key and consumer secret that Twitter API provides and using Python to connect to the stream of data. A Twitter streamer was created however due to the fact that the news streamer had become too costly was never utilized to scrape data.

Past Twitter data was obtained using the Java code written by Jefferson Henrique [44]. A description on how to use the code to obtain tweets is located at https://github.com/Jefferson-Henrique/GetOldTweets-java

Though the code seemed to work best when scraping monthly tweets rather then yearly, tweets from 09-2016 to 06-2019 where scraped. A total of 404,016 tweets were scraped that contained the word Bitcoin. The data contained the dimensions id, mentions permalink, retweets, text username, hashtags, date, favorites and status. Table 3.2 shows a snapshot of the data frame created, the dimensions of the data frame are:

Id: the id of the specific tweet itself

Permalink: the link to the tweet

Username: not given, however using the permalink can be deduced.

For example https://twitter.com/avsa/status/95885179767802 is a Twitter permalink, the user that posted this tweet is avsa. The username is contained in the tweet permalink itself and therefore t can be extracted from the url. This was done in Python using Python's regex library.

Hashtags: hashtags in a tweet

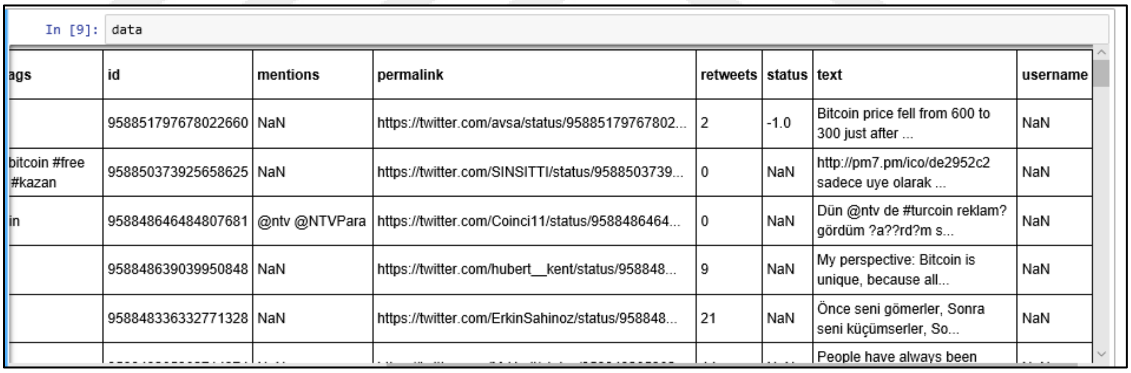
Date: date that tweet was sent. The date contains the hour and minute information as well.

Favorites: number of favorites

Retweets: amount of times tweet got retweeted

Status: dimension was created to label tweets as positive, negative and neutral

Table 3.2: Dataframe of Tweet Data

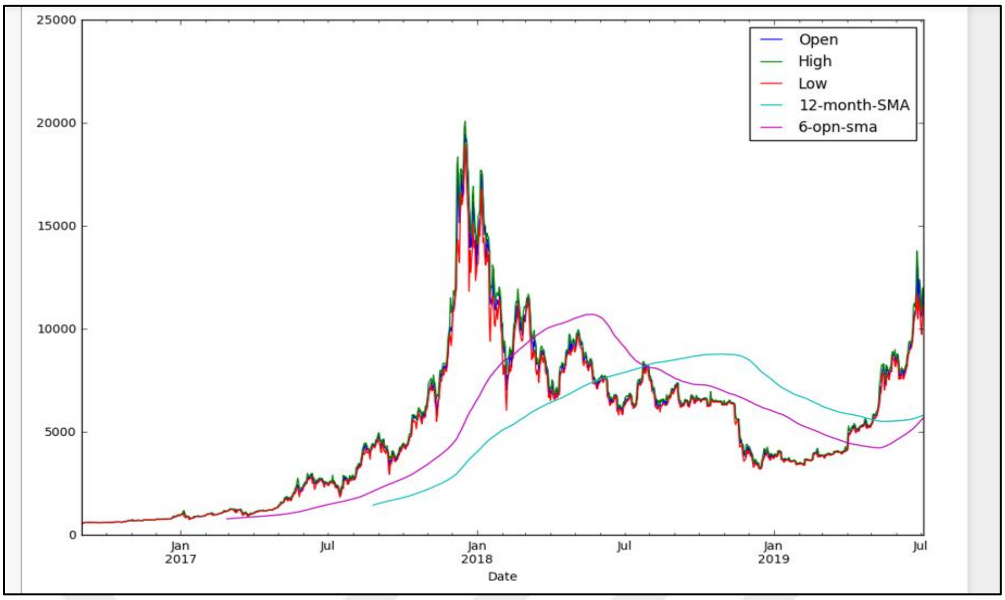


```
In [9]: data
```

ags	id	mentions	permalink	retweets	status	text	username
	958851797678022660	NaN	https://twitter.com/avsa/status/95885179767802...	2	-1.0	Bitcoin price fell from 600 to 300 just after ...	NaN
bitcoin #free #kazan	958850373925658625	NaN	https://twitter.com/SINSITTI/status/9588503739...	0	NaN	http://pm7.pm/ico/de2952c2 sadece uye olarak ...	NaN
in	958848646484807681	@ntv @NTVPara	https://twitter.com/Coinci11/status/9588486464...	0	NaN	Dün @ntv de #turcoin reklam? gördüm ?a??rd?m s...	NaN
	958848639039950848	NaN	https://twitter.com/hubert__kent/status/958848...	9	NaN	My perspective: Bitcoin is unique, because all...	NaN
	958848336332771328	NaN	https://twitter.com/ErkinSahinoz/status/958848...	21	NaN	Önce seni gömerler, Sonra seni küçümserler, So...	NaN
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	People have always been	[illegible]

3.1.3 Price of Bitcoin

Figure 3.3: Price of bitcoin

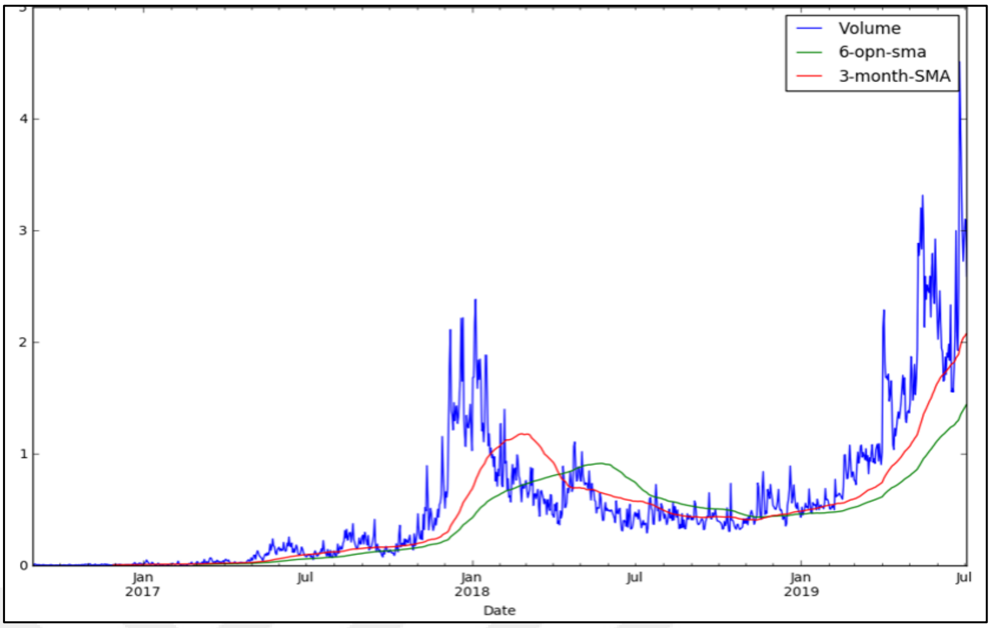


The historical price of Bitcoin was taken from https://coinmarketcap.com/. Coinmarketcap provide the historical closing, opening, low, high, and market cap. The prices from the months 2016 -09 to 2019-06 where taken from the website. Figure 3.3 shows the opening price highs and lows with 12 month and 6 month moving average of the opening price of Bitcoin.

The average opening price range for Bitcoin in this time period was 5161.24635 dollars. The highest price ever reached in this time frame was 20,089.00 dollars on Dec 17, 2017. The lowest it ever reached was 570.81 dollars on Sep 02, 2016.

Within our time interval of 1037 days, 496 dates closing price of the next day was 0-5% higher than the closing price of the end of the day. 345 dates closing price of the next day was 0-5% lower than the closing price of the end of the day. 101 dates closing price of the next was 5% higher than the closing price of the end of the day, 95 dates closing price of the next day was more than 5% lower than the closing price of the end of the day. The rise in volume is given in Figure 3.4 with its 6 month and 3 month moving averages as well.

Figure3.4: Volume of Bitcoin



3.2 PREPROCESSING

Due to the fact the Sentiment Analyzer Textblob and the Analyzer Vader didn't perform as well as expected a sentiment classifier was created utilizing machine learning approach using Naive Bayes algorithm. The machine learning approach, as mentioned in the literature review, is when a given piece of text has labeled data and the labeled data is used to teach a classifier patterns by utilizing different algorithms like Naive Bayes. Therefore data needed to be prepared to be analyzed. Therefore preprocessing is split up into the parts labeling and preprocessing for both Twitter data and news article data.

3.2.1 Sentiment Analysis Pre-Processing

While this study was originally focused on a lexicon based approach to conduct Sentiment Analysis due to findings a machine learning method was utilized instead. This didn't change the necessary preprocessing since both methodologies can use the same preprocessing as it is a benefit and helps get more accurate results.

Due to the fact a machine learning method was taken a set of data needed to be labeled. A group of English and Turkish random tweets were selected and hand annotated either positive, negative or neutral. For tweets they were labeled and it was annotated 1 for positive -1 for negative and 0 for neutral and for news articles they were only two labels 1 for positive sentiment and -1 for negative sentiment.

A tweet was annotated positive when it talked about the future price increase of Bitcoin, mentioned how you should buy Bitcoin, mentioned how much money you would have made or anything that could be seen as a positive sign for Bitcoin. A news article was positive, if it mentioned about new companies that accept Bitcoin as payment or the increase of Bitcoin's price. The tweets were classified negative when tweets mentioned the decrease of Bitcoin, or if the tweet stated that bitcoin was a bubble.

Tweets where annotated, 1 for positive, -1 for negative and 0 for neutral. The tweet was annotated 1 when the tweet talked about Bitcoins price increase or the tweet mentioned companies that are willing to accept Bitcoin as a way of payment, or when people tweeted about the future booms of Bitcoin. It was annotated negative when the tweet talked about how Bitcoin was collapsing in price, when the tweet talked about regulations that may affect Bitcoin, when the tweet referred to Bitcoin as a bubble or when Bitcoin was referred to as a scam. It was marked 0 when only the price of Bitcoin at that moment was stated, or when even though the tweet contained the word Bitcoin it wasn't about Bitcoin. For example, "I met a boy who was 14 who talked to me about Bitcoin," the sentence contains the word Bitcoin however it is not about the current price of Bitcoin. Although one could argue that it was positive because if the boy knew about Bitcoin it shows how well known a currency it is.

For news article data a set of data was marked to be either positive or negative. The articles were marked positive if they were talking about the price increase of Bitcoin and the global usage of Bitcoin. It was marked negative if the article was about government regulations or scams that have to do with Bitcoin. The news articles were labeled according to the brief description instead of the headlines themselves, this is due to the fact some headlines from some sources are frequently written as a question, for example, "Will the price of Bitcoin rise?" One couldn't label this to be positive or negative, however in the description of the article it goes into a bit more detail and one can therefore say if it is positive or negative.

3.2.1.1 News articles data

For the preprocessing of news data each letter in the title or context was converted to lower case in the dataframe. A stemmer was used to stem each word in the news headline and stop words, common words in the English language, were removed. All of these processes will be talked about in more detail in section 3.2.1.2.

Then numbers followed by the dollar sign were removed, this was done since a price in one point in time does not represent a value today. For example, “Bitcoin skyrocketed to $10000.” During the time this news article was printed this would be marked as positive. If the number wasn't removed during preprocessing then the machine learning algorithm would associate $10000 with a positive sentiment. However if Bitcoin’s price is around $50000 and an article is published stating, “Bitcoin might see the $10000 level by the end of the day”, due to the fact the number 10000 is associated as positive this tweet might be wrongly labeled as positive. However it would be accepted as that while using the machine learning approach. This issue wouldn’t have to be dealt with if the lexicon method was accurate. Results of the lexicon sentiment classifier will be given in section 3.2.1.2.

3.2.1.2 Tweets data

For preprocessing the tweets all of the capital letters in a tweet were set to lowercase. Our lexicon method or machine learning method lexicons use terms as vectors. Therefore if, for example, "good" is in our positive lexicon however the sentence contains the term "Good" due to the fact the “g” is capitalized it is seen as a different term entirely. Unlike a human a machine cannot tell that Good is the same as good. Therefore each word in each sentence of each tweet was converted to lower case.

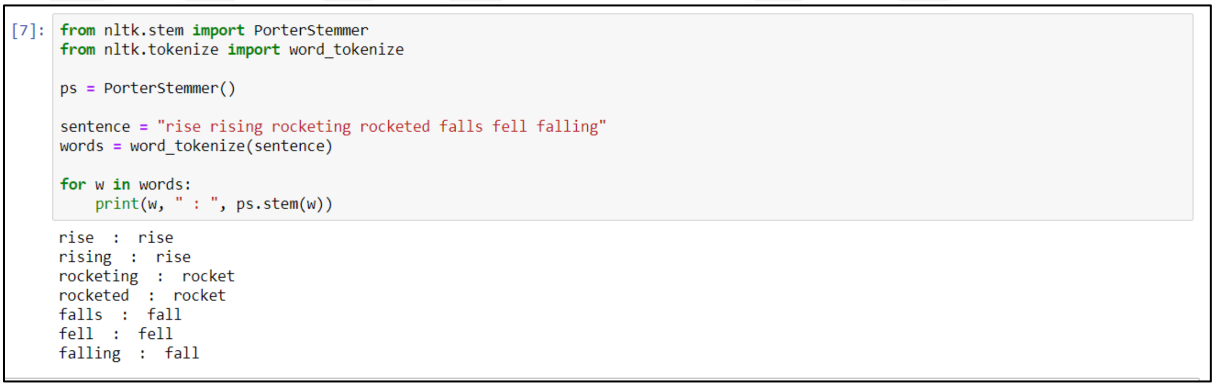
Stopwords were removed. Stop words here refers to words that do not have any impact on sentiment. Words like “this, or, and”, that are common words that are used frequently that have no significance on sentiment. Removing words like, and, the common words that don't really give the sentence that much meaning where removed. Pythons Nltk stop word dictionary was downloaded and utilized for the list of stop words. This was done for news articles as well.

Hashtag words were removed from each sentence and replaced with the word themselves. So for example if a tweet contained #bitcoin, it was replaced with bitcoin.

If the tweet contained a url the url was stripped and the text url was put in replacement for the word url.

User names were generated from the tweet links in the twitter data. As mentioned the twitter data did not contain ids of users only ids of tweets and the url to that tweet, however the url contains the user information. So the url was used to extract the user name during preprocessing

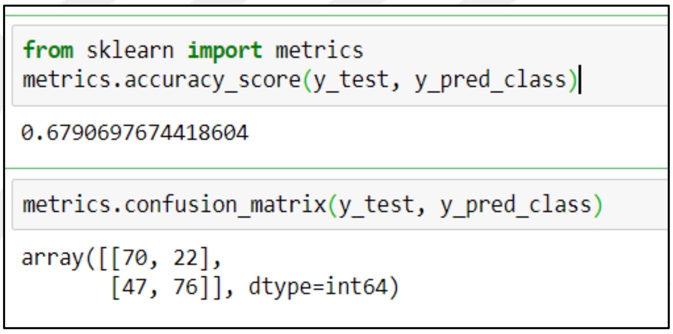
Figure 3.5: PorterStemmer output



```
[7]: from nltk.stem import PorterStemmer
     from nltk.tokenize import word_tokenize

     ps = PorterStemmer()

     sentence = "rise rising rocketing rocketed falls fell falling"
     words = word_tokenize(sentence)

     for w in words:
         print(w, " : ", ps.stem(w))

     rise  :  rise
     rising  :  rise
     rocketing  :  rocket
     rocketed  :  rocket
     falls  :  fall
     fell  :  fell
     falling  :  fall
```

After that each word in each tweet was put through a stemmer. “Stemming is a technique used to remove affixes from a word replacing them with their roots reducing different forms of a word such as nouns, verbs, adjectives etc. to a common base form.”(PBarnaghi, Ghaffari, & Breslin, (2016)). Porter's stemmer was utilized, which can be found in the Python nltk library. Stemming is the process of stripping each word into their base form. So for example in Figure 3.5, the words, “rise, rising, rocketing, rocketed, falls, fell, falling,” when put through the same stemmer was utilized to bring back the results of “rise, rise, rocket, rocket, fall, fell, fall”. The same stemmer was used while stemming the newspaper articles. To understand why this helps for analysis picture a negative lexicon that contains the word “fall”, however if the news article or tweet contained the word “falling” it wasn't in our negative lexicon and it wouldn't be given a negative score. By stemming both your documents and your lexicon you are increasing the odds of finding the same word that is within your lexicon, otherwise due to a couple of different letters a word won't be classified. It is the same case for machine learning methodology if you train your classifier on stemmed words and then feed it documents in which the words are stemmed you increase the chances that the words in the predicted documents will have been in the training data.

Figure 3.6: Sentiment Classifier Results Without Stemmer



```
from sklearn import metrics
metrics.accuracy_score(y_test, y_pred_class)
```

```
0.6790697674418604
```

```
metrics.confusion_matrix(y_test, y_pred_class)
```

```
array([[70, 22],
       [47, 76]], dtype=int64)
```

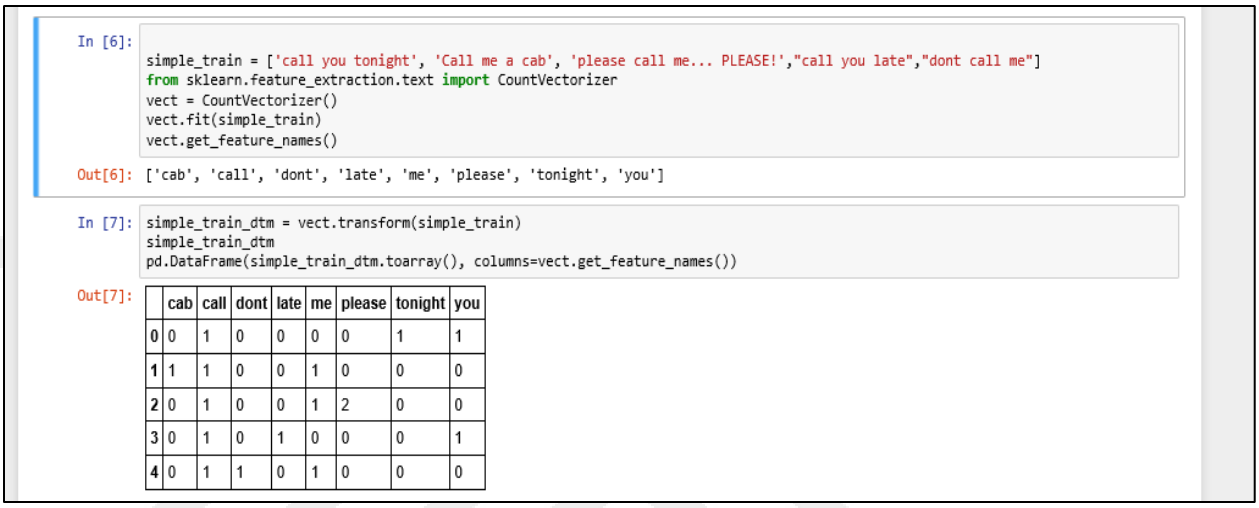
Figure 3.6 is a printout of the news articles sentiment classifier results when stemming wasn't accounted for, 67.9%, whereas for the classifier that was given stem words the accuracy was 69.3%

The preprocessing step was set up by a Python’s script that utilized the libraries Pandas Re and NLTK. It took about 1 and a half day, approximately 36 hours, to preprocess the Twitter data alone using a simple house computer. 400000/3600 so data preprocessing for 3 tweets occurred every second.

Then the test data that contained the labeled data for English and Turkish tweets was separated from the tweets. This was done to train the sentiment classifier with just the prelabeled tweets.

A document term matrix vector was created for words in the training data and the original tweet data. A document term matrix is a matrix that contains all tokens as features and documents as rows and represent which token is located in which document.

Figure 3.7: Document term matrix

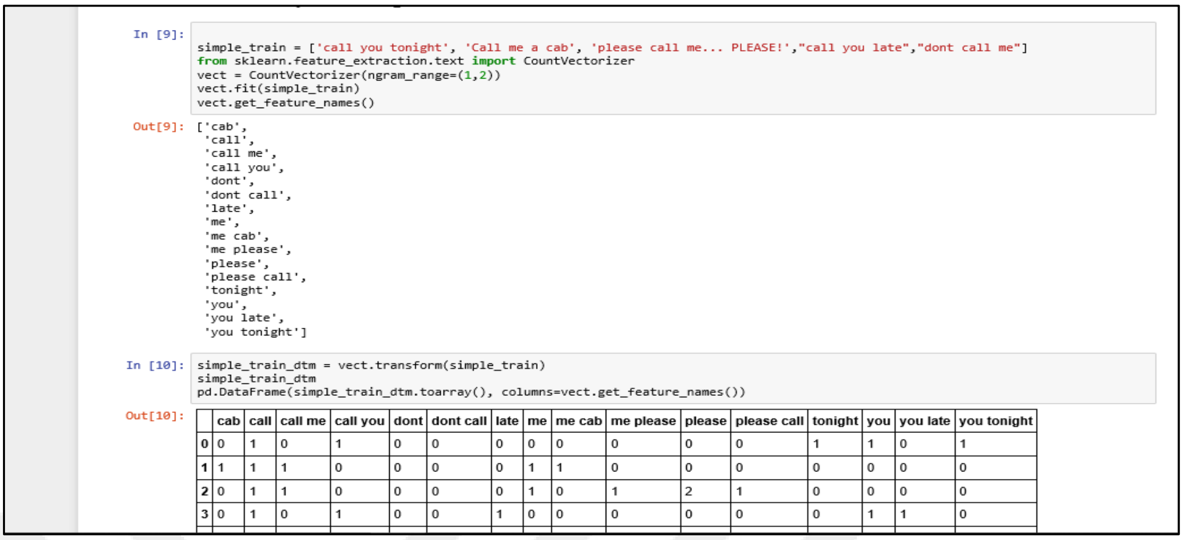


```
In [6]:
simple_train = ['call you tonight', 'Call me a cab', 'please call me... PLEASE!',"call you late","dont call me"]
from sklearn.feature_extraction.text import CountVectorizer
vect = CountVectorizer()
vect.fit(simple_train)
vect.get_feature_names()

Out[6]: ['cab', 'call', 'dont', 'late', 'me', 'please', 'tonight', 'you']

In [7]: simple_train_dtm = vect.transform(simple_train)
simple_train_dtm
pd.DataFrame(simple_train_dtm.toarray(), columns=vect.get_feature_names())

Out[7]:
```

	cab	call	dont	late	me	please	tonight	you
0	0	1	0	0	0	0	1	1
1	1	1	0	0	1	0	0	0
2	0	1	0	0	1	2	0	0
3	0	1	0	1	0	0	0	1
4	0	1	1	0	1	0	0	0

Figure 3.7 is an example of a document term matrix, for the document stated in the cell above as simple train. As you can see, each word that all of the documents contained, are in the columns of the document term matrix and the rows indicate the document number. Figure 3.8 is an example of a document term matrix that was created by utilizing bigrams and unigrams. As previously mentioned unigrams take each word as an individual token, however bigrams take two words as a token into account, the word “please” was used 2 times in the indexed 2 document and wasn't used in any other sentences for example. You can see the features of the document term matrix in Figure 3.8.

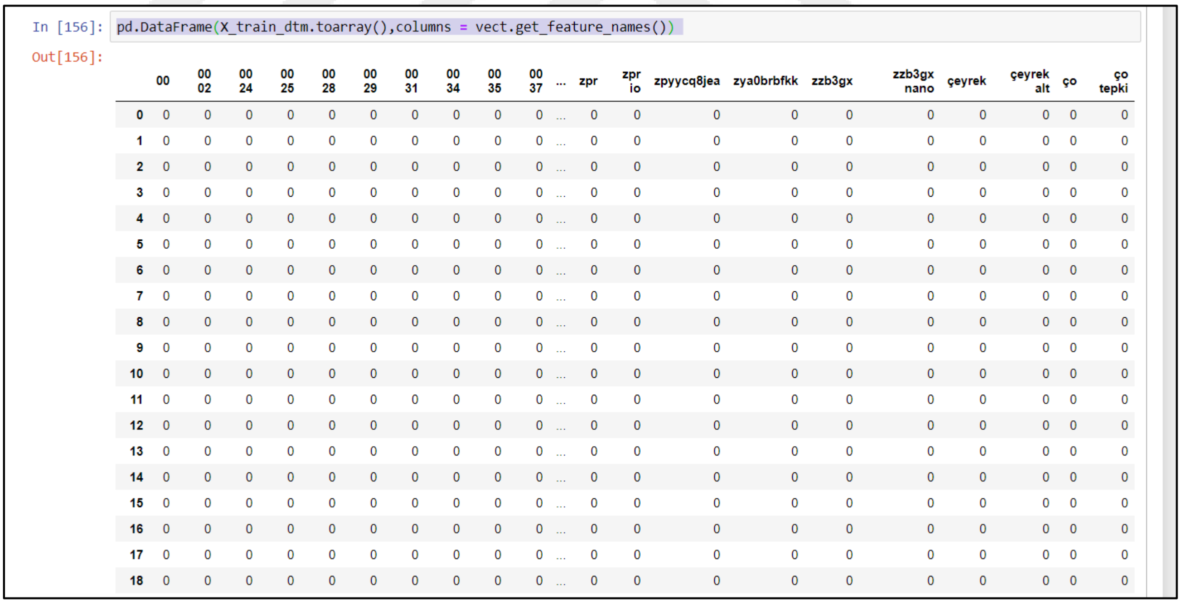
Figure 3.8: Bigram and Unigram Document Term Matrix



```
In [9]:  simple_train = ['call you tonight', 'Call me a cab', 'please call me... PLEASE!',"call you late","dont call me"]
         from sklearn.feature_extraction.text import CountVectorizer
         vect = CountVectorizer(ngram_range=(1,2))
         vect.fit(simple_train)
         vect.get_feature_names()

Out[9]:  ['cab',
          'call',
          'call me',
          'call you',
          'dont',
          'dont call',
          'late',
          'me',
          'me cab',
          'me please',
          'please',
          'please call',
          'tonight',
          'you',
          'you late',
          'you tonight']

In [10]: simple_train_dtm = vect.transform(simple_train)
         simple_train_dtm
         pd.DataFrame(simple_train_dtm.toarray(), columns=vect.get_feature_names())

Out[10]:
```

	cab	call	call me	call you	dont	dont call	late	me	me cab	me please	please	please call	tonight	you	you late	you tonight
0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	1
1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
2	0	1	1	0	0	0	0	1	0	1	2	1	0	0	0	0
3	0	1	0	1	0	0	1	0	0	0	0	0	0	1	1	0

Figure 3.9: Document term matrix of Turkish tweets



```
In [156]: pd.DataFrame(X_train_dtm.toarray(),columns = vect.get_feature_names())

Out[156]:
```

	00	00 02	00 24	00 25	00 28	00 29	00 31	00 34	00 35	00 37	...	zpr	zpr io	zpyycq8jea	zya0brbfkk	zzb3gx	zzb3gx nano	çeyrek	çeyrek alt	ço	ço tepki
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Figure 3.9 is the document term matrix created for Turkish tweets. The rows represent each tweet and the columns represent words contained in all tweets. .

3.3 SENTIMENT ANALYSIS RESULTS

3.3.1 News Sentiment Analysis Results

Both the lexicon method and the machine learning methodology were used while analyzing news articles as mentioned in the data preprocessing section. Text Blobs and Vader's sentiment lexicons were used, they are Python libraries that you can installed. Bing Lius (2004). positive and negative lexicon was utilized as well to see if the lexicon would be able to classify news articles correctly.

Out of the labeled data 447 were positive and 410 were negative, besides preprocessing, nothing needed to be done to feed the labeled data to Textblob and Vader.

Figure 3.10: Lexicon Method Accuracy

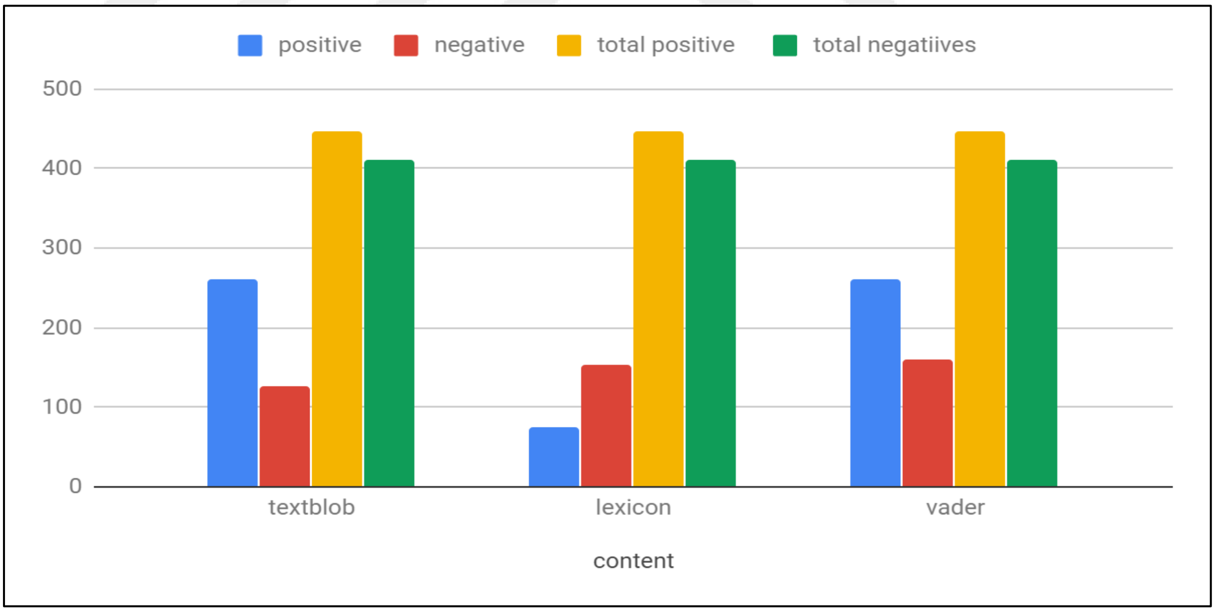
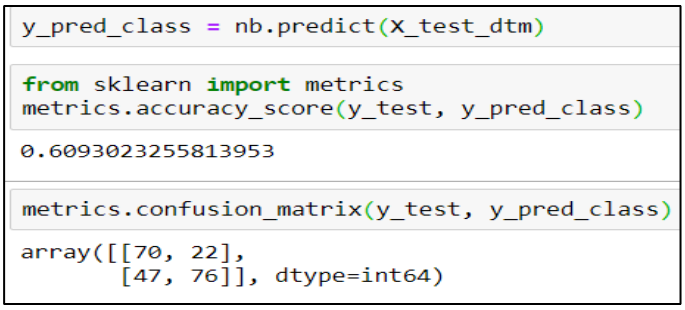


Figure 3.10 shows how many correct positives and negatives each classifier got correct when labeling the labeled data set. Vader sentiment classifier managed to classify the labeled data set 49% accurately, labeling 260 positives correctly positive and 160 negative correctly negative. Textblob managed to classify each news article with a 45%

accuracy and the lexicon that was downloaded managed to achieve a 26% accuracy. Since if the sentiment classifier labeled everything positive it would of gotten a 52% accuracy rate none of these methodologies was seen as an acceptable rate. However Vader was the most accurate of the three. It got a lot of labeled data as either positive or negative. As mentioned in the literature review, sentiment classifiers are better when subject specific or industry terms are taken into account. For example, “rise” may not be valued as positive in the Vader lexicon, however when someone is talking about Bitcoin the word “rise” should have a positive value. Or a news article that mentions government regulations, something negative for Bitcoin, without having keywords in your lexicon the accuracy rate may be at these levels. Figure 3.10 shows the accuracy rate when the header of the news article was taken into account rather than the content, Textblob got a 23% accuracy while Vader managed to predict 33% accurately.

Due to the low accuracy rate of the lexicon method, machine learning methodology was utilized to properly label the news article data. To do so, the data that was preprocessed and labeled data was split into 642 training data and 215 testing data for a Naive Bayes algorithm. A document term matrix of the testing and the training set were created using n-grams of 1 and 3. Initially the title of the news articles were used to classify the news article, this resulted in a 60.9% accuracy rate as observed in Figure3.11. 76 articles were correctly classified as positive, while 47 positive articles were classified as negative, whereas 22 articles that were negative were classified as positive and 70 negatives were classified correctly. We had 122 negative tweets and 92 positive tweets in our testing data, all had been classified as negative the accuracy rate would be 56.7%.

Figure 3.11: Sentiment classifier results



```
y_pred_class = nb.predict(X_test_dtm)
```

```
from sklearn import metrics
metrics.accuracy_score(y_test, y_pred_class)
```

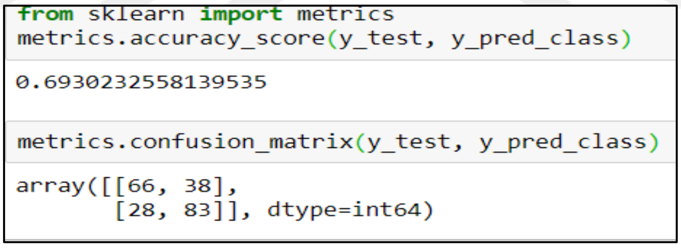
```
0.6093023255813953
```

```
metrics.confusion_matrix(y_test, y_pred_class)
```

```
array([[70, 22],
       [47, 76]], dtype=int64)
```

When the content of the news articles were analysed by converting them in to unigram and the classifier managed to classify the articles with an accuracy rate of 69.3% as shown in Figure3.12. All tests were conducted with a random split of the training data, this test had 104 negative and 111 positive values, if all were marked positive this would give us an accuracy of 51%. A few of tests were conducted using different numbers of n-grams and by using stop words or removing them with a specific test and training set instead of randomizing the training and testing to compare results.

Figure 3.12: Sentiment Classifier Results



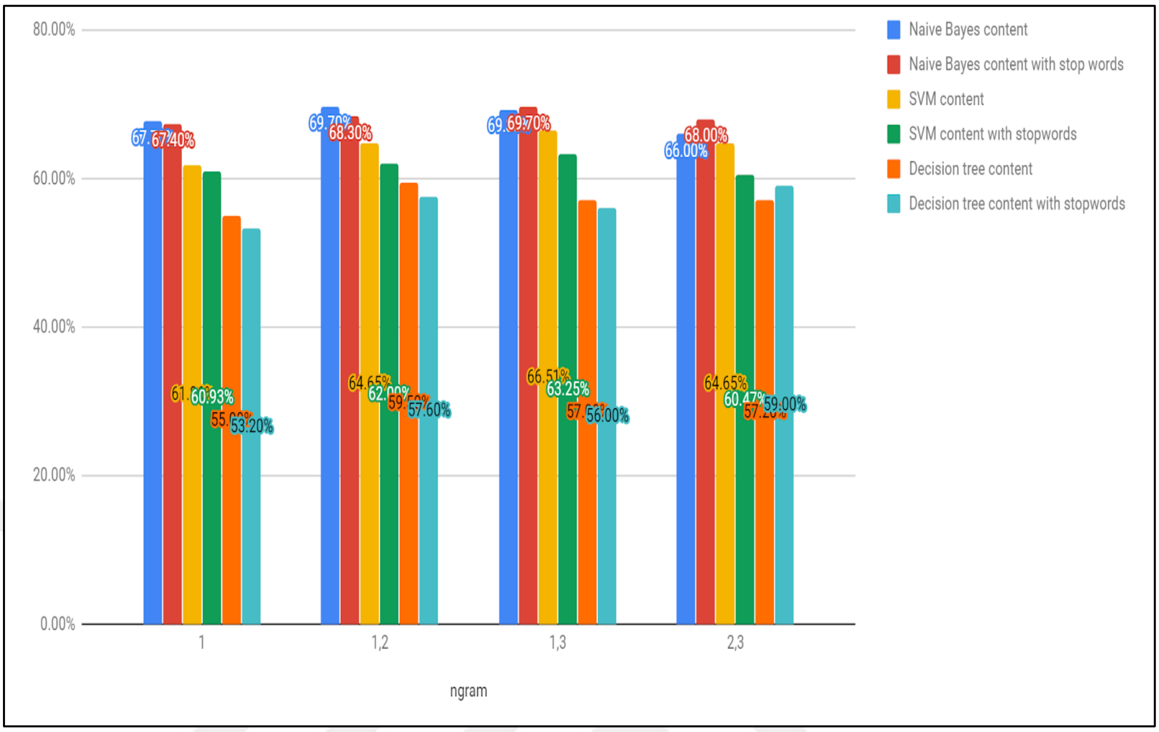
```
from sklearn import metrics
metrics.accuracy_score(y_test, y_pred_class)
```

```
0.6930232558139535
```

```
metrics.confusion_matrix(y_test, y_pred_class)
```

```
array([[66, 38],
       [28, 83]], dtype=int64)
```

Figure 3.13: Sentiment Classifier Results



Multiple different machine learning algorithms were utilized while including stopwords and removing stop words in the document term matrix of the training set and testing set while using different number of ngrams. Figure 3.13 shows the accuracy of each method that was utilized. As you can see from Figure 3.13 Naïve Bayes performed the best out of the three machine learning algorithms used. Decision tree algorithm performed the worst out of all the algorithms that were utilized.

Bo Pang and Lillian Lee (2002) showed that Naïve Bayes or SVM performed well while trying to create a sentiment classifier. Our results confirm this, however Naive Bayes outperformed SVM. The classifier that used 1-grams and 3-grams without erasing stop words as well as the classifier that used 1-gram and 2-grams with erasing stop words performed the best with an accuracy of 69.7%. The ROC curve of the classifier is given below in graph Figure 3.14, as you can see the ROC curve shows that the classifier is better than a random guess . A ROC curve is a two-dimensional chart where the y axis value is Recall and the x axis value is equal to Specificity. Specificity is calculated by dividing all true negatives with the sum of all false positives and all true negatives (Beckmann 2017).

Figure 3.14: NB Sentiment ROC curve

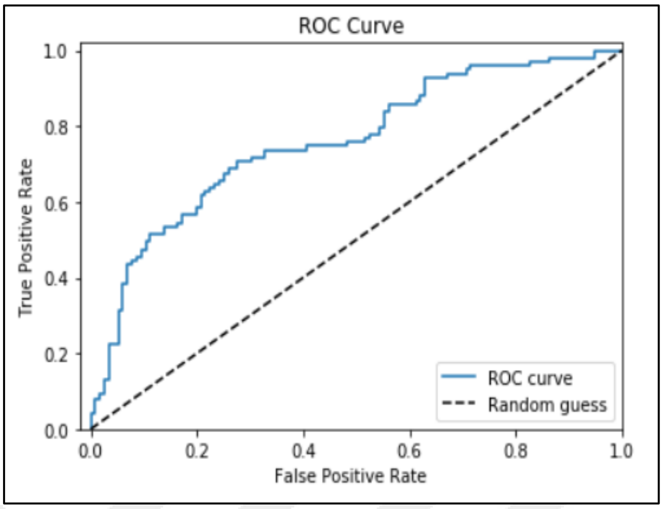
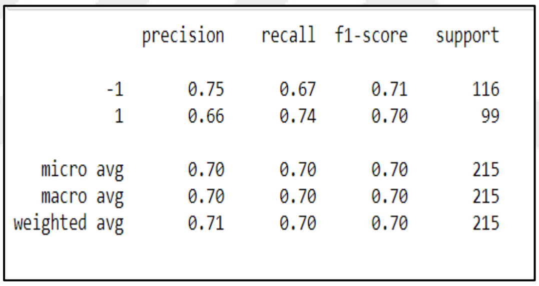


Figure 3.15: NB Sentiment Classifier Results



```
              precision    recall  f1-score   support

          -1       0.75      0.67      0.71       116
           1       0.66      0.74      0.70        99

   micro avg       0.70      0.70      0.70       215
   macro avg       0.70      0.70      0.70       215
weighted avg       0.71      0.70      0.70       215
```

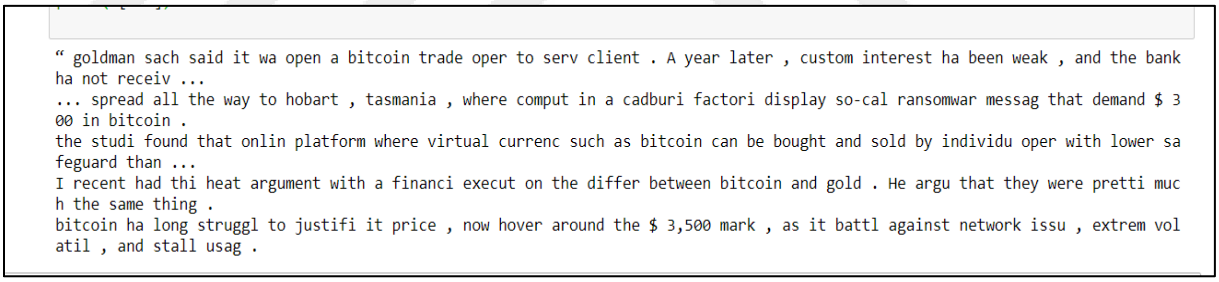
The precision and recall results for both negative and positive is given in Figure 3.15 “The Precision is a measure of exactitude, and it denotes the percent of hits related to all positive objects.” (Beckmann 2017). In other words precision is the number of true positives for a label divided by the true positives and the false positives. So the classifier managed to correctly label 75 % of negatives, as you can see by the results.

“The Recall, also denominated as Sensitivity, is a completeness measure, and it denotes the percent of positive objects identified by the classifier” (Beckmann 2017). In other words precision is the number of true positives for a label divided by the true positives and the false negatives. So the classifier was 67% accurate for all the negative labels it classified.

To label the sentiment of the unlabeled data a Naive Bayes classifier was used by creating document term matrices that contained unigrams and bigrams while removing stop words from articles.

A print out of the negative labeled articles that were marked as positive is given in Figure 3.16. The first one, for example, “Goldman Sachs said it will open a Bitcoin trade operation to serve clients. A year later, customer interest has been weak and the bank has not received…”, this content begins positive about Bitcoin, in that Goldman Sachs made a decision about opening trade to Bitcoin, however it goes on to say something negative and is the reason why it was labeled as negative, though it's easy to see why it may have been classified as positive. The classifier classified 5233 news articles as negative and 7240 news articles as positive out of the total articles collected for the time period.

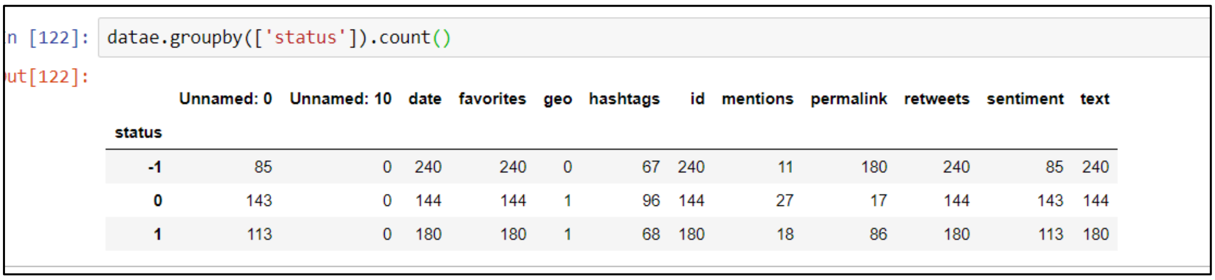
Figure 3.16: False Positive



```
“ goldman sach said it wa open a bitcoin trade oper to serv client . A year later , custom interest ha been weak , and the bank
ha not receiv ...
... spread all the way to hobart , tasmania , where comput in a cadburi factori display so-cal ransomwar messag that demand $ 3
00 in bitcoin .
the studi found that onlin platform where virtual currenc such as bitcoin can be bought and sold by individu oper with lower sa
feguard than ...
I recent had thi heat argument with a financi execut on the differ between bitcoin and gold . He argu that they were pretti muc
h the same thing .
bitcoin ha long struggl to justifi it price , now hover around the $ 3,500 mark , as it battl against network issu , extrem vol
atil , and stall usag .
```

3.3.2 Twitter Sentiment Analysis

Figure 3.17: Tweet Training Data

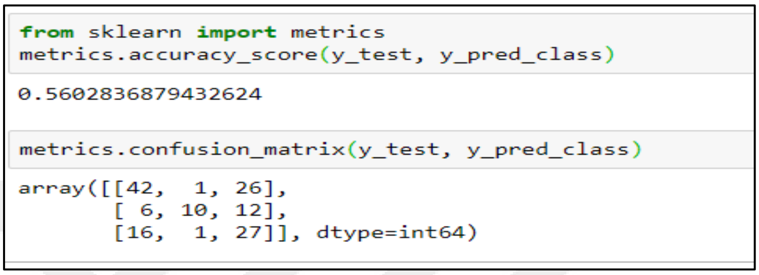


```
n [122]: datae.groupby(['status']).count()
ut[122]:
```

	Unnamed: 0	Unnamed: 10	date	favorites	geo	hashtags	id	mentions	permalink	retweets	sentiment	text
status												
-1	85	0	240	240	0	67	240	11	180	240	85	240
0	143	0	144	144	1	96	144	27	17	144	143	144
1	113	0	180	180	1	68	180	18	86	180	113	180

The English tweets training data contained 240 negative tweets, 144 neutral tweets and 18- positive tweets as shown in Figure 3.17. Both unigrams and a combination of unigrams and bigram document term matrices were created and utilizing Naive Bayes machine learning algorithm two tests were conducted. The labeled training data itself was split into training data and test data.

Figure 3.18: Twitter sentiment classifier results



```
from sklearn import metrics
metrics.accuracy_score(y_test, y_pred_class)
```

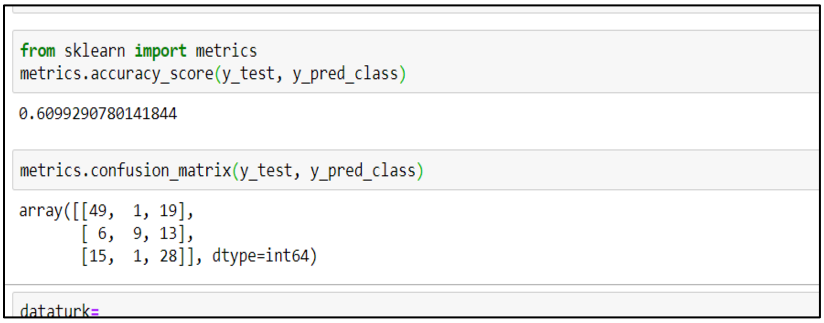
```
0.5602836879432624
```

```
metrics.confusion_matrix(y_test, y_pred_class)
```

```
array([[42,  1, 26],
       [ 6, 10, 12],
       [16,  1, 27]], dtype=int64)
```

While only taking unigrams into account the Naive Bayes algorithm managed to accurately label 56 percent of English tweets as shown in Figure 3.18. Classifying 42 correctly negative, 6 that were neutral as negative and 16 that were positive as negative. 1 positive and negative tweets were misclassified as neutral and 10 neutral were classified correctly. 27 were true positives while 26 negatives were marked positive and 12 neutrals were marked positive.

Figure 3.19: Twitter sentiment classifier results



```
from sklearn import metrics
metrics.accuracy_score(y_test, y_pred_class)
```

```
0.6099290780141844
```

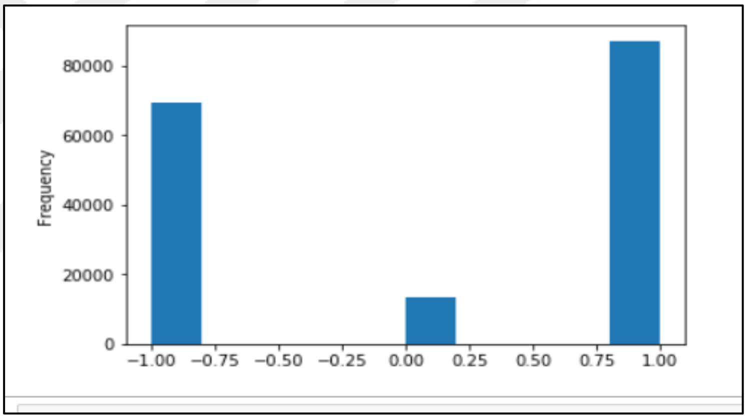
```
metrics.confusion_matrix(y_test, y_pred_class)
```

```
array([[49,  1, 19],
       [ 6,  9, 13],
       [15,  1, 28]], dtype=int64)
```

```
dataturk=
```

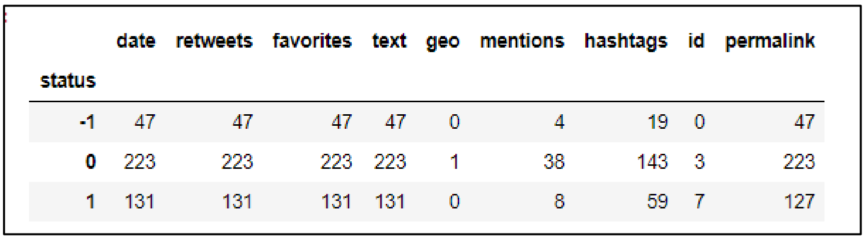
While using bigrams and unigrams the Naive Bayes algorithm managed to get an accuracy rate of .609 once stop words were taken into account as shown in Figure 3.19. As you can see from the accuracy metric, we predicted 49 negative, miss predicted 6 neutral tweets as negative and miss predicted 15 positive tweets as negative. 1 positive and negative tweet miss predicted as neutral while getting 9 correct neutral labels. We had 29 true positives. However there were a lot of false positives, 19 negative tweets were classified as positive and 13 neutral tweets were classified as positive. The training data that was utilized had 180 positive tweets, -1. .The test data contained 141 tweets, 69 of which were negative. If the classifier classified all the tweets negative it would have a 48% accuracy. Due to this the labeled data was split into a document term matrix of unigrams and bigrams and stop words were taken into account when the Naive Bayes machine learning algorithm classified each tweet in the un-labeled English tweet data.

Figure 3.20: English Tweets Labels



As shown in Figure 3.20 the English classifier, classified 88425 tweets as negative, 11863 tweets as neutral and 69614 tweets as positive. The maximum amount of tweets a person sent was 4167, the mean of the number of tweets sent was 6.6 and the median was 1, so most people that have tweeted about Bitcoin have only tweeted about Bitcoin once in the last 3 years. This may affect our results while analysing the accuracy of each user since most users have only made one prediction.

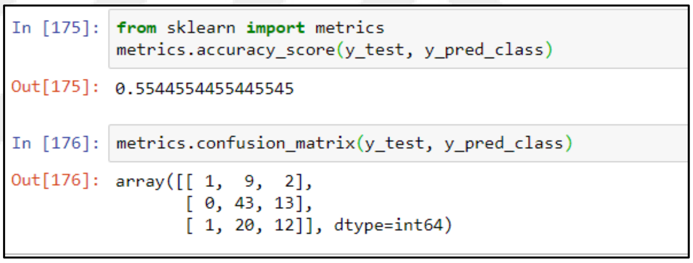
Figure 3.21: Turkish tweet training data



status	date	retweets	favorites	text	geo	mentions	hashtags	id	permalink
-1	47	47	47	47	0	4	19	0	47
0	223	223	223	223	1	38	143	3	223
1	131	131	131	131	0	8	59	7	127

As shown in Figure 3.21, the Turkish tweet labeled training set contained 47 negative, 131 positive and 223 neutral tweets. Both unigrams and a combination of unigrams and bigram document term matrices were created and by utilizing the Naive Bayes machine learning algorithm two tests were conducted. The labeled training data itself was split into training data and test data.

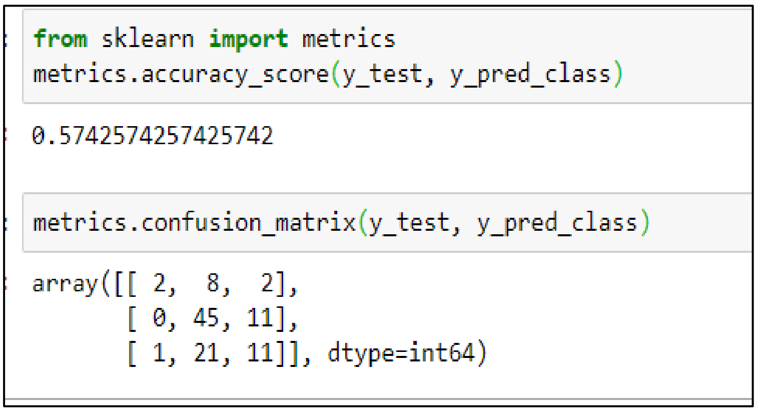
Figure 3.22: Turkish tweet sentiment classifier results



```
In [175]:  from sklearn import metrics
           metrics.accuracy_score(y_test, y_pred_class)

Out[175]:  0.5544554455445545

In [176]:  metrics.confusion_matrix(y_test, y_pred_class)

Out[176]:  array([[ 1,  9,  2],
                  [ 0, 43, 13],
                  [ 1, 20, 12]], dtype=int64)
```

As shown in Figure 3.22, the unigram results for the Turkish data had a 55 percent accuracy. It correctly labeled one negative as negative and incorrectly labeled 1 positive as negative. It correctly labeled 43 tweets as neutral while it incorrectly labeled 20 positive and 9 negative tweets as neutral and it labeled 2 negative as positive, 13 neutral as positive and 12 positives as positive. Looking at this data you can see that the main predictions made by the classifier were neutral or positive. This may be due to the fact that there weren’t that many negative Turkish tweets labeled in the training set.

Figure 3.23: Turkish Tweet Sentiment Classifier Results



```
from sklearn import metrics
metrics.accuracy_score(y_test, y_pred_class)
```

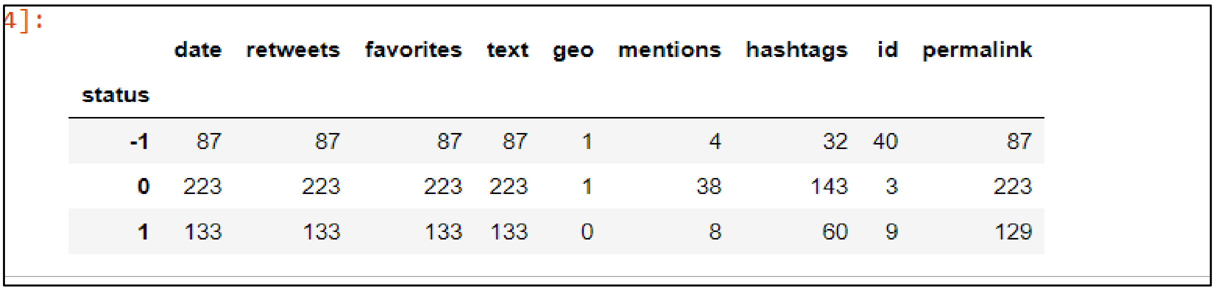
```
0.5742574257425742
```

```
metrics.confusion_matrix(y_test, y_pred_class)
```

```
array([[ 2,  8,  2],
       [ 0, 45, 11],
       [ 1, 21, 11]], dtype=int64)
```

While the classifier was given unigrams and bigrams to analyse, the analysis was conducted with bigrams and the accuracy increased to 57 percent as shown in Figure 3.23. As you can observe by the confusion matrix, 2 were labeled correctly negative and 1 positive was labeled negative. 45 neutral were correctly labeled neutral, 21 positive were labeled neutral, 8 negative were labeled as neutral, 2 negatives were labeled as positive, 11 neutral were labeled as positive and 11 positive were correctly labeled as positive. Due to the fact that there weren't many negative tweets when randomly classifying tweets and more neutral the classifier may be miss labeling our data. It has 45 neutral true positives however 29 false tweets that were either positive or negative that was labeled as neutral as well. To solve this issue due to the fact that there were a very little amount of negative tweets labeled, tweets were once again hand sorted through and hand labeled. This time 87 negative tweets, 223 neutral tweets and 129 negative tweets were labeled as shown in Figure 3.24. The test was run again and the accuracy rate increased to 59%.

Figure 3.24: Turkish Tweet Training Data

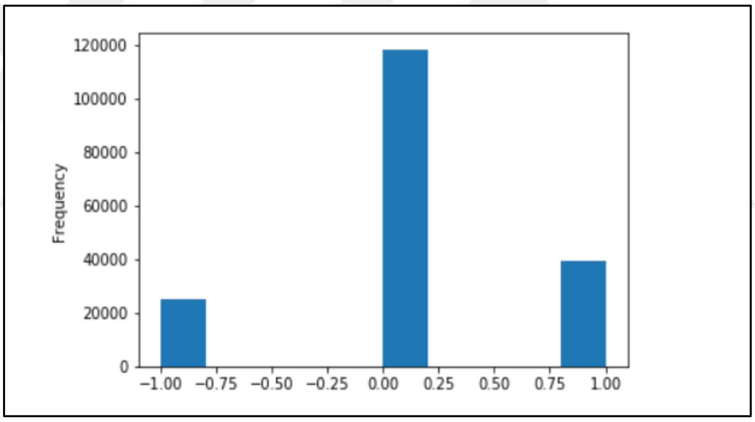


4]:

	date	retweets	favorites	text	geo	mentions	hashtags	id	permalink
status									
-1	87	87	87	87	1	4	32	40	87
0	223	223	223	223	1	38	143	3	223
1	133	133	133	133	0	8	60	9	129

The classifier was then trained on the entire training set and then used to predict the unlabeled data set. As you can see from Figure 3.25 the classier classified 24941 Turkish tweets as negative, 1185452 as neutral and 39483 tweets as positive.

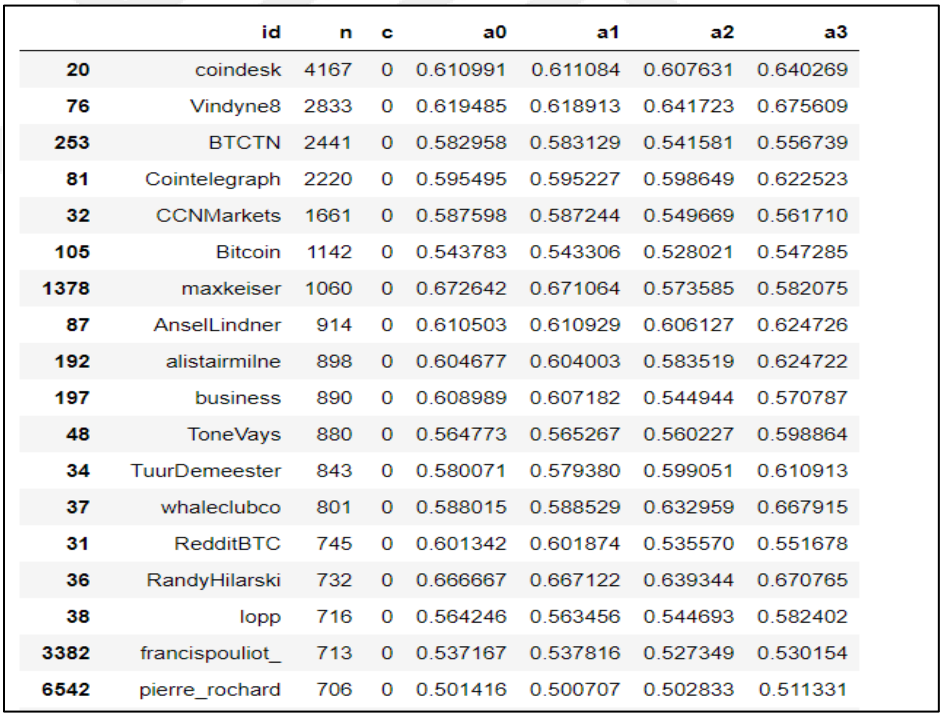
Figure 3.25: Turkish Tweets Labels



3.4 ACCURACY DATAFRAME AND PREPROCESSING OF DATA

After each tweet was classified as to be either positive, negative or neutral, the accuracy metrics were calculated. If a user had positive sentiment about Bitcoin and the price went up during the day, or the next 3 days the users accuracy rate would increase by one. So n being the total times a user tweeted tweets containing Bitcoin if the users sentiment was correct then the users new accuracy would be a=(N+1)/(N+1). If the user or articles sentiment was incorrect then his new accuracy would be equal to a=(N)/(N+1), n being the total amount of guesses a user has made. A snapshot of the Dataframe created is given in Figure 3.26.

Figure 3.26: Accuracy Dataframe



	id	n	c	a0	a1	a2	a3
20	coindesk	4167	0	0.610991	0.611084	0.607631	0.640269
76	Vindyne8	2833	0	0.619485	0.618913	0.641723	0.675609
253	BTCTN	2441	0	0.582958	0.583129	0.541581	0.556739
81	Cointelegraph	2220	0	0.595495	0.595227	0.598649	0.622523
32	CCNMarkets	1661	0	0.587598	0.587244	0.549669	0.561710
105	Bitcoin	1142	0	0.543783	0.543306	0.528021	0.547285
1378	maxkeiser	1060	0	0.672642	0.671064	0.573585	0.582075
87	AnselLindner	914	0	0.610503	0.610929	0.606127	0.624726
192	alistairmilne	898	0	0.604677	0.604003	0.583519	0.624722
197	business	890	0	0.608989	0.607182	0.544944	0.570787
48	ToneVays	880	0	0.564773	0.565267	0.560227	0.598864
34	TuurDemeester	843	0	0.580071	0.579380	0.599051	0.610913
37	whaleclubco	801	0	0.588015	0.588529	0.632959	0.667915
31	RedditBTC	745	0	0.601342	0.601874	0.535570	0.551678
36	RandyHilarski	732	0	0.666667	0.667122	0.639344	0.670765
38	lopp	716	0	0.564246	0.563456	0.544693	0.582402
3382	francispouliot_	713	0	0.537167	0.537816	0.527349	0.530154
6542	pierre_rochard	706	0	0.501416	0.500707	0.502833	0.511331

Dataframe dimensions listed below:

Id: the source of news or twitter id

n=number of guess made

a0=accuracy if the price went up during that day

a1=accuracy if the price went up the next day
a2=accuracy if the went up two days after the tweet or articles publication
a3=accuracy if price went up days after the tweet or articles publication

The data was ordered from last date to the most present date and by using iterating through each row was fed into the accuracy matrix row by row using the price information taken online. Before saving the new accuracy in the accuracy dataframe for the current user, the current accuracy was saved in the news article or the tweet data frame, so each tweet or news item had the current accuracy of the user at that moment of time. Then each tweet user's accuracy at the time the tweet was sent was multiplied with the tweets sentiment score.

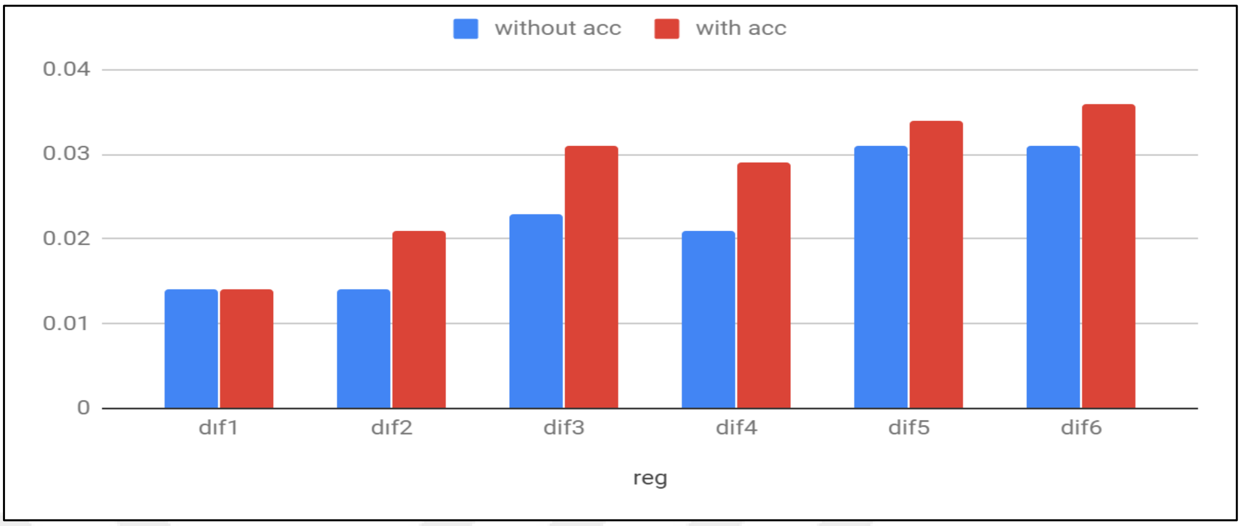
Using the tweets with their accuracy of the time that the tweet was sent and sentiment labelled, a data frame was created that indicated the sum positive tweets, negative tweets, positive news stories, negative news stories, sentiment score within that day, date, opening price closing price, sum of positive tweets with the accuracy taken into account, the sum of negative tweets with accuracy taken into account, sum of positive news stories with the accuracy taken into account, sentiment score that takes accuracy into account.

Then the percentage price change of 1 day 2 day and 3 days were calculated and grouped into 5 different categories, 0 for a neutral group, 1 for slight increase, 2 for a high increase -1, for a slight decrease and -2 for a high decrease .

4. FINDINGS

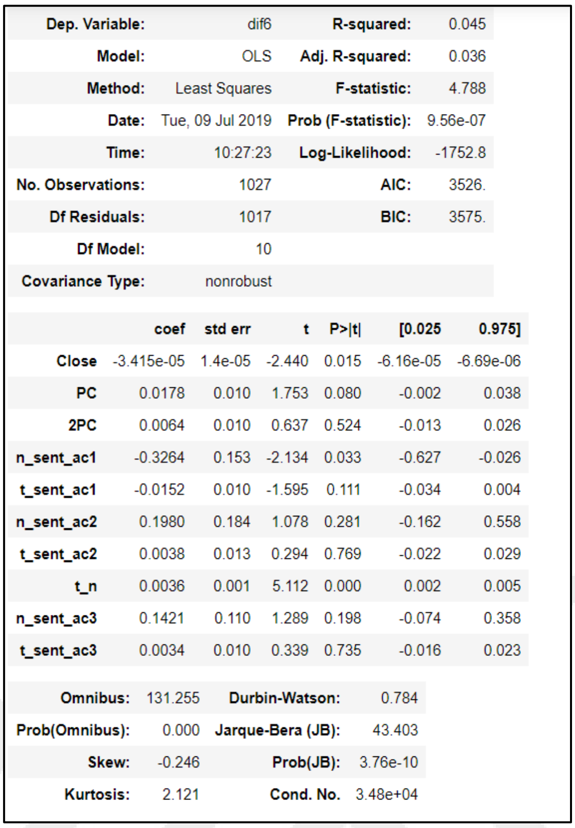
After the dataframe was created a few regressions were run to determine if the price of Bitcoin could be predicted using Twitter sentiment and news sentiment of a given day as variables. The adjusted r-square of the results is shown in Figure 4.1. dif 1 is the price change from today to tomorrow with the price groups of 5% change or above being labeled as 2, 5-1% price change labeled as 1, 1%change to -1% change labeled as 0 , -1 to -5 labeled as -1 and negative 5% and lower labeled as -2. dif2 4is price change of tomorrow however the neutral group is between 2% and negative 2% . dif 3 is the price change for 2 days in to the future with a neutral level of 1 and -1.

Figure 4.1: Regression Results



Dif 4 is the price change for 2 days with the neutral level of %2 and -2%. Dif 5 price change of 3 days with a neutral level of 1% and -1%. Dif 6 is the price change of 3 days with a neutral level of 2 and -2. Even though the regressions adjusted R-square results were not higher than .04 which would indicate that this is not a good model to predict the price of Bitcoin, the adjusted r square once accuracy has been taken into has account was slightly higher. Figure 4.1 shows the adjusted r square results where only the total sentiment, number of neutral sentiment, closing price and the previous day's price change was only taken in to account for both the model with accuracy included and the model that did not.

Figure 4.2: Regression Results with Accuracy

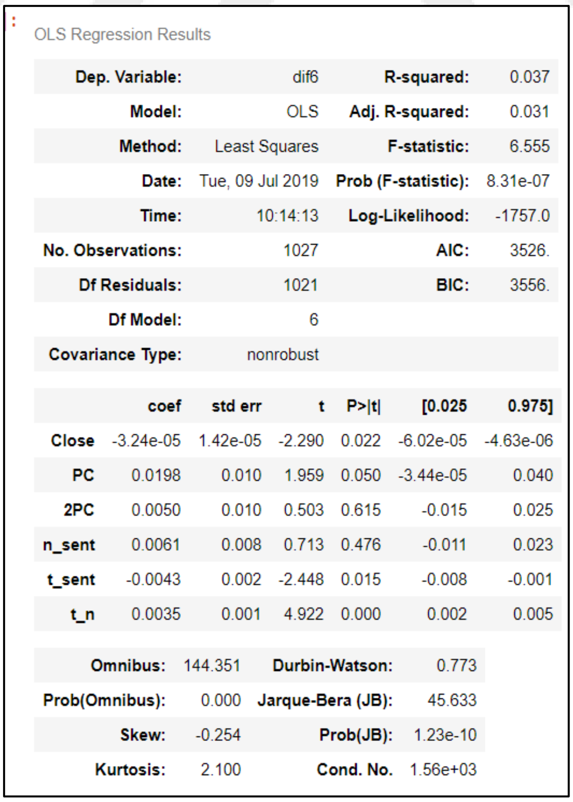


Dep. Variable:	dif6	**R-squared:**	0.045
Model:	OLS	**Adj. R-squared:**	0.036
Method:	Least Squares	**F-statistic:**	4.788
Date:	Tue, 09 Jul 2019	**Prob (F-statistic):**	9.56e-07
Time:	10:27:23	**Log-Likelihood:**	-1752.8
No. Observations:	1027	**AIC:**	3526.
Df Residuals:	1017	**BIC:**	3575.
Df Model:	10		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Close	-3.415e-05	1.4e-05	-2.440	0.015	-6.16e-05	-6.69e-06
PC	0.0178	0.010	1.753	0.080	-0.002	0.038
2PC	0.0064	0.010	0.637	0.524	-0.013	0.026
n_sent_ac1	-0.3264	0.153	-2.134	0.033	-0.627	-0.026
t_sent_ac1	-0.0152	0.010	-1.595	0.111	-0.034	0.004
n_sent_ac2	0.1980	0.184	1.078	0.281	-0.162	0.558
t_sent_ac2	0.0038	0.013	0.294	0.769	-0.022	0.029
t_n	0.0036	0.001	5.112	0.000	0.002	0.005
n_sent_ac3	0.1421	0.110	1.289	0.198	-0.074	0.358
t_sent_ac3	0.0034	0.010	0.339	0.735	-0.016	0.023

Omnibus:	131.255	**Durbin-Watson:**	0.784
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	43.403
Skew:	-0.246	**Prob(JB):**	3.76e-10
Kurtosis:	2.121	**Cond. No.**	3.48e+04

Figure 4.3: Regression Results



OLS Regression Results

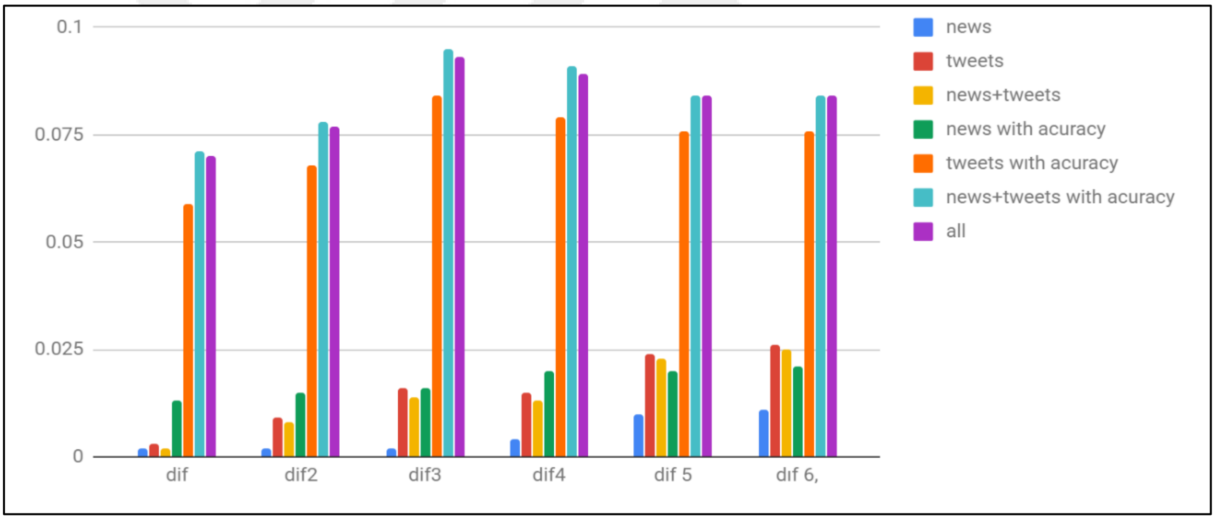
Dep. Variable:	dif6	**R-squared:**	0.037
Model:	OLS	**Adj. R-squared:**	0.031
Method:	Least Squares	**F-statistic:**	6.555
Date:	Tue, 09 Jul 2019	**Prob (F-statistic):**	8.31e-07
Time:	10:14:13	**Log-Likelihood:**	-1757.0
No. Observations:	1027	**AIC:**	3526.
Df Residuals:	1021	**BIC:**	3556.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Close	-3.24e-05	1.42e-05	-2.290	0.022	-6.02e-05	-4.63e-06
PC	0.0198	0.010	1.959	0.050	-3.44e-05	0.040
2PC	0.0050	0.010	0.503	0.615	-0.015	0.025
n_sent	0.0061	0.008	0.713	0.476	-0.011	0.023
t_sent	-0.0043	0.002	-2.448	0.015	-0.008	-0.001
t_n	0.0035	0.001	4.922	0.000	0.002	0.005

Omnibus:	144.351	**Durbin-Watson:**	0.773
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	45.633
Skew:	-0.254	**Prob(JB):**	1.23e-10
Kurtosis:	2.100	**Cond. No.**	1.56e+03

As you can see from the results of the regression in Figure 4.2 and Figure 4.3, when sentiment of a given day is taken into account with closing price of the day and change in price for one day ago and two days ago the resulting r squares are .045 and .031. For the test where accuracy was taken into account news sentiment with 1 day accuracy rates taken into account, the amount of neutral tweets, and the closing price had the smallest p-values. There P values were .033 0.00 and 0.015. In the regression where accuracy rates were not taken into account neutral tweets and Twitter sentiment as well as price range were found to be significant in the test. When compared, the results that take the accuracy rate of each user into account is a better predictor of the price of Bitcoin, however both adjusted R square weren`t higher than .04 percent. Therefore neither is a good model overall to predict the price movement of Bitcoin.

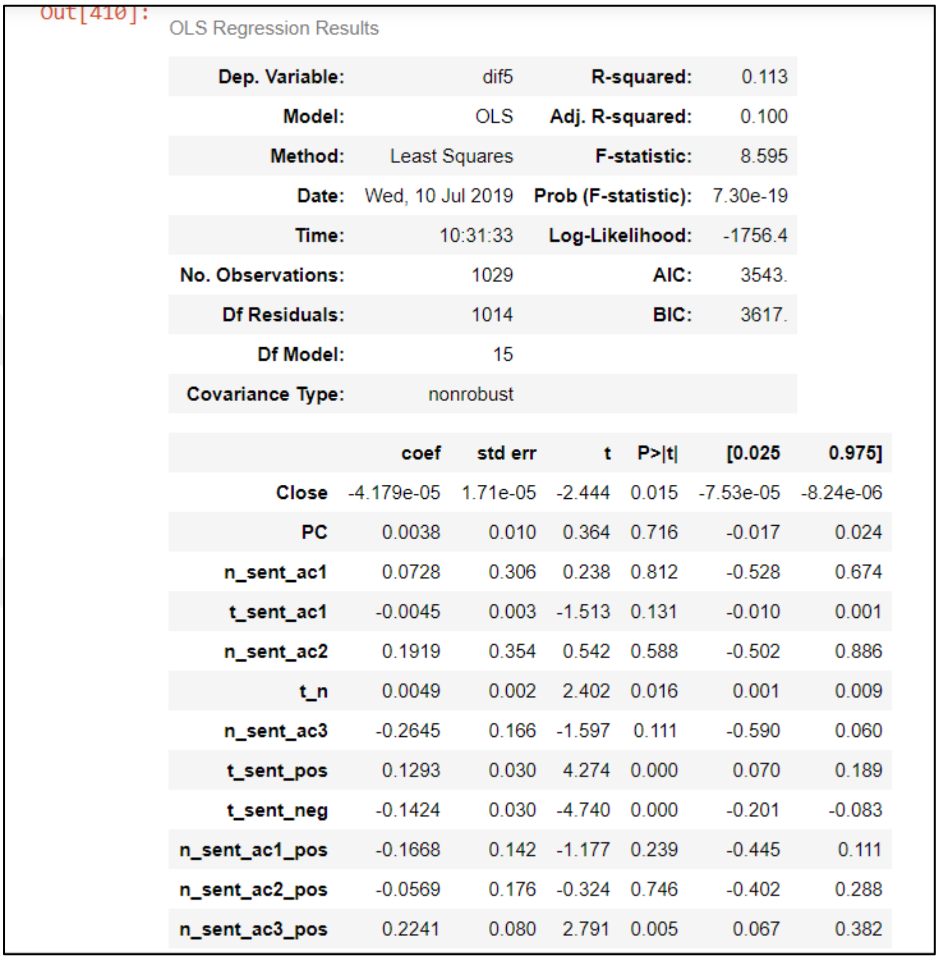
Figure 4.4: Regression Adjusted R-square Results



Multiple regressions where conducted, to see how accuracy rate affected the adjusted r square for tweets and news and how it affected the both of them together. As the results indicate in figure 4.4 the best model that predicted the price of bitcoin was news and tweets with accuracy taken in to account for dif3, which is the price change for two days into the future with the neutral class equals -1% and 1%.

The highest adjusted r squared reached from all the tests that were conducted was .100 as shown in Figure 4.5. Price change of the day before, news sentiment with all accuracy taken into account and twitter sentiment with all accuracy measurements taken into account were used as independent variables. The price change for 3 days into the future with the neutral class between 1% and -1 % was used as the dependent class. The p score for news and Twitter sentiment score with accuracy taken into account are lower than 0.002 .

Figure 4.5: Regression Results



Out[410]:

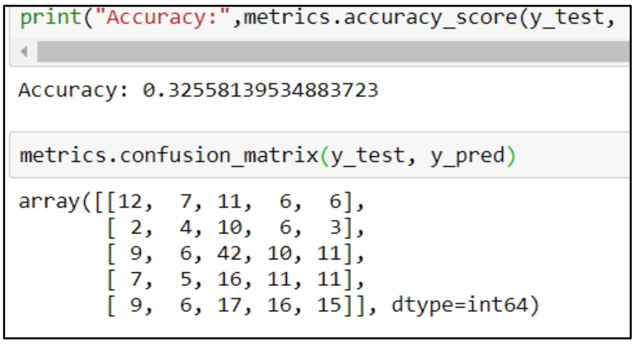
OLS Regression Results

Dep. Variable:	dif5	**R-squared:**	0.113
Model:	OLS	**Adj. R-squared:**	0.100
Method:	Least Squares	**F-statistic:**	8.595
Date:	Wed, 10 Jul 2019	**Prob (F-statistic):**	7.30e-19
Time:	10:31:33	**Log-Likelihood:**	-1756.4
No. Observations:	1029	**AIC:**	3543.
Df Residuals:	1014	**BIC:**	3617.
Df Model:	15		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Close	-4.179e-05	1.71e-05	-2.444	0.015	-7.53e-05	-8.24e-06
PC	0.0038	0.010	0.364	0.716	-0.017	0.024
n_sent_ac1	0.0728	0.306	0.238	0.812	-0.528	0.674
t_sent_ac1	-0.0045	0.003	-1.513	0.131	-0.010	0.001
n_sent_ac2	0.1919	0.354	0.542	0.588	-0.502	0.886
t_n	0.0049	0.002	2.402	0.016	0.001	0.009
n_sent_ac3	-0.2645	0.166	-1.597	0.111	-0.590	0.060
t_sent_pos	0.1293	0.030	4.274	0.000	0.070	0.189
t_sent_neg	-0.1424	0.030	-4.740	0.000	-0.201	-0.083
n_sent_ac1_pos	-0.1668	0.142	-1.177	0.239	-0.445	0.111
n_sent_ac2_pos	-0.0569	0.176	-0.324	0.746	-0.402	0.288
n_sent_ac3_pos	0.2241	0.080	2.791	0.005	0.067	0.382

The variables that got the highest r squared for the regression results were used with a decision tree algorithm to predict the price change of bitcoin, the model result matrix is in Figure 4.6.

Figure 4.6: Decision tree results



```
print("Accuracy:",metrics.accuracy_score(y_test,
```

```
Accuracy: 0.32558139534883723
```

```
metrics.confusion_matrix(y_test, y_pred)
```

```
array([[12,  7, 11,  6,  6],
       [ 2,  4, 10,  6,  3],
       [ 9,  6, 42, 10, 11],
       [ 7,  5, 16, 11, 11],
       [ 9,  6, 17, 16, 15]], dtype=int64)
```

The decision tree results are given in Figure 4.6. Even though the model`s accuracy rate was 32% it managed to predict on 12 occasions that Bitcoin`s price was going to go lower than 5 percent, however on 9 occasions where it predicted the price would go lower than 5% within 3 days the price actually increased by more than 5%, the decision tree managed to predict the increase of 5% 15 times and was only wrong about the price direction 9 times, 6 when the price went lower than 5% and 3 where the price drop was between 5% and 2% .

Figure 4.7: Open Price of Bitcoin and Sentiment

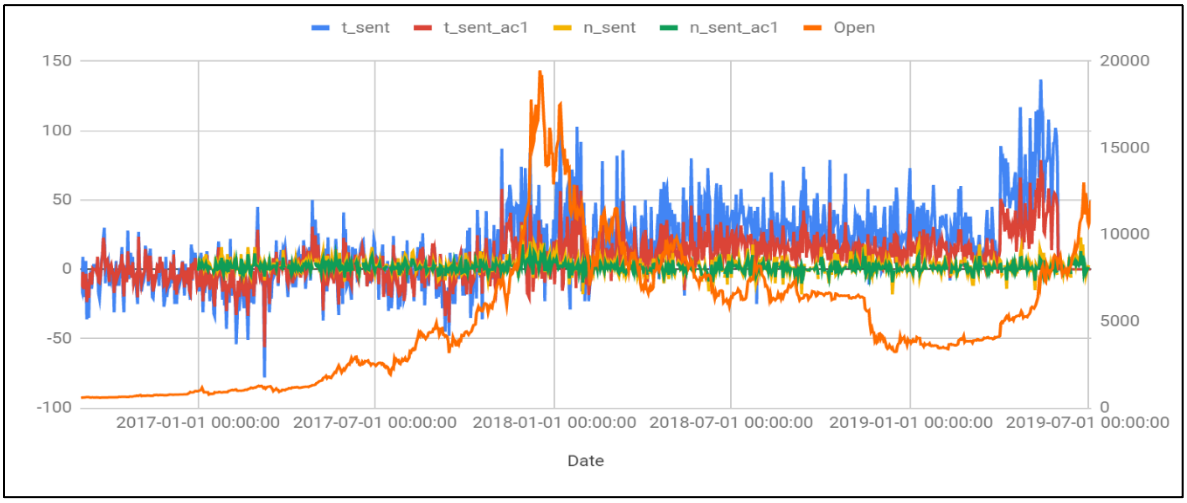


Figure 4.7 shows the price of Bitcoin and the sentiment for news and Twitter, it appears that the Twitter sentiment for Bitcoin increased before it reached its peak price of 20000 and peaked once again before the increase it has made this year. This may indicate that even though it is not feasible to predict price of Bitcoin with the total of a day’s

sentiment it may be accurate if a larger prediction window is taken into account. Figure 4.8shows the opening price of Bitcoin and the predicted price change of Bitcoin.

Figure 4.8: Open Price of Bitcoin and Predictive Price Change

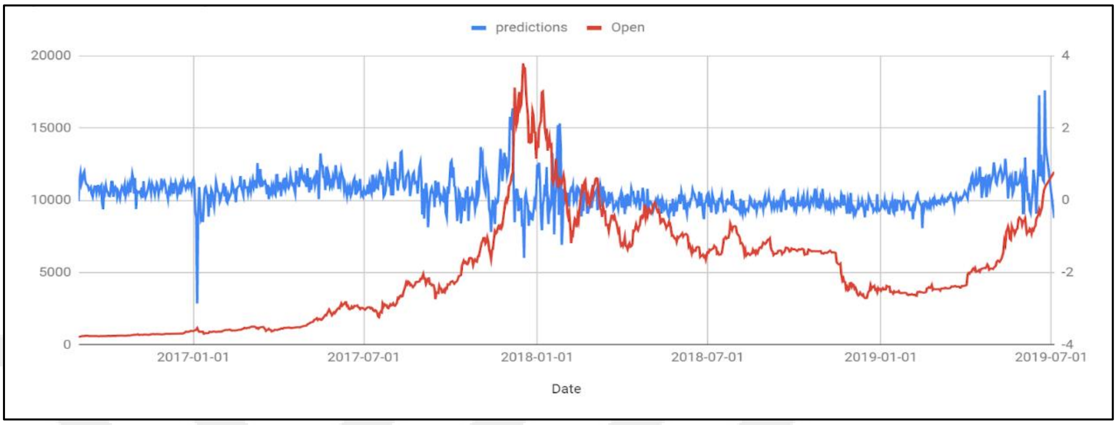
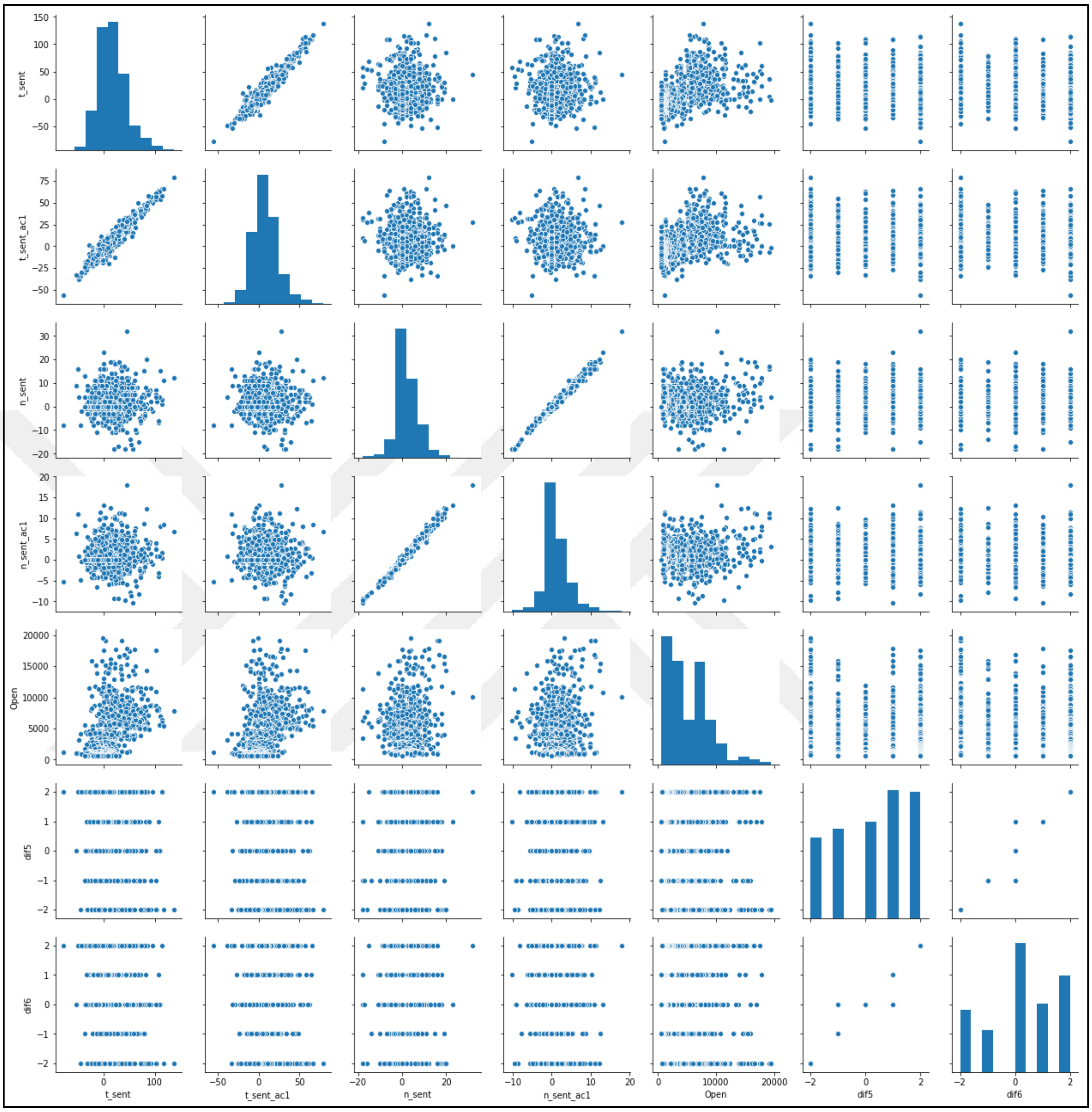


Figure 4.9: Pairplot of Sentiment and Price and Price Change



The pairplot of sentiment and price and the price difference of 3 days in percentage categories and sentiment is given in Figure 4.9. As you can see by the pair plot in Figure 4.9 there is no clear correlation between sentiment or sentiment with accuracy taken in to account with the price change of Bitcoin

5. DISCUSSION

The aim of the study conducted was to determine if accuracy rates of users and news sources should be taken into account when trying to predict the price change of an asset. The underlying hypothesis was that if the sentiment of a given day is calculated by giving users with a higher accuracy a higher weight it would increase the accuracy of the model. Both the regression results and results from the decision tree model, accuracy have indicated that they are bad models to use to predict the price change of Bitcoin. However the results that take the accuracy of each user into account has a higher adjusted r square than the results where the accuracy of each user is not taken into account. Therefore there is evidence that looking at individual accuracy rates may help predict the price of an asset like cryptocurrency or stock.

The accuracy rate of our sentiment classifiers may have been the reason why the accuracy rate to predict the price difference of Bitcoin was very low. The news sentiment classifier had an accuracy of 69%, however the Twitter sentiment classifier had an accuracy of 60%. Another limitation and possible reason why there was a low accuracy rate was that the labeled dataset to train the classifiers was labeled by one person. As discussed positivity is a subjective matter. A sentence may seem positive to one person but negative to another. A investor for example may take the fact that Bitcoin`s price has hit a, all time low as sign to invest, assuming that it will not go any lower and therefore invest. That being said the example was labeled negative due to the fact that any tweet or news article mentioning Bitcoins price drop was marked as negative. However the low accuracy may be caused by the fact that the price of Bitcoin is not correlated with Twitter sentiment of Bitcoin as previous research indicates. Sekan Ayvaz and Mohammed O. Shiha (2018) as well, found that there was no strong correlation between the price change of cryptocurrencies and Twitter sentiment. The price may not be correlated due to Bitcoins speculative nature and Twitters users over exaggerated speculations. By segmenting users based on their accuracy and then looking at the segment that has the highest accuracy we may therefore improve results.

Using the RSS feeds and Twitter streaming API the same methodology could be used in a streaming architecture. The tweets were put in through the accuracy dataframe starting from September 2016 to June 2019 and the accuracy rate of each tweet was equal to the accuracy rate of that user for that given time. Therefore a streamer architecture would have had the same results. If the accuracy rate was calculated by each closing day marking tweets with the previous days accuracy we would get the same results.

Even though using Sentiment Analysis with the accuracy of each user and news source taken into account is not a reliable way to predict the price change of Bitcoin, it does not mean that accuracy shouldn't be taken into account while trying to predict the price of a given asset. Due to the fact that there is no visible correlation between Twitter sentiment and Bitcoins price change the next step in my future work will be to see whether Twitter prediction accuracy would increase the prediction rate of stock prices. While creating dummy variables for the accuracy rate of each tweet rather than having accuracy multiplied with sentiment. Then by grouping Twitter users according to their current accuracy and seeing how results will be impacted if only the Twitter users with high accuracy scores are taken into account when calculating a given days sentiment.

In future work, Seghal Vivek (2007) approach to creating a sentiment classifier by using prelabeled yahoo posts that are labeled by the user posting the post will be utilized. Hand labeling tweets was a long process, time consuming and due to the fact that the person labeling the data may be biased may not be the best approach to create a training set for a sentiment classifier. However if the data is labeled by the users themselves then , one wouldn't need to waste time hand labeling all the training data and would be sure that the training data was labeled correctly since the one posting the post is stating the sentiment themselves. There may be issues training a sentiment classifier on yahoo data and then using it to analyze Twitter data since they are two different domains, the type of text on each site may differ. In my future work after training the classifier on the prelabeled yahoo data, using the labeled tweets from this research the accuracy of the created classifier that was trained on the yahoo data set will be calculated to see if the classifier would be effective on twitter data as well.

Also in future work, the total sentiment of a week will be used to predict the price of a stock or Bitcoin using the same methodology. An investor is unlikely to buy an asset according to one days of information, therefore in future work the overall positive sentiment and negative sentiment towards an asset over a different period of time will be analysed to determine if it will increase the accuracy rate of the final prediction.

6. CONCLUSION

In conclusion, this study aimed to determine if the accuracy rate of Twitter users and news sources should be taken into account when trying to predict the price of an asset while using Sentiment Analysis. Two different sentiment classifier to classify tweets and news articles that was relevant to Bitcoin were created. Using the tweets and news articles sentiment as price change predictions, an accuracy for each Twitter user and for each news source was calculated. Using the sentiment of tweets and news articles, price change of Bitcoin was predicted. Even though the regression results indicate that there is no clear correlation between the price change of Bitcoin and twitter sentiment and news sentiment even with accuracy taken into account , the l adjusted r square for the model where accuracy rates were taken into account was higher. There for results support the hypothesis that accuracy rates of news sentiment and tweets should be taken into account.

The results indicate that there is a higher accuracy when taking the accuracy of each user into account however, neither the model where accuracy was taken into account and the model were accuracy wasn't taken in to account where both a bad model to predict the price change of Bitcoin. Similar results were achieved by Vivek Sehgal (2007). When he used Trust score instead of accuracy rate to predict the price of stocks, Vivek (2007). Also found using the Trust score increased the accuracy to predict the price of a stock. The regressions that take news sentiment as well as Twitter sentiment achieved the highest adjusted r -squared indicating that if both Twitter sentiment and news sentiment with accuracy taken into account will increase the accuracy to predict an asset future price.

REFERENCES

Books

Liu, B.(2015) Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press

Miner, Gary. *Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications*. Academic Press, 2012.

Periodicals

Fama, E. (1965) The Behaviour of Stock Market Prices. Journal of Business, 64, 34-105

Gupta, Bhumika & Negi, Monika & Vishwakarma, Kanika & Rawat, Goldi & Badhani, Priyanka. (2017). Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python. International Journal of Computer Applications.

Perveen, Nasira & Missen, Malik Muhammad Saad & Rasool, Qaisar & Akhtar, Nadeem. (2016). Sentiment Based Twitter Spam Detection. International Journal of Advanced Computer Science and Applications. 7. 568-573. 10.14569/IJACSA.2016.070777.

Salloum, Said & Al-Emran, Mostafa & Abdel Monem, Azza & Shaalan, Khaled. (2017). A Survey of Text Mining in Social Media: Facebook and Twitter Perspectives. Advances in Science, Technology and Engineering Systems Journal. 2.

Yadav, Sameer. (2017). STOCK MARKET VOLATILITY - A STUDY OF INDIAN STOCK MARKET. Global Journal for Research Analysis. 6. 629-632.

Other Publications

Asur, Sitaram & Huberman, Bernardo. (2010). Predicting the Future with Social Media. Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010. 1. 10.1109/WI-IAT.2010.63.

Ayvaz, Serkan & O. Shiha, Mohammed. (2018). A Scalable Streaming Big Data Architecture for Real-Time Sentiment Analysis. 10.1145/3264560.3266428.

Beckmann, Marcelo. (2017). Stock Price Change Prediction Using News Text Mining.

Birjali, M., Beni-Hssane, A., & Erritali, M. (2017). Machine Learning and Semantic Sentiment Analysis based Algorithms for Suicide Sentiment Prediction in Social Networks. Procedia Computer Science, 113, 65-72. doi:10.1016/j.procs.2017.08.290

Bollen, Johan & Pepe, Alberto & Mao, Huina. (2009). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. Computing Research Repository - CORR.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *J. Comput. Science, 2,* 1-8.

Bovet, A., Morone, F., & Makse, H. A. (2018). Validation of Twitter opinion trends with national polling aggregates: Hillary Clinton vs Donald Trump. Scientific Reports, 8(1). doi:10.1038/s41598-018-26951-y

Brooke, Julian. (2009). A semantic approach to automated text sentiment analysis.

Chen, R.J., & Lazer, M. (2011). Analysis of Twitter Feeds for the Prediction of Stock Market Movement

Cortis, Keith & Freitas, Andre & Daudert, Tobias & Hurlimann, Manuela & Zarrouk,Manel & Handschuh ,Siegfried & Davis, Brian (2017).Semeval-2016 task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News (SemEval-2017)

Dung Nguyen, Vu & Varghese, Blesson & Barker, Adam. (2013). The Royal Birth of 2013: Analysing and Visualising Public Sentiment in the UK Using Twitter. 10.1109/BigData.2013.6691669.

Fama, E. F., Fisher, L., Jensen, M. C., & Roll, R. (1969). The Adjustment of Stock Prices to New Information. International Economic Review, 10(1), 1. doi:10.2307/2525569 h, Chong & Sheng, Olivia. (2011). Investigating Predictive Power of Stock Micro Blog Sentiment in Forecasting Future Stock Price Directional Movement.. International Conference on Information Systems 2011, ICIS 2011. 4.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *KDD*.

Kolchyna, Olga & Souza, Thársis & Treleaven, Philip & Aste, Tomaso. (2015). Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination.

Li, Mengdi & Ch'ng, Eugene & Chong, Alain & See, Simon. (2016). Twitter Sentiment Analysis of the 2016 U.S. Presidential Election Using an Emoji Training Heuristic.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167. doi:10.2200/s00416ed1v01y201204hlt016

Liu, B. (2015). Sentiment Analysis. doi:10.1017/cbo9781139084789

Liu, B. (n.d.). Sentence Subjectivity and Sentiment Classification. Sentiment Analysis, 70-89. doi:10.1017/cbo9781139084789.005

Liu, Y., Huang, X., An, A., & Yu, X. (2007). ARSA: a sentiment-aware model for predicting sales performance using blogs. SIGIR.

Mäntylä, M. V., Graziotin, D., & Kuutila, M. (2018). The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. Computer Science Review, 27, 16-32. doi:10.1016/j.cosrev.2017.10.002

Mittal, A. (2011). Stock Prediction Using Twitter Sentiment Analysis

Nakov, Preslav & Ritter, A & Rosenthal, Sara & Sebastiani, Fabrizio & Stoyanov, V. (2016). Semeval- 2016 task 4: Sentiment analysis in twitter. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). 1-18.

Nann, Stefan & Krauss, Jonas & Schoder, Detlef. (2013). Predictive analytics on public data-The case of stock markets. ECIS 2013 - Proceedings of the 21st European Conference on Information Systems.

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

Pak, Alexander & Paroubek, Patrick. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Proceedings of LREC. 10.

Pang, Bo & Lee, Lillian & Vaithyanathan, Shivakumar. (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques. EMNLP. 10. 10.3115/1118693.1118704.

PBarnaghi, Peiman & Ghaffari, Parsa & Breslin, John. (2016). Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment. 52-57. 10.1109/BigDataService.2016.36.

Saif, H., He, Y., & Alani, H. (2012). Semantic Sentiment Analysis of Twitter. The Semantic Web – ISWC 2012 Lecture Notes in Computer Science, 508-524. doi:10.1007/978-3-642-35176-1_32

Sarlan, Aliza & Nadam, Chayanit & Basri, Shuib. (2014). Twitter sentiment analysis. 212-216. 10.1109/ICIMU.2014.7066632.

Schumaker, Rob & Chen, Hsiu-chin. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Trans. Inf. Syst.. 27. 10.1145/1462198.1462204.

Sehgal, Vivek & Song, Charles. (2007). SOPS: Stock Prediction Using Web Sentiment.. 21-26.

Taboada, Maite & Brooke, Julian & Tofiloski, Milan & D. Voll, Kimberly & Stede, Manfred. (2011). Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics. 37. 267-307. 10.1162/COLI_a_00049.

Zhang, X.S., Fuehres, H., & Gloor, P.A. (2011). Predicting Stock Market Indicators Through Twitter “I hope it is not as bad as I fear”.

Bitcoin (BTC) Historical Data. (n.d.). Retrieved from https://coinmarketcap.com/currencies/bitcoin/historical-data/?start=20160901&end=20190705

Twitter: Number of active users 2010-2019. (n.d.). Retrieved from https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

Jefferson-Henrique. (2016, April 15). Jefferson-Henrique/GetOldTweets-java. Retrieved from https://github.com/Jefferson-Henrique/GetOldTweets-java