INTERNATIONAL HELLENIC UNIVERSITY

Forecasting Cryptocurrency Price Movements with Predictive Social Media Analytics

Michail Vlachos - Giovanopoulos

SID: 3308200024

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2022

THESSALONIKI – GREECE

Forecasting Cryptocurrency Price Movements with Predictive Social Media Analytics

Michail Vlachos - Giovanopoulos

SID: 3308200024

Supervisor:	Assoc. Prof. C. Tjortjis
Supervising Committee Members:	Assoc. Prof. M. Drakaki
	Dr. S. Stavrinidis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2022

THESSALONIKI – GREECE

Abstract

This dissertation was written as a part of the MSc in Data Science at the International Hellenic University, by student Michail Vlachos – Giovanopoulos under the supervision of Prof. Christos Tjortjis.

The rapid growth of social media has significantly increased the value of social media data. Social media platforms have inserted in people's everyday lives, giving analysts the opportunity to analyze the information and use them to make future predictions for events like cryptocurrency prices, stock market movements, and many more. This project aims at collecting Twitter data about cryptocurrencies, and specifically about Bitcoin, Ethereum, and Dogecoin, conduct sentiment analysis, and perform price predictions based on the sentiment scores. After examining the results of the correlation analysis, it was decided to gather tweets volume and volume of transactions for a longer period, which finally produced very good results. Various machine learning techniques were used in order to further improve the predictions. Finally, the best forecasts were achieved with 91.5%, 93.3%, and 83.6% R-Squared for Bitcoin, Ethereum, and Dogecoin respectively, accompanied by 0.5%, 0.3%, and 0.4% Mean Squared Error for each of the three crypto-currencies respectively.

At this point, I would like to thank my supervisor Prof. Christos Tjortjis for his excellent mentoring and for giving me the opportunity to work with him on a very interesting and challenging project, as well as Dr. Paraskevas Koukaras for his valuable guidance on the practical part of the dissertation.

I would also like to thank my family for their support and for believing in me, especially during the period of my studies.

Michail Vlachos - Giovanopoulos

07/01/2022

Contents

List of Figures

List of Tables

1 Introduction

The first chapter is going to introduce the challenge of the project. First, the fundamentals of Cryptocurrencies are going to be presented, and then the reasons why Social Media Analytics are the ideal tool to perform forecasting of their prices. Finally, it will be explained why Twitter is the best option for this purpose.

1.1 The Problem

Cryptocurrency price forecasting can give a significant boost to cryptocurrency investors for making proper investment decisions in order to achieve higher profits, while it can also support financial researchers for studying cryptocurrency markets behavior. Cryptocurrency price prediction can be considered as a common type of time series problems, such as stock price prediction [1].

As it will be explained in the next section, there are high fluctuations observed in the cryptocurrencies' prices over short time. This intense instability in the value of cryptocurrencies signifies that both people who have in mind to use them as actual currencies, and investors, can not be assured. This can be attributed to the fact that cryptocurrencies are a relatively new form of money and the actual reason for these large fluctuations is still an area of debate [2]. The inevitable fragility of decentralized systems based on blockchain technology might also play a significant role in these fluctuations [3]. Another opinion states that this high volatility has to do with people's perception, and bearing in mind that cryptocurrency values do not behave like actual currencies, the prediction is really difficult [2], [4].

1.2 Cryptocurrency

Cryptocurrency is an alternative, virtual or digital form of money, that can be exchanged online for goods and services, and is based on the principles of cryptography for the network's assurance, the control of the creation of additional assets, and the verification of the coins' transfer [5]. In more details, cryptocurrencies use a decentralized technology called blockchain, which is a public digital ledger of transactions that records information

in a way that makes it difficult to hack or alter [5]–[7]. The first decentralized cryptocurrency, Bitcoin, was released in 2009 by Satoshi Nakamoto, as open-source software [5], [8]. This event draw the attention and evoked the creation of more than 10,000 altcoins (alternative cryptocurrencies) by December 2021 [9]. Now, the most important cryptocurrencies are Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC) and Cardano (ADA).

The world of the cryptocurrencies has grown exponentially the last years, with their total market capitalization being over $2 trillion, where Bitcoin occupies a percentage of approximately 40%, Ethereum 21%, Litecoin around 0.5% and Cardano almost 2% [10]. Considering their massive value in the market, these currencies attract not only people who look at them as investment opportunities, but also those who consider them as actual currencies. As a consequence, there are huge fluctuations in cryptocurrencies' prices over short time [5]. For instance, the price of Bitcoin climbed from $6,483 in March 2020, to $58,734 in March 2021, an increase of more than 800%, while the value of Ethereum recorded a rise of almost 1,020% from $384 in October 2020, to $4,298 in October 2021 [11], [12]. Similarly, Litecoin's value dropped almost 87%, from $238 in December 2017, to $31 one year later, while the price of Cardano decreased more than 50% in 2 months, from $2.15 in May 2021, to $1.05 in July 2021 [13], [14].

This intense instability in the value of cryptocurrencies signifies that both people who have in mind to use them as actual currencies, and investors, can not be assured [5]. This can be attributed to the fact that cryptocurrencies are a relatively new form of money and the actual reason for these large fluctuations is still an area of debate [2]. The inevitable fragility of decentralized systems based on blockchain technology might play a significant role [3]. Another opinion states that this high volatility has to do with people's perception, and bearing in mind that cryptocurrency values do not behave like actual currencies, the prediction is really difficult [3], [5].

1.3 Social Media Analytics as a Forecasting Tool

The development of the World Wide Web (www) has created the social media platforms, which have become a main part of people's daily routine [15]. These technologies allow people to communicate and express themselves by sharing and exchanging information and ideas, through texts, photos and videos [16]. The most popular social media sites such as Twitter, Instagram, Facebook and LinkedIn have millions (even billions, some of them) of monthly active users all over the world . Therefore, there is a vast amount of

data generated every day, which is easily accessible and can be referred to as Social Media Data [15].

Social media data can be helpful and exploited in various ways, and because of that, many companies focus on social media in order to gain insights and make the most of it. More particularly, data retrieved from social media platforms hide unique information about customers' habits, interests and their experiences, opinions about the company and its products, pricings, reviews, and more [16], [17]. Alternatively, it would cost businesses more to gain this knowledge using traditional methods [18]. In general, if social media data are processed and analyzed correctly, they can be really valuable and useful for the company in various ways, such as in deciding about future moves and improvements, and in stock market or cryptocurrency price prediction.

1.4 Why Twitter

Twitter is one of the most popular social media networks, with around 400 million users and more than 200 million daily active users [19]. What distinguishes it from other social media platforms such as Facebook, Instagram, LinkedIn, is the fact that it has the most textual context, which makes it the best choice to perform analysis based on text, like sentiment analysis. Other social media networks contain images which can also be analyzed, but text is much easier and more accurate [20]. People post on social media to express themselves about various issues. Hence, if someone wants to make a statement about a serious topic, it is more likely to post on Twitter rather than other social media [21].

As for the technical aspect, Twitter offers users the ability to create a developer account and use the API services that enable for easy data gathering. Twitter pays much attention to privacy and security, however, tweets can be downloaded by making simple requests to the API [21]. More details about the data gathering will be presented in following chapter.

1.5 Dissertation Objectives

The first objective of this dissertation is to gather Twitter data for a three month period, from September 15, 2021 until December 15, 2021, and conduct a sentiment analysis to explore and analyze the public opinion about three particular cryptocurrencies. Then, it will be examined to what extent the public sentiment affects their prices, and based on

that, try to forecast the cryptocurrencies' prices. The final objective is the attempt to improve the performance of the predictions, compared to the ones of the literature.

2 Background

The following section encompasses a few background information about the tools and methodologies that were used and aim at assisting in understanding the practical part of the dissertation.

2.1 Twitter

As already explained in the previous chapter, Twitter offers easy data gathering, which can be achieved with python libraries, either with or without using Twitter API, as presented below.

2.1.1 Twint library

Twint is an advanced, open-source Twitter scraping tool that enables tweets retrieval from Twitter profiles without using Twitter's API. Users can choose to scrape tweets by filtering for search word, date (not more than 7 days old), username, minimum number of likes, retweets, or followers, etc. Since twint does not require any authentication from Twitter, the initial setup can be done quickly, with only prerequisite a Python IDE. Some more benefits are the fact that Twitter API's limitations do not apply to this tool, so almost all tweets can be fetched, and it can be used incognito without Twitter sign up [22].

2.1.2 Twitter's Search API

Twitter's Search API, which is part of Twitter's REST API, is a powerful tool that gives access to recent tweets with the help of queries, and behaves in a similar way like the Search feature of Twitter's mobile or web version [23]. The retrieved data will belong to the last 7 days and will comprise only a segment of the tweets published during this period. The standard Search API does not focus on completeness, but in relevance. Hence, some tweets and users will not be part of the results, unless somebody chooses the premium or enterprise Search APIs. Furthermore, one serious drawback of Twitters' API is the rate limit, which allows users to make up to 900 requests per 15 minutes [24].

2.1.3 Tweepy library

In contrary to Twint, Tweepy interacts with Twitter through Twitter's Search API. Tweepy is also a non-proprietary Python library which supports both OAuth 1a (application-user) and OAuth 2 (application-only) authentication [25]. The main difference between these two is that in OAuth 2 an application makes API requests without the user context, and this method of authentication is used in cases of read-only access to public information. In both methods, the first to be done is apply -free of charge- for a Twitter Developer License, which may take a day or two to get approval, and obtain credentials. Once obtained consumer keys and access tokens, one is ready to use them and access Twitter API.

2.2 MySQL and PHPMyAdmin

MySQL is a free, easy to use relational database management system based on Structured Query Language, that offers various advantages [26]. Some of its advantages are that it offers security through a solid data security layer, it is easily scalable and can deal with big amounts of data, and it owns a unique storage engine architecture which makes it more reliable, faster and cheaper.

PHPMyAdmin is a free software tool written in PHP, for the purpose of administrating MySQL over the Web [27]. It has become the most popular application for MySQL database management. Users can perform various operations not only via user interface, but also by executing SQL statements, operations such as creating, updating, dropping, altering, deleting, importing and exporting MySQL database tables [28].

2.3 Sentiment Analysis

As already mentioned, Internet has significantly evolved the last few decades, and has greatly changed peoples' lives and everyday habits [21]. Using the internet, people can easily communicate, read reviews about products and services, and make purchases. After the completion of their actions, people have the need to share their experience and opinion with their virtual friends. That is why there are many online forums and most websites offer comment sections. However, the most common way of expressing their feelings is by posting on their preferred social media platforms. This fast-growing amount of data is really valuable for businesses that analyze the public opinion about a particular subject. Some topics, such as new products, services and, in this case, cryptocurrencies, are really

attractive and awaken great interest for gathering feedback and analyzing how the public feels and reacts.

In the past, a company could simply inform people through an advertisement who they are and what they sell, and gain people's interest by telling them how precious their products or services are. However, this does not apply anymore. Nowadays, it is not a matter of what the company says about the product, but what customers tell each other that it is [29]. As a result, public opinion can have an extreme effect on the brand or, in general, on the topic of analysis.

Sentiment analysis, also known as opinion mining or emotion AI, is the use of *Natural Language Processing* (NLP) to identify, extract and study affective states and subjective information [30]. In other words, it is the process of determining whether the text is positive, negative or neutral [31]. The categorization of the semantic orientation of a text segment in one of these three classes is the process of *polarization* [16]. Sentiment analysis is generally considered a classification problem. One of it's main challenges is that people express and interpret sentiment polarity in different ways. Also, a particular word can have different meanings in different sentences based on the context, which is known as *polysemy* [32]. There are two main approaches that analysts use to deal with this problem:

- *Rule-Based Approach*: In rule-based sentiment analysis algorithms, the algorithm calculates the sentiment score from a set of manually created rules [33]. For instance, the analyst is going to label words like "bad", "horrible", "boring" as negative, and words like "excellent", "nice", "awesome" as positive. Human judgement is crucial here since the rules that they are going to define will determine the polarity of the input data.
- *Machine Learning Approach*: The classifier uses a training dataset with already labeled data as positive or negative, to get trained, and then classifies the test data [20]. Consequently, the human factor is less important in this approach.

2.3.1 TextBlob

TextBlob is a python library (part of *NLTK* library) for textual data processing. It offers API access to various NLP tasks such as sentiment analysis, part-of-speech tagging and classification. In the context of sentiment analysis, this sentiment analyzer uses lexicons with already scored words, and returns two properties for a sentence [34], [35]:

- *Polarity*, which lies between -1 and 1, and indicates negative and positive sentiments respectively.
- *Subjectivity*, which lies in between 0 and 1, and is assigned based on personal opinion, emotion or judgment.

2.3.2 Vader

Valance Aware Dictionary for sEntiment Reasoning is another popular python library (also part of *NLTK* library) which operates also as a rule-based sentiment analyzer. It calculates the sentiment of the text based on a set of lexical features which are already classified as positive or negative depending on their semantic orientation. Vader sentiment analyzer returns four scores [34], [36]:

- the probability of a given sentence to be positive, negative, and neutral. These three probabilities will add up to 100%, and
- the *Compound Score* which is a metric that calculates the sum of all the lexicon ratings that have been normalized between -1 (most extreme negative) and +1 (most extreme positive)
    - Positive sentiment: compound score >= 0.05.
    - Neutral sentiment: compound score > -0.05 and compound score <0.05.
    - Negative sentiment: compound score <= -0.05.

2.4 Machine Learning

Machine Learning is an application of Artificial Intelligence (AI) that enables systems to automatically learn from data. The aim is to explore patterns in the data and utilize them for better future decisions, by making the appropriate actions autonomously [37]. There are two main categories of machine learning algorithms:

- *Supervised learning*, which refers to utilizing already labeled data to train the algorithm and predict future events [37]. There is also the possibility to have a comparison between the actual and the predicted values, in order to indicate errors and make the appropriate modifications on the model to improve the performance. Based on the numerical type of the output, the problems are categorized into *Regression* and *Classification* problems. *Regression* refers to predicting a real or continuous value, e.g. stock market prediction, while *Classification* is used to predict a discrete class label, e.g. loan eligibility (yes or no) [38].

- *Unsupervised learning*, which deals with unlabeled data aiming to discover previously unknown patterns and information [38], [39]. It provides users the ability to perform more complex processing tasks than supervised learning, and simplify a dataset to allow for easier supervised learning. Two fundamental subcategories of unsupervised learning are *dimensionality reduction* and *clustering*, which reduce the dimensions of the input data, and determine a small dataset sufficiently describing the initial one, in order to represent it, respectively.

2.4.1 Algorithms

Machine Learning Algorithms are the means for turning a dataset into a model. The best choice of algorithm depends on the kind of problem, the computer resources, and the nature of the data. The algorithms that were used in the literature and in this project are going to be presented below [40].

2.4.1.1 Linear Regression – Multiple Linear Regression (MLR)

Linear Regression is used to predict continuous or real variables and shows the relationship between the independent variable (X-axis) and the dependent variable (Y-axis). The linear regression is called *simple linear regression* if there is a single input variable (x), and *multiple linear regression* if there are more than one input variables [41].

2.4.1.2 Polynomial Regression

In *Polynomial Regression*, the relationship between the independent variable x and the dependent variable y is described as an n^{th} degree polynomial in x. This type of regression describes the fitting of a nonlinear relationship between the value of x and the conditional mean of y [42].

2.4.1.3 Recurrent Neural Network (RNN)

A *Recurrent Neural Network* is a type of artificial neural network that uses sequential data or time series data to learn. What differentiates it is its memory, as it uses information from prior inputs to influence the current input and output. The output of RNN depend on the prior elements within the sequence. Some of the applications of recurrent neural networks are natural language processing (nlp), language translation, and speech recognition [43].

2.4.1.4 Long Short-Term Memory (LSTM)

Long Short Term Memory Network is an advanced RNN that enables information to remain. It remembers the previous information and uses it to process the current input. LSTMs are devised in a way to stay away from long-term dependency problems, as they are not able to remember long-term dependencies [44].

2.4.1.5 Extreme Gradient Boosting (XGBoost)

XGBoost is a decision tree-based ensemble machine learning algorithm that uses a gradient boosting framework. *Decision Tree* is mostly preferred in classification problems, and it is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules, and each leaf node represents the outcome. *Gradient boosting* is a type of machine learning boosting which relies on the intuition that the best possible next model minimizes the overall forecast error when combined with previous models. In prediction tasks with unstructured data (images, text, etc.) artificial neural networks achieve better results than all other algorithms or frameworks [45]–[47].

2.4.1.6 Multilayer Perceptron (MLP)

Multilayer Perceptron is a complement of feed forward neural network. It comprises three types of layers, the input, output and hidden layer. The input layer is responsible for receiving the input signal to be processed. The output layer takes care of performing the required task such as prediction and classification. In between of these two layers, there is a random number of hidden layers which define the computational engine of the MLP. The direction of the data is from input to output layer, similar to a feed forward network. The neurons in the MLP are trained with the back propagation learning algorithm, which allows the multilayer perceptron to iteratively adjust the weights in the network, by aiming to minimize the cost function. Multilayer perceptrons are devised in a way to estimate any continuous function and can deal with not linearly divisible problems. The most common applications of MLP are pattern classification, recognition, prediction and approximation [48], [49].

2.4.1.7 Support Vector Regression (SVR)

Support Vector Regression is a supervised learning algorithm used for discrete values prediction. The straight line that is required to fit the data is referred to as hyperplane. SVR tries to fit the best line within a threshold value, unlike other regression models that

try to minimize the error among the actual and the predicted value. The threshold value is the distance between the hyperplane and the boundary line. The fit time complexity of SVR is more than quadratic with the number of samples which makes it hard to scale to datasets with more than a couple of 10000 samples. For large datasets, Linear SVR or SGD Regressor is used. Linear SVR achieves quicker implementation than SVR but takes into account only the linear kernel. The SVR's model relies only on a portion of the training data, because the cost function does not consider samples whose prediction is close to their target [50].

2.4.2 Autoregressive Integrated Moving Average (ARIMA)

An *ARIMA* model is used for forecasting time series data and is generally denoted as ARIMA (p, d, q), where *p* is the order of autoregressive model, *d* is the degree of differencing, and *q* is the order of moving-average model. It uses differencing to convert a non-stationary time series into a stationary one, and the predict future values from historical data. Such models use autocorrelations and moving averages over residual errors in the data to forecast future values [51].

2.4.3 Evaluation Metrics

Evaluation metrics explain the performance of a model. A common question is "how accurate are the results?" which refers only to classification, not regression. The performance of a regression model has to be calculated in terms of an error in the predictions. The most common error metrics are *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE)*, and *Mean Average Error (MAE)* [52]. MSE represents the average of the squared difference among the actual and predicted values in the dataset [53]. In other words, it measures the variance of the residuals. RMSE is the square root of MSE and measures the standard deviation of the residuals. MAE represents the average of the absolute difference among the original and predicted values in the dataset. It measures the average of the residuals in the dataset. Moreover, *R Squared* explains how well the independent variables in the linear regression model explain the variability in the dependent variable. It is similar to accuracy for classification. However, for comparing the accuracy among different linear regression models, RMSE is a better choice than R Squared.

Another metric that was used in the project is *Maximum Error (ME)*, which indicates the absolute value of the most significant difference between a predicted variable and its real value [54].

2.4.4 Normalization – MinMaxScaler

When dealing with real data, a common issue is that the values are of different scales. Unscaled input variables can result in a slow or unstable learning process, whereas unscaled target variables on regression problems can result in exploding gradients causing the learning process to fail.

Normalization is a rescaling of the data from the original range so that all values are within the same range (usually between 0 and 1). *MinMaxScaler* is an object of python's *scikit-learn* library, that normalizes a value as follows:

$$y = (x - min) / (max - min)$$

where the minimum and maximum values pertain to the value *x* being normalized [55].

2.4.5 Pearson's Correlation Coefficient

Correlation coefficients are used to indicate how strong a relationship is between two different variables. There are several types of correlation coefficients, of whom the most popular is *Pearson's* [56]. *Pearson's correlation* (also called Pearson's R) is known as the best method of measuring the association between variables, as it is based on the method of covariance [57]. The covariance method assumes that the returns will be normally distributed around the mean of a normal or bell-shaped probability distribution [58].

Correlation coefficients have the following properties [57]:

- They can take values from -1 to +1.
- The absolute value of the indicator refers to the strength of the relationship.
- The sign refers to the direction of the relationship (positive or negative).
- If the value is near ±1, there is **perfect correlation**, which means that if one variable increases, the other variable will also increase, if the correlation is positive, or decrease, if the correlation is negative.
- If the value ranges between ±0.5 and ±1, there is **strong correlation**.
- If the value ranges between ±0.3 and ±0.49, there is **medium correlation**.
- If the value is less than ±0.29, there is **small correlation**.
- If the value is zero, there is **no correlation**.

3 Literature Review

In this section, a review of the most relevant researches that have been conducted on the same problem will be presented. Cryptocurrency price prediction with predictive social media analytics is a relatively new idea and has gained a lot of attention since more people are interested in cryptocurrency trading, and more data are available on the web. The approaches are going to be examined and the findings and results of the following papers are going to be described:

1. Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis
2. Short-Term Bitcoin Price Fluctuation Prediction Using Social Media and Web Search Data
3. Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model

3.1 Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis

A research attempt for cryptocurrency price prediction considered tweet volumes and sentiment analysis [2]. The main goal was to collect tweets and google trends data referring to the two largest cryptocurrencies and try to predict changes in their prices. Bitcoin's and Ethereum's market capitalization in 2018 was more than $160 billion, and both cryptocurrencies had experienced large price fluctuations. More particularly, what they wanted is to find out if a relationship between Twitter sentiment and cryptocurrency price changes could be determined.

In order to collect the data, the authors used Twitter's API and Google Trends to retrieve tweets and Google search data respectively.

On one hand, Tweepy, a Python library, is used to get access to Twitter API and collect the data. This tool enables the user to filter their search based on hashtags or words. In this case, the selected search terms were "#bitcoin" and "#ethereum", referring only to

English tweets. The script used was scheduled to run for 2 months, every 15 minutes and collect 1,500 tweets each time. So the final dataset consisted of 30,420,063 tweets.

On the other hand, Google gives access to search data through Google Trends. Google is the most popular search engine which is used for more than 74% of all web searches, so, beyond a doubt, Google search data can be really valuable by providing information about what people are interested in, and how interested they are in specific topics. These data provided by Google, are not search volumes, but a search volume index (SVI). The SVI is determined by the division of each data point, by the total searches within a geographic region and time range. Then, the numbers will be scaled between 0 and 100 on a search term's proportion to all searches on all topics. In case that the query refers to trends data for a period longer than 3 months, then the search volume indices are summed at a weekly level. In order to weigh up these SVIs, the technique explained by Erik Johansson has been used. So, firstly all the daily SVI data were collected in 90 day increments and combined into a single increment for the entire time period analyzed. Then the weekly SVI were calculated by sorting the data for the same time span, combined at a weekly level. Thirdly, when the dates overlapped, the weekly SVI were divided with the daily SVI, in order to define an adjustment factor. Finally, the daily SVI values were multiplied by the adjustment factor. It is worth mentioning that in instances when the SVI was less than 1, Google Trends did not give any information about the specific value, it just returned "< 1". So, the authors decided to replace it with 0.5, which is the halfway value, in order to enable an adjustment calculation.

As in the case of tweets retrieval, the selected search terms were "bitcoin" and "ethereum", which were the least ambiguous ones, compared for instance to their abbreviations, "BTC" and "ETH" respectively.

As regards the Tweet Volume, Twitter's API which was used by the writers to retrieve tweets through Tweepy, sets a limit of 1,500 tweets that can be collected per instance. Consequently, they weren't able to know the total number of tweets about their search terms for a given day, so they made use of www.bitinfocharts.com, which gave them the daily total number of tweets for free.

The last step before the data analysis, was Tweets cleaning. Tweets contain characters that are not useful for a sentiment analysis. For the purpose of preprocessing they used readily available packages and regular expressions. So, all the words came in the correct

form, and the regular expressions removed hashtags, quotes, question marks and https links.

When it comes to the sentiment analysis, VADER (Valence Aware Dictionary for sEntiment Reasoning) sentiment analysis was used. VADER analysis measures both subjectivity and polarity. It turned out that only half of the tweets had any objective VADER score, while all the others were neutral, as shown in Figure 1 for Bitcoin, and in Figure 2 for Ethereum.

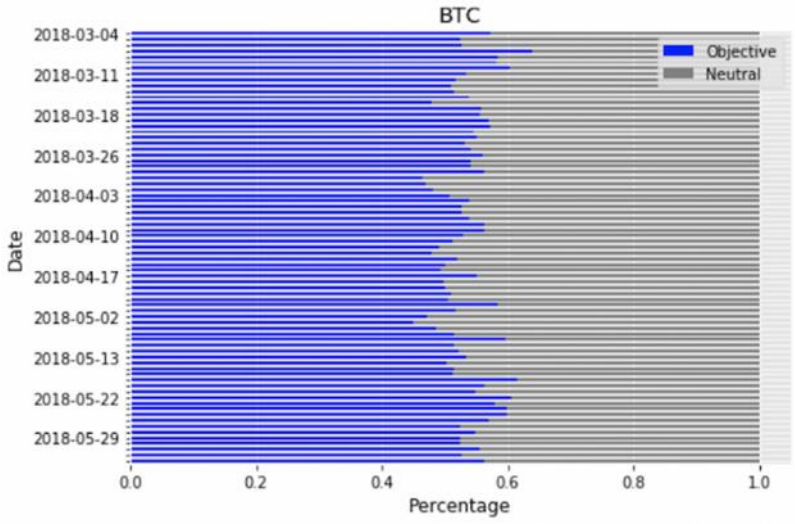

Figure 1. Bitcoin percent of objective (having a positive or negative sentiment) versus neutral tweets.

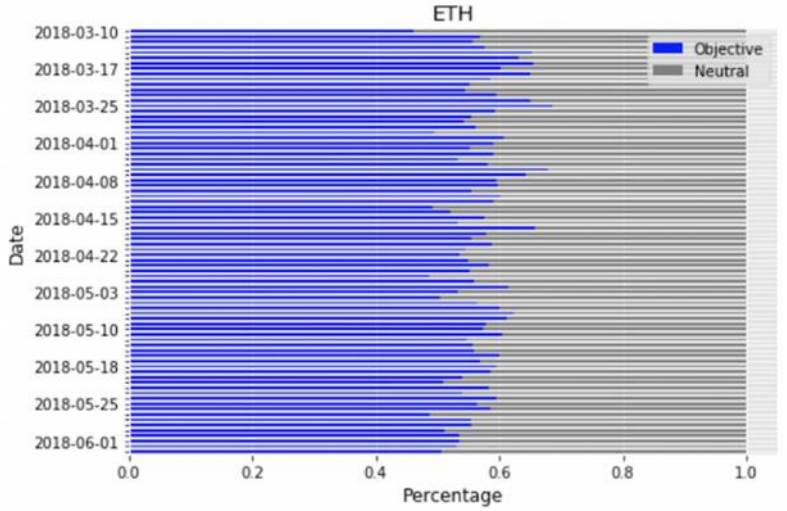

Figure 2. Ethereum percent of objective (having a positive or negative sentiment) versus neutral tweets.

Figure 3 and Figure 4 show that based on VADER sentiment analysis, tweets were more neutral than objective.

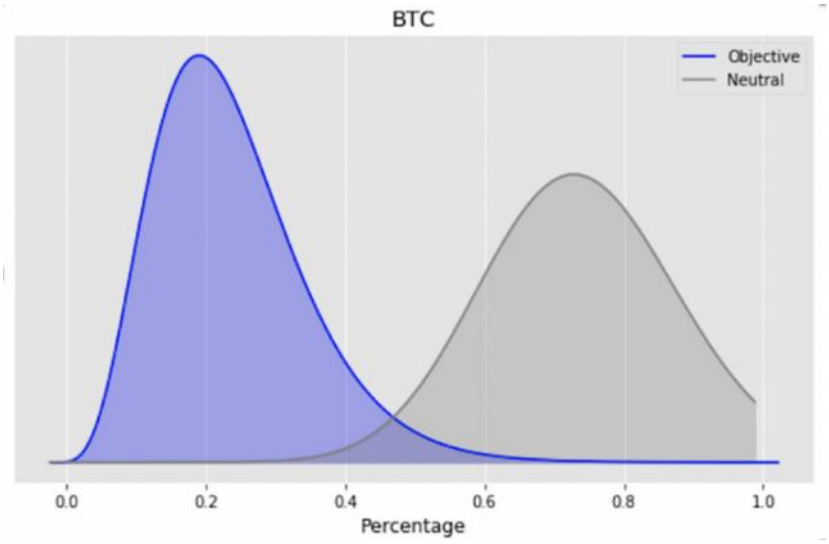

Figure 3. Bitcoin objectivity distribution.

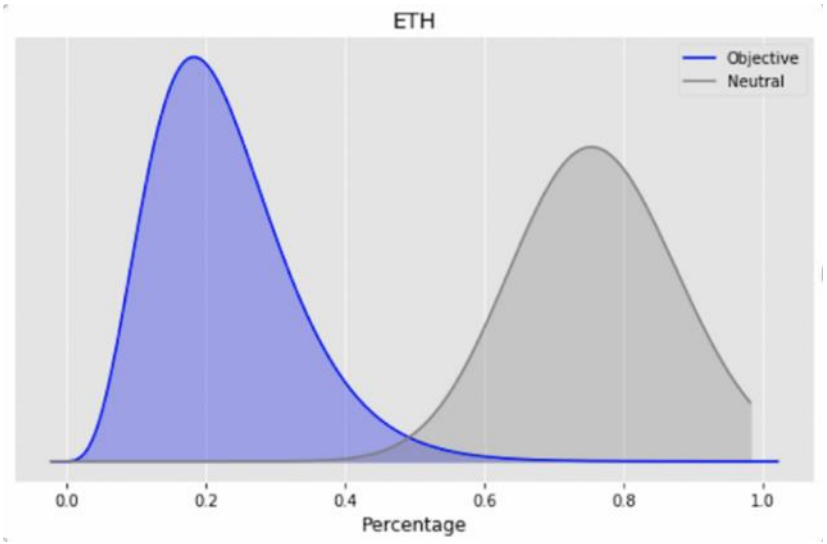

Figure 4. Ethereum objectivity distribution.

The authors still believed that despite the fact that half of the tweets had a neutral sentiment, it was still possible that the positive or negative sentiment of the remaining tweets could give important insights to the model if there was a correlation between sentiment and price changes. In Figure 5 and Figure 6 it can be observed that despite the price fluctuations, the tweet sentiment does not change (only one day the sentiment became negative for Bitcoin, not a single day for Ethereum).

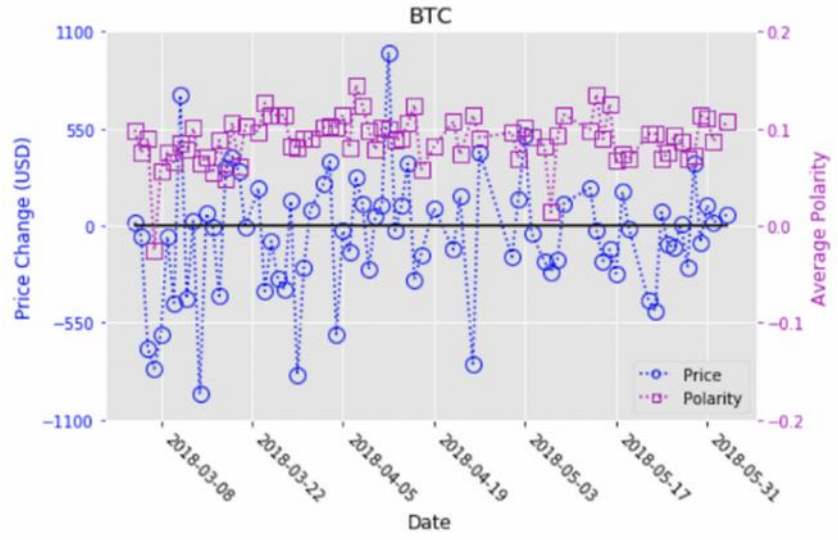

Figure 5. Bitcoin price change and Daily Average Tweet Polarity by date.

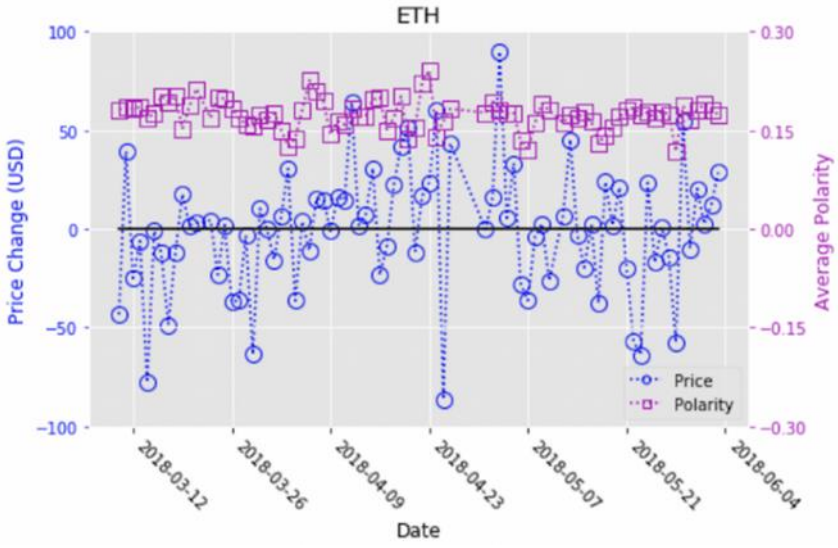

Figure 6. Ethereum price change and Daily Average Tweet Polarity by date.

Consequently, the sentiment analysis was not used in the model.

As regards the correlation between Google Trends search data and cryptocurrency price changes, the writers decided to use the “Pearson R” and the “p-value” metrics. The Pearson R is 0.817, which means that the price is positively correlated with Google Trends data, with a p-value of 0.000, which means that the result is statistically significant.

When it comes to Tweet Volume and Cryptocurrency Prices, the analysis showed up that the sentiment of the tweets remained positive even when the price decreased. However, this could be attributed to the fact that people who still post on Twitter are interested in

cryptocurrencies for other reasons, such as privacy, when prices decrease. Given that the number of users tweeting about cryptocurrencies may fluctuate with prices, it could be agreed that tweet volume would be a more reliable indicator than sentiment.

Finally, out of the three model inputs, tweets' sentiment proved to be unreliable when cryptocurrency prices decreased, therefore Google Trends and tweet volume were selected, which displayed a high correlation with price. 80% of the data were split into training set, and the remaining 20% were used for testing the model. As the machine learning algorithm, multiple linear regression was used.

As it can be observed in Figure 7 and Figure 8, the authors managed to show that search volume index and tweet volume are highly correlated with cryptocurrency prices in both cases, when the prices increase and decrease. Through multiple linear regression they accurately predicted price changes for Bitcoin.

Figure 7. Model fit shown as actual price for Bitcoin, training data shown by green dots, and test results as red dots.

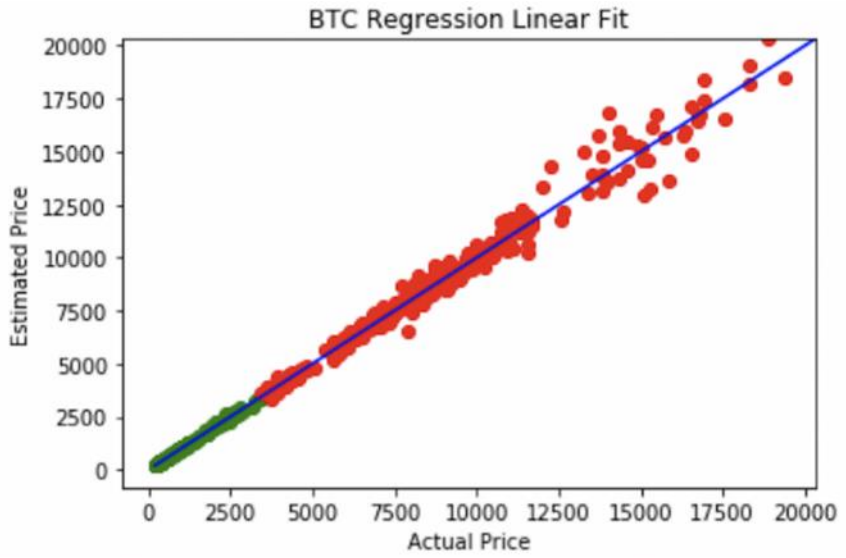

Figure 8. Bitcoin regression fit shown as estimated price on the y-axis and actual price on the x-axis. Green dots are training data. Red dots are testing results.

Finally, the challenges that this paper introduced are using more complex models to examine if there would be further improved results, and checking if these findings are still present in varying pricing situations.

3.2 Short-Term Bitcoin Price Fluctuation Prediction Using Social Media and Web Search Data

In 2019, Aditi Mittal, Vipasha Dhiman, Ashi Singh and Chandra Prakash from the Department of Information Technology of the Indira Gandhi Delhi Technical University for Women in India, published a paper about "Short-Term Bitcoin Price Fluctuation Prediction Using Social Media and Web Search Data" [59]. Their purpose was to discover a relationship between Bitcoin price and Twitter and Google search data, and predict price fluctuations. Bitcoin was the most prosperous cryptocurrency with most of the 4 million cryptocurrency users using it, back in 2017, either as a currency, or as investment. They alleged that with the use of machine learning they would be able to predict the fluctuations of Bitcoin prices and help people do the correct moves for their investments.

In order to achieve this, they tried various machine learning approaches, such as Linear Regression, Polynomial Regression, Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM).

First of all, Bitcoin data were gathered from bitcoincharts for the period from 9 April 2014 until 7 January 2019. The dataset that was created consisted of three features, Timestamp, Weighted Price and Volume of Transactions. Every row contained data for

every minute, and after the appropriate calculations on the price, they finally had the average value for each day.

Afterwards tweets were retrieved by using Tweepy, a Python library, to gain access to Twitter API, from 16 December 2017 until 20 February 2019. The selected search term was the hashtag "#bitcoin", while retweets and non-English tweets were excluded. At the end of the tweet extraction, the dataset was made up of approximately 7.5 million records described by the ID, the Timestamp and the Text. At this point, the dataset needed some preprocessing tasks, such as similar patterns discovery, and removal of hashtags, links, emoticons and capital letters, which were accomplished through Python's preprocessing packages and regular expressions. Moreover, the tweets were classified as positive with a compound score larger than 0.05, negative with a score less than -0.05, or neutral otherwise, based on VADER.

Next, the authors collected the number of tweets concerning Bitcoin posted daily, through www.bitinfocharts.com. So, the Tweet Volume's dataset contained 1735 records referring to the same time period as the collected tweets.

The final step of the data collection was to gather Google Trend Data for the same period. Google makes search data available in the form of Search Volume Index (SVI), but not for a period longer than three months. In such a case, the data returned were in a monthly basis and some further processing was required based on Erik Johansson approach. First of all, the SVI data were collected in 90-day segments and then added all together. Secondly, the monthly SVI was computed by lining up the data for the same period, aggregated at a monthly level. Then, an adjustment factor was defined for the overlapping days, with the division of the monthly with the daily SVI. Finally, the adjustment factor was multiplied with the daily SVI.

With respect to the machine learning approaches, as already mentioned, various algorithms were applied for the purpose of discovering a relationship between Bitcoin price and Twitter and Google search data.

The authors first applied Linear Regression (as shown in 1) to model the linear relationship between Bitcoin price and Tweet sentiment, Google Trends, and Tweet Volume, and make predictions. 70% of the dataset was used for training and the remaining 30% for testing the model.

$$Y = A0 + A1x \quad (1)$$

Table 1 displays the R2 Score and the Pearson R correlation coefficient for the three inputs of the model. Based on these results, it turned out that Google Trends SVI and Tweet Volume are strongly correlated with Bitcoin price, with Pearson R-value as 0.79 and 0.74, and R2 Score as 0.755 and 0.69 respectively. On the contrary, it was proven that Tweet Sentiments have a poor relationship with Bitcoin price, with Pearson R-value as -0.3 and R2 Score as 0.049, by applying VADER sentiment analysis to get the sentiments polarity, and cross validation with 70% training and 30% testing sets on the dataset with the tweets' daily average scores. The outcome of the sentiments linear regression is represented in Figure 9.

Table 1. Linear Regression Results.

Parameter	R2 Score	Correlation Coefficient
Tweet Volume	0.690	0.740
Google Trends	0.755	0.790
Tweet Sentiments	0.049	-0.300

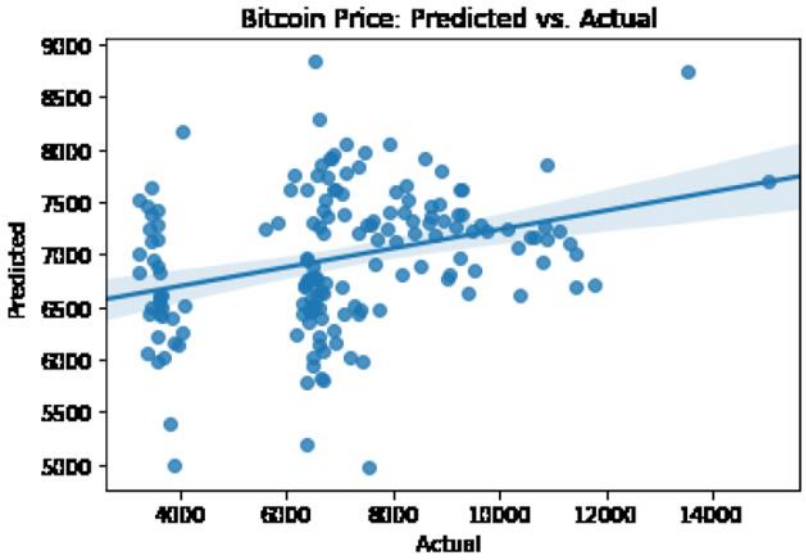

Figure 9. Linear Regression Results for Tweets Average Polarity.

The second algorithm the writers used was Polynomial Regression, to fit a non-linear model to the data. While modeling and comparing the best-fit curve of various degrees, it was noticed that there was a better R2 Score at higher degree, until the stage where it began reducing again, as shown in Figure 10 and Figure 11.

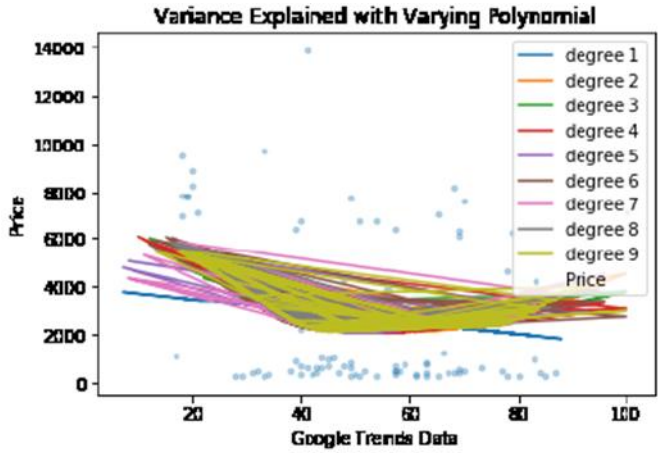

Figure 10. Polynomial regression with Price vs Google trends.

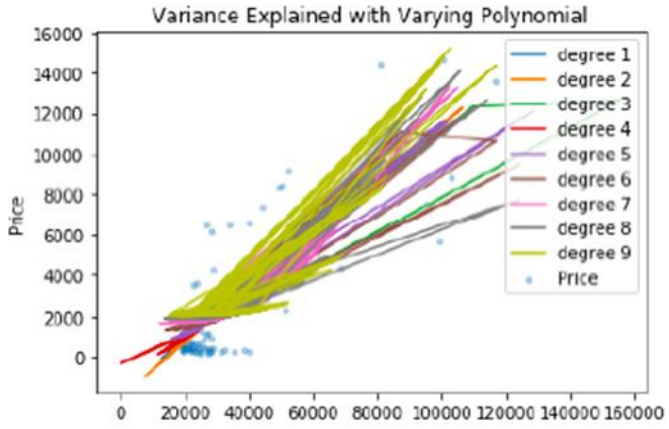

Figure 11. Polynomial regression with Price vs Tweet volume.

The outcomes of the Bitcoin price prediction based on Google search data and Tweet Volume are represented in Table 2. It can be observed that Tweet Volume gave more accurate results than Google Trends, with an overall accuracy of 77.01% instead of 66.66%.

Table 2. Polynomial Regression results.

Measure	Google Trends	Tweet Volume
Accuracy	66.66%	77.01%
500$ margin acc.	6.89%	10.34%
100$ margin acc.	3.44%	1.14%
50$ margin acc.	2.29%	1.14%
25$ margin acc.	2.29%	1.14%

What the authors did next, is they applied Recurrent Neural Network and Long Short Term Memory on Tweet Volume and Google search data. 70% of the data was used in order to train the model and the remaining 30% to test it. Various parameters were used

for the two algorithms, so they received different results. The best-performed parameters with respect to the accuracy, are presented in Table 3, while Table 4 shows the outcomes of Bitcoin price prediction based on Google search data and Tweet Volume using RNN and LSTM.

Table 3. RNN and LSTM parameters used.

	RNN		LSTM	
	Trends	Volume	Trends	Volume
Activation Function	sigmoid	sigmoid	sigmoid	sigmoid
Units	5	4	6	4
Learning Rate	0.001	0.01	0.01	0.01
Epochs	70	60	80	70
Batch size	5	5	4	6

Table 4. RNN and LSTM results for Google Trends and Tweet Volume.

	RNN		LSTM	
Measure	Trends	Volume	Trends	Volume
Accuracy	62.45%	53.46%	50.00%	49.89%
500$ margin acc.	83.33%	80.80%	82.62%	80.93%
100$ margin acc.	33.75%	32.7%	37.71%	30.50%
50$ margin acc.	18.14%	17.51%	22.24%	15.46%
25$ margin acc.	8.86%	8.5%	11.86%	6.35%

Moreover, it can be observed that Bitcoin's predicted price, in both cases, using Google Trends, and Tweet Volume, accord with the actual price in the biggest part of the time period, except of some periods with very high prices. This can also be seen in Figure 12 and Figure 13.

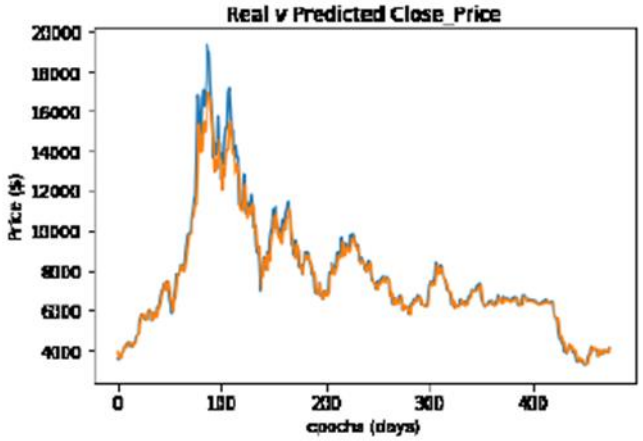

Figure 12. Actual and Price of Bitcoin predicted using Google Trends & RNN.

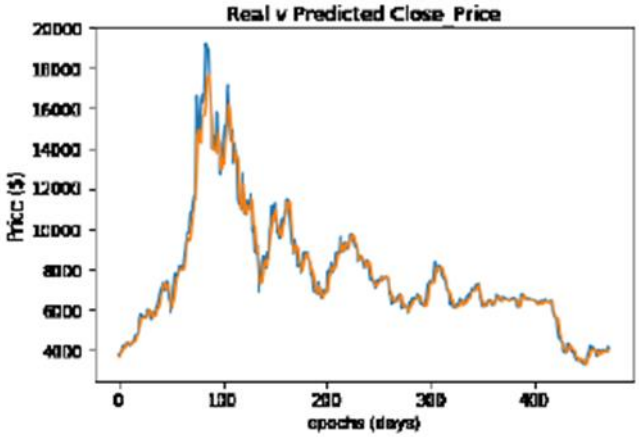

Figure 13. Actual and Price of Bitcoin predicted using Tweet Volume & LSTM.

Finally, RNN, LSTM and ARIMA models were applied on daily, weekly, monthly and annual data using only previous Bitcoin prices. It turned out that Recurrent Neural Network and Long Short Term Memory managed to accurately predict if there would be an increase or decrease in price, by 43.78% and 42.98% respectively, with respect to daily data. The worst predictions were performed by ARIMA, with an accuracy of 38.02%. Table 5 shows the comparison of the three approaches.

Table 5. Comparison of RNN, LSTM and ARIMA on daily data.

Measures	RNN	LSTM	ARIMA
Accuracy	43.78%	42.98%	38.02%
500$ margin acc.	35.23%	30.60%	28.020%
100$ margin acc.	25.56%	23.98%	21.10%
50$ margin acc.	23.21%	21.43%	12.32%
25$ margin acc.	32.45%	31.75%	20.47%

The conclusion that is drawn from this paper is that Tweet Sentiment demonstrated the worst results. With the use of the machine learning algorithms – Long Short Term Memory, Recurrent Neural Network, Polynomial Regression – on Tweet Volume and Google search data, Polynomial Regression performed the best and predicted Bitcoin price with an accuracy of 77.01% based on Tweet Volume and 66.66% based on Google Trends.

According to the authors, future research should aim at Wikipedia search data and Facebook posts and examine how accurate the results will be. Moreover, the collected data can be added together to examine the overall accuracy, except of investigating them apart from each other.

3.3 Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model

In 2019, Tianyu Ray Li, Anup S. Chamrajnagar, Xander R. Fong and Nicholas R. Rizik from the Department of Mathematics of Dartmouth College in Hanover New Hampshire, and Feng Fu from the Department of Biomedical Data Science of the Geisel School of Medicine at Dartmouth in Lebanon New Hampshire published a paper called "Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model" [3]. Their goal was to study and build a machine learning model in order to predict cryptocurrency price fluctuations. In more details, they were planning to work with alternative cryptocurrencies, a.k.a. alt-coins, and gain beneficial knowledge to generate an attainable arbitrage opportunity for other nascent alt-coins.

After a thorough research, they decided that ZClassic (ZCL) would be the best fit for their study. ZClassic is a private, decentralized, quick, open-source community-driven digital currency, which is ready to entrain a major change to Bitcoin Private's blockchain protocol and make previously invalid blocks and transactions valid. This major change is known as "hard fork", and before this, ZClassic gets separated into two, ZClassic and Bitcoin Private. Older hardforks contained Bitcoin Cash and Bitcoin Private, and the history of each implied that ZClassic's price rises and falls would heavily rely on speculation concerning the future prosperity of Bitcoin Private. As a consequence, real-time tweet analysis works for measuring investor sentiment after, for instance, a positive information about Bitcoin Private. Also, the lower trading rate of ZClassic compared to other alt-coins implies that sentiment-based price fluctuations might have a stronger effect on it.

As for the data collection, RStudio was used, and through the open-sourced rtweet package the authors gained access to Twitter's REST and stream APIs. So, they retrieved tweets containing the terms "ZClassic", "ZCL", and "BTCP", for a period of 3,5 weeks. Moreover, the datasets were joined, and after removing duplicate tweets, the final dataset consisted of 130,000 records.

What the writers did next, was to create an algorithm used for the positive, negative or neutral sentiment classification of the tweets by utilizing natural language processing. Through Python's package "Textblob", every tweet is given a polarity value between -1 and 1, by examining the important words and phrases of it. If the whole tweet phrase has a positive nonzero polarity value, the program considers the sentiment as positive, +1, if

negative nonzero, then negative, -1, and if the polarity value is zero, then the program scores a sentiment value of 0.

Attention needed to be paid on retweets, as they might hold new positive or negative information. This can lead to the chained network effect and expand the dispersion of the original post. In order to avoid this, the authors invented a second sentiment index which received a value of -2 for every negative retweet and +2 for every positive one. Now the retweets are extra weighted as there was the assumption that they were more trustworthy and the investors would pay more attention on them. After that, they also had to sum up the weights of all tweets that were repeated, in order to calculate the hourly values of both weighted and unweighted sentiment indices and make direct comparisons to ZCL price data.

When it comes to model selection, the authors applied 10-fold cross validation on 589 data points to identify the ideal model between linear regression, tree model, and support vector machine regression. It turned out that Extreme Gradient Boosting Regression performed the best with respect to accuracy levels. Then, 70% of the dataset was used for training and the remaining 30% was used for testing the model.

It is important to highlight that the natural language processing classification algorithm performed pretty well and gave remarkably accurate results with regards to correctly recognizing the sentiment of the tweets. However, the authors decided to extra measure the accuracy of their algorithm, by manually classifying 100 random tweets and contrast them with their algorithm's results. In this way, they tried to make certain that they would control the computer-generated bot tweets - which many times have positive content - even though most of them were correctly classified as neutral. The results of the algorithm's versus the manual classification are shown in Table 6. It can be observed that in all three cases the algorithm managed to correctly predict the sentiment in more than 50% of the tweets, while almost 80% and 0% of the positive and negative ones respectively, were effectively characterized as positive.

Table 6. Validation analysis of algorithm sentiment prediction by manual inspection.

		Algorithm sentiment prediction		
		Positive	**Neutral**	**Negative**
Manual sentiment decision	Positive	79%	21%	0%
	Neutral	34%	51%	15%
	Negative	25%	0%	75%

Once the sentiment classification algorithms have been put into effect, the authors chose to use the following features to train the model: Pure Positive Sentiment, Pure Negative Sentiment, Neutral Sentiment, An Unweighted Sentiment Index, A Weighted Sentiment Index, and Hourly Trading Volume. These six features provided the highest and most accurate overall correlation with the testing data (as presented in Table 7), as they were diverse enough and it turned out that they successfully trained the model.

Table 7. 5-factor correlation coefficients between the chosen feature and the price data, respec-tively.

	Correlation coefficients
Trading volume	0.605
Neutral	0.302
Positive	0.250
Negative	0.156
Unweighted index	0.150
Weighted index	0.103

As regards the testing of the model, the produced results followed the actual price movements to a large extent, with a Pearson correlation of 0.806 and statistical significance at $p < 0.0001$. Although they managed to correctly predict the direction of the price movement, they found out that there is a difference of around $30 between the predicted and the actual price, as shown in Figure 14. This can be attributed to the fact that the training and testing sets did not contain equal amount of positive, negative and neutral tweets.

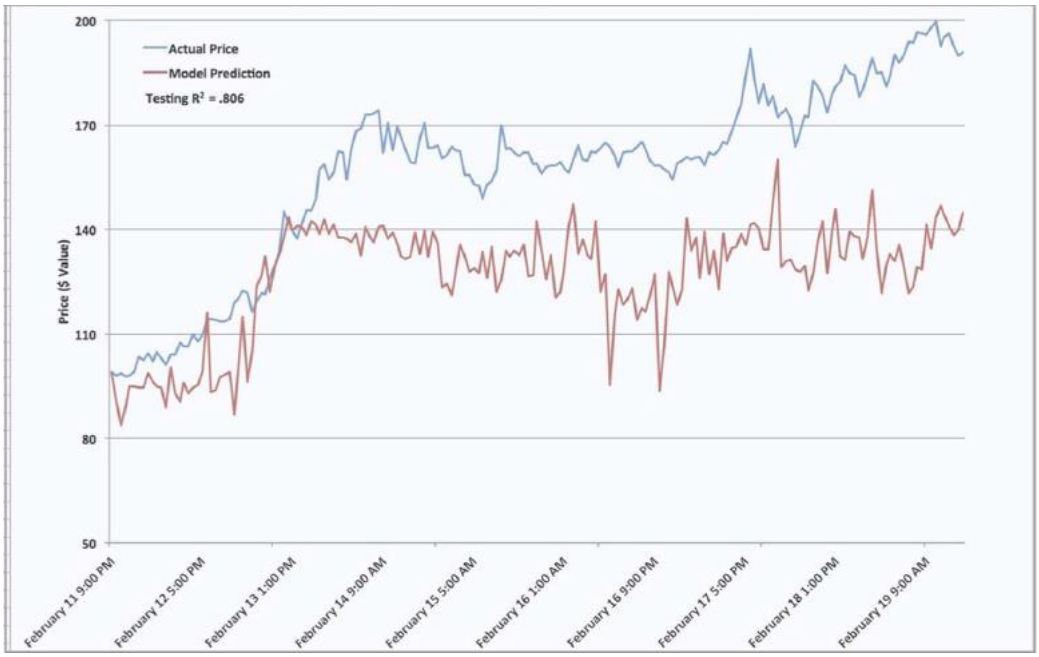

Figure 14. Comparison of model prediction and actual price data. Details the model prediction price data as compared to the testing real price data.

These discrepancies are presented in Table 8. As one may notices, the data that were used for training the model, demonstrated a negative trend, hence the model took into account more the negative records, than the positive ones. It can also be observed that regarding the testing data, there was a decrease of around 3% in the positive records, while there was an increase of around 0.5% in the negative records, and around 15% in the average hourly tweets. These fluctuations caused the model to produce a little lower predictions compared to the actual data. The correlation remained strong, hence the model would have achieved better predictions if it was trained also on data with positive trend.

Table 8. Discrepancies of Twitter sentiments between testing and training data.

	Positive	Negative	Neutral	Total
Average hourly tweets in training data	73.7	18.8	75.2	167.7
Percentage of total	43.9%	11.2%	44.8%	–
Average hourly tweets in testing data	82.4	23.5	95.5	201.4
Percentage of total	40.9%	11.7%	47.4%	–

The authors of this paper concluded that an Extreme Gradient Boosting Regression Tree Model is a good tool for forecasting price changes within the ZClassic cryptocurrency market, based on the analysis of Twitter sentiment and trading volume. Their model suggests that twitter is a good source to analyze data and utilize them to get a first idea about future price changes in altcoins. They found out that the algorithm's weak point is to correctly classify negative tweets, and this is attributed to the fact that the algorithm is not able to recognize sarcasm in the tweets.

The future challenges that the authors introduced are examining if the predictions could be further improved by exploiting Google Search data, Wikipedia queries, or content that is being posted on Facebook and Reddit. Also, the prediction's accuracy might be boosted by improving the algorithm so that it recognizes sarcasm in the content of the post. One last point they introduced is gathering data for a longer period and using them for testing and training the model, as they could only collect data for 3.5 weeks because of Twitter's limitations.

4 Implementation

This chapter is going to present the data that were collected and used, as well as the methodologies that were followed.

4.1 Data Gathering

Social media platforms have become a necessary daily tool for the communication among millions, or even billions, of users locally and worldwide. Possibly the most used social media tool is social networking sites, such as Twitter, Facebook, Tumblr. Most of this kind of websites are free and accessible to everyone to be used to express themselves by sharing their thoughts and opinions on various topics. This means that people share their emotions on social media, that's why they have become a valuable source of data, available for social or marketing analyses. [20], [60]

4.1.1 Tweets Gathering

Twitter data were collected first with the help of python libraries, as explained below.

4.1.1.1 Twint Library

What was first implemented is tweets collection by using Python's library, Twint. 2.1.21 version of the library was installed through Jupyter Notebook and used with the following configurations in order to retrieve every English tweet that contained either the word "bitcoin" or the hashtag "#bitcoin", and for a specific period of days each time. The data were saved as csv files in the working directory. This procedure was repeated for both ethereum and dogecoin.

However, after some period of scrapping, it was observed that it did not perform as expected and was not fetching all the tweets of a day, but sometimes it stopped at different times, before the end of the day. Therefore, a combination of Twint and Tweepy libraries was followed for the remaining time period, in order to maintain higher credibility.

4.1.1.2 Tweepy Library

Tweepy 3.10.0 was used along with OAuth 1a authentication in PyCharm 2021.2 (Edu), as shown in the figure below. An OAuthHandler instance was created, into which the

authentication details were passed. Consumer keys and access tokens are confidential, hence they have been blurred for security reasons.

```
# Import library
import tweepy

# Consumer keys and access tokens
consumer_key = 'RMX[redacted]DNM'
consumer_secret = 'YQz[redacted]0eO'
access_token = '134[redacted]KuA'
access_secret = 'Mzu[redacted]d7t'

# OAuth authentication using the keys and tokens
authentication = tweepy.OAuthHandler(consumer_key, consumer_secret)
authentication.set_access_token(access_token, access_secret)
```

Figure 15. Twitter API authentication using Tweepy OAuth 1a.

In order to moderate Twitter's API limitation on the number of requests, two additional parameters for the *tweepy.API* class have been set to *True*. The first parameter is "wait_on_rate_limit", which defines whether or not to automatically wait for rate limits to replenish, while the second one, "wait_on_rate_limit_notify" enables printing a notification when Tweepy is waiting for rate limits to replenish [61]. So, the program halts for some seconds and then it automatically continues scrapping tweets.

4.1.2 Tweets Volume

Twitter's API offers access to Twitter for tweets extraction with a limited amount of requests, as already mentioned. Besides that, the collected tweets are a random sample of all the tweets that have been posted, comprising large datasets, but it does not allow extraction of the number of the total tweets for a specific day.

Therefore, one good source to scrape tweets volume is www.bitinfocharts.com, which provides cryptocurrency statistics for free. In order to extract data from HTML file, Python library *BeautifulSoup* was required. The data about tweets volume were retrieved from 09-04-2014 until 15-12-2021 for Bitcoin and Dogecoin, and from 16-03-2016 until 15-12-2021 for Ethereum.

4.1.3 Data Storing

Python does also have the ability to connect with *MySQL* databases using *MySQL* driver called *MySQL Connector* [62]. After installing and importing *mysql-connector* 2.2.9 version, the database was managed through *PHPMyAdmin*. The database that was

constructed under the name *social_media*, contained tables with daily tweets retrieved with the search word "bitcoin" or "#bitcoin", "ethereum" or "#ethereum", and "dogecoin" or "#dogecoin", from the 11th of November 2021 until the 15th of December 2021. The tables were manually created by running on phpMyAdmin's SQL terminal the corresponding query specifying the appropriate columns, each time before the execution of the Python code for the tweets extraction.

Each table then held information about various entities of the content posted on Twitter, such as the tweet's id, text, number of retweets, the name of the user who posted the tweet, and more. The structure of the database but also of the table are depicted in Figure 16, Figure 17, and Figure 18. Finally each table was exported as csv in order to be in the appropirate format for further processing.

Table	Action	Rows	Type	Collation	Size	Overhead
btc11-09	Browse Structure Search Insert Empty Drop	5,294	InnoDB	latin1_swedish_ci	6.5 MiB	-
btc11-09(#)	Browse Structure Search Insert Empty Drop	41,431	InnoDB	latin1_swedish_ci	44.6 MiB	-
btc11-10	Browse Structure Search Insert Empty Drop	18,027	InnoDB	latin1_swedish_ci	18.5 MiB	-
btc11-10(#)	Browse Structure Search Insert Empty Drop	11,672	InnoDB	latin1_swedish_ci	12.5 MiB	-
btc11-10(all)	Browse Structure Search Insert Empty Drop	18,028	InnoDB	latin1_swedish_ci	17.5 MiB	-
btc11-11	Browse Structure Search Insert Empty Drop	17,757	InnoDB	latin1_swedish_ci	18.6 MiB	-
btc11-11(#)	Browse Structure Search Insert Empty Drop	13,351	InnoDB	latin1_swedish_ci	14.5 MiB	-
btc11-12	Browse Structure Search Insert Empty Drop	6,975	InnoDB	latin1_swedish_ci	7.5 MiB	-
btc11-12(#)	Browse Structure Search Insert Empty Drop	4,331	InnoDB	latin1_swedish_ci	5.5 MiB	-
btc11-12(all)	Browse Structure Search Insert Empty Drop	6,976	InnoDB	latin1_swedish_ci	8.5 MiB	-
btc11-13(#)	Browse Structure Search Insert Empty Drop	~88,687	InnoDB	latin1_swedish_ci	106.7 MiB	-
btc11-14	Browse Structure Search Insert Empty Drop	~59,268	InnoDB	latin1_swedish_ci	71.6 MiB	-

Figure 16. Screeshot of MySQL database's structure.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None			Change Drop More
2	search_val	varchar(60)	latin1_swedish_ci		No	None			Change Drop More
3	created_at	text	latin1_swedish_ci		Yes	NULL			Change Drop More
4	tweet_id	text	latin1_swedish_ci		Yes	NULL			Change Drop More
5	text	text	latin1_swedish_ci		Yes	NULL			Change Drop More
6	source	text	latin1_swedish_ci		Yes	NULL			Change Drop More
7	in_reply_to_status_id	text	latin1_swedish_ci		Yes	NULL			Change Drop More
8	in_reply_to_user_id	text	latin1_swedish_ci		Yes	NULL			Change Drop More
9	in_reply_to_screen_name	text	latin1_swedish_ci		Yes	NULL			Change Drop More
10	user_name	text	latin1_swedish_ci		Yes	NULL			Change Drop More
11	user_screen_name	text	latin1_swedish_ci		Yes	NULL			Change Drop More
12	user_location	text	latin1_swedish_ci		Yes	NULL			Change Drop More
13	user_url	text	latin1_swedish_ci		Yes	NULL			Change Drop More
14	user_description	text	latin1_swedish_ci		Yes	NULL			Change Drop More
15	user_verified	tinyint(1)			Yes	NULL			Change Drop More
16	user_followers_count	int(11)			Yes	NULL			Change Drop More
17	user_friends_count	int(11)			Yes	NULL			Change Drop More
18	user_listed_count	int(11)			Yes	NULL			Change Drop More
19	user_favourites_count	int(11)			Yes	NULL			Change Drop More
20	user_created_at	text	latin1_swedish_ci		Yes	NULL			Change Drop More
21	user_id	varchar(50)	latin1_swedish_ci		Yes	NULL			Change Drop More

Figure 17. Screenshot of the tables' structure.

22	user_profile_image_url_https	text	latin1_swedish_ci	Yes	NULL	Change Drop More
23	coordinates_lat	text	latin1_swedish_ci	Yes	NULL	Change Drop More
24	coordinates_lon	text	latin1_swedish_ci	Yes	NULL	Change Drop More
25	place_country	text	latin1_swedish_ci	Yes	NULL	Change Drop More
26	place_country_code	text	latin1_swedish_ci	Yes	NULL	Change Drop More
27	place_full_name	text	latin1_swedish_ci	Yes	NULL	Change Drop More
28	place_id	varchar(50)	latin1_swedish_ci	Yes	NULL	Change Drop More
29	place_type	varchar(30)	latin1_swedish_ci	Yes	NULL	Change Drop More
30	quoted_status_id	varchar(50)	latin1_swedish_ci	Yes	NULL	Change Drop More
31	quoted_status	text	latin1_swedish_ci	Yes	NULL	Change Drop More
32	retweeted_status	text	latin1_swedish_ci	Yes	NULL	Change Drop More
33	quote_count	int(11)		Yes	NULL	Change Drop More
34	reply_count	int(11)		Yes	NULL	Change Drop More
35	retweet_count	int(11)		Yes	NULL	Change Drop More
36	favorite_count	int(11)		Yes	NULL	Change Drop More
37	retweeted	tinyint(1)		Yes	NULL	Change Drop More
38	entities	text	latin1_swedish_ci	Yes	NULL	Change Drop More
39	lang	varchar(30)	latin1_swedish_ci	Yes	NULL	Change Drop More
40	feeds_link	text	latin1_swedish_ci	Yes	NULL	Change Drop More
41	feeds_video	text	latin1_swedish_ci	Yes	NULL	Change Drop More
42	feeds_image	text	latin1_swedish_ci	Yes	NULL	Change Drop More
43	created_date	datetime		Yes	current_timestamp()	Change Drop More

Figure 18. Screenshot of the tables' structure

On the other hand, as already mentioned, the storage of the data retrieved with twint was simpler, as they were directly exported to csv files. Similarly, the tables contained metadata and additional corresponding information about the collected tweets. The same applies for tweets volume data, which were saved into csv files for all three cryptocurrencies.

Finally, when the tweets retrieval was completed in the 15th of December, all the csv files where merged and formed the final dataset for each cryptocurrency. This was achieved through python, by importing the corresponding csv files into variables and concatenating these variables with *pandas* object *concat*.

4.1.4 Cryptocurrencies Prices

There are numerous reliable sources to obtain historical market data of cryptocurrencies through Python, with or without the use of crypto APIs. Some of the best sources are *Alpha Vantage*, *Pandas DataReader*, *Yfinance* and *Cryptocompare*, which are easy to use with good documentation. Pandas DataReader version 0.10.0 was used, which is a sub-package of Pandas, and retrieves historical data from the *Yahoo Finance API* [63]. The extracted information refer to:

- the *High* and *Low Prices*, which are the highest and lowest prices of the day respectively [64],
- the *Open* and *Close Prices*, which are the first and last prices of the day respectively,
- the *Volume*, which is the total amount of coins traded in the last 24 hours, and
- the *Adjusted Closing Price*, which is the closing price of the day, adjusted for splits and dividend and/or capital gain distributions.

The console output of the cryptocurrencies' historical data extraction is presented below.

Date	High	Low	Open	Close	Volume	Adj Close
2021-09-15	48450.468750	46773.328125	47098.000000	48176.347656	30484496466	48176.347656
2021-09-16	48486.828125	47079.558594	48158.906250	47783.359375	31764293754	47783.359375
2021-09-17	48160.921875	46832.523438	47771.003906	47267.519531	28727713711	47267.519531
2021-09-18	48791.781250	47087.285156	47273.527344	48278.363281	28575630451	48278.363281
2021-09-19	48328.367188	46919.804688	48268.855469	47260.218750	26967722648	47260.218750

Figure 19. Output of Bitcoin's historical data extraction from Yahoo Finance API.

4.2 Sentiment Analysis

After having all the necessary data gathered, the first main task of the project can take place, which is the Sentiment Analysis. Essentially, it is a classification problem that aims in extracting the public opinion of Twitter users and, based on it, predicting the price of Bitcoin, Ethereum and Dogecoin. So, the gathered data get imported, and specifically only those columns of the csv files that are needed for the analysis, which are *Date* and *Tweet*. The console's output is shown below.

	Date	Tweet
0	2021-09-15	Nonprofit Highlight: Rising Tide Montessori wo...
1	2021-09-15	Nonprofit Highlight: Rising Tide Montessori wo...
2	2021-09-15	@DastardlyDaniel @Pickle_cRypto Without Chainl...
3	2021-09-15	@DastardlyDaniel @Pickle_cRypto Without Chainl...
4	2021-09-15	Güncel #bitcoin $btc son 3 mumdur ema144 den r...

Figure 20. Output of Bitcoin's dataset imported for preprocessing.

4.2.1 Text Pre-Processing

The first and most important task of the analysis, is the pre-processing of the data. Especially when it comes to unstructured data like text, text pre-processing is even more crucial. The steps that were applied on "Tweets" column are:

- *Remove Duplicate Tweets*: some users post the same content more than once for various reasons, such as to remind their audience about something or to get attention. Another reason is that it might be the case that the same tweets got retrieved more than once either by accident, or because, twint and tweepy libraries were both used for the retrieval during the second half of the period of examination. Each tweet needs to count only once, so the duplicates get eliminated.
- *Remove URLs*: when users post on social media, they sometimes want to share or refer to another website and, since Twitter offers this possibility, they include the website's URL in their tweets. These links need to be cleared from the tweet's text.
- *Remove Emojis*: the use of emojis on Twitter improves the tweet's engagement and makes it much more noticeable [65]. They get removed to clean the text from any emoticon, symbols and pictographs, transport and map symbols, flags.
- *Remove non-English characters*: the project focuses only on English tweets, so those that contain also other languages characters will only add noise and need to be removed.
- *Remove Mentions and Hashtags*: with typing "*@username*" users can tag each other on social media platforms. People that have been mentioned in somebody else's Tweet can see it in their *Notifications tab* [66]. On the other hand, a hashtag – written with a "#" symbol – is used to categorize tweets and appear more easily in Twitter search. Clicking or tapping on it, shows other posts containing that hashtag [67]. An example of a hashtag could be "#bitcoin". Since the text that is going to be analyzed needs to contain only words that add some information, hashtags and mentions need to be removed.
- *Remove Punctuations*: symbols such as "@, #, %" and other special characters are often used in social media posts and add noise in the dataset, hence they need to be removed.

- *Drop Empty Tweets*: after all these steps of preprocessing, there might be tweets that come up with empty content. These tweets do not add any value in the analysis, thus they get deleted.

After preprocessing, Bitcoin's dataset ended up containing 4,335,906 rows, Ethereum's 1,376,312 rows and Dogecoin's 845,941 rows. A screenshot of Bitcoin's dataset after preprocessing is shown below.

	Date	Tweet
0	2021-09-15	nonprofit highlight rise tide montessori work ...
2	2021-09-15	without chainlink best version defi bitcoin
4	2021-09-15	g ncel btc son mumdur ema den red yiyoruz muml...
6	2021-09-15	cryptoshit sir
8	2021-09-15	none option bitcoin

Figure 21. Output of Bitcoin's dataset after preprocessing.

Then it was time for sentiment analysis. Two different methods were used, as explained below.

4.2.2 TextBlob

The first approach used for the sentiment analysis was *TextBlob*. As already mentioned, it is a sentiment analyzer that uses lexicons with already scored words, and returns two metrics, *Polarity* and *Subjectivity*, defining a negative or positive sentiment, and qualifying the amount of personal opinion respectively [68]. Since the aim was to count the general public opinion regardless of personal opinion, only polarity was used. The latest version of the library was used, which is 0.17.1 and was released on October 22, 2021 [69].

So, TextBlob calculated the polarity for every tweet and assigned a score from -1 to 1 to each one. Based on the scores, the tweets were then classified as *Negative* for negative scores, *Positive* for positive scores, and *Neutral* if the score was zero, as presented below.

	Date	Tweet	Sentiment_Score	Sentiment_Type
0	2021-09-15	nonprofit highlight rise tide montessori work ...	0.0	Neutral
2	2021-09-15	without chainlink best version defi bitcoin	1.0	Positive
4	2021-09-15	g ncel btc son mumdur ema den red yiyoruz muml...	0.0	Neutral
6	2021-09-15	cryptoshit sir	0.0	Neutral
8	2021-09-15	none option bitcoin	0.0	Neutral

Figure 22. Output of Bitcoin's dataset after TextBlob assigned sentiment scores and classified the tweets.

Figure 23 presents the average sentiment score and type of the tweets in daily basis.

Date	Sentiment_Score	Sentiment_Type
2021-09-15	0.063580	Positive
2021-09-16	0.063606	Positive
2021-09-17	0.071595	Positive
2021-09-18	0.066628	Positive
2021-09-19	0.068357	Positive

Figure 23. Output of the average sentiment score and type of the tweets in daily basis for Bitcoin using TextBlob.

Afterwards, it would be interesting to see how many positive, negative and neutral tweets were posted, so the sentiment types were calculated both totally and in daily basis, as shown in Figure 24 and Figure 25.

```
Neutral     2649979
Positive    1258197
Negative     427730
```

Figure 24. Output of the total amount of Positive, Negative, and Neutral tweets for Bitcoin, using TextBlob.

```
Date        Sentiment_Type
2021-09-15  Neutral          4068
            Positive         2075
            Negative          804
2021-09-16  Neutral          3913
            Positive         1944
            Negative          714
2021-09-17  Neutral          3692
            Positive         1868
            Negative          600
```

Figure 25. Output of the daily amount of Positive, Negative, and Neutral tweets for Bitcoin, using TextBlob.

4.2.3 VADER

In order to conduct a better analysis and be able to compare the findings, one more approach was used. As explained in the Background chapter, Valence Aware Dictionary for sEntiment Reasoning is a model based on lexicons of sentiment-related words and returns four scores, the probability a tweet to be positive, negative, neutral, and the compound score. The library's version that was used was again the latest one, 3.6.5, released on October 11, 2021 [70].

Consequently, VADER processed the tweets' text and allocated the corresponding scores. So, every tweet now has two extra features, the probabilities, and the compound score. Since the compound score is essentially the normalized summation of the valence scores, this is the most useful metric of sentiment, therefore this will be used for the analysis. Following that, the tweets will be classified as *Negative*, *Positive*, and *Neutral* based on the compound scores, if they are >= 0.05, or > -0.05 and < 0.05, or <= -0.05, respectively, as shown below.

	Date	Tweet	Sentiment_Score	Sentiment_Type	Score
0	2021-09-15	nonprofit highlight rise tide montessori work ...	-0.2023	Negative	{'neg': 0.234, 'neu': 0.565, 'pos': 0.202, 'co...
2	2021-09-15	without chainlink best version defi bitcoin	-0.5216	Negative	{'neg': 0.402, 'neu': 0.598, 'pos': 0.0, 'comp...
4	2021-09-15	g ncel btc son mumdur ema den red yiyoruz muml...	0.0000	Neutral	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...
6	2021-09-15	cryptoshit sir	0.0000	Neutral	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...
8	2021-09-15	none option bitcoin	0.0000	Neutral	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...

Figure 26. Output of Bitcoin's dataset after VADER assigned sentiment scores and classified the tweets.

Figure 27 presents the average sentiment score and type of the tweets in daily basis.

Date	Sentiment_Score	Sentiment_Type
2021-09-15	0.093003	Positive
2021-09-16	0.110673	Positive
2021-09-17	0.106223	Positive
2021-09-18	0.113747	Positive
2021-09-19	0.106492	Positive

Figure 27. Output of the average sentiment score and type of the tweets in daily basis for Bitcoin using VADER.

Similarly to TextBlob, the amount of positive, negative and neutral tweets were calculated totally, as well as in daily basis, as shown in Figure 28 and Figure 29.

```
: Neutral     2226001
  Positive    1521395
  Negative     588510
```

Figure 28. Output of the total amount of Positive, Negative, and Neutral tweets for Bitcoin, using VADER.

```
Date        Sentiment_Type
2021-09-15  Neutral          3583
            Positive         2305
            Negative         1059
2021-09-16  Neutral          3208
            Positive         2367
```

Figure 29. Output of the daily amount of Positive, Negative, and Neutral tweets for Bitcoin, using VADER.

The same procedure was followed also for the other two cryptocurrencies, Ethereum and Dogecoin.

4.3 Correlation

The main purpose of this project is to examine and find a correlation between the particular cryptocurrencies and the tweets posted about them. Generally, the aim is to ascertain whether the public opinion of social media users has an impact on the price of the cryptocurrencies. Consequently, after the process of sentiment analysis, the next step is to examine the tweets in terms of correlation.

First of all, the csv containing the cryptocurrency's prices was imported and assigned to a variable, which was then merged once with the variable holding TextBlob's sentiment scores, and once with the variable holding VADER's compound scores. Then, *Pearson's Correlation Coefficient* was used, as it is recognized as the best indicator of association between variables [57]. Python's module *Numpy* offers *corrcoef()* method, which took as inputs the sentiment scores and the prices, first for the case of TextBlob and then for VADER. The correlation values are shown in Table 9.

Table 9. Results of the Correlation Analysis between sentiment scores and cryptocurrencies' prices for TextBlob and VADER.

Pearson's Correlation Coefficient		
	TextBlob	**VADER**
Bitcoin	0.4502	0.6035
Ethereum	-0.427	-0.5699
Dogecoin	0.2602	0.2038

As it can be observed, VADER's sentiment scores are more correlated with the cryptocurrencies' prices of Bitcoin and Ethereum, while the opposite goes for Dogecoin. TextBlob provided results that are more correlated with Dogecoin's prices. Therefore, VADER's sentiment scores are going to be used for building machine learning models for Bitcoin's and Ethereum's price predictions, while TextBlob's scores are going to be used in Dogecoin's case.

4.4 Machine Learning

At this point, where the sentiment analysis was completed, it is time for the second main task of the project, the Machine Learning Approach. The final goal was to build a model able to predict the price of the cryptocurrencies of the study.

4.4.1 Sentiment Score and Tweets Volume as Inputs

The first attempt was to build a model taking as inputs the sentiment scores and the volume of the tweets posted in the period from September 15, 2021, until December 15, 2021, and predicting the cryptocurrency's average price.

To start with, the three csv files containing the sentiment scores, the tweets volume, and the cryptocurrency's prices, along with the volume of transactions, were imported and assigned to variables, which afterwards were merged. *Daily Return* was also computed, which measures the price's change as a percentage of the previous day. The final dataset consists of six columns, as presented in Figure 30.

	Date	Sentiment_Score	Tweets_Volume	Transactions_Volume	Daily_Return	Avg_Price
0	2021-09-15	0.093003	109407.0	3.048450e+10	0.035374	47611.898438
1	2021-09-16	0.110673	90855.0	3.176429e+10	0.003598	47783.193359
2	2021-09-17	0.106223	101046.0	2.872771e+10	-0.005995	47496.722656
3	2021-09-18	0.113747	82649.0	2.857563e+10	0.009323	47939.533203
4	2021-09-19	0.106492	74123.0	2.696772e+10	-0.006580	47624.085938

Figure 30. Output of Bitcoin's dataset containing sentiment scores, tweets volume and prices for the 3-month period.

Before starting constructing the model, a correlation analysis, as well as a hypothesis test were conducted, in order to examine if there is an association between the candidate features and the price. From the hypothesis test, it was discovered that the p-value is close to 0 in all cases, which means that the correlation between the features and the cryptocurrencies' prices is statistically significant. As it can be observed in the table Table 10, *Sentiment Score* is strongly correlated and *Tweets Volume* has a medium correlation with Bitcoin's price. When it comes to Ethereum, *Tweets Volume* has a strong positive correlation with the price, while *Sentiment Score* has a strong negative correlation, meaning that when the sentiment score increases, the price decreases, and vice versa. On the other hand, *Sentiment Score* is weakly correlated with Dogecoin's price, while *Tweets Volume* and *Transactions Volume* have a medium correlation. Consequently, Bitcoin's and Ethereum's models are going to use *Sentiment Score* and *Tweets Volume* as inputs, while *Sentiment Score* has to be excluded from Dogecoin's model, and use *Transactions Volume* instead.

Table 10. Results of the Correlation Analysis between sentiment scores and cryptocurrencies' prices, and tweets volume and cryptocurrencies' prices.

	Correlation Coefficient		
Features	**Bitcoin**	**Ethereum**	**Dogecoin**
Sentiment_Score	0.6	-0.57	0.26
Tweets_Volume	0.3	0.73	0.37
Transactions_Volume	0.12	0.01	0.34
Daily_Return	0.08	0.09	0.2

After discovering the correlations between the features and the price for every cryptocurrency, it is time for the machine learning models. For all three cryptocurrencies, three

models were built. Firstly, only sentiment scores were used as input, then only tweets volume, and finally both features -with the only difference transactions volume instead of sentiment score for Dogecoin-. Because of the fact that the values of the features and the label were of different ranges, *MinMaxScaler* was used in order to normalize them between 0 and 1. For every model, *Linear Regression*, *XGBoost*, *Multilayer Perceptron*, and *Support Vector Regression* were used to compare their performance. 70% of the data were used for training, while the remaining 30% for testing the model. However, the size of the dataset (92 records) was not adequate for the model's training.

4.4.2 Tweets Volume and Volume of Transactions as Inputs

Due to the small size of the dataset containing the sentiment scores, an alternative should be found and used for training the models. Therefore, the idea was to use tweets volume and volume of transactions for a longer period, and particularly since October 15, 2014 for Bitcoin and Dogecoin, and since April 15, 2016 for Ethereum.

Like previously, the two csv files containing the tweets volume, and the cryptocurrency's prices, along with the volume of transactions, were imported and assigned to variables, which afterwards were merged. The final dataset consists of five columns, as presented in Figure 31.

	Date	Tweets_Volume	Transactions_Volume	Daily_Return	Avg_Price
189	2014-10-15	23992.0	25267100.0	-0.014980	395.496490
190	2014-10-16	24445.0	26990000.0	-0.024167	385.938507
191	2014-10-17	21491.0	13600700.0	-0.014264	380.433502
192	2014-10-18	20572.0	11416800.0	0.017430	387.064499
193	2014-10-19	18259.0	5914570.0	0.008096	390.197998

Figure 31. Output of Bitcoin's dataset containing tweets volume, volume of transactions and prices for the long period.

The hypothesis test produced p-values close to 0 for all cases, while the results of the correlation analysis are presented in Table 11.

Table 11. Results of the Correlation Analysis between tweets volume and cryptocurrencies' prices, and volume of transactions and cryptocurrencies' prices.

	Correlation Coefficient		
Features	**Bitcoin**	**Ethereum**	**Dogecoin**
Tweets_Volume	0.79	0.79	0.55
Transactions_Volume	0.74	0.65	0.62
Daily_Return	0.012	0.01	0.024

It can be observed that, for all three cryptocurrencies, *Tweets Volume and Transactions Volume* are strongly correlated with their prices. On the other hand, *Daily Return* is weakly correlated with the cryptocurrencies' prices, hence it will not be used for training the model.

Consequently, the first model is fed only with tweets volume, the second only with volume of transactions, and the third with both features. In all three cases the features and the label were normalized between 0 and 1, and the dataset was split to 70-30 training-testing sets.

4.4.3 Ensemble Model

The last attempt was to build an *Ensemble Model* in order to examine if the overall performance could be improved. To achieve this, the three models that had the best performance were combined. Then, the ensemble technique used is *Weighted Averaging*, where the models are being assigned weights according to their performance [71]. For instance, in the case of Bitcoin, Multilayer Perceptron had the least Mean Absolute Error (MAE) so it is given the highest weight, then XGBoost, and finally Support Vector Regression, which performed the worst of these three models. Similarly, Multilayer Perceptron, Support Vector Regression and XGBoost were used for Ethereum's ensemble model, while Support Vector Regression, Multilayer Perceptron and XGBoost were the best-performed algorithms for Dogecoin. In order to assign the optimal weights to the algorithms, a trial-and-error method was followed and various weights have been examined.

5 Results

In this section, the results of the two main tasks of the project are going to be presented. In more details, the results of the sentiment analysis for Bitcoin ,Ethereum and Dogecoin, as well as those of the best-performed machine learning algorithms.

5.1 Sentiment Analysis

To start with, by analyzing 4,335,906 rows for Bitcoin, 1,376,312 rows for Ethereum, and 845,941 rows for Dogecoin, referring to the period from 15-09-2021 until 15-12-2021, it turned out that VADER's method performed better for Bitcoin and Ethereum, in terms of correlation with the cryptocurrencies' prices, while TextBlob achieved better results for Dogecoin. In all three cases, most of the tweets were, neutral, then positive and a smaller portion was negative. This can also be seen in Figure 32 for Bitcoin, Figure 33 for Ethereum and in Figure 34 for Dogecoin.

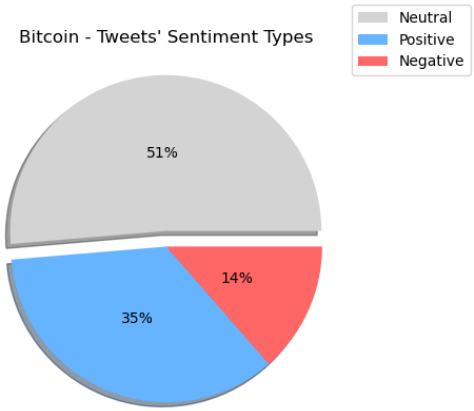

Figure 32. Distribution of Positive, Negative, and Neutral tweets for Bitcoin.

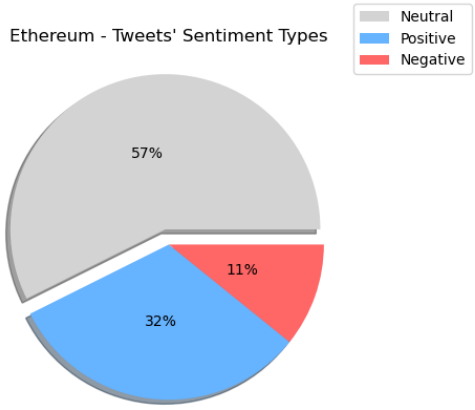

Figure 33. Distribution of Positive, Negative, and Neutral tweets for Ethereum.

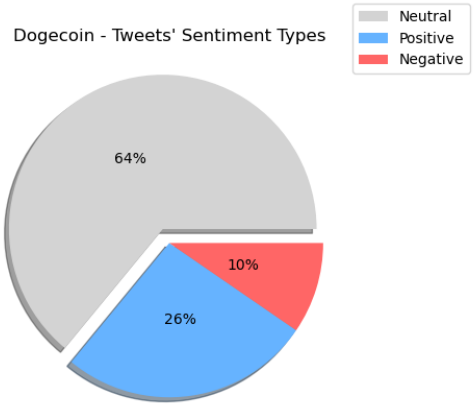

Figure 34. Distribution of Positive, Negative, and Neutral tweets for Dogecoin.

The following graphs present the fluctuations of the sentiment score along with the price over time. Figure 35 refers to Bitcoin, Figure 36 to Ethereum, and Figure 37 to Dogecoin respectively.

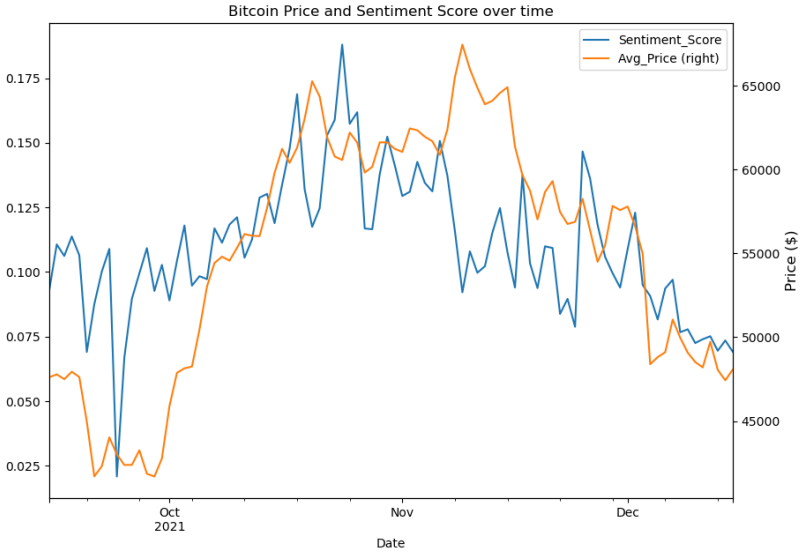

Figure 35. Fluctuations of the sentiment score along with the price over time for Bitcoin.

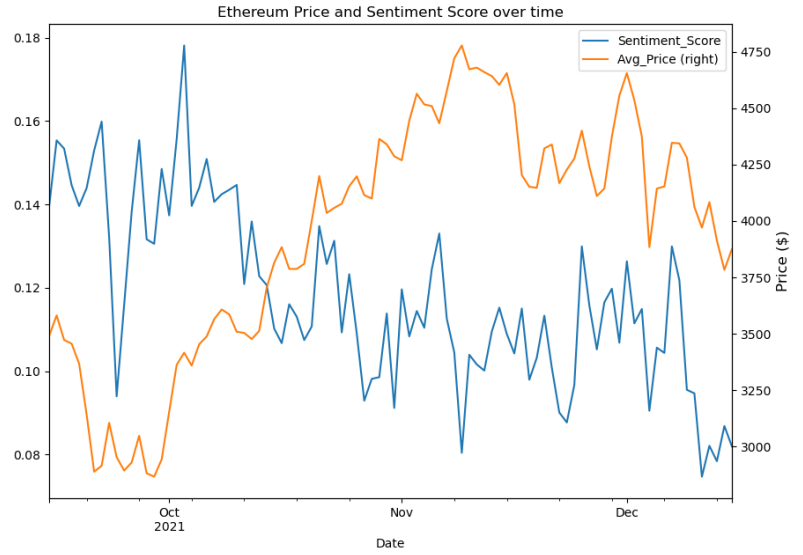

Figure 36. Fluctuations of the sentiment score along with the price over time for Ethereum.

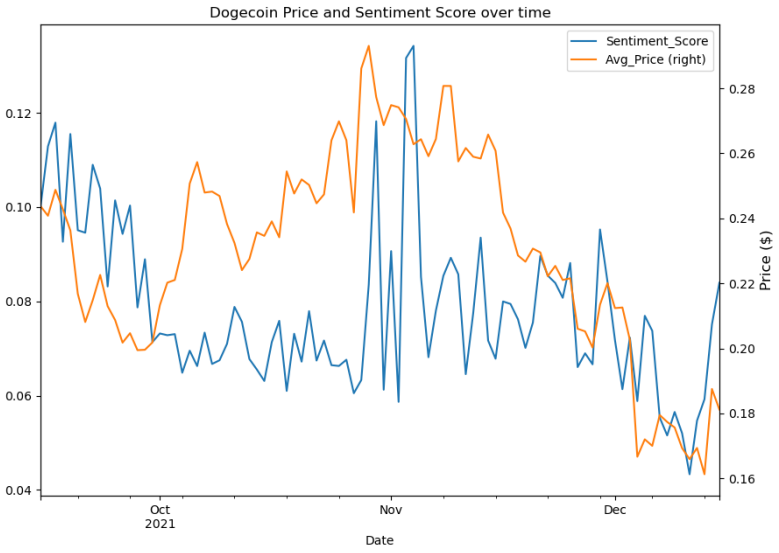

Figure 37. Fluctuations of the sentiment score along with the price over time for Dogecoin.

By observing the above graphs, the correlations that were calculated previously are visible, and especially the negative one of Ethereum's sentiment score and price.

The graph below depicts the trajectory of the sentiment scores of all three cryptocurrencies during the three months period.

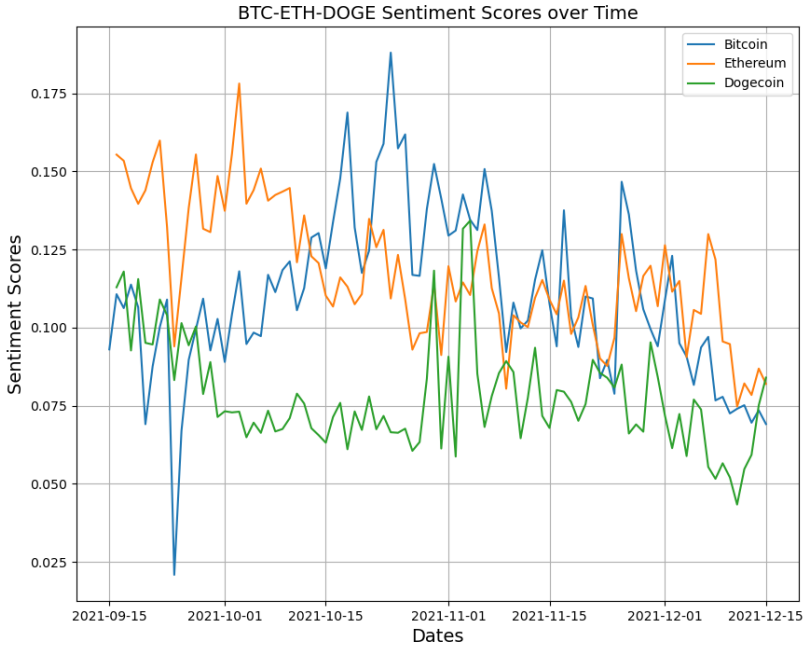

Figure 38. the trajectory of the sentiment scores of all three cryptocurrencies during the three months period.

It can be noted that there is a correlation between the three lines, as there are similar fluctuations in some specific time instants.

5.2 Machine Learning

5.2.1 Sentiment Score and Tweets Volume as Inputs

As described in the previous section, the first attempt was to build a model taking as inputs the sentiment scores and the volume of the tweets posted in the period from September 15, 2021, until December 15, 2021, and predicting the cryptocurrency's average price. However, the correlation analysis proved that sentiment score is weakly correlated with Dogecoin's price, so it stayed out of Dogecoin's model and transactions volume was used instead.

The results for Bitcoin, Ethereum and Dogecoin can be seen in Table 12, Table 13 and Table 14 respectively.

Table 12. Results of Bitcoin's price prediction by using sentiment score and tweets volume.

Evaluation Metrics	Sentiment_Score	Tweets_Volume	Sentiment_Score + Tweets_Volume
	Linear Regression		
R2	19.6%	18.6%	43.4%
MAE	20.3%	20.8%	17.4%
MSE	6.2%	6.2%	4.3%
	XGBoost		
R2	-0.6%	-0.6%	-0.6%
MAE	23.9%	23.9%	23.9%
MSE	7.7%	7.7%	7.7%
	Multilayer Perceptron		
R2	21.9%	7.5%	28.8%
MAE	19.4%	22.3%	17.5%
MSE	6%	7.1%	5.4%
	Support Vector Regression		
R2	38.9%	19.8%	65.2%
MAE	17%	20.5%	12.2%
MSE	4.8%	6.2%	2.7%

Table 13. Results of Ethereum's price prediction by using sentiment score and tweets volume.

Evaluation Metrics	Sentiment_Score	Tweets_Volume	Sentiment_Score + Tweets_Volume
	Linear Regression		
R2	40.6%	15.9%	28.6%
MAE	17.6%	18.9%	17.4%
MSE	4.6%	6.5%	5.6%
	XGBoost		
R2	-1.7%	-1.7%	-1.7%
MAE	24.5%	24.5%	24.5%
MSE	7.9%	7.9%	7.9%
	Multilayer Perceptron		
R2	21.1%	46.8%	64.5%
MAE	18.3%	14.8%	12.2%
MSE	6.1%	4.1%	2.8%
	Support Vector Regression		
R2	46.9%	63.7%	78.4%
MAE	15.3%	12.7%	10%
MSE	4.1%	2.8%	1.7%

Table 14. Results of Dogecoin's price prediction by using sentiment score and tweets volume.

Evaluation Metrics	Tweets_Volume	Transactions_Volume	Tweets_Volume + Transactions_Volume
	Linear Regression		
R2	19.8%	18.7%	21.3%
MAE	14.7%	15.3%	14.7%
MSE	3.9%	4%	3.8%
	XGBoost		
R2	-2.2%	-2.2%	-2.2%
MAE	17%	17%	17%
MSE	5%	5%	5%
	Multilayer Perceptron		
R2	7%	-37.5%	16.4%
MAE	15.9%	20.3%	14.7%
MSE	4.5%	6.7%	4.1%
	Support Vector Regression		
R2	29.6%	21.6%	41.7%
MAE	15.4%	16.5%	14.1%
MSE	4%	4.5%	3.3%

By observing the above tables, it can be deduced that, generally, the algorithms perform better when the models take as inputs both features. The best performance for all three cryptocurrencies was achieved with *Support Vector Regression*, where the data fit the model by 65.2 %, 78.4% and 41.7% respectively. These results have been produced after setting *random_state* parameter to 42 when splitting the data, as the algorithms generated diverse results after every run. This can be attributed to the small size of the dataset (92 records), which was not adequate for the model's training.

5.2.2 Tweets Volume and Volume of Transactions as Inputs

The second attempt was to use tweets volume and volume of transactions for training the models, as an alternative to the first attempt because of the small dataset containing the sentiment scores. The datasets refer to the period from October 15, 2014 until December 15, 2021 for Bitcoin and Dogecoin, and from April 15, 2016 until December 15, 2021 for Ethereum.

Table 15, Table 16 and Table 17 present the performance of the algorithms of the three models for Bitcoin, Ethereum and Dogecoin respectively.

Table 15. Results of Bitcoin's price prediction by using tweets volume and volume of transactions.

Evaluation Metrics	**Tweets_Volume**	**Transactions_Volume**	**Tweets_Volume + Transactions_Volume**
	Linear Regression		
R2	65.6%	60.7%	79.4%
MAE	10%	8.8%	6.4%
MSE	1.9%	2.2%	1.2%
	XGBoost		
R2	73.8%	63.8%	85.3%
MAE	8.3%	7.9%	5.3%
MSE	1.5%	2%	0.8%
	Multilayer Perceptron		
R2	74.8%	64.4%	91.5%
MAE	8.5%	8.1%	3.7%
MSE	1.4%	2%	0.5%
	Support Vector Regression		
R2	68.7%	53.9%	84.3%
MAE	9%	11.1%	7.1%
MSE	1.7%	2.4%	0.8%

Table 16. Results of Ethereum's price prediction by using tweets volume and volume of transactions.

Evaluation Metrics	Tweets_Volume	Transactions_Volume	Tweets_Volume + Transactions_Volume
	Linear Regression		
R2	63.5%	36.5%	74.2%
MAE	8.5%	10.7%	7.2%
MSE	1.7%	3%	1.2%
	XGBoost		
R2	64.9%	47.6%	76.9%
MAE	7.8%	8.9%	5.9%
MSE	1.7%	2.5%	1.1%
	Multilayer Perceptron		
R2	67.9%	45.5%	93.3%
MAE	7.7%	8.5%	3%
MSE	1.5%	2.6%	0.3%
	Support Vector Regression		
R2	60.6%	43.7%	85.8%
MAE	10%	11.2%	6.9%
MSE	1.8%	2.6%	0.7%

Table 17. Results of Dogecoin's price prediction by using tweets volume and volume of transactions.

Evaluation Metrics	Tweets_Volume	Transactions_Volume	Tweets_Volume + Transactions_Volume
	Linear Regression		
R2	41.3%	31.5%	31.5%
MAE	5.6%	5%	5%
MSE	1%	1.2%	1.2%
	XGBoost		
R2	78.8%	78.8%	80.2%
MAE	2.1%	2.3%	2.1%
MSE	0.4%	0.4%	0.4%
	Multilayer Perceptron		
R2	75.9%	78.5%	80.1%
MAE	3.2%	2.5%	2.1%
MSE	0.4%	0.4%	0.4%
	Support Vector Regression		
R2	79.8%	76%	83.6%
MAE	2.2%	3.8%	3.4%
MSE	0.3%	0.4%	0.3%

As in the previous case where sentiment score and tweets volume were used as inputs, the algorithms perform better in all terms of *R2, Mean Absolute Error,* and *Mean Squared Error,* when both tweets volume and transactions volume are considered as inputs for the models,.

The best performance for Bitcoin and Ethereum was achieved with *Multilayer Perceptron*, where the data fit the model by 91.5% and 93.3% respectively, while for Dogecoin it was achieved with *Support Vector Regression* with 83.6% R2 score.

5.2.3 Ensemble Model

Finally, an attempt to improve the overall performance was made with the help of ensemble modeling and weighted averaging. Table 18 presents various weights that have been tried along with the results they produced for Bitcoin, Table 19 for Ethereum, and Table 20 for Dogecoin respectively.

Table 18. Ensemble model results of Bitcoin's price prediction.

	Weights	**R2**	**MAE**	**MSE**
MLP	0.56	91.1%	4.3%	0.5%
XGBoost	0.35			
SVR	0.14			
MLP	0.6	90.9%	4.2%	0.5%
XGBoost	0.41			
SVR	0.09			
MLP	0.77	91.4%	3.8%	0.5%
XGBoost	0.26			
SVR	0.01			

Table 19. Ensemble model results of Ethereum's price prediction.

	Weights	R2	MAE	MSE
MLP	0.5	90.3%	4.2%	0.5%
XGBoost	0.45			
SVR	0.1			
MLP	0.8	91%	3.7%	0.4%
XGBoost	0.3			
SVR	0.01			
MLP	0.73	92.7%	3.4%	0.3%
XGBoost	0.16			
SVR	0.1			

Table 20. Ensemble model results of Dogecoin's price prediction.

	Weights	R2	MAE	MSE
MLP	0.14	80.1%	1.9%	0.4%
XGBoost	0.15			
SVR	0.56			
MLP	0.16	80.6%	1.9%	0.3%
XGBoost	0.1			
SVR	0.79			
MLP	0.13	81.7%	1.9%	0.3%
XGBoost	0.31			
SVR	0.6			

From the above tables, it can be observed that when it comes to Bitcoin and Ethereum, the ensemble model had a worse performance, while in the case of Dogecoin, it performed better, as it reduced the Mean Absolute error by 1.5%.

5.2.4 Forecasting

Out of the three attempts, the model with the optimal performance was the one taking as input tweets volume and transactions volume for all three cryptocurrencies. In particular, Table 21 shows the evaluation metrics of the algorithms that performed the best.

Table 21. Evaluation metrics of the algorithms that performed the best.

Algorithms	**Evaluation Metrics**		
	R2	**MAE**	**MSE**
Bitcoin **Multilayer Perceptron**	91.5%	3.7%	0.5%
Ethereum **Multilayer Perceptron**	93.3%	3%	0.3%
Dogecoin **Support Vector Regression**	83.6%	3.4%	0.3%

To take a closer look, for instance, 93.3% R2 means that 93.3% of the price records can be explained by the model, while 3% Mean Absolute Error means that the absolute distance between the real and the predicted data is 3%, and 0.3% Mean Squared Error means that the squared average distance between the real and the predicted data is 0.3% [53]. To have a better understanding, Table 22 presents a comparison between some of the actual and predicted values that the aforementioned algorithms achieved, for Bitcoin, Ethereum, and Dogecoin respectively.

Table 22. Comparison between actual and predicted values for Bitcoin, Ethereum, and Dogecoin.

Bitcoin			
Actual:	5,934.3	**Predicted:**	5,460.8
Actual:	10,107.99	**Predicted:**	10,330.55
Actual:	45,513.36	**Predicted:**	45,492.9
Ethereum			
Actual:	149.77	**Predicted:**	166.06
Actual:	260.8	**Predicted:**	262.37
Actual:	2,660.78	**Predicted:**	2,737.81
Dogecoin			
Actual:	0.0172	**Predicted:**	0.1563
Actual:	0.1111	**Predicted:**	0.0958
Actual:	0.2786	**Predicted:**	0.2079

6 Testing and Evaluation

This chapter consists of two parts. The first part is the testing, and it is very important as it ensures that the code for all tasks -Data Gathering, Sentiment Analysis, Machine Learning- work properly and without any malfunctions. One the other hand, the second part is the evaluation of the project, which is equally important, assessing if the techniques that were used can provide reliable results [20].

6.1 Testing

For the purpose of testing the written code, all the necessary actions were performed in Data Gathering, Sentiment Analysis, and Machine Learning python code.

To start with, for Data Gathering, various requests were made before starting collecting the data needed for the project, to ensure that the retrieved tweets comply to the search words and the time period that were specified. It turned out that for some requests, the collected data did not refer to the whole period, but the retrieval stopped somewhere in between. That is why it was decided to also use a second method, utilizing tweepy library and Twitter API, as analyzed in previous chapter. It seems that neither this way collected all the existing tweets for the whole specified time period, but at least the combination of the two methods gathered an adequate amount of tweets.

Moreover, in order to test the sentiment analysis' code, the code was executed multiple times to check its consistency, a sample was examined if the tweets were correctly classified as positive, negative, and neutral, and the csv files were checked if they are structured as expected. It was found that a small portion of the tweets were misclassified because of the fact that their content was sarcastic and the sentiment analyzers were not able to recognize that.

As regards the machine learning code, it was executed multiple times in order to ensure that there is not any malfunction.

6.2 Evaluation

The purpose of the project was to predict cryptocurrencies' prices. In a regression problem like this, the aim is not to predict the values exactly (this is practically difficult), but to make as closer predictions to the actual values as possible. Hence, in order to evaluate

the results of this prediction project, the best way is to make a comparison between the actual and the predicted data, and take into account the error metrics. A part of the comparison can be seen in Table 22 of the previous chapter. The error metrics that refer to these results are the following:

- 0.5% MSE and 3.7% MAE for Bitcoin.
- 0.3% MSE and 3% MAE for Ethereum.
- 0.3% MSE and 3.4% MAE for Dogecoin.

The lower the errors, the better the performance of the regression problem. MSE and MAE equal to 0 indicate an ideal model. The aforementioned error metrics are close to 0, so it can be concluded that the predictions were carried out well.

7 Conclusions

7.1 Summary

This dissertation dealt with the problem of predicting cryptocurrency prices using social media data, and specifically prices of Bitcoin, Ethereum, and Dogecoin by using Twitter data. After receiving the first results, it came to light that three-months Twitter data were not enough to achieve accurate results, and sentiment score was not useful in the case of Dogecoin. Therefore, an alternative had to be found in order to reach the target. Based on a correlation analysis, the alternative was to use the tweets volume and the transactions volume for a longer period, since October 15, 2014 for Bitcoin and Dogecoin, and since April 15, 2016 for Ethereum.

The practical part of the project started with Data Gathering. Both Twint and Tweepy are excellent open-source libraries for Twitter data extraction. One would say that twint is simpler to use, without applying for developer access and by overcoming Twitter's limitations on the number of tweets, but with consistency as a trade-off. On the other side of the coin, tweepy requires some more steps, but achieves more trustworthy results. Furthermore, it comes with good documentation and is closely related to Twitter API with its great documentation, hence it is much easier to understand and learn. However, both methods of collecting tweets produced the final datasets consisted of 15,639,318 rows for Bitcoin, 4,955,384 rows for Ethereum, and 3,483,494 rows for Dogecoin.

When it comes to Data Storing, MySQL, one of the most well-liked databases worldwide, proved to be a good choice for storing Twitter data extracted with twint. It was used only for storing and not for managing or processing the data, so a positive conclusion only for some of its advantages has been drawn, which are reliability, compatibility, speed, and ease of use.

During the Sentiment Analysis, TextBlob and VADER techniques assigned scores to all tweets. The first sentiment analyzer proved more useful for Dogecoin, while the second one for the other two cryptocurrencies, based on the correlation analysis between the sentiment scores of each analyzer, and the cryptocurrencies' prices.

As, for the Machine Learning task, the best results were produced by Multilayer Perceptron algorithm for Bitcoin and Ethereum by taking as inputs the tweets volume and transactions volume, while Support Vector Regression performed the best for Dogecoin with

the same inputs. The algorithms achieved really good predictions with 91.5% R2 for Bitcoin, 93.3% for Ethereum, and 83.6% for Dogecoin. For a clearer and better evaluation of a regression model, however, the error metrics need to be considered, which are 3.7% MAE and 0.5% MSE for Bitcoin, 3% MAE and 0.3% MSE for Ethereum, and 3.4% MAE and 0.3% MSE for Dogecoin.

Overall, it can be said that gathering and mining Twitter data and analyzing them properly, can lead to really valuable conclusions about cryptocurrencies' prices. Also, collecting the appropriate data for an adequate time period can accomplish really good price predictions. However, there are various external factors that affect the price of a cryptocurrency, which always need to be taken into account. Moreover, Python is a great tool for the purpose of this project, offering a variety of useful libraries with great documentation.

Everything considered, this dissertation has shown that utilizing the appropriate data can lead to pretty accurate predictions for Bitcoin, Ethereum and Dogecoin prices. Social media data can uncover really valuable information and have high potentials in many fields.

7.2 Limitations

Since the data for the sentiment analysis were collected from Twitter, the biggest limitations that were encountered, are those that the social media platform establishes. Firstly, the fact that tweets older than 7 days could not be extracted, and secondly, that for some requests, not all tweets for a specified time period were retrieved. This is for both twint and tweepy libraries.

7.3 Further Research

Predicting with social media data is a topic with great potential. First of all and most important, researchers are encouraged to collect Twitter data for a longer period, so that they will end up having an adequate dataset for predictions based on the sentiment scores. As for the sentiment analysis, various other techniques could be used, such as other sentiment analyzers, or even better, to manually classify a sample of tweets, and let this classifier assign scores to the remaining data. By doing this, some tweets, and especially those with sarcastic content, might get classified more accurately. Moreover, regarding the machine learning part, more predicting algorithms could be tried, such as *Neural Networks*, which in the literature were used a lot. Lastly, as two theoretical tasks, what could also

be examined is the time lag between the announcement of an event or some news that affected the cryptocurrency, and the change of its price. The second task could be to investigate what affected the similar fluctuations in the sentiment scores for all three cryptocurrencies, which was not examined due to lack of time. It would be really interesting to extract some valuable information and upgrade this project.

References

[1] E. Pintelas, I. E. Livieris, S. Stavroyiannis, T. Kotsilieris, and P. Pintelas, "Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach," *IFIP Advances in Information and Communication Technology*, vol. 584 IFIP, pp. 99–110, Jun. 2020, doi: 10.1007/978-3-030-49186-4_9.

[2] J. Abraham *et al.*, "Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis." [Online]. Available: https://scholar.smu.edu/datasciencereviewhttp://digitalrepository.smu.edu.Availableat:https://scholar.smu.edu/datasciencereview/vol1/iss3/1

[3] J. T. Kjellstadli, E. Bering, M. Hendrick, S. Pradhan, and A. Hansen, "Can local stress enhancement induce stability in fracture processes? Part I: Apparent stability," *Frontiers in Physics*, vol. 7, no. JULY, Jul. 2019, doi: 10.3389/fphy.2019.00098.

[4] K. Wołk, "Advanced social media sentiment analysis for short-term cryptocurrency price prediction," *Expert Systems*, vol. 37, no. 2, Apr. 2020, doi: 10.1111/exsy.12493.

[5] C. Baru, Institute of Electrical and Electronics Engineers, and IEEE Computer Society, *2019 IEEE International Conference on Big Data : proceedings : Dec 9 - Dec 12, 2019, Los Angeles, CA, USA*.

[6] "Blockchain: What it is and How it Works - NerdWallet." https://www.nerdwallet.com/article/investing/blockchain (accessed Jan. 02, 2022).

[7] "Blockchain Definition: What You Need to Know." https://www.investopedia.com/terms/b/blockchain.asp (accessed Jan. 02, 2022).

[8] "Cryptocurrency - Wikipedia." https://en.wikipedia.org/wiki/Cryptocurrency (accessed Jan. 02, 2022).

[9] "Altcoin Investing: What Investors Need to Know." https://www.investopedia.com/terms/a/altcoin.asp (accessed Jan. 02, 2022).

[10] “Crypto Market Cap and DeFi Market Cap Charts — TradingView.” https://www.tradingview.com/markets/cryptocurrencies/global-charts/ (accessed Jan. 02, 2022).

[11] “• Bitcoin price history 2013-2021 | Statista.” https://www.statista.com/statistics/326707/bitcoin-price-index/ (accessed Jan. 02, 2022).

[12] “• Ethereum price history 2015-2021 | Statista.” https://www.statista.com/statistics/806453/price-of-ethereum/ (accessed Jan. 02, 2022).

[13] “• Litecoin price history 2013-2021 | Statista.” https://www.statista.com/statistics/807160/litecoin-price-monthly/ (accessed Jan. 02, 2022).

[14] “Cardano (ADA) Latest Prices, Charts & Data | Nasdaq.” https://www.nasdaq.com/market-activity/cryptocurrency/ada (accessed Jan. 02, 2022).

[15] F. Emmert-Streib, O. P. Yli-Harja, and M. Dehmer, “Data Analytics Applications for Streaming Data From Social Media: What to Predict?,” *Frontiers in Big Data*, vol. 1. Frontiers Media S.A., Sep. 11, 2018. doi: 10.3389/fdata.2018.00002.

[16] Sid, “Forecasting with Predictive Social Media Analytics Eleana Tsiara,” 2018.

[17] D. Rousidis, P. Koukaras, and C. Tjortjis, “Social Media Prediction: A Literature Review.”

[18] M. Barakos, “5 th IBA Bachelor Thesis Conference,” 2015.

[19] “How Many People Use Twitter in 2021? [New Twitter Stats].” https://backlinko.com/twitter-users#twitter-users (accessed Jan. 03, 2022).

[20] Sid, “Predicting Stocks Movement using Social Media Analytics Touparis Fotios.”

[21] Sid, “Predicting the USA presidential elections using Twitter Data Student: Lazaros Oikonomou,” 2016.

[22] “How to Use Twint OSINT Tool on Google Cloud Console ? - GeeksforGeeks.” https://www.geeksforgeeks.org/how-to-use-twint-osint-tool-on-google-cloud-console/ (accessed Jan. 03, 2022).

[23] “overview @ developer.twitter.com.” https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/overview (accessed Jan. 04, 2022).

[24] “Rate-Limits @ Developer.Twitter.Com.” https://developer.twitter.com/en/docs/twitter-api/v1/rate-limits (accessed Jan. 04, 2022).

[25] “Authentication Tutorial — tweepy 4.4.0 documentation.” https://docs.tweepy.org/en/stable/auth_tutorial.html#introduction (accessed Jan. 03, 2022).

[26] “MySQL Features - javatpoint.” https://www.javatpoint.com/mysql-features (accessed Jan. 03, 2022).

[27] “phpMyAdmin.” https://www.phpmyadmin.net/ (accessed Jan. 03, 2022).

[28] “What is phpMyAdmin: Tutorial, Features, Installation - javatpoint.” https://www.javatpoint.com/phpmyadmin (accessed Jan. 03, 2022).

[29] “A brand is no longer what we tell the consumer it is – it is what consumers tell each other it is.” https://quotes.amazingmarketer.in/a-brand-is-no-longer-what-we-tell-the-consumer-it-is-it-is-what-consumers-tell-each-other-it-is/ (accessed Jan. 03, 2022).

[30] “Sentiment analysis - Wikipedia.” https://en.wikipedia.org/wiki/Sentiment_analysis (accessed Jan. 03, 2022).

[31] “Sentiment Analysis | Lexalytics.” https://www.lexalytics.com/technology/sentiment-analysis (accessed Jan. 03, 2022).

[32] “Sentiment Analysis — Comparing 3 Common Approaches: Naive Bayes, LSTM, and VADER | by Kevin C Lee | Towards Data Science.” https://towardsdatascience.com/sentiment-analysis-comparing-3-common-approaches-naive-bayes-lstm-and-vader-ab561f834f89 (accessed Jan. 03, 2022).

[33] “What is Sentiment Analysis? Types and Use Cases - TAUS.” https://blog.taus.net/what-is-sentiment-analysis-types-and-use-cases (accessed Jan. 03, 2022).

[34] “Sentiment Analysis in Python: TextBlob vs Vader Sentiment vs Flair vs Building It From Scratch - neptune.ai.” https://neptune.ai/blog/sentiment-analysis-python-textblob-vs-vader-vs-flair (accessed Jan. 03, 2022).

[35] “Sentiment Analysis — Let TextBlob do all the Work! | by Abdul Hafeez Fahad | Red Buffer | Medium.” https://medium.com/red-buffer/sentiment-analysis-let-textblob-do-all-the-work-9927d803d137 (accessed Jan. 03, 2022).

[36] “Python | Sentiment Analysis using VADER - GeeksforGeeks.” https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/ (accessed Jan. 03, 2022).

[37] “What is the Definition of Machine Learning? | Expert.ai | Expert.ai.” https://www.expert.ai/blog/machine-learning-definition/ (accessed Jan. 03, 2022).

[38]. “book -- machine-learning-refined”.

[39] “Unsupervised Machine Learning: What is, Algorithms, Example.” https://www.guru99.com/unsupervised-machine-learning.html (accessed Jan. 03, 2022).

[40] “Machine learning algorithms explained | InfoWorld.” https://www.infoworld.com/article/3394399/machine-learning-algorithms-explained.html (accessed Jan. 03, 2022).

[41] “Linear Regression Algorithm To Make Predictions Easily.” https://www.analyticsvidhya.com/blog/2021/06/linear-regression-in-machine-learning/ (accessed Jan. 03, 2022).

[42] “Understanding Polynomial Regression Model - Analytics Vidhya.” https://www.analyticsvidhya.com/blog/2021/10/understanding-polynomial-regression-model/ (accessed Jan. 03, 2022).

[43] “What are Recurrent Neural Networks? | IBM.” https://www.ibm.com/cloud/learn/recurrent-neural-networks (accessed Jan. 03, 2022).

[44] “LSTM | Introduction to LSTM | Long Short Term Memor.” https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/ (accessed Jan. 03, 2022).

[45] “XGBoost Algorithm: Long May She Reign! | by Vishal Morde | Towards Data Science.” https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d (accessed Jan. 03, 2022).

[46] “Machine Learning Decision Tree Classification Algorithm - Javatpoint.” https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm (accessed Jan. 03, 2022).

[47] “Gradient Boosting Explained - The Coolest Kid on The Machine Learning Block - Displayr.” https://www.displayr.com/gradient-boosting-the-coolest-kid-on-the-machine-learning-block/ (accessed Jan. 03, 2022).

[48] “Multilayer Perceptron - an overview | ScienceDirect Topics.” https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron (accessed Jan. 03, 2022).

[49] “Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis | by Carolina Bento | Towards Data Science.” https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141 (accessed Jan. 03, 2022).

[50] “Unlocking the True Power of Support Vector Regression | by Ashwin Raj | Towards Data Science.” https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0 (accessed Jan. 03, 2022).

[51] “Understanding ARIMA Models for Machine Learning | Capital One.” https://www.capitalone.com/tech/machine-learning/understanding-arima-models/ (accessed Jan. 03, 2022).

[52] “Regression Metrics for Machine Learning.” https://machinelearningmastery.com/regression-metrics-for-machine-learning/ (accessed Jan. 03, 2022).

[53] “MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better? | by Akshita Chugh | Analytics Vidhya | Medium.” https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e (accessed Jan. 03, 2022).

[54] “Regression Algorithms: which Machine Learning Metrics? | MyDataModels.” https://www.mydatamodels.com/learn/guide/a-path-to-discover-ai/regression-algorithms-which-machine-learning-metrics/ (accessed Jan. 03, 2022).

[55] “How to use Data Scaling Improve Deep Learning Model Stability and Performance.” https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/ (accessed Jan. 03, 2022).

[56] “Correlation Coefficient: Simple Definition, Formula, Easy Calculation Steps.” https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/ (accessed Jan. 03, 2022).

[57] “Pearson’s Correlation Coefficient - Statistics Solutions.” https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/pearsons-correlation-coefficient/ (accessed Jan. 03, 2022).

[58] “Variance-Covariance Method for Calculating Value at Risk - Video & Lesson Transcript | Study.com.” https://study.com/academy/lesson/variance-covariance-method-for-calculating-value-at-risk.html (accessed Jan. 03, 2022).

[59] S. S. Iyengar, V. Saxena, IEEE Computer Society. Technical Committee on Parallel Processing, Institute of Electrical and Electronics Engineers, Jaypee Institute of Information Technology University, and University of Florida. College of Engineering, *2019 Twelfth International Conference on Contemporary Computing (IC3-2019) : 8-10 August 2019, Jaypee Institute of Information Technology, Noida, India*.

[60] L. Oikonomou and C. Tjortjis, “A Method for Predicting the Winner of the USA Presidential Elections using Data extracted from Twitter.” [Online]. Available: http://gnip.com

[61] “API Reference — tweepy 3.5.0 documentation.” https://docs.tweepy.org/en/v3.5.0/api.html (accessed Jan. 04, 2022).

[62] “Connecting MySQL to Python. Python has the ability to connect with… | by Dea Venditama | Chatbots Life.” https://chatbotslife.com/connecting-mysql-to-python-ceef881a19ef (accessed Jan. 04, 2022).

[63] “10 Best Resources to Fetch Cryptocurrency Data in Python | by Nikhil Adithyan | CodeX | Medium.” https://medium.com/codex/10-best-resources-to-fetch-cryptocurrency-data-in-python-8400cf0d0136 (accessed Jan. 04, 2022).

[64] “BTC-USD 46,379.12 -689.96 -1.47 : Bitcoin USD - Yahoo Finance.” https://finance.yahoo.com/quote/BTC-USD/history?p=BTC-USD (accessed Jan. 04, 2022).

[65] “5-ways-to-increase-your-tweet-engagements-with-emojis @ business.twitter.com.” https://business.twitter.com/en/blog/5-ways-to-increase-your-tweet-engagements-with-emojis.html (accessed Jan. 04, 2022).

[66] “Mentions-and-Replies @ Help.Twitter.Com.” https://help.twitter.com/en/using-twitter/mentions-and-replies (accessed Jan. 04, 2022).

[67] “How-To-Use-Hashtags @ Help.Twitter.Com.” https://help.twitter.com/en/using-twitter/how-to-use-hashtags (accessed Jan. 04, 2022).

[68] “Sentiment Analysis — Let TextBlob do all the Work! | by Abdul Hafeez Fahad | Red Buffer | Medium.” https://medium.com/red-buffer/sentiment-analysis-let-textblob-do-all-the-work-9927d803d137 (accessed Jan. 04, 2022).

[69] “textblob · PyPI.” https://pypi.org/project/textblob/ (accessed Jan. 04, 2022).

[70] “NLTK :: Release Notes.” https://www.nltk.org/news (accessed Jan. 04, 2022).

[71] “Ensemble Technique | Ensemble Techniques in Machine Learning.” https://www.analyticsvidhya.com/blog/2021/03/basic-ensemble-technique-in-machine-learning/ (accessed Jan. 04, 2022).