University of Warwick

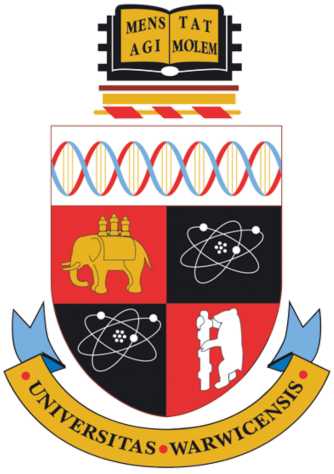
Department of Computer Science

Dissertation Report

Stock Market Prediction Using Sentiment Analysis of News Articles and Stock Brokers Opinions

Author
Jack Wells

Supervisor
Dr. Tanaya Guha



September 10, 2020

Abstract

The area of stock price prediction is a widely studied area, with many people attempting to model it over many years, however, attempting to make accurate predictions has been a consistent challenge. This project attempts to use multiple different sources of data to offer a greater insight into the causality behind the change in a particular stock price. These data sources include news articles where the use of the BERT language model, can be used to analyse the sentiment of the news articles. The project also takes into account the opinions of stock brokers, who are experts in this area.

The BERT model is able to give an 85% accuracy on the sentiment labelling part of this project. This allows for the results to be used as a feature later on, and to offer improvements to the predictions of future stock price.

However, even with these additional sources of data the models failed to learn how the stock prices would be affected, this is most likely due to the highly volatile and complex nature of the stock market.

Keywords - BERT, stock price prediction, NLP, sentiment analysis.

Acknowledgements

I would sincerely like to thank my tutor, Dr. Tanaya Guha, for her continued support during this project and for her invaluable advice and guidance.

Declarations

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for degree except for some points that are from my interim report and research proposal. Except where states otherwise by reference or acknowledgement, the work presented is entirely my own. The data used in the project is collected from publicly available sources and will only be used for research purposes.

Abbreviations

LSTM	Long term Short Term Memory
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
SVM	Support Vector Machine
EMH	Efficient Market Hypothesise
SVR	Support Vector Regression
XGBoost	Extreme Gradient Boosting Model

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Stock market prediction is one of the most attractive research topics, as the successful prediction can lead to significant profits, which has allowed for glamorisation, as in movies, such as, "The Wolf of Wall Street", and "The Big Short". Historically, predictive methods are based on the analysis of historic market data, such as stock prices, moving averages and daily returns [3]. However, recent advantages in computing have allowed for the fast processing of much more of the data available in this area, and so the context of the world around the company can now to be used to improve the modelling of stock prices.

Stock market prediction is considered one of the most challenging time series prediction problems due to the volatile nature and the noise the data contains [4]. Over the last few decades machine learning models, such as Support-Vector-Machines (SVM) and Support-Vector-Regression (SVR) have been used in the prediction of time series data, such as the financial data in this project [5] [6] [7] [8] [9]. Methods in these financial literature exploit sentiment signal features, which are inherently limited, by not considering factors such as events and news context. So does not consider or attempt to model the causality of the stock price change.

How to accurately predict stock price movement is still an open question. The Efficient-Market Hypothesis (EMH) [10] suggests that the stock price of a company reflects all available information and that any price changes are based on newly revealed relevant information, and so suggests the collection of additional data sources will lead to greater accuracy in the predictive models.

In this project it is hoped that the problem of not utilising all the data available will be addressed by leveraging deep neural models to extract rich semantic features from news articles and from the opinion of stock brokers. It is hoped that this method is competitive with other state-of-the-art approaches demonstrating the effectiveness of Natural Language Processing (NLP) for computational finance. The use of NLP will hopefully be able to capture trends in data that are easily predictable given the information, e.g. a change in the law or the private licensing of a product expiring, and how it may substantially effect a company,

and therefore the stock price. The motivation behind the use of sentiment analysis in particular is as a result of the many papers that have shown significant improvements when predicting the direction of price movements using a similar methodology. This include the use of sentiment in news articles and other sources of data such as tweets [11] [12] [13]. The main goal of the addition of this data source is to further understand the causality behind the stock price movements, and as such will result in improved accuracy of the models.

1.2 Motivation

The area of stock market prediction is a well explored area going back many decades [14], attracting a considerable amount of research, with the methods changing over the years from using SVM to more modern methods such as neural networks. The use of NLP in this area is a more recent development, with most research focusing on a binary classification, providing a simplification of the problem of predicting stock price (classifying either an increase or decrease in stock price). Some of the most widely studied approaches rely solely on analysing the recent prices and volumes of the traded stock [15] [16] [17] [18] [19] and it is hoped that the model made in this project is competitive with these approaches.

A model with the name Enalyst was introduced in Lavrenko et al. and uses similar methods as this project; that being the use of the headline of news articles. Vivek Sehgal et al. [20] and Michal Skuza et al. [21] used the sentiment analysis of texts to develop their models, with Michal Skuza paper using the analysis of tweets. All of these methods have limitations, including unveiling the rules that may govern the dynamics of the market which makes the model incapable of capturing the impact of recent trends in the overall stock market. Using the sentiment analysis of just textual data to predict an increase or decrease does not offer an accurate prediction on the exact actual price. Additional sources of data are required to make the predictions better, as this project demonstrates through the use of brokers opinions, news articles, and stock market price data.

Recent developments in the use of neural networks to learn dense representations of text, have been shown to be effective on a wide range of NLP problems; given enough training data. This provides strong grounds for the exploration of deep-learning based models in the prediction of stock prices. For example, Ding et al. [22] has shown that the use of deep-learning representations of event structures, yields better accuracy compared to discrete event features. This is a crucial reason why this project uses NLP, specifically in the use of sentiment analysis of news articles.

Objectives of the Project

The simplest way to look at stock market prediction is the act of trying to determine the future value of a company [14]. The problem in this project can be seen as a way of attempting to calculate the value of a company at a specific time, that may or may not be reflected in the stock price of a company. [4] For

example, it is possible for a company to be worth more than the current stock price suggests if the future of that company is bright. The opposite is also true. In the case of a failing company, they may have assets that give them a current value but if the company fails they may become worthless. However, with this information the future stock price should reflect the value of the company.

The problem this project intends to provide some insight into is the development of predictive models to accurately predict the future stock price of a company based on the information that is available. It is believed that the more contextual (textual data, brokers opinions) data that the model is given, the better the predictions should be. It is fully possible that the increase in data does not offer any advantage as it may add noise, perhaps to an extent that no improvements can be made, or that the extra data hinders the modelling.

The exact task to be solved will be, given the data of the companies; broker opinions, news article and the financial data. They will be used to build data points with a rolling window approach. These data points will be used to predict the direction of the price change some days in the future. In this paper 10 days will be used to create each of the data points. From the last day used in the data point the closing value on that day will be compared to the closing price 10 days in the future to determined whether the price has increased or decreased. The exact details are outlined in chapter 3 and 4.

This project builds models to predict the future stock price movements. This will be done using combinations of the three data sets that have been collected for this project, these being the news articles, stock brokers opinions, and the financial data. The objectives are outlined below.

- Investigate the use of stock market data in the prediction of stock price changes.
- Develop an accurate model trained on financial news articles to understand the sentiment of financial news articles. This can then be used to predict the sentiment of the news articles collected for this project.
- Use the sentiment analysed news articles as an additional feature in the prediction of the future stock price.
- Build models that incorporate the use of brokers opinions and stock price data, both separately and together to attempt to improve the modelling.
- Combine all the data sources into one model that is capable of understanding the sentiment, brokers opinions and financial values of a company over a small window and make predictions about the future price of a stock.

Differences Between the Project and Other Work

One of the major differences of this project has compared to others, is how the sentiment is derived from the news article headlines. It is intended that multiple news articles will be used together as this will hopefully give a better context

of the company. This project will also use the BERT model [23] (Bidirectional Encoder Representations from Transformers) created by Google in late 2018, which has been pre-trained on English Wikipedia and BookCorpus. This is a state-of-the-art model that can relate the content of many sentences together. In this project, it was considered to be used to relate the meaning of many news articles together, for instance the ten most recent or all articles that have been published in the last month. When doing this, the resulting points had multiple unrelated articles, this removes one of the key advantages of the BERT model. That being the ability to relate sentences that have a common meaning or relation, as the relation between all of these news articles can be so unrelated the model fails to learn the specific sentiment and predicts neutral at all times. So, the combination of the news articles will be done in a different way, described in chapter 3.3.

Another key difference is the use of brokers opinions to help predict the future price. Which my understanding is, has not been explored in other papers and research. This, and the combining of multiple different sources of data, e.g. the brokers opinions, news articles and financial data, is something that is not often undertaken. Most papers focus on one of these or maybe two but not three or more. The use of more of these data sources is hoped to improve the accuracy by offering greater context.

Hypothesis to be Tested in the Project

This project follows the assumption that the use of additional sources of data for features will be able to improve the models. Especially when considering things such as the use of news articles that will hopefully be able to offer an insight into the reasons behind a price change, based on the sentiment of the article.

1.3 Possible Future Challenges that Could be Faced

As in many research projects it's likely that there will be a variety of challenges that will come up in areas such as project methodology, resource availability and project managements. These attempt to be mitigated by trying to foresee as many problems as possible to reduce or overcome the effects. Some of the most likely problems are described below.

- Stock prices are very volatile and subject to a large number of factors that are unforeseeable, this makes the problem very difficult and gives rise to the possibility that the project may not result in success.
- Sentiment is subjective to the view of an individuals, so the training of sentiment analysers can be difficult due to the ambiguity in the textual data, this will be difficult to overcome but as described later, on deep-learning models will be used to mitigate this problem successfully.

- The time dependencies in stock prices can be highly complex, and may be dependent on more factors than are available to the models, than in the data that is being provided. So the models may not learn the reasons why a stock price has changed.

1.4 Structure and Overview of this Project

This paper begins with an introduction to the project and the motivations and importance of studying this area. It then moves on to give an overview of the previous work in this area, and some of the important pre-built models and practices that will be used within this project. It then goes onto explain how the database used in this project for building models to predict the future stock price was created. It then goes on to outline the methodology that will be used in the project. The modelling will be done through the successive addition of different data sources, these being the stock price data, brokers data, and news articles. This will hopefully allow for the analysis of each of these data sources to see if any specific one is a particularly good feature. The paper concludes by giving an overview of the challenges and problems that have occurred throughout the project and offers some possible improvements that could be undertaken in the future.

Chapter 2

Background for the Project

2.1 Description of Some of the Key Models Used in the Project

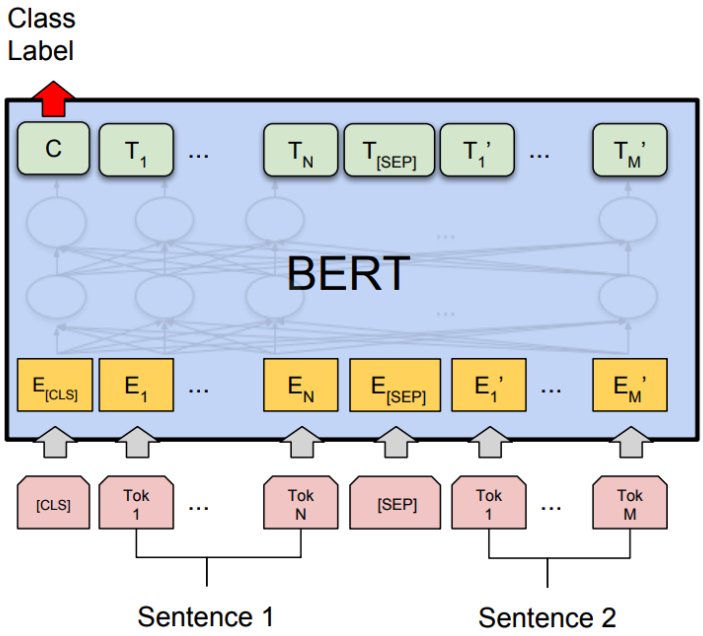
Background of LSTM's Neural Networks

An LSTM (Long short-term memory) is a type of recurrent neural network that consists of LSTM units. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). These are particularly capable of understanding the time relation between events in the data. This could be used to gain the context of a word in a given sentence or a delay in time series data. [24] This is one of the key reasons why it is expected that the use of LSTM's for both the analysis of news articles and the prediction of future stock prices will perform better than most other models.

Background and Examples of the BERT Model

LSTM's have the ability to relate data points from before a specified time point allowing for the theorised improved results. The use of bidirectional nodes allows for the ability for the model to look forward and backwards in the data. This makes LSTM's very good at understanding text as they can look at words before and after the current target word, to understand the word's context and so the meaning of the target word. This is what the BERT model implements.

Figure 2.1: Visual representation of the BERT model.



The BERT (Bidirectional Encoder Representations from Transformers) model [25] is being used because of its state-of-the-art performance on tasks such as sentiment analysis and other NLP tasks like information retrieval, and summarising. [26] [1] [27]

The results from "Exploiting BERT for End-to-End Aspect-based Sentiment Analysis" [1] are shown below. In this paper, the authors focus on the aspect termlevel End-to-End Aspect-Based Sentiment Analysis (E2E-ABSA) problem setting [28]. This is one of the SemEval tasks that serves as a bench-marking project. As can be seen the BERT model out performs the previous state-of-the-art models presented in Li et al. [2], in the summary table below.

Table 2.1: Main results from Li and Lidong et al. [1]. The symbol * denotes the numbers that are officially reported in Li et al. The results are retrieved from Li et al. (2019a). [2]

Model		LAPTOP			REST		
		P	R	F1	P	R	F1
Existing Models	(Li et al., 2019a)*	61.27	54.89	57.90	68.64	71.01	69.80
	(Luo et al., 2019)*	–	–	60.35	–	–	72.78
	(He et al., 2019)*	–	–	58.37	–	–	–
LSTM-CRF	(Lample et al., 2016)*	58.61	50.47	54.24	66.10	66.30	66.20
	(Ma and Hovy, 2016)*	58.66	51.26	54.71	61.56	67.26	64.29
	(Liu et al., 2018)*	53.31	59.40	56.19	68.46	64.43	66.38
BERT Models	BERT+Linear	62.16	58.90	60.43	71.42	75.25	73.22
	BERT+GRU	61.88	60.47	**61.12**	70.61	76.20	73.24
	BERT+SAN	62.42	58.71	61.12	72.92	76.72	**74.72**
	BERT+TFM	63.23	58.71	60.49	72.39	76.64	74.41
	BERT+CRF	62.22	59.49	60.78	71.88	76.48	74.06

As shown in the results of these projects the BERT model offers a significant improvement on the performance of these tasks, and in the others that have been referenced. This gives a good indication that the model will have a similar effect on the performance of the predictive models within this project, as they are both variants of sentiment analysis projects.

One of the main advantages the BERT model, as seen in the figure 2.1 is its ability to understand the sentiment of multiple news articles at the same time, through the use of several sentences. The BERT model has specialised transformer layers that convert the tokenized words and interpret them to perform so well on these tasks. The BERT model does require the input of a mask, which can be shown in the image as $E_{[CLS]}, E_1, \ldots$ where E_i represents actual words and the other ($E_{[CLS]}$) special characters that indicate the start, end, and padding spaces within sentences. The model then transforms the words and uses these to produce a sentiment label for the sentence based on the training data, which is expected to teach the BERT model what information each category should expect to see.

2.2 Past Work

A key piece of work for this project is, "Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network" [3] by Huicheng. This project will develop on and use key insights, such as the use of news article headline instead of using more of the textual data, from, for instance the abstract of the article or the entire article.

In the paper the author aims to leverage publicly released financial news and train an LSTM model to make predictions on the directional change of the both

the Standard and Poor's 500 index as well as companies within them. The model used by the author consists of Recurrent Neural Network (RNN) to encode the news text and capture the contextual information; self attention mechanism is applied to distribute attention on most relative words, news and days. The model outlined above then demonstrates its competitiveness with other state-of-the-art approaches, showing the effectiveness of recent advances in NLP technology for computational finance.

The paper goes on to explain some of the key background in the methodology, such as "Bag-of-Words" and "LSTM" models. These methods and models will be used throughout that project. However, there is no need to review work and models at depth that have previously been addressed in other papers and is not specifically relevant to this project.

The paper uses a pre-trained 100 dimensional word embeddings with a skip-gram method, trained and developed on $445,262$ articles for training and $55,658$ articles for development, and a testing set will consist of $55,658$ articles. The training vocabulary has a size of $153,214$ words. The word embeddings are fine-tuned during the model training. This project attempts something similar when using the 100 dimensional GloVe pre-trained word embedding to build an LSTM sentiment analyser in chapter 5.1.

The results of this paper are shown below for several different models. Those being an SVM a form of classical machine learning, the others are forms of deep neural networks. The Bag-At-LSTM and At-LSTM show that sentence encoding with LSTM models offers a slight accuracy improvement over the Bag-of-Words model. The WEB-LSTM and At-LSTM indicates that character level composition helps improve the models accuracy. The comparison between CNN-LSTM and At-LSTM shows that the news level self-attention layer can help capture more relevant news titles and their temporal features. The author reaches the same conclusion as Ding et al. [29], that the use of news headlines offers a higher accuracy than using the whole article or just its abstract, both of which have some negative effect on the modelling. As shown in the Ab-At-LSTM (abstract) and Doc-At-LSTM (document). The model proposed by the author offers a lower accuracy than the accuracy of KGEB-CNN proposed in [29]. As discussed by Huicheng [3] this is likely because Knowledge Graph Event Embedding (KGEB) is a more powerful method than the sequence embedding preformed in Huicheng's paper. These results are hoped to give this project an estimate of the current performance of many of the current methods that are being used.

Table 2.2: Results showing the accuracy of using different types of neural networks with different forms of data and data pre-processing. From "Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network" paper.

Model	Average Accuracy (%)	Max Accuracy (%)
SVM	56.38	–
Bag-At-LSTM	61.93	63.06
WEB-At-LSTM	62.51	64.42
Ab-At-LSTM	60.6	61.93
Doc-At-LSTM	59.96	60.6
Tech-At-LSTM	59.96	64.42
CNN-LSTM	62.51	63.06
E-NN	58.83	–
EB-CNN	64.21	–
KGEB-CNN	**66.93**	–
At-LSTM	63.06	**65.53**

As can be seen the SVM model performs significantly worse than the neural network approaches with the accuracy of the models made by Huicheng average about 62% accuracy that is over 5% better than the SVM model. This again goes to show that the hypothesis surrounding the use of LSTM's is likely to be correct.

There are however more promising results shown when the analysis is done on an individual company (Table 2.3). This is likely to be due to the predictions of those news articles related to specific companies being more relevant. In contrast to using the entire corpus to predict the s&P 500 which adds noise and so reduces accuracy. The accuracy of predicting WALMART is over 70% which is better than the best model when predicting the entire S&P 500.

Table 2.3: Results of the accuracy of using the At-LSTM model on predicting individual companies in the S&P 500 index. From "Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network" paper.

Company	Average Accuracy (%)	Max Accuracy (%)
GOOG	68.75	71.25
AMZN	67.32	69.46
CSCO	66.82	67.62
MSFT	67.92	69.89
APPL	67.52	69.42
INTC	67.12	67.63
IBM	69.49	71.41
AMD	66.12	69.10
NVDA	69.35	70.51
QCOM	68.53	69.70
WMT	**70.36**	**72.06**
T	68.53	69.70

Another key piece of work is "Using Structured Events to Predict Stock Price Movement: An Empirical Investigation" [29]. This paper explains how the use of Open Information Extraction (Open IE) has enabled the extraction of structured events from web-scaled data. This method is shown to offer significant improvements in the accuracy over the use of a Bag-of-Words approach in the sentiment analysis. The paper mainly contains information on how to effectively use these NLP methods for the prediction of stock prices. The model uses an additional evaluation metric based on the Matthews Correlation Coefficient (MCC) to avoid bias due to data skew, in this project the F1 score will be used. This is because some companies may have a large number of increases in their stock price, this data would have a high accuracy even if the predictive model classified all points as positive. The paper is particularly useful as it develops a way of processing financial textual data and makes available the large amount of data that they used.

2.3 Observations

Previous pieces of related work offer many key insights into the problem of stock price prediction, and offer many approaches that can be implemented in this project.

The paper mentioned above gives two major insights that this project will follow, these being;

- the use of news article headlines instead of the whole text or just the abstract,

- as the performance of the models trained on the individual companies seemed to be better than when only looking at the S&P 500 as a whole, individual companies will be used in this project. They will also be combined to create better predictive models; the FTSE 100 index itself will not be used.

Chapter 3

Database Creation

3.1 Data Collection

To acquire a large sample of data to enable this project to come to the best and most accurate conclusion possible, the data will come from all the companies in the entire FTSE 100. These are the 100 largest companies (that are listed on the London stock exchange), which will hopefully have the most news articles and brokers opinions written about them. Although, one company has data that has many occasions that the price does not change (3i), and another has only been listed for 70 days which is an insufficient number of days to model from. As such, these two companies will not be used in this project. So, that leaves 98 companies that will be used to model on and create data points for use in the combined and individual models.

The models are trained on the individual companies, these are company specific models. The models use 1,000 data points to train the model on, this equates to around 4 years of data. The remaining data points are used for testing, there are 583 data points that are tested on which represents 2 years approximately. All the data is then combined; this is done in a similar way to the individual example, that is by taking the first 1,000 data samples and testing on all data points after which is 583 data points for each of the companies.

The data has been collected between 1^{st} January 2014 and 27^{th} March 2020. Choosing a larger span of time risks having to remove more companies that haven't been trading before January 2014. There are four sources of data for this project including Yahoo finance, that offers a API that allows the download of the financial information of each company over a period of time. This data includes the closing price, high price, low price, volume traded, adjusted close and the opening price of each company for each day the stock markets are open, and an example can be seen below:

Table 3.1: An example of the stock price data that has been collected through the Yahoo API. This is specifically for the ASHTEAD GROUP PLC stock.

Date	**Open**	**High**	**Low**	**Close**	**Adj Close**	**Volume**
2014-01-02	765.5	785.5	760.5	783.5	707.394714	2448870
2014-01-03	780.5	782.5	769.5	776.5	701.074646	1086984
2014-01-06	776.0	792.5	772.5	790.0	713.263306	1704991
2014-01-07	790.0	807.5	790.0	804.0	725.903442	1812159
2014-01-08	805.0	811.5	789.0	790.0	713.263306	1172443

The second source of data was a website [30] that contained brokers opinions on the stocks. To extract the data from this website a web scraper was developed that would go through and load all the web-pages that contained relevant data about a company and then move onto the next page or pages, then the next company. These opinions are given in statements such as "buy" or "sell". There are also more ambiguous categories such as "house stock". So that the model can learn better, some of the noise is removed by forming five categories out of the the statements which relate to the relative "goodness" of the statement; they are "buy", "outperform", "hold", "underperform", and "sell". A breakdown of the different categories and the number of samples within can be seen below:

Table 3.2: This shows the proportions of brokers opinions within each of the five categories. Combined from all companies.

	Percentage of all (%)	Number of samples
Buy	31.15	7,013
outperform	19.23	4,329
hold	35.34	7,956
Underperform	7.83	1,763
Sell	6.45	1,451
Total	**100**	**22,512**

As can be seen the data is heavily skewed towards the positive categories, this may mean that it becomes harder to learn from as the data will often be very similar.

The third source of data in my project was another website [31], used to gather news articles about the companies over the last 6 years. This was done with the use of web scrapers similarly to the way the brokers opinion data was collected. The data collected was the news headline, the date of the article, and the company that the article related to, e.g. Tesco. The data is more prolific in recent times, probably due to the increased use in technology in the last few years. See figure 3.1.

The final data set used in this project is an openly available data set that contains sentiment labelled financial news articles headlines. These will be used to train the sentiment analysing models. The data set is labelled by eight humans

where each human gives a category that they think the sentiment of the article falls into, the possibilities being "positive", "neutral", and "negative". Multiple humans do this for each article so there can be some differences in the opinions. To over come any issues associated with that, this project uses the collection of news articles that 66% of the people agree on for training and fine-tuning models for sentiment analysis purposes. The trained sentiment models will be used to predict the sentiment of the news articles collected for this project. The two news article data sets are very similar so the models trained on the labelled data set can be used on the news articles collected for this project.

3.2 Data Processing

Stock Data Processing

The stock price data will be scale, this means the data from multiple companies can be combined even if they have large differences in the average stock price. A statistical transform is used that models the data with a Gaussian distribution, with mean zero. Combine all the data together and it is hoped that the models will improve as there will be more data available to learn from. This should benefit the LSTM model, as it is particularly data hungry.

Brokers Data Processing

The brokers data needed to be pre-processed, this included turning the brokers opinions into the five categories, mentioned above. This data was one-hot-encoded as below:

Table 3.3: The brokers opinion data after being reduced to the 5 categories.

Date	Buy	Outperform	Hold	Underperform	Sell
2014-01-13	0	0	1	0	0
2014-01-15	0	0	0	1	0
2014-01-31	0	0	0	1	0
2014-02-03	1	0	0	0	0
2014-02-05	0	0	1	0	0
2014-02-11	1	0	0	0	0
2014-02-19	1	0	0	0	0
2014-02-20	1	0	0	0	0

The brokers opinions are a long term prediction, and therefore it would make sense to use the same opinion for multiple days. The opinion is presented and accounted for in each of the subsequent 100 days. The 100 days represent 100 days the stock market was open, given from the stock price data set. 100 days of trading was chosen as being a reasonable length of time for the opinion would

remain relevant for (in real time this is approximately 120 days). This is shown below:

Table 3.4: Brokers data after being propagated through the data for 100 days, this is the data for WPP (Wire & Plastic Product PLC).

	Buy	**Outperform**	**Hold**	**Underperform**	**Sell**
1573	2.0	1.0	1.0	0.0	0.0
1574	2.0	1.0	1.0	0.0	0.0
1575	2.0	1.0	1.0	0.0	0.0
1576	2.0	1.0	1.0	0.0	0.0
1577	2.0	1.0	1.0	0.0	0.0

This is then combined with the stock price data that a has been scaled, as described in chapter 3.1, they are combined on the date that the opinions were released.

Text Processing

The text has undergone a classic NLP pre-processing method that is out lined as follows.

- **Lower Case** : All words and letters within them are changed to lower case letters.
- **Stop Word Removal** : these are words that don't carry much information such as "The", being removed.
- **Stemming** : this aims to reduce the vocabulary and increase the number of overlapping words within the training and the testing data set collected. Stemming reduces words such as "consult", "consulting", "consultant" all become the word "consult".
- **Word Substitution** : words like "ftse" and words highly related to it such as "ftse 100" have been changed into the word "index", they have a similar meaning but will have a better word embedding, as "FTSE" isn't a word but index is.
- **Increasing Word Overlapping** : Any mention or abbreviation of a company name will be changed into the single word "company". This is because it's not expected that a specific company name will offer any additional information that can be used. However, when using the word "company", a better predictive model will hopefully be produced.

In later parts of this project other pre-processing methods where required to be used to make the text data usable by the pre-trained BERT model. These include adding special start, stop, and end of sentence characters to the text.

Sentences were then also tokenized, enabling a computer to understand the word that is being given. In all of the models, word embeddings where used, as they have the ability to carry contextual information that a word may have. For instance words with similar meaning have similar embeddings, e.g. "good" and "great" have similar word embedding. The sentences are then padded with zeros to make them all the same length. An example for a sentence can be seen below:

Original: according finnair technical service measure due employment situation

Tokenized: ['according', 'finn', '##air', 'technical', 'service', 'measure', 'due', 'employment', 'situation']

Token IDs: [2429, 9303, 11215, 4087, 2326, 5468, 2349, 6107, 3663]

3.3 NLP Modelling of News Articles

To make use of the collected text it has analysed. This is done through sentiment analysis to give a perspective of what the news article that means and then how it will possibly effect a companies stock price.

The motivation for using news articles can be easily seen within recent times, for instance due to coronavirus. Many companies have had large drops in their stock price, resulting in the last month or so of data being anomalous, as the price dropped "unexpectedly". This big price drop was unexpected in the stock market and brokers data, whereas in reality it was foreseeable. Using news articles to build a better intuition of the price seems to be a good idea when creating better models. The articles should be able to offer an improved context to the problem, and so increase accuracy.

There are two ways to do this:

- Build model that takes text and stock price data together.
- Build a separate model to interpret the text and then feed this into a separate model to predict the future price changes.

In this project the second approach is used. Combining multiple different types of data such as, numbers and text is very difficult, and it will be hard to interpret the text in that kind of model. Whereas there exists many good methods to model text by itself which have been widely studied, understood, and are not overly difficult to implement. This project chooses to use the sentiment of the news articles as a feature in the prediction of the future stock prices.

As will be described in chapter 5.1 the models are trained on a set of labelled financial news articles. These pre-trained models are later used to predict the sentiment of news articles, these are the ones that have been collected for this project, and which that relate to the data sets that have been collected and relate to the individual companies. Using the fine-tuned BERT model, that will

be described later, can predict the sentiment of the news articles collected in this project. Looking at the labels produce by the best preforming model, the BERT model, we can see the sentiments have been well calculated. For example a random sample of the data is shown below.

Table 3.5: Example of news articles after pre-processing and having the sentiment analysed by the fine-tuned BERT model.

Price change (%)	Sentence	Predicted Sentiment
-1.1	forget cash isa id rather buy company share price yield	neutral
-8.503	coronavirus pandemic gut hotel stay demand data	negative
-0.878	britain company raise fiscal year profit outlook	positive
-2.58	big oil billion fund back new cement engine technology	positive

Adding the Sentiment Feature to the Data

The results of the BERT model that will be presented in chapter 5.1 are very encouraging. This will allow for the possibility to use this data source effectively to improve the accuracy of the current models, as has been done in other papers which has this project follows.

The data from this was then combined in a similar way as the brokers opinions where, in this case the articles sentiment was added up over a 10 day period (2 weeks in real time). A negative sentiment article was given the value -1, a positive 1, and a neutral 0. Articles released on the same day had the sentiment added together prior to being reported for the 10 day period. This allowed for a development of the sentiment over a reasonable time period and hopefully capture the length of time that an article remains relevant for. An example of the data is shown below after being added to the current data set is shown below.

Table 3.6: An example from one of the companies complete data set after all processing and analysis. Subsets of this data set are used to model on in all of the results chapter.

Index	Date	Open	High	Low	Close	Adj Close	Volume	Buy1	Outperform1	Hold1	Underperform1	Sell1	BERT combined
1596	2020-03-13	579.60	602.80	537.80	557.60	557.60	8640055.0	1.0	1.0	1.0	0.0	0.0	-1.0
1597	2020-03-16	530.40	538.40	467.00	501.60	501.60	15163156.0	1.0	1.0	1.0	0.0	0.0	-1.0
1598	2020-03-17	521.20	531.80	458.60	489.90	489.90	10467616.0	1.0	1.0	1.0	0.0	0.0	0.0
1599	2020-03-18	468.40	496.20	450.00	491.70	491.70	13306849.0	1.0	1.0	0.0	0.0	0.0	-4.0
1600	2020-03-19	490.70	513.00	468.20	497.30	497.30	8173602.0	1.0	1.0	0.0	0.0	0.0	-4.0
1601	2020-03-20	519.40	529.40	480.70	492.20	492.20	18207035.0	1.0	1.0	0.0	0.0	0.0	-4.0
1602	2020-03-23	462.00	498.30	455.94	490.00	490.00	11429443.0	1.0	1.0	0.0	0.0	0.0	-5.0

This is done for each of the 100 companies, the complete database can be found in the appendix A.

3.4 Data Summary

Stock price data

The stock price data is the most important data set, as it is used to calculate whether a stock price has changed, and to give the time points for other data sets. That is, only the days that are in the stock price data are used to create data points. For events in other data sets that happen outside of these days, the points are moved to the next day that the stock price data has. For example on the 1^{st} an opinion is released and the 1^{st} is not in the stock price data set the opinion will be placed on the next day in the stock data set which could be the 2^{nd} or 3^{rd}. Every company that is used has 1,583 data points giving a total of 155,134 data points for the 98 companies that have been used to model.

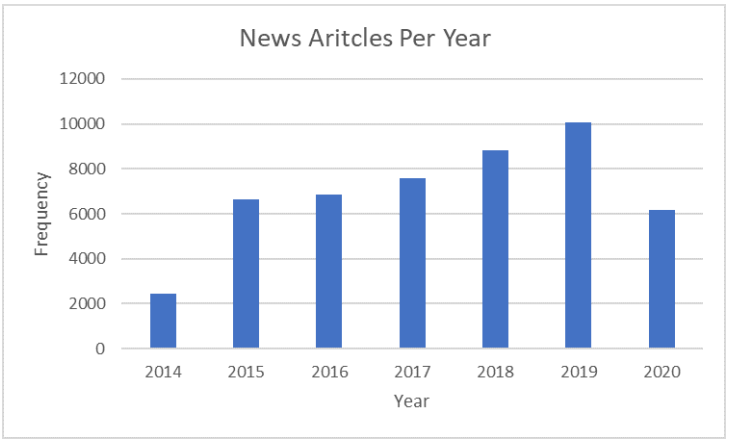
Brokers

Within the brokers data set there is 22,512 examples of brokers opinions with the companies ranging between having 4 and 500 opinions. The mean number of opinions for each company is 225.12, with a standard deviation of 119 this represents about 14% of the days in the stocks data set having a brokers opinions. The median number of brokers for the number of brokers opinions is 203. A breakdown of the data set can be seen in Table 3.2.

News articles

48,631 news article titles were collected, with a total 23,439 being unique. This is because a news article can relate to multiple companies and so can appear multiple times in the data set. The companies have between 21 and 3,989 individual articles written about each company with the mean being 491 with a standard deviation of 622 with the median being 284. This represents a news article being written about a company every three days. However, the distribution of the news articles means that the articles are concentrated towards the more recent dates.

Figure 3.1: The number of news articles per year.



Chapter 4

Proposed Methodology

4.1 Types of Models Used in the Project to Make Stock Price Predictions

Four different types of models are used to model the data on, these are; a logistic regression, SVM, XGBoost, and an LSTM models. The hope is that the more complex LSTM model out performs the other models. These models have been chosen as the SVM and logistic regression are often used in the prediction of stock prices in similar research papers. The use of the XGBoost model is used as this is a type of decision tree method has performed very well on many Kaggel tasks and occasionally in similar projects to this project [32]. These models represent a good selection of the current machine learning techniques, as they include a deep learning neural network (LSTM), a SVM model, a regression method, and a decision tree method. However, due to the size of the data set when combined the SVM takes too long to train, and as such will not be used to model the combined data set in chapter 5.3.

4.2 Rolling Windows for Time-Series Analysis

Rolling windows are a common approach when choosing to model time-series data as they feed in the data for multiple days into a model, instead of using the features for just one day. The use of multiple days allows for the models to see how the data changes day by day to be able to predict future trends [33]. They work by looking at a specified number of days data before a specific day and the data and using the combined data to create data points to be modelled on.

The data points are formed through a rolling window approach where the 10 days of data are used to create each of the windows. The data is split into two parts; the part that get scaled and the part that does not get scaled. The scaled data is the financial pricing data (daily high, low, open, close, adjusted close, and volume) this data is scaled over the 10 day window with a "Standard Scaler" this models the data with a mean 0 and standard deviation 1 distribution, calculated from the data points included within each individual window. The

data is scaled as companies can have a large difference in exact price of the for instance AstraZeneca trades around 8,000 pence per share and Tesco only trades around 220 pence per share. So these different companies can be combined and used together to build models they need to be scaled. This is not a problem with some other sources of data such as the brokers opinions or the sentiment analysis part, as such they are not scaled. This does offer an easy way to separate the modelling through including and not including different data sets. As will be explained with this code snippet:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

def window(stock, brokers, size, delay):

    windows = []
    binary_pred = []
    pred = []

    for i in range(len(stock) - delay - size):

        scaler.fit(stock[i: i + size])

        win = []
        for j in range(size):
            # comment top line to only include brokers opinions/text
            win.append(np.concatenate(scaler.transform(stock[i+j:i+j+1])))
            win.append(brokers[i+j])

        windows.append(np.concatenate(win))

        # Exact future price scaled according to historic data
        pred.append(scaler.transform(stock[i+size:i + size +
            delay])[-1][-3] - scaler.transform(stock[i:i +
            size])[-1][-3]) # The actual value scaled appropriately

        #Binary pred category
        if scaler.transform(stock[i+size:i + size + delay])[-1][-3] <
            scaler.transform(stock[i:i + size])[-1][-3]:

            binary_pred.append(1)
        else:
            binary_pred.append(0)

    return windows, binary_pred, pred
```

NOTE - The inputs can include the text within the brokers processing and information can be removed when the data set, containing all data, is split into the two categories; the one for scaling and the one which will not be scaled.

This code is used to create the windows, as well as the appropriate label, e.g. increase or decrease for the window in question. It also produces the exact future value as an output label, although that part of the project is now no longer needed.

- INPUT :
  - **Data Sets** : two data sets containing the data that will be scaled and the data that will not be scaled separately.
  - **Delay**: the number of days in the future that will be predicted.
  - **Size** : the number of days each window will be made up of.
- The code initialises the window, prediction and binary prediction lists.
- It then takes the start of each window element (win) by creating an appropriate scaler, that being the data used in the window element.
- It then scales the data that requires scaling one day at a time and adds the unscaled data (brokers and text) to that day as well. This is repeated for "size" number of days.
- The prediction is then updated with the scaled value "delay" days in the future.
- The binary prediction is updated by comparing the price on the last day of the window with the future price of the company "delay" days in the future.
- OUTPUT :
  - **Windows** : a list containing each of these windows that contain the data for "size" number of days.
  - **Targets** : two lists that contain either the exact scaled price "delay" days in the future or the binary prediction based on whether the price has increased or decreased between the last day and the price "delay" days in the future.

4.3 Metrics

As this project only attempts to model the direction of the price change and not predict the exact future price, it requires a metric that can be used on categoric classification. As the data is near a 50/50 split between increase and decrease it is appropriate to use accuracy as a metric. The variance in the accuracy is also used in the cases where a model has been created for each company, this helps to understand if a particular model was causing the mean of the accuracy to be affected.

This project will also use the F1 score as a measure of how effective these models are, even in the presence of class imbalance. The F1 score relates the precision and the recall of the models, this allows for the instances where there is a class imbalance.

4.4 Experimental Settings

Hardware

For all models other than BERT they were trained on a NVidia GeForce 920MX GPU, with a Intel i7-6500U CPU, on a Windows 10 machine.

The BERT model was trained on Google Colabs Tesla T4 GPU's, this was done because the NVidia GeForce 920MX GPU's 4 GB of dedicated memory was not large enough to hold the BERT model. Google's GPU's also have the advantage of being substantially faster at training than the 920MX GPU.

Dependencies

Python – version 3.7.6

Packages

- pandas – 1.0.1
- beautifulsoup4 – 4.8.2
- sklearn – 0.22.1
- XGBoost – 1.0.2
- Tensorflow-gpu – 2.1.0
- PyTorch – 1.5.0
- Karas – 2.3.1
- DateTime – 3.7.6
- NLTK – 3.4.5
- Transformers – 2.11.0
- Numpy – 1.18.1
- re – 2020.6.8
- matplotlib – 3.1.3
- time – 3.7.6
- random – 3.7.6
- seaborn – 0.10.0

Note : Within the NLTK package three database where downloaded, these are contain the stopwords, lemmatizing references, and the pre-trained NLTk sentiment analyser. The downloads are "stopwords", "wordnet", and the "Vader lexicon".

Parameters Used in the Models

Table 4.1: The parameters of each of the models

SVM		Logistic Regression		XGBoost	
Parameter	Type/Value	Parameter	Type/Value	Parameter	Type/Value
C	1.0	C	1.0	eta (learning rate)	0.3
kernel	rbf	dual	False	gamma	0
degree	3	tol	0.0001	max depth	6
gamma	scale	warm start	False	min child weight	1
coef0	0.0	fit intercept	True	max delta step	0
shrinking	True	intercept scalling	1	subsample	1
probability	False	solver	lbfgs	sampling method	uniform
tol	0.001	tol	0.0001	colsample by tree	1
cache size	200	penelty	l2	lambda	1
class weight	None	class weight	None	alpha	0
max iter	-1	multi class	auto	tree method	auto
random state	None	random state	None	updater	grow_colmaker
decision function shape	ovr	n jobs	None	grow policy	depthwise

LSTM Parameters For Stock Price Prediction

The LSTM model used for predicting any change in the future price is described below, it uses three LSTM layers with one fully connected layer. The model uses "binary_crossentropy" as a loss function, this is specifically developed to make predictions between two classes. The model uses the "adam" optimiser this is an update to the RMSProp optimiser. In this optimisation algorithm, running averages of both the gradients and the second moments of the gradients are used. The metric will be accuracy, this monitors how well the models are learning.

The model will be trained for 10 epochs with a 20% validation set (this is 20% of the training data, not the entire data set), the data will not be shuffled so as to keep the data in chronological order, in order to preserve temporal features within the data.

Table 4.2: A description of LSTM model that was used to model the data set described in the previous chapter.

Layer	Output Shape	No. of Parameters
LSTM_1	(None, 1, 180)	389,520
LSTM_2	(None, 1, 90)	97,560
LSTM_3	(None, 30)	14,520
Dense	(None, 1)	31

Total Parameters : 501,631
Trainable Parameters : 501,631
Non-trainable Parameters : 0

LSTM Parameters For Sentiment Analysis

The LSTM model for sentiment has an initial embedding layer that contains the pre-trained 100 dimensional GloVe word embeddings. Due to the three categories that the sentiment can take a softmax activation has been used. The optimizer was the rmsprop, with a categorical_crossentropy loss function. The metric used for this model was accuracy. The details of the model are outlined below. The embeddings assume a maximum vocabulary size of $10,000$ words and only take into account the first 20 words before padding. The drop out layers dropout 20% of the network connections.

Table 4.3: A description of the LSTM model to predict the sentiment of the news articles.

Layer	Output Shape	No. of Parameters
Embedding_1	(None, 20, 100)	1,000,000
LSTM_1	(None, 20, 200)	240,800
Dropout_1	(None, 20, 200)	0
LSTM_2	(None, 20, 80)	89,920
LSTM_3	(None, 50)	26,200
Dropout_6	(None, 50)	0
Dense	(None, 3)	153

Total Parameters : 1,357,073
Trainable Parameters : 1,357,073
Non-trainable Parameters : 0

Chapter 5

Results

This results section is split into 3 sections, firstly the results and performance of the NLP models in predicting the sentiment of the news articles. It then moves on to present the results when training the models on different data sets as individual companies. Finally, the last section is similar to the previous, however, the models are trained on the combined data for all the companies.

The data used to test and train contains $152,043$ examples generated through the rolling window approach using 10 days of data in each window and attempting to predict 10 days in the future. The data is broken into $98,094$ (64.5%) training examples and $53,949$ (35.5%) testing examples.

Table 5.1: Support of the data for the combined data models

	Test	Train	Total
Increase	26,415	45,061	71,476
Decrease	27,534	53,033	80,567
Increase (%)	48.96	45.94	47.01
Decrease (%)	51.04	54.06	52.94

These numbers remain unchanged for each of the models, as the changes in features does not change the number of days of data available. This is the support for all the combined data and does not relate to the individual models that may have different class imbalances.

For the uncombined data set each model is built from that individual companies data, this includes 1,583 data points split into training and testing sets as described in chapter 3.4.

The way this project has chosen to analyse the data is by presenting the situations where the models have performed best, along with the average performance of the models represented in the "ALL" row. The best performance will be determined by the highest F1 score, this will mitigate any discrepancies that a class imbalance will cause.

Immediate Observations

The results for predicting the direction of price movements are substantially worse than hoped for from the outset of this project. Some possible reasons for this are outlined in chapter 6.

Another observation, mentioned previously is the performance of the BERT model on the sentiment analysis; where the results were better than would have initially been hoped for.

5.1 Training and Fine-Tuning NLP Models for the Sentiment Analysis of News Articles

At the beginning of this project, testing whether pre-trained sentiment analysers available from python packages, such as NLTK, are able to work on this task was done. These models are particularly good at getting sentiment from normal text. When testing on the set of labelled financial news articles the accuracy was only 60%. This is far too low to be used and to still expect good results when modelling the data later on in the project. To improve this an LSTM model that has had the weights initialised by the GloVe word embedding was created. It was trained on a data set of sentiment pre-labelled financial news articles. However, this only gave a 55% accuracy. A better approach was needed. This was achieved with the use of the BERT model. Trained with a 71% training, 7.2% validation and 21.8% test split, an 85% accuracy was achieved on this task. This is an excellent result, especially when considering that even humans struggle to understand the sentiment, (8 people where given these articles and only 66% or more of the people agreed on the sentiment). A summary of results, including the BERT model is shown below.

Table 5.2: The performance of various models on the task of predicting sentiment of the labelled news articles

Model	Test (%)	Validation (%)
NLTK vader	60.3	–
LSTM - GloVe	55.3	53.2
BERT	**84.51**	**83**

Table 5.3: Summary of the training of the BERT model over the 4 training epochs

epoch	**Training Loss**	**Valid. Loss**	**Valid. Accur. (%)**	**Training Time**	**Validation Time**
1	0.78	0.59	80	0:00:16	0:00:00
2	0.43	0.47	82	0:00:16	0:00:00
3	0.27	0.49	84	0:00:16	0:00:00
4	0.19	0.49	84	0:00:16	0:00:00

Figure 5.1: Graph of the learning of the BERT model

When looking at the training of the BERT model, we see that the model during its first epoch is still better than any of the other models tested before. After further training we see that the model continues to improve and reaches a maximum validation accuracy on the third and fourth epoch. However, the validation loss does increase after the second epoch. This suggests that any further epochs would not result in improvements in the testing and validation accuracy. This supports the BERT documentation that states the BERT model should be fine-tuned for up to 4 epochs.

Figure 5.2: Confusion matrix of the test results of the BERT model.

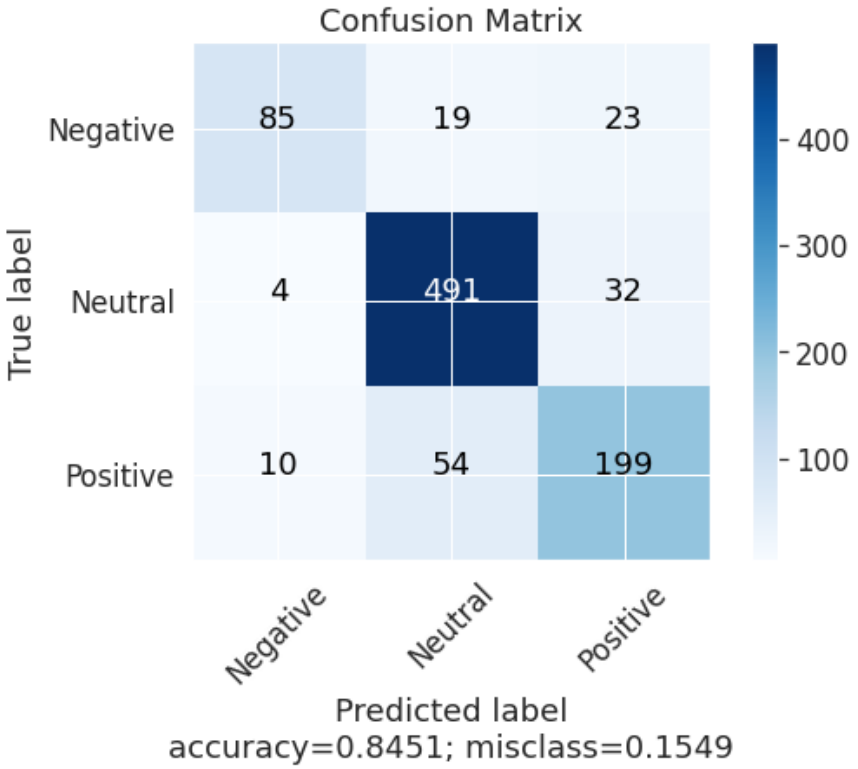
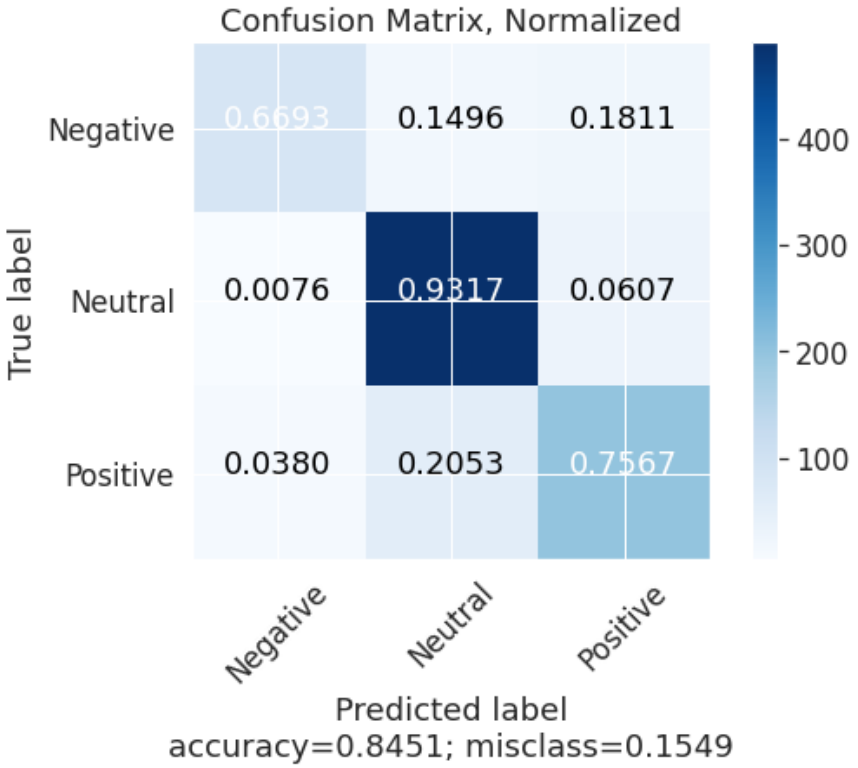


Figure 5.3: Confusion matrix of the BERT models results that has been normalised.



As demonstrated the BERT model is very good at classification, rarely identifying the incorrect sentiment, and if so more often identifying it as neutral instead of positive or negative. This is to be expected as the sentiment of these types of news articles can be quite ambiguous.

5.2 Modelling the Database to Predict Future Prices, Individual Companies

Stock Price Data Only

To begin only the stock price data will be used to train the models, the results are shown in the following table. The All row shows the average accuracy and F1 score for the 98 companies that have been modelled. As the LSTM is hoped to be the best, the top 5 results for the LSTM's F1 scores are presented, as well as the average.

Table 5.4: The accuracy and F1 score of the top 5 and average models using only the stock price data; making new models for each company.

	SVM Accur. (%)	SVM F1	Logistic Accur. (%)	Logistic F1	XBGoost Accur. (%)	XGBoost F1	LSTM Accur. (%)	LSTM F1
KGF.L	55.30	0.551	51.89	0.514	54.58	0.544	56.19	0.562
EZJ.L	50.27	0.494	50.81	0.503	49.91	0.497	55.30	0.552
BHP.L	55.30	0.551	54.76	0.546	55.48	0.555	55.11	0.550
IAG.L	49.01	0.469	51.17	0.499	45.60	0.443	55.12	0.547
ANTO.L	44.34	0.437	48.47	0.483	46.69	0.465	54.58	0.544
ALL	**51.27**	0.456	50.74	0.472	50.88	**0.487**	51.05	0.474

This model performs does perform better than just guessing for some of the models having accuracy of 55% and F1 scores of about 0.55. Despite this, the mean accuracy and F1 scores are lower than would be hoped for, with the SVM model performing the best. Further investigation into building a combined model will be done in an attempt to improve the results.

Results when Adding the Brokers Data

Brokers and Stock Data

The addition of the brokers opinion is aimed to improve the results by trying to give an understanding of professional opinions about the future price of the stock. Using both the stock price and brokers data, the results are shown below:

Table 5.5: The accuracy and F1 score of the top 5 and average models using the stock price data and the brokers opinions; making new models for each company.

	SVM Accur. (%)	SVM F1	Logistic Accur. (%)	Logistic F1	XBGoost Accur. (%)	XGBoost F1	LSTM Accur. (%)	LSTM F1
HSBA.L	48.11	0.466	49.19	0.471	47.40	0.447	55.12	0.549
BHP.L	55.48	0.520	54.76	0.546	55.48	0.555	54.76	0.545
BT-A.L	51.70	0.508	48.65	0.453	52.06	0.518	52.42	0.524
WTB.L	53.32	0.509	49.91	0.491	50.27	0.495	53.14	0.513
RMV.L	58.71	0.451	48.65	0.384	47.22	0.447	55.83	0.512
ALL	50.65	0.413	50.30	0.455	49.98	**0.462**	**51.27**	0.398

Initially, looking at the results we see that they are slightly worse when compared to the financial data only, for all models other than the LSTM. The best models are comparatively worse, warranting further investigation into other factors and approaches that may improve the results.

Using Only the Brokers Data

Now to see how good using only the broker data is:

Table 5.6: The accuracy and F1 score of the top 5 and average models using only the brokers opinions; making new models for each company.

	SVM Accur. (%)	SVM F1	Logistic Accur. (%)	Logistic F1	XBGoost Accur. (%)	XGBoost F1	LSTM Accur. (%)	LSTM F1
AZN.L	49.19	0.330	50.45	0.504	49.91	0.495	58.35	0.583
EZJ.L	49.01	0.330	49.91	0.408	51.17	0.412	60.14	0.573
AUTO.L	46.43	0.317	54.37	0.534	45.24	0.388	57.54	0.572
BT-A.L	48.83	0.483	52.06	0.498	50.45	0.475	53.86	0.524
LLOY.L	50.09	0.480	61.04	0.529	48.65	0.486	53.68	0.517
ALL	50.73	0.403	50.52	0.417	49.40	**0.448**	**51.26**	0.373

Using only the brokers opinions does offer some improvement compared with using both the financial and brokers data together. It is comparable to using only the financial data when looking at the average performance. Although some of the top performing models are better, with this data source achieving a 58% accuracy and an F1 score of 0.58 for the LSTM model. As can be seen the SVM model is significantly worse at predicting the categories that the LSTM does well in.

Using News Articles to Improve the Predictions

In this part of the analysis the sentiment analysis of the news articles will be included as an additional data source in an attempt to improve the performance of these models. The sentiment and how it was derived was outlined in chapter 3 and earlier in this chapter.

Text and stock

This analysis is with the news articles; combining the news articles and stock data will be undertaken.

Table 5.7: The accuracy and F1 score of the top 5 and average models using the stock data and the sentiment of news articles; making new models for each company.

	SVM Accur. (%)	SVM F1	Logistic Accur. (%)	Logistic F1	XBGoost Accur. (%)	XGBoost F1	LSTM Accur. (%)	LSTM F1
ANTO.L	44.88	0.448	51.89	0.516	47.22	0.472	58.53	0.579
TSCO.L	49.01	0.463	50.45	0.502	51.89	0.512	55.66	0.556
SDR.L	51.89	0.498	52.78	0.515	53.68	0.536	56.19	0.551
EZJ.L	49.91	0.490	48.47	0.476	50.45	0.497	54.76	0.548
RSA.L	52.96	0.527	50.81	0.505	52.24	0.522	55.48	0.544
ALL	50.81	0.460	50.88	0.483	50.50	**0.491**	**50.85**	0.462

On the whole these results are comparable with the stock only model, however the more accurate models are better with this data source, than compared with using just the stock price data.

Results when Using Text, Brokers and Stock this Represents All the Data.

These models are hoped to perform the best as they have access to the most complete and extensive data sets that is available in this project.

Table 5.8: The accuracy and F1 score of the top 5 and average models using the stock data, sentiment of news articles and the opinions of brokers; making new models for each company.

	SVM Accur. (%)	SVM F1	Logistic Accur. (%)	Logistic F1	XBGoost Accur. (%)	XGBoost F1	LSTM Accur. (%)	LSTM F1
EXPN.L	45.78	0.458	45.60	0.455	48.65	0.466	60.68	0.545
RR.L	52.42	0.524	54.40	0.536	50.99	0.462	53.50	0.532
BARC.L	54.94	0.355	53.68	0.423	49.73	0.480	52.96	0.528
BHP.L	58.35	0.560	52.24	0.522	51.89	0.518	53.86	0.524
BT-A.L	52.24	0.498	48.65	0.465	54.22	0.542	52.60	0.520
ALL	50.49	0.417	50.32	**0.464**	50.02	0.462	**51.07**	0.418

The expectations of this method to result in one of the best performances has not been achieved. In fact, this is one of the worst performing data sets.

Text Only

Table 5.9: The accuracy and F1 score of the top 5 and average models using sentiment of news articles only; making new models for each company.

	SVM Accur. (%)	SVM F1	Logistic Accur. (%)	Logistic F1	XBGoost Accur. (%)	XGBoost F1	LSTM Accur. (%)	LSTM F1
RDSA.L	49.37	0.493	47.04	0.468	50.63	0.501	57.45	0.560
BA.L	52.96	0.514	58.71	0.537	51.89	0.516	59.61	0.526
ANTO.L	42.73	0.412	53.14	0.515	44.88	0.445	54.40	0.524
SPX.L	62.48	0.522	59.96	0.430	63.02	0.494	63.91	0.500
ULVR.L	44.70	0.411	50.63	0.462	47.40	0.469	50.09	0.496
ALL	51.59	0.452	**51.80**	0.452	51.27	**0.472**	51.20	0.389

This is a surprising result with all four of the different models performing well on average and with reasonable F1 scores recorded for the top performing models.

Text and Brokers data

Table 5.10: The accuracy and F1 score of the top 5 and average models using the sentiment of news articles and the opinions of brokers; making new models for each company.

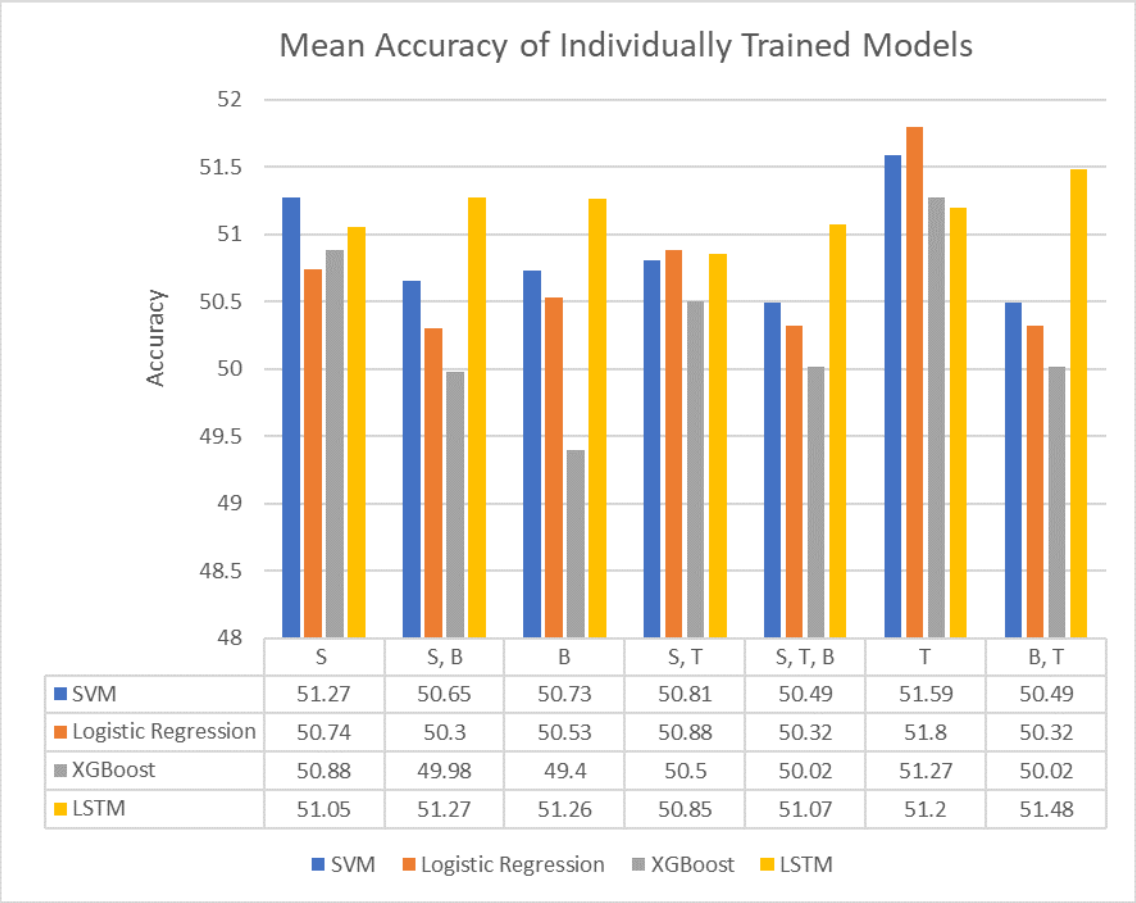
	SVM Accur. (%)	SVM F1	Logistic Accur. (%)	Logistic F1	XBGoost Accur. (%)	XGBoost F1	LSTM Accur. (%)	LSTM F1
BHP.L	58.35	0.560	52.24	0.522	51.89	0.518	56.91	0.561
BT-A.L	52.24	0.498	48.65	0.465	54.22	0.542	54.76	0.547
HSBA.L	46.68	0.452	50.09	0.480	45.78	0.426	53.86	0.535
RR.L	52.42	0.524	54.40	0.536	50.99	0.462	52.60	0.526
LLOY.L	46.86	0.459	52.60	0.496	48.65	0.486	53.14	0.525
ALL	50.49	0.417	50.32	0.464	50.02	**0.462**	**51.48**	0.418

Summary of Previous Results

These graphs show the average performance of the models when they are trained on individual companies data, but trained on different subsets of the database.

- S : Stock data
- B : Brokers data
- T : Text data

Figure 5.4: Mean accuracy of all trained models, one model for each company in the FTSE 100. The accuracy is gained after modelling the data for each of the four models.



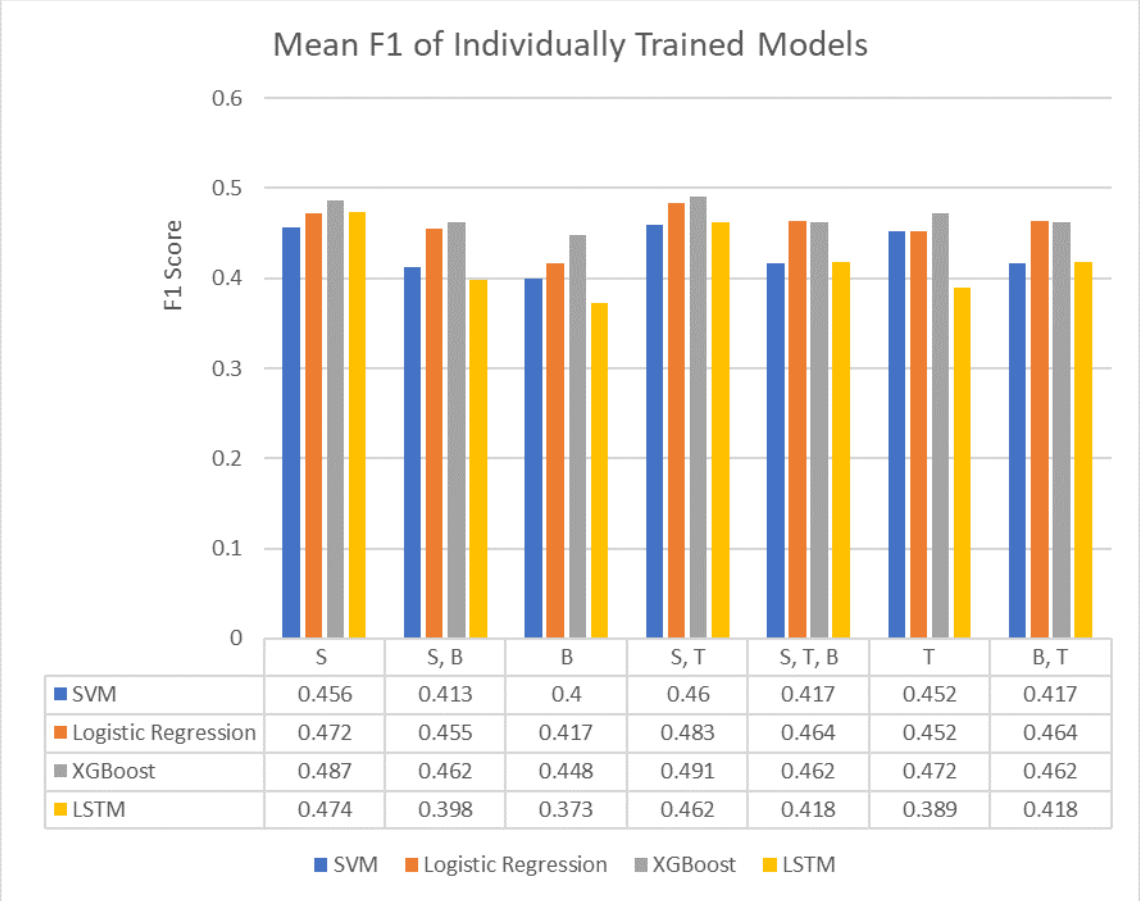
	S	S, B	B	S, T	S, T, B	T	B, T
SVM	51.27	50.65	50.73	50.81	50.49	51.59	50.49
Logistic Regression	50.74	50.3	50.53	50.88	50.32	51.8	50.32
XGBoost	50.88	49.98	49.4	50.5	50.02	51.27	50.02
LSTM	51.05	51.27	51.26	50.85	51.07	51.2	51.48

As can be seen there are a few notable features from the average accuracy.

- The models all perform well on the text only data set.
- The XGBoost model consistently performs worse than the other models.
- The LSTM model performs better than most of the other models when looking at the overall trend between data sets.
- Only the LSTM model consistently performs better than a majority classifier.

The graph below looks at the F1 scores of these models.

Figure 5.5: Mean F1 score of all trained models, one model for each company in the FTSE 100. These are for each of the data sets and each of the four model types.



	S	S, B	B	S, T	S, T, B	T	B, T
SVM	0.456	0.413	0.4	0.46	0.417	0.452	0.417
Logistic Regression	0.472	0.455	0.417	0.483	0.464	0.452	0.464
XGBoost	0.487	0.462	0.448	0.491	0.462	0.472	0.462
LSTM	0.474	0.398	0.373	0.462	0.418	0.389	0.418

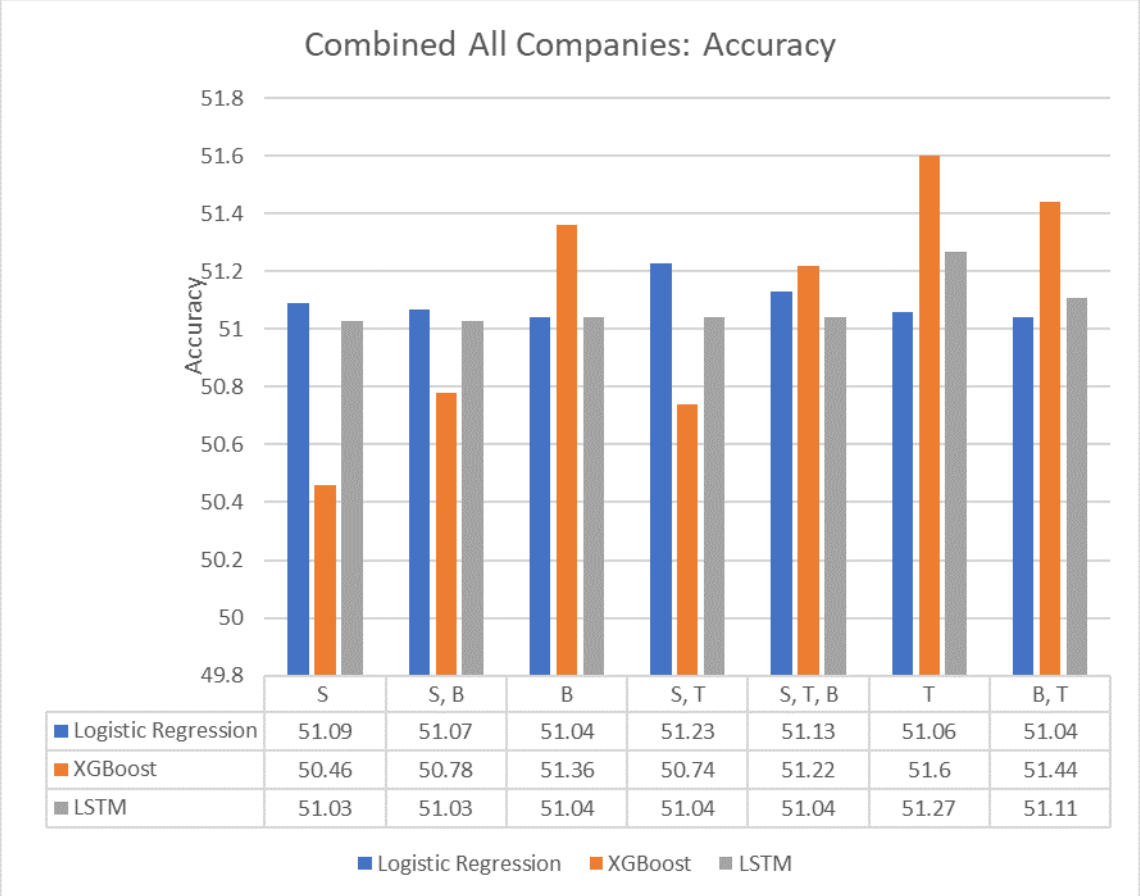
Again looking at some of the notable features of the F1 score.

- Here we see that unlike with the accuracy the XGBoost model for all data sets performs better than the other models.
- There is a much smaller range of F1 scores between the individual data sets.
- The variance in F1 score between the different groups is less than within the accuracy. All of these models seem to have very consistent F1 scores whereas the accuracy had larger variances, both for different models from a particular data set, and between data sets.
- These F1 scores are all very low, the expected result for a random classifier would be 0.5. Our results are less than this which is particularly worrying.
- The best performance is achieved from the stock and text data. This combined with the accuracy being highest for the stock price data, does give a suggestion that text analysis has had some positive effect.

5.3 Combining All the Data for the Companies to Investigate whether this improves results

This project now moves on to look at whether using the data from the companies combined offers improved results. It is hoped that the increase in the amount of data that these models have to work, will result in improved performance. See appendix B for further details about the results. As mentioned in chapter 4, the SVM is not used in this section. This is because the size of the training data is so large that the model fails to train in a reasonable amount of time. The details of the models performance is shown below beginning with the accuracy.

Figure 5.6: Accuracy when combining data from all the FTSE 100 companies for different subsets of the database.



	S	S, B	B	S, T	S, T, B	T	B, T
Logistic Regression	51.09	51.07	51.04	51.23	51.13	51.06	51.04
XGBoost	50.46	50.78	51.36	50.74	51.22	51.6	51.44
LSTM	51.03	51.03	51.04	51.04	51.04	51.27	51.11

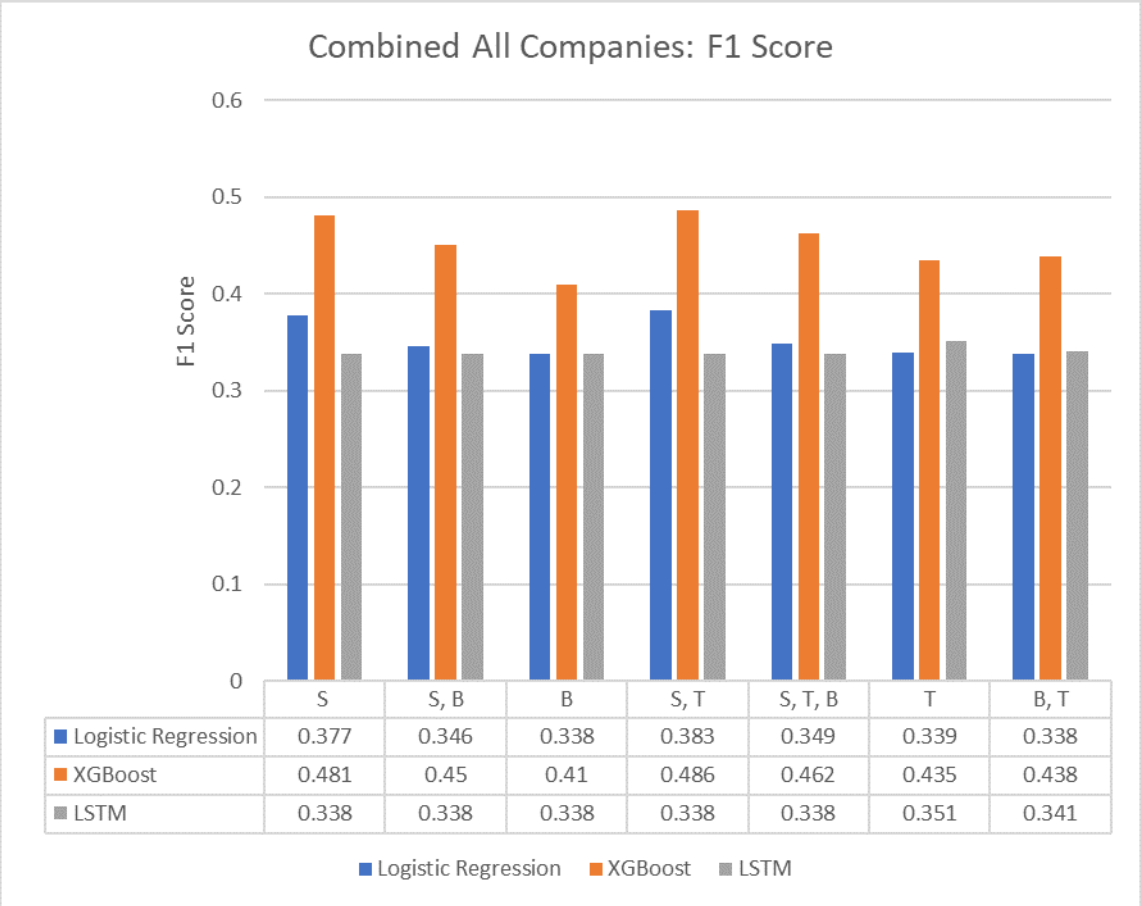
Looking at some of the most notable points of the results, they are shown below.

- The XGBoost model in this part is either the best, by a notable margin, or the worst it seems to perform well in the data sets that contain the text data.

- The use of text analysis does seem to offer some improvements as these models having some of the highest accuracy levels.
- Most of the models fail to achieve results better than a majority classifier.

The F1 scores for the models are now shown below in a similar way to above.

Figure 5.7: F1 score when combining data from all of the FTSE 100 companies for different subsets of the database.



	S	S, B	B	S, T	S, T, B	T	B, T
Logistic Regression	0.377	0.346	0.338	0.383	0.349	0.339	0.338
XGBoost	0.481	0.45	0.41	0.486	0.462	0.435	0.438
LSTM	0.338	0.338	0.338	0.338	0.338	0.351	0.341

Finally, looking at the notable findings within the combined F1 scores.

- The F1 score for all of the models except the XGBoost is significantly less than the performance of the XGBoost.
- The LSTM has some of the lowest F1 scores, which is surprising given the hoped performance of the LSTM.
- There is little difference between the performance of any of the different data sets and no suggestion that any data is specifically useful.
- The performance when taking the text data into account the F1 scores seem to be higher than when not using the text data.
- The use of brokers opinions again seems to result in worse results.

5.4 Analysis of Results

As can be seen the results are not as hoped. The hypotheses seem to be true as there is a slight improvement when adding some different data sources, the text data that is the main focus of this project does seem to offer an improvement but the brokers opinions do not seem to help. The best results occurred when only considering the text data and training models on individual companies.

The sentiment analyser has achieved good results on the test set and on the data collected during this project (the results for all the texts can be found by following the link in appendix A). This part of the project can be seen as a great success, with the model performing better than some of the most widely used bench marking models (LSTM, NLTK-package).

However, the models made for the main focus of this project have not performed as well as was hoped. Using the brokers opinion data does not seem to offer any improvement to the models, in fact it appears that the use of the brokers opinions results in worse performing models. This can be seen both from the combined data set and models trained on individual companies. As can be seen the models often over predict the decrease category, this is the majority class of the data. The models do occasionally perform better than a majority classifier. However, not by much, and in some models worse. As can be seen the LSTM model as hoped for, does perform the best on most of the data sets on the individually trained data sets. However, the best average performance comes from the logistic regression model used on the text only data set where models have been trained on individual companies.

One of the key observations that can be made is that the performance when combining the companies is consistently worse than then when individual models are trained. This can be seen in figures 5.4 and 5.6. As such any modelling in the future should focus on company specific models.

An interesting observation that can be made is that the XGBoost model is consistently out performing all other models in regards to F1 score. When looking at accuracy of the XGBoost it is consistently good in the combined data set, however on the individual companies the LSTM model performs better. The reason the XGBoost model may perform so well is due to the way it trains on data, it does this by focusing on points where the model performs the worse and uses an ensemble method of many gradient boosted trees to build a model from.

There is no clear model that performs the task of accurately predicting the future direction of stock prices well. This project presented 4 very different machine learning methods that are all well known for their performance on tasks similar to this project. So, it seems unlikely that it is the models that are causing the poor performance; more realistic reasons for this are outlined in chapter 6.

The models have in some cases learned relations in the data to make better predictions achieving F1 scores of up to 0.561 and accuracy of 60.68%. In general the performance of the models in predicting the direction of price movements is worse than desired, but the sentiment analysis section was a success.

Chapter 6

Conclusion

6.1 Observations of Results

There have been very minor improvements when using different sources of data. It seems that the use of news articles did offer some improvements, but that and the addition of the opinions of stock brokers did not give any benefit and may have even resulted in worse performance. Not only that, but a majority classifier may in some cases have worked better than the predictive models. That being so, there are examples where the models do learn from the data and achieve reasonable results that out perform a majority classifier.

There are many possible reasons for this lack of performance, some of which are listed below.

- **Volatility** : Due to the extremely volatile nature of the stock market, making any form of prediction is difficult in the best of circumstances.
- **Choice of Sentiment** : The use of sentiment within news articles may not have been the best approach. A better approach may have been using the price changes to give a suggestion of the actual meaning of a sentence.
- **NLP** : There are many other methods that may have improved results other than sentiment analysis, such as the ones mentioned in chapter 2.3.
- **Methodology** : Changing the methodology may improve the results by using an end-to-end model that takes in the news articles and models them together. This was suggested in earlier parts of this project, however, not implemented, because of difficultly.

One reason that the project results are significantly worse compared with other projects, especially ones mentioned in chapter 2.3, maybe due to the market that is being modelled. The papers mentioned are predicting movements within the S&P 500. When looking at this index compared with the FTSE 100 there is a clear difference between them. Both FTSE and S&P has increased over the last 12 years, however the S&P has had a much larger increase compared with the FTSE. This means the data split within the data of the S&P is much larger than in the FTSE hence it is easier to obtain higher accuracy and F1 scores.

6.2 Challenges Faced in this Project

This project initially attempted to model the exact future price of a stock, but after developing it further it became clear that this would not yield the desired result and after some initial experimentation where the Mean Squared Error (MSE) proved to be larger than just predicting no change. The direction of these changes was also investigated. It found that the models did not offer any improvements over attempting to predict the binary categories. As such, this project moved on to exclusively look at a binary classification problem.

The project has faced many challenges, these include a lack of performance in the prediction in all areas. This began with an initially poor performance of the sentiment analysis achieving approximately 60%. However, the BERT model performed well after its implementation, achieving a highly respectable performance. It is unlikely that the sentiment analysis is an issue, rather, that the issue in regards to stock price prediction performance is a result of the methodology. The major challenge in relation to lack of performance has been in the predicting of the binary classification problem which has been very poor. Chapter 6.1 goes over some of the possible reasons as to why this was the case.

Another issue identified early on was within the scaling of the data during the presentation. The problem identified itself through highly accurate results in both the binary classification and exact price prediction. This was caused by scaling all the data as a whole, meaning that the data had been scaled on points in the future. This is clearly an issue as in real world examples the future price is something that would not be known. This problem was solved by scaling each data point independently as described in chapter 4.

6.3 Future Developments

This project, and many like it, have a large scope for future development. For instance, in this project, only 3 source of data were used. Many more where available, such as the use of the type of company and the size of the company. Also, some factors relating to the data that was collected was not included. For example, within the news articles the author was not taken into account. It is quite possible that some sources are more reputable than others and therefore more significant. However, these factors were not taken into effect.

Although many previous pieces of work have used just the news article headline, the BERT model is specially designed to work with large text samples that have many sentences. The model has the ability to relate the meaning of multiple sentences together. This may have resulted in even better performance, and/or understanding of the context of the problem. So improving the performance of the models. In addition to this, building an end-to-end model may also be better than including two separate models. This means that the methodology itself could be adjusted for future projects.

Another future development would be to create additional categories for the classification. Instead of using just increase and decrease, perhaps add a category

that explains large increase, and decrease with another one that is for small changes? This may improve the predictive models as the volatility and variation in the data will hopefully be reduced when predicting larger increase and decrease categories. This leaves events that have smaller price movements, and therefore harder to classify, because they are more susceptible to volatility, in their own category.

One final improvement and future suggestion, would be the analysis of these models for predicting long term or shorter effects (more/less than 10 days in the future) or using different length of time windows on which to base the predictions. As an example, predicting 2 days in the future with 3 days of data in the window. This type of investigation was briefly explored in the project with minimal effect to the results.

Finally, the project concludes that more extensive investigation may yield potentially interesting results and be a profitable avenue to explore.

6.4 Appraisal, Reflection, and Project Management

At this stage in the project the work has now been finished, concluding with writing up the results, methodology, and the analysis within this report. The project progressed well throughout, especially when considering the scale of the project and its complexities. Much of work was needed in the preliminary stages to acquire suitable data and then to process this information to form a reasonable database. Especially true with regards to the sentiment analysis that required the finding of a large data set, so models such as BERT, can learn the semantic features that can be used to find the sentiment of the news articles collected for this project.

Time Management

The time-line for this project ranges from 01/01/2020 to 01/09/2020. Various stages have been set out in order to monitor the progress of the project and allow for the right level depth required to keep up with the time-line. For instance, there is no need to develop a LSTM model with many layers, if doing so will, offer little benefit its accuracy, and may even cause time pressure on more important parts of the project, such as the NLP modelling.

This was an extensive project that required a lot of time during its many stages. This included, areas such as the researching of the problem, the possible approaches to take, and creating a database and all the methods required to do that. Developing a methodology and implementing it, through various machine learning techniques were used that required a deep understanding of them and how they work. Also, attempts were made throughout to improve the performance of various models, such as developing the BERT model for sentiment analysis.

On reflection, the time-line was adhered to well, especially when considering the issues that appeared within the project. This shows that the time-line and the consistent work take, has resulted in a project that has been produced on time.

Bibliography

[1] X. Li, L. Bing, W. Zhang, and W. H. Lam, "Exploiting bert for end-to-end aspect-based sentiment analysis," *ArXiv*, vol. abs/1910.00883, 2019.

[2] X. Li, L. Bing, P. Li, and W. Lam., "A unified model for opinion target extraction and target sentiment prediction.," *AAAI*, p. 6714–6721, 2019.

[3] H. Liu, "Leveraging financial news for stock trend prediction with attention-based recurrent neural network," *ArXiv*, vol. abs/1811.06173, 2018.

[4] B. Wang, H. Huang, and X. Wang, "A novel text mining approach to financial time series forecasting," *Neurocomputing*, Dec 2011.

[5] S. P. D. Basak and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, 2017.

[6] E. O. M. A. Hearst, S. T. Dumais, "Support vector machines," *IEEE Intelligent Systems and their applications 13*, 1998.

[7] G. F. A. N. Refenes, A. Zapranis, "Stock performance modeling using neural networks: a comparative study with regression models, neural networks," *International Journal of Computer Applications*, 1994.

[8] S. P. Das and S. Padhy, "Support vector machines for prediction of futures prices in indian stock market," *International Journal of Computer Applications*, vol. 41, pp. 22–26, 2012.

[9] C. C. C.J. Lu, T.S. Lee, "Financial time series forecasting using independent component analysis and support vector regression, decision support systems," 2009.

[10] B. G. Malkiel, "The efficient market hypothesis and its critics," *Journal of economic perspectives*, 2003.

[11] M. Kanakaraj and R. M. R. Guddeti, "Performance analysis of ensemble methods on twitter sentiment analysis using nlp techniques," in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 169–170, 2015.

[12] S. A. Phand and J. A. Phand, "Twitter sentiment classification using stanford nlp," in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, pp. 1–5, 2017.

[13] M. R. Vargas, B. S. L. P. de Lima, and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 60–65, 2017.

[14] K. M. B. G. Malkiel, "A random walk down wall street," *Norton New York*, 1985.

[15] E. Schoneburg, "Stock price prediction using neural networks: A project report," *Neurocomputing 2*, pp. 17–27, 1990.

[16] V. Akgiray, "Conditional heteroscedasticity in time series of stock returns: Evidence and forecasts," *Journal of business*, 1989.

[17] A. B. M. Gocken, M. Oz calıcı, "Integrating meta-heuristics and artificial neural networks for improved stock price prediction," *Expert Systems with Applications 44*, pp. 320–331, 2016.

[18] C. K. A. A. Adebiyi, A. O. Adewumi, "Comparison of arima and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, 2014.

[19] H. A. K.-J. Kim, "Simultaneous optimization of artificial neural networks for financial forecasting," *Applied Intelligence 36*, pp. 887–898, 2012.

[20] C. S. V. Sehgal, "Sops: stock prediction using web sentiment," *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on, IEEE*, pp. 21–26, 2007.

[21] A. R. M. Skuza, "Sentiment analysis of twitter data within big data distributed environment for stock prediction," *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on, IEEE*, pp. 1349–1354, 2015.

[22] T. L. X. Ding, Y. Zhang, "Deep learning for event-driven stock prediction.," *in: Ijcai*, pp. 2327–2333, 2015.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *CoRR*, 2018. cite arxiv:1810.04805Comment: 13 pages.

[24] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning natural language inference using bidirectional lstm model and inner-attention," 2016.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North*, 2019.

[26] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with bert," *IEEE Access*, vol. 7, pp. 154290–154299, 2019.

[27] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," 2019.

[28] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "Semeval-2015 task 12: Aspect based sentiment analysis.," *SemEval*, p. 486–495, 2015.

[29] X. Ding, Y. Zhang, T. Lui, and J. Duan, "Using structured events to predict stock price movement," *An empirical investigation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[30] "www.lse.co.uk."

[31] "uk.investing.com."

[32] L. Jidong and Z. Ran, "Dynamic weighting multi factor stock selection strategy based on xgboost machine learning algorithm," in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, pp. 868–872, 2018.

[33] L. Li, F. Noorian, D. Moss, and P. Leong, "Rolling window time series prediction using mapreduce," *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration, IEEE IRI 2014*, pp. 757–764, 02 2015.

[34] J. S. S. Hochreiter, "Long short-term memory," *Neural computation 9*, 1997.

[35] Q. V. L. I. Sutskever, O. Vinyals, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, 2014.

[36] J. S. F. A. Gers, N. N. Schraudolph, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, Aug 2002.

[37] S. K. H. Y. Zhai, A. Hsu, "Combining news and technical indicators in daily stock price trends prediction," *International symposium on neural networks, Springer*, pp. 1087–1096, Aug 2007.

[38] "Yahoo finance, https://uk.finance.yahoo.com/."

Appendices

Appendix A

Database and Code used in this project

The code used throughout this project can be found on my GitHub page:
https://github.com/Jack-Wells/Stock-market-prediction-NLP-BERT

Table 6.1: A summary of the files that are available on the GitHub related to this project, the "Results" folder contains 7 files and the "all_data" folder contains 100 files.

Jack-Wells Added file	Size
Folder: RESULTS	∼70 KB
Folder: all_data	∼18 MB
BERT_model_sentiment.ipynb	207.3 KB
Broker opinion scraper.ipynb	69.73 KB
Brokers both .ipynb	71.07 KB
Brokers3.csv	1.61 MB
CSV maker.ipynb	14.36 KB
Dissertation_Interim_Report(2).pdf	1.26 MB
Dissertation_proposal.pdf	1.49 MB
FINAL MODELLING.ipynb	17.39 KB
README.mb	17.39 KB
Sentences_66Agree.txt	566.65 KB
stocks_name.txt	3.71 KB
text scraper.ipynb	16.13 KB
text.csv	4.34 MB

Some of the important files are the ones that relate to the BERT model file, "BERTmodelsentiment.ipynb". The three web-scrapers and API's being "Broker opinion scraper.ipynb", "text scraper.ipynb", and "CSV maker.ipynb". Finally, the "all_data" folder contains the complete database of processed data, these contain one file for each company and are the files used in the "FINAL MODELLING.ipynb" file to generate results.

Appendix B

Tables of Results Used When Combining Data

Table 6.2: The accuracy and F1 results of the combined data set (that contains 98 companies) using only the stock price data.

Model	Accuracy (%)	F1 Score
Logistic Reg.	**51.09**	0.377
XGBoost	50.46	**0.481**
LSTM	51.03	0.338

Table 6.3: The accuracy and F1 results of the combined data set (that contains 98 companies) using the stock price and opinions of brokers data.

Model	Accuracy (%)	F1 Score
Logistic Reg.	51.07	0.346
XGBoost	50.78	**0.45**
LSTM	**51.03**	0.338

Table 6.4: The accuracy and F1 results of the combined data set (that contains 98 companies) using only the brokers data.

Model	Accuracy (%)	F1 Score
Logistic Reg.	51.04	0.338
XGBoost	**51.36**	**0.41**
LSTM	51.04	0.338

Table 6.5: The accuracy and F1 results of the combined data set (that contains 98 companies) using the stock price and the sentiment analysis of news articles.

Model	Accuracy (%)	F1 Score
Logistic Reg.	**51.23**	0.383
XGBoost	50.74	**0.486**
LSTM	51.04	0.338

Table 6.6: The accuracy and F1 results of the combined data set (that contains 98 companies) using all available data, this includes sentiment of news articles, brokers opinions, and stock price data.

Model	Accuracy (%)	F1 Score
Logistic Reg.	51.13	0.349
XGBoost	**51.22**	**0.462**
LSTM	51.04	0.338

Table 6.7: The accuracy and F1 results of the combined data set (that contains 98 companies) using only the sentiment analysis of news articles.

Model	Accuracy (%)	F1 Score
Logistic Reg.	51.06	0.339
XGBoost	50.6	**0.435**
LSTM	**51.27**	0.351

Table 6.8: The accuracy and F1 results of the combined data set (that contains 98 companies) using opinions of stock brokers and the sentiment analysis of news articles.

Model	Accuracy (%)	F1 Score
Logistic Reg.	51.04	0.338
XGBoost	**51.44**	**0.438**
LSTM	51.11	0.341