Applying Machine Learning Algorithms and Sentiment Analysis for the Prediction of Stock Prices

Adriana María Trigueros Schmidt

Dublin Business School

This dissertation is submitted for the degree of
Master of Science in Business Analytics

September 2022

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Adriana María Trigueros Schmidt
September 2022

Acknowledgements

I would like to thank my supervisor, Obinna Izima, for providing relevant guidance while choosing my topic of research. Additionally, I would like to thank my professors for providing me with the essential knowledge needed to complete this dissertation and my classmates for providing support. Most importantly I would like to thank my parents for their unconditional support throughout the completion of the course.

Abstract

The prediction of stock prices has been a popular topic for research and its is considered one of the most challenging tasks due to the volatile and noisy nature of stock prices. Furthermore, considering exogenous variables such as the public sentiment regarding a particular task is challenging due to data being in an unstructured format. This work proposes the application of machine learning and deep learning algorithms for the prediction of stock prices, including in the final model the sentiment of public opinions about a company expressed through the social media platform Twitter. First the data is collected through the Twint scraping tool for tweets and Yahoo Finance for historic stock prices. Then the twitter dataset is cleaned, and a sentiment analysis is performed using VADER. The final dataset is obtained by merging the sentiment analysis results of the twitter dataset and the historic stock prices. A rolling window technique is applied to incorporate lagged features in the model, and finally Random Forest, XGBoost, Linear Regression, LSTM and Bidirectional LSTM are tested for the prediction of stock prices and their evaluation is compared with R-Squared and RMSE metrics against a traditional model like ARIMA. The results indicate that the proposed models outperform by far traditional time series models, such as, ARIMA. Additionally, a relation between sentiment scores and closing prices of stocks was found using several machine learning techniques. The performance of the models is tested on five different datasets, from the companies Tesla, Amazon, Google, Meta, and Apple.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 General Overview

Two main ways a company can raise funds for its economic activities are acquiring debt and selling equities (part of the business) to investors. Selling shares is often the preferred choice because it is less risky for companies since they don't have to repay the money with interest. Therefore, their credit record is not threatened if their activities fail to succeed. The shares of a company are bought and sold in the stock market, which has been around for several years and serves as an important channel for companies to raise capital and investors with extra money looking to benefit from a company's growth and distribution of earnings. Furthermore, stock markets play a crucial role in promoting economic growth in a country by facilitating capital flow between investors and the users of money.

However, the mechanism in which the price of a stock is formed is highly complex (Jin *et al.*, 2019). The ever-changing nature of stock prices is mainly due to the several variables that influence the price, ranging from macroeconomic indicators to political factors, technology, markets, and investor sentiment. This constant change can give rise to highly speculative behavior, increasing the risk of investing in stocks even more (Jin *et al.*, 2019), as prices might not always reflect the real value but rather what investors believe. With this kind of risk, investors might experience substantial economic losses, but also the economic activity of a country can be negatively impacted (Jin *et al.*, 2019). One example is the US housing crisis in 2008, where the nominal price of houses rose above the real price, giving rise to a financial bubble that brought catastrophic consequences.

When researchers first started to automate the task of predicting stock prices the most conventional method was to treat the problem as a time series forecasting analysis. A time series variable can be defined as any sequence of observations taken sequentially in time (Loukas, 2020). The most popular models applied by researchers belong to the ARIMA

family: (AR) auto-regressive model, (MA) moving average model, (ARMA) auto-regressive and moving average model, and (ARIMA) auto-regressive integrate moving average model. However all these approaches don't include other influential variables, they focus on the stock price as a time series. The models are uni-variate and operate under the assumption that information from the stocks price past can predict the future behaviour and also assume a linear relationship. Several factors indicate that this traditional approaches are not effective for the task at hand, the main being that due to it's complexity the stock market tends to exhibit non-linear characteristics (Jin *et al.*, 2019).

More recently the successful application of machine learning algorithms on other time series problems has led to many scholars to investigate their performance in the prediction of the stock market. Furthermore, deep learning models have also been incorporated on undergoing research. With their superior ability to capture non-linear relationships and extract complex patterns from data, they have become a trend in stock market prediction (Jin *et al.*, 2019).

In addition, as more advances in the field of behavioural economics and finance are being made, it becomes more evident that investor's psychology plays an important role in their trading strategy (Fataliyev *et al.*, 2021), suggesting that they will not always act in a rational way (buying stocks only if they think their profits will rise), as emotions can produce irrational behaviour, leading to the urgent need of incorporating new variables into the models. One of the main complexities of this task is that the data to represent investor's sentiment is in an unstructured format, therefore additional steps need to be taken to pre-process the data into a format that can be fed to the algorithms. Advances in the field of NLP (natural language processing) have made this possible, and several studies have emerged to test the correlation between the investors sentiment and the stock market movements.

In this research project, the main aim is to effectively predict the stock market movement. The major contributions of this work can be stated as follows:

- Include investor's sentiment as a variable in the model to increase prediction accuracy. More specifically, the proposed method is to use twitter as the main source to capture the sentiment of investors of a specific stock by fetching all tweets that mention the companies ticker symbol. Pre-process the data with NLP techniques and use a pre-trained algorithm specifically designed for social media to capture the polarity on the comments and the final score as the mean of every tweets score.

- Use a rolling window technique to treat the problem as a time series and include lagged features into the data.

- Test and compare the performance of different machine learning and deep learning algorithms against a traditional model like ARIMA to test if they actually perform better.

In a summary, this work aims to apply machine learning techniques to provide investors with a tool that can help them predict stock prices in a more accurate and efficient way. Accurately predicting stock prices plays an important role in the above-mentioned problem by reducing the risk for investors and providing stability to a country's economy (Jin *et al.*, 2019), hence the motivation of this research. The rest of the paper is organized as follows: next subsections of this chapter explore relevant studies performed by other authors; chapter 2 reviews the research questions and rationale for the study; chapter 3 describes the methodology and experimental design of the work; chapter 4 displays the development and implementation of the proposed scheme; chapter 5 describes the results of the implementation and main findings; and chapter 6 provides a conclusion of the work and, limitations of the study, and further studies suggestions.

1.2 Problem Statement

In the past, it was possible to generate stock price predictions that were somewhat reliable and yielded a certain degree of accuracy relying only on experience (Strader *et al.*, 2020). However, nowadays, relying on studies that use technical and fundamental analysis manually is very difficult and quite a tedious task given the size of the market and the speed at which the trading's are executed (Strader *et al.*, 2020). A simple statistical analysis could produce some valuable conclusions. However, with the rise of Big Data, it has become possible to use artificial intelligence to automate the task of predicting stock prices. As a result, many investors have decided to invest in finding methods that produce predictions with the greatest accuracy. These artificial intelligence systems can use a massive amount of information produced in real-time to find patterns in the data and reach conclusions on price variations that are much more accurate than traditional methods.

Several problems are associated to the prediction of stock prices, mainly because as was stated before, there are too many forces that influence its behaviour. More specifically related to the application of machine learning algorithms in an attempt to atomize the task and produce more reliable results, these problems can be summarized as follows:

- Data from stock prices is non-stationary, posing a challenge as it is in nature a time-series problem.

- The volatility of the stock prices gives rise to the need of more complex algorithms that are able to capture non-linear relationships.
- The need to capture the sentiment of investors to produce better results, also gives rise to the need to apply several techniques that can capture information from sources that reflect investors moods and translates them into a structure that can be used by the algorithms.

1.3 Literature Review

A principal characteristic of the stock market in every region of the world is that it encompasses a vast amount of wealth (Strader *et al.*, 2020). The value of global equities has surpassed trillions of dollars over the years, a value that increases over time. As long as markets exist, investors will always try to find ways to beat the market. It is an area of great interest, due to the attractive result of becoming richer. Ever since Machine Learning started to be applied to the task of predicting stock markets, the number of research papers started to increase dramatically. A numerous amount of research papers are available and this chapter aims to discuss the most relevant contributions to the field.

In 2017, (Chong *et al.*, 2017) test the performance of deep neural networks for the prediction of stock prices, using the stock returns of KOSPI (a major stock market in South Korea). In their data collection process they gather data of stock prices every five minutes from a period of January 2010 to December 2014 and then calculate the returns. Their results indicate that deep neural networks highly outperform traditional methods. A factor that might influence the positive results is the robust method chosen by the authors to select the correct feature set. They generate 7 training sets, consisting of different representation methods such as PCA and autoencoder using the lagged stock returns as an input. They later test the predictive power of each training set by employing classification techniques. Few other papers apply so many steps to carefully select the final data that will be fed to the models. However, their predictions only apply for short-term decisions, since the values consists of returns every 5 minutes, even if the models perform well, they can only predict very short periods, therefore only high-frequency traders can benefit from it.

Apart from deep neural networks, several other models have been applied for the task at hand. In 2018, (Khan *et al.*, 2018) test the performance of linear regression algorithm on the prediction of stock prices. They use information of historic prices extracted from Yahoo Finance of several technology companies, such as, Google and Microsoft. Their results show very low RMSE scores, indicating that the model performance is good. One of the main advantages of using a linear regression model is that they were able to identify how each

feature relates to the closing price, as this model provides high interpretability. However, the study also has several drawbacks, the main being that they don't compare the results among stocks from companies with similar characteristics, therefore it is hard to tell if their methodology would work well with datasets containing stock prices of different companies. The volatile nature of stock prices limit the predictive power of a model like linear regression that assumes the shape of the underlying function is linear, and the good performance scores would most likely indicate that the model is overfitting.

One of the main limitations of studies that use only historic data of the stock price as an input is that they operate under the assumption that the price of the stock is able to capture it's real value. However, in reality, this assumption is very unlikely to hold, as it has been stated by many researcher's that the main reason behind the high volatility of the stock market is that there are many variables that influence it's behaviour. Furthermore, as it has been stated by scholar's studying behavioral finance and economics, emotion influences the way humans make decisions, a type of information that is clearly not captured by historic data on price. The next paragraphs discuss academic papers that use other kind of data as an input.

There are several research papers that either make use of historic stock prices data or data from fundamental analysis, but few use a combination of both. In the paper (Nti *et al.*, 2020) makes a systematic literature review of machine learning applications for stock market predictions and finds that out of the total 122 papers that were selected for the review, 66 percent focus on technical analysis, 23 percent focus on fundamental analysis and only 11 percent use a combination of both. The following paragraphs discuss some of the relevant literature that combines the approaches for stock market prediction.

In 2016, (Pagolu *et al.*, 2016) use information from twitter, news, and historic prices to try to find if there is an existing correlation between the sentiments extracted during the process, which are classified into positive, negative and neutral, and the price of the stock. Their conclusion is that the study finds a strong correlation between the up or downward movement in stock prices to the public sentiments expressed in opinions on twitter or news about a certain company. One of the main drawbacks of this study is that they create their own sentiment classifier by training the classification models, labelling the extracted data with tweets or news according to the rise or fall on the stock price that day. In reality, it has not been proved with certainty that a negative sentiment about a company expressed through social media will always reflect a positive or negative movement on the stock's price.

In 2019, (Jin *et al.*, 2019) use LSTM with a combination of variables from historic prices of stocks and investors sentiment to predict the trend of the stock price. Their method proposed to extract investor's sentiment is based on a CNN (convolutional neural network) combined with the Word2vec model to learn the semantics of all the comments made on the

market in an unsupervised day and classify them into bearish or bullish. The final sentiment index is represented by the amount of bearish or bullish comments on a day. Additionally, they propose to decompose the time series data of historic prices with EMD (empirical decomposition mode), and ultimately feed the combined data to an LSTM model with an extra attention layer. Their results show that when the sentiment index of investors is added to the data model, LSTM performs better.

Chapter 2

Research Main Overview

2.1 Research Questions

After reviewing the main methodologies and models that have been previously applied to the task at hand it's clear that the challenge of predicting the prices of stocks has not been fully mastered yet and there is a lot of research to be done. The main contribution of this paper will to test if an endogenous variable, such as, investor's sentiment can improve the performance accuracy of the models. Furthermore, different types of algorithms will be applied to compare their performance and find which ones perform better. The main questions that this research aims to answer are the following:

- Is the sentiment of investors about a company in each day correlated to the opening price of the next day?
- Do machine learning algorithms (Random Forest, Linear Regression, and XGBoost) perform better than traditional models like ARIMA for predicting stock prices ?
- Do deep learning algorithms (LSTM and Bidirectional LSTM) perform better than traditional models like ARIMA for predicting stock prices?
- What type of models, machine learning or deep learning, perform best for the prediction of stock prices?

2.2 Research Objectives

The main objective of this research paper is to test different machine learning techniques for the prediction of stock prices and compare them to traditional methods that will be used as a

baseline, to test if they actually perform better. Additionally, the aim is to apply different techniques to try to find if there is an existing correlation between the sentiments of the public about a company expressed through the social media twitter and the closing price of a stock the next day. The specific objectives of this research paper can be summarized as follows:

- Improve prediction accuracy of stock price.
- Explore the relationship between sentiment of investors expressed through twitter and the closing prices of a stock.
- Review methods to extract tweets of a certain company.
- Explore the performance of machine learning algorithms against traditional models for the prediction of stock prices.
- Explore the performance of deep learning algorithms against traditional models for the prediction of stock prices.
- Compare the different proposed models performance on different datasets relating to several companies to see if their performance is able to hold.

2.3 Proposed Scheme

This section of the chapter is aimed to provide a general description of the proposed methodology, which will be discussed in more detail in the next chapter. It consists of several steps, similar to the cross-industry standard process for data mining (CRISP-DM). Figure **??** outlines the overall methodology proposed in this research project for the task of predicting stock prices.

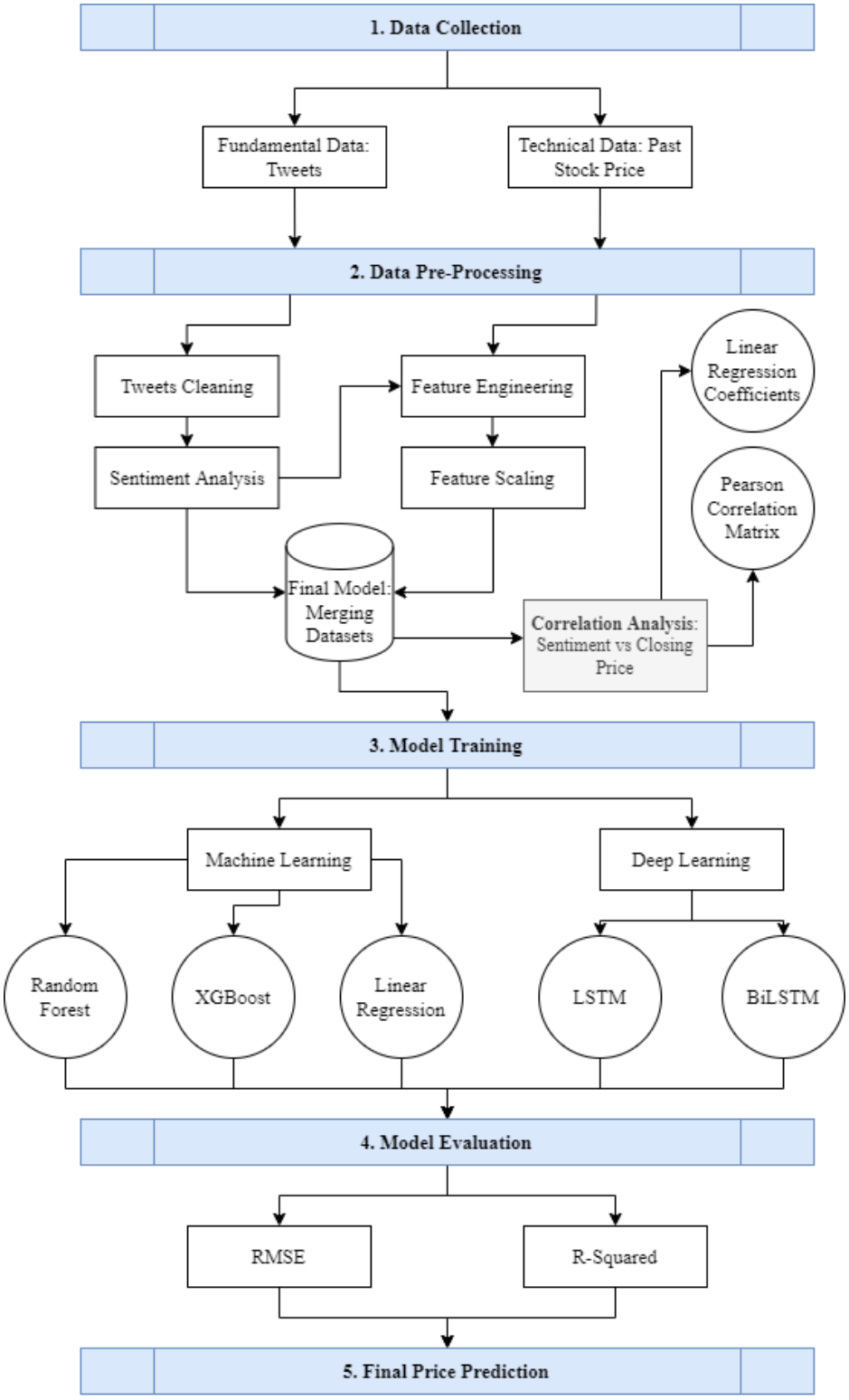


Fig. 2.1 Flowchart of Proposed Methodology

Chapter 3

Experimental Design

The methodology proposed by this work consists of several stages that will be discussed in detail over this chapter. It consists of a five steps approach: from the collection of data, pre-processing steps of data, model training, model evaluation and interpretation of results.

It is important to understand that this methodology is not proposed for long-term predictions of the stock market, rather it can serve as a tool for investors to update the positions of their portfolios stocks on a daily basis, as it has been developed for the prediction of short-term movements in the market. The reason is that, as it was mentioned before, the stock market is highly volatile and complex, making it a very hard task to predict in the long run.

Additionally, to prove if the proposed methods can be considered useful for investors that trade on a daily basis, they will be compared with a baseline model. The baseline used for this project is a traditional ARIMA model, which is one of the most popular models applied in the task at hand as it serves specifically for time series problems. This simple method will be compared with the machine learning models proposed in this paper. Additionally, to try to determine the relation between sentiment expressed on social media and closing prices, several techniques will be applied.

3.1 Data Collection

In this project two different datasets will be used for every stock that is being analyzed. The stocks selected for the purpose of this project are the following: Google, Meta, Amazon, Tesla, and Apple. The first dataset obtained consists of twitters mentioning the companies keyword listed on Yahoo Finance, for example, the tweets from Tesla are fetched using the TSLA keyword. This data will be used to extract the sentiment of investors on a given day. The second dataset consists of the historical stock prices of the company, and it will be used

to extract patterns from past movements of the stock's price. In this section of the chapter the methods applied to collect the data and its characteristics are discussed.

3.1.1 Twitter Datasets

The extraction of the twitter dataset was done through the Twint scraping tool. Twint is a tweet advanced scraping tool that was designed specifically to extract tweets from Twitter (Paudyal, 2020). The other method explored for the extraction of this dataset was the twitter API and Tweepy module, however, one of the major advantages of the chosen method is that no authentication or API is required. Additionally, the set-up of Twint is quick and less complicated, also providing no limit on the amount of data that can be scraped, whereas the twitter API allows a maximum of 3,200 tweets

The tweets were fetched from the social media site using the company's keyword listed on Yahoo Finance, since most traders use this to talk about their opinions on the company that they are investing on. The tweets extracted range from July 1st 2015 to June 30th 2022, to analyze 7 years of data. The Twint tool fetched every tweet per day mentioning the company's keyword on the specified date range. The following table **??** contains a summary of each dataset obtained containing the tweets:

Table 3.1 Twitter Datasets Description

Company	Keyword	Date Range	Total Number of Tweets	Total Number of Rows
Tesla	TSLA	7/01/2015–6/30/2022	35,977	1,394
Google	GOOG	7/01/2015–6/30/2022	40,459	1,606
Meta Platforms	META	7/01/2015–6/30/2022	36,438	1,260
Amazon	AMZN	7/01/2015–6/30/2022	45,686	1,450
Apple	AAPL	7/01/2015–6/30/2022	46,864	1,416

The following figure 3.1 displays an example of the initial twitter dataset containing all the tweets mentioning the company, in this example, Tesla, with the first 5 columns and 3 rows. Before producing the table, several unnecessary columns such as, "id", "conversation-id", "created-at", "timezone", "place", "hashtags", etc. were dropped.

As it can be observed in figure 3.1, the initial twitter dataset contains several problems. For example it contains dates in which the stock market is closed and the dates would not match with the stocks dataset. It also comes in an unstructured format which can't be feed to the algorithms. To address this issues, several pre-processing steps were taken which will be discussed in the next section of the chapter.

	Date	0	1	2	3	4
0	7/3/2015	Top price gainers http://t.co/JaPBkU75Az $MKL...	XLP Consumer Staples Select Sector SPDR Invest...	XLK Technology Select Sector SPDR Yield http:...	Tesla could have a problem with one of the bes...	@darrenheiberg #Tesla #tsla beats quarterly #e...
1	7/4/2015	KRFT Kraft Foods Group, Inc. Tick http://t.co...	TSLA Price Update: $280.02 #Tesla via @Flutter...	Tesla could have a problem with one of the bes...	#Tesla is at another crossroads. #$TSLA must ...	Tesla is at another crossroads (TSLA) http://...
2	7/5/2015	Early movers: hnt tsla pot an fdo ge & mor...	Tesla is at another crossroads (TSLA) http://...	@howardlindzon add $TSLA is almost as big as A...	TSLA Tesla Motors Inc. Bid http://t.co/4jo6rg...	Tesla could have a problem with one of the bes...

Fig. 3.1 Tesla - Initial Twitter Dataset

3.1.2 Stock Prices Data

The stock prices dataset was downloaded directly from yahoo finance, a site that provides the entire history of the stock prices opening, close, adjusted close and volume on a given day. The following table 3.2 contains a summary of each dataset obtained by company with the information about the stock's prices.

Table 3.2 Stock Datasets Description

Company	Keyword	Date Range	Total Number of Rows
Tesla	TSLA	7/01/2015–6/30/2022	1,762
Google	GOOG	7/01/2015–6/30/2022	1,762
Meta Platforms	META	7/01/2015–6/30/2022	1,762
Amazon	AMZN	7/01/2015–6/30/2022	1,762
Apple	AAPL	7/01/2015–6/30/2022	1,762

This dataset does not require as many pre-processing steps as the twitter dataset, however, some unnecessary columns were dropped to maintain only the relevant variables for the final model. The following figure 3.2 displays an example of the initial stocks dataset downloaded from Yahoo Finance, containing the first 5 rows of the company Tesla.

	Date	Open	High	Low	Close	Adj Close	Volume
0	7/1/2015	54.222000	54.523998	53.570000	53.830002	53.830002	10506000
1	7/2/2015	56.040001	56.490002	54.661999	56.004002	56.004002	35819500
2	7/6/2015	55.776001	56.338001	55.259998	55.944000	55.944000	20609500
3	7/7/2015	55.000000	55.040001	52.153999	53.576000	53.576000	30525500
4	7/8/2015	51.863998	52.160000	50.862000	50.992001	50.992001	31105500

Fig. 3.2 Tesla Initial Stock Dataset

3.2 Data Pre-Processing

The second section of this chapter discusses the data pre-processing steps of the methodology.This section of the chapter explains in detail every step that was taken to prepare the data for the models.

3.2.1 Tweets Cleaning

The data is initially in an unstructured format when extracting the tweets related to a stock on a given day through the Twint scraping tool. This unstructured format refers to the data not having the structure usually expected from it, meaning rows and columns with well-defined features. The text has a linguistic structure suited for communication between people who can navigate through misspelled words, faulty grammar, uncommon abbreviations, random punctuation, synonyms, and, most importantly, context to deduce meaning. For computers to understand text, it is necessary to transform it into a structured format that can be fed to the several algorithms applied in this project.

With the use of the NLTK (natural language toolkit) the data containing the twitters in an unstructured format was transformed into a structured format with several steps. The method used in this project is the bag of words approach, in which each word of the tweets will represent a token or features. The cleaning steps applied to the data to convert it into a structured format are the following:

- Pulling data out of HTML and XML files using the Beautiful Soup Python library.
- Remove all possible hashtags, @mentions, URLs, etc. and substituting with white space using regular expressions.

- Removing all non-alphabetic characters such as numbers and punctuation's and substituting with white space using regular expressions.
- Break the reviews into individual words using the word tokenize function of the NLTK library.
- Converting all words into lower case using the lower method.
- Removing all the stop words.
- Take care of plurals and synonyms with lemmatization.

Initially the words in every tweet are split into individual tokens for the cleaning process. After they were retokenized (put back together) to form reconstructed sentences of the tweets in their cleaned version. The following figure 3.3 represents an example of how the final twitter dataset looks after pre-processing of Tesla.

	0	1	2	3	4
0	rise robocar uber ceo say buy k self driving c...	tesla motor receives sell rating bank america ...	weekend review video greece spy iwm qqq aapl g...	tesla could problem one best thing owning car ...	tesla motor inc reiterated rating bank america...
1	polgarstocks tsla velocityshares daily x vix s...	tesla stock drop concern baked big opportunity...	ziop investor opinion ziop ilmn tsla xle ziop ...	tsla ceo sun valley tell model x well received...	tsla velocityshares daily x vix short term etn...
2	everyone ready market get bid globex shot tomo...	commented stocktwits tesla actually taken look...	tsla investor use faith investing strategy mea...	believe look really cool photo fun	love earnings season guy rooting money mnk aap...

Fig. 3.3 Tesla - Final Tweets Dataset

3.2.2 Sentiment Analysis

After cleaning the data, it is ready to perform a sentiment analysis to determine the investor's sentiment on a given day. Sentiment analysis basically means the deduction of the sentiment (emotion, moods) of a given text. It is a Natural Language Processing (NLP) technique that

allows to complete the task of analyzing sentiments in a text computationally. There are currently two solutions available for sentiment analysis:

- Rule-based system: make use of a list of lexical features to classify a particular set of words or text with the use of pre-defined vocabularies with words that are already labeled as per their semantic orientation.
- Machine learning systems: make use of machine learning models to predict the classification of a new text, using a dataset that contains text that has already been classified, generally into positive or negative.

For this project a rule-based system called VADER Sentiment Analyzer was chosen, particularly because the dataset containing the tweets has not been labeled and it is a system that was particularly created with the aim of analyzing text from social media, providing several advantages from other packages, since things like slangs and emoticons are incorporated in its vocabulary and manipulate the intensity of the scores. By including these features, it will be possible to obtain a more reliable classification in the sentiments of the tweets. Since VADER is a pre-trained model for classifying tweets that uses a list of lexical features that already have been labeled as per their semantic orientation, to obtain the sentiment scores the step was to simply download the necessary packages and get the compound scores which are a combination of the negative scores (stating how negative the tweet is), positive scores (stating how positive the tweet is), and neutral score (stating how neutral the tweet is). The closest the compound score is to -1, the more negative it is, the closest it is to 0 the more neutral it is and the closest it is to +1 the more positive it is. To obtain a single score for every day, the mean of the compound scores for all the tweets fetched on a given day is used as a final score. The following figure 3.4 displays the final mean compound score assigned for each day of the company Tesla:

3.2.3 Feature Engineering

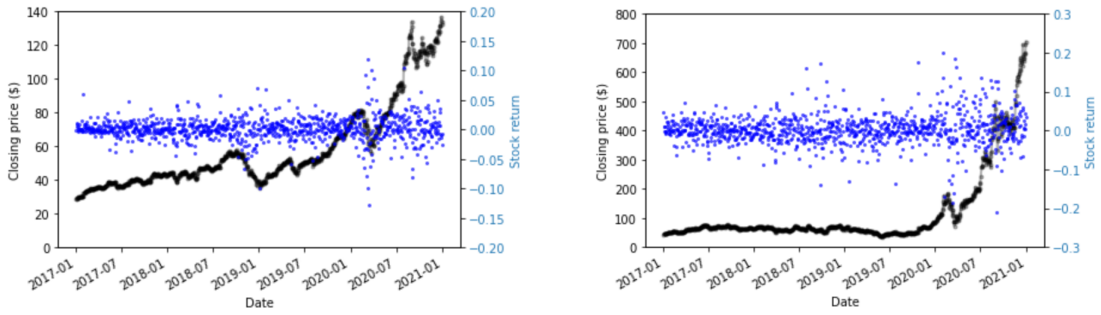
Calculating Stocks Return

An important step taken in the stocks dataset is to calculate the stocks return on a given day. This can be computed simply as the percentage of change in the stock's closing price from one day to another. The main reason is that providing the stocks closing price of previous days alone as features can result in bad performance and its highly unreliable due to it's non-stationary and unpredictable characteristics. Figure 3.5 shows an example of plotting the closing prices and returns of the companies Apple and Tesla where in both cases can

Date	Mean Compound Score
2015-07-06	0.155000
2015-07-08	0.066000
2015-07-09	0.022308
2015-07-10	0.160000
2015-07-14	0.088750

Fig. 3.4 Tesla - Sentiment Scores

be observed that while the closing price remains volatile and unsteady, the returns are more stationary in time.



(a) a. APPLE (b) b. TESLA

Fig. 3.5 Closing Price vs Returns

Dropping Unnecessary Variables

Another step of the feature selection process was to drop some unnecessary columns to feed the model only the most relevant variables. From the stocks dataframe the columns 'Open', 'High', 'Low' and 'Close' were dropped and a column to represent the stock's return that was caltulated on the previous step was added.

3.2.4 Merging Datasets and Feature Selection

To arrive to the final dataset, the stocks data with the dropped and added variables and the sentiment scores as the mean compound score for each day were merged together using the pandas dataframe merge function. From the sentiment analysis another column was added to represent 1 for positive sentiment, 2 for neutral sentiment and 3 for negative sentiment. Additionally the date column was set to index. The following figure 3.6 displays and example of the final dataset for the company Tesla:

Date	Adj Close	Volume	Sentiment_Score	Sentiment_Label	%_Change
2015-07-08	50.992001	31105500	0.066000	1	-0.048231
2015-07-09	51.584000	16670500	0.022308	1	0.011610
2015-07-10	51.830002	13054500	0.160000	1	0.004769
2015-07-13	52.431999	14801500	0.000000	2	0.011615
2015-07-14	53.130001	9538000	0.088750	1	0.013313

Fig. 3.6 Tesla - Final Dataset

3.2.5 Rolling Window Technique

When predicting a time series variable it is important to consider the auto-correlation factor. This basically means that since it is a sequential variable, each value is affected by the values just preceding it. For example, if when predicting the traffic level at a certain time, it is logical to think that if the traffic is high at time t, it will remain high for the next hour. The same logic applies for predicting the closing price of a company's stock. In a regular regression, only the values at values at time t are considered, while due to auto-correlation, x(t-1), x(t-2), etc. will also affect the outcome (Perera, 2018).

To address this problem the rolling window technique was applied. The idea behind this technique is to feed into the algorithm not only the features at time (t), but also at previous times depending on the size of the window.

3.2.6 Training Test Split

Before modelling an additional step is to split the data into training and test, so that the algorithm can train on some data and the be evaluated with the test set. For this project the data was split into 90 percent for training and 10 percent for testing.

3.2.7 Feature Scaling

The final pre-processing step of the data before feeding it to the models is to scale it so that every feature can be considered with equal weights in the model. This is done because some values have different means and variance, for example, the percentage of change has values that will range from 0 to 1, while the sentiment labels have values that range from 1 to 3. This might cause the model to consider the features with the highest values more important for the prediction. To avoid this all the data was scaled using the MinMaxScaler function from sklearn.

3.3 Model Training

The next step in this project is to feed the final dataset to several models so that it can be trained for predictions. The chosen models range from machine learning and deep learning algorithms and will be discussed further in this section.

3.3.1 Machine Learning Models

Machine learning is a sub-field of artificial intelligence built on the idea that systems can learn from data, identify underlying patterns from it and finally make decisions without following explicit instructions, therefore requiring little or no human interaction (S and R, 2021). This data analysis technique is used in this project to automate the process of predicting stock prices. The models selected for this project are Random Forest, XGBoost and Linear Regression.

Random Forest

Random Forest is a popular ensemble model, which means that it relies on the prediction of several models, instead of just one. Random Forest is an ensemble of decision trees (S and R, 2021). It builds multiple decision trees by first creating multiple training sets using the bootstrap sampling technique (sampling with replacement), meaning that it can use a sample the same observation more than once.

For a classification problem, random forest predicts the class outcomes using the majority vote, which means that it picks the prediction of the majority of trees. For a regression problem, which is the application of this project, random forest picks the prediction using the mean value of the predictions of all the trees.

The steps followed to implement the model are the following: creating an instance of the Random Forest Regression model, fitting the model using the training sets, making predictions with the X-test set features and finally evaluating the model by calculating the performance metrics using the y-test set and the predictions.

XG Boost

XGBoost stands for Extreme Gradiant Boosting and it is the implementation of an algorithm developed by Tianqui Chen on 2014 (Biswas *et al.*, 2021). It is also an ensemble model built on multiple decision trees similar to the Random Forest model, however, the difference is on how the trees are built and combined. While Random Forest builds trees sequentially using bagging to build decision trees from random bootstrap samples of the data, XGBoost builds treed in parallel using boosting (improving a weak model by combining a series of other weak models), therefore generating a collectively strong model.

The main difference between the two ensemble models XGBoost and Random Forest is that Random Forest minimizes variance (chance of overfitting) with bagging technique and XGBoost minimizes bias (chance of under-fitting) with boosting technique.

The steps followed to implement the model are the following: creating an instance of the XGBoost Regression model, fitting the model using the training sets, making predictions with the X-test set features and finally evaluating the model by calculating the performance metrics using the y-test set and the predictions.

Linear Regression

Linear Regression is a mathematical model used to forecast the relationships between a dependent variable and a single or several independent variables (Biswas *et al.*, 2021). This model has the disadvantage that it assumes the shape of the underlying function of the data that is trying to predict is linear, which in the case of stock price prediction it will fail to capture non-linear relationships in the data. However for the purpose of this project it will be implemented for comparison with other models performance.

To better understand this model, it is useful to visualize its equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon_i \tag{3.1}$$

where:

Y is target variable (opening price of stock)

Betas are the coefficients of each independent variable

X1 is independent variable 1 (closing price of stock)

X2 is independent variable 2 (volume of stock)

X3 is independent variable 3 (Sentiment Score)

X4 is independent variable 4 (Sentiment Label)

X5 is independent variable 5 (Percent of Change Stock Price)

The linear regression model can be seen as an optimization problem that tries to find the value for predictions with the lower squared error, which means finding the values of the independent variables with betas at which the squared loss function is minimized. This is achieved through the Stochastic Gradient Descent algorithm, which in the case of having multiple dimensions (independent variables), it takes the partial derivative (slope) taking the direction of the minima by subtracting the slope from x at each step.

The steps followed to implement the model are the following: creating an instance of the Linear Regression model, fitting the model using the training sets, making predictions with the X-test set features and finally evaluating the model by calculating the performance metrics using the y-test set and the predictions.

3.3.2 Deep Learning Models

Deep learning as machine learning is a sub-field of artificial intelligence that uses algorithms to find hidden patterns in the data and atomize tasks to rely on no human interaction. However, deep learning uses a different approach to learn from the data and it can be regarded as a complex mathematical and sophisticated evolution of machine learning algorithms (Wolfewicz, 2022).The deep learning algorithms analyze the data with a logic structure that is similar to how human make decisions (Wolfewicz, 2022), by using a layered structure of algorithms called ANN (artificial neural network). The design of the ANN is inspired on the biological network of human brains, leading to results that are more capable than machine learning models (Wolfewicz, 2022). The models selected for this project are LSTM and BiLstm.

LSTM

LSTM is deep learning model that stands for long short term memory. LSTM is a type of recurrent neural network that was specifically designed to overcome the long-term dependency problem of RNN's. This basically means that an LSTM is capable of retaining longer

sequences of information than a traditional neural network, making it attractive for problems where the predictions don't depend only on recent information, such as the case of time series problems and stock price prediction.

The output of an LSTM basically depends on three things:

- Cell state: current long-term memory.
- Hidden state: output at previous point in time.
- Input data of current time period.

To control how the information flows across the network, LSTM uses a series of gates. A forget gate to control how much information leaves, an input gate to determine how much of the new information will be added to the networks, and an output gate which determines the way the information will be stored in the network.

The steps followed for the implementation of the LSTM are: reshape the features data vertical vectors for the model, define the LSTM model structure by adding three layers including a dropout after each LSTM layer (fraction of nodes that will be dropped in each epoch) to prevent overfitting, compile the model using the Adam optimizer and use the mean square error as a loss function, and finally train the model using 25 epochs. After that, the model is evaluated using data of features and target from the test sets, the model uses the X-test data to make predictions, and then the final evaluation is made by calculating the performance metrics based on those predictions and the real values of the Y-test data.

BiLSTM

Bidirectional long short term memory is similar to an LSTM, however, it allows the input layer to blow in two direction, from past to future or from future to past, and in a traditional LSTM it is only possible to choose one direction.

3.4 Evaluation

To select the best models for deployment they have to be evaluated on the test set to select the one that performs better. This section of the chapter discusses the evaluation methods and scores that were used in the project to select the best model. As the task at hand is a regression problem the evaluation metrics chosen are the Root Mean Squared Error, Mean Absolute Percentage Error and R-squared.

3.4.1 RMSE

The RMSE metric stands for root mean squared error. It can be defined as the square root the the averaged squared error, by the following formula:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{j=1}^{n}(y_j - \overline{y_j})^2} \tag{3.2}$$

For this metric the smaller the value is, the better the model is performing as it means that the average magnitude of the error in the predictions is low.

3.4.2 R2 Score

Sometime the RMSE metric can be difficult to interpret as it depends on the scale of the target variable. For this reason, the Coefficient or Determination R-Squares was also chosen as a metric to evaluate the models, as it provides a relative value that ranges from 0 to 1. This metric represents a comparison between the residual sum of squares and the total sum of squares. The formula can be defined by:

$$R^2 = 1 - \frac{SUM(y_i - \overline{y_i})^2}{SUM(y_i - y_{avg})^2} \tag{3.3}$$

For this metric, the results can be interpreted as the closer the value is to 1, the better the model performance is. It tells in relative numbers, how close the model is to fitting the data that has been feed to it.

Chapter 4

Results and Discussion

This chapter discusses the results obtained from the selected algorithms. Section 4.1 discusses the results of the correlation matrix and linear regression technique to try to find if the sentiment of investors is related to the closing price of stocks, to find an answer for research question 1. Section 4.2 and 4.3 discusses the results of machine learning and deep learning algorithms for the prediction of closing prices and the results are compared with the performance of an ARIMA model which is used as a baseline, to find an answer for research question 2.

4.1 Correlation Between Sentiment and Closing Price

4.1.1 Sentiment Score vs Closing Prices

One of the ways to try to determine if the sentiment score is related to the closing prices of stocks is to simply plot one against the other and observe their relation. The following figure 4.1 displays the plot of the five different companies being analyzed of the mean compound score on the right side and the closing price of the stock on the left side of the plot. Theoretically as the mean compound score approaches negative one, the negativity in the tweets increases, so one should observe a fall in the price of the stock.

As can be observed in the graphs above, it is hard to tell the relationship between the sentiment score and the closing price of the stock as the majority of tweets concentrate on positive values, which means that much more positivity has been captured in the sentiment analysis. In some cases it is possible to observe that right before there is a fall in the price there are also some negative tweets that have been posted, however, it is not so evident and further research needs to be done to determine the exact relationship between the two variables.

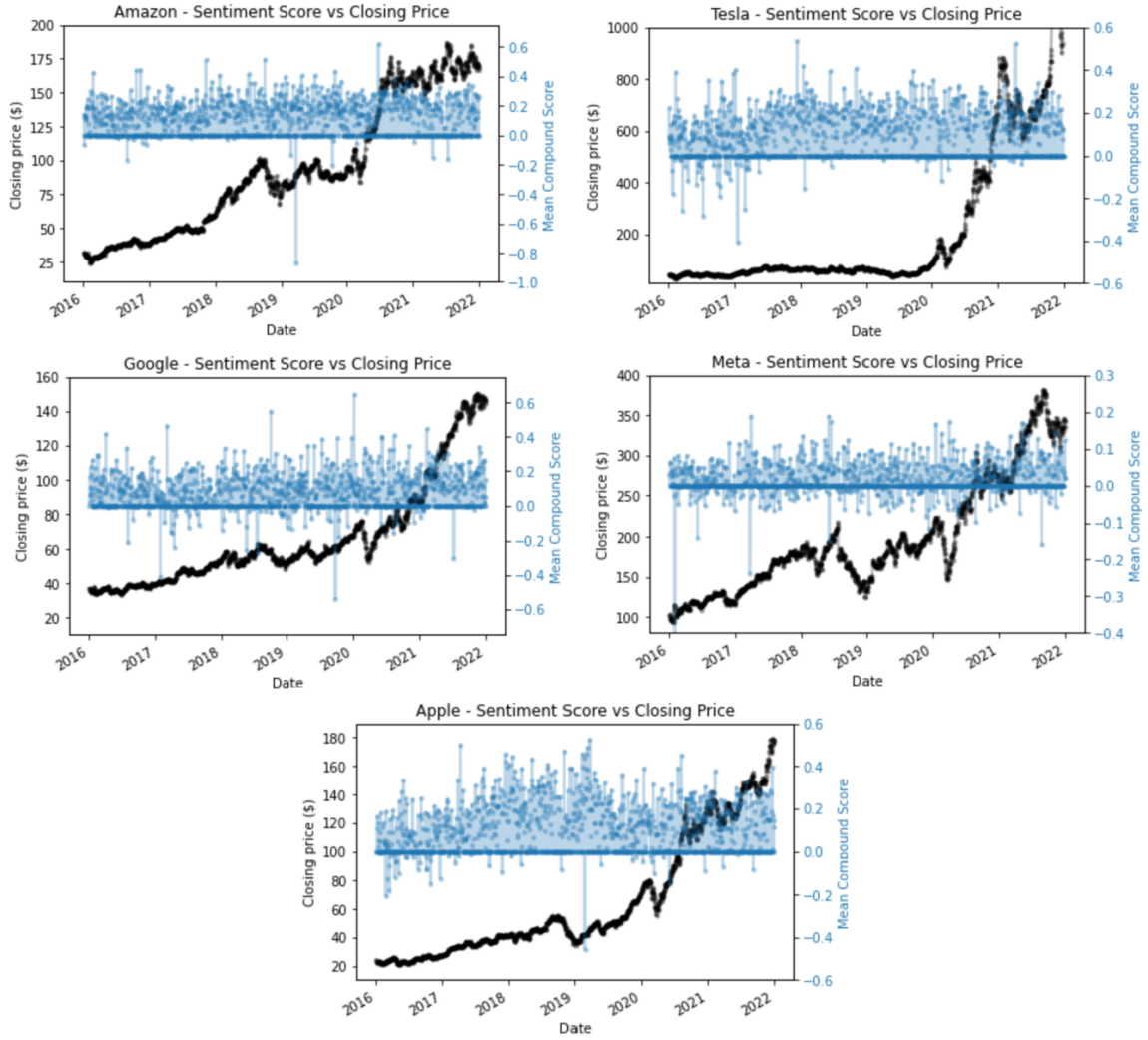


Fig. 4.1 Sentiment Score vs Closing Prices

4.1.2 Pearson Correlation Matrix

Another way to find if there is a relation between the closing price of the stocks and the sentiment is to plot the Pearson Correlation Matrix and interpret the results. A positive value would indicate a positive correlation, as the value of one variable increases, the value of the other also increases. On the other hand, a negative value indicates a negative correlation, as the value of one variable increases, the value of the other decreases. And a value close to zero indicates that there is very little or no correlation between one variable and the other.The following table 4.1 displays the correlation coefficients between the variables sentiment and mean compound score of the closing stock's price of each company being analyzed.

Table 4.1 Pearson Correlation Matrix Results

Company Name	Pearson Correlation Coefficients	
	Sentiment	Mean Compound Score
Tesla Closing Price	-0.1	-0.01
Google Closing Price	-0.004	0.06
Amazon Closing Price	-0.05	0.03
Meta Closing Price	-0.06	0.08
Apple Closing Price	-0.05	0.02
Average Value	**-0.05**	**0.03**

4.1.3 Linear Regression Coefficients

Additionally, another way to find if there is a relation between two variables, if one is the dependent variable and the other an independent variable of a linear regression model is to observe the values of the coefficients of beta of the particular independent variable which relation with the target variable we are trying to find, in this case the coefficient of the sentiment and mean compound score variable. This value tells how much the target variable is expected to increase or decrease when the particular independent variable goes up by one. The following table 4.2 displays the results of the linear regression coefficients of each independent variable relating to sentiment and the target variable (closing price) of each company being analyzed.

Table 4.2 Linear Regression Results

Company Name	Regression Betas	
	Sentiment	Mean Compound Score
Tesla	0.91	-0.01
Google	0.83	0.34
Amazon	0.91	0.08
Meta	1.04	0.02
Apple	0.92	0.13
Average Value	**0.92**	**0.11**

4.2 Machine Learning Algorithms

4.2.1 Random Forest Regressor

The following table 4.3 displays the RMSE and R-Squared results for each of the datasets obtained of the five different companies, which can be compared to the ARIMA model scores which was used as a baseline.

Table 4.3 Random Forest vs ARIMA Model

Company Name	RFR		ARIMA	
	RMSE	R-Squared	RMSE	R-Squared
Tesla	0.0614	93.83%	49.36	-75.77%
Google	0.0457	97.50%	5.07	-84.64%
Amazon	0.0471	95.76%	6.96	-4.18%
Meta	0.0370	98.24%	24.94	-175.91%
Apple	0.0442	97.07%	8.38	-101.67%

The results of the random forest model turn out to be very good across the five different datasets. A can be observed in the figure 4.2, the model is able to capture the majority of changes in the closing price.

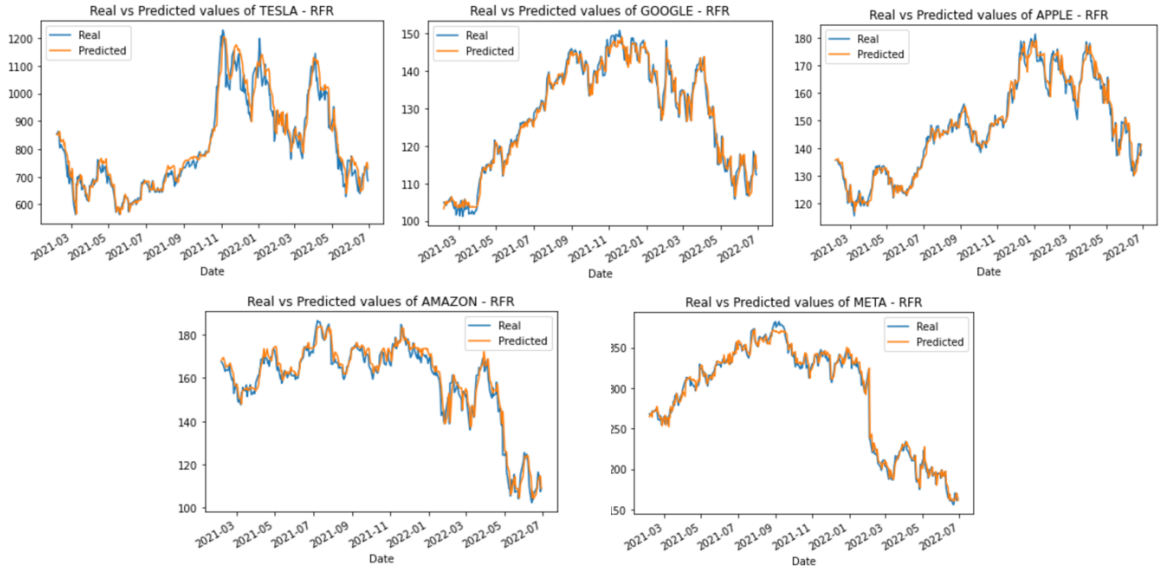


Fig. 4.2 Real vs Predicted Values - RFR

4.2.2 XG Boost Regressor

The following table 4.4 displays the RMSE and R-Squared results for each of the datasets obtained of the five different companies, which can be compared to the ARIMA model scores which was used as a baseline.

Table 4.4 XGBoost vs ARIMA Model

Company Name	XGBoost		ARIMA	
	RMSE	R-Squared	RMSE	R-Squared
Tesla	0.0633	93.45%	49.36	-75.77%
Google	0.0479	97.26%	5.07	-84.64%
Amazon	0.0470	95.77%	6.96	-4.18%
Meta	0.0384	98.10%	24.94	-175.91%
Apple	0.0439	97.10%	8.38	-101.67%

The results of the random forest model turn out to be very good across the five different datasets. A can be observed in the figure 4.3.

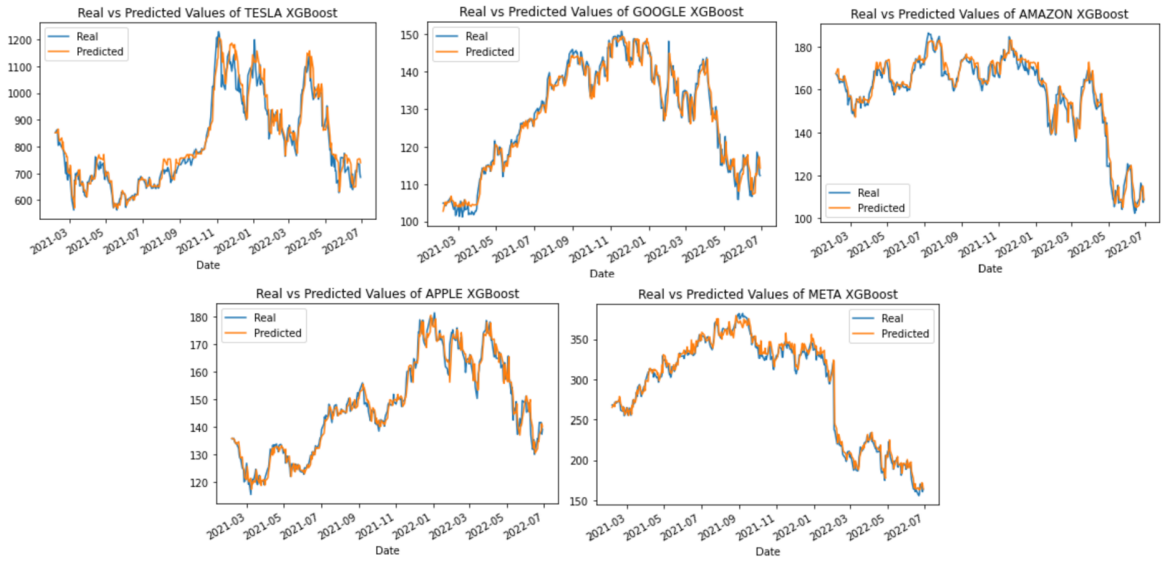


Fig. 4.3 Real vs Predicted Values - XGBoost

4.2.3 Linear Regression

The following table 4.5 displays the RMSE and R-Squared results for each of the datasets obtained of the five different companies, which can be compared to the ARIMA model scores which was used as a baseline.

Table 4.5 Linear Regression vs ARIMA Model

Company Name	Linear Regression		ARIMA	
	RMSE	R-Squared	RMSE	R-Squared
Tesla	0.0496	95.97%	49.36	-75.77%
Google	0.0480	97.25%	5.07	-84.64%
Amazon	0.0413	96.75%	6.96	-4.18%
Meta	0.0351	98.41%	24.94	-175.91%
Apple	0.0414	97.43%	8.38	-101.67%

The linear regression, though originally intended to find the regression coefficients to test the relation between sentiment scores and closing prices, turned out to perform very well in terms of evaluation metrics. As can be observed below on figure 4.4, the model consistently fits to the original data in the 5 different datasets.

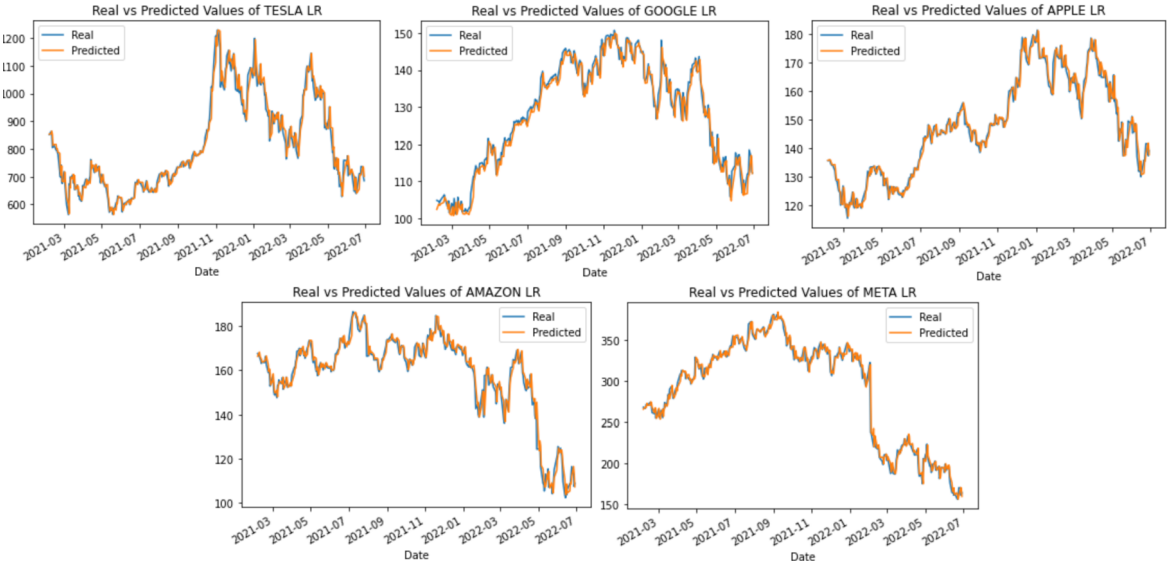


Fig. 4.4 Real vs Predicted Values - Linear Regression

4.3 Deep Learning Algorithms

4.3.1 Long Short Term Memory (LSTM)

The following table 4.6 displays the RMSE and R-Squared results for each of the datasets obtained of the five different companies, which can be compared to the ARIMA model scores which was used as a baseline.

Table 4.6 LSTM vs ARIMA Model

Company Name	LSTM		ARIMA	
	RMSE	R-Squared	RMSE	R-Squared
Tesla	0.0762	90.50%	49.36	-75.77%
Google	0.1134	84.63%	5.07	-84.64%
Amazon	0.1807	37.74%	6.96	-4.18%
Meta	0.1437	73.44%	24.94	-175.91%
Apple	0.0735	91.88%	8.38	-101.67%

The results of the LSTM model using 25 epochs and a dropout rate of 0.2 are of an r-squared of 75.64%. Which means that on average the model is able to fit to the real data in 75%. The results are consistent in almost all companies except Amazon, with the lowest r-squared of 37.74%.

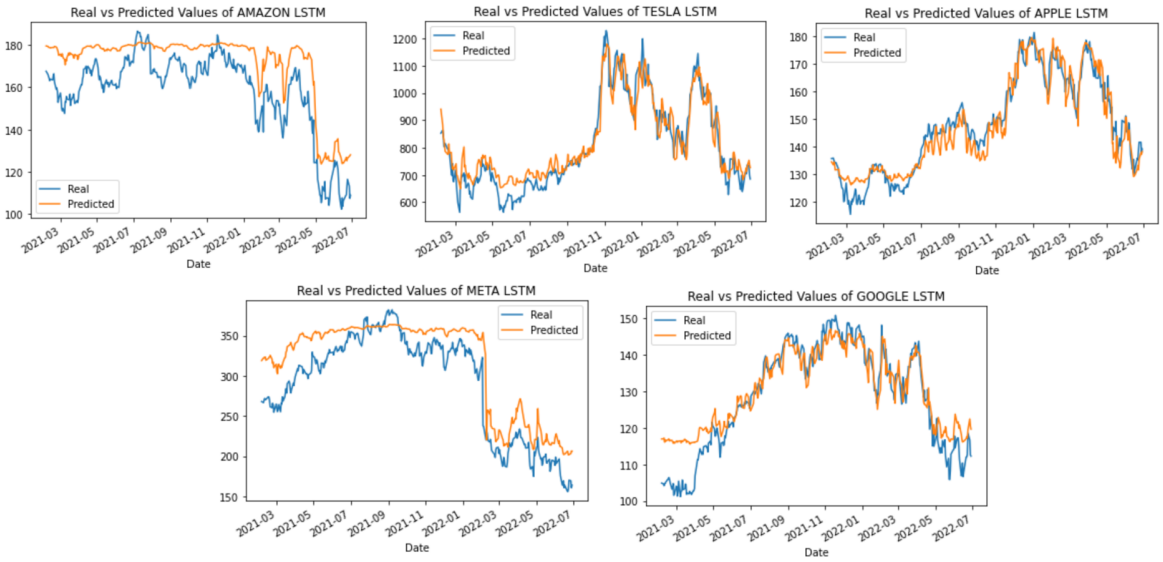


Fig. 4.5 Real vs Predicted Values - LSTM

4.3.2 Bidirectional LSTM

The following table 4.7 displays the RMSE and R-Squared results for each of the datasets obtained of the five different companies, which can be compared to the ARIMA model scores which was used as a baseline.

The results of the bidirectional LSTM have an average of 77.68% r-squared considering all five companies. The best results are for Google with 90.68% and Meta with 96.33% and

Table 4.7 BiLSTM vs ARIMA Model

Company Name	BiLSTM		ARIMA	
	RMSE	R-Squared	RMSE	R-Squared
Tesla	0.1511	62.66%	49.36	-75.77%
Google	0.0883	90.68%	5.07	-84.64%
Amazon	0.0995	81.13%	6.96	-4.18%
Meta	0.0534	96.33%	24.94	-175.91%
Apple	0.1680	57.59%	8.38	-101.67%

the worst performance is observed on Apple with 57.59%. The results are more consistent than the LSTM and they outperform the baseline model by far.

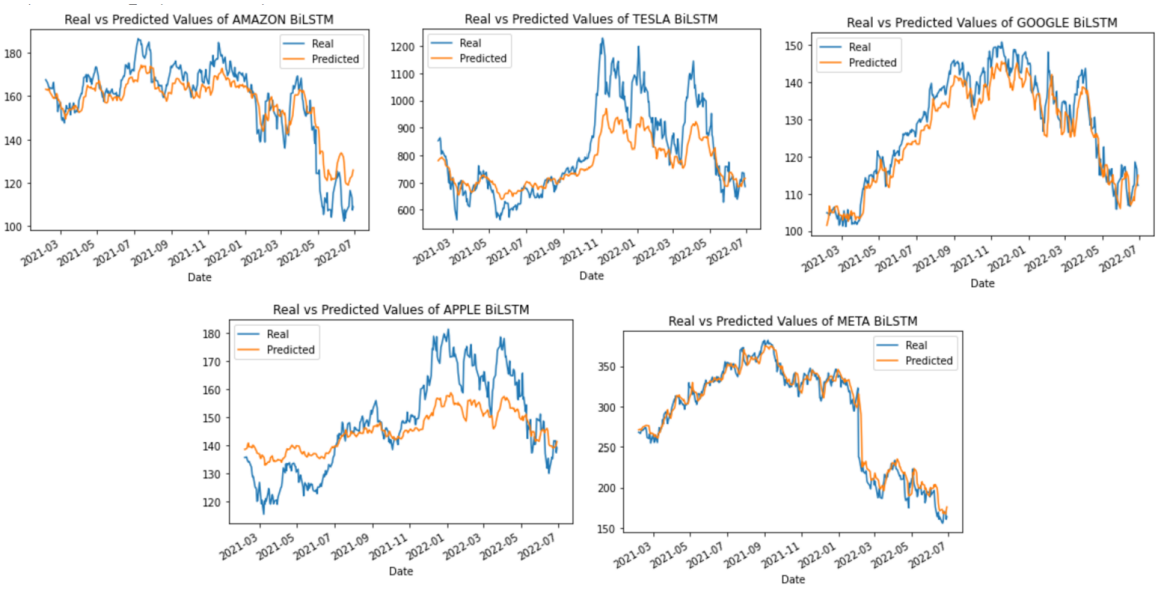


Fig. 4.6 Real vs Predicted Values - BiLSTM

Chapter 5

Conclusions and Future Work

The aim of this research is to find the relation between stock's closing price and sentiment of public opinion expressed through a social media platform like twitter. It was found that on average the mean compound scores of all twitters posted about the companies being analyzed and their closing prices have a correlation coefficient of 0.036 meaning that the relation is positive but very weak. Additionally, the regression coefficients as an average value of all companies taken into account in this study is of 0.11 between the mean compound score and the closing prices, indicating that as the score increases, the closing price will also increase by 0.11.

Another aim of this research is to compare the performance scores of machine learning and deep learning models with a traditional time series model like ARIMA. All the proposed models outperformed the ARIMA model, as is the case on previous studies.

The implemented code can be used to successfully predict the next day closing price of a stock. However, the usefulness is limited to only short-term predictions, as using it for predictions greater than a day can produce unreliable results, as the stock market is highly volatile and unpredictable. Taking this into account, traders can make use of it by updating their positions on a daily basis, if the algorithms predict a rise in the next day closing price, the convenient strategy would be to acquire a long position of the stock to profit from the rise in price. On the other hand, if the algorithm predicts a fall in the next day closing price, the wiser strategy would be to go short in the position of the stock to avoid major loses.

One of the main advantages of this research is that it included an analysis of several companies, therefore the performance of the models was tested on five different companies. So it can be expected that it will work well with different stocks and it's use is not limited to one type of dataset. Additionally, it incorporates exogenous variables, in this case the sentiment analysis results of the tweets, rather than just considering the stocks closing price. However the drawbacks of the research are that only one source of information was used to

capture the public sentiment of the company, when there are several other sources that could also have an impact on the closing price. Furthermore, the study doesn't take into account many other variables, such as, macroeconomic indicators to train the models.

Further research needs to be done to determine the exact relation between public sentiments and the stocks closing prices, by taking into account different sources where opinions about companies are expressed. Additionally, more robust models that incorporate all the variables that could have an effect on closing prices needs to be done, a difficult task considering the myriad amount of factors that can impact a stocks price.

References

Biswas, M., Shome, A., Islam, M.A., Nova, A.J. and Ahmed, S. (2021) Predicting stock market price: A logical strategy using deep learning in: *2021 IEEE 11th IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)* pp. 218–223

Chong, E., Han, C. and Park, F.C. (2017) Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies *Expert Syst. Appl.* **83**, pp. 187–205

Fataliyev, K., Chivukula, A., Prasad, M. and Liu, W. (2021) Stock market analysis with text data: A review

Jin, Z., Yang, Y. and Liu, Y. (2019) Stock closing price prediction based on sentiment analysis and lstm *Neural Computing and Applications* **32**, pp. 9713–9729

Khan, U., Aadil, F., Ghazanfar, M.A., Khan, S., Metawa, N., Muhammad, K., Mehmood, I. and Nam, Y. (2018) A robust regression-based stock exchange forecasting and determination of correlation between stock markets *Sustainability* **10**(10)

Loukas, S. (2020) Time-series forecasting: Predicting stock prices using an arima model [online] https://towardsdatascience.com/time-series-forecasting-predicting-stock-prices-using-an-arima-model-2e3b3080bd70

Nti, I.k., Adekoya, A. and Weyori, B. (2020) A systematic review of fundamental and technical analysis of stock market predictions *Artificial Intelligence Review* **53**

Pagolu, V.S., Challa, K.N.R., Panda, G. and Majhi, B. (2016) Sentiment analysis of twitter data for predicting stock market movements *CoRR* **abs/1610.09225**

Paudyal, P. (2020) Scraping tweet using twint and analyzing with nlp [online] https://medium.com/@pragya_paudyal/scraping-tweet-using-twint-and-analyzing-with-nlp-932e01ad5587

Perera, S. (2018) Rolling window regression: A simple approach for time series next value predictions [online] https://medium.com/making-sense-of-data/time-series-next-value-prediction-using-regression-over-a-rolling-window-228f0acae363

S, P. and R, V.P. (2021) Stock price prediction using machine learning and deep learning in: *2021 IEEE Mysore Sub Section International Conference (MysuruCon)* pp. 660–664

Strader, T.J., Rozycki, J.J., Root, T.H. and Huang, Y.H. (2020) Machine learning stock market prediction studies: Review and research directions *Journal of International Technology and Information Management* **28**, pp. 63–83

Wolfewicz, A. (2022) Deep learning vs. machine learning – what's the difference? [online] https://levity.ai/blog/difference-machine-learning-deep-learning