Comparison of Machine Learning Techniques in Sentiment Analysis of Tweets and BTC/USD Price Prediction using Deep Learning and Time Series Models

Yerai Díaz Castro[1]

Supervisor: Xiaochun Meng[2]

September 6, 2022

Abstract

In recent years, interest in sentiment analysis has increased. Sentiment analysis is an analytical method for identifying the emotional meaning of communications. It combines machine learning, natural language processing, and statistics. The first part of this study will be to analyse different Machine Learning Algorithms (Logistic Regression, Gaussian Naïve Bayes and Support Vector Machine) to see which one provides the best results for the sentiment analysis prediction of our datasets. Our datasets for this analysis will be the tweets from Elon Musk, Barack Obama and Jim Cramer for the periods between 01/06/2017 to 01/06/2022.

Stock price prediction studies have been conducted for a long time. Many stock market prediction techniques use time series analysis as a key tool to create forecasts by looking at observed points in the series. Deep learning models have demonstrated an excellent performance on tasks that include stock market prediction thanks to their great capacity to handle massive data and understand the nonlinear relationship between input characteristics and prediction target. Considering the rise of cryptocurrencies in the last decade, in the second part of this study we are going to analyse the adjusted close price of BTC/USD for the same historical period as for sentiment analysis and compare time series models (Arima and Holt-Winters Exponential Smoothing) with a deep learning model (Long Short-Term Memory) to see which ones can better predict its price.

Keywords: Sentiment Analysis, Machine Learning, Logistic Regression, Gaussian Naïve Bayes, Support Vector Machine, Stock Price Prediction, Deep Learning, LSTM, Time Series, Arima, Holt-Winters Exponential Smoothing.

1 Introduction

Sentiment analysis is the study of people's emotions, attitudes, responses and opinions towards an issue, problem, product or service. In recent years, research and studies on sentiment analysis have multiplied due to the great value that governments and private entities place on it. Getting to know the behaviour of the population and its future action in the face of a problem or a product can lead to a very important economic benefit or saving. For this reason, private companies are becoming more and more interested in this analysis.

This sentiment analysis is also known as Opinion Mining (OM). These two expressions express a mutual meaning and are interchangeable although there are different studies that claim that they are slightly different (Tsytsarau and Palpanas, 2012).

[1] MSc student in Fintech, Risk and Investment Analysis, University of Sussex, United Kingdom. Email: yeraidiazcastro@gmail.com

[2] Lecturer in Finance, University of Sussex, United Kingdom. Email: xiaochun.meng@sussex.ac.uk. Tel: +44-(0)1273 873428 Ext.3428.

[3] I really appreciate Dr. Xiaochun Meng's major contribution to this project.

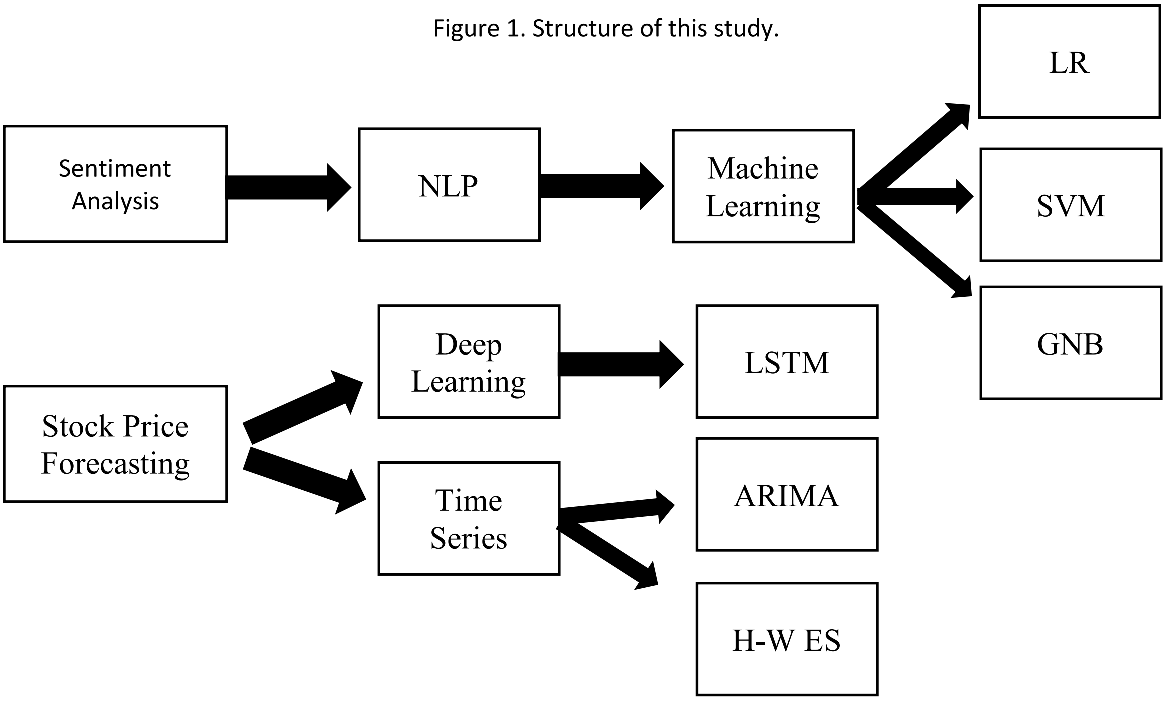
Twitter is one of the most important social networks in the world with more than 320 million users. Different Twitter features such as the number of followers of each user, the number of retweets each comment generates, the number of likes each tweet has, can be used as a parameter to create a sentiment analysis.

For this reason, this platform was chosen for the extraction of information through the use of the Twiter API for subsequent analysis using the Python Tweepy library. An exploratory analysis of datasets has been performed after cleaning irrelevant data from the datasets. Subsequently, different Machine Learning models have been implemented to see which one offers the most accurate sentiment predictions.

By utilizing the modelling of the three classifiers stated above, the machine learning algorithms used for sentiment analysis in this study give the intuition behind the task of sentiment categorization and their accuracy. The implementation will be in the third part of this study denominated methodology.

In this figure 1 you can see how the structure of this study, mentioned previously in the abstract.

Figure 1. Structure of this study.



For several decades, stock price forecasting has always been of great interest, especially in scientific studies. Many studies have concluded that when efficiently developed and optimised forecasting models can predict these prices quite reliably. Also, the selected variables used for this construction considerably affect the accuracy obtained. In contrast, there are numerous studies that do not support these theories.

Due to its simplicity, transparency, and efficiency, Bitcoin, which is a decentralized digital currency and payment system that is supported by neither a central bank nor an administration, has drawn considerable interest. Bitcoin is the most widely used and successful virtual currency right now.

Deep learning models have evolved into a state-of-the-art neural network architecture that improves prediction accuracy in various domains. It is considered a part of machine learning family methods based on artificial neural networks with representational learning. Deep learning techniques have been used in a variety of programs, including those for computer vision, natural language processing, machine translation, speech recognition, climate science, medical image analysis, drug design, bioinformatic, materials inspection, and many others. In these programs, the results were comparable to those of human experts, and in some cases, they even outperformed them. The analysis of time-dependent data has become more effective with the introduction of LSTM (Hochreiter and Schmidhuber, 1997). These networks can store historical data and have been applied to the prediction of stock prices.

An investment time series keeps track of data points at regular intervals and plots the changes in the selected data points over time, such as the price of an asset. For this reason, time series models have been used to forecast the stock market price since decades ago.

ARIMA models are based on statistical models. According to the literature, prediction may generally be done from two perspectives: statistics and artificial intelligence approaches (Wang et al., 2012). Other statistic model is Holt-Winter Exponential Smoothing. The Holt-Winters method is used to exponentially smooth a large number of historical data in order to predict "typical" values for the present and the future. Using an exponentially weighted moving average, exponential smoothing is the process of "smoothing" a time series (EWMA).

For all the reasons mentioned aboved, in the second part of this study we will focus on the Bitcoin/usd adjusted price for this 5 years mentioned and compare which of these 3 models outperform each other.

This study is structured as follows: section 2 proposes the related work, section 3 shows the methodology used, section 4 are the results obtained and section 5 are the conclusions.

2 Related Work / Literature Review

2.1 Twitter Data

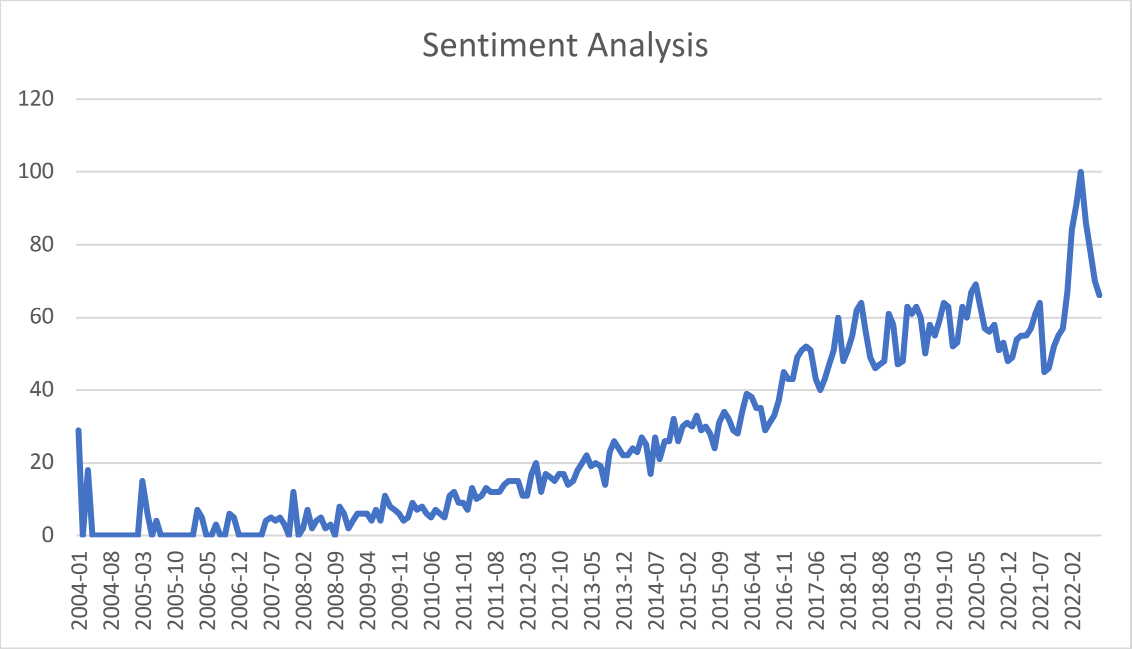
There are many reasons why Twitter has become one of the most important sources of data for social and business research. Twitter messages have different limitations such as the total number of characters (280) that can be included in messages, although it allows these messages to be spontaneous and easy to interpret (Chinnov et al., 2015; Díaz et al., 2016).
In addition, there are different modalities such as mentions, retweets, and hashtags that offer different inputs for different analyses, both at the level of social networks and communication, as well as at the level of influence, marketing and popularity (Bruns and Stieglitz., 2013; Philander and Zhong., 2016).

These characteristics show that Twitter is one of the ideal places to observe in real time the behaviour of the population both globally and regionally. This makes it very attractive for companies to know the future purchasing ideas of a specific audience before launching a new product on the market. In addition, they can also get to know the opinion and feedback generated by consumers before or after the launch of a new product.
It is important to mention that there are negative characteristics in tweets to be taken into account such as spam (Chinnov et al. 2009), nonsensical opinions (Fang and Zhan, 2015) and irony (Reyes et al., 2013).

2.2 Sentiment Analysis

Sentiment analysis has been increasingly in demand and developed over the last decade. The interest of society and companies in the results of these analyses has caused the number of sentiment analyses and their searches to skyrocket, as can be seen in Fig 2.

Fig 2. Global Searches on Google Trends for Sentiment Analysis.



The figures represent search interest in relation to a graph's greatest value for a particular area and time period. A term's maximum popularity is indicated by a value of 100, whereas values of 50 and 0 indicate that the term is only halfway as popular as its maximum value or that there are insufficient data for the term, respectively. According to data provided by Google Trends, in April 2022 it reached a score of 100 and then dropped to almost 70 points.
Financial investors who wish to learn about and react to market sentiment can also benefit from the analytical views that sentiment analysis can offer as you can see in the studies carried on by Oliveira et al., (2013) and Bollen et al., (2010).

The studies conducted make use of several techniques from many fields of computer science. Some of them use machine learning techniques that frequently use supervised classification strategies and require label data to train the classifiers (see Rawat et al., 2021; Mukhtar et al., 2018; Duwairi et al., 2014; Yadav et al., 2021; Habernal et al., 2013, June).

Natural Language Processing is the name given to the technique of providing the ability to understand text and words provided by humans to computers for subsequent interpretation. This branch comes from computer science and more specifically from artificial intelligence. NLP is characterised by the use of computational linguistics with statistical models of deep and machine learning. Nowadays it is very likely that you have used a product or service that supports this technique, from voice assistants such as Siri and Alexa, GPS, customer service chatbot and many more.

Numerous studies on the impact of NLP on sentiment analysis have been carried out in recent years (see Yildirim et al., 2014; Ahuja et al., 2019; Yu et al., 2013; Carvalho et al., 2020).

2.2 Machine Learning Algorithms (MLA)

Modern organizations and services already heavily rely on machine learning to function. Machine learning models are used in many different contexts, including social networking platforms, healthcare, finance and many more.

Machine Learning relies on different algorithms to solve different data problems. It is a very common statement in the data scientist world that there is no single type of algorithm that is capable of solving a problem, but that there are several. Depending on the type of problem to be solved, the number of variables, a different type of algorithm can be used to see which model adapts and offers a better solution.

However, depending on the job at hand and the data that is available, different procedures will be required to train and deploy a model.

2.2.1 Supervised and Unsupervised Learning

There are two main approaches to machine learning model development, supervised and unsupervised learning, as examples. They differ in terms of how the models are developed and the quality of the necessary training data.

A supervised learning model will often encounter a different task or difficulty than an unsupervised learning model since each method has different capabilities.
Understanding the data at hand and the issue that needs to be resolved will help a firm decide whether to install a machine learning model. We will now examine the key distinctions between supervised and unsupervised machine learning, as well as how to use both.

Supervised machine learning needs labelled input and output data during the learning phase of the machine learning lifecycle. Before using it to train and test the model, a data scientist would commonly label this training data during the pre-processing stage. Once the model has figured out how the input and output data are connected, it may be used to classify previously unknown datasets and predict outcomes.

Because an algorithm learning from the training dataset can be compared to a teacher supervising the learning process, it is known as supervised learning. Most of the information that is available is unlabelled, raw information. For data to be adequately labelled and suitable for supervised learning, human input is typically necessary. Naturally, this approach might be resource-intensive because to the large amounts of properly labelled training data that are needed.

Using learning patterns in training data, supervised machine learning is frequently used to categorize various file kinds, including photos, documents, and written words, as well as to predict future trends and results (see Nasteski, 2017; Berry et al., 2019).
Unsupervised machine learning is the training of models on raw and unlabelled training data. It is often used to find different patterns in unprocessed data in order to sort them into categories and classify them. This process is often a strategy used in the initial stage of understanding the data we have.

Compared to supervised machine learning, unsupervised machine learning adopts a more passive strategy. Although humans provide the model's hyperparameters, such as the number of cluster points, the model effectively processes large amounts of data without human oversight. As a result, they are well-suited to reply to questions about hidden connections and patterns in the data. However, since there is less human oversight, unsupervised machine learning merits more consideration.

Most of the information that is available is unlabelled, raw information.
Unsupervised machine learning is mostly used to cluster datasets on the basis of similarity in features or data segments, comprehend relationships between various data points, such as automatic music suggestions, and carry out preliminary data analysis (see Chambers and Jurafsky, 2008; Celebi and Aydin, 2016; Berry et al., 2019).

2.2.2 Logistic Regression, Gaussian Naïve Bayes and Support Vector Machine

Over the last few years several machine learning algorithms have been implemented in sentiment analysis. One of the most often used Machine Learning algorithms, within the category of Supervised Learning, is logistic regression. It is used to forecast the categorical dependent variable using a specified set of independent variables.
When a categorical dependent variable is used, logistic regression is used to predict the result, and the predicted value should be discrete or categorical and provide probabilistic values between 0 and 1, not binary values of 0 and 1. It can be binary form either Yes or No, 0 or 1, etc (Tyagi and Sharma, 2018).

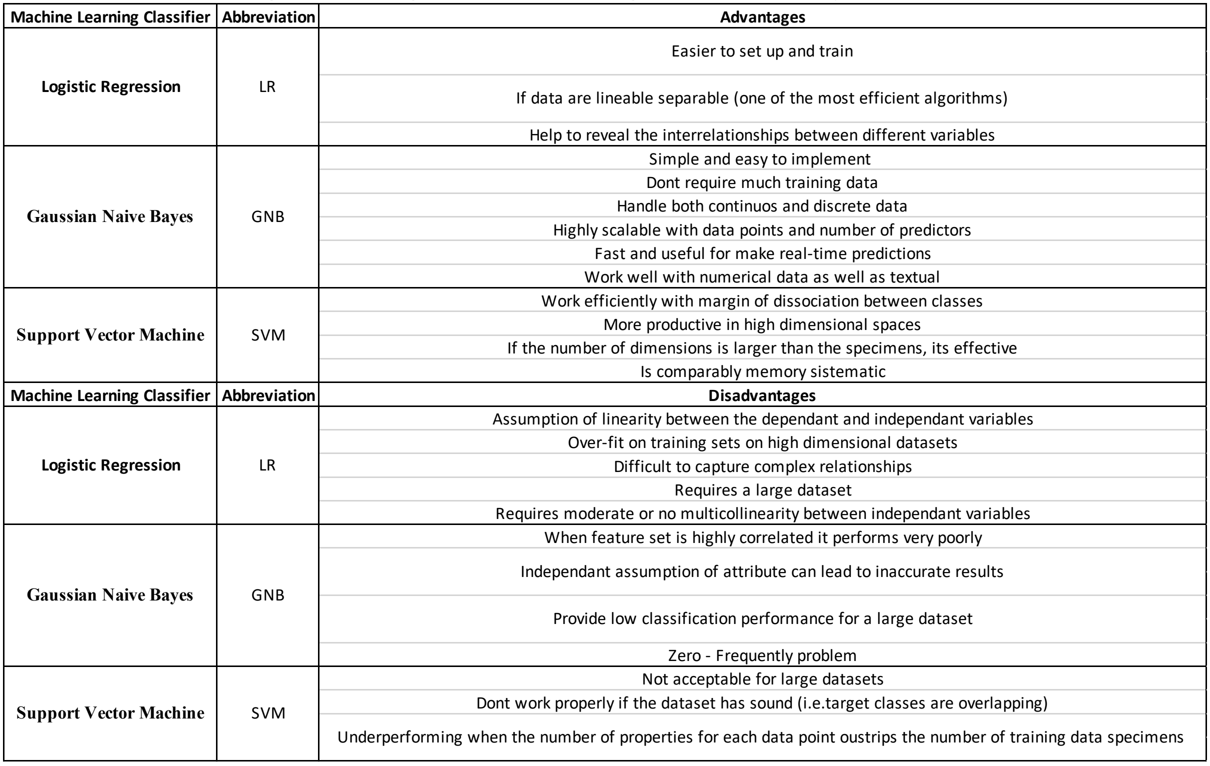
Logistic regression is used to solve classification problems and linear regression is used to solve regression problems, which makes them similar. Poornima and Priya (2020), affirm that Logistic Regression performs better comparing to other supervised machine learning algorithms for Twitter sentiment analysis of sentence embedding getting an approximately accuracy of 86%.

Gaussian Naive Bayes is a variant of Naive Bayes that uses continuous data and follows the Gaussian normal distribution. Naive bayes, a group of supervised machine learning classification techniques based on the Bayes theorem, is a simple yet effective categorization technique. When the inputs are highly dimensional, they are useful. The Naive Bayes Classifier may be used to solve complex classification issues as well as text classification and spam identification although to estimate parameters these classifiers need training data. Rathee et al., (2018) have provided an accuracy of 72.4% for IMDB results after applying the Gaussian Naïve Bayes for sentiment analysis.

The most effective classifier for situations involving voice classification is known as Support Vector Machine (SVM). They succeeded by developing a hyperplane with the shortest training instances separated by the greatest Euclidean distance. A very small fraction of the training data sets, which are used as support vectors, entirely resolves the Support Vector Machine hyperplane. There is no access to the trained classifier for the remaining training data sets. Therefore, the classifier SVMs have been effectively employed for text classification and have also been used in many sequence processing applications. SVMs are utilized in text categorization and hypertext since they don't need labelled training data.
Alsaeedi and Khan (2019) classified the techniques for Twitter sentiment analysis using machine learning approaches getting an accuracy of 85% for SVM.

In figure 3 we can see different advantages and disadvantages of the models mentioned above.

Fig 3. Advantages and Disadvantages of the MLA implemented.



Machine Learning Classifier	Abbreviation	Advantages
Logistic Regression	LR	Easier to set up and train
		If data are lineable separable (one of the most efficient algorithms)
		Help to reveal the interrelationships between different variables
Gaussian Naive Bayes	GNB	Simple and easy to implement
		Dont require much training data
		Handle both continuos and discrete data
		Highly scalable with data points and number of predictors
		Fast and useful for make real-time predictions
		Work well with numerical data as well as textual
Support Vector Machine	SVM	Work efficiently with margin of dissociation between classes
		More productive in high dimensional spaces
		If the number of dimensions is larger than the specimens, its effective
		Is comparably memory sistematic
Machine Learning Classifier	**Abbreviation**	**Disadvantages**
Logistic Regression	LR	Assumption of linearity between the dependant and independant variables
		Over-fit on training sets on high dimensional datasets
		Difficult to capture complex relationships
		Requires a large dataset
		Requires moderate or no multicollinearity between independant variables
Gaussian Naive Bayes	GNB	When feature set is highly correlated it performs very poorly
		Independant assumption of attribute can lead to inaccurate results
		Provide low classification performance for a large dataset
		Zero - Frequently problem
Support Vector Machine	SVM	Not acceptable for large datasets
		Dont work properly if the dataset has sound (i.e.target classes are overlapping)
		Underperforming when the number of properties for each data point oustrips the number of training data specimens

2.3 Deep Learning – Long Short-Term Memory (LSTM)

Since 2010, deep learning has developed rapidly covering different areas such as image recognition, speech recognition, natural language, stock price prediction and many more (Yao and Guan, 2018).

A typical form of artificial neural network used in voice recognition and natural language processing is the recurrent neural network. One of the numerous varieties of Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), is able to capture information from earlier stages and utilize it to make predictions for the future (Patterson and Gibson, 2017).
An Artificial Neural Network (ANN) typically has three layers: the input layer, the hidden layers, and the output layer. The size of the data always determines the number of nodes in the input layer of a NN with a single hidden layer, and the input layer nodes are connected to the hidden layer via connections known as synapses.

The weight coefficient, which is present in every two-node connection from the input to the hidden layer, controls how signals are handled.

The Artificial NN will have the best weights for each synapses once the learning process is over. Continuous weight changes are a natural part of learning.

The hidden layer nodes alter the sum of input layer weights using the activation function, which is a transformation that results in values with a lower error rate between train and test data.

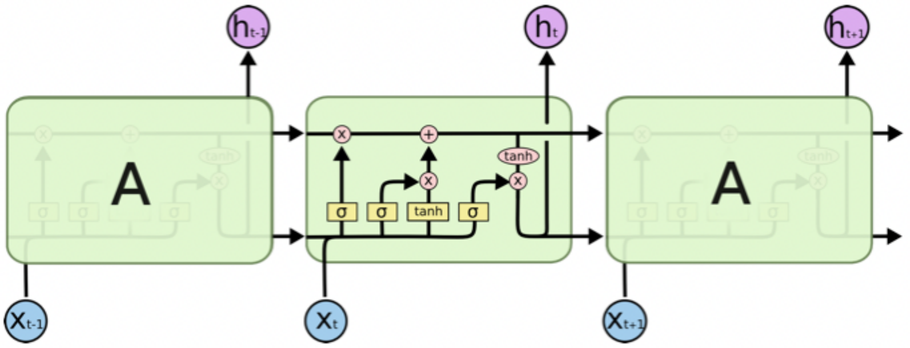
The output layer of our NN is comprised of the values acquired after this change, however they might not be the best. In this instance, the best weight and the ideal error will be targeted using a back propagation procedure, which connects the output layer to the hidden layer and delivers a signal for the specified number of epochs.

We'll go through this procedure again in an effort to refine our forecasts and reduce prediction error. After finishing this step, the model will be trained. Recurrent Neural Networks (RNN) are a class of NN that use previous stages to learn from data and forecast future trends. They estimate future value based on previous sequences of observations (Selvin et al., 2017).

To forecast and anticipate future values, it is important to keep in mind the previous phases of the data. In this situation, the hidden layer serves as a repository for historical data from the sequential data. The method of utilizing components of previous sequences to predict future data is referred to as recurrent.

Since RNN cannot retain long time memory, Long Short-Term Memory (LSTM) based on "memory line" has been discovered to be extremely useful in anticipating scenarios with long time data.
The early phases of an LSTM can be remembered using integrated gates along a memory line.

Fig 4. LSTM Structure.



The LSTM is a unique class of RNNs since it has the capacity to memorize data sequences.

Each cell's upper line serves as a transport line connecting the models and transferring historical data to contemporary models. Through the addition of values from one cell to another, the independence of the cells helps the model's disposal filter. A group of cells responsible for storing passed data streams must be present in every LSTM node.

The sigmoidal neural network layer that constitutes the gates ultimately drives the cell to an optimum value by excluding or permitting input to flow. A binary value (0 or 1) is assigned to each sigmoid layer, with 0 meaning "allow nothing pass through" and 1 meaning "let everything pass through."

The Forget Gate generates a value between 0 and 1, with 1 denoting "totally keep this" and 0 denoting "completely disregard this."

Memory Gate selects the fresh data that will be kept in the cell. The initial "input door layer" of a sigmoid layer selects the values to be modified. A tanh layer is then used to construct a vector of potential new candidate values that could be added to the state.

filtered and most recent data contributed as well as the current state of the cell to get its value. Figure 5 shows the advantages and disadvantages of this model.

Fig 5. Advantages and Disadvantages of LSTM

Long-Short Term Memory
Advantages
Process inputs of any length
Remember information throughout the time
Large range of parameters (learning rates, input and output biases)
Low complexity to update weights
The weights can be shared across the time steps
Disadvantages
Take longer to train
Require more memory to train
Easy to overfit
Dropout is much harder to implement in LSTMs
Sensitive to random weight initializations

2.4 Time Series Models

Time-series forecasting models are those that can anticipate future values based on values that have already been seen. For non-stationary data, time-series forecasting is frequently employed. Non-stationary data are those whose statistical characteristics, such as the mean and standard deviation, do not remain constant throughout time but rather change.

These models' non-stationary input data are typically referred to as time-series. Time-series examples include changes in temperature, stock prices, home prices, and other variables throughout time. As a result, the input is a signal (time-series) that is made up of observations that were made in chronological order.

2.4.1 ARIMA

Auto-Regressive Integrated Moving Average (ARIMA) is formed by three terms. Autoregression (AR), Integrated (I) and Moving average (MA). The ARIMA follows the next equation:

$$Y_t = \alpha + \beta Y_{t-1} + \gamma \varepsilon_{t-1} + \varepsilon_t$$

This equation basically tells us that if we want to predict the adjusted close price of BTC/USD at time t, Y_t, we need to consider alpha, beta and gamma as fixed parameters. Then, our Y_{t-1}will be the expected price of BTC/USD from the previous day. If the expected price is different from the real price, we have an error which is gonna be introduced into the equation as ε_{t-1}. This process will be done again for all the data introduced.
The Autoregressive terms (AR) in my model consists of Adj Close Price of BTC/USD from 01/06/2017 to 01/06/2022 (1827 records). This is to represent the past information. This term is expected to contain predicted power for predicting future based on past information.

The Moving Average (MA) of our model will be the forecast of the price of BTC/USD for this period. It was obtained averaging the residuals from the past observations. This combination is considered as ARMA model. If Y_t is non-stationary, things become problematic. In our study we find our data is non-stationary. To solve this, we apply the ADF test to check it and this process can be read in the methodology part. The process to convert the data from non-stationary to stationary is referred as Integrated.

2.4.1 Holt-Winters Exponential Smoothing

Time series data with a trend and seasonal variation are forecasted using Holt-Winters Exponential Smoothing. The following four forecasting methods are built one on top of the other to make up the Holt-Winters methodology. These forecasting methods are: Holt-Winters ES, and Holt ES , Exponential smoothing, Weighted Averaging (Goodwin, 2010).

Using a weighted average of all prior values, where the weights decrement exponentially from the most recent to the oldest historical value, the Exponential Smoothing (ES) approach predicts the following result.
This method follows the next equation:

$$y_t = b1 + b_2 t + S_t + \in_t$$

Where:

$b1$ is the permanent component which is our training data.
b_2 is the trend component which is our test data.
S_t is a additive seasonal factor, which is 10 in our model.
$\in_t$ is the random error component

When utilizing ES, you must consider that the most recent values of the time series have a bigger bearing on your decisions than those from earlier in the series. The ES approach has two key drawbacks: It cannot be used when your data has a pattern or seasonal fluctuations.

3 Methodology

The Twitter API is a collection of programmatic endpoints that may be utilised to comprehend or create the Twitter discourse. You may locate and get, interact with, or create a number of resources using this API. With the help of the free source Python module Tweepy, I have worked to get all the tweets of the 3 datasets (Jim Cramer, Barack Obama and Elon Musk). It is important to mention that you have an easy access to the Twitter API with this library.

Once we have all the tweets of these 3 datasets for a period of 5 years (01/06/2017 to 01/06/2022) we will proceed to apply NLP[1] The goal of NLP, a subfield of AI[2] in computer science, is to enable computers to comprehend how people write and communicate. 61835 tweets from the 3 datasets are gathered. The different stages carried out in NLP can be seen in Fig 6.

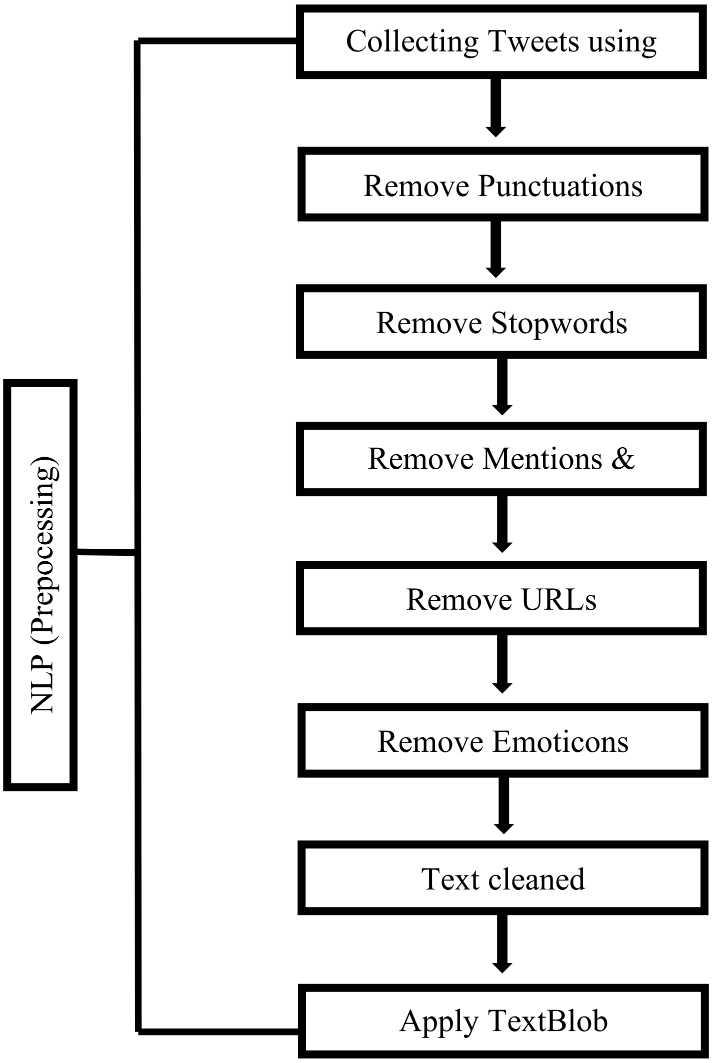


Fig 6. Dataset construction using NLP.

3.1 Dataset construction using NLP

These stages consist of gathering the dataset of tweets from Elon Musk, Jim Cramer and Barack Obama, pre-processing the tweets and annotations, removing empty words, removing Stopwords, removing mentions and retweets made by these 3 datasets, removing URLs, managing and removing emoticons. Once the text is clean, we can apply TextBlob.

We grab the tweets from the 3 datasets using the Tweepy python library by connecting to the Twitter API. Check characters to see if they are in punctuation to remove them. Then we join the characters again to form a string before apply Stopwords. To speed up the processing of the data, a set of undesired Stopwords are eliminated. Unneeded words are removed (326 words). When mining unstructured data, removing unwanted terms is a crucial task. The next step is to remove mentions, retweets, URL links and emoticons(e.g. @Elonmusk, RT which means Retweet and :D, :)).

Once we have done all the above steps, the text is cleaned and we can start using the obtained text to perform sentiment analysis applying TextBlob. Being a Lexicon-based sentiment analyzer, TextBlob contains certain predetermined guidelines. We can say a word and weight dictionary, which has some scores that assist in determining the polarity and subjectivity of a statement. We create subjectivity and polarity columns in our dataset to see whether the impact of tweets is positive, negative or neutral.

Once we have this data, we set out to perform an exploratory data analysis to see what percentage of tweets in each dataset are positive, negative or neutral, to get general data on the length of tweets in each dataset, to see the total number of samples for each, to see what percentage of Bitcoin return has been higher for each dataset, and to see the polarity and subjectivity of each before proceeding to use Machine Learning Algorithms for these datasets to see which model performs better and gives us greater accuracy.

3.2 Machine Learning Algorithms (MLA)

Machine Learning Algorithms (MLA) are applied. These algorithms are tied with learning structures and includes different classification algorithms for the text. The suggested dataset, which is made up of training and test data, is subjected to several supervised learning methods. This paper utilizes the Logistic Regression (LR), Gaussian Naive Bayes (GNB), Support Vector Machine (SVM) algorithms for classifying the dataset and comparing the results obtained using these algorithms to come up with an efficient algorithm that results in an accurate classification.

In the construction of this model, we take the text column for the variable X and the analysis column obtained previously through the results with the TextBlob library which marks whether the sentiment is positive, negative or neutral for the variable Y. We transform the variable X into vectors for further analysis. Subsequently, we divide the data into train and test split, giving a value of 80% to train and 20% to test.

After importing the models from Sklearn we proceed to train the data in each model and then load and fit the model. Subsequently, we will calculate the prediction and accuracy for each of the implemented models.

- Logistic Regression: A predictive analytic technique based on the idea of probability. This MLA is used for categorization problems. It used the sigmoid function representing the cost (Tyagi and Sharma, 2018). After train the data into the model, load and fit it, we predict and calculate the accuracy of the model.
- Gaussian Naive Bayes: An extension of the Naïve Bayes that follows Gaussian normal distributions. This algorithm supports continuous and discrete data. (John and Langley, 2013). Additionally, it excels in multiclass prediction.
- Support Vector Machine: It is used for classification and regression. Finding a hyperplane in an N-dimensional space that clearly classifies the data points is the goal of the SVM.

3.3 Parameters optimization

It is worth mentioning that GridSearchCV techniques are also applied to these machine learning algorithms to see if they can improve the accuracy results obtained for these algorithms. A method for finding the optimum parameter values in a grid of parameters is called GridSearchCV (Shuai et al., 2018).. Essentially, it is a cross-validation approach (Ranjan et al., 2019). Fit and forecast are the two primary approaches that may be used with GridSearchCV. The settings vary depending on the type of algorithm that is being used to analyse the dataset at hand. The crucial parameters require a separate set of settings from the user. The parameters and the model must be entered. The forecasts are created after extracting the best parameter values.

In order to prevent either underfitting or overfitting, machine learning classifiers employ one or more tuning parameters The model's performance is maximized by using that set of these tuning parameters, which minimizes a predetermined loss function and produces better outcomes with fewer mistakes while preventing overfitting or underfitting. To train each machine learning algorithm and fine-tune its parameters, a grid of adjusted classification algorithms is created using the fit method from the Scikit-learn GridSearchCV class. The whole training data set is utilised to generate the final model once the parameters have been set to their ideal levels.

3.4 Classifiers performance evaluation

The studies were carried out in Python 3.9.12 with an 8GB RAM utilising the Natural Language Tool Kit (NLTK) and Scikit-learn where the effectiveness of the chosen classifiers is assessed using different evaluation metrics such as Accuracy, Precision, Recall and F-1 Score. It is important to mention that we will also analyse the performance of the algorithms using confusion matrix.

A classification metric for evaluating classifiers is accuracy. Accuracy may not be an appropriate criterion for assessing classifier performance when the class distribution in the dataset is not uniform. As a result, the accuracy, recall, and F-measure of a confusion matrix are determined in order to assess classification performance.

Accuracy is expressed using the following formula:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

The ability of the model to correctly categorise samples is known as precision, and it can be obtained as follows.

$$Precision = \frac{TP\ (True\ Positive\ Rate)}{TP + FP(False\ Positive\ Rate)}$$

It is important to mention that TP is the true positive rate of the algorithm which is the estimation made correct whereas FP is the false positive rate which is the estimation made wrongly and taking into account as positive as well as for the recall, FN is the false negative rate of the algorithm which is the wrongly estimation when it was positive.

Recall, which is expressed by the following equation, demonstrates the model's ability to categorise the largest number of potential samples.

$$Recall = \frac{TP}{TP + FN\ (False\ Negative\ Rate)}$$

The F1-score conveys the balance between the precision and the recall.

$$F1 - Score = \frac{Precision\ x\ Recall}{Precision + Recall}$$

3.5 Deep Learning Algorithms (DLA)

Bitcoin prices have been obtained from the Yahoo Finance website for the periods from 01/06/2017 to 01/06/2022.

In this part we are going to compare the results obtained in the price forecast of bitcoin price using LSTM and later compare these results with the predictions obtained in the time series methods(ARIMA and H-W ES) and observe which models perform better in the prediction.

Long Short-Term Memory networks (LSTM) are a type of recurrent neural network. These networks are able to learn order dependence in sequence prediction problems. A LSTM network is an RNN in which the internal memory-equipped cell "remembers" the value for a number of time intervals, making it ideal for forecasting time series.

To apply this techniques in Python, we will use Tensorflow and Keras. In order to make the implementation of neural networks simple, Google established the high-level Keras deep learning API, which was developed in Python.

The mean squared error with the Adam Optimizer (MSE) is then selected as the main loss function and the deep learning model (LSTM) is trained in a supervised learning environment. The MSE follow the next equation:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_1 - \widehat{y_1})^2$$

The MSE is excellent for guaranteeing that our trained model does not contain any outlier predictions with significant errors. Due to the squaring component of the function, the MSE gives larger weight on these errors. However, the main disadvantage is that the squaring portion of the function increases the error if our model produces a single really poor forecast.

For deep learning the LSTM model was implemented for model building. The BTC/USD price data from 01/06/2017 to 01/06/2022 was used. In a univariate analysis, just the Adj Close column was considered. For the separation of the data 80% (1461 records) has been taken into account as training set while the remaining 20% (366 records) was used as testing set. LSTM is sensitive to the scale of the data. So we will apply MinMax Scaler to scale it and convert the data in a range of 0 to 1. After this process the training and testing data was ready. As a time step, a 100 days rolling window was implemented. Our goal was to make X- train on the first 100 days (index: 0 to 99) of the train set.

The X-train was an array of scaled values where the 101st day value will be the y-train (index:100). Later on, the sliding window will move ahead by one day. The process will be repeated and X-train will have array values for the next one hundred days (index: 1 to 100) and the y-train will have the 102nd which belongs to the index 101 value. In the same way, it will keep going until the Y-train has the last index of the train set and the X-train has values up to the second-to-last record of the train set.

The train set will have 1361 days in all. The counter will then be moved to the testing set after that. Once more, on the test set, the value for day 100 will be designated as the X-test, and the value for day 101 will be the Y-test. This will continue till the test set is finished. The test set will include 266 days total.

Simple machine learning models operate according to the idea of giving data one by one, and predictions are only created using that. But with LSTM, values were further projected based on historical data. Here, the order is crucial.

The components kept in short-term memory are used by RNN. These components aid in the prediction of the sequence. At least one of its connections is a feedback one, which aids in looping activations.

The LSTM model was selected for model construction. It will have a sequential of LSTM layers followed by a dropout layer (50%).

A regularization technique called dropout excludes input and recurrent connections to LSTM units probabilistically from weight and activation updates during network training. As a result, overfitting is decreased and model performance is enhanced.

The last layer employed is a dense layer, which is a layer of a neural network with many connections. One hundred time steps and fifty features are present in the input layer of the LSTM. 50 nodes make up the first hidden LSTM layer, which is followed by a 50% dropout (dropout 1). 50 nodes make up the second hidden LSTM layer, which is then dropped out by 50% (dropout 1).

At the end, a dense output layer (dense). The training set is used to train the model (X-train and y-train). With the aid of Adam optimizer, the model is built, and Root Mean Squared Error is used to calculate the error (RMSE). With a 64 batch size, the network is trained across 100 epochs.

Each of the near values is based on the training data from the preceding 100 days. Figure 7 shows the summary of the LSTM model.

Layer (type)	**Output Shape**	**Parameters**
lstm (LSTM)	(None, 100, 50)	10400
dropout (Dropout)	(None, 100, 50)	0
lstm_1 (LSTM)	(None, 100, 50)	20200
dropout_1 (Dropout)	(None, 100, 50)	0
lstm_2 (LSTM)	(None, 50)	20200
dropout_2 (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51
Total params	50851	
Trainable params	50851	
Non-trainable params	0	

Fig 7. LSTM Model Summary.

On the other hand, figure 8 shows the LSTM model used for BTC/USD (Adj Close) prediction.

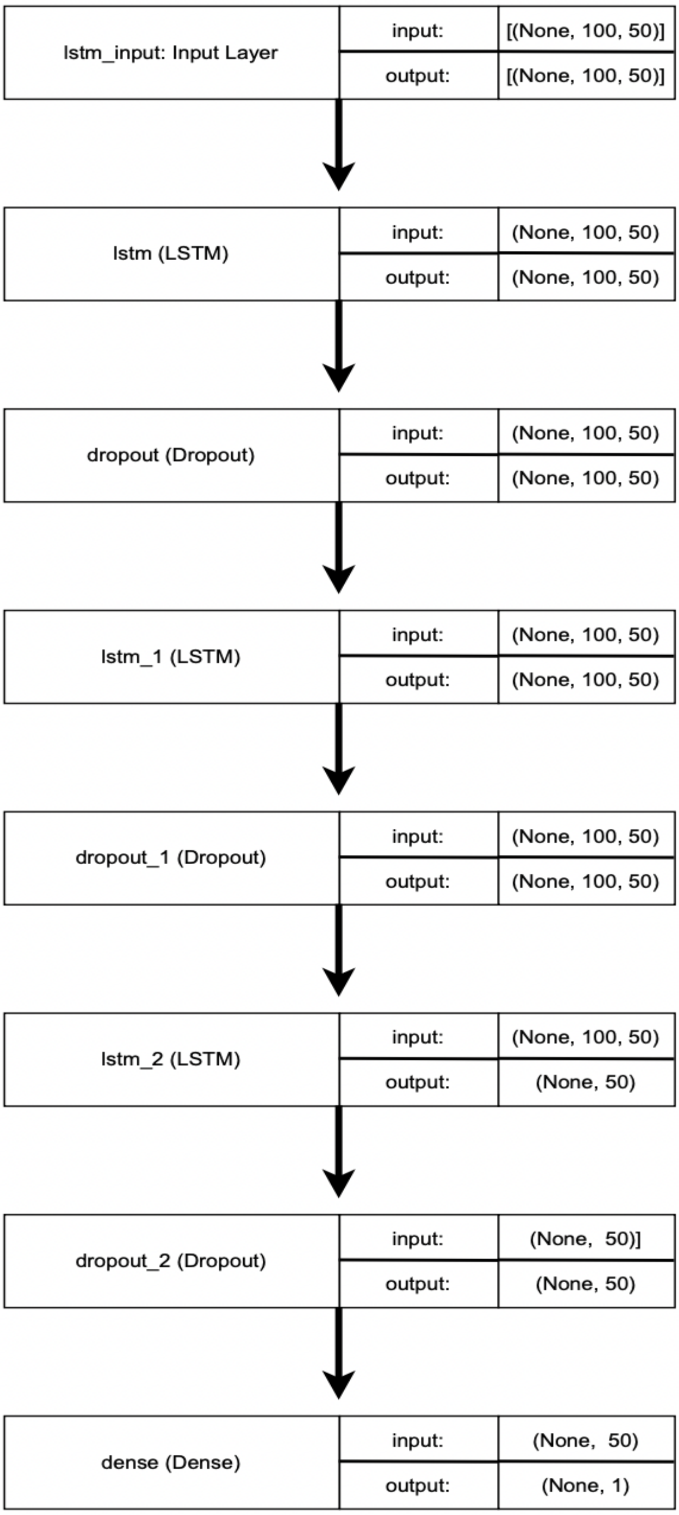


Fig 8. LSTM Model for BTC/USD prediction.

3.6 Time Series Models

A statistical package called PMDARIMA was created to fill the gap in Python's time series analysis capabilities. This package was used in our study.

Univariate analysis was done using ARIMA, applying it on the Adj Close column of BTC/USD for the 5 years historical data (01/06/2017 to 01/06/2022).

Initially, there were 1827 records. After splitting our data into train and test size, 1461 records were used as a training dataset, and 366 records were used as a testing dataset.

We have been applying the ADF test to check whether the price series is stationary. Since the ADF test's null hypothesis states that the time series is non-stationary, we can reject the null hypothesis and conclude that the time series is stationary if the test's p-value is less than the significance level of 0.05. To calculate this data we imported adfuller from statsmodels.tsa.stattools.

The ARIMA() function was imported from the arima_model sub-package of statsmodel.tsa package(statsmodels.tsa.arima_model). The ideal values of p, d, and q were determined using the auto arima function.

To determine the forecasted BTC/USD price value, the optimal values of p, d, and q were passed into the ARIMA() algorithm as an input. To obtain d we have used the pandas differencing function to see if the time series has become stationary and to see the noise in the data for each differencing. To obtain the p-value we looked at the PACF traffic. This graph can be used to draw a correlation between the time series and its lag to obtain this value. The ACF plot was used to determine the value of q and will show us how much moving average is needed to eliminate autocorrelation from stationary time series.

When comparing y-test and y-pred, the ratio RMSE score and test mean of the Adj Close was determined to assess the model's performance.

Besides, a univariate analysis was done using Holt- Winters Exponential Smoothing, applying it on the Adj Close column of BTC/USD for this 5 years historical period. The same as for ARIMA, there were 1827 records and after splitting our data into train and test size, 1461 records were used as a training dataset, and 366 records were used as a testing dataset.

The statsmodels package's Exponential Smoothing () method was imported (statsmodels.tsa.holtwinters). Since seasonality remained constant across the dataset, the seasonal parameter was set to additive instead of multiplicative. Given that there are 10 seasonal periods, the data will show seasonality every ten days.

The performance of the model was calculated using the ratio between the RMSE score and the mean of the test section of the Adj Close of BTC/USD. The performance of the model was calculated using the ratio between the RMSE score and the mean of the test section of the Adj Close of BTC/USD.

4 Empirical Results

In this first part of the results, we will analyse the exploratory data analysis for sentiment analysis together with the results obtained for the classifiers of the machine learning algorithms and their confusion matrix.

Fig 9. Counts of length by frequency.

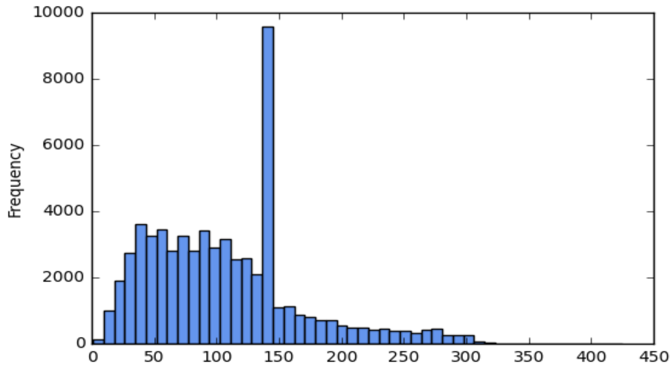


Fig 10. Counts of length by name.

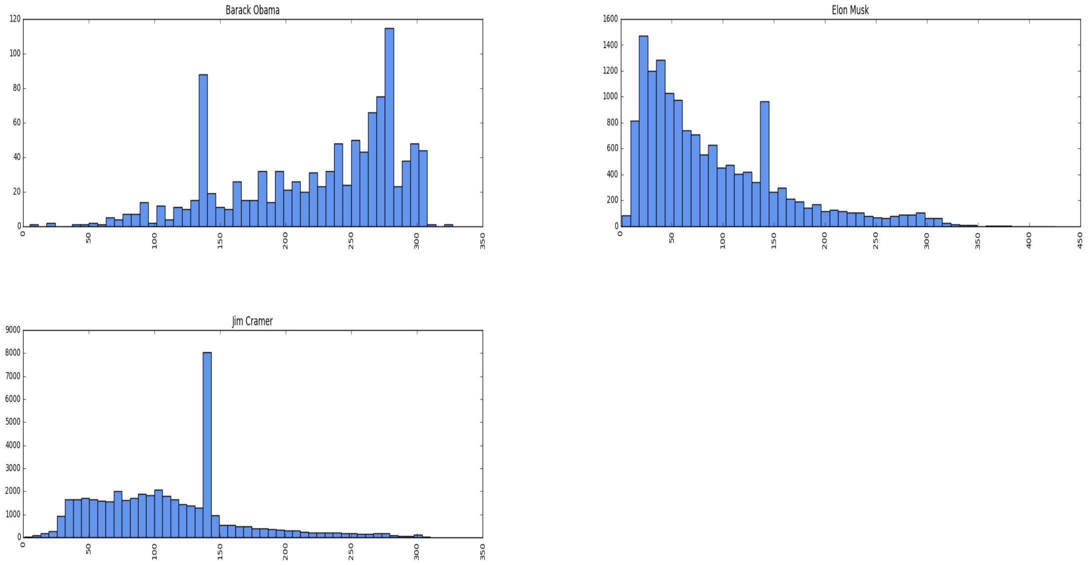


Fig 11. Counts of length by name (Boxplots).

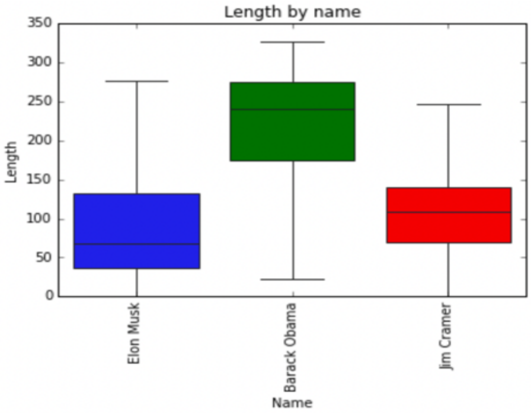


Figure 9 shows the frequency of the length of our sample. Figure 10 and 11 show the length of the tweets of the 3 datasets, with Barack Obama's tweets being the longest and Elon Musk's being the shortest in general.

Figure 12 shows the number of total tweets analysed from each dataset (Jim Cramer 45542, Elon Musk 15203, and Barack Obama 1090). Figure 13 shows the total number of tweets analysed as positive, negative and neutral, and figure 14 shows the total percentage they represent.

Fig 12. Counts of tweets by name.

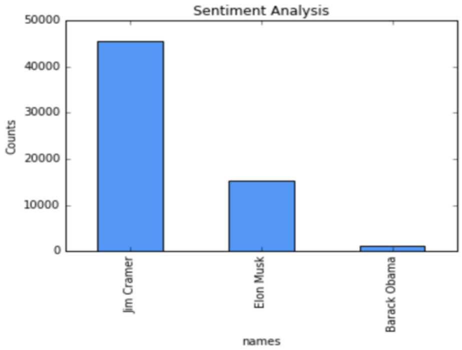


Fig 13. Count of Sentiment Analysis.

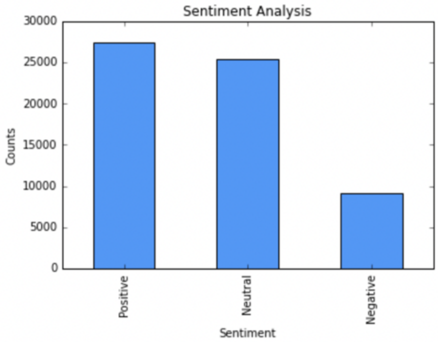


Fig 14. Percentage of Sentiment Distribution.

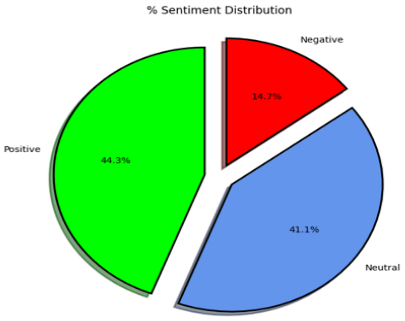


Fig 15. Count of Analysis by name.

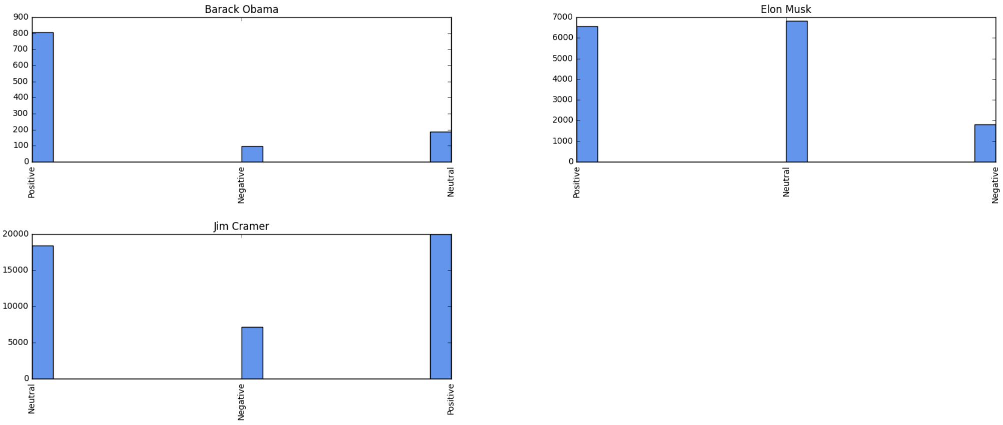


Figure 15 shows the number of positive, negative and neutral tweets for each dataset, noting that Elon Musk creates a greater positive impact than Jim Cramer and Elon Musk. Note that the number of tweets is much lower. The positive impact is much higher than the negative impact for all three datasets, with the neutral impact being fairly even with the positive impact for Elon Musk and Jim Cramer. Figures 16,17,18 and 19 show the total number of polarities and subjectivities calculated to obtain the sentiments along with the impact on each dataset.

Fig 16. Counts by Polarity

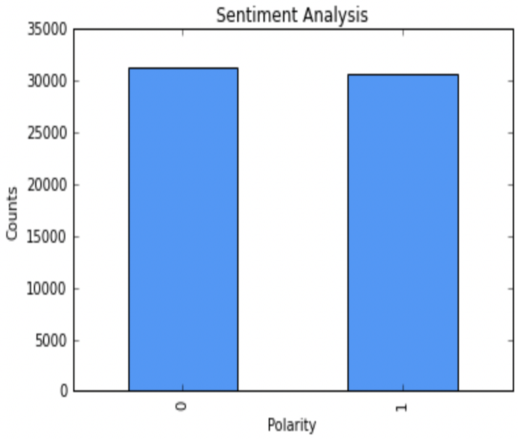


Fig 17. Polarity by name.

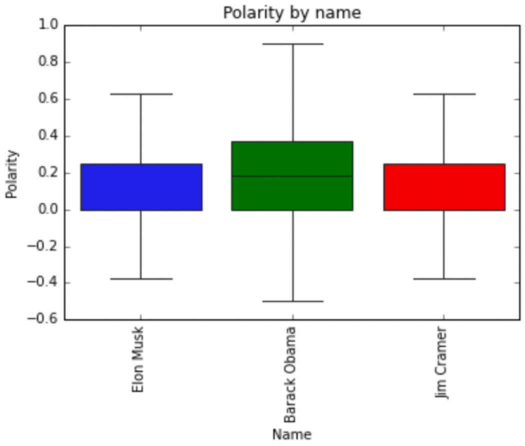


Fig 17. Counts by Subjectivity.

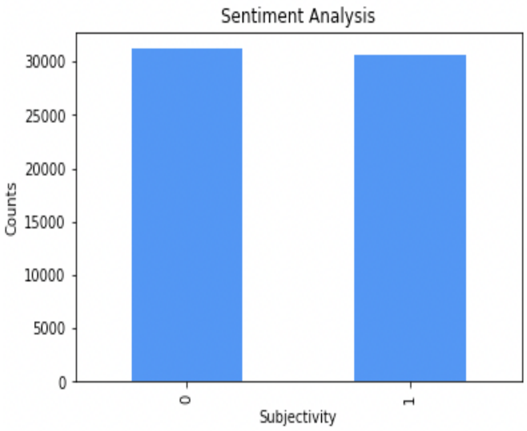


Fig 18. Subjectivity by name.

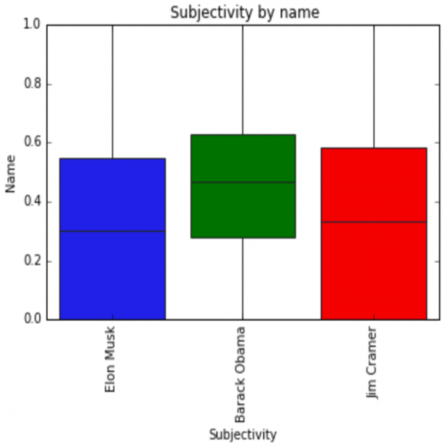


Fig 19. Return by name.

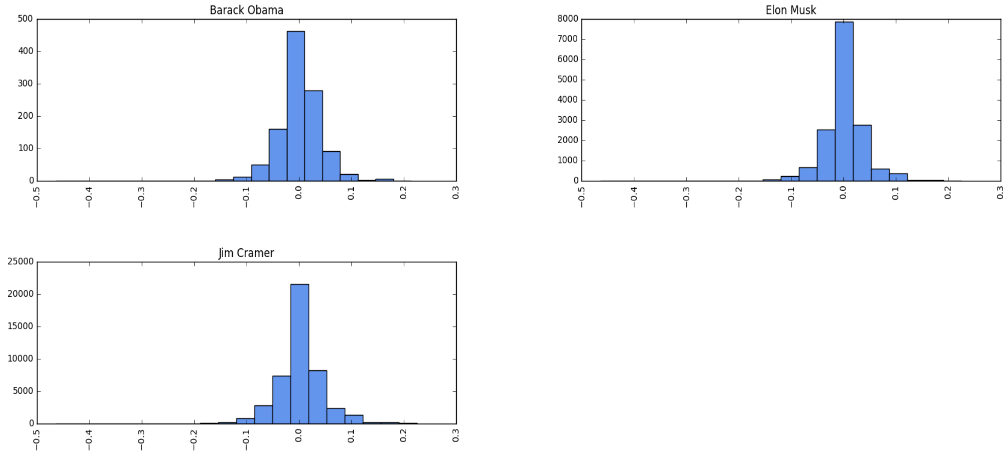


Figure 19 shows the frequency of the generated daily return of BTC/USD for each dataset. We can see in Figure 20 that Jim Cramer's impact on the return is slightly higher.

Fig 20. Return by name (Boxplot).

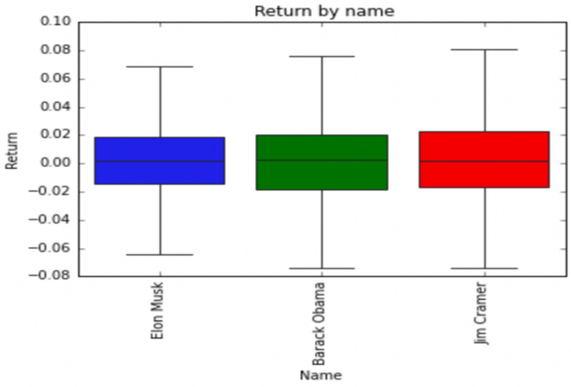


Fig 21. Return by Analysis.

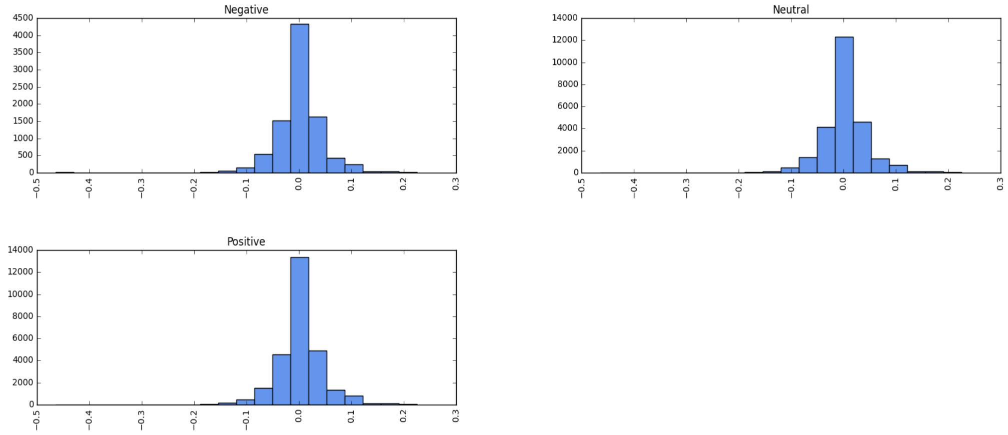


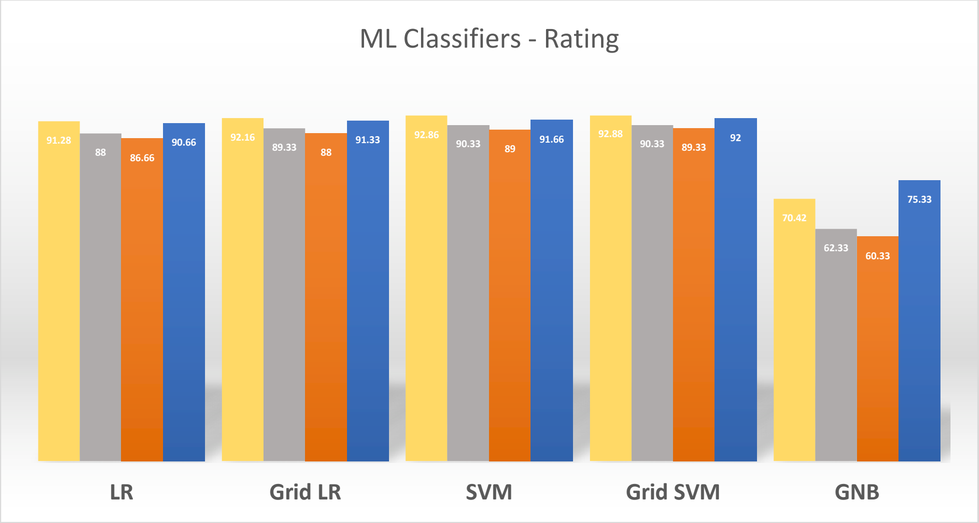
Figure 21 shows the frequency of the return generated as a function of the impact of the total sentiment analysis obtained.

Fig 22. ML Classifiers Performance.

Classifier	Ratings			
	Precision	Recall	F1-Score	Accuracy
LR	90.66%	86.66%	88%	91.28%
Grid LR	91.33%	88%	89.33%	92.16%
SVM	91.66	89%	90.33%	92.86%
Grid SVM	92%	89.33%	90.33%	92.88%
GNB	75.33%	60.33%	62.33%	70.42%

Figure 22 shows the results obtained for the performance evaluation classifiers after applying the machine learning algorithms and their optimisers. Figure 23 shows these classifiers in graph to show a better observation.

Fig 23. Comparison MLA.



Next, we will show in figures 24,25,26,27 and 28 the confusion matrix obtained that define the performance and classification of the algorithms applied.

Fig 24. Confusion Matrix RL.

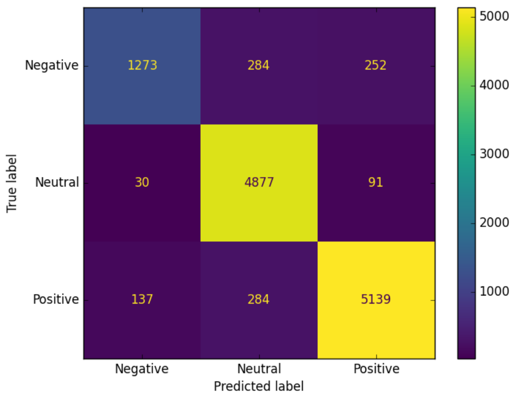


Fig 25. Confusion Matrix Grid RL.

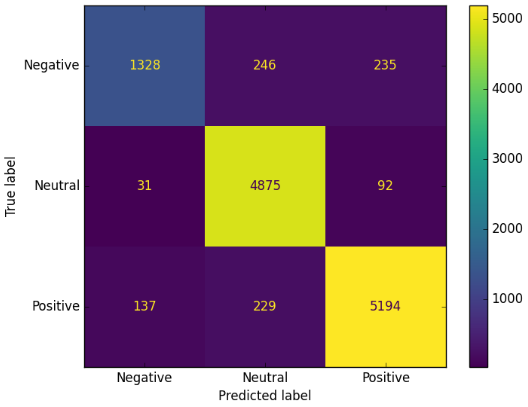


Fig 26. Confusion Matrix SVM.

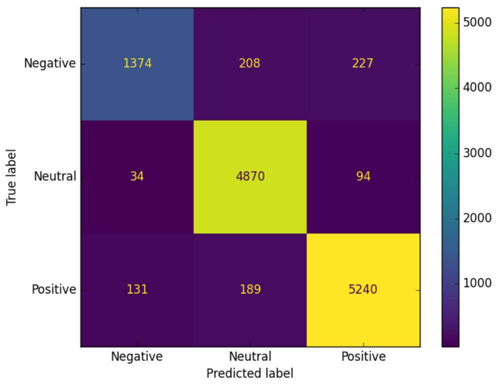


Fig 27. Confusion Matrix Grid SVM.

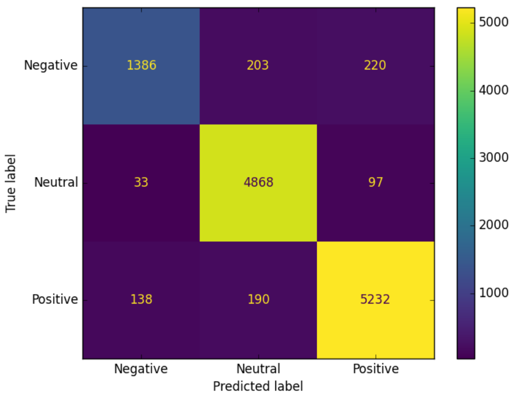
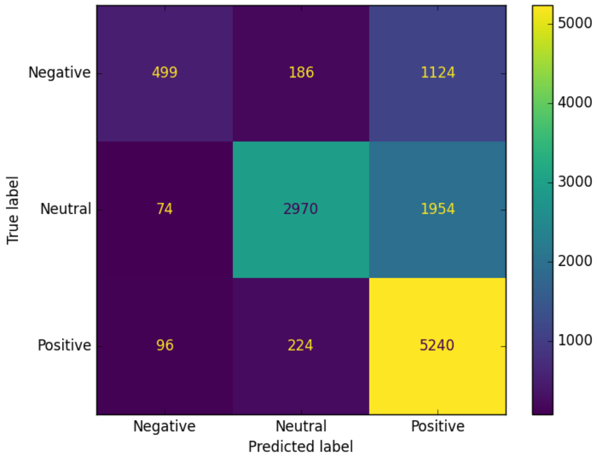


Fig 28. Confusion Matrix Grid GNB.



In the second part of the results we will analyse the predictions obtained by the following models. Figure 29 shows the predictions made for the train and test data (orange line and green line) using LSTM.

Fig 29. Forecast BTC/USD - LSTM.

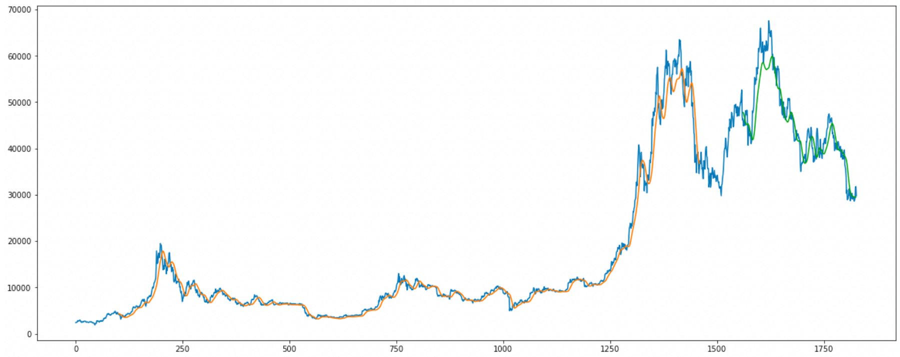


Fig 30. Forecast BTC/USD - ARIMA.

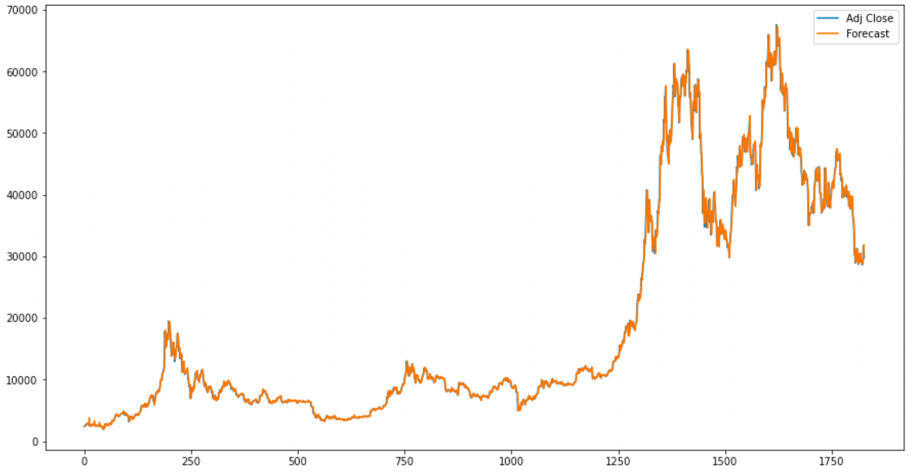
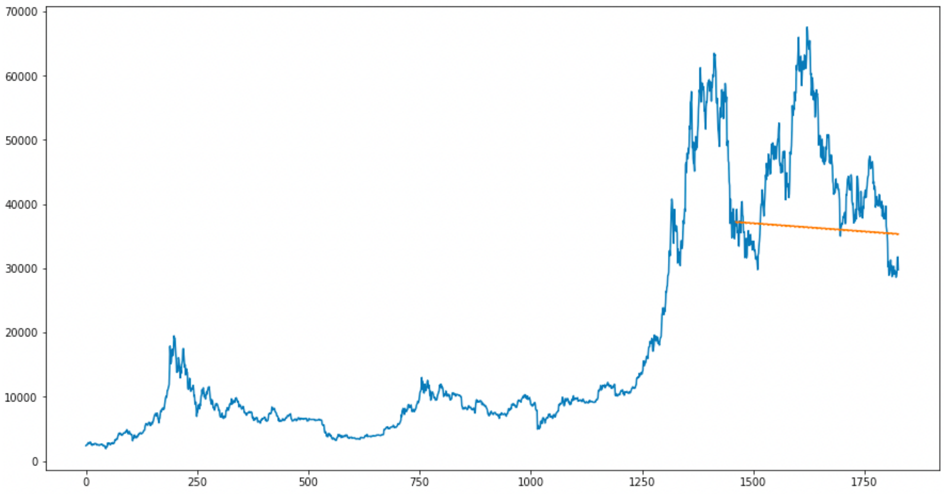


Figure 30 shows the predictions obtained using the ARIMA model for those 5 years of historical BTC/USD data.

Figure 31 shows the predictions obtained for the test data using the Holt-Winter Exponential Smoothing using additive seasonality model.

Fig 31. Forecast Test data BTC/USD – Holt Winter ES.



5 Conclusions

In the first part of our study, we will conclude that Barack Obama's tweets are longer and have a greater positive impact than those of Elon Musk and Jim Cramer. It is important to mention that Barack Obama's sample is much smaller than that of the other datasets. Jim Cramer generates more positive tweets (due to the number of tweets) but lower percentage than Barack Obama. Elon Musk's tweets tend to be neutral or positive, generating more neutral than positive tweets and having a very short sample of negative tweets. The total number of positive tweets for the 3 datasets is 44.3%, neutral 41.1% and negative 14.7%. Likewise, Barack Obama generates the most subjective and polarised tweets. Jim Cramer's impact on BTC-USD returns is greater due to the higher number of tweets he generates. Analysing the evaluation of the machine learning algorithms, we can conclude that Support Vector Machine analyses our samples better, especially when we apply the Grid optimising parameter, obtaining an accuracy of 92.88%. On the other hand, Gaussian Naive Bayes is the one that gives us the lowest accuracy, 70.42%.

In the second part of our study, after analysing the behaviour of BTC-USD for that 5-year period, we can conclude that the model that best predicts the price, according to our study is the ARIMA time series model. It is important to mention that LSTM produces quite good results, but the sample is not very large (1827 records) so ARIMA in this case predicts the price better. The Holt-Winter ES model does not provide us with accurate estimates.

References

1. Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P. (2019). The impact of features extraction on the sentiment analysis. Procedia Computer Science, 152, 341-348.
2. Alsaeedi, A., & Khan, M. Z. (2019). A study on sentiment analysis techniques of Twitter data. *International Journal of Advanced Computer Science and Applications*, *10*(2).
3. Berry, M. W., Mohamed, A., & Yap, B. W. (Eds.). (2019). Supervised and unsupervised learning for data science. Springer Nature.
4. Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of computational science*, *2*(1), 1-8.
5. Bruns, A., & Stieglitz, S. (2013). Towards more systematic Twitter analysis: metrics for tweeting activities. *International journal of social research methodology*, *16*(2), 91-108.
6. Carvalho, L., & Scornavacca, E. (2020). Chatbots an exploratory analysis on the impact of NLP and customer sentiment analysis. ACR North American Advances.
7. Celebi, M. E., & Aydin, K. (Eds.). (2016). Unsupervised learning algorithms. Berlin: Springer International Publishing.
8. Chambers, N., & Jurafsky, D. (2008, June). Unsupervised learning of narrative event chains. In Proceedings of ACL-08: HLT (pp. 789-797).
9. Chinnov, A., Kerschke, P., Meske, C., Stieglitz, S., & Trautmann, H. (2015). An overview of topic discovery in Twitter communication through social media analytics.
10. Chinnov, A., Kerschke, P., Meske, C., Stieglitz, S., & Trautmann, H. (2015). An overview of topic discovery in Twitter communication through social media analytics.
11. Diaz, F., Gamon, M., Hofman, J. M., Kıcıman, E., & Rothschild, D. (2016). Online and social media data as an imperfect continuous panel survey. *PloS one*, *11*(1), e0145406.
12. Dickey, D. A. (2015). Stationarity issues in time series models. *SAS Users Group International*, *30*.
13. Duwairi, R. M., & Qarqaz, I. (2014, August). Arabic sentiment analysis using supervised classification. In 2014 International Conference on Future Internet of Things and Cloud (pp. 579-583). IEEE.
14. Fang, X., & Zhan, J. (2015). Sentiment analysis using product review data. *Journal of Big Data*, *2*(1), 1-14.
15. Goodwin, P. (2010). The holt-winters approach to exponential smoothing: 50 years old and going strong. *Foresight*, *19*(19), 30-33.

16. Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645-6649). Ieee.
17. Habernal, I., Ptáček, T., & Steinberger, J. (2013, June). Sentiment analysis in czech social media using supervised machine learning. In Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp. 65-74).
18. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.
19. John, G. H., & Langley, P. (2013). Estimating continuous distributions in Bayesian classifiers. *arXiv preprint arXiv:1302.4964*.
20. Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?. *Journal of econometrics*, *54*(1-3), 159-178.
21. Mukhtar, N., & Khan, M. A. (2018). Urdu sentiment analysis using supervised machine learning approach. International Journal of Pattern Recognition and Artificial Intelligence, 32(02), 1851001.
22. Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons. b*, *4*, 51-62.
23. OliveiraN,CortezP,ArealN(2013)Onthepredictabilityofstockmarketbehaviorusingstocktwitssentimentand posting volume. In: Progress in artificial intelligence. Lecture notes in computer science, vol 8154. Springer, Heidelberg, pp 355-365
24. Patterson, J., & Gibson, A. (2017). *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc.".
25. Philander, K., & Zhong, Y. (2016). Twitter sentiment analysis: Capturing sentiment from integrated resort tweets. *International Journal of Hospitality Management*, *55*(2016), 16-24.
26. Poornima, A., & Priya, K. S. (2020, March). A comparative sentiment analysis of sentence embedding using machine learning techniques. In *2020 6th international conference on advanced computing and communication systems (ICACCS)*(pp. 493-496). IEEE.
27. Ranjan, G. S. K., Verma, A. K., & Radhika, S. (2019, March). K-nearest neighbors and grid search cv based real time fault monitoring system for industries. In *2019 IEEE 5th international conference for convergence in technology (I2CT)* (pp. 1-5). IEEE.
28. Rathee, N., Joshi, N., & Kaur, J. (2018, June). Sentiment analysis using machine learning techniques on Python. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 779-785). IEEE.
29. Rawat, R., Mahor, V., Chirgaiya, S., Shaw, R.N., Ghosh, A. (2021). Sentiment Analysis at Online Social Network for Cyber-Malicious Post Reviews Using Machine Learning Techniques. In: Bansal, J.C., Paprzycki, M., Bianchini, M., Das, S. (eds) Computationally Intelligent Systems and their Applications. Studies in Computational Intelligence, vol 950. Springer, Singapore.
30. Reyes, A., Rosso, P., & Veale, T. (2013). A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, *47*(1), 239-268.
31. Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 1643-1647). IEEE.
32. Shuai, Y., Zheng, Y., & Huang, H. (2018, November). Hybrid software obsolescence evaluation model based on PCA-SVM-GridSearchCV. In *2018 IEEE 9th international conference on software engineering and service science (ICSESS)* (pp. 449-453). IEEE.
33. Tsytsarau, M., & Palpanas, T. (2012). Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, *24*(3), 478-514.
34. Tyagi, A., & Sharma, N. (2018). Sentiment analysis using logistic regression and effective word score heuristic. *International Journal of Engineering and Technology (UAE)*, *7*(2.24), 20-23.
35. Wang, J. J., Wang, J. Z., Zhang, Z. G., & Guo, S. P. (2012). Stock index forecasting based on a hybrid model. *Omega*, *40*(6), 758-766.
36. Yadav, N., Kudale, O., Rao, A., Gupta, S., & Shitole, A. (2021). Twitter sentiment analysis using supervised machine learning. In Intelligent Data Communication Technologies and Internet of Things (pp. 631-642). Springer, Singapore.
37. Yao, L., & Guan, Y. (2018, December). An improved LSTM structure for natural language processing. In *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)* (pp. 565-569). IEEE.
38. Yıldırım, E. , Çetin, F. S. , Eryiğit, G. & Temel, T. (2014). The Impact of NLP on Turkish Sentiment Analysis . Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi , TBV BBMD , 43-51 . Retrieved from https://dergipark.org.tr/en/pub/tbbmd/issue/22247/238817

39. Yu, Y., Duan, W., & Cao, Q. (2013). The impact of social and conventional media on firm equity value: A sentiment analysis approach. Decision support systems, 55(4), 919-926.