Predicting Stock Market Movements Using Social Media And Machine Learning

Tsichli Vasiliki

SID: 3308180022

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

December 2, 2019

THESSALONIKI – GREECE

Predicting Stock Market Movements Using Social Media And Machine Learning

Tsichli Vasiliki

SID: 3308180022

Supervisor:	Assistant Professor Christos Tjortjis
Supervising Committee Member:	Dr. Christos Berberidis
Supervising Committee Member:	Dr Stavros Stavrinides

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

December 2, 2019

THESSALONIKI – GREECE

Abstract

Using data from microblogging websites and analyze them to obtain their sentiment has become a popular approach for market prediction. However, many authors that analyzed this kind of data, stress the noise these data contain, and how difficult is to distinguish truly valid information. In this dissertation we collected 782.459 tweets starting from $2018-11-01$ until $2019-31-07$. For each user day, we create a graph (271 graphs in total) with the users that have tweeted and their followers, finally, we use this graph to obtain a PageRank score for each user. This score is then multiplied with the sentiment data. Our results indicate that using an importance-based measure, such as PageRank, can improve the scoring ability of the models, as the PageRank data set achieved, on average, a lower mean squared error than the economic data set and the sentiment data set. Lastly, we tested multiple machine learning models, the results show that XGBoost is the best model, with the random forest being the second best and LSTM being the worst.

Acknowledgements

To my beloved husband,

who stands by me even when I kneel.

Contents

List of Figures

List of Tables

Listings

1 Introduction

Stock market forecasting is an important academic topic, which has attracted academic interest since the early 1960's [1]. Although a lot of effort and time has been spent on predicting financial time series, the results of the research are not robust. In recent years a lot of researchers have shifted their focus from classical econometric approaches to machine learning approaches. With the rise of microblogging platforms, such as Twitter, StockTwits and others, information is more available than ever before and given that emotions can have a significant effect on economic decisions [2], alongside with herding phenomena [3], one can assume that mining information through such microblogging platforms might be the key to achieve better results in predicting stock market movements.

1.1 Background

As we already stated stock market forecasting has drawn a lot of academic attention since the 1960's. The first model that revolutionized how the stock was evaluated is the Capital Asset Pricing Model (or CAPM for short). CAPM was developed[1] by William Sharpe [5] who built on top of Markowitz's diversification theory. The model is fairly simple and is based on the sensitivity that a stock's returns exhibit over the systemic risk (or market risk), which is expressed quantitatively through the use of a factor, called beta and which is symbolized by β.

An important remark we have to make is that CAPM measures the return of a stock in accordance with the market risk. Every other risk that stems from the stock itself can be diversified as Markowitz proved in the portfolio theory and thus, there is no point in measuring it. Although CAPM has been a fundamental tool with which asset managers make decisions, it has been criticized a lot by academics because by its nature, it has a lot of problems. It has been proven that the model is not robust, and that it fails to give accurate results consistently. Fama and French [6] stated that the model is not robust and that a model that takes into account the size and the ratio of accounting over stock market value is more accurate.

Fama's and French's research gave the incentive to start looking for other factors that may be affecting the returns of a stock and this gave birth to a whole new way of evaluating a stock, which is called technical analysis. Technical analysis is not an academic principle, rather it is based on ratios and indicators that capture the momentum of the stock market. Although technical analysis is not based purely on academic research, it is used extensively and it is a common practice.

In recent years there has been a lot of effort to construct indicators or ratios based on the information of the microblogging community. Essentially, those indicators provide an overall sentiment over the market or a particular stock, and thus, the trader can have a more objective metric about the "feelings". Moreover, this data might contain useful information that would be unavailable otherwise. On the other hand, this approach contradicts one of the most fundamental economic theories, which is the Efficient Market Hypothesis. As Fama [1] suggested in his seminal paper, the price of a given stock embodies all the prior information available and thus it is impossible to forecast future values since the current ones reflect everything. Moreover, in efficient-market

[1] There is a dispute on who deserves credit about CAPM, for more information check [4]

hypothesis (EMH), it is believed that the market adjusts the prices instantly as the news spread, and as Fama [1] noted, the most probable future price is the current price.

Nevertheless, recent empirical research provided evidence that sentiment plays an important role and can act as a determining factor of the stock market returns.

1.2 Problem Statement

One of the biggest problems encountered by the researchers that used data from Twitter of other relevant sources is that they are noisy [7, 8, 9, 10], thus yielding spurious results. To deal with that problem, the authors either choose a specific news source, such as MarketWatch [11] or Thomson Reuters [12], but this approach might lead to overlooking important information. Another issue is that they use a lot of data which might hinder their research in terms of efficiency and statistical robustness [13].

1.3 Objective of the Study and Key Findings

Our objective is to provide a more efficient way of handling those massive data, by looking for and distinguishing those data that matter the most. To achieve that, we use graphs that are constructed based on users and their data accordingly. We believe that our approach solves the problem of noisy microblogging data, without disregarding any useful information that might exist. Given our hypothesis we expect that the dataset which accounts for the noise in the data have a better score than the simple sentiment dataset.

1.4 Structure of the Study

The rest of the dissertation is organized as follows. The next chapter describes the existing literature, what methodologies and data were used and what has been achieved in this particular field of study. Afterward, we present the data and the methodology we used. In the next section, we present the results of the models we constructed and tested. Lastly, in chapter 5, we conclude and summarize by stating the limitations of this work as well as our suggestions for further advancing this field of study.

2 Literature Review

The current chapter provides a review of the relevant literature of sentiment analysis on microblogging platforms and machine learning techniques to predict stock market returns. Sentiment analysis on financial news or forum posts is not a relatively new idea [14], but it has recently gained a lot of attention since more data are available and the tools for processing the data are becoming more and more trivial. There exist numerous papers that examine this subject, and there are a plethora of methodologies used. Thus, we opted to break the literature in two main parts. In the first part, we provide the reader with an overview of the studies which use statistical approaches, such as correlations and OLS Regression. The other part examines the literature in which some machine learning approaches are used, such as decision trees, neural networks, etc.

To have a far-reaching variety of papers to examine, we decided to use the ACM Digital library, the IEEE Xplore Digital Library, the Science Direct, and the Springer Link (Figure 1). The keywords we used were "Stock Market Sentiment Analysis" and "Stock Prediction Sentiment".

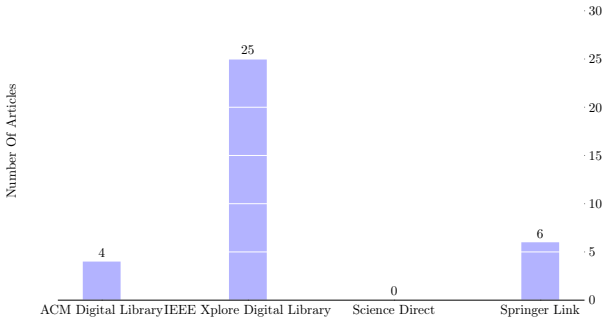

Figure 1: Final Results Per Digital Library

In order for our research to yield a substantial, but manageable, number of papers we decided to pose a restriction on the year of publication; we chose the years $2016 - 2019$[2] (Figure 2), but we did not limit our research to any type of publication, with the exceptions that it had to be written in English and be accessible to us.

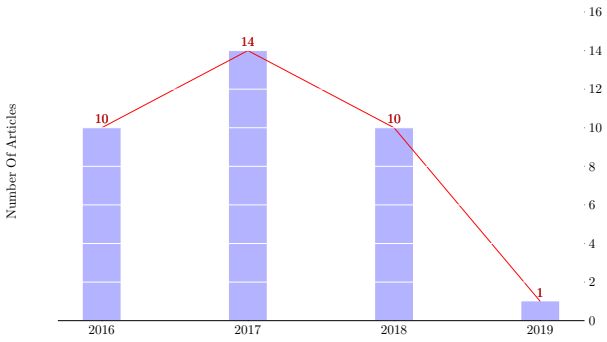

Figure 2: Results Per Year

[2]We cited papers from years before those, but these papers did not originate from our search, rather from the papers we collected.

2.1 Statistical Approaches

The sentiment is an opinion of a view on a subject that is carried by a person. In recent years, and because of the more available data from social media, blogs, and forums, sentiment analysis has attracted a lot of academic research. In recent years, there has been a plethora of studies examining the prospect of sentiment analysis as a predictive factor of the stock market. It started from the seminal papers of Bollen et. al [2] and Tetlock et. al [14]. In [2], the authors used the price of the Dow Jones Industrial Average and obtained the sentiment by OpinionFinder and GPOMS. GPOMS variables described the moods of the public, which allowed the authors to have a more accurate result. They chose a period in which both elections and Thanksgiving were included. Afterward, all the variables were normalized.

The Granger causality showed that the time lag that has the biggest predictive power is 3 days. Moreover, when the authors dropped the OpinionFinder variable and examined only the calm variable from GPOMS, the score was improved. Also, they showed that using the happy variable does improve the MAPE of the model but it drops the direction accuracy, which, according to the authors, indicates that there is a non-linear relationship between calm and happy variables. Moreover, authors in [14] analyzed financial news about specific firms and calculated a ratio of the negative words contained in the news articles. In their model, the authors included economic variables as well. More specifically, they used lagged earnings, size, book-to-market ratio, and trading volume. All of those variables are commonly used in the prediction of stock market returns. Although the authors use simple OLS regression, they find that the ratio of negative words can forecast low firm earnings. Furthermore, the stock prices seem to underreact to the information contained in the negative words. Lastly and more importantly, the best predictor for stock market prices is the ratio of the negative words that came only from the news that focus on the fundamentals.

Another attempt to capture the sentiment was made by [15]. The authors used Google Trends data to obtain a proxy for market sentiment. They hypothesized that the more an investor demands information about a stock, the more risk-averse they will be. To test their hypothesis, the authors used 30 stocks traded on NYSE combined with the internet search volume in the Google Trends database. The model used was a GARCH$(1, 1)$, a model that is designed to specifically capture volatility and idiosyncratic shocks. The results showed that the weekly search volume indeed affects stock returns. On top of that, it seems that market-related information is what investors take into account the most. Lastly, the main finding was that information demand has a positive relationship with the risk aversion.

Sprenger et. al [16] explored the same concepts. To do so, they built eight hypotheses around the behavior of investors and the microblogging activity. More specifically, they examined whether an increase in message volume is associated with higher stock returns, higher trading volume, and higher volatility. Moreover, they focused on users, examining the belief that users who give quality advice often are more influential and that those advices spread more widely. To examine those hypotheses, the authors focused on stocks traded on S & P 100 for six months. The 250.000 tweets were classified using Naive Bayes classification. Afterward, the authors used OLS regression

and Fama-MacBeth cross-sectional regressions. Their results indicated that a tweet does contain valuable information and that is associated highly with returns, trading volume, and volatility. Moreover, the authors did accept the hypothesis that users with higher quality content are more influential in terms of followers and retweets. On the other hand, higher-quality tweets do not receive the attention they should.

Another study that focuses on how to extract information based on the quality of the tweets is [17]. The authors used 5 highly traded stocks from the technology sector, and the related tweets to examine the relationship between the returns and the market sentiment. To achieve that, the authors designed a data cleansing process. This process can be decomposed in 5 steps. The first step is the lowercase transformation, where all the letters become lowercase – this is because the words in the lexicon are described in lowercase letters. Afterward, they remove spurious tweets or tweets with no context, such as those only with URLs or those that were just retweets. In step 3, the authors cleanse the tweets from punctuation marks using regular expressions. In the next process, the stop words are removed. The last part of the process is tokenization. In this part, each tweet is being split based on the space character and then, converted into a list. All those steps lead to the formation of a bag-of-words, and this, in turn, is transformed into N-grams. Each word in those N-grams is labeled according to the lexicon as "very positive", "positive", "neutral", "negative", and "very negative".

Another aspect that authors in [17] pay a lot of attention to is the influential users. A metric to distinguish those users is the number of followers a user has. The authors drew two main conclusions from their analysis. First, using sentiment alongside historical data helps improve the prediction accuracy of VAR models. Their best model achieved a 73.96% accuracy in predicting the direction of a stock. Secondly, they concluded that the minimum daily tweets that are needed for stock prediction are 2.500.

An effect that has been extensively examined is if there is causation between sentiment and stock market movements. Authors in [18] used the Granger Causality test to test this hypothesis. They employed a similar to [17] procedure in preprocessing to extract the sentiment. Afterwards, they classified the emotions into eight categories, namely "afraid", "amused", "angry", "annoyed", "don't care", "happy", "inspired", and "sad". After constructing the eight variables, they tested for Granger causality between those variables and the Korean Stock Price Index. Granger causality is a statistical test developed by Granger [19] in 1969. According to its definition, a variable y_t Granger-causes a variable x_t if the x_t can be predicted with better accuracy by using past values of y_t rather than not using them [20]. The results indicated that different emotions affect stock market behavior with different time lags and different metrics. For example, the emotions classified as "amused" and "happy" affect the next day's stock price, whilst other emotions affect the next day's trading volume and the day's after.

Granger causality tests have been utilized by many authors to test if there is a relationship between microblogging activity and stock market movements. For example, authors in [21] use this test. The authors collected data for the period between November 15, 2010 and April 20, 2011. The restrictions that were placed were that all tweets had to be a retweet only, that they had to contain the words "Hope" and "Fear" or "Worry", and that the location of the users had to be

in the United States. Lastly, they limited their research only for tweets of economic sentiment by using keywords such as "dollar", "$", "gold", "oil", "job", and "economy". Their metric for the public sentiment was simply the counting of the tweets that contained the above words. Thus, they built six public opinion times series. The market data were time series of Dow Jones, S & P 500, and NASDAQ, WTI Crude Oil, and the exchange rate between USD and CHF for the same period.

Afterward, they proceeded to the statistical analysis. As a first indication, they used pairwise correlations between the Twitter keywords and the market data. Correlations were calculated for 3 time lags in the Twitter keywords. Their results showed that the dollar keyword was strongly correlated with all of the economic variables, the oil and the economic keywords with the oil movements, and the gold keyword with the exchange rate, but not with the gold variable. All the coefficients were positive, which means that when the users speak more about a topic, that indicates an increase in the corresponding economic variable. Moreover, the effect seems to decay after the 1 day time lag. Lastly, they employed pairwise Granger causality tests. The results showed that the dollar keyword has the highest predictive power in the stock market data, that the oil keyword affects the oil prices, and that the gold and job keywords have significant predictive power over the gold economic variables. The most interesting observation was that the gold keyword does not have a causation relationship with the gold but that it does have one with the exchange rate.

Authors in [22] tried to explain the volatility of the stock market using the Twitter measures that [13] has presented. That is, they used Bullishness and Agreement variables constructed by the number of positive and negative tweets per day. The economic variable used was the Dow Jones, the NASDAQ-100 and the returns of technological companies, such as Google, Amazon, and Yahoo. Moreover, they constructed a measure for volatility using the Garman-Klass equation. The correlation analysis indicated that there are significant relationships between all economic variables and Twitter features. On the other hand, those relationships differ significantly from one economic variable to another, which indicates that for each economic variable, a different model must be built. Moving on to the Granger causality tests, the results showed that only the same Twitter features have a significant causal relationship with the economic variables. Given those predictors, EMMS models were used for the Dow Jones and NASDAQ-100 variables. The results showed that the predictors had a positive impact on all the statistical metrics, such as R-squared, MaxAPE and Direction. Lastly, the authors examined what time series can be more accurately predicted. They ran the same experiments for daily, weekly, 2-weekly, 3-weekly, monthly, 5-weekly, and 6-weekly times series. The results showed that the monthly time series are more appropriate to detect anomalies in economic time series.

Detecting economic anomalies, such as bubbles and crashes, in time, has been an important research topic for decades. In [23], the authors emphasized on detecting such anomalies using tweets to identify them. More specifically, the authors studied the relationship that does not focus on trivial fluctuations, since they believed that some volatility exists due to the daily movements or other factors, but rather emphasize the relationship between the financial news and stock market returns. Their experiments were conducted using Hong Kong's stock market. The results showed that the density (or volume) of the financial news contributes to the prediction of stock market

anomalies. On the other hand, the authors noted that the time lag that has the highest predictive power is one day.

Many authors do not examine the relationship between the returns of stocks' but the volatility. Volatility is a term used in finance to indicate the fluctuations of stock returns – it is measured with the variance. Because time series data exhibit a phenomenon called volatility clusters, simple regression techniques have been proved to provide spurious results. That is because different fluctuations in different periods result in residuals with non-constant variance, which is a necessary and sufficient condition to have robust results. Examining the relationship between volatility and microblogging data, like authors in [24] did, solves this problem. The authors constructed an emotional index based on microblogging activity in Chinese forums and examined the relationship with the volatility of the SSE Composite Index for the second semester of 2016. Their results indicated that there is a strong relationship between those two indices.

2.2 Machine Learning Approaches

As we have seen, statistical approaches focus on determining if there is a causal relationship between microblogging data and stock market movements and do not put a lot of effort into predicting. That is because of the volatility and the problems that it creates [3]. So in recent years a lot of researchers use machine learning approaches, such decision trees, XGBoost and neural networks. In this section of the literature, we are going to review papers, which use those or similar techniques.

One of the first studies that employed such techniques is [25]. The authors combined both economic and textual data. The news source comes from two well-known sites for business news, Forbes and Reuters. The economic variables include β, the price per earnings ratios, and profit ratios amongst others for five selected stocks, Cisco, eBay, Microsoft, Teva Pharmaceutical, and Yahoo. Afterward, they proceeded to forecast the trend of each stock. Here, the authors tried to forecast the direction of the movement for each stock. To do that, they used Decision Tree using the algorithm C4.5. Lastly, to evaluate their results, they built 3 trading strategies. The first is a random strategy, the second is based on the recommendation of the system, and the third is a single day trading strategy. The last strategy is a trading strategy where the investor sells all of their stocks at the end of each day. The strategy with the biggest ROI[4] was the third one, but the authors highlighted that the number of switches on positions is a lot and that those expenses are not accounted for in their research. Nevertheless, both strategies that followed the recommendation of the system were more profitable than a random one, with their best accuracy being 83.3% and the worst one 77.0%.

Decision trees have been proven useful in predicting the direction of stock market movements. In [26], the authors used a stacked classifier with Decision Tree and SVM to forecast the Bombay Stock Market. At first, they used the latent Dirichlet allocation algorithm to extract relevant topics from a financial news repository. They extracted 25 such topics and disregarded texts that did not correspond to those topics. From those 25 topics only 8 exhibited high correlations with the BSE Sensex index or with its volatility. Lastly, they used the stacked classifier to forecast the

[3] Usually, volatility creates non BLUE estimators

[4] Return on Investment is given by $\frac{\text{Profit}}{\text{Expenses}}$

stock market. The best accuracy of their system was 62.02% with a training sample that ranged from August 2005 to December 2007 and a testing sample that ranged from January to April of 2008. Authors in [27] used another source of information for sentiment data. Alongside economic variables such as a book for market value and opening prices, they used the number of visits to each company's Wikipedia page. Afterward, they used the random forest classifier, which is an ensemble of decision trees, to test for relationships. The authors concluded that although the number of visits to a company's Wikipedia page affects the results, no causal relationship can be established.

Another model that is extensively used to forecast the stock market is Naive Bayes [28]. Many authors have used Naive Bayes either to classify the sentiment of the text or to study the movement of the stock market. For example, in [29], authors used Naive Bayes as a classifier to study the relationship between sentiment and Apple's stock. The authors mined data from a specific site for financial news called Stocktwits and classified them using Naive Bayes. Their results indicated that there is a relationship between the sentiment and the stock market movement, which becomes strong when investors exhibit a bearish[5] behavior. Furthermore, authors in [30] used 5 machine learning algorithms to classify tweets: a support vector machine, a Naive Bayes classifier, a Decision Tree, a Random Forest, and a neural network. The best accuracy was achieved by the Random Forest classifier, with 60.39%, whilst Naive Bayes scored 56.50%. Afterward, they used simple linear regression to predict stock prices. They found that using the positive tweets percentage produced by Naive Bayes classifier yielded a better result than using the same percentage from the other classifiers. Another comparative study is [31]. The authors used TF and TF-IDF to extract features from 42.000 tweets. They used those features as inputs to a neural network-based classifier and to a Logistic Regression classifier. The results indicated that TF-IDF has better overall and average accuracy in both models. The logistic regression classifier had better accuracy only with the term frequency features.

Classifying textual data based on Naive Bayes or other machine learning approaches is not currently used a lot. Most of the recent literature focuses on extracting the sentiment based on a lexicon such as Vader Sentiment or using TF-IDF and a bag-of-words to calculate the sentiment. So the research community has shifted from classifying the textual data to embedding those classifications to models in order to more accurately predict stock market movements. For example, in [32], authors used TF-IDF and a bag-of-words with SentiWordNet 3.0 to calculate sentiments. They mined tweets for Apple from October 2011 until March 2012. They classified each tweet to represent a feature, such as the CPU, of the iPhone and calculated the respective association rules. Those association rules were used as weights to their model. They contrasted their results to three other well-known algorithms: SVM, decision tree, and Naive Bayes. The best score in terms of accuracy was achieved by their algorithm, with 76.12%. SVM ranked second with 70.75%, decision tree ranked third and Naive Bayes last. In addition to these papers, authors in [33] used 3 machine learning algorithms to classify tweets. The authors represented the corpus both with Word2Vec and with N-gram. The best algorithm in both representations was the Random Forest, with an

[5]Bull/Bear: Financial terminology to indicate when the investors feel positive/negative respectively towards a stock.

accuracy of 70.18% for Word2Vec and with 70.49% for N-grams.

One of the most interesting machine learning models used is Support Vector Machines. SVMs have been used considerably [28, 34]. In [28] the authors considered the same models but also introduced a sequential minimal optimization. The data used covered a period of one month, and they were mined from credible news sources, such as The Wire, Bloomberg, CNN, etc. They trained 3 models, one with only the extracted news, one with extracted news and the baseline of stocks' prices and the regressed estimate for stocks' prices. The authors evaluated the result based on the mean squared error and directional accuracy. The model with the best score was the one that utilized the strength of both the public sentiment data and the economic variables, with an MSE of 0.04621 and a directional accuracy of 57.1%. Similar results were achieved by [35]. This research presented interesting results in terms of posts in microblogging platforms. The authors used Yahoo's Message Boards and financial data from Yahoo Finance. They divided those messages into two categories, those in which there was an explicit trading strategy (buy, hold, and sell) and those in which there was none. The authors classified the messages from the second category. Their results indicated that out of all the messages, 68.8% were spam and that messages posted on trading hours had a lower probability of being spam. To have more accurate results, they tried to identify spammers and then, to disregard the message coming from such users. Lastly, using an artificial neural network, they achieved an accuracy of 57.38% when they included the sentiment as a variable. The ANN without the sentiment achieved 56.67%. Also, the results were better when the overall sentiment was to sell a stock, that is if the behavior of the investors was bearish.

More recent results in predicting the direction of stock prices with SVM have yielded better results. In [33], when they trained the model with 90% of the data, the LibSVM model achieved a 71.82%. Similar results were obtained by [36]. The author compared two classifiers for tweets and afterward, used the features that were obtained by the best classifier as an input to predict the stock's price. At first, the author gathered all of the relevant data from twitter for 8 companies and for 7 days, which is a small period compared to other studies. They used all of the regular steps for data prepossessing and filtering. For the first part, the author noted that the SVM classifier outperformed Naive Bayes, achieving an accuracy score of 81.51%, whilst Naive Bayes achieved 80.04%. Lastly, with an SVM, the accuracy of the predicted stock market movement was 84.8%. Kordonis et. al [37] obtained similar results. The authors compared Naive Bayes and SVM in sentiment classification, and they found that the two classifiers achieved analogous results in terms of accuracy. More specifically, Naive Bayes had an accuracy of 80.6%, whilst SVM had 79.3%. Afterward, they proceeded to extract the features in which their market model would be based upon. The authors extracted 7 features, which were:

1. percentage positive sentiment score
2. percentage negative sentiment score
3. percentage neutral sentiment score
4. close price

5. HLPCT

6. PCTchange

7. volume

Lastly, given these features, they trained an SVM model to both predict the movement and the price (i.e. both a classifier and a regressor). Their results showed that capturing the sentiment effect strengthens the prediction result. In terms of movement direction, the model achieved an 87% accuracy, whilst in terms of the stock price, the average error was under 10%. Moreover, their results on a specific date achieved an average error of 1.668%. In [38], authors compared a plethora of models both in sentiment classification and in stock market direction classification. More specifically they trained and tested 5 different classification methods: Logistic Regression, SVM, Decision Tree, Boosted Tree, and Random Forest. The results indicated how useful SVM can be, given that it has the best accuracy score (on both training and test sets). On predicting a stock's price, they found that using stock-specific cashtags[6].

Although SVM can be a very powerful tool, more advanced techniques may yield better results. For example, authors in [39] compared multiple models as well. Their selection included Logistic Regression, LSTM (Long short-term memory neural networks), SVM, Naive Bayes SVM, and an ensemble of methods. The most accurate method in sentiment classification was the ensemble of methods, which consisted of a weighted model of the Naive Bayes SVM and the LSTM models. The model achieved an accuracy of 71.3%. Lastly, the authors used classified information on stocks to optimize their portfolio strategy. The average total return for seven months was 19.54%, and the authors noted that their strategy was more stable than the market itself, which also indicated less risk.

There is a vast and growing literature that uses artificial neural networks comparing them to other machine learning techniques. In most of this literature, the best results are achieved by ANNs. For example in [40], the authors compare multiple techniques to test if the efficient market hypothesis (EMH) holds in exchange rates. More specifically, their data consisted of four exchange rates, the USD/EUR, the USD/JPY, the USD/GBP, and the USD/AUD for the period between January 2, 2013 and December 26, 2013. The models used were Logistic Regression, SVM, Naive Bayes, KNN, Decision Trees with boosting (AdaBoost and LogitBoost), and ANN. Moreover, to test the efficiency of using sentiment data, they constructed 5 different input sets. The first one was simply the past values of the exchange rates. The second one was the number of bearish and bullish posts per day. The third was the second set plus the total posts per day. Lastly, the fourth and fifth data sets were a combination of the first data set with the addition of the second and third. Their results indicated that there is not a methodology that consistently outperforms all of the others, rather than the outcome is susceptible to the exchange rate. On the other hand, KNN, SVM, and ANN exhibit higher forecasting accuracy than the other methodologies. Lastly, the authors did obtain their best results from the data sets which included sentiment information, thus the weak form of EMH can be rejected.

[6]Much like the hashtag (#), financial analysts use the cashtag ($)

More on comparative analysis, authors in [41] used 4 difference models. Their data set consisted of 6 stocks of the Shanghai Stock Exchange, which are not named. The models used in this research are LSTM, Regression, Multi-Layer Perceptron, and Recurrent Neural Networks. Their results showed that the LSTM model outperformed all of the other methodologies for all of the six stocks with an average mean error of 0.21. The second best was the recurrent neural network (mean error of 0.43), and the third one was the regression model (mean error of 1.51). The MLP (mean error of 1.98) was last. Moreover, authors in [42] used an artificial neural network with one hidden layer to model the stock price movements. Their data set included 5 technological companies (Apple, Google, Microsoft, Oracle, and Facebook) for the period from 01/01/2015 to 22/02/2016. Their results were on par with those of [40] on that each stock must be treated as a different case and thus, a different model must be tuned. Although the authors used only an ANN, the number of neurons changed significantly from stock to stock. More specifically, for Apple's stock, the ANN achieved the best score: the mean square error was 0.14 with 10 neurons. On the other hand, for Oracle, the best model had an MSE of 0.22 with 9 neurons. Google's model consisted of 12 neurons with an MSE of 0.27, Microsoft's model used 10 neurons and achieved an MSE of 0.18, and lastly, Facebook's model used the largest number of neurons, 15, to obtain an MSE of 0.28.

In [11], the author combined economic variables and semantic features extracted from Market-Watch. The economic variables included the ratios of liquidity, the beta, the price per earnings ratio, the price to book and the ROE. The semantic features extracted from news stories were a certainty, optimism, realism, activity, and commonality as described by Hart el. al [7]. To test the hypothesis, the author compared 1.402 stocks from the New York Stock Exchange or the Nasdaq Stock Market with a reported stock price of a least 3$ before the 10−k fillings. The best prediction result was achieved by an artificial neural network with one hidden layer that used all of the data. The accuracy of the model was 0.6184. Moreover, the author exploited different time windows in predictions. The findings indicated that the most suitable time window is 3 days as the prediction accuracy lessens when a larger time window is considered. Moreover, authors in [43] integrated economic variables and news stories from Thomson Reuters for 6 years and for 15 stocks listed in the New York Stock Exchange. The features extracted from news stories were the sentiment and the number of topics. To acquire the sentiment the authors used a lexicon approach, whilst to extract the number of topics they used the latent Dirichlet allocation method. Lastly, they employed an SVM and a deep neural network with dropout regularization and rectified linear units, which they trained and tested with sentiment features only, once with topics features only and once with both topics and sentiment features. The best model, which outperformed all the others, was the deep neural network that was trained with both several topics and the sentiment as features. The model's average accuracy was 57.2%. Moreover, to test more thoroughly their results, the authors compared two trading strategies. The first one was a simple buy and hold strategy, whereas the other one was based on the DNN's results. On average, the DNN's strategy achieved a return on 5.76%, whilst the B & H strategy 1.24%.

Authors in [44] tested different ANNs with different training methods: Bayesian regulariza-

[7] Hart RP (2001) Redeveloping DICTION: theoretical considerations (new). In: West MD (ed) Theory, method, and practice in computer content analysis. CT Ablex, Westport, pp 43–60

tion backpropagation, Levenberg Marquardt backpropagation, Conjugate gradient backpropagation with Powell-Beale restarts, and Gradient descent with adaptive learning rate backpropagation. Their data set consisted of 10 years in total for Dow Jones Industrial Average and of news articles from The New York Times. According to the results, the best backpropagation method was gradient descent. Afterward, the authors further exploited the optimization of this method. First, they experimented with different window sizes and their results indicated that the best window size was 6 days. This size has yielded the least mean squared error of $3.72E-05$. The second-best was a 3 days window which achieved a $4.88E-05$ MSE. Afterward, they employed a sentiment effect parameter called α. This variable was used to control the effect of the sentiment in the model. The model with lowest MSE ($3.57E-05$) was obtained when α was set to 0.01%, while the second-best ($3.85E-05$) result was obtained when it was set to 0.05%. Fine-tuning the artificial neural networks can be an important factor in predictions. For example, in [45], authors tested different options in hyperparameters. After doing multiple tests, the authors concluded that the optimal setting for the number of hidden neurons was 3, that the learning rate should be set to 0.6, and that the best option for the momentum parameter was 0.7. After the hyperparameter optimization, the authors used 70% of the sample for training and 15% for validation. The last 15% was left to be the testing set. The success rate that the model achieved was 99.95%, which is phenomenal.

Lastly, many major financial companies, such as Bloomberg and Thomson Reuters, are now providing sentiment metrics. The authors in [46] used 6 different metrics from the Bloomberg database to test the importance of sentiment in 18 stocks. The variables included several news articles for each company, the number of positive articles and the number of negative articles. Moreover, they acquired the same variables for tweets. They tested two artificial neural network models, one that included the sentiment variables and one without them. In terms of root mean square error when forecasting the price of the stock, the model which included the sentiment variables outperformed the basic model. On the other hand, in terms of direction prediction, the models obtained similar results.

2.3 Graph Theory

The noisy nature of Twitter data has been noted by a lot of authors [7, 8, 9, 10]. Authors in [10] conducted two experiments to test this nature. The authors gathered a significantly big corpus of financial data for a 5-year period for the stocks that were traded in DJIA. As a first step, they classified tweets given the financial news corpus. Then, they used the SVM and the sparse logistic regression with 1-gram and 2-grams. Their initial results showed that the 1-gram leads to higher accuracy, with the best overall accuracy being achieved by the logistic regression classifier. Afterward, they proceeded to the predictions, at first, with all of the tweets and then, with only breaking-news tweets. Their results showed that the model which accounted only for the important tweets achieved better accuracy on most of the stocks' movement direction. Moreover, authors in [34] recognizing that Twitter data exhibit noisy behavior employed TF-IDF in the preprocessing phase. But TF-IDF is not an efficient way to handle massive data.

A more efficient way to handle these data is to identify the importance of its author using

graphs. Graphs have been utilized to map importance in very efficient ways. The most famous of all is PageRank [47]. PageRank was created by Sergey Brin and Lawrence Page [48] and is the basis that Google was built on. What PageRank does is map every web page in the world wide web as a node. Each node gets a ranking according to the edges that lead up to it. What this algorithm achieves is that when a web page is referenced by other non-important web pages, its score is lower and thus, it will not be in the top suggested pages. A similar mapping can be used to model Twitter users.

Authors in [49] constructed a graph based on Twitter data. The authors modeled tweets, users, URLs, and hashtags as nodes. The edges were the annotations, retweets, mentions, citations and the author. From this graph, they extracted numerous features: the number of nodes, the number of edges, the number of connected components, the maximum diameter of any component, the PageRank, the statistics of each connected component, and the degree of each node. When the authors correlated the graph features with the price of specific stocks they found that the number of components and the number of nodes is more important than the PageRank. On the other hand, when they constructed an index with 20 companies and employed the same tests, they found that the PageRank and the degree of each node are better and more robust estimators. This implies that the PageRank and the degree act more as global estimators.

Since the microblogging activity has been increased enormously, many new algorithms have been implemented. For example, in [50], the authors provided an improved LeaderRank algorithm. More specifically, the authors used both Wikivote and Twitter networks to identify influential users. To test the results of their algorithm, they also extracted the LeaderRank and the PageRank from each graph. Their algorithm identified users who affect more nodes than the other two algorithms. Another known algorithm is the HITS (Hypertext-Induced Topic Search). The authors designed an HITS algorithm that is based on the topic-decision method and afterward, they employed an LDA model that identified the critical events and the influential spreaders. As the authors noted, their approach largely reduced the impact of unrelated posts, which in turn increased the efficiency and accuracy of identifying critical events.

In [51], authors utilized graphs to test the sentiment significance with times series. The authors modeled only 13 very influential users, such as Barack Obama. Afterward, the model took into account all of the users that repled, retweeted or mentioned any of these 13 users. The final graph consisted of $499,756$ nodes. Lastly, using the users who interacted with Barack Obama and the sentiment from Barack Obama's tweets, they tested if those correlate with the Job Approval rating. They found that the sentiment of those tweets can be used to landscape the offline phenomena. Lastly, another approach was implemented in [52]. The authors constructed a bipartite graph, the one part being the users and the second one the tweets. By extracting features from the graph and computing the stationary distribution of Markov Chain, they ranked both the influential users and the important tweets.

Although graphs have not been used extensively to model stock market prediction, the literature suggests that modeling the Twitter opinion space as a graph and extracting features, such as PageRank, can provide a solution to noisy data and also, act as estimators.

3 Data & Methodology

In this chapter, we present the data we chose and the methodology we followed. In the first part, we explore the data in multiple ways. Firstly, we provide an overview of the economic variables we chose and the reasons behind these choices. Afterward, we proceed with Twitter data. In this subsection, we present the data we gathered from Twitter and the tools we used. Finally, we close the data part of the chapter with the descriptive statistics of our final data sets. In the second part, we elaborate on the methodology part. This part includes a summary of the main modules we created, such as how we identified the most influential users, and in each part, we describe in detail the algorithms and the packages we used.

3.1 Data

3.1.1 Economic Variables

As we discussed in the previous chapter, economic variables can act as predictors. There is a humongous number of such variables ranging from fundamental analysis of a company's balance sheet to technical indicators specially designed to capture specific events. In this dissertation, we chose to use technical indicators for multiple reasons. Firstly, technical analysis is based on examining a stock's trend, thus it constitutes a more robust tool for prediction. Moreover, one of the core principles of technical analysis is that a stock's price reflects all the available information, thus it is focused more on past behavior of the market. Although technical analysis has been dismissed by academics [53], many of the leading trading companies use technical indicators to identify signals and trends on time [54].

For the reasons discussed above, we concluded that technical indicators are more suited for our research. Since technical indicators do not focus on news events, our final data set will be more balanced and have features that try to capture different aspects of trading. From all the available technical indicators, we opted for 5 of the most common ones. Those are:

1. Aroon: The Aroon Oscillator is a trend indicator that measures the power of an ongoing trend and the probability to proceed by using elements of the Aroon Indicator (Aroon Up and Aroon Down). Readings above zero show an upward trend, while readings below zero show a downward trend. To signal prospective trend changes, traders watch for zero line crossovers [55].

2. CCI: The CCI was created to determine the rates of over-bought and over-sold stocks. This is done by evaluating the price-to-moving average (MA) relationship or, more specifically, by evaluating ordinary deviations from that median [56].

3. OBV: On-balance volume (OBV), is a momentum indicator that measures positive and negative volume flows [57].

4. RSI: The RSI is a momentum index, measuring the magnitude of the latest price modifications, that is used to assess which stocks are over-bought or over-sold. The RSI is an oscillator. Traditionally, traders interpret a score of 70 or higher as a sign that a stock is

overbought or overestimated, which might lead to a trend reversal. An RSI of 30 or lower signals that a stock is undervalued [58].

5. STOCH: The Stochastic Oscillator attempts to predict price turning points by comparing the last closing price of a security to its price range. It takes values from 0 up to 100. A value of 70 or higher signals an overbought security [59].

These indicators were chosen for two main reasons. Firstly, they are very robust and are used extensively in the industry. Secondly, these indicators belong to a special category which is called "Oscillators". Oscillators are indicators that fluctuate within a range and are used to capture short term trends. Our sample period ranges from December, 1st of 2018 to July, 31st of 2019. This is a period that is characterized by high fluctuations and small but powerful shocks (Trade War, No Deal Brexit, etc.), thus we believe that using such variables will provide more accurate results than using fundamental analysis.

To collect the economic variables, we used the API of Alpha Vantage. We obtained a free API key and constructed a module called "Stocks.py". The module contains a class with two functions. The first function is the one that downloads all the data and appends it to the appropriate comma-separated values file. Since we used the free version of the API, Alpha Vantage imposes rate limits. To automate the downloading process and to avoid having to manually restart the module every time the rate limit is encountered, we designed a second function. Alpha Vantage does not provide an API endpoint to check the status of the rate limit so we have to check the contents of the response to find out if the rate limit has been encountered. Thus, if the response contained "Thank you for using Alpha Vantage!", we knew that the rate limit has been reached, and the module went for sleep for 5 minutes. After these 5 minutes, which is the time limit that is being imposed by the service, the module restarts itself.

```
def _access_api_endpoint(mode, resource_url, ticker, api_key):
    while True:
        response = requests.get(url)
        content = response.content.decode()
        if "Thank you for using Alpha Vantage!" in content:
            sleep_ends_at = datetime.now() + timedelta(minutes=5)
            print("The rate limit has been encountered, " +
                  "sleeping until " +
                  sleep_ends_at.strftime("%Y/%m/%d %H:%M:%S"))
            time.sleep(5 * 60)
        else:
            break
```

Listing 1: Accessing The API Endpoint

3.1.2 Twitter Data

A large part of this dissertation depends on Twitter data. More specifically, we are interested in two categories of data, the tweets and the users that wrote those tweets. The main problem reported in the literature is the noisy nature of Twitter data [60, 61, 62, 63, 64]. To overcome this

problem, we used the "cashtag" or "$" in the tweets, which as [38, 65] notes, is more suited for gathering stock related data.

The module for gathering Twitter data is built upon a library called Twint. This library can provide tweets, users' statistics (followers, following, likes, etc.) and also, it can gather users' followers. Moreover, it also has a built-in function for storing those data directly to a database. Our Twitter module has one class, called Twitter, which in turn has three functions. The first function gathers all the relevant tweets and has three inputs, the ticker symbol, a starting date, and an ending date. Afterward, from the downloaded tweets, we take all the tweets authors' usernames and gather metrics for them. These metrics are used when we are checking the validity of our data. The last function of the module is the gathering of all users' followers, a metric that is going to be used in the graph module.

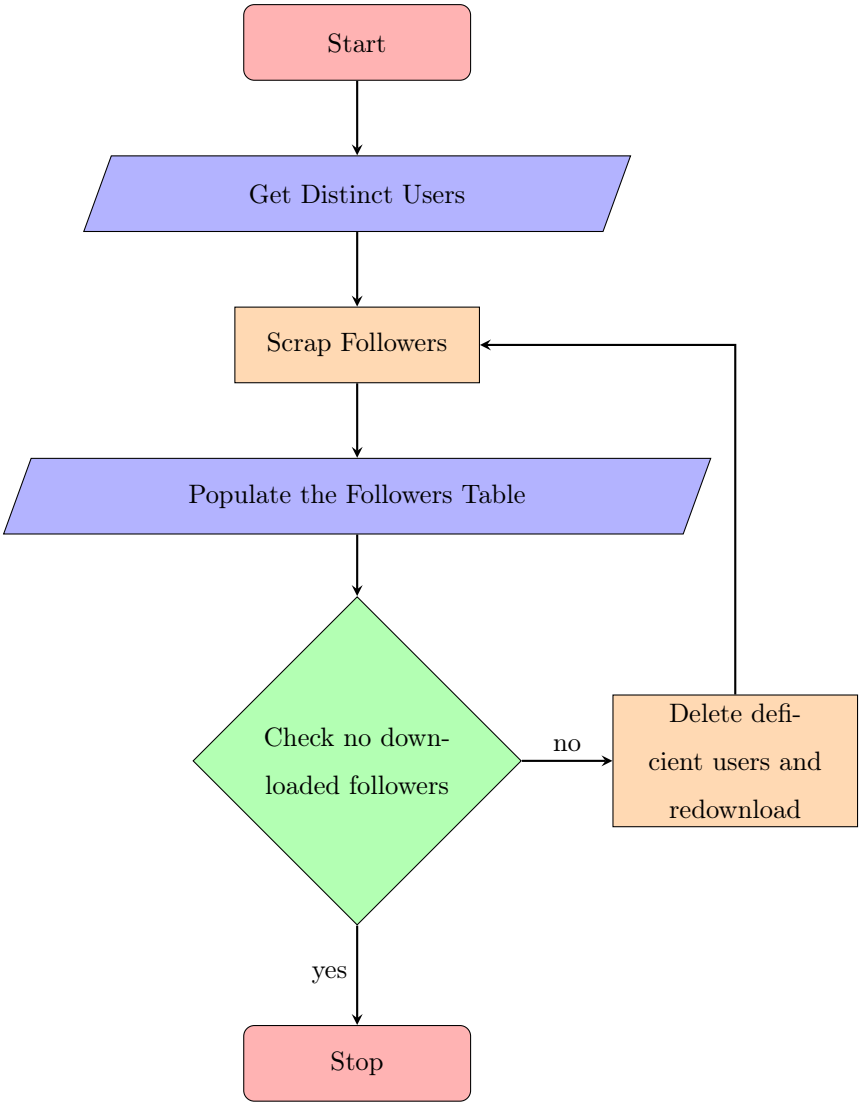

Figure 3: Flowchart of Users Download

After the first trials, we detected that Twint did not download all users' data correctly. The reason is that because Twint does not access the proprietary API that Twitter develops and supports, the gathering of all of the user data takes a few days to complete, and during that period, the total number of followers of some users was not equal to the number that was expected based on the initial step of the algorithm. To overcome this limitation, we built a function that detects

the users whose total number of the followers that have been download has a difference greater than 100 than the total number of expected (a number that is calculated with the user metrics function). That number was chosen to skip the accounts that have their followers number modified (e.g. due to follows or unfollows or due to a suspension) since the time we first gathered their personal information.

```
def find_follower_deficient_users(self, min_deficit):
    """Returns a list of usernames for which we have gathered less followers
    than expected, where "less" is defined by the "min_deficit" parameter.
    """
    data = self.execute("""
        SELECT username, total_expected - total_gathered AS deficit
        FROM
            (SELECT followers_names.follower as username,
                    count(followers_names.user) AS total_gathered,
                    users.followers AS total_expected
            FROM followers_names
            INNER JOIN users ON users.username = follower
            GROUP BY follower)
        WHERE deficit >= %s
        ORDER BY deficit DESC
    """ % min_deficit)
    # Convert the list of tuples to a simple list
    users = []
    for row in data:
        users.append(row[0])
    return users
```

Listing 2: Finding Deficient Users

After identifying the deficient users, we removed their data from the database with the below function.

```
def remove_follower_data(self, usernames):
    print("Purging the data of " + str(len(usernames)) + " users")
    for username in tqdm(usernames):
        self.execute("""
            DELETE FROM followers_names
            WHERE follower = "%s"
        """ % username)
    self.commit()
```

Listing 3: Deleting Deficient Users

Then we ran the whole process again until no deficient user can be identified.

```
def get_followers(db):
# Get the usernames for whom we a tweet that has been retweeted at least
# once.
users = db.search_users_in_tweets(min_retweets=1)
Twitter.get_user_metrics(users)
# Get the followers of the users that have at least 100 and at most 3000
# follower count.
users = db.search_users_in_users(min_followers=100, max_followers=3000)
```

```
Twitter.get_followers(users)
while True:
    deficient_users = db.find_follower_deficient_users(min_deficit=100)
    count = len(deficient_users)
    if (count > 0):
        print("We have identified " + str(count) + " users that need to " +
              "be processed again")
        db.remove_follower_data(usernames=deficient_users)
        db.remove_user_data(usernames=deficient_users)
        Twitter.get_user_metrics(deficient_users)
        Twitter.get_followers(deficient_users)
    else:
        print("No follower deficient users have been identified, the " +
              "process has completed")
        break
```

Listing 4: Gathering Users' Data Function

3.2 Methodology

The methodology we followed can be broken up into three main parts. At first, we designed a Graph for the users to obtain their importance using the PageRank algorithm [47]. Afterward, we analyzed the tweets that were obtained using two different lexicons and lastly, we estimated five different machine learning models. This section describes the above parts in more detail.

3.2.1 Identifying Influential Users

As we stated in the previous chapter, Graphs have not been used in the literature extensively, although most of the literature recognizes the problem with the noisy Twitter data. To solve this problem, we used Graphs. Because of the complexity of the project, we designed a class called Graph. This class contains three functions: the function that generates the graph, the function that computes the PageRank score for each edge, and the function that computes the hub and authority scores. The Graph class is fairly simple and is based on the NetworkX library [66]. Moreover, the PageRank and HITS algorithms are also implemented in the NetworkX library [66].

As we stated in the literature, PageRank and HITS are two algorithms that are often used to measure the importance of nodes on directed graphs. Both of the algorithms were designed to rank websites. The PageRank algorithm is a recursive algorithm, where an internet page is important if and only if important pages are linked with it. As it is usually described [67], a website's score is the probability that any random person who is browsing on the web will end up on this website. This is by definition a Markov Process. Markov Processes have been used extensively to model recursive phenomena, such as the weather. The PageRank algorithm starts with a set of websites (denoting the number of those websites with N). On each website, we assign a score of $1/N$. Afterward, we sequentially update the score of each website by adding up the weight of every other website that links to it divided by the number of links emanating from the referring website. But if the website does not reference any other website, we distribute its score to the remaining websites. This process is executed until the scores are stable.

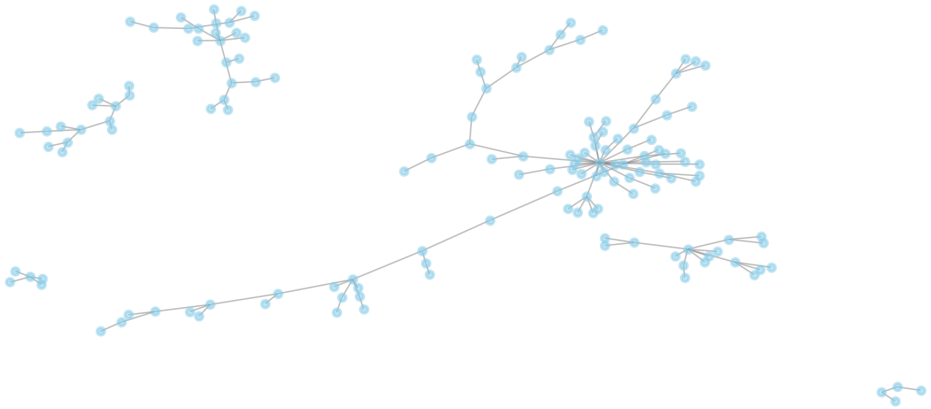

Figure 4: Graph of users' connections

The HITS algorithm was developed by prof. Kleinberg around the same time PageRank was developed [68]. HITS stands for Hypertext-Induced Topic Search and provides two scores, the "Authority" and the "Hub". To explain better the reasoning behind this algorithm, let us think of a website that contains only a phrase, for example, "automobile makers". If we evaluate a query about cars based only on its similarity to the web site's information, then the website we previously made up should always be first in any web search engine, but this site does not contain any information. Thus, what we need to identify are "good" or "authoritative" sites. If we imagine the web as a graph where websites link to each other, then we can count how many incoming and outcoming nodes each one has. Then we can compute these two scores. The hub score is calculated based on some of the authority scores of the sites that the website references. The authority scores are calculated based on the sum of the hub scores of the sites that reference the website. HITS algorithm is also recursive, and self-referential, which makes it difficult to compute. In our dissertation, we tried to compute the HITS algorithm, but all our efforts were in vain as the algorithm never achieved convergence. Since we wanted to compute the hubs and the authorities for each day in our sample, the recursiveness of the algorithm poses a significant barrier.

On the computing part, for each date, we needed to create a graph that references the follower relationships of the users that have tweeted on that specific date, which is from $2018-12-01$ to $2019-07-31$. This means that we created 242 graphs. We transformed the data into a pandas DataFrame objects using the below function.

```
def run_link_analysis(db):
    Functions.log("Creating the PageRank table in the database")
    db.create_pagerank_table()
    Functions.log("Identifying the users that have tweeted")
    df = db.get_users_with_tweets_per_day()
    dates = df["date"].unique()
    # For each date we need to create a graph that references the follower
    # relationships of the users that have tweeted on that specific date
    Functions.log("Starting the processing of " + str(len(dates)) + " days")
    for date in dates:
        start_time = time.time()
        current_users = df[df["date"] == date]["screen_name"]
```

```
        followers_df = pd.DataFrame(columns=["user", "follower"])
        Functions.log(
            "Preparing the graph for " + date +
            ", processing a total of " + str(len(current_users)) + " users"
        )
        for user in current_users:
            followers = db.get_followers(user)
            temp_df = pd.DataFrame(followers, columns=["user", "follower"])
            followers_df = followers_df.append(temp_df, ignore_index=True)
        graph = Graph(followers_df)
        Functions.log("Running PageRank")
        results = graph.run_pagerank()
```

Listing 5: Data Transformation and Graph Computing

3.2.2 Sentiment Analysis

As noted by [69], lexicon analysis outperforms other methodologies. In our approach, we used VADER (Valence Aware Dictionary and sEntiment Reasoner) [70] and TextBlob. Both of these tools are part of the nltk library and are pretty easy to use. VADER analyzer returns four scores, the negative, the positive, the neutral, and the compound score, whilst TextBlob returns two scores, the polarity (which should be very close to the compound score) and the subjectivity. We decided to use all of these variables as features in our models, which allows us to compare those two analyzers. Furthermore, to achieve better accuracy on the scores, the tweets must be stripped from any special characters, etc. More specifically, tweets often contain Unicode characters such as the non-breaking space. These characters should be normalized so as not to negatively affect the scoring of the analyzers.

```
def clean_tweet(tweet):
    tweet = re.sub(r"http\S+", "", tweet)
    tweet = normalize('NFKD', tweet)
    return tweet
```

Listing 6: Clean Tweets Function

3.2.3 Machine Learning Models

3.2.3.1 Decision Tree

The decision tree builds regression or classification models in the form of a tree structure. This means that the model breaks the dataset into smaller subsets by asking different questions each time. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches (Decisions), each one representing values for the attribute that was tested. Leaf nodes (Terminal Nodes) represent decisions on the numerical targets. The questions and their order is determined by the model itself using Information Gain (for classification) or ID3 (for regression) [71, 72, 73]. For each question, the model must make a strategic split using a criterion. Usually, this criterion is the mean squared error (MSE), but sklearn (the Python library that is used for machine learning) [74] provides other options as well, such as mean absolute error (MAE). Decision trees have a lot of advantages. First of all, trees, in general, are not affected by missing values or

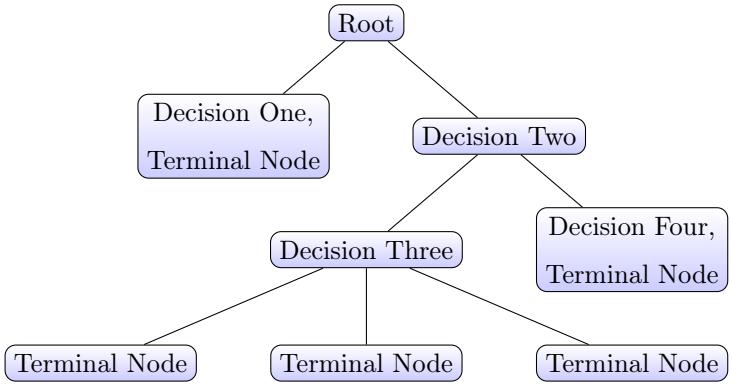

Figure 5: Simple Decision Tree

outliers. Moreover, they can handle both numerical and categorical values, and they are very easy to understand. Lastly, trees can capture non-linear relationships. Although decision trees are very useful, they present a lot of disadvantages. The most important one is that they tend to overfit to the training sample. Secondly, a small difference in data might produce a completely different tree. Lastly, there is no guarantee that the tree will be the global optimal [73].

3.2.3.2 Random Forest

Random Forest is another method that uses a tree structure to solve a regression or a classification problem. A random forest is a collection of decision trees, with each tree voting on the final decision. In the training phase, each tree on the forest considers only a random sample of the data. In the predicting phase, each tree will make a prediction and the average of all of the trees will be considered as the final value [75].

3.2.3.3 XGBoost

Boosting and bagging are two methods commonly used in weak prediction trees, such as decision trees, to improve their performance. Those two methods work sequentially, meaning that a new model is added to correct the error of the existing models until no further improvements can be made. XGBoost stands for eXtreme Gradient Boosting, which is a method where new models are created that predict the residuals or errors of existing models and then, added together to make the final prediction. Its name comes from the algorithm used to minimize the loss function, which is called gradient descent [75].

3.2.3.4 κ-Nearest Neighbors

k-Nearest Neighbors is one of the most basic and essential machine learning algorithms. Like the trees, it belongs to the supervised machine learning algorithms. κ-NN is a non-parametric method, meaning that it does not make any assumptions about the distribution of the data. κ-NN is a fairly simple model that calculates similarities based on the distances between the data points. When a new entry needs to be classified, the algorithm measures the distances between the new data and all the already classified data. The new entry is then assigned to the class that has the minimum distance to the new data point. There are multiple methods to measure the distance, such as the Euclidean or the Manhattan distance.

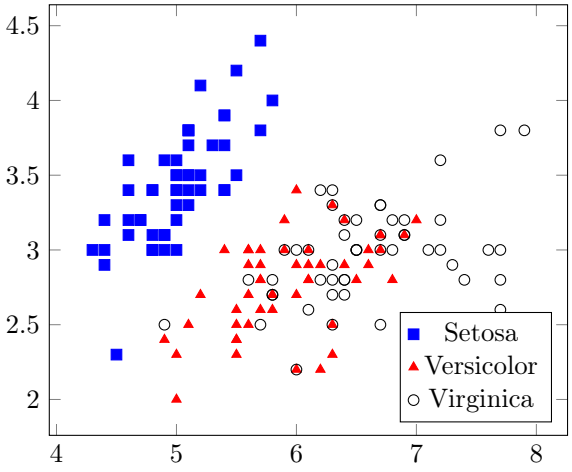

Figure 6: κ-Nearest Neighbors on Iris Dataset

3.2.3.5 LSTM

Simple neural networks cannot understand the context and the order of the data. For that, we need some sort of memory. Recurrent neural networks are a special form of neural networks where their units are connected between each other so the values depend not only on all the units [76, 77]. RNNs are extremely important and have been successfully used in a lot of applications, such as speech recognition. But as [76] showed, RNNs suffer from the vanishing gradients problem. This problem refers to the hidden neuron activation functions that are used. If those functions are saturating nonlinearities, like the tanh function, then the derivatives can be very small, even close to zero. Multiplying many such derivatives leads to zero, which means that the neural network cannot propagate back for too many instances [75].

Hochreiter & Schmidhuber [78] introduced another kind of recurrent neural networks, called long short term memory or LSTM. Those models have the same "chain" like structure, but the module responsible for the "repetition" part has a different structure. In a classic RNN, the repetition module is a neural network with a hidden layer, usually with tanh as the activation function.

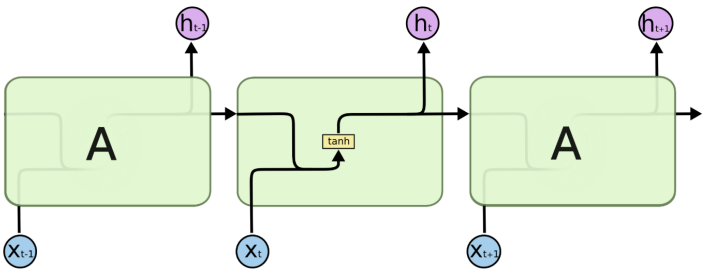

Figure 7: A Recurent Neural Network

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

On an LSTM, instead of having a single hidden layer, there are four. On the first stage or gate, as it is called, the neural network decides which information to throw away from the cell state. Continuing to the second stage, the model incorporates the new information and decides what to

keep and what to throw away. The model updates the old cell state into the new cell state. In the third stage, the model throws away the old information and adds new information. In this stage, the candidate values are estimated. Lastly, the output values depend on the state of the first and the third layer [75].

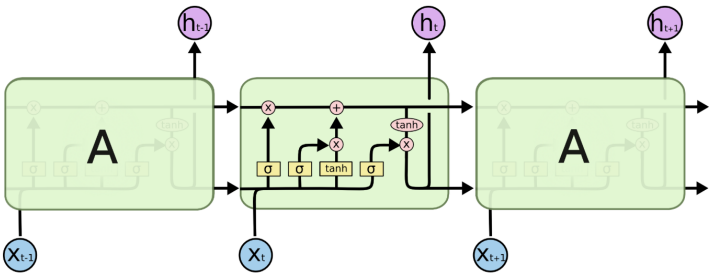

Figure 8: An Long Short Term Memory Network

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

4 Results

In this section, we present the results of our research. We begin with feature selection and explanatory data analysis. The second part consists of a summary of the results per data set and per model, presenting them through charts. Afterward, we proceed with the description of the results in more detail. We do that only for all data sets for all the models and all the tickers. All of the scores refer to the mean squared error, thus the best score is the lowest. Lastly, we evaluate our results using a naive trading strategy and compare this strategy across all data sets.

4.1 Feature Selection

This section describes the features we created in this dissertation, as well as the descriptive statistics of those features per ticker. We have to note that all of the variables are not available for the day we want to predict, thus all the features created are values of previous days. Since there is no consensus on the literature on which time lag is the most important, for every variable we created the lags from 1 to 3 days prior [2].

Moreover, because one major aspect of this dissertation is to determine if the sentiment data are noisy, and how this can be redeemed, we decided to create three different datasets. The first dataset contains only the lagged economic variables (discussed in detail in section 3.1.1) and the lags of previous days' closing prices. The second dataset contains all the features of the economic dataset as well as the sentiment data (discussed in detail in section 3.2.2). Lastly, the PageRank dataset contains all of the features from the sentiment dataset, but the sentiment variables are all multiplied by the PageRank value of the respective user.

One major drawback of calculating daily PageRank values for each user is that the algorithm does not always estimate the importance for all of the users. Thus, we decided to fill all those dates with the mean of each user. After that process, we fill all the residual not-estimated PageRank values with 0. The reasoning behind this process is that we wanted to have a timely measure of the importance of the user, and in the cases where this was not feasible, we theorized that the number of the followers of a user does not alter significantly from day to day, so it was a logical assumption to fill any missing values with their respective mean. Lastly, if there was no mean, then that means that the PageRank algorithm did not find any importance in the user for any day, thus we filled the residual empty values with 0, as we considered them noisy and not important.

4.1.1 Descriptive Statistics

Since our collection of data consists of 30 stocks and 16 features for each stock, we decided to provide the reader with two stocks' data explanatory data analysis (for more descriptive statistics please see A).

Our data are time-series, which means that the component of time plays an important role. Usually, time-series data bare some specific characteristics, such as trend and seasonality, and tend to exhibit high orders of autocorrelation. Moreover, most of the time-series are not stationary, which means that their statistical properties vary over time and that their mean and variance are volatile. On the other hand, a stationary time series (a random walk process is a stationary

process) does not change its statistical properties over time. All of the econometric, statistical and machine learning models try to predict such statistical properties, thus it would be easier if the data we are trying to predict is indeed stationary. Sadly, most of the time-series data and more specifically stock market data are non-stationary. Augmented Dickey-Fuller (the simple DF test considers only the first-order difference, whilst the ADF tests multiple time lags) is used to test whether a time-series is stationary or not. ADF's null hypothesis is that the time-series is non-stationary. We can reject the null hypothesis if the p-value is less than 0.05. In figure 8, Apple's explanatory analysis is presented.

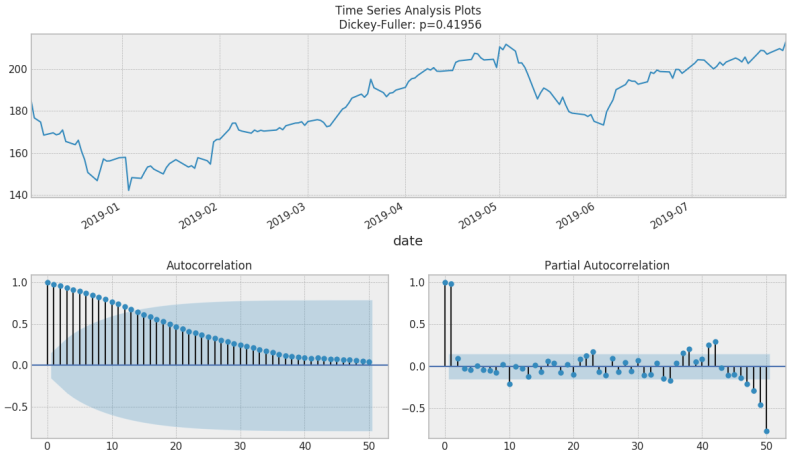

Figure 9: Apple's Explanatory Data Analysis

As it is evident from the plots, our data exhibit higher-order autocorrelation and a unit root, so our data are non-stationary. In such cases, there are a lot of approaches that can combat non-stationarity. We can take the first difference of the series, which in our case corresponds to calculating the return of the stock. Other well-documented approaches include smoothing, removing the trend, and seasonality or transformation of the data using the Box-Cox transformations.

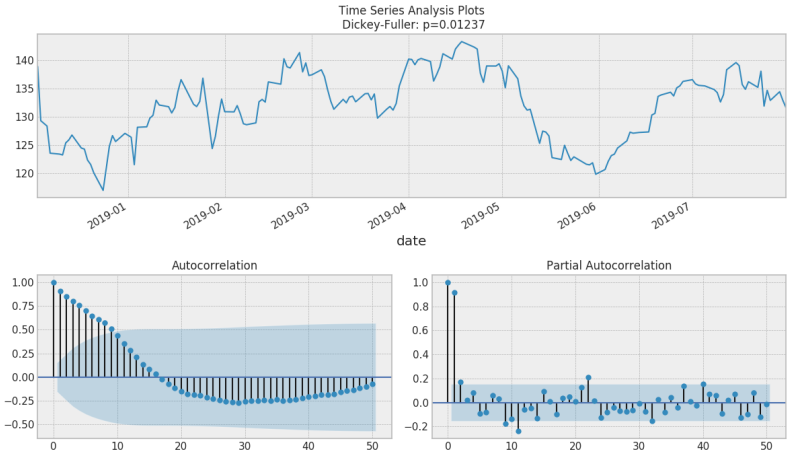

Figure 10: Caterpillar Explanatory Data Analysis

Although, as we stated, most of the time-series data are non-stationary, in our sample, we did encounter 3 stocks that are stationary. One such case is the Caterpillar stock. Although the data exhibits a high order of autocorrelation, the ADF test estimated a probability value of 0.01237,

which can be translated to the rejection of the null hypothesis of a unit root.

4.1.2 Granger Causality

Earlier studies that examined sentiment data integration with the stock market found significant evidence of causality between the stocks' closing prices and the sentiment data [2, 17, 18, 21, 22]. In-line with literature, we tested for Granger Causality, and the results showed that we cannot conclude if there is a causal relationship between the closing price and sentiment. In our sample, the test for Granger Causality showed that only in 2 out of 30 stocks there is a causal relationship.

Table 1: Apple's Close Price Granger Causality Tests

	Lag 1		Lag 2		Lag 3	
Exogenous Variable	χ^2	p-value	χ^2	p-value	χ^2	p-value
compound_score	0,99	0,32	0,88	0,64	1,17	0,76
negative_score	0,99	0,32	0,88	0,64	1,17	0,76
neutral_score	0,99	0,32	0,88	0,64	1,17	0,76
polarity	0,99	0,32	0,88	0,64	1,17	0,76
positive_score	0,99	0,32	0,88	0,64	1,17	0,76
subjectivity	0,99	0,32	0,88	0,64	1,17	0,76

The tests were performed for time lags 1, 2, and 3 days through the statsmodels functionality for Granger Causality. In Apple's case, the results show that there is no causality between the closing price and the exogenous variables. On the other hand, Intel's stock shows an order I(3) integration. The other stock that exhibits the same behavior is IBM's. The results from Boeing's stock showed that there is a weak causality for 2 days time lag, at a significance level of 10%.

Table 2: Intel's Close Price Granger Causality Tests

	Lag 1		Lag 2		Lag 3	
Exogenous Variable	χ^2	p-value	χ^2	p-value	χ^2	p-value
compound_score	12,20	0,00	16,59	0,00	15,56	0,00
negative_score	12,20	0,00	16,59	0,00	15,56	0,00
neutral_score	12,20	0,00	16,59	0,00	15,56	0,00
polarity	12,20	0,00	16,59	0,00	15,56	0,00
positive_score	12,20	0,00	16,59	0,00	15,56	0,00
subjectivity	12,20	0,00	16,59	0,00	15,56	0,00

4.2 Summary

4.2.1 Results Per Data Set

Figure 11 presents the average results per data set in descending order. As we can see, the data set that has the minimum error is the PageRank data set, whilst the maximum error is encountered on the sentiment data set. This result is an indication that the noisy nature of Twitter data can be redeemed if the algorithm takes into account the importance of the user. Moreover, the usage of the PageRank sentiment data did have a positive effect on the score since this data set achieved a lower score than the economic one. On the other hand, the statistical test showed that the average

Figure 11: Results Per Data Set

scores do not have a statistical difference between any data set.

Table 3: Test for Statistical Differences between Data Set Scores

Data Set 1	Data Set 2	t - Stat	t Critical one-tail	t Critical two-tail
Economic	Sentiment	-0,05	1,67	2,00
Economic	PageRank	0,05	1,67	2,00
Sentiment	PageRank	0,10	1,67	2,00

Examining the models used, the best score was achieved by the XGBoost model and the second-best by the Random Forest model. Although trees are not usually good in handling time-series data, those two models outperformed both the LSTM and the kNN regressor. We have to note that neural networks and especially LSTM need a lot of data and a lot of training, our training data set contained 175 data points, approximately six months of data, which are not considered enough. The average mean squared error of XGBoost is $11,44$. For Random Forest it is $15,62$, for Decision Tree it is $30,38$, for k-NN it is $46,80$, and lastly, for LSTM it is $107,43$.

It is important to note that the above ranking of the models persists through all the models, as it is shown in figure 14. Moreover, most of the models achieve their lowest scores on the PageRank

Figure 12: Results Per Model

data set, with the economic data set being second and the sentiment data set being third. There are two exceptions. The first of them is the Random Forest model, where the best score is achieved on the economic data set in the XGBoost model and the lowest score is achieved on the sentiment data set. For the second exception, we have to note that although this is true on average, 19 from 30 stocks record a lower score on the PageRank data set, but the other 11 has a significantly higher error.

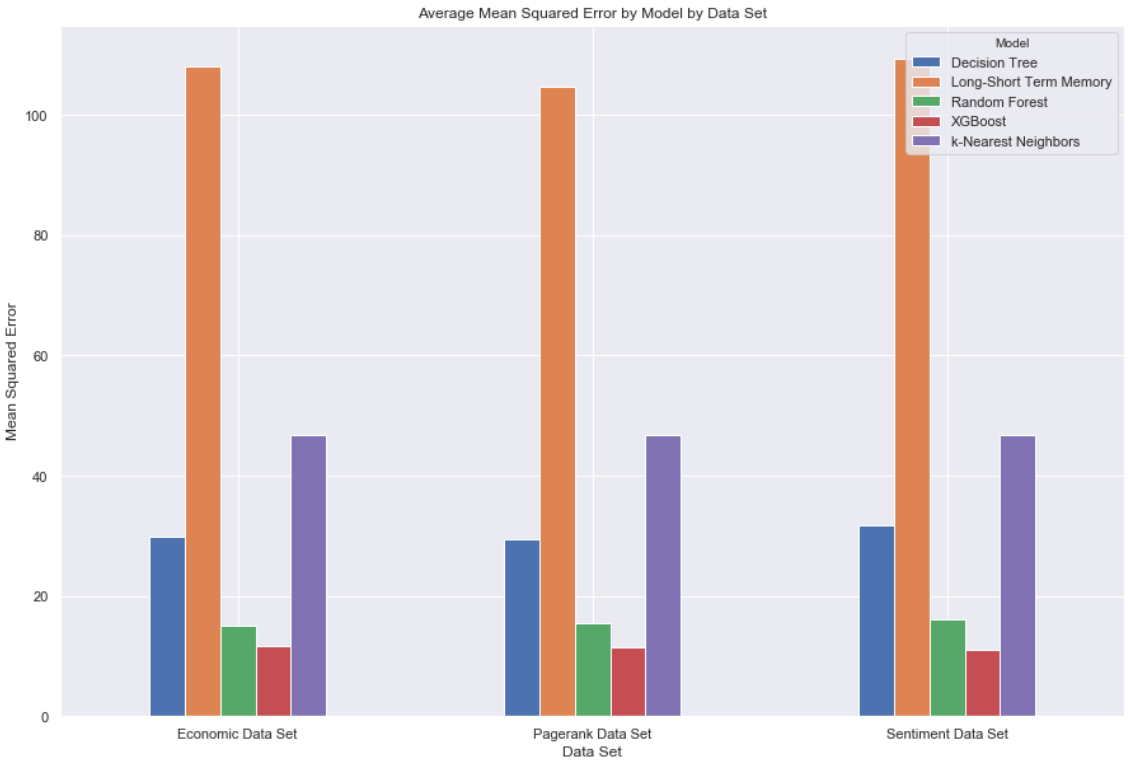

Figure 13: Results Per Model Per Data Set

4.2.2 Results Per Ticker

Figures 14 - 16 provide a graphical representation of the average, the minimum, and the maximum mean squared error per data set for each ticker. It is worth noting that the stocks with the worst results record similar results across all the data sets. The final results indicate that PageRank

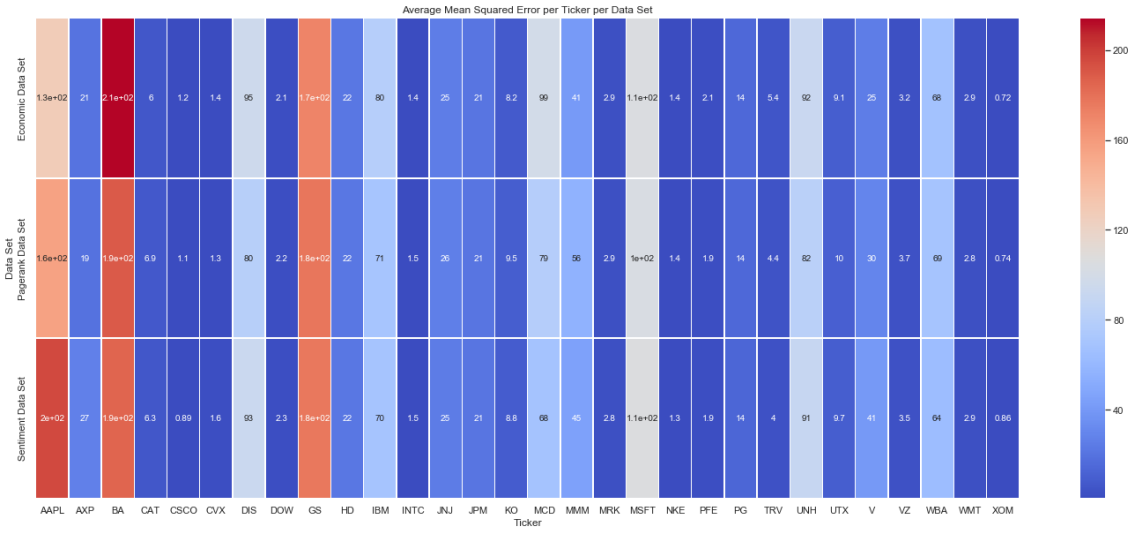

Figure 14: Average MSE Per Ticker Per Data Set

helped improve the score of all models. More specifically, 15 out of 30 stocks achieved their best score with the PageRank data set. On the other hand, in 9 stocks, the economic data set is where the lowest error was observed. Finally, the remaining 6 stocks have their lower scores on the sentiment data. Furthermore, the superiority of PageRank data set is confirmed by the average error: For the 15 stocks in the PageRank data set, the mean squared error is $2,70$, whilst on the other data sets the error is $8,37$ and $20,46$ for the economic and the sentiment data set respectively.

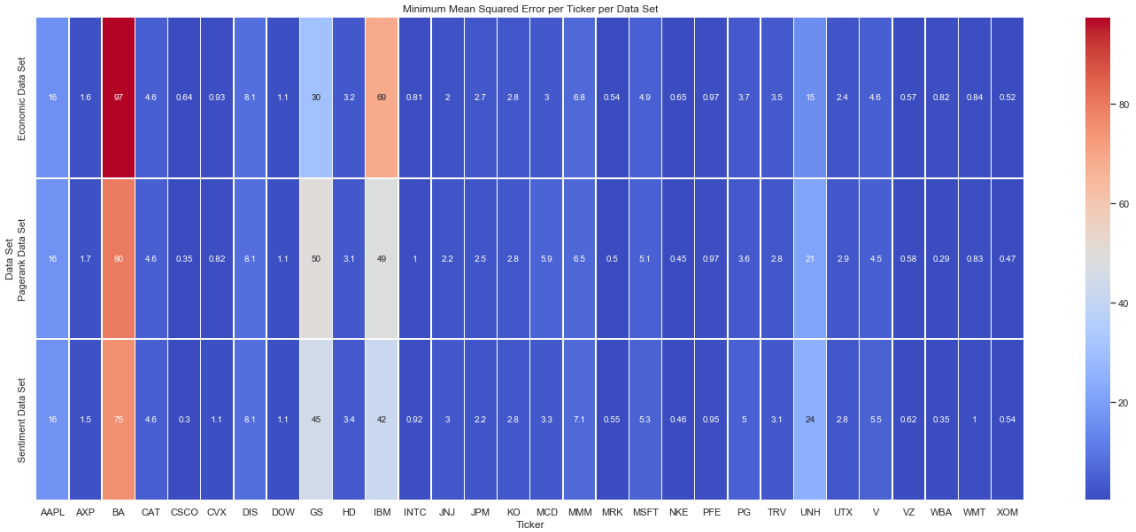

Figure 15: Minimum MSE Per Ticker Per Data Set

The worst data set is the economic one. 16 out of 30 stocks achieved their worst scores with this data set. The second worst is the PageRank data set with 9 stocks, and the final is the sentiment data set.

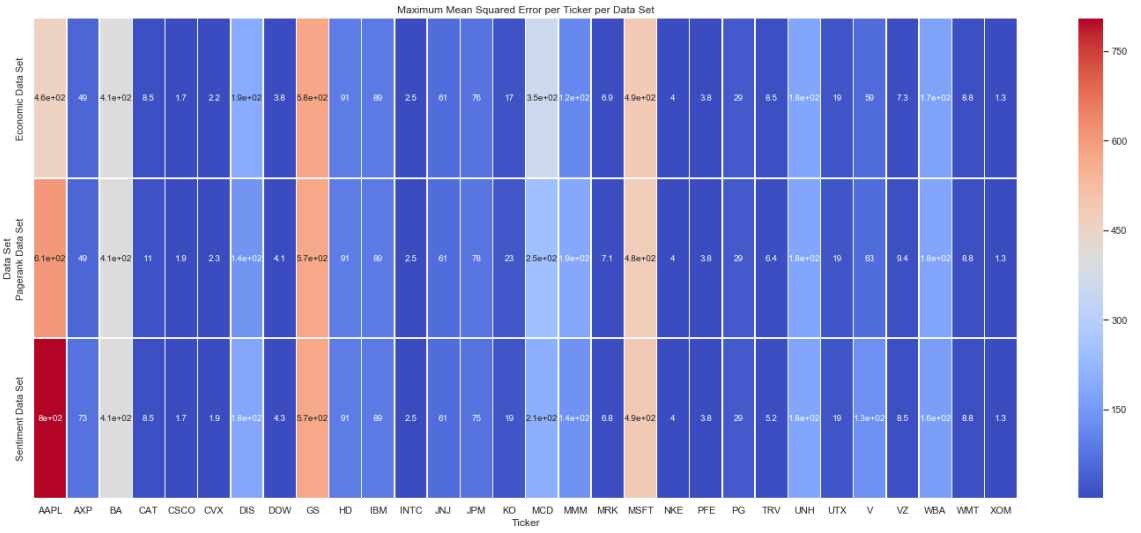

Figure 16: Maximum MSE Per Ticker Per Data Set

Figures 17 - 19 provide a graphical representation for the average, the minimum, and the maximum mean squared error per model for each ticker. As we have seen, the model with the best average mean squared error is the XGBoost for 11 out of 30 stocks. Random Forest is the second-best with 7, LSTM is third with 5, k-Nearest Neighbors is the fourth. The worst model is the Decision Tree with only 3 stocks out of 30, recording their best score with that model.

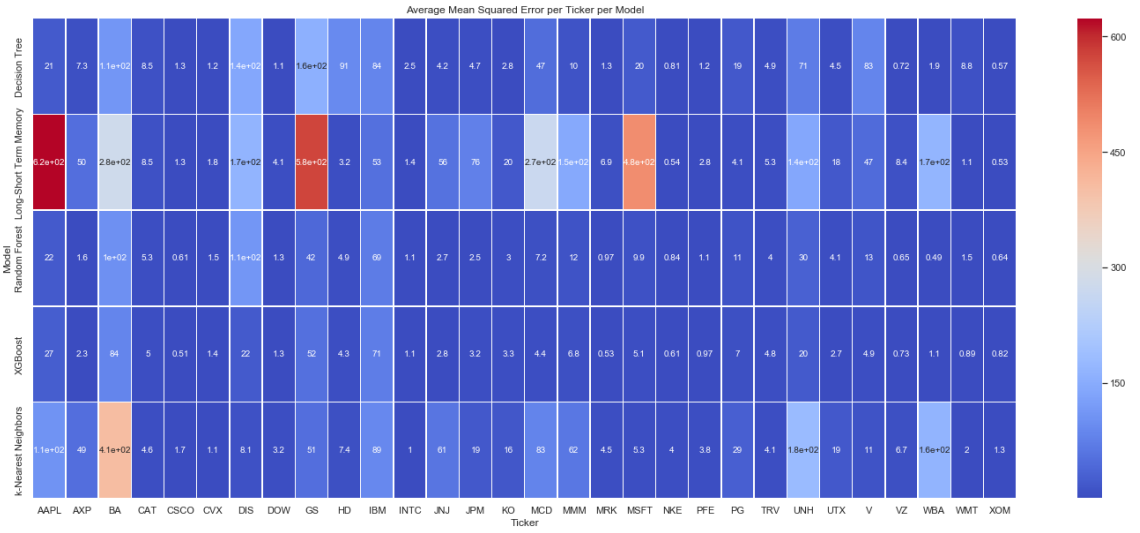

Figure 17: Average MSE Per Ticker Per Model

The results for the minimum error do not change by far. The only difference is that for the minimum mean squared error, the worst model is the k-Nearest Neighbors, for 2 out of 30 stocks.

Lastly, the worst model is the long short term memory neural network. 16 out of 30 stocks achieved their worst scores with this model. The second worse is the k-Nearest Neighbors with 9 stocks. The third is the decision tree (4 stocks). Finally. XGBoost estimated the worst error for 1 stock only.

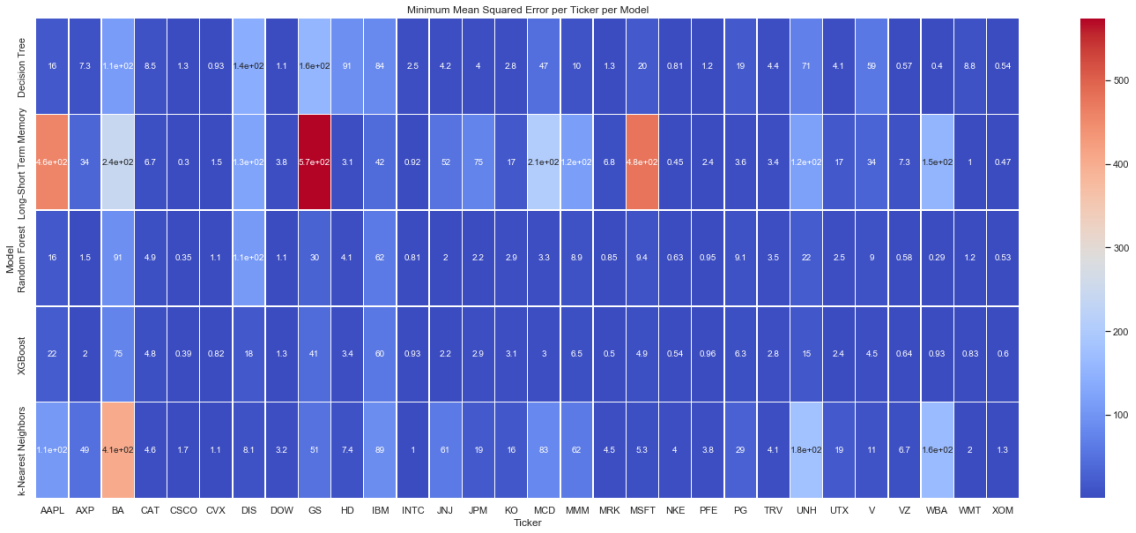

Figure 18: Minimum MSE Per Ticker Per Model

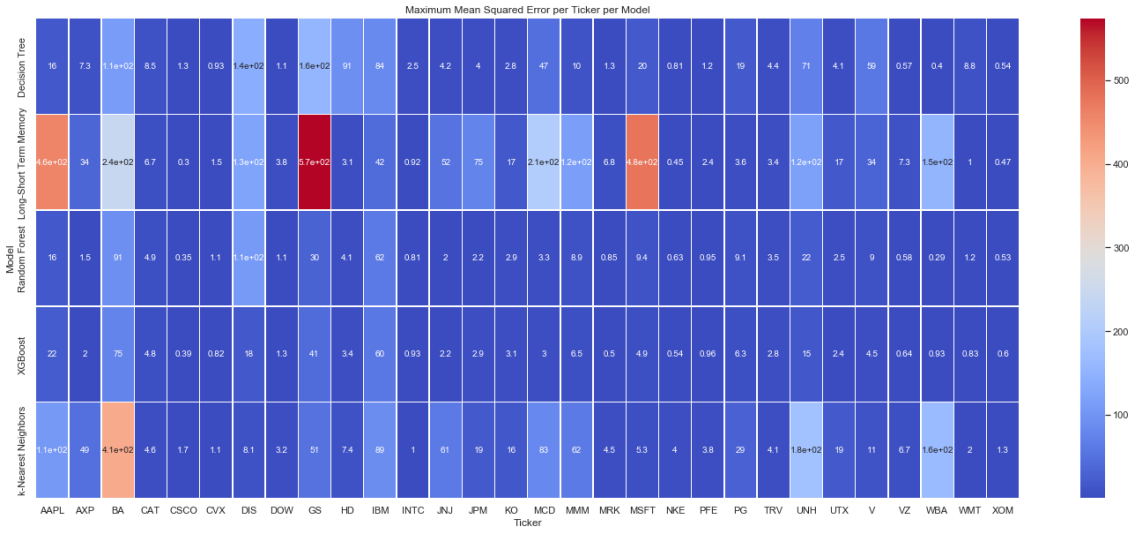

Figure 19: Maximum RMSE Per Ticker Per Model

4.3 Results Per Model

In this section, we present the results for each model on the PageRank data set. To obtain the best model, we performed a grid search for all of the models, except for the LSTM network. Thus, for each model, we present the optimum parameters obtained for each ticker, alongside the mean squared error of that model. Moreover, for each model, we provide a chart for the tickers that achieved the best and the worst score.

4.3.1 Decision Tree Results

Decision trees, in general, failed to provide good estimators. The best model achieved a $0,40$ mean squared error in WBA, whilst the worst score was recorded on Goldman Sachs, with a mean squared error of $156,50$. We have to note that despite the small errors, decision trees failed to keep up with any changes in the price, and most of the time produced only flat lines.

In the decision trees algorithms, we chose to optimize five parameters: the criterion, the minimum samples split, the maximum depth, the minimum samples in leaf, and the maximum leaf

nodes. In the criterion, we tested two options, the mean squared error and mean absolute error. Continuing with the parameter of maximum leaf nodes, we tested three possible values: 5, 20, and 100. For the minimum sample leaf, we also tested three possible values: 20, 40, and 100. The last parameter we optimized is the minimum samples split, where we chose three values: 10, 20, and 40.

4.3.1.1 Economic Data Set Results

Beginning with the economic data set the best model was in Verizon's stock with a mean squared error of $0,57$. On the other hand, the worst model was observed in Disney's stock, with an MSE of $156,50$.

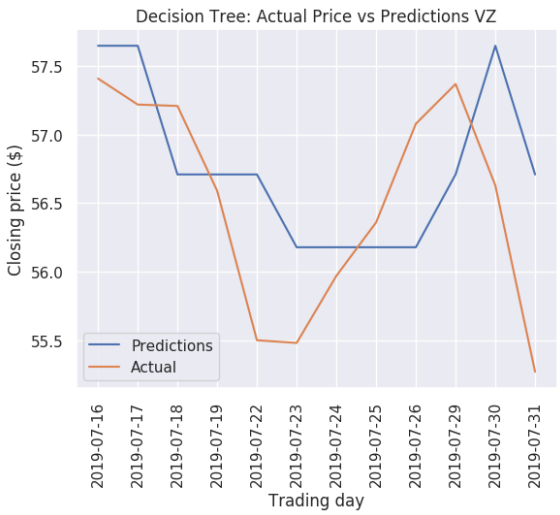

Figure 20: The Best Economic Model in Decision Tree

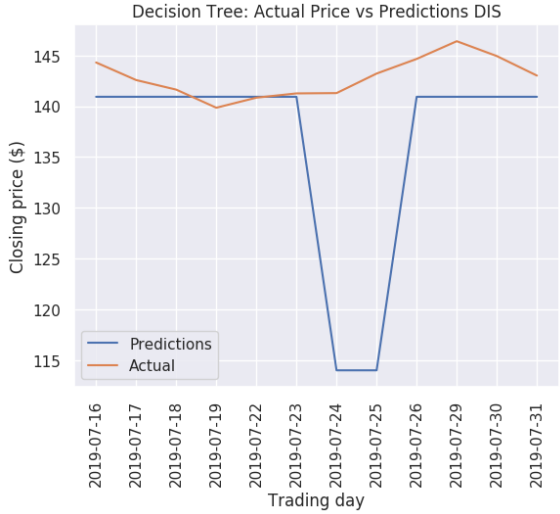

Figure 21: The Worst Economic Model in Decision Tree

On the optimization part, 19 out of 30 stocks found the best criterion to be the mean squared error. On the maximum depth parameter, in 12 stocks a maximum depth of 2 was chosen, whilst the options of 8 and 6 were the best for 9. Continuing with the parameter of maximum leaf nodes, for 14 stocks the optimum value was 5. For 4 of them, it was 20, and for 12 stocks the value of 100 was chosen. In the minimum samples leaf, for all stocks but one, the optimum value was set to 20; for Caterpillar's stock, the optimum variable was estimated to 40. Lastly, for the minimum samples split. the results showed that for 29 stocks the optimum value was 20, and for one stock, that of Caterpillar, 40.

Table 4: Economic Data Decision Tree Results Per Ticker

Ticker	Mean Squared Error	Criterion	Max Depth	Max Leaf Nodes	Min Samples Leaf	Min Sample Split
AAPL	31,28	mse	8,00	5,00	20,00	20,00
AXP	7,25	mse	6,00	100,00	20,00	20,00
BA	110,26	mae	6,00	5,00	20,00	20,00
CAT	8,50	mae	2,00	5,00	40,00	40,00
CSCO	1,33	mae	2,00	100,00	20,00	20,00
CVX	0,93	mse	8,00	100,00	20,00	20,00
DIS	139,47	mse	8,00	5,00	20,00	20,00
DOW	1,12	mse	2,00	5,00	20,00	20,00
GS	156,50	mae	6,00	5,00	20,00	20,00

Table 4: Economic Data Decision Tree Results Per Ticker

Ticker	Mean Squared Error	Criterion	Max Depth	Max Leaf Nodes	Min Samples Leaf	Min Sample Split
HD	90,65	mae	2,00	5,00	20,00	20,00
IBM	84,36	mse	2,00	100,00	20,00	20,00
INTC	2,55	mae	6,00	5,00	20,00	20,00
JNJ	4,20	mae	6,00	5,00	20,00	20,00
JPM	6,17	mae	6,00	5,00	20,00	20,00
KO	2,80	mae	8,00	5,00	20,00	20,00
MCD	47,07	mse	2,00	100,00	20,00	20,00
MMM	10,37	mse	6,00	20,00	20,00	20,00
MRK	1,33	mse	8,00	100,00	20,00	20,00
MSFT	20,13	mse	2,00	100,00	20,00	20,00
NKE	0,81	mse	8,00	100,00	20,00	20,00
PFE	1,22	mse	8,00	20,00	20,00	20,00
PG	18,86	mse	2,00	20,00	20,00	20,00
TRV	5,19	mse	2,00	20,00	20,00	20,00
UNH	70,62	mse	2,00	5,00	20,00	20,00
UTX	4,10	mse	8,00	100,00	20,00	20,00
V	59,04	mae	2,00	5,00	20,00	20,00
VZ	0,57	mse	8,00	100,00	20,00	20,00
WBA	2,64	mse	2,00	100,00	20,00	20,00
WMT	8,77	mae	6,00	5,00	20,00	20,00
XOM	0,62	mse	6,00	100,00	20,00	20,00

4.3.1.2 Sentiment Data Set Results

On the sentiment data set the best model was, also, in XOM' stock with a mean squared error of $0,54$, whilst Verizon's stock was the second-best, with MSE equal to $0,79$. On the other hand, the worst model was observed, again in Disney's stock, with an MSE of $156,50$.

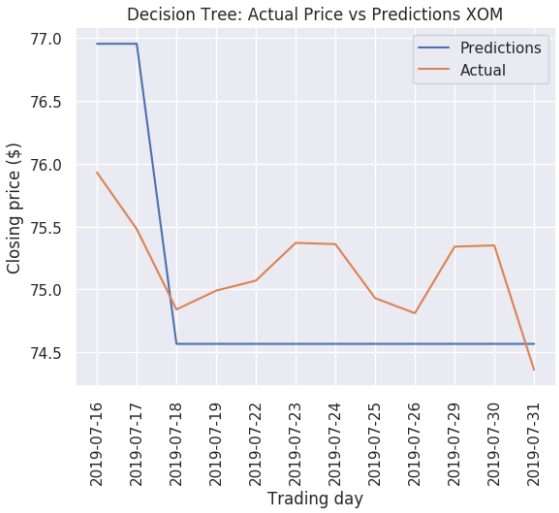

Figure 22: The Best Sentiment Model in Decision Tree

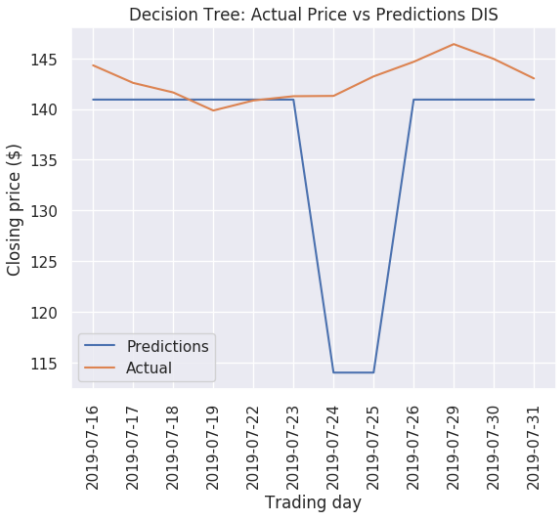

Figure 23: The Worst Sentiment Model in Decision Tree

In the table below we summarize the results of the grid search. 19 out of 30 stocks found the best criterion to be the mean squared error. On the maximum depth parameter, in 15 stocks a maximum depth of 6 was chosen, whilst the options of 8 and 2 were the best for 6 and 9 stocks, respectively. Continuing with the parameter of maximum leaf nodes, for the vast majority of stocks, 20 out of 30 stocks the optimum value was 5. For 7 of them, it was 20, and for only 3

stocks the value of 100 was chosen. In the minimum samples leaf, again the only exception is Caterpillar's stock, in which the optimum variable was estimated to 40. For all the other stocks the optimum value was set to 20. Lastly, for the minimum samples split the results do not indicate a clear preference in any option. For 12 stocks the optimal value was 10, and the options 20 and 40 were found to be optimal for 9 stocks.

Table 5: Sentiment Data Decision Tree Results Per Ticker

Ticker	Mean Squared Error	Criterion	Max Depth	Max Leaf Nodes	Min Samples Leaf	Min Sample Split
AAPL	16,32	mse	6,00	20,00	20,00	10,00
AXP	7,25	mse	2,00	20,00	20,00	20,00
BA	110,26	mae	6,00	5,00	20,00	10,00
CAT	8,50	mae	2,00	5,00	40,00	10,00
CSCO	1,33	mae	6,00	5,00	20,00	10,00
CVX	0,93	mse	8,00	100,00	20,00	20,00
DIS	139,47	mse	6,00	5,00	20,00	20,00
DOW	1,12	mse	2,00	5,00	20,00	10,00
GS	156,50	mae	6,00	5,00	20,00	10,00
HD	90,65	mae	2,00	5,00	20,00	10,00
IBM	84,36	mse	2,00	100,00	20,00	10,00
INTC	2,55	mae	6,00	5,00	20,00	10,00
JNJ	4,20	mae	6,00	20,00	20,00	10,00
JPM	3,98	mae	6,00	5,00	20,00	10,00
KO	2,80	mae	6,00	20,00	20,00	20,00
MCD	47,07	mse	6,00	100,00	20,00	40,00
MMM	10,37	mse	8,00	20,00	20,00	20,00
MRK	1,33	mse	8,00	5,00	20,00	40,00
MSFT	20,13	mse	8,00	5,00	20,00	20,00
NKE	0,81	mse	6,00	20,00	20,00	40,00
PFE	1,20	mse	8,00	20,00	20,00	40,00
PG	18,86	mse	8,00	20,00	20,00	20,00
TRV	4,39	mse	6,00	20,00	20,00	10,00
UNH	70,62	mse	2,00	5,00	20,00	10,00
UTX	5,44	mse	8,00	5,00	20,00	20,00
V	59,04	mae	2,00	5,00	20,00	10,00
VZ	0,79	mse	6,00	5,00	20,00	10,00
WBA	2,64	mse	2,00	5,00	20,00	20,00
WMT	8,77	mae	6,00	5,00	20,00	20,00
XOM	0,54	mse	6,00	5,00	20,00	10,00

4.3.1.3 Pagerank Data Set Results

Decision trees, in general, failed to provide good estimators. The best model achieved a $0,54$ mean squared error, whilst the worst score was recorded on Goldman Sachs, with a mean squared error of $156,50$. The error recorded in the best model may be small but as it is shown in the chart below, the model failed to keep up with any changes in the price and produced only a flat line.

The grid search for 19 out of 30 stocks found the best criterion to be the mean squared error. On the maximum depth parameter, in 15 stocks a maximum depth of 6 was chosen, whilst for 8 and 7 stocks a maximum depth of 2 and 8, respectively, were found to be the most appropriate. Continuing with the parameter of maximum leaf nodes, for 18 stocks the optimum value was 5. For 9 of them, it was 20, and only for 3 stocks, the value of 100 was chosen. For the minimum samples leaf, for all stocks but one, the optimum value was set to 20; for Caterpillar's stock, the optimum variable was estimated to 40. This parameter remained constant across all data sets.

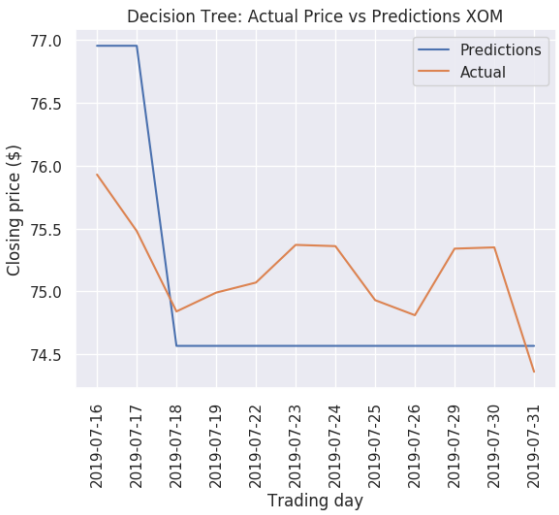

Figure 24: The Best PageRank Model in Decision Tree

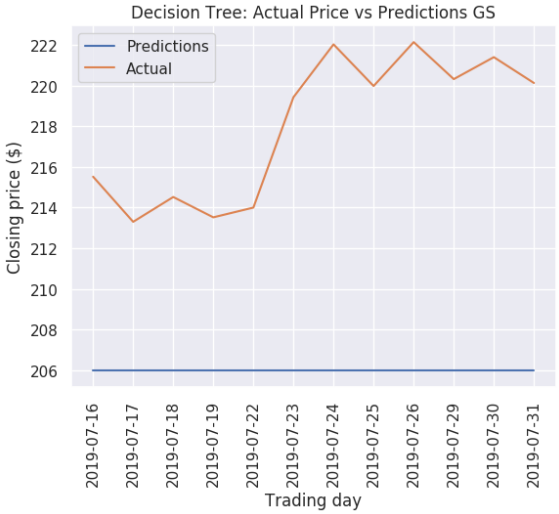

Figure 25: The Worst PageRank Model in Decision Tree

Lastly, for the minimum samples split, the results showed that for 16 stocks the optimum value was 10, for 10 stocks 20 was deemed as the optimum value and lastly, for 4 stocks 40 was found to be the optimum.

Table 6: PageRank Data Decision Tree Results Per Ticker

Ticker	Mean Squared Error	Criterion	Max Depth	Max Leaf Nodes	Min Samples Leaf	Min Sample Split
AAPL	16,32	mse	6,00	20,00	20,00	10,00
AXP	7,25	mse	2,00	20,00	20,00	20,00
BA	110,26	mae	6,00	5,00	20,00	10,00
CAT	8,50	mae	2,00	5,00	40,00	10,00
CSCO	1,33	mae	6,00	5,00	20,00	10,00
CVX	0,93	mse	8,00	100,00	20,00	20,00
DIS	139,47	mse	6,00	5,00	20,00	20,00
DOW	1,12	mse	2,00	5,00	20,00	10,00
GS	156,50	mae	6,00	5,00	20,00	10,00
HD	90,65	mae	2,00	5,00	20,00	10,00
IBM	84,36	mse	2,00	100,00	20,00	10,00
INTC	2,55	mae	6,00	5,00	20,00	10,00
JNJ	4,20	mae	6,00	20,00	20,00	10,00
JPM	3,98	mae	6,00	5,00	20,00	10,00
KO	2,80	mae	6,00	20,00	20,00	20,00
MCD	47,07	mse	6,00	100,00	20,00	40,00
MMM	10,37	mse	8,00	20,00	20,00	20,00
MRK	1,33	mse	8,00	5,00	20,00	40,00
MSFT	20,13	mse	8,00	5,00	20,00	20,00
NKE	0,81	mse	6,00	20,00	20,00	40,00
PFE	1,20	mse	8,00	20,00	20,00	40,00
PG	18,86	mse	8,00	20,00	20,00	20,00
TRV	4,39	mse	6,00	20,00	20,00	10,00
UNH	70,62	mse	2,00	5,00	20,00	10,00
UTX	5,44	mse	8,00	5,00	20,00	20,00
V	59,04	mae	2,00	5,00	20,00	10,00
VZ	0,79	mse	6,00	5,00	20,00	10,00
WBA	2,64	mse	2,00	5,00	20,00	20,00
WMT	8,77	mae	6,00	5,00	20,00	20,00
XOM	0,54	mse	6,00	5,00	20,00	10,00

4.3.2 Random Forest Results

The random forest algorithm was the second-best model, only behind the XGBoost. The best model had an estimated mean squared error of $0,29$, which is the best score estimated on any model on any data set. The score was recorded on the WBA stock in the PageRank dataset. On the other hand, the calculated error on the worst model was $119,35$, on Disney's stock in the economic data set.

For the random forest, we chose to optimize four parameters: the number of estimators, the maximum number of features, the minimum sample splits, and the use of bootstrap. Tables 7, 8 and 9 present the final parameters for each stock. On the maximum features parameter, we tested three options, the automatic, which is equal to the number of features, the "sqrt", when then number of features considered is equal to the $\sqrt{\text{number of features}}$, and the "log2", which is used then the number of features equals to the $\log_2$ (number of features). For the minimum number of samples required to split an internal node, we tested three options: 2, which is the default value, 4, and 8. Lastly, the parameter number of estimators was given three values: 10, which is the default, 20, and 30.

4.3.2.1 Economic Data Set Results

In the economic data set the random forest model was the second best model. The lowest mean squared error was estimated on XOM's stock, whilst the worst recorded on Disney's, again.

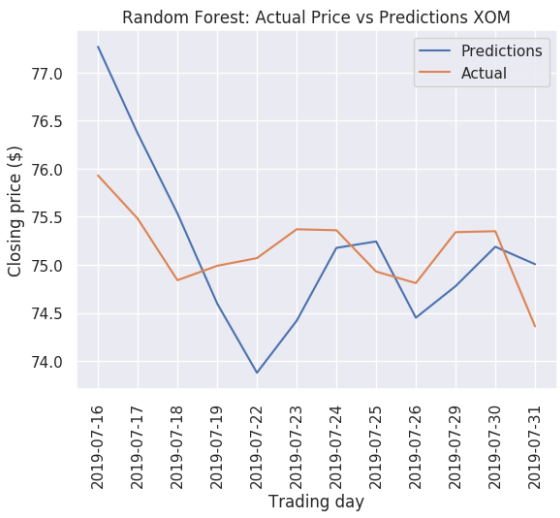

Figure 26: The Best Economic Model in Random Forest

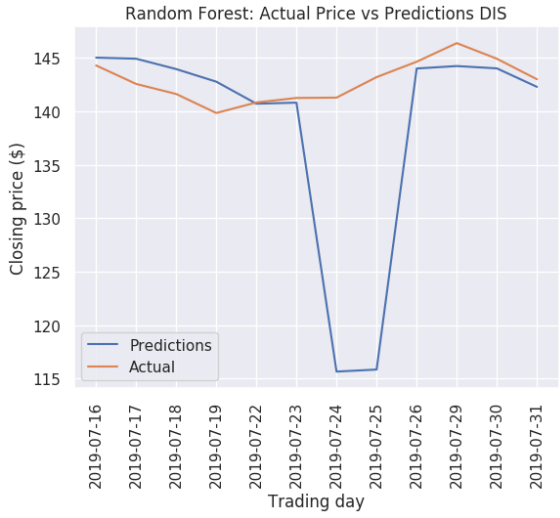

Figure 27: The Worst Economic Model in Random Forest

The results on the grid search for 24 out of 30 stocks the usage of bootstrap was deemed the most appropriate. On the maximum features parameter for 27 stocks, the optimal option was set to the automatic one, whilst for Cisco, Dow Jones and IBM the option was set to the "sqrt". Continuing with the next optimized parameter, the minimum number of samples required to split an internal node, in 18 out of 30 stocks, the optimized values were 8. For 8 stocks, it was set to 2, and for 4 of them, the default value was given. Lastly, for the number of estimators for most of the stocks (14), the optimal value was 20. For 8, it was set to the default, and for the remaining 8

tickers the option of 30 was deemed appropriate.

Table 7: Economic Random Forest Results Per Ticker

Ticker	Mean Squared Error	Bootstrap	Max Features	Min Samples Split	Estimators
AAPL	16,03	True	auto	8,00	30,00
AXP	1,60	True	auto	8,00	20,00
BA	112,75	True	auto	2,00	20,00
CAT	4,87	True	auto	8,00	10,00
CSCO	0,91	Fals	sqrt	4,00	10,00
CVX	1,06	True	auto	8,00	10,00
DIS	119,35	Fals	auto	2,00	20,00
DOW	1,13	True	sqrt	2,00	20,00
GS	30,38	True	auto	8,00	20,00
HD	4,09	True	auto	8,00	30,00
IBM	71,21	True	sqrt	8,00	10,00
INTC	0,81	True	auto	2,00	20,00
JNJ	1,95	True	auto	8,00	20,00
JPM	2,71	True	auto	2,00	20,00
KO	2,99	True	auto	8,00	30,00
MCD	5,84	True	auto	2,00	30,00
MMM	11,13	True	auto	8,00	10,00
MRK	1,19	True	auto	8,00	30,00
MSFT	10,24	Fals	auto	8,00	30,00
NKE	0,90	True	auto	4,00	10,00
PFE	1,23	True	auto	2,00	20,00
PG	10,93	Fals	auto	8,00	10,00
TRV	3,52	True	auto	8,00	20,00
UNH	22,42	True	auto	8,00	20,00
UTX	2,54	Fals	auto	2,00	20,00
V	8,98	True	auto	8,00	10,00
VZ	0,74	True	auto	8,00	20,00
WBA	0,82	True	auto	8,00	30,00
WMT	1,69	Fals	auto	4,00	20,00
XOM	0,55	True	auto	4,00	30,00

4.3.2.2 Sentiment Data Set Results

The second data set that we ran the random forest was the sentiment. For this data set the lowest mean squared error was observed in WBA's stock, whilst the worst, was for Disney's, just like the economic data set.

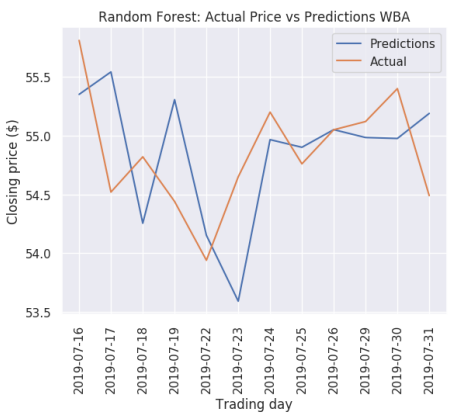

Figure 28: The Best Sentiment Model in Random Forest

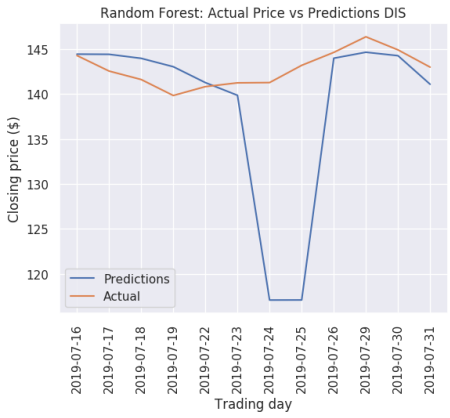

Figure 29: The Worst Sentiment Model in Random Forest

On the grid search, the optimal values we acquired do not differ a lot with the ones from the economic data set. For 25 out of 30 stocks, the usage of bootstrap was considered the most suitable. On the maximum features parameter for 28 stocks, the optimal option was set to the automatic one, whilst the other two options were used only in two stocks: for Dow Jones, the option was set to the "log2", and for IBM, it was set to the "sqrt". For the next parameter, the minimum number of samples, in 12 out of 30 stocks, the optimized values were 2. For 9 stocks, it was set to 4, and for the rest of them, the default value was given. Lastly, for the parameter number of estimators, For 12 stocks, the optimal value was the default. For 10, it was set to 20, and for 8 tickers the option of 30 was deemed appropriate.

Table 8: Sentiment Random Forest Results Per Ticker

Ticker	Mean Squared Error	Bootstrap	Max Features	Min Samples Split	Estimators
AAPL	30,52	TRUE	auto	8,00	10,00
AXP	1,53	TRUE	auto	2,00	20,00
BA	90,93	TRUE	auto	4,00	20,00
CAT	5,47	TRUE	auto	2,00	20,00
CSCO	0,58	TRUE	auto	8,00	30,00
CVX	1,87	TRUE	auto	8,00	20,00
DIS	108,16	FALSE	auto	2,00	10,00
DOW	1,55	TRUE	log2	2,00	10,00
GS	44,66	TRUE	auto	2,00	30,00
HD	5,52	TRUE	auto	4,00	20,00
IBM	74,61	FALSE	sqrt	8,00	20,00
INTC	1,43	TRUE	auto	4,00	10,00
JNJ	3,02	TRUE	auto	8,00	30,00
JPM	2,22	TRUE	auto	2,00	10,00
KO	2,99	TRUE	auto	2,00	30,00
MCD	3,27	FALSE	auto	2,00	10,00
MMM	8,90	TRUE	auto	4,00	10,00
MRK	0,86	TRUE	auto	2,00	30,00
MSFT	9,36	TRUE	auto	8,00	10,00
NKE	0,63	TRUE	auto	8,00	10,00
PFE	0,95	TRUE	auto	8,00	30,00
PG	9,08	TRUE	auto	4,00	20,00
TRV	4,28	TRUE	auto	4,00	10,00
UNH	44,89	TRUE	auto	8,00	10,00
UTX	4,48	FALSE	auto	2,00	30,00
V	20,65	TRUE	auto	4,00	20,00
VZ	0,62	TRUE	auto	4,00	30,00
WBA	0,35	TRUE	auto	2,00	10,00
WMT	1,45	FALSE	auto	4,00	20,00
XOM	0,85	TRUE	auto	2,00	20,00

4.3.2.3 Pagerank Data Set Results

The random forest algorithm was the second-best model, only behind the XGBoost. The best model had an estimated mean squared error of $0,29$, which is the best score estimated on any model on any data set. On the other hand, the calculated error on the worst model was $108,68$ on Disney's stock.

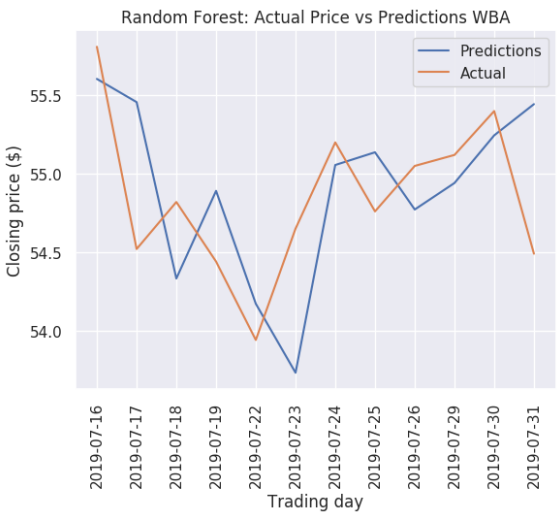

Figure 30: The Best PageRank Model in Random Forest

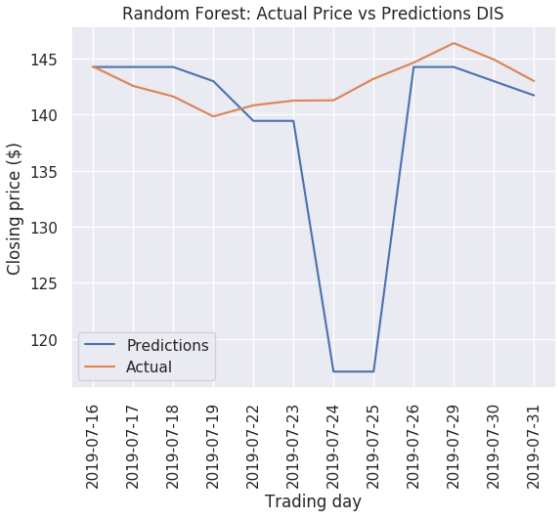

Figure 31: The Worst PageRank Model in Random Forest

As we can see, for 24 out of 30 stocks the usage of bootstrap was deemed the most appropriate. On the maximum features parameter for 28 stocks, the optimal option was set to the automatic one, whilst the other two options were used only in two stocks: for Dow Jones, the option was set to the "sqrt", and for IBM, it was set to the "log2". For the minimum number of samples required to split an internal node, in 17 out of 30 stocks, the optimized values were 8. For 8 stocks, it was set to 4, and for 5 of them, the default value was given. Lastly, for the parameter number of estimators, for most of the stocks (14), the optimal value was the default. For 11, it was set to 20, and only for 5 tickers, the option of 30 was deemed appropriate.

Table 9: PageRank Random Forest Results Per Ticker

Ticker	Mean Squared Error	Bootstrap	Max Features	Min Samples Split	Estimators
AAPL	18,38	TRUE	auto	8,00	10,00
AXP	1,72	TRUE	auto	4,00	30,00
BA	102,37	TRUE	auto	8,00	20,00
CAT	5,45	TRUE	auto	2,00	30,00
CSCO	0,35	TRUE	auto	8,00	20,00
CVX	1,42	TRUE	auto	8,00	10,00
DIS	108,68	FALSE	auto	8,00	10,00
DOW	1,24	TRUE	sqrt	8,00	10,00
GS	49,82	TRUE	auto	8,00	20,00
HD	5,05	TRUE	auto	4,00	20,00
IBM	61,68	TRUE	log2	8,00	10,00
INTC	1,06	TRUE	auto	4,00	30,00
JNJ	3,11	TRUE	auto	8,00	30,00
JPM	2,52	TRUE	auto	8,00	20,00
KO	2,88	TRUE	auto	4,00	20,00
MCD	12,56	TRUE	auto	2,00	10,00

Table 9: PageRank Random Forest Results Per Ticker

Ticker	Mean Squared Error	Bootstrap	Max Features	Min Samples Split	Estimators
MMM	15,66	TRUE	auto	8,00	10,00
MRK	0,85	TRUE	auto	8,00	10,00
MSFT	10,24	FALSE	auto	8,00	20,00
NKE	1,00	FALSE	auto	8,00	10,00
PFE	1,16	TRUE	auto	4,00	20,00
PG	13,03	FALSE	auto	8,00	30,00
TRV	4,33	TRUE	auto	8,00	20,00
UNH	22,37	TRUE	auto	8,00	20,00
UTX	5,15	FALSE	auto	2,00	10,00
V	10,76	TRUE	auto	2,00	10,00
VZ	0,58	TRUE	auto	4,00	10,00
WBA	0,29	TRUE	auto	2,00	10,00
WMT	1,21	FALSE	auto	4,00	10,00
XOM	0,53	TRUE	auto	4,00	20,00

4.3.3 XGBoost Result

As we noted previously, XGBoost is the model which estimated error was the lowest on average. The best model of XGBoost was recorded in the CSCO stock and it was equal to $0,39$. On the other hand, in BA's stock, a mean squared error of $79,80$ was calculated. As it is evident, XGBoost did not provide the lowest score on all variables, but the errors from XGBoost have a lower average and even lower standard deviation, which means that the model is a more robust estimator in general.

In XGBoost, we chose to optimize five parameters: the colsample bytree, the gamma, the maximum depth, the minimum child weight, and the subsample. Beginning from the colsample bytree, this parameter sets the subsample ratio of the columns when constructing each tree. Here, we chose to assign values ranging from $0,1$ to $1,00$. Continuing to the gamma parameter, which is the parameter that controls the minimum loss reduction required to make a further partition on a leaf node of the tree, we tried three values: $0,3$, $0,4$, and $0,5$.

Furthermore, for the maximum of the depth of the tree, we tested each integer within the range $(0, 21)$. The next parameter is the minimum child weight. This parameter sets the minimum sum of instance needed in a child, and it usually takes values in a range of $[0, 3]$. The last parameter we optimized is the subsample ratio of the training instance. We tested values from $0,6$ to 1.

4.3.3.1 Economic Data Set Results

Starting with our first data set, the economic one, we observed that the minimum mean squared error was achieved on XOM's stock, while the worst one on IBM's stock.

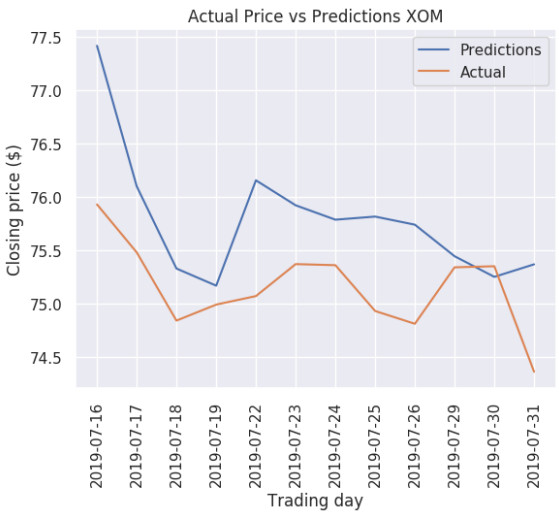

Figure 32: The Best Economic Model in XGBoost

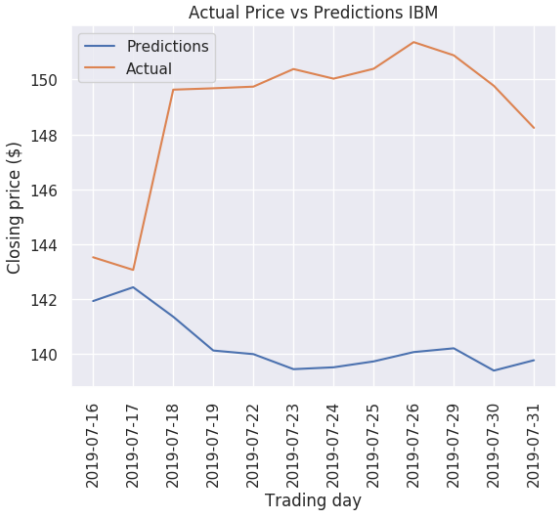

Figure 33: The Worst Economic Model in XGBoost

Figure 34 shows the importance of features in the stock with the best result. As we can see the most important feature is the previous day's price. Moreover, all the lags from the variable volume are important, whilst the technical indicators are less important.

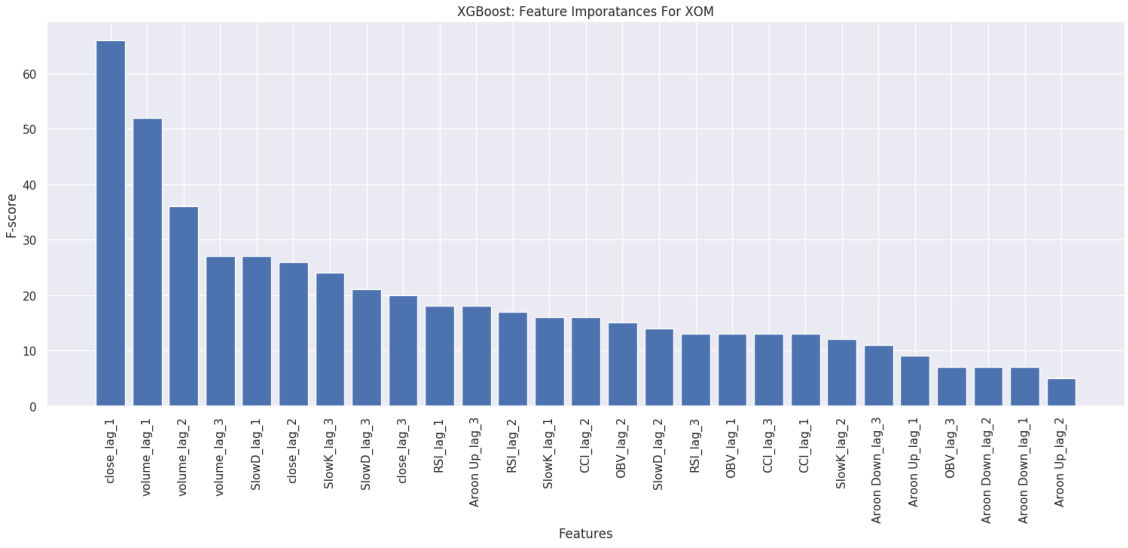

Figure 34: Ecominc Data Set Feature Importance in XOM stock

Table 10 summarizes the grid search results for the tickers. Beginning from the colsample bytree, the results show that the optimal value depends on the stock, and none stands out. For 8 stocks, the value was set to $0,9$, for 6 to 1, and for 4, it was set to $0,8$. The minimum value we encountered was $0,4$, which was observed for only one stock, that of IBM. Continuing to the gamma parameter, for 16 stocks, the optimal value of the parameter was $0,3$, whilst for 5 it was $0,4$, and for 9 it was the $0,5$ option.

Furthermore, for the maximum of the depth of the tree, the results for the optimal value are inconclusive since the value for each stock differs. But, for the option 1, was the optimal value for 7 stocks. Proceeding to the next parameter, that is the minimum child weight. For 18 stocks the optimal value was 0, for 3 was the option of 2 and, for 9 stocks, the option was set to 3. Lastly,

for the subsample ratio, the results in table 10 show that for 13 stocks the optimal value is $0,60$, but for all the other options none stands out.

Table 10: Economic XGBoost Results Per Ticker

Ticker	Mean Squared Error	Colsample Bytree	Gamma	Max Depth	Min Child Weight	Subsample
AAPL	24,27	0,90	0,30	1,00	3,00	0,60
AXP	2,00	0,90	0,30	5,00	0,00	0,70
BA	97,48	0,90	0,30	9,00	3,00	0,90
CAT	5,29	1,00	0,30	1,00	0,00	1,00
CSCO	0,64	0,50	0,50	8,00	3,00	0,80
CVX	2,23	1,00	0,30	5,00	0,00	0,90
DIS	17,69	0,40	0,30	8,00	0,00	1,00
DOW	1,32	0,60	0,50	3,00	2,00	0,60
GS	40,92	0,90	0,30	1,00	0,00	0,60
HD	4,71	0,70	0,50	10,00	0,00	0,60
IBM	85,01	0,50	0,30	8,00	0,00	0,60
INTC	0,93	0,80	0,50	7,00	3,00	0,60
JNJ	2,73	0,90	0,30	1,00	0,00	0,80
JPM	2,90	0,70	0,50	6,00	3,00	0,60
KO	3,07	0,80	0,30	1,00	2,00	0,60
MCD	3,05	0,80	0,30	6,00	0,00	1,00
MMM	6,76	1,00	0,30	1,00	3,00	1,00
MRK	0,54	1,00	0,50	9,00	3,00	0,80
MSFT	4,94	0,90	0,30	9,00	2,00	1,00
NKE	0,65	0,60	0,30	1,00	0,00	0,60
PFE	0,97	0,80	0,30	6,00	3,00	0,60
PG	8,33	0,90	0,40	3,00	0,00	0,60
TRV	8,50	0,50	0,50	10,00	0,00	0,60
UNH	15,28	0,50	0,30	4,00	0,00	0,80
UTX	2,36	1,00	0,40	9,00	0,00	1,00
V	4,61	0,60	0,50	12,00	0,00	1,00
VZ	0,70	0,40	0,50	3,00	0,00	0,70
WBA	0,93	1,00	0,40	7,00	0,00	0,70
WMT	0,84	0,90	0,40	7,00	3,00	0,90
XOM	0,60	0,70	0,40	4,00	0,00	0,60

4.3.3.2 Sentiment Data Set Results

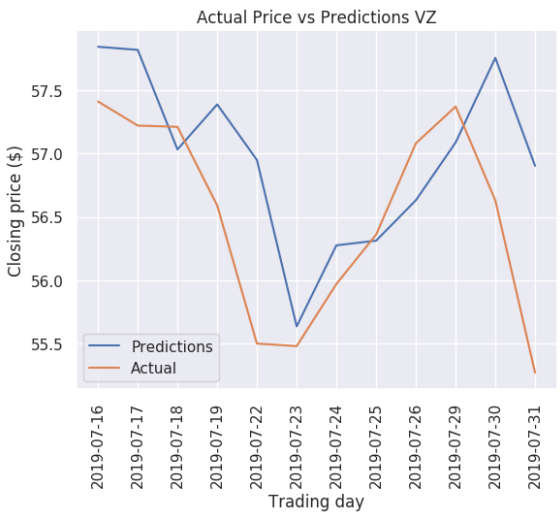

Figure 35: The Best Sentiment Model in XGBoost

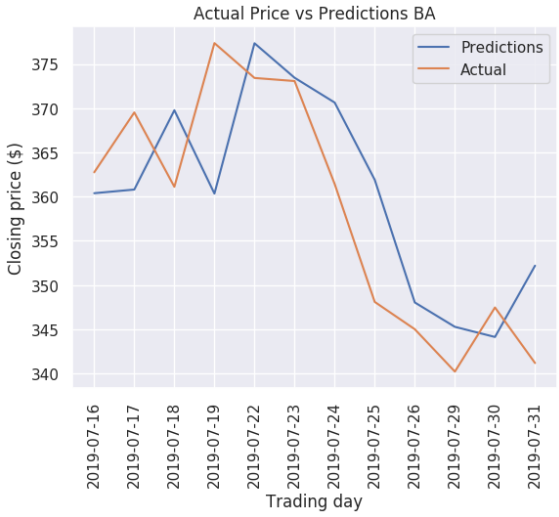

Figure 36: The Worst Sentiment Model in XGBoost

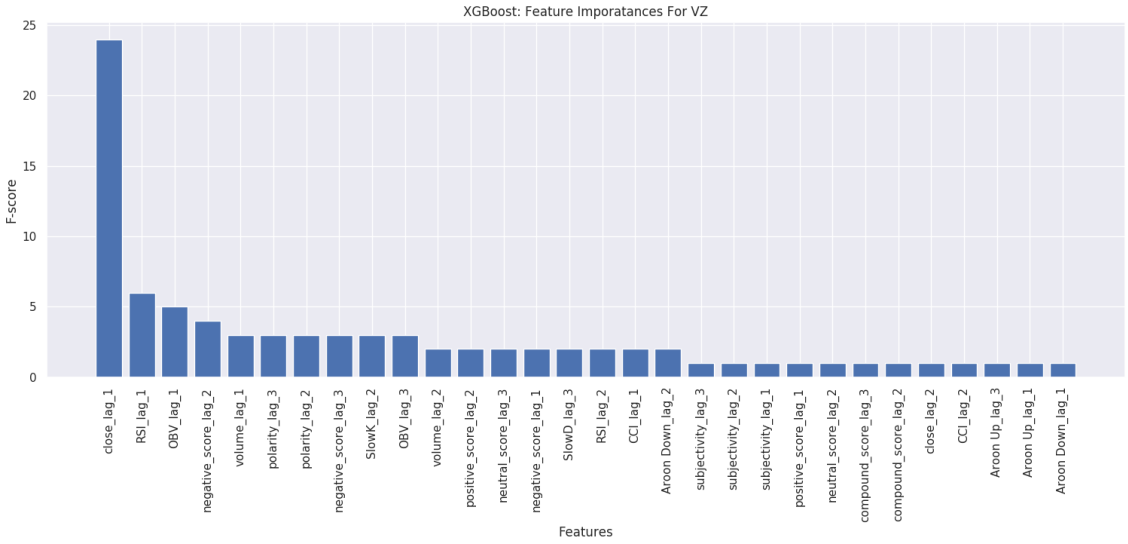

Figure 37: Sentiment Data Set Feature Importance in VZ stock

Figures 35 and 36 exhibit the best and the worst model of the sentiment data set in the XGBoost model. The best mode, in terms of mean squared error, was observed in Verizon's stock, whilst the worst on Boeing's.

Continuing with the importance of the features we see that the previous day's price is still the most important feature, but the importance of the volume variable has been decreased. Instead, the importance of two technical indicators has been improved and the new variable that captures the negative sentiment has been observed.

The grid search results showed, do not differ significantly. For colsample bytree, for most of the stocks (22), the optimal values were close to 1. For 10 stocks, the value was set to $1,00$, for 9 to $0,9$, and for 5, it was set to $0,8$. The minimum value we encountered was $0,3$, which was observed for only one stock, that of IBM. Continuing to the gamma parameter, the results showed that for 18 stocks, the optimal value of the parameter was $0,3$, whilst for 7 it was $0,5$, and for 5 it was the $0,4$ option. Furthermore, for the maximum of the depth of the tree, the results for the optimal value are inconclusive since the value for each stock differs. Proceeding to the next parameter, that is the minimum child weight. In most cases, the optimal value was 0, but 8 was the option of 2 and, for 9 stocks, the option was set to 3. The last parameter we optimized is the subsample ratio of the training instance. The results in table 12 show that the optimal value depends on the stock, and none stands out.

Table 11: Sentiment XGBoost Results Per Ticker

Ticker	Mean Squared Error	Colsample Bytree	Gamma	Max Depth	Min Child Weight	Subsample
AAPL	22,46	0,80	0,40	6,00	0,00	0,60
AXP	2,17	0,50	0,30	1,00	3,00	0,90
BA	75,38	0,90	0,40	13,00	2,00	0,80
CAT	4,82	0,80	0,30	1,00	3,00	1,00
CSCO	0,50	0,80	0,30	8,00	2,00	0,60
CVX	1,29	1,00	0,30	2,00	0,00	0,90
DIS	27,48	0,80	0,30	9,00	0,00	0,90
DOW	1,34	0,70	0,50	3,00	0,00	0,60
GS	57,85	0,70	0,30	1,00	0,00	0,70
HD	4,78	0,60	0,30	11,00	0,00	0,70
IBM	59,59	0,30	0,50	2,00	3,00	0,70

Table 11: Sentiment XGBoost Results Per Ticker

Ticker	Mean Squared Error	Colsample Bytree	Gamma	Max Depth	Min Child Weight	Subsample
INTC	1,37	0,50	0,50	2,00	3,00	0,70
JNJ	3,41	0,90	0,30	1,00	0,00	1,00
JPM	3,53	1,00	0,50	7,00	0,00	0,70
KO	3,57	1,00	0,40	3,00	3,00	0,60
MCD	4,12	0,90	0,30	11,00	0,00	0,90
MMM	7,09	0,90	0,30	1,00	3,00	0,70
MRK	0,55	0,90	0,50	6,00	3,00	0,80
MSFT	5,34	0,70	0,40	9,00	2,00	1,00
NKE	0,54	0,80	0,30	1,00	3,00	0,90
PFE	0,96	0,90	0,40	5,00	0,00	0,60
PG	6,39	0,90	0,40	8,00	2,00	0,60
TRV	3,14	1,00	0,50	5,00	2,00	0,60
UNH	24,41	0,90	0,40	3,00	2,00	0,60
UTX	2,78	0,80	0,30	11,00	0,00	1,00
V	5,54	0,70	0,50	7,00	2,00	0,70
VZ	0,64	0,90	0,30	1,00	0,00	0,60
WBA	1,16	0,80	0,40	7,00	0,00	0,70
WMT	1,00	1,00	0,30	6,00	3,00	1,00
XOM	1,02	0,60	0,30	6,00	3,00	0,60

4.3.3.3 Pagerank Data Set Results

As we noted previously, XGBoost is the model which estimated error was the lowest on average. The best model of XGBoost was recorded in the CSCO stock and it was equal to $0,39$. On the other hand, in BA's stock, a mean squared error of $79,80$ was calculated. As it is evident, XGBoost did not provide the lowest score on all variables, but the errors from XGBoost have a lower average and even lower standard deviation, which means that the model acts better in general.

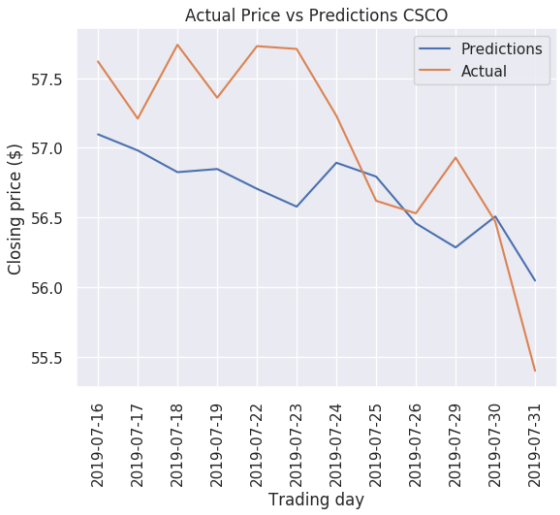

Figure 38: The Best PageRank Model in XGBoost

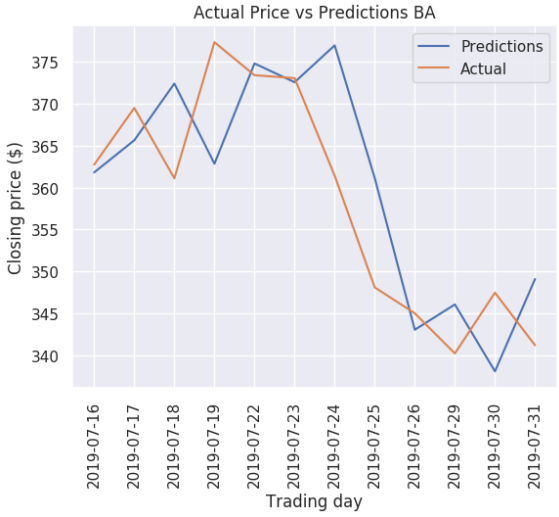

Figure 39: The Worst PageRank Model in XGBoost

As it was noted in the literature, the negative sentiment score was more important than the positive or compound scores. This can be confirmed based on our research as well. Moreover, we could not confirm the importance of a specific lag; the mixed results showed that which time lag is important is highly dependant on the feature.

Beginning from the colsample bytree, for most of the stocks (22), the optimal values were close

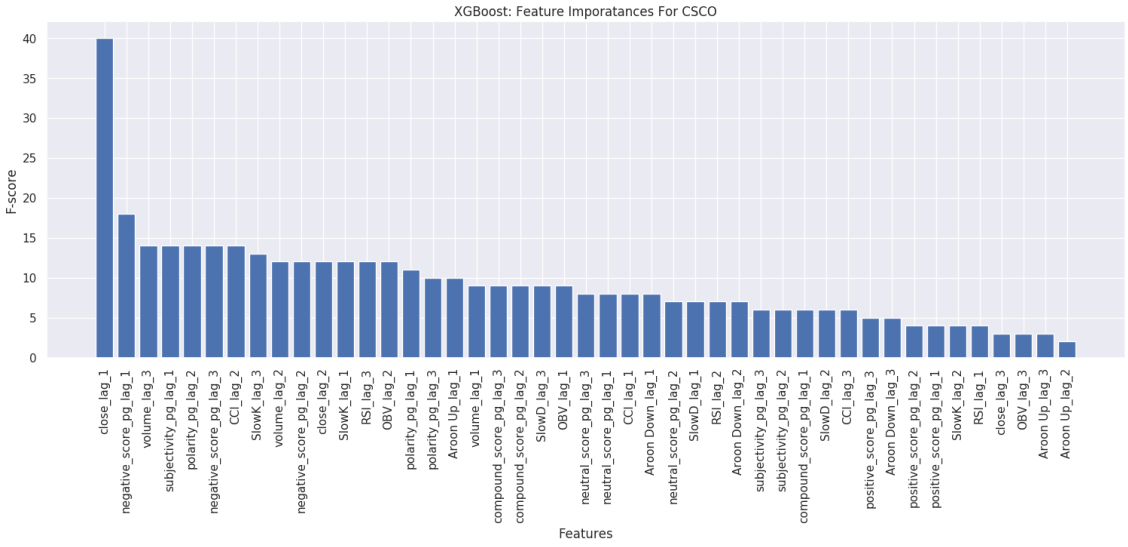

Figure 40: Feature Importance in CSCO stock

to 1. For 10 stocks, the value was set to $1,00$, for 9 to $0,9$, and for 5, it was set to $0,8$. The minimum value we encountered was $0,3$, which was observed for only one stock, that of IBM. Continuing to the gamma parameter, the results showed that for 18 stocks, the optimal value of the parameter was $0,3$, whilst for 7 it was $0,5$, and for 5 it was the $0,4$ option. Furthermore, for the maximum of the depth of the tree, the results for the optimal value are inconclusive since the value for each stock differs. Proceeding to the next parameter, that is the minimum child weight. In most cases, the optimal value was 0, but 8 was the option of 2 and, for 9 stocks, the option was set to 3. The last parameter we optimized is the subsample ratio of the training instance. The results in table 12 show that the optimal value depends on the stock, and none stands out.

Table 12: PageRank XGBoost Results Per Ticker

Ticker	Mean Squared Error	Colsample Bytree	Gamma	Max Depth	Min Child Weight	Subsample
AAPL	33,79	0,80	0,40	9,00	0,00	0,60
AXP	2,71	0,90	0,30	6,00	2,00	0,60
BA	79,80	0,90	0,50	9,00	0,00	0,80
CAT	4,94	0,90	0,30	1,00	3,00	1,00
CSCO	0,39	0,80	0,40	7,00	3,00	0,60
CVX	0,82	1,00	0,50	11,00	0,00	0,90
DIS	20,27	0,50	0,30	4,00	0,00	0,90
DOW	1,37	0,80	0,30	2,00	3,00	0,60
GS	56,56	0,90	0,30	1,00	0,00	0,60
HD	3,39	0,90	0,50	8,00	2,00	0,70
IBM	69,75	0,30	0,40	5,00	3,00	0,60
INTC	1,14	0,40	0,30	1,00	3,00	0,70
JNJ	2,21	1,00	0,30	1,00	0,00	0,90
JPM	3,12	1,00	0,50	6,00	2,00	0,70
KO	3,35	1,00	0,40	2,00	2,00	0,60
MCD	5,91	1,00	0,30	4,00	0,00	1,00
MMM	6,53	1,00	0,30	1,00	2,00	1,00
MRK	0,50	1,00	0,50	6,00	3,00	0,80
MSFT	5,08	1,00	0,30	7,00	2,00	0,90
NKE	0,64	0,90	0,30	3,00	0,00	0,80
PFE	0,97	1,00	0,50	3,00	0,00	0,60
PG	6,29	0,70	0,30	5,00	0,00	0,60
TRV	2,82	0,40	0,30	4,00	0,00	0,70
UNH	21,24	0,50	0,50	8,00	2,00	0,80

Table 12: PageRank XGBoost Results Per Ticker

Ticker	Mean Squared Error	Colsample Bytree	Gamma	Max Depth	Min Child Weight	Subsample
UTX	2,94	0,80	0,30	10,00	0,00	1,00
V	4,54	0,60	0,40	10,00	0,00	0,70
VZ	0,85	0,90	0,30	6,00	3,00	0,70
WBA	1,15	1,00	0,30	7,00	2,00	0,90
WMT	0,83	0,80	0,30	6,00	3,00	1,00
XOM	0,84	0,60	0,30	3,00	3,00	0,60

4.3.4 LSTM Results

The Long Short Term Memory neural network did not produce good results on average. The main problem of the model was that when it failed, it failed tremendously. The most profound case is that of Apple's stock, in which the model produces just a flat line, and it fails to capture any shifts or the volatility of the data. On the other hand, in some cases, the LSTM produced the best score; Nike's stock is such an example 41, 43 45.

Since in LSTM we cannot perform a grid search for the best parameters, we provide the reader with the code of the LSTM network that we designed. It is a two-layer neural network, which we trained for 2.000 epochs. In table 13, we present the mean squared errors for all of the stocks.

```
model = Sequential()
model.add(LSTM(
    units=50,
    input_shape=(x_train.shape[1], x_train.shape[2])))
model.add(Dense(1, activation="linear"))
opt = Adadelta()
model.compile(loss='mean_squared_error', optimizer=opt)
model.fit(x_train, y_train, epochs=2000, batch_size=1, verbose=2)
```

Listing 7: The LSTM model

Table 13: Economic LSTM Results Per Ticker

Ticker	Economic	Sentiment	PageRank
AAPL	458,80	804,92	606,43
AXP	43,37	72,84	34,43
BA	344,27	243,50	251,22
CAT	6,70	8,03	10,78
CSCO	1,59	0,30	1,91
CVX	1,47	1,76	2,26
DIS	190,41	182,01	125,17
DOW	3,77	4,33	4,06
GS	576,77	574,77	574,04
HD	3,19	3,37	3,10
IBM	68,89	42,36	48,80
INTC	1,50	0,92	1,84
JNJ	55,68	51,99	60,83
JPM	75,68	74,86	77,88
KO	16,68	19,24	22,81
MCD	354,48	205,22	245,83
MMM	115,79	137,20	185,08
MRK	6,87	6,78	7,09
MSFT	487,04	488,77	478,40
NKE	0,72	0,46	0,45
PFE	3,52	2,38	2,42

Table 13: Economic LSTM Results Per Ticker

Ticker	Economic	Sentiment	PageRank
PG	3,67	4,99	3,61
TRV	5,95	3,42	6,40
UNH	173,24	134,08	117,14
UTX	17,25	17,74	19,06
V	43,41	33,73	62,81
VZ	7,34	8,54	9,44
WBA	172,32	153,63	177,62
WMT	1,20	1,21	1,01
XOM	0,52	0,59	0,47

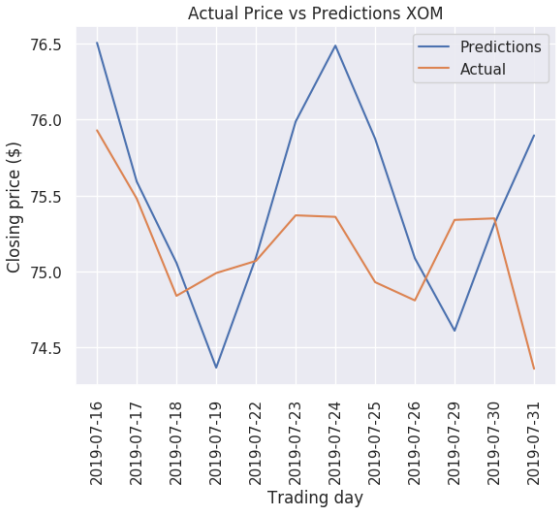

Figure 41: The Best Economic Model in LSTM

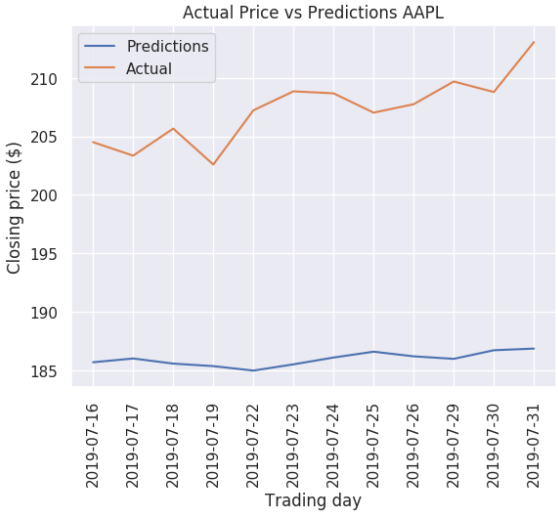

Figure 42: The Worst Economic Model in LSTM

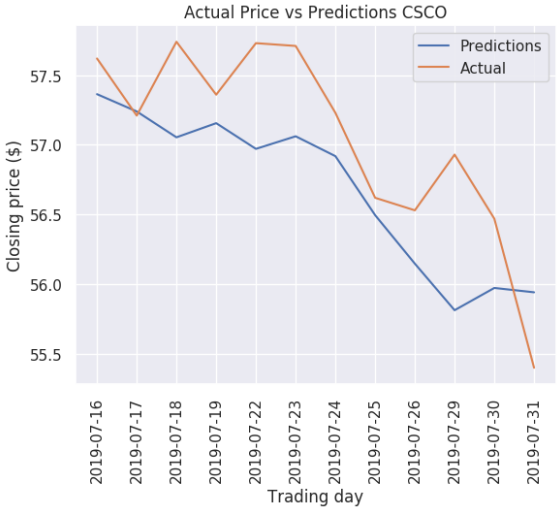

Figure 43: The Best Sentiment Model in LSTM

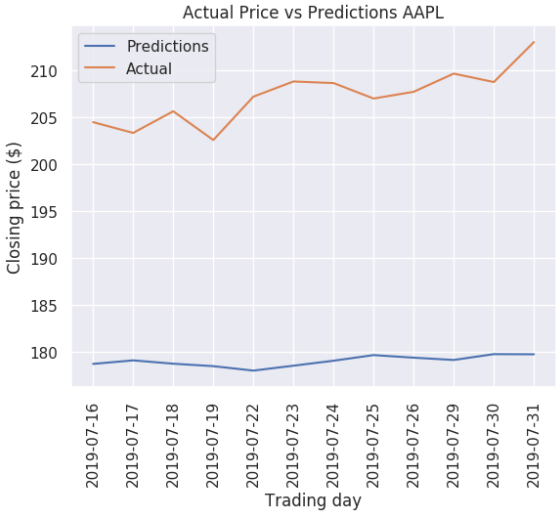

Figure 44: The Worst Sentiment Model in LSTM

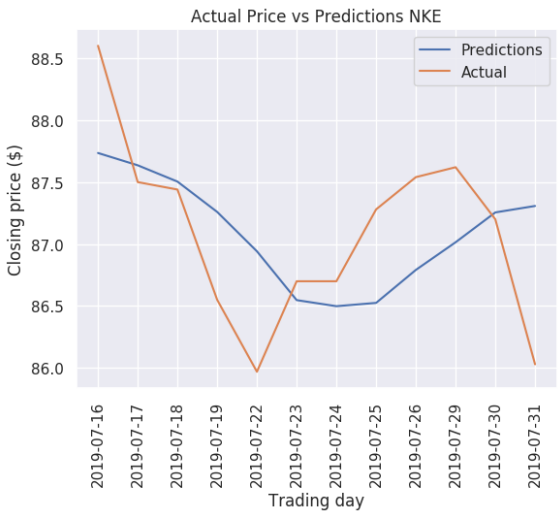

Figure 45: The Best PageRank Model in LSTM

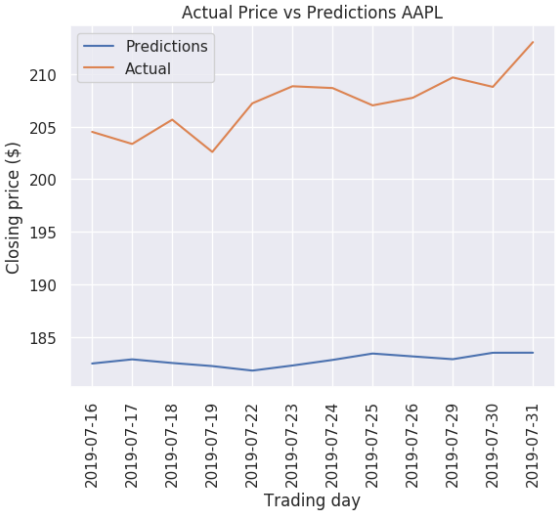

Figure 46: The Worst PageRank Model in LSTM

4.3.5 k-Nearest Neighbors Results

k-Nearest Neighbors had surprisingly good results even if it was not the best model. For example, in Intel's stock, k-NN produced the best score from any other model. On the other hand, in Boeing's stock, the model failed to keep up with both the trend and the volatility.

Table 14: PageRank k-Nearest Neighbors Results Per Ticker

Ticker	Economic	Sentiment	PageRank
AAPL	106,91	106,91	106,91
AXP	49,26	49,26	49,26
BA	405,93	405,93	405,93
CAT	4,63	4,63	4,63
CSCO	1,73	1,73	1,73
CVX	1,12	1,12	1,12
DIS	8,08	8,08	8,08
DOW	3,23	3,23	3,23
GS	51,17	51,17	51,17
HD	7,42	7,42	7,42
IBM	88,52	88,52	88,52
INTC	1,04	1,04	1,04
JNJ	61,04	61,04	61,04
JPM	19,08	19,08	19,08
KO	15,54	15,54	15,54
MCD	82,74	82,74	82,74
MMM	61,96	61,96	61,96
MRK	4,51	4,51	4,51
MSFT	5,28	5,28	5,28
NKE	3,97	3,97	3,97
PFE	3,79	3,79	3,79
PG	28,95	28,95	28,95
TRV	4,08	4,08	4,08
UNH	179,18	179,18	179,18
UTX	19,42	19,42	19,42
V	11,13	11,13	11,13
VZ	6,67	6,67	6,67
WBA	164,42	164,42	164,42
WMT	1,96	1,96	1,96
XOM	1,32	1,32	1,32

For k-Nearest Neighbors, the only parameter we could optimize was the number of neighbors. We tested values in the range of $[2, 9]$. The results showed that for three stocks, the optimal value of neighbors was 3, 4, and 7. For all the other stocks, the results balanced through all of the other options. The surprising result was that the number of neighbors is independent of the data set, meaning that for each stock the number of neighbors remains constant for all the data sets.

Table 15: PageRank k-Nearest Neighbors Results Per Ticker

Ticker	Economic	Sentiment	PageRank
AAPL	8,00	8,00	8,00
AXP	5,00	5,00	5,00
BA	9,00	9,00	9,00
CAT	8,00	8,00	8,00
CSCO	6,00	6,00	6,00
CVX	9,00	9,00	9,00
DIS	2,00	2,00	2,00
DOW	5,00	5,00	5,00
GS	9,00	9,00	9,00
HD	5,00	5,00	5,00
IBM	5,00	5,00	5,00
INTC	3,00	3,00	3,00
JNJ	9,00	9,00	9,00
JPM	9,00	9,00	9,00
KO	6,00	6,00	6,00
MCD	9,00	9,00	9,00
MMM	9,00	9,00	9,00
MRK	2,00	2,00	2,00
MSFT	4,00	4,00	4,00
NKE	2,00	2,00	2,00
PFE	9,00	9,00	9,00
PG	2,00	2,00	2,00
TRV	6,00	6,00	6,00
UNH	6,00	6,00	6,00
UTX	5,00	5,00	5,00
V	6,00	6,00	6,00
VZ	6,00	6,00	6,00
WBA	6,00	6,00	6,00
WMT	7,00	7,00	7,00
XOM	5,00	5,00	5,00

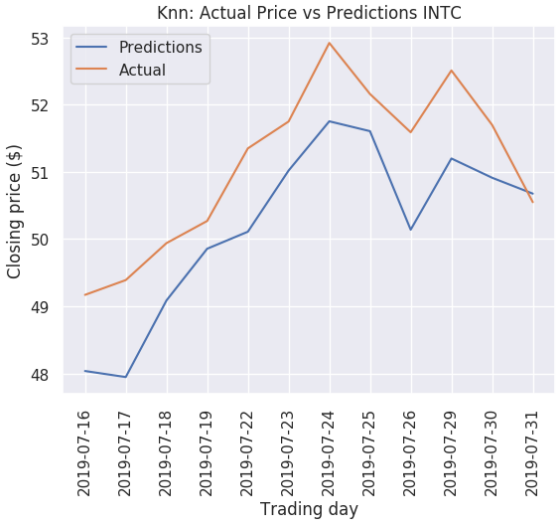

Figure 47: The Best Economic Model in kNN

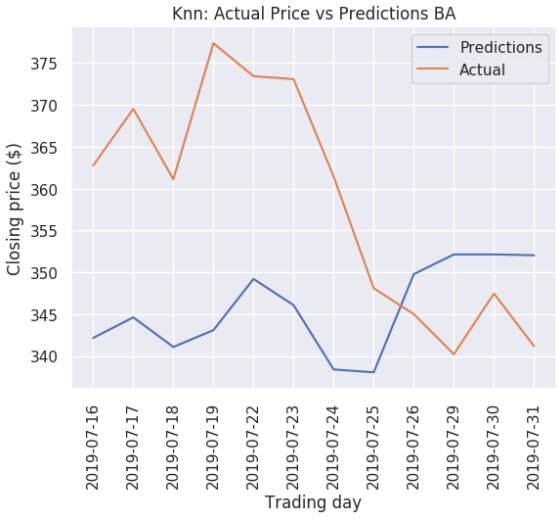

Figure 48: The Worst Economic Model in kNN

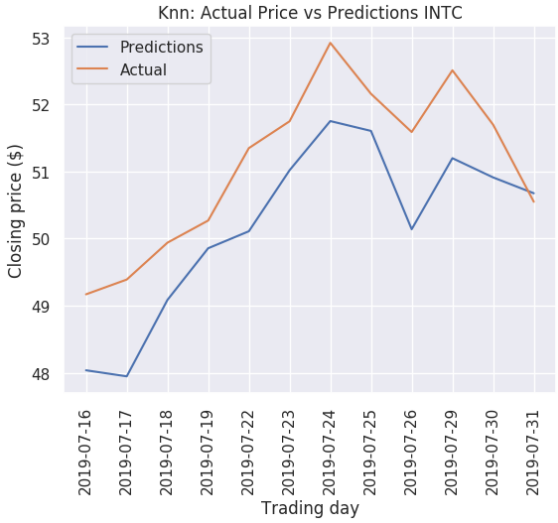

Figure 49: The Best Sentiment Model in kNN

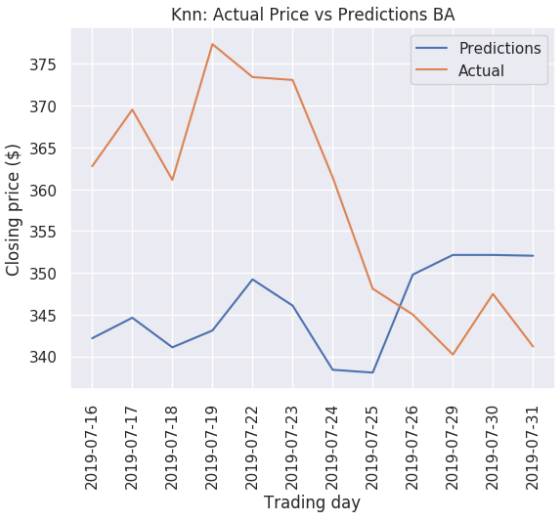

Figure 50: The Worst Sentiment Model in kNN

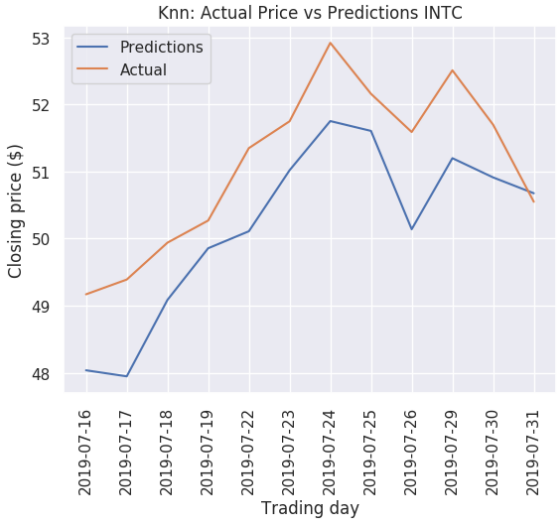

Figure 51: The Best PageRank Model in kNN

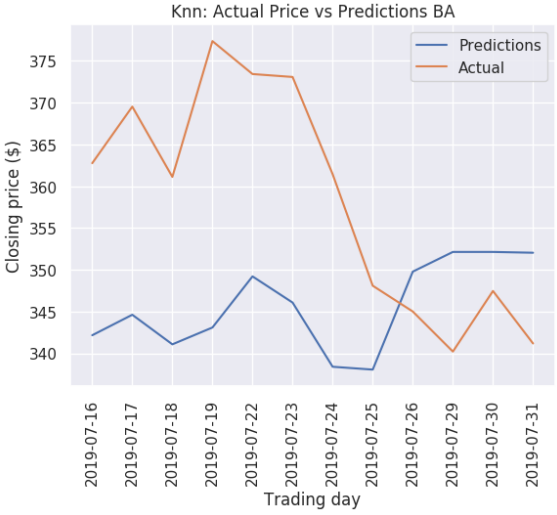

Figure 52: The Worst PageRank Model in kNN

5 Evaluation

This chapter evaluates the results of the methodology we followed. To evaluate our methodology we designed a simple greedy strategy and tested it for all data sets. The strategy was very simple and naive. For each day we used the next day's prediction and we sold all the stocks that we predicted were going to have a negative return and we bought as much as we could from the stocks that were expected to have a positive return. For all the data sets we began with the same Portfolio, which is presented in table 16. We acknowledge that this is not the optimal strategy, because we do not take into account the beneficial effects of diversification. Our strategy bears a significant risk, that of buying one particular stock based on the predictions, we may lose all of our budgets. Moreover, we used only the predictions of the XGBoost models since this model was our best overall, but in a real-life scenario we would use the model that gave the minimum squared error on the testing sample for each particular stock.

Table 16: Initial Portfolio

Ticker	Quantity	Price	Amount
AAPL	1	204,5	204,5
CAT	1	139,09	139,09
HD	1	217,26	217,26
UNH	1	264,66	264,66
XOM	1	75,93	75,93
IBM	1	143,53	143,53
TRV	1	154,59	154,59
V	1	179,31	179,31
BA	1	362,75	362,75
INTC	1	49,17	49,17
GS	1	215,52	215,52
JNJ	1	132,5	132,5
WBA	1	55,81	55,81
DOW	1	52,32	52,32
VZ	1	57,41	57,41
JPM	1	115,12	115,12
PG	1	115,89	115,89
KO	1	52,14	52,14
MSFT	1	137,08	137,08
CVX	1	124,76	124,76
MRK	1	81,59	81,59
CSCO	1	57,62	57,62
UTX	1	133,19	133,19
MMM	1	176,49	176,49
WMT	1	114,76	114,76
MCD	1	213,72	213,72
PFE	1	42,85	42,85
AXP	1	128,06	128,06
DIS	1	144,3	144,3

5.1 Economic Data Set Evaluation

We began with the evaluation of the economic data set results, on the XGBoost model. Our predictions suggested that we should sell MRK, MCD, MSFT, V, PFE, DOW, JNJ, WMT, DIS, BA, HD, AXP, CAT, IBM, TRV, MMM, JPM, AAPL, NKE, KO, CSCO, GS, and PG and buy 4 shares of Intel's stock. Our predictions proved correct and Intel's stock recorded a gain, so our

portfolio now had a total evaluation of $4.041,61$\$. Our decisions for 2019/7/18 also proved correct and, again, we recorded a gain of $0,30\%$. On the contrary, for 2019/7/19 our decisions lead to a negative return of $-0,47\%$. The biggest gain was observed on 2019/7/30 with a daily return of $1,87\%$, whilst our worst day was the next day, where we lost most of our gains (-75,99\$). Finally, our cumulative return for the whole period was positive, $0,75\%$. The table below (17) describes the daily transactions alongside the daily and cumulative returns.

5.2 Sentiment Data Set Evaluation

In the sentiment's data set we began by selling most of our stocks in our portfolio and buying only one. More specifically, we sold 23 stocks and bought WBA's stock. This decision was wrong, as we sold Intel's stock, which as we have seen in the previous data set leads to a significant gain. These decisions naturally lead to a significant loss of $-1,91\%$. Although the next day (2019/7/18) our predictions resulted in a daily positive return of $0,26\%$, that was not enough to overturn the cumulative negative return. Our best return was 2019/7/29 of $1,39\%$. Even that return could not reverse our losses, thus the final result of this data set was a cumulative loss of $-3,05\%$. Table 18 describes all the transactions and the returns.

5.3 PageRank Data Set Evaluation

For the PageRank data set in the first day, we sold the following stocks, V, MRK, PFE, JNJ, HD, AXP, WMT, MCD, NKE, CAT, TRV, CVX, JPM, MMM, CSCO, INTC, IBM, KO, PG, DIS, and GS. This decreased the value of bought stocks to $1.343,65$\$ and increased the available funds to $2.686,87$\$. At this point, 10 units of ticker UNH were bought at $264,66$ per unit. This updated the value of bought stocks to $3.990,25$\$ and the available funds to $40,27$\$. Since we were still on the same day, the evaluation of the portfolio had not changed, because we had not updated the prices yet. On the next day, after updating the prices, we saw that our portfolio had a value of $4.051,48$\$, which meant that our strategy and predictions resulted in a positive return of $1,5\%$.

On the second day, we decided to sell the stocks of VZ, AAPL, and UTX and buy 3 units of Nike's stock. This decision resulted in a loss of $75,88$\$ and a total return of $-1,3\%$. The decision was based on the prediction that Nike's stock would have a positive return. On the contrary, the actual result was a loss of $-1,07\%$. We followed the same strategy for every day. We ended up having two stocks, that of XOM's and Intel's on 25/7/19. From then and onwards, the predictions showed that Intel's stock would have a positive return, so according to our strategy we held on to our stocks. This never happened, and our overall return was negative, resulting in a loss of -122\$ or $-3,03\%$. Lastly, the above and all the transactions for this portfolio are described in detail in table 19.

Table 17: Economic's Data Set Daily Transactions

Date	Action	Ticker	Price per Unit	Quantity	Current Position	Available Funds	Total Valuation	Daily Return	Cumulative Return
16/7/19	Prices Update				4.030,52	0,00	4.030,52		
16/7/19	Sell	MRK	81,59	1,00	3.948,93	81,59	4.030,52		
16/7/19	Sell	MCD	213,72	1,00	3.735,21	295,31	4.030,52		
16/7/19	Sell	MSFT	137,08	1,00	3.598,13	432,39	4.030,52		
16/7/19	Sell	V	179,31	1,00	3.418,82	611,70	4.030,52		
16/7/19	Sell	PFE	42,85	1,00	3.375,97	654,55	4.030,52		
16/7/19	Sell	DOW	52,32	1,00	3.323,65	706,87	4.030,52		
16/7/19	Sell	JNJ	132,50	1,00	3.191,15	839,37	4.030,52		
16/7/19	Sell	WMT	114,76	1,00	3.076,39	954,13	4.030,52		
16/7/19	Sell	DIS	144,30	1,00	2.932,09	1.098,43	4.030,52		
16/7/19	Sell	BA	362,75	1,00	2.569,34	1.461,18	4.030,52		
16/7/19	Sell	HD	217,26	1,00	2.352,08	1.678,44	4.030,52		
16/7/19	Sell	AXP	128,06	1,00	2.224,02	1.806,50	4.030,52		
16/7/19	Sell	CAT	139,09	1,00	2.084,93	1.945,59	4.030,52		
16/7/19	Sell	IBM	143,53	1,00	1.941,40	2.089,12	4.030,52		
16/7/19	Sell	TRV	154,59	1,00	1.786,81	2.243,71	4.030,52		
16/7/19	Sell	MMM	176,49	1,00	1.610,32	2.420,20	4.030,52		
16/7/19	Sell	JPM	115,12	1,00	1.495,20	2.535,32	4.030,52		
16/7/19	Sell	AAPL	204,50	1,00	1.290,70	2.739,82	4.030,52		
16/7/19	Sell	NKE	88,60	1,00	1.202,10	2.828,42	4.030,52		
16/7/19	Sell	KO	52,14	1,00	1.149,96	2.880,56	4.030,52		
16/7/19	Sell	CSCO	57,62	1,00	1.092,34	2.938,18	4.030,52		
16/7/19	Sell	GS	215,52	1,00	876,82	3.153,70	4.030,52		
16/7/19	Sell	PG	115,89	1,00	760,93	3.269,59	4.030,52		
16/7/19	Buy	INTC	49,17	66,00	4.006,15	24,37	4.030,52		
17/7/19	Prices Update				4.017,24	24,37	4.041,61	0,28%	0,28%
17/7/19	Sell	UTX	130,10	1,00	3.887,14	154,47	4.041,61		
17/7/19	Sell	UNH	266,65	1,00	3.620,49	421,12	4.041,61		
17/7/19	Sell	VZ	57,22	1,00	3.563,27	478,34	4.041,61		
17/7/19	Sell	XOM	75,48	1,00	3.487,79	553,82	4.041,61		
17/7/19	Sell	INTC	49,39	67,00	178,66	3.862,95	4.041,61		
17/7/19	Buy	DOW	51,60	74,00	3.997,06	44,55	4.041,61		
18/7/19	Prices Update				4.009,00	44,55	4.053,55	0,30%	0,57%
18/7/19	Sell	CVX	124,68	1,00	3.884,32	169,23	4.053,55		
18/7/19	Buy	DIS	141,63	1,00	4.025,95	27,60	4.053,55		
19/7/19	Prices Update				4.006,77	27,60	4.034,37	-0,47%	0,10%
19/7/19	Sell	DOW	51,52	74,00	194,29	3.840,08	4.034,37		
19/7/19	Buy	XOM	74,99	51,00	4.018,78	15,59	4.034,37		
22/7/19	Prices Update				4.023,35	15,59	4.038,94	0,11%	0,21%
22/7/19	Sell	DIS	140,84	1,00	3.882,51	156,43	4.038,94		
22/7/19	Buy	VZ	55,50	2,00	3.993,51	45,43	4.038,94		
23/7/19	Prices Update				4.009,48	45,43	4.054,91	0,40%	0,61%
23/7/19	Buy	PFE	43,09	1,00	4.052,57	2,34	4.054,91		
24/7/19	Prices Update				4.053,39	2,34	4.055,73	0,02%	0,63%
24/7/19	Sell	WBA	55,20	1,00	3.998,19	57,54	4.055,73		
24/7/19	Buy	VZ	55,97	1,00	4.054,16	1,57	4.055,73		
25/7/19	Prices Update				4.033,18	1,57	4.034,75	-0,52%	0,10%
25/7/19	Sell	PFE	42,67	1,00	3.990,51	44,24	4.034,75		
26/7/19	Prices Update				3.986,55	44,24	4.030,79	-0,10%	0,01%
26/7/19	Sell	VZ	57,08	3,00	3.815,31	215,48	4.030,79		
26/7/19	Buy	MMM	173,98	1,00	3.989,29	41,50	4.030,79		
29/7/19	Prices Update				4.019,10	41,50	4.060,60	0,74%	0,75%
29/7/19	Sell	XOM	75,34	51,00	176,76	3.883,84	4.060,60		
29/7/19	Sell	MMM	176,76	1,00	0,00	4.060,60	4.060,60		
29/7/19	Buy	BA	340,21	11,00	3.742,31	318,29	4.060,60		
29/7/19	Buy	VZ	57,37	5,00	4.029,16	31,44	4.060,60		
30/7/19	Prices Update				4.105,21	31,44	4.136,65	1,87%	2,63%
31/7/19	Prices Update				4.029,33	31,44	4.060,77	-1,83%	0,75%

Table 18: Sentiment's Data Set Daily Transactions

Date	Action	Ticker	Price per Unit	Quantity	Current Position	Available Funds	Total Valuation	Daily Return	Cumulative Return
16/7/19	Prices Update				4.030,52	0,00	4.030,52		
16/7/19	Sell	CVX	124,76	1,00	3.905,76	124,76	4.030,52		
16/7/19	Sell	HD	217,26	1,00	3.688,50	342,02	4.030,52		
16/7/19	Sell	MRK	81,59	1,00	3.606,91	423,61	4.030,52		
16/7/19	Sell	V	179,31	1,00	3.427,60	602,92	4.030,52		
16/7/19	Sell	MSFT	137,08	1,00	3.290,52	740,00	4.030,52		
16/7/19	Sell	MCD	213,72	1,00	3.076,80	953,72	4.030,52		
16/7/19	Sell	WMT	114,76	1,00	2.962,04	1.068,48	4.030,52		
16/7/19	Sell	PFE	42,85	1,00	2.919,19	1.111,33	4.030,52		
16/7/19	Sell	AAPL	204,50	1,00	2.714,69	1.315,83	4.030,52		
16/7/19	Sell	BA	362,75	1,00	2.351,94	1.678,58	4.030,52		
16/7/19	Sell	TRV	154,59	1,00	2.197,35	1.833,17	4.030,52		
16/7/19	Sell	DIS	144,30	1,00	2.053,05	1.977,47	4.030,52		
16/7/19	Sell	CAT	139,09	1,00	1.913,96	2.116,56	4.030,52		
16/7/19	Sell	AXP	128,06	1,00	1.785,90	2.244,62	4.030,52		
16/7/19	Sell	IBM	143,53	1,00	1.642,37	2.388,15	4.030,52		
16/7/19	Sell	CSCO	57,62	1,00	1.584,75	2.445,77	4.030,52		
16/7/19	Sell	NKE	88,60	1,00	1.496,15	2.534,37	4.030,52		
16/7/19	Sell	INTC	49,17	1,00	1.446,98	2.583,54	4.030,52		
16/7/19	Sell	KO	52,14	1,00	1.394,84	2.635,68	4.030,52		
16/7/19	Sell	PG	115,89	1,00	1.278,95	2.751,57	4.030,52		
16/7/19	Sell	JPM	115,12	1,00	1.163,83	2.866,69	4.030,52		
16/7/19	Sell	GS	215,52	1,00	948,31	3.082,21	4.030,52		
16/7/19	Buy	WBA	55,81	55,00	4.017,86	12,66	4.030,52		
17/7/19	Prices Update				3.940,70	12,66	3.953,36	-1,91%	-1,91%
17/7/19	Sell	VZ	57,22	1,00	3.883,48	69,88	3.953,36		
17/7/19	Buy	DOW	51,60	1,00	3.935,08	18,28	3.953,36		
18/7/19	Prices Update				3.945,29	18,28	3.963,57	0,26%	-1,66%
19/7/19	Prices Update				3.918,63	18,28	3.936,91	-0,67%	-2,32%
19/7/19	Sell	UTX	132,39	1,00	3.786,24	150,67	3.936,91		
19/7/19	Sell	DOW	51,52	2,00	3.683,20	253,71	3.936,91		
19/7/19	Sell	UNH	256,65	1,00	3.426,55	510,36	3.936,91		
19/7/19	Buy	MMM	172,61	2,00	3.771,77	165,14	3.936,91		
19/7/19	Buy	XOM	74,99	2,00	3.921,75	15,16	3.936,91		
22/7/19	Prices Update				3.898,41	15,16	3.913,57	-0,59%	-2,90%
23/7/19	Prices Update				3.947,91	15,16	3.963,07	1,26%	-1,67%
23/7/19	Sell	MMM	177,52	3,00	3.415,35	547,72	3.963,07		
23/7/19	Buy	VZ	55,48	9,00	3.914,67	48,40	3.963,07		
24/7/19	Prices Update				3.950,79	48,40	3.999,19	0,91%	-0,78%
24/7/19	Sell	WBA	55,20	56,00	859,59	3.139,60	3.999,19		
24/7/19	Buy	MMM	179,42	17,00	3.909,73	89,46	3.999,19		
24/7/19	Buy	XOM	75,36	1,00	3.985,09	14,10	3.999,19		
25/7/19	Prices Update				3.966,29	14,10	3.980,39	-0,47%	-1,24%
26/7/19	Prices Update				3.901,35	14,10	3.915,45	-1,63%	-2,85%
29/7/19	Prices Update				3.955,63	14,10	3.969,73	1,39%	-1,51%
29/7/19	Sell	MMM	176,76	17,00	950,71	3.019,02	3.969,73		
29/7/19	Buy	WBA	55,12	54,00	3.927,19	42,54	3.969,73		
29/7/19	Buy	PFE	41,45	1,00	3.968,64	1,09	3.969,73		
30/7/19	Prices Update				3.973,54	1,09	3.974,63	0,12%	-1,39%
31/7/19	Prices Update				3.906,39	1,09	3.907,48	-1,69%	-3,05%

Table 19: PagerRank's Data Set Daily Transactions

Date	Action	Ticker	Price per Unit	Quantity	Current Position	Available Funds	Total Valuation	Daily Return	Cumulative Return
16/7/19	Prices Update				4.030,52	0,00	4.030,52		
16/7/19	Sell	V	179,31	1,00	3.851,21	179,31	4.030,52		
16/7/19	Sell	MRK	81,59	1,00	3.769,62	260,90	4.030,52		
16/7/19	Sell	PFE	42,85	1,00	3.726,77	303,75	4.030,52		
16/7/19	Sell	JNJ	132,50	1,00	3.594,27	436,25	4.030,52		
16/7/19	Sell	HD	217,26	1,00	3.377,01	653,51	4.030,52		
16/7/19	Sell	AXP	128,06	1,00	3.248,95	781,57	4.030,52		
16/7/19	Sell	WMT	114,76	1,00	3.134,19	896,33	4.030,52		
16/7/19	Sell	MCD	213,72	1,00	2.920,47	1.110,05	4.030,52		
16/7/19	Sell	NKE	88,60	1,00	2.831,87	1.198,65	4.030,52		
16/7/19	Sell	CAT	139,09	1,00	2.692,78	1.337,74	4.030,52		
16/7/19	Sell	TRV	154,59	1,00	2.538,19	1.492,33	4.030,52		
16/7/19	Sell	CVX	124,76	1,00	2.413,43	1.617,09	4.030,52		
16/7/19	Sell	JPM	115,12	1,00	2.298,31	1.732,21	4.030,52		
16/7/19	Sell	MMM	176,49	1,00	2.121,82	1.908,70	4.030,52		
16/7/19	Sell	CSCO	57,62	1,00	2.064,20	1.966,32	4.030,52		
16/7/19	Sell	INTC	49,17	1,00	2.015,03	2.015,49	4.030,52		
16/7/19	Sell	IBM	143,53	1,00	1.871,50	2.159,02	4.030,52		
16/7/19	Sell	KO	52,14	1,00	1.819,36	2.211,16	4.030,52		
16/7/19	Sell	PG	115,89	1,00	1.703,47	2.327,05	4.030,52		
16/7/19	Sell	DIS	144,30	1,00	1.559,17	2.471,35	4.030,52		
16/7/19	Sell	GS	215,52	1,00	1.343,65	2.686,87	4.030,52		
16/7/19	Buy	UNH	264,66	10,00	3.990,25	40,27	4.030,52		
17/7/19	Prices Update				4.011,21	40,27	4.051,48	0,52%	0,52%
17/7/19	Sell	VZ	57,22	1,00	3.953,99	97,49	4.051,48		
17/7/19	Sell	AAPL	203,35	1,00	3.750,64	300,84	4.051,48		
17/7/19	Sell	UTX	130,10	1,00	3.620,54	430,94	4.051,48		
17/7/19	Buy	NKE	87,50	4,00	3.970,54	80,94	4.051,48		
17/7/19	Buy	XOM	75,48	1,00	4.046,02	5,46	4.051,48		
18/7/19	Prices Update				3.970,14	5,46	3.975,60	-1,87%	-1,36%
19/7/19	Prices Update				3.939,27	5,46	3.944,73	-0,78%	-2,13%
19/7/19	Sell	BA	377,36	1,00	3.561,91	382,82	3.944,73		
19/7/19	Sell	DOW	51,52	1,00	3.510,39	434,34	3.944,73		
19/7/19	Sell	UNH	256,65	11,00	687,24	3.257,49	3.944,73		
19/7/19	Buy	XOM	74,99	43,00	3.911,81	32,92	3.944,73		
22/7/19	Prices Update				3.914,40	32,92	3.947,32	0,07%	-2,06%
22/7/19	Sell	WBA	53,94	1,00	3.860,46	86,86	3.947,32		
22/7/19	Sell	MSFT	138,43	1,00	3.722,03	225,29	3.947,32		
22/7/19	Buy	VZ	55,50	4,00	3.944,03	3,29	3.947,32		
23/7/19	Prices Update				3.960,37	3,29	3.963,66	0,41%	-1,66%
23/7/19	Sell	NKE	86,70	4,00	3.613,57	350,09	3.963,66		
23/7/19	Buy	VZ	55,48	6,00	3.946,45	17,21	3.963,66		
24/7/19	Prices Update				3.950,90	17,21	3.968,11	0,11%	-1,55%
25/7/19	Prices Update				3.935,45	17,21	3.952,66	-0,39%	-1,93%
25/7/19	Sell	VZ	56,36	10,00	3.371,85	580,81	3.952,66		
25/7/19	Buy	INTC	52,16	11,00	3.945,61	7,05	3.952,66		
26/7/19	Prices Update				3.933,94	7,05	3.940,99	-0,30%	-2,22%
29/7/19	Prices Update				3.967,91	7,05	3.974,96	0,86%	-1,38%
30/7/19	Prices Update				3.959,45	7,05	3.966,50	-0,21%	-1,59%
31/7/19	Prices Update				3.902,25	7,05	3.909,30	-1,44%	-3,01%

6 Conclusion

6.1 Summary

In this dissertation, we dealt with the problem of predicting stock market data using Twitter data. As it was noted in the literature, sentiment data can have a significant positive impact on the forecasting ability of the models. However, many authors noted the noisy nature of these data. To redeem that, we proposed a new methodology. By using graphs, we obtained a daily importance measure for all of the users and we weighted their tweets with this measure.

Table 20 summarizes the results for the computed errors of all of the stocks. It is shown that the PageRank data set performed better than both the economic and the simple sentiment data set. Moreover, we were able to confirm that the most important feature, on the sentiment data, is the negative score of the tweet. However, we were not able to confirm which time lag is the most important, as the results were highly dependant on the feature.

Table 20: Best Data Set Per Ticker

	Ticker	AAPL	AXP	BA	CAT	CSCO	CVX	DIS	DOW	GS	HD	IBM	INTC	JNJ	JPM	KO	MCD	MMM	MRK	MSFT	NKE	PFE	PG	TRV	UNH	UTX	V	VZ	WBA	WMT	XOM
	PageRank				✓		✓	✓	✓		✓					✓		✓	✓		✓		✓	✓			✓		✓	✓	✓
Data Set	Sentiment		✓	✓		✓						✓			✓							✓									
	Economic	✓								✓			✓	✓			✓			✓					✓	✓		✓			

In terms of the models that were used, we tested five different ones. More specifically for each stock and for each data set, we estimated a Decision Tree, a Random Forest, an XGBoost, an LSTM, and a k-Nearest Neighbors. Table 21 presents a summarized version of the results in the PageRank data set. As we can see, the best model was XGBoost because it achieved the lowest scores at 13 stocks. Furthermore, it was the most robust model, having the lowest average error and the lowest standard deviation.

Table 21: Best Data Set Per Model on PageRank Data Set

	Ticker	AAPL	AXP	BA	CAT	CSCO	CVX	DIS	DOW	GS	HD	IBM	INTC	JNJ	JPM	KO	MCD	MMM	MRK	MSFT	NKE	PFE	PG	TRV	UNH	UTX	V	VZ	WBA	WMT	XOM
	Decision Tree	✓			✓				✓							✓															
	k-Nearest Neighbors				✓			✓					✓																		
Model	LSTM										✓	✓									✓		✓								✓
	Random Forest		✓			✓				✓					✓													✓	✓		
	XGBoost			✓			✓							✓			✓	✓	✓	✓		✓		✓	✓	✓	✓			✓	

Although PageRank's data set provided the best scores for most of the stocks, on the evaluation the economic data set proved the only profitable $(0,75\%)$. The other two data sets recorded of loss of $-3,05\%$ and $-3,01\%$ for Sentiment and PageRank.

6.2 Limitations

This study acts like a proof of concept that microblogging data can be a powerful feature in predicting stock market data, if and only if we can determine and distinguish the important ones. As the results showed, this is feasible but the required data pose a tremendous obstacle.

This is the biggest limitation this study has. Since all of our data come from the Twint library, and not from the official Twitter API, we could collect a specific amount of tweets. Moreover, this library is significantly slower than the official, thus it was very difficult to obtain data for a longer

period. We believe that if we had two years' worth of data and all the tweets per day, then our results would be significantly better.

Furthermore, the computing power needed for all of those tasks is another obstacle. In a machine with Intel Core i5-7600K @ 3.8GHZ (with 4 threads) and 32GB of RAM, it took two days to run all the models for all the data sets. The most demanding model was the XGBoost, which needed half an hour for each stock per data set.

Lastly, on the evaluation part, we choose a greedy strategy and not an optimal one. The optimal solution would require an extra module that would implement diversification according to Markowitz's Portfolio Theorem [79] and the extraction of optimal weights per stock. Moreover, every transaction should move us alongside the efficient frontier.

6.3 Further Research

There are a lot of aspects of our research we want to explore in the future. Firstly, we could estimate more models, such as SVM, which in the literature was used a lot. Another dimension we would like to explore is the economic variables we can choose. There are other useful economic variables that we should embed in our research. Moreover, we could expand our methodology to other financial instruments to explore the possibility that sentiment data can act as features on government and corporate bonds, or even on derivates. Lastly, as we observed in some models, there were cases where the mean squared error was low, but the fit between the actual and the predicted price was not good. Thus, it would be very helpful if we could define a new measure that can capture the fit better.

References

[1] E. F. Fama, "The behavior of stock-market prices," *The journal of Business*, vol. 38, no. 1, pp. 34–105, 1965.

[2] J. Bollen, H. Mao, and X.-J. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, pp. 1–8, Mar. 2011. arXiv: 1010.3003.

[3] A. Devenow and I. Welch, "Rational herding in financial economics," *European Economic Review*, vol. 40, no. 3-5, pp. 603–615, 1996.

[4] J. L. Treynor, "Jack treynor's' toward a theory of market value of risky assets'," *Available at SSRN 628187*, 1962.

[5] W. F. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *The journal of finance*, vol. 19, no. 3, pp. 425–442, 1964.

[6] E. F. Fama and K. R. French, "Common risk factors in the returns on stocks and bonds," *Journal of financial economics*, vol. 33, no. 1, pp. 3–56, 1993.

[7] E. W. K. See-To and Y. Yang, "Market sentiment dispersion and its effects on stock return and volatility," *Electronic Markets*, vol. 27, pp. 283–296, Aug. 2017.

[8] A. Alshahrani Hasan and A. C. Fong, "Sentiment Analysis Based Fuzzy Decision Platform for the Saudi Stock Market," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, (Rochester, MI), pp. 0023–0029, IEEE, May 2018.

[9] F. Wex, N. Widder, M. Liebmann, and D. Neumann, "Early Warning of Impending Oil Crises Using the Predictive Power of Online News Stories," in *2013 46th Hawaii International Conference on System Sciences*, (Wailea, HI, USA), pp. 1512–1521, IEEE, Jan. 2013.

[10] H. Alostad and H. Davulcu, "Directional Prediction of Stock Prices Using Breaking News on Twitter," in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, (Singapore, Singapore), pp. 523–530, IEEE, Dec. 2015.

[11] P. Hájek, "Combining bag-of-words and sentiment features of annual reports to predict abnormal stock returns," *Neural Computing and Applications*, vol. 29, pp. 343–358, Apr. 2018.

[12] M.-a. Mittermayer and G. Knolmayer, "NewsCATS: A News Categorization and Trading System," in *Sixth International Conference on Data Mining (ICDM'06)*, (Hong Kong, China), pp. 1002–1007, IEEE, Dec. 2006.

[13] W. Antweiler and M. Z. Frank, "Is all that talk just noise? the information content of internet stock message boards," *The Journal of finance*, vol. 59, no. 3, pp. 1259–1294, 2004.

[14] P. C. Tetlock, M. Saar-Tsechansky, and S. Macskassy, "More than words: Quantifying language to measure firms' fundamentals," *The Journal of Finance*, vol. 63, no. 3, pp. 1437–1467, 2008.

[15] N. Vlastakis and R. N. Markellos, "Information Demand and Stock Market Volatility," *Journal of Banking & Finance*, vol. 36, no. 6, pp. 1808–1821, 2012.

[16] T. O. Sprenger, A. Tumasjan, P. G. Sandner, and I. M. Welpe, "Tweets and Trades: the Information Content of Stock Microblogs: Tweets and Trades," *European Financial Management*, vol. 20, pp. 926–957, Nov. 2014.

[17] J. Park, H. Leung, and K. Ma, "Information fusion of stock prices and sentiment in social media using granger causality," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 614–619, IEEE, 2017.

[18] C. Wong and I.-Y. Ko, "Predictive Power of Public Emotions as Extracted from Daily News Articles on the Movements of Stock Market Indices," in *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, (Omaha, NE, USA), pp. 705–708, IEEE, Oct. 2016.

[19] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[20] D. Asteriou and S. G. Hall, *Applied econometrics*. Macmillan International Higher Education, 2015.

[21] X. Zhang, H. Fuehres, and P. A. Gloor, "Predicting Asset Value through Twitter Buzz," in *Advances in Collective Intelligence 2011* (J. Altmann, U. Baumöl, and B. J. Krämer, eds.), vol. 113, pp. 23–34, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[22] T. Rao and S. Srivastava, "Twitter Sentiment Analysis: How to Hedge Your Bets in the Stock Markets," in *State of the Art Applications of Social Network Analysis* (F. Can, T. Özyer, and F. Polat, eds.), pp. 227–247, Cham: Springer International Publishing, 2014.

[23] F. Wang, X. Li, and C. Dou, "Analysis of Financial News Impact on Stock Based on a Statistical Learning Method with News Density," in *2011 Fourth International Conference on Business Intelligence and Financial Engineering*, (Wuhan, Hubei, China), pp. 122–125, IEEE, Oct. 2011.

[24] R. Li, D. Fu, and Z. Zheng, "An Analysis of the Correlation between Internet Public Opinion and Stock Market," in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, (Changsha), pp. 150–153, IEEE, July 2017.

[25] G. Rachlin, M. Last, D. Alberg, and A. Kandel, "ADMIRAL: A Data Mining Based Financial Trading System," in *2007 IEEE Symposium on Computational Intelligence and Data Mining*, (Honolulu, HI, USA), pp. 720–725, IEEE, 2007.

[26] A. Mahajan, L. Dey, and S. M. Haque, "Mining Financial News for Major Events and Their Impacts on the Market," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, (Sydney, Australia), pp. 423–426, IEEE, Dec. 2008.

[27] P. Wei and N. Wang, "Wikipedia and Stock Return: Wikipedia Usage Pattern Helps to Predict the Individual Stock Movement," in *Proceedings of the 25th International Conference*

Companion on World Wide Web - WWW '16 Companion, (Montréal, Québec, Canada), pp. 591–594, ACM Press, 2016.

[28] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFin text system," *ACM Transactions on Information Systems*, vol. 27, pp. 1–19, Feb. 2009.

[29] N. Suman, P. K. Gupta, and P. Sharma, "Analysis of Stock Price Flow Based on Social Media Sentiments," in *2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)*, (Jammu), pp. 54–57, IEEE, Dec. 2017.

[30] Y. E. Cakra and B. Distiawan Trisedya, "Stock price prediction using linear regression based on sentiment analysis," in *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, (Depok, Indonesia), pp. 147–154, IEEE, Oct. 2015.

[31] M. Qasem, R. Thulasiram, and P. Thulasiram, "Twitter sentiment classification using machine learning techniques for stock markets," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (Kochi, India), pp. 834–840, IEEE, Aug. 2015.

[32] L. Bing, K. C. Chan, and C. Ou, "Public Sentiment Analysis in Twitter Data for Prediction of a Company's Stock Price Movements," in *2014 IEEE 11th International Conference on e-Business Engineering*, (Guangzhou, China), pp. 232–239, IEEE, Nov. 2014.

[33] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, (Paralakhemundi, Odisha, India), pp. 1345–1350, IEEE, Oct. 2016.

[34] R. Chiong, Z. Fan, Z. Hu, M. T. P. Adam, B. Lutz, and D. Neumann, "A sentiment analysis-based machine learning approach for financial market prediction via news disclosures," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion on - GECCO '18*, (Kyoto, Japan), pp. 278–279, ACM Press, 2018.

[35] R. Michael, S. W. Nick, and S. Padmini, "Stock chatter: Using stock sentiment to predict price direction," *Algorithmic Finance*, no. 3-4, pp. 169–196, 2013.

[36] S. Urolagin, "Text Mining of Tweet for Sentiment Classification and Association with Stock Prices," in *2017 International Conference on Computer and Applications (ICCA)*, (Doha, United Arab Emirates), pp. 384–388, IEEE, Sept. 2017.

[37] J. Kordonis, S. Symeonidis, and A. Arampatzis, "Stock Price Forecasting via Sentiment Analysis on Twitter," in *Proceedings of the 20th Pan-Hellenic Conference on Informatics - PCI '16*, (Patras, Greece), pp. 1–6, ACM Press, 2016.

[38] P. Chakraborty, U. S. Pria, M. R. A. H. Rony, and M. A. Majumdar, "Predicting stock movement using sentiment analysis of twitter feed," in *2017 6th International Conference on*

Informatics, Electronics and Vision & 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMHT), pp. 1–6, IEEE, 2017.

[39] T. Sun, J. Wang, P. Zhang, Y. Cao, B. Liu, and D. Wang, "Predicting Stock Price Returns Using Microblog Sentiment for Chinese Stock Market," in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, (Chengdu), pp. 87–96, IEEE, Aug. 2017.

[40] V. Plakandaras, T. Papadimitriou, P. Gogas, and K. Diamantaras, "Market sentiment and exchange rate directional forecasting," *Algorithmic Finance*, no. 1-2, pp. 69–79, 2015.

[41] G. Zhang, L. Xu, and Y. Xue, "Model and forecast stock market behavior integrating investor sentiment analysis and transaction data," *Cluster Computing*, vol. 20, pp. 789–803, Mar. 2017.

[42] S. K. Khatri and A. Srivastava, "Using sentimental analysis in prediction of stock market investment," in *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, (Noida, India), pp. 566–569, IEEE, Sept. 2016.

[43] P. Hajek and A. Barushka, "Integrating Sentiment Analysis and Topic Detection in Financial News for Stock Movement Prediction," in *Proceedings of the 2nd International Conference on Business and Information Management - ICBIM '18*, (Barcelona, Spain), pp. 158–162, ACM Press, 2018.

[44] Z. Wang, S.-B. Ho, and Z. Lin, "Stock Market Prediction Analysis by Incorporating Social and News Opinion and Sentiment," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, (Singapore, Singapore), pp. 1375–1380, IEEE, Nov. 2018.

[45] M. A. Asraf Roslan and M. H. Fazalul Rahiman, "Stock Prediction Using Sentiment Analysis in Twitter for Day Trader," in *2018 9th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, (Shah Alam, Malaysia), pp. 177–182, IEEE, Aug. 2018.

[46] B. J. Vanstone, A. Gepp, and G. Harris, "Do news and sentiment play a role in stock price prediction?," *Applied Intelligence*, Apr. 2019.

[47] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

[48] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[49] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes, "Correlating financial time series with micro-blogging activity," in *Proceedings of the fifth ACM international conference on Web search and data mining - WSDM '12*, (Seattle, Washington, USA), p. 513, ACM Press, 2012.

[50] Z.-H. Zhang, G.-P. Jiang, Y.-R. Song, L.-L. Xia, and Q. Chen, "An Improved Weighted LeaderRank Algorithm for Identifying Influential Spreaders in Complex Networks," in *22017 IEEE*

International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), (Guangzhou, China), pp. 748–751, IEEE, July 2017.

[51] Y. Bae and H. Lee, "A Sentiment Analysis of Audiences on Twitter: Who Is the Positive or Negative Audience of Popular Twitterers?," in *Convergence and Hybrid Information Technology* (G. Lee, D. Howard, and D. Ślęzak, eds.), vol. 6935, pp. 732–739, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[52] R. E. Bacha and T. Thi Zin, "Ranking of Influential users based on User-Tweet bipartite graph," in *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, (Singapore), pp. 97–101, IEEE, July 2018.

[53] B. G. Malkiel and E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[54] "7 technical indicators to build a trading toolkit."

[55] C. Mitchell, "Aroon oscillator definition and tactics," Jun 2019.

[56] J. Kuepper, "Timing trades with the commodity channel index," Sep 2019.

[57] I. Staff, "On-balance volume: The way to smart money," Aug 2019.

[58] D. Blystone, "Overbought or oversold? use the relative strength index to find out," May 2019.

[59] "Stohastic oscillator."

[60] D. Rousidis, P. Koukaras, and C. Tjortjis, "Social media prediction: A literature review," *Multimedia Tools and Applications*, 2019.

[61] P. Koukaras, C. Tjortjis, and D. Rousidis, "Social media types: introducing a data driven taxonomy," *Computing*, pp. 1–46.

[62] P. Koukaras and C. Tjortjis, "Social media analytics, types and methodology," in *Machine Learning Paradigms*, pp. 401–427, Springer, 2019.

[63] D. Beleveslis, C. Tjortjis, D. Psaradelis, and D. Nikoglou, "A hybrid method for sentiment analysis of election related tweets," in *2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pp. 1–6, IEEE, 2019.

[64] L. Oikonomou and C. Tjortjis, "A method for predicting the winner of the usa presidential elections using data extracted from twitter," in *2018 South-Eastern European Design Automation, Computer Engineering, Computer Networks and Society Media Conference (SEEDA_ CECNSM)*, pp. 1–8, IEEE, 2018.

[65] K. Golmohammadi and O. R. Zaiane, "Sentiment Analysis on Twitter to Improve Time Series Contextual Anomaly Detection for Detecting Stock Market Manipulation," in *Big Data Analytics and Knowledge Discovery* (L. Bellatreche and S. Chakravarthy, eds.), vol. 10440, pp. 327–342, Cham: Springer International Publishing, 2017.

[66] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[67] A. Papadopoulos, "Lecture notes in big data," March 2019.

[68] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[69] S. Sohangir, N. Petty, and D. Wang, "Financial Sentiment Lexicon Analysis," in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, (Laguna Hills, CA, USA), pp. 286–289, IEEE, Jan. 2018.

[70] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth international AAAI conference on weblogs and social media*, 2014.

[71] P. Tzirakis and C. Tjortjis, "T3c: improving a decision tree classification algorithm's interval splits on continuous attributes," *Advances in Data Analysis and Classification*, vol. 11, no. 2, pp. 353–370, 2017.

[72] C. Tjortjis and J. Keane, "T3: a classification algorithm for data mining," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 50–55, Springer, 2002.

[73] C. Tjorjis, "Lecture notes in data mining," March 2019.

[74] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

[75] K. Diamantaras, "Lecture notes in advanced machine learning," March 2019.

[76] S. Hochreiter, "Investigations on dynamic neural networks," *Diploma, Technical University*, vol. 91, no. 1, 1991.

[77] Y. Bengio, P. Simard, P. Frasconi, *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[78] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[79] H. M. Markowitz, "Foundations of portfolio theory," *The journal of finance*, vol. 46, no. 2, pp. 469–477, 1991.

Appendices

A Appendix I

Descriptive Statistics

Table 22: Features' Means Per Ticker

Ticker	Negative Score	Neutral Score	Positive Score	Compound Score	Polarity	Subjectivity	Pagerank	Close	Volume	SlowD	SlowK	RSI	CCI	OBV	Aroon Down	Aroon Up
AAPL	41,29	983,37	120,48	219,56	138,32	414,50	0,19	183,97	32.048.103,80	55,29	55,25	50,50	17,85	23.275.501.321,20	46,58	55,83
AXP	0,78	40,46	2,70	5,98	2,30	10,44	0,00	111,82	3.541.903,80	59,10	58,82	56,00	63,80	382.744.840,22	27,41	71,56
BA	16,32	302,43	23,98	25,78	26,97	100,71	0,04	363,78	5.556.747,28	55,05	54,72	50,03	-11,26	475.442.559,78	51,41	46,18
CAT	2,58	79,08	5,19	9,66	5,80	24,70	0,01	131,55	4.677.307,07	53,54	53,36	49,73	15,84	148.805.290,76	43,32	51,38
CSCO	1,62	74,69	6,31	15,92	9,15	26,22	0,01	51,23	22.345.147,83	58,36	58,04	55,64	42,41	3.293.837.147,83	31,96	65,05
CVX	1,74	64,10	5,37	11,61	7,62	23,15	0,01	119,19	7.002.596,74	54,33	54,07	51,54	20,01	713.897.246,74	39,80	61,85
DIS	7,15	217,44	18,98	35,85	23,89	74,33	0,03	122,81	10.337.052,72	56,28	56,26	55,46	38,31	1.166.837.350,54	33,84	57,30
DOW	1,90	47,14	2,75	2,74	3,91	16,26	0,01	50,46	4.877.817,74	44,30	43,75	50,27	-44,79	13.472.153,23	55,70	23,33
GS	6,37	168,22	18,04	35,10	14,64	64,84	0,02	196,27	3.212.979,35	53,25	53,19	48,78	4,24	-358.986.339,13	51,59	51,20
HD	1,96	70,80	6,02	12,52	7,32	23,37	0,01	190,66	4.594.721,20	59,02	59,04	52,95	31,58	230.379.951,63	41,99	56,32
IBM	2,51	91,47	7,21	15,43	7,88	27,97	0,01	133,24	4.314.543,48	59,13	59,36	51,73	25,79	68.068.448,91	42,30	59,40
INTC	5,48	158,60	11,35	20,17	16,74	54,72	0,02	49,55	24.358.534,78	56,80	56,59	51,35	47,11	2.393.134.138,59	39,08	65,85
JNJ	4,75	110,33	9,20	14,63	11,72	38,83	0,01	136,87	7.894.702,17	56,38	56,30	50,83	16,80	334.630.695,11	41,07	69,33
JPM	5,36	173,00	13,17	26,80	14,09	60,93	0,02	107,24	13.571.994,57	54,68	54,49	50,92	21,88	-2.026.105.947,83	43,50	55,29
KO	1,77	70,42	5,44	11,26	6,96	22,67	0,01	48,67	14.129.502,72	57,67	57,46	54,36	54,59	2.145.114.945,11	29,80	68,57
MCD	3,50	78,33	7,25	10,01	9,33	27,96	0,01	191,12	3.264.045,65	58,67	58,47	58,23	65,76	1.389.596.610,87	26,85	74,55
MMM	1,72	56,82	3,58	6,81	3,90	17,80	0,01	191,38	2.733.838,59	54,20	53,92	46,96	16,58	359.788.935,33	45,00	56,96
MRK	2,94	88,79	7,41	12,78	11,27	30,63	0,02	78,99	11.552.931,52	56,90	56,79	55,03	38,20	322.458.873,37	36,97	68,51
MSFT	7,51	279,32	23,97	55,32	34,56	97,23	0,07	118,02	29.014.478,26	61,84	61,80	56,03	61,34	-2.331.783.783,70	33,08	65,51
NKE	3,00	78,98	6,26	9,78	6,76	25,14	0,02	81,80	6.774.944,02	56,30	56,17	52,43	26,68	1.302.158.864,13	46,36	61,72
PFE	2,94	96,42	7,58	15,72	11,01	35,16	0,01	42,45	24.510.370,11	54,69	54,40	50,19	-5,53	-3.261.354.246,74	45,94	49,84
PG	1,77	69,00	6,13	14,16	10,24	26,30	0,01	101,58	8.724.636,41	62,76	62,78	59,67	80,14	527.802.416,30	28,51	82,61
TRV	0,42	17,40	1,21	2,45	1,42	4,58	0,00	135,63	1.427.075,54	62,17	61,96	56,34	65,70	343.210.404,35	27,72	71,72
UNH	1,83	63,50	5,56	12,33	6,63	21,32	0,01	251,80	4.619.586,96	53,73	53,77	49,33	3,53	711.291.041,85	42,59	54,95
UTX	0,96	40,95	6,60	15,49	3,98	13,75	0,00	126,35	4.408.949,46	58,00	57,78	50,80	16,08	175.094.483,70	40,34	47,64
V	5,32	134,29	10,44	16,51	12,47	42,23	0,01	153,49	8.386.393,48	62,83	62,74	57,19	67,87	2.915.909.511,41	33,57	75,60
VZ	2,18	80,87	6,28	14,22	8,67	27,94	0,00	57,23	14.915.396,20	51,85	51,97	51,56	-2,93	85.617.007,61	44,57	43,42
WBA	1,30	37,75	2,23	3,24	2,61	10,26	0,01	63,72	6.224.611,41	50,93	50,82	44,25	-32,79	18.647.442,93	55,51	46,61
WMT	3,57	115,03	9,94	19,99	12,12	38,82	0,02	100,91	7.125.435,33	56,95	56,78	55,46	49,37	202.790.409,24	36,20	63,17
XOM	2,93	99,68	7,65	16,60	8,92	35,77	0,01	76,58	12.480.637,50	52,71	52,44	48,43	-5,29	296.418.583,15	48,06	52,88
Overall Average	4,78	132,96	12,28	22,95	14,71	47,95	0,02	125,61	10.287.566,19	56,26	56,11	52,40	26,30	1.177.147.345,22	40,35	59,14

Table 23: Features' Standard Deviation

Ticker	Negative Score	Neutral Score	Positive Score	Compound Score	Polarity	Subjectivity	Pagerank	Close	Volume	SlowD	SlowK	RSI	CCI	OBV	Aroon Down	Aroon Up
AAPL	44,77	569,55	63,68	109,52	61,89	224,09	0,14	17,87	13.799.345,85	23,02	24,98	9,87	127,48	185.451.019,68	40,28	41,36
AXP	1,06	25,70	1,85	3,83	2,24	6,47	0,01	9,26	1.342.625,24	22,93	24,93	6,94	98,96	27.296.417,22	30,36	32,72
BA	23,99	312,45	24,48	20,97	26,76	106,79	0,06	29,59	4.696.750,73	25,66	27,47	7,75	111,77	20.580.005,16	34,19	34,78
CAT	7,09	84,99	6,14	7,12	4,87	27,87	0,03	5,87	1.912.330,68	22,75	24,69	5,22	98,00	17.653.152,45	33,86	35,37
CSCO	1,34	55,27	4,92	10,91	7,84	18,47	0,03	4,95	10.469.152,17	24,87	26,89	7,37	97,50	189.008.782,93	31,52	34,57
CVX	1,56	47,65	4,29	9,98	6,67	16,00	0,02	5,04	3.974.790,06	21,34	23,75	5,37	101,76	22.983.814,13	34,95	34,82
DIS	8,15	197,42	20,55	33,32	25,38	71,90	0,06	13,06	6.793.395,19	23,84	25,91	8,04	108,96	57.177.428,08	32,18	34,86
DOW	1,39	21,19	1,66	2,79	2,22	5,98	0,02	1,76	1.758.817,00	25,69	28,45	3,37	76,35	15.610.184,22	35,77	26,19
GS	5,13	86,85	7,20	14,34	8,72	28,57	0,03	13,43	1.762.033,14	25,09	26,51	8,68	116,32	20.241.218,71	37,26	35,74
HD	2,63	59,36	4,77	7,98	5,82	19,14	0,02	14,15	2.005.200,70	23,73	25,57	8,34	100,97	24.403.045,03	34,59	37,47
IBM	2,42	74,88	6,71	13,93	8,53	22,60	0,03	9,85	2.374.012,15	22,53	24,32	9,04	113,17	36.340.315,52	34,95	38,28
INTC	6,45	104,87	7,61	11,21	9,87	33,97	0,04	3,59	10.150.729,87	22,01	24,05	7,75	109,05	118.536.159,50	35,44	35,47
JNJ	7,20	99,51	8,46	11,34	7,42	34,56	0,04	5,53	5.379.057,38	22,33	24,43	7,10	127,37	32.134.904,83	34,39	29,76
JPM	5,36	94,52	9,38	18,54	10,88	32,78	0,05	5,40	5.270.643,63	23,10	25,01	6,62	107,16	55.973.154,05	32,48	36,03
KO	2,30	58,81	4,79	9,28	7,17	20,00	0,02	2,24	6.604.642,61	21,03	23,41	7,00	119,75	33.323.745,59	32,15	33,68
MCD	7,01	50,83	5,67	17,79	9,41	22,87	0,02	11,81	1.298.688,47	22,33	24,37	5,92	97,59	15.262.408,89	29,94	31,65
MMM	3,83	52,63	3,34	5,76	4,20	17,29	0,02	16,41	1.417.856,15	26,34	28,44	8,86	110,40	17.719.695,41	36,14	37,24
MRK	1,70	35,88	3,92	7,98	5,81	12,91	0,04	3,52	4.971.900,30	21,92	23,96	6,27	109,54	58.767.840,49	32,02	33,45
MSFT	4,37	123,32	12,67	26,65	19,15	45,53	0,09	12,46	12.238.597,80	21,48	23,59	6,86	97,94	263.789.466,52	33,39	33,83
NKE	4,91	81,71	6,81	9,41	8,32	26,51	0,04	5,28	2.924.514,62	24,25	26,19	6,16	98,53	34.473.766,04	34,99	34,45
PFE	2,18	54,52	4,29	9,01	6,25	17,18	0,03	1,30	10.321.562,05	19,75	22,07	4,77	109,10	120.342.866,70	32,66	35,53
PG	1,38	53,11	5,51	13,81	9,56	20,89	0,02	7,77	3.597.060,09	21,84	23,86	4,60	87,59	57.802.897,69	27,17	22,76
TRV	0,55	12,76	0,91	2,05	1,69	3,48	0,00	11,25	499.736,02	22,37	24,41	7,67	95,47	15.889.820,85	31,97	34,13
UNH	2,83	61,13	5,63	10,29	5,55	19,85	0,02	13,63	2.959.121,08	24,03	26,22	5,61	116,29	20.851.299,93	32,43	34,12
UTX	1,16	45,72	6,76	15,74	5,76	16,02	0,01	8,76	2.585.491,02	23,81	25,52	8,48	110,12	18.212.555,07	35,47	37,91
V	3,69	64,42	5,64	11,94	8,43	22,21	0,02	15,44	3.478.492,61	21,27	23,23	7,32	90,16	80.722.809,22	30,26	31,74
VZ	1,54	37,75	3,57	8,94	4,50	13,39	0,01	1,56	5.642.301,35	20,40	22,86	5,34	99,00	78.252.832,59	34,19	34,20
WBA	3,31	46,87	3,25	3,69	3,24	14,05	0,03	10,81	3.556.750,70	23,86	25,80	10,21	127,74	37.386.544,00	36,09	37,05
WMT	2,70	72,47	7,18	13,67	8,98	23,87	0,04	6,43	3.027.413,60	24,81	26,18	7,83	102,66	55.415.197,12	34,42	33,02
XOM	1,96	41,68	4,67	12,23	7,86	18,62	0,02	3,69	4.785.022,51	24,42	26,02	7,16	114,13	52.304.654,30	37,40	37,16
Overall Average	5,47	90,93	8,54	15,13	10,17	32,13	0,03	9,06	4.719.934,49	23,09	25,10	7,05	106,03	59.463.600,07	33,76	34,31

Table 24: Features' Maximum Values

Ticker	Negative Score	Neutral Score	Positive Score	Compound Score	Polarity	Subjectivity	Pagerank	Close	Volume	SlowD	SlowK	RSI	CCI	OBV	Aroon Down	Aroon Up
AAPL	540,02	6.179,58	546,39	622,79	429,03	2.316,46	0,85	213,04	95.744.600,00	94,75	97,22	69,13	256,98	23.627.603.900,00	100,00	100,00
AXP	9,89	226,34	14,77	20,97	17,69	52,10	0,05	128,57	11.005.300,00	93,64	96,75	66,88	246,70	416.893.300,00	100,00	100,00
BA	194,49	2.446,91	199,60	168,05	200,30	849,07	0,41	440,62	36.922.600,00	94,17	97,85	70,26	248,97	525.396.900,00	100,00	100,00
CAT	87,75	982,16	71,09	50,66	39,54	336,58	0,24	143,36	17.421.400,00	92,43	96,80	60,45	248,27	185.026.800,00	100,00	100,00
CSCO	12,59	500,90	40,52	78,46	68,63	147,36	0,27	58,05	103.123.400,00	93,68	97,72	70,21	185,49	3.563.844.600,00	100,00	100,00
CVX	10,81	499,49	44,02	97,62	59,98	154,93	0,15	126,68	42.693.700,00	92,87	94,76	62,30	203,44	757.043.100,00	100,00	100,00
DIS	98,12	2.262,66	248,22	380,32	305,51	865,64	0,48	146,39	65.253.500,00	93,50	97,98	76,06	419,59	1.299.385.000,00	100,00	100,00
DOW	8,29	120,82	9,07	8,34	9,27	33,81	0,07	53,46	13.932.700,00	93,50	96,17	55,74	129,17	36.603.000,00	100,00	100,00
GS	36,04	804,00	76,02	130,84	94,73	280,84	0,22	231,65	15.194.200,00	96,35	97,40	65,17	243,17	-317.781.900,00	100,00	100,00
HD	24,91	526,41	38,69	60,85	38,70	165,29	0,22	218,70	14.274.000,00	94,53	97,77	67,06	237,71	282.208.200,00	100,00	100,00
IBM	17,66	760,39	69,95	143,28	90,15	232,24	0,24	151,36	22.063.700,00	93,29	94,41	67,73	260,05	132.852.800,00	100,00	100,00
INTC	55,08	833,90	53,50	72,06	60,13	260,23	0,32	58,82	86.455.700,00	91,85	95,00	67,02	249,57	2.639.385.500,00	100,00	100,00
JNJ	82,07	956,72	80,21	76,77	47,57	348,87	0,40	147,84	58.140.200,00	94,39	94,66	65,21	282,97	387.440.200,00	100,00	100,00
JPM	66,72	900,60	84,90	183,46	105,74	304,96	0,38	116,83	41.313.900,00	91,13	94,68	66,28	244,94	-1.879.887.700,00	100,00	100,00
KO	25,85	526,11	46,32	96,37	73,27	178,25	0,24	54,33	58.905.400,00	95,24	96,25	68,22	356,12	2.260.968.400,00	100,00	100,00
MCD	65,95	402,13	37,89	70,82	89,75	183,08	0,09	215,91	10.440.100,00	96,09	97,15	67,92	334,94	1.414.775.500,00	100,00	100,00
MMM	46,92	571,07	32,00	33,00	25,94	190,20	0,15	219,50	14.646.200,00	94,12	98,01	61,34	216,37	391.866.800,00	100,00	100,00
MRK	11,65	287,48	28,87	54,00	39,35	92,06	0,40	86,90	44.546.600,00	92,53	93,39	67,31	217,67	449.322.900,00	100,00	100,00
MSFT	32,86	1.061,95	106,43	195,74	148,52	382,85	0,65	141,34	111.242.100,00	94,19	97,11	72,67	301,37	-1.816.607.400,00	100,00	100,00
NKE	34,79	573,19	50,68	80,75	72,40	194,74	0,32	89,48	28.487.900,00	92,87	98,15	63,75	190,86	1.360.936.100,00	100,00	100,00
PFE	19,59	474,51	28,10	50,90	38,59	125,58	0,20	46,23	90.834.600,00	92,62	95,88	61,03	252,52	-3.000.149.600,00	100,00	100,00
PG	10,27	470,40	42,32	99,23	68,72	156,82	0,20	120,41	30.802.700,00	94,78	96,25	68,27	245,34	643.413.400,00	100,00	100,00
TRV	4,51	101,34	5,77	9,76	12,54	26,33	0,03	154,83	4.523.600,00	95,23	96,98	67,95	246,99	367.227.500,00	100,00	100,00
UNH	27,99	624,67	57,35	89,99	51,68	203,45	0,18	286,33	27.361.400,00	93,54	95,76	62,17	371,59	750.621.700,00	100,00	100,00
UTX	10,32	402,55	60,41	135,36	32,20	144,87	0,08	142,61	16.869.700,00	92,99	96,44	69,39	340,48	206.203.700,00	100,00	100,00
V	19,08	389,66	30,26	63,93	47,67	115,72	0,22	183,69	25.448.600,00	93,22	97,55	68,91	259,49	3.049.353.100,00	100,00	100,00
VZ	14,42	354,82	23,75	42,98	26,74	107,73	0,07	60,88	55.406.300,00	94,00	95,58	66,61	208,06	224.306.500,00	100,00	100,00
WBA	40,91	506,25	37,83	34,20	24,94	159,21	0,23	85,69	36.877.800,00	94,74	98,29	67,66	289,07	101.208.200,00	100,00	100,00
WMT	20,92	746,25	79,49	156,14	95,66	247,38	0,34	114,98	20.697.300,00	95,03	96,67	73,02	313,71	316.647.500,00	100,00	100,00
XOM	18,50	408,74	39,47	86,14	38,44	123,92	0,15	83,38	47.287.300,00	94,16	96,20	61,09	212,13	385.686.200,00	100,00	100,00
Overall Average	54,97	863,40	76,13	113,13	81,78	299,35	0,26	144,06	41.597.216,67	93,85	96,49	66,56	260,46	1.292.059.806,67	100,00	100,00

Table 25: Features' Minimum Values

Ticker	Negative Score	Neutral Score	Positive Score	Compound Score	Polarity	Subjectivity	Pagerank	Close	Volume	SlowD	SlowK	RSI	CCI	OBV	Aroon Down	Aroon Up
AAPL	6,66	288,37	20,77	33,42	26,14	93,00	0,00	142,19	11.362.000,00	7,20	3,37	30,97	-290,34	22.842.145.400,00	0,00	0,00
AXP	0,00	3,00	0,00	-2,33	-1,33	0,13	0,00	89,50	938.800,00	3,79	1,94	32,40	-308,86	316.199.200,00	0,00	0,00
BA	1,82	43,15	2,62	-24,38	0,27	18,11	0,00	294,16	1.599.400,00	8,85	5,60	37,01	-286,49	432.068.900,00	0,00	0,00
CAT	0,05	14,89	0,54	-7,84	-0,76	4,23	0,00	116,95	2.378.300,00	10,83	5,76	37,96	-280,75	97.118.400,00	0,00	0,00
CSCO	0,00	4,48	0,52	0,94	0,98	1,17	0,00	40,28	8.809.700,00	8,18	3,74	37,76	-258,14	2.949.257.100,00	0,00	0,00
CVX	0,04	5,75	0,05	-2,03	-0,55	1,57	0,00	100,99	2.725.800,00	7,49	6,08	33,29	-280,88	654.345.000,00	0,00	0,00
DIS	0,84	25,25	0,91	2,01	2,43	6,12	0,00	100,35	3.242.300,00	7,23	5,05	33,59	-294,88	994.030.100,00	0,00	0,00
DOW	0,13	19,62	0,91	-3,91	0,35	4,48	0,00	46,76	2.479.400,00	8,58	5,99	43,38	-156,46	-14.842.600,00	0,00	0,00
GS	0,49	20,87	2,64	-0,12	1,22	7,86	0,00	156,35	978.800,00	8,19	4,91	26,81	-212,45	-415.605.000,00	0,00	0,00
HD	0,00	8,00	0,00	0,00	0,00	0,37	0,00	158,14	2.172.600,00	11,06	5,78	34,23	-219,26	184.222.000,00	0,00	0,00
IBM	0,00	12,24	0,76	-2,27	-0,53	2,78	0,00	107,57	1.849.800,00	11,41	6,55	33,24	-288,19	-7.766.400,00	0,00	0,00
INTC	1,18	22,38	1,22	-2,94	2,41	7,87	0,00	43,46	8.906.900,00	3,94	1,20	32,20	-285,69	2.151.687.600,00	0,00	0,00
JNJ	0,16	9,97	0,87	-6,46	0,68	1,86	0,00	122,84	3.404.900,00	7,10	5,53	32,73	-391,20	259.950.800,00	0,00	0,00
JPM	0,27	8,87	0,86	0,91	0,26	2,37	0,00	92,14	6.488.400,00	7,54	5,03	31,42	-248,82	-2.135.277.300,00	0,00	0,00
KO	0,00	9,79	0,21	0,05	-0,43	2,23	0,00	44,69	4.792.500,00	7,75	1,34	39,06	-295,22	2.056.381.500,00	0,00	0,00
MCD	0,32	13,25	1,10	-127,52	0,52	1,54	0,00	170,28	1.559.200,00	7,01	4,95	42,95	-339,81	1.345.224.400,00	0,00	0,00
MMM	0,00	6,90	0,00	-11,87	-1,82	0,63	0,00	159,75	999.900,00	5,14	4,11	27,01	-289,77	326.300.400,00	0,00	0,00
MRK	0,13	17,81	1,33	-5,33	-0,87	7,53	0,00	71,15	3.521.800,00	4,20	2,69	33,92	-337,20	195.765.500,00	0,00	0,00
MSFT	0,06	31,56	2,38	6,60	2,00	7,68	0,00	94,13	13.629.300,00	17,18	10,89	38,29	-257,17	-2.741.579.900,00	0,00	0,00
NKE	0,00	4,58	0,00	-2,01	-2,54	0,72	0,00	67,53	2.582.300,00	10,17	8,30	37,12	-219,96	1.226.574.200,00	0,00	0,00
PFE	0,40	35,93	2,16	-2,14	-0,17	7,58	0,00	38,79	8.390.900,00	11,67	5,69	35,38	-455,13	-3.506.527.400,00	0,00	0,00
PG	0,17	18,30	0,35	-0,41	0,38	4,34	0,00	87,36	4.018.800,00	10,73	9,68	45,01	-302,84	424.473.400,00	0,00	0,00
TRV	0,00	1,00	0,00	-2,55	-1,26	0,00	0,00	112,63	405.800,00	9,63	3,02	33,84	-187,84	311.338.100,00	0,00	0,00
UNH	0,00	3,26	0,59	-0,08	-0,49	0,54	0,00	216,84	1.159.300,00	6,81	3,55	35,51	-368,11	651.821.900,00	0,00	0,00
UTX	0,00	4,81	0,33	-1,03	-2,87	0,07	0,00	102,06	1.605.600,00	6,54	2,85	27,27	-236,70	123.525.300,00	0,00	0,00
V	0,13	7,71	0,16	-9,56	-0,70	2,19	0,00	121,73	3.676.000,00	14,13	9,29	37,71	-239,99	2.766.744.300,00	0,00	0,00
VZ	0,03	3,97	0,00	-0,13	-1,39	0,07	0,00	53,05	6.938.500,00	11,92	7,85	39,87	-289,59	-117.429.400,00	0,00	0,00
WBA	0,00	1,78	0,14	-2,70	-2,23	0,92	0,00	49,34	2.716.300,00	3,62	2,88	29,78	-454,64	-46.596.500,00	0,00	0,00
WMT	0,33	14,23	1,08	1,18	0,62	2,23	0,00	85,82	2.688.500,00	7,01	6,23	34,20	-167,23	77.794.900,00	0,00	0,00
XOM	0,10	7,44	0,46	-4,56	-1,90	3,07	0,00	65,51	5.246.100,00	5,74	3,24	29,98	-277,57	150.354.100,00	0,00	0,00
Overall Average	0,44	22,30	1,43	-5,90	0,61	6,44	0,00	105,08	4.042.263,33	8,35	5,10	34,80	-284,04	1.051.723.266,67	0,00	0,00

Table 26: Features' 1st Quantile Values

Ticker	Negative Score	Neutral Score	Positive Score	Compound Score	Polarity	Subjectivity	Pagerank	Close	Volume	SlowD	SlowK	RSI	CCI	OBV	Aroon Down	Aroon Up
AAPL	22,22	728,49	75,59	132,11	99,48	293,09	0,09	171,03	21.863.975,00	38,04	34,62	40,96	-99,63	23.166.347.200,00	5,83	10,00
AXP	0,35	29,78	1,71	3,32	0,93	7,43	0,00	106,50	2.685.700,00	40,82	39,33	52,42	12,66	363.638.875,00	3,33	55,83
BA	6,00	155,56	11,82	12,62	12,41	48,84	0,01	347,13	3.263.050,00	31,49	28,71	44,68	-83,69	461.281.175,00	16,67	13,33
CAT	0,93	50,64	2,74	4,91	2,76	14,83	0,00	126,75	3.435.375,00	33,08	29,92	46,02	-45,19	142.954.325,00	10,00	16,67
CSCO	0,90	47,93	3,37	8,89	4,74	15,29	0,00	46,84	17.336.450,00	35,13	32,76	50,61	-21,55	3.119.321.300,00	3,33	39,17
CVX	0,98	42,10	3,06	5,62	3,14	13,56	0,00	116,07	4.965.500,00	37,05	34,65	47,57	-28,57	695.550.825,00	6,67	32,50
DIS	3,46	120,46	10,16	20,12	13,39	42,17	0,00	111,87	6.973.625,00	36,68	34,03	49,18	-16,81	1.135.288.050,00	6,67	23,33
DOW	1,03	34,80	1,63	1,03	2,22	11,96	0,00	49,13	3.940.775,00	21,75	20,48	47,53	-111,43	-1.805.600,00	20,83	3,33
GS	3,70	123,86	14,12	27,20	10,12	50,33	0,00	191,29	2.177.825,00	29,07	28,30	43,50	-109,47	-368.139.025,00	13,33	16,67
HD	0,83	45,64	3,36	6,65	3,71	13,94	0,00	179,72	3.314.225,00	38,44	34,88	47,08	-55,23	212.835.375,00	6,67	20,00
IBM	1,38	59,90	4,12	8,05	3,69	18,15	0,00	123,50	2.908.650,00	40,18	39,06	43,98	-61,90	27.379.025,00	6,67	19,17
INTC	2,87	111,88	7,56	13,56	9,95	36,61	0,00	47,08	17.847.500,00	43,60	41,61	47,78	-24,14	2.296.241.375,00	6,67	35,83
JNJ	2,06	65,77	4,43	6,64	6,56	20,46	0,00	132,33	5.562.100,00	39,15	36,54	44,69	-52,02	321.452.625,00	10,00	50,00
JPM	3,42	129,61	8,67	16,18	8,90	45,14	0,00	103,36	10.111.375,00	36,17	33,79	47,36	-34,51	-2.061.541.825,00	15,83	19,17
KO	0,91	48,03	3,26	6,31	3,11	13,57	0,00	46,94	10.507.525,00	41,29	38,28	48,81	-34,10	2.122.970.975,00	3,33	43,33
MCD	1,49	53,33	4,28	7,02	4,87	17,06	0,00	182,14	2.374.300,00	41,95	38,68	53,89	25,31	1.382.173.325,00	3,33	62,50
MMM	0,63	37,52	1,69	2,97	1,04	10,35	0,00	174,66	1.917.850,00	31,03	26,71	42,29	-54,54	340.938.700,00	10,00	20,00
MRK	1,80	68,10	5,05	7,46	7,44	23,22	0,00	75,96	8.605.675,00	39,48	37,33	50,39	-42,44	283.269.600,00	6,67	45,83
MSFT	5,07	222,71	16,97	37,42	22,20	70,65	0,01	106,90	21.460.575,00	40,01	41,06	50,49	9,81	-2.580.370.575,00	3,33	42,50
NKE	1,22	42,75	2,99	4,47	2,64	12,98	0,00	77,76	5.008.300,00	35,31	33,25	48,23	-42,17	1.271.629.025,00	10,00	33,33
PFE	1,59	68,23	4,48	8,85	7,07	23,83	0,00	41,80	18.708.400,00	40,31	37,69	47,49	-60,33	-3.366.866.375,00	16,67	16,67
PG	0,85	41,05	2,85	5,68	4,49	12,12	0,00	93,55	6.574.975,00	47,71	44,96	56,83	48,58	477.426.500,00	3,33	73,33
TRV	0,10	10,74	0,61	1,03	0,37	2,50	0,00	126,52	1.098.275,00	46,84	44,89	51,50	33,56	329.106.425,00	3,33	52,50
UNH	0,66	35,86	2,73	5,87	3,90	11,45	0,00	242,59	3.158.000,00	33,93	31,71	45,63	-51,31	693.203.900,00	13,33	22,50
UTX	0,29	20,94	2,24	5,27	0,41	5,77	0,00	122,33	2.754.375,00	38,66	36,72	45,28	-62,82	169.049.950,00	6,67	10,00
V	1,79	82,47	6,00	8,09	6,69	25,98	0,00	139,66	6.085.075,00	44,05	43,12	50,78	32,63	2.841.665.425,00	6,67	63,33
VZ	1,31	58,64	3,75	8,06	5,37	18,59	0,00	56,39	11.601.275,00	36,32	34,39	47,85	-72,31	26.311.275,00	10,00	10,00
WBA	0,35	20,98	1,15	1,40	0,88	5,01	0,00	53,89	4.288.925,00	32,58	29,95	35,33	-108,77	-8.320.750,00	20,00	10,00
WMT	2,01	83,31	6,67	13,17	7,41	27,37	0,00	96,90	5.077.175,00	36,43	35,94	50,81	-15,88	163.416.850,00	3,33	35,83
XOM	1,67	71,75	4,31	6,99	3,77	24,03	0,00	74,52	9.426.875,00	33,33	30,10	43,62	-94,67	262.637.025,00	10,00	13,33
Overall Average	2,40	90,43	7,38	13,23	8,79	31,21	0,00	118,84	7.501.123,33	37,33	35,12	47,42	-40,70	1.130.634.832,50	8,75	30,33

Table 27: Features' 2nd Quantile Values

Ticker	Negative Score	Neutral Score	Positive Score	Compound Score	Polarity	Subjectivity	Pagerank	Close	Volume	SlowD	SlowK	RSI	CCI	OBV	Aroon Down	Aroon Up
AAPL	32,90	897,21	124,82	214,13	136,05	416,63	0,16	186,32	28.596.050,00	57,62	57,03	51,85	71,02	23.295.640.450,00	38,33	73,33
AXP	0,55	36,37	2,26	5,06	1,81	9,43	0,00	110,48	3.289.700,00	63,03	62,46	57,24	85,32	396.924.750,00	15,00	86,67
BA	9,59	207,89	17,21	22,52	19,72	69,90	0,02	361,52	4.487.550,00	57,72	57,55	48,01	-20,23	469.979.350,00	56,67	40,00
CAT	1,39	63,31	3,89	8,56	4,69	20,23	0,00	132,05	4.261.600,00	53,52	53,68	50,18	13,16	151.697.350,00	40,00	56,67
CSCO	1,36	63,94	5,36	13,40	6,72	21,63	0,00	52,66	20.255.800,00	63,95	61,25	55,04	74,13	3.374.172.700,00	21,67	76,67
CVX	1,41	56,69	4,38	9,68	6,16	19,57	0,00	119,75	6.153.000,00	53,81	54,61	51,46	36,43	713.284.200,00	33,33	70,00
DIS	5,28	165,78	14,48	28,80	18,31	60,41	0,01	115,47	8.526.350,00	59,72	59,18	54,60	49,08	1.159.428.650,00	23,33	63,33
DOW	1,56	41,20	2,34	2,56	3,38	15,04	0,00	50,39	4.591.000,00	39,24	38,40	50,02	-58,90	18.869.050,00	63,33	13,33
GS	4,94	145,60	16,59	33,02	13,04	58,58	0,01	197,16	2.753.300,00	57,47	57,36	51,19	26,34	-357.627.050,00	51,67	53,33
HD	1,40	59,11	5,10	11,49	5,26	18,65	0,00	189,77	4.094.550,00	63,45	63,21	52,80	63,25	227.117.150,00	40,00	63,33
IBM	2,06	76,82	5,87	12,53	6,10	24,29	0,00	136,00	3.732.550,00	64,66	65,39	52,90	50,68	82.399.450,00	36,67	75,00
INTC	3,78	131,32	9,18	18,77	15,29	47,06	0,00	48,58	21.891.900,00	59,59	57,63	51,75	70,78	2.397.289.750,00	30,00	83,33
JNJ	3,08	87,49	7,45	12,18	9,89	31,27	0,00	137,77	6.665.250,00	60,20	60,67	52,54	50,55	341.753.900,00	33,33	80,00
JPM	4,47	154,72	10,80	23,44	11,84	54,04	0,01	107,28	12.514.100,00	57,69	56,24	50,96	31,63	-2.033.883.200,00	40,00	63,33
KO	1,26	58,28	4,68	9,24	5,37	18,60	0,00	48,65	12.300.400,00	59,66	59,45	55,88	71,85	2.145.675.050,00	16,67	83,33
MCD	2,01	65,51	5,99	10,59	7,05	22,51	0,00	187,49	3.022.750,00	63,92	65,55	59,51	89,33	1.391.564.350,00	15,00	90,00
MMM	1,00	47,42	3,07	6,39	3,05	14,63	0,00	193,23	2.360.500,00	60,07	58,80	48,03	33,11	360.621.100,00	41,67	70,00
MRK	2,48	84,12	6,83	11,65	10,86	29,01	0,00	79,29	10.451.150,00	60,02	60,13	56,04	71,47	319.526.850,00	30,00	83,33
MSFT	6,92	265,99	22,15	51,06	30,45	88,16	0,04	117,28	26.627.400,00	66,75	65,39	56,78	81,64	-2.344.323.800,00	20,00	73,33
NKE	1,81	58,51	4,46	7,70	4,80	18,27	0,00	83,42	6.260.000,00	59,29	59,35	52,51	42,81	1.295.870.100,00	50,00	71,67
PFE	2,50	87,62	7,07	14,93	10,08	33,09	0,00	42,51	22.546.150,00	56,25	58,16	50,48	-7,40	-3.267.588.000,00	45,00	48,33
PG	1,50	56,58	4,73	10,65	7,15	19,33	0,00	101,93	7.848.100,00	66,97	67,52	60,46	97,76	544.436.550,00	23,33	93,33
TRV	0,27	14,82	0,98	2,11	0,99	3,76	0,00	134,64	1.349.950,00	67,35	68,33	58,44	96,20	344.272.550,00	13,33	90,00
UNH	1,30	47,83	4,27	9,84	5,71	16,35	0,00	248,42	3.952.600,00	56,22	56,98	49,88	17,22	709.110.400,00	36,67	63,33
UTX	0,75	30,81	5,20	11,70	1,97	9,60	0,00	127,77	3.543.200,00	62,19	61,21	51,41	18,84	178.329.950,00	33,33	41,67
V	5,43	136,85	10,24	14,90	10,21	39,76	0,00	154,41	7.566.950,00	66,50	66,40	58,52	97,26	2.906.306.650,00	23,33	90,00
VZ	1,83	71,92	5,17	11,30	8,48	24,12	0,00	57,21	13.856.750,00	52,32	54,26	50,79	11,71	98.150.450,00	40,00	38,33
WBA	0,67	26,36	1,49	2,98	1,95	7,61	0,00	61,91	5.500.100,00	50,08	51,44	43,94	-34,76	14.459.000,00	61,67	43,33
WMT	3,13	103,60	8,86	17,69	11,29	35,98	0,00	99,54	6.324.400,00	59,89	60,56	53,92	46,66	195.320.400,00	26,67	70,00
XOM	2,69	95,21	6,59	12,71	6,37	31,19	0,00	76,56	11.359.200,00	52,24	53,84	47,44	-4,73	293.156.700,00	46,67	55,00
Overall Average	3,64	114,63	11,05	20,72	12,80	42,62	0,01	125,32	9.222.743,33	59,05	59,07	52,82	42,41	1.180.731.160,00	34,89	66,78

Table 28: Features' 3ND Quantile Values

Ticker	Negative Score	Neutral Score	Positive Score	Compound Score	Polarity	Subjectivity	Pagerank	Close	Volume	SlowD	SlowK	RSI	CCI	OBV	Aroon Down	Aroon Up
AAPL	45,33	1.126,28	146,29	292,59	170,11	479,35	0,27	199,76	38.833.500,00	73,89	76,32	58,28	112,50	23.394.966.050,00	90,00	96,67
AXP	0,90	44,99	3,16	7,67	3,16	11,81	0,00	119,14	4.053.675,00	78,75	79,43	60,78	129,34	404.953.525,00	46,67	96,67
BA	15,94	337,72	26,40	32,94	30,97	110,41	0,04	376,54	6.031.625,00	79,38	79,84	53,69	96,72	488.606.125,00	80,83	80,83
CAT	2,31	78,25	5,74	12,49	7,57	25,74	0,00	135,81	5.418.425,00	72,60	72,56	53,87	93,88	159.275.525,00	73,33	83,33
CSCO	2,02	81,76	7,92	20,65	11,67	32,40	0,01	55,86	24.301.050,00	80,62	82,22	61,96	115,97	3.464.493.500,00	53,33	96,67
CVX	2,01	69,80	6,81	16,19	10,34	28,61	0,00	123,31	7.750.625,00	73,22	74,27	55,68	93,67	734.063.750,00	70,83	93,33
DIS	8,54	242,64	20,70	40,58	25,76	77,47	0,04	135,08	11.077.700,00	78,39	80,89	61,12	98,15	1.204.587.525,00	53,33	90,00
DOW	2,30	55,05	3,30	4,74	5,18	19,72	0,01	51,97	5.400.775,00	62,52	66,00	53,37	9,45	27.179.300,00	89,17	32,50
GS	7,20	184,53	19,98	40,80	16,51	71,90	0,02	202,56	3.699.125,00	75,53	76,20	54,70	97,96	-349.421.475,00	90,00	86,67
HD	2,03	76,41	7,41	16,60	9,99	27,93	0,00	202,21	5.123.075,00	81,29	82,90	60,21	106,94	242.290.275,00	73,33	93,33
IBM	2,70	94,73	8,00	19,24	9,24	30,06	0,01	139,76	4.989.875,00	79,76	80,96	59,79	107,71	94.772.000,00	74,17	96,67
INTC	5,79	165,93	12,50	24,05	19,60	57,96	0,01	52,44	28.026.125,00	74,69	76,98	57,11	125,60	2.495.476.700,00	70,00	96,67
JNJ	5,43	113,52	10,70	21,34	15,28	45,77	0,01	140,41	8.510.525,00	75,02	76,80	55,60	110,07	357.714.450,00	73,33	93,33
JPM	5,92	185,29	15,15	33,10	16,01	66,18	0,02	111,30	15.230.525,00	75,09	78,11	54,60	100,39	-1.995.410.175,00	70,00	90,00
KO	1,97	72,06	5,99	13,69	9,11	25,27	0,01	49,86	16.009.375,00	74,26	77,98	59,72	135,87	2.164.648.925,00	46,67	96,67
MCD	2,92	84,43	7,88	14,58	10,13	30,38	0,01	199,22	3.847.525,00	75,64	78,02	62,98	121,40	1.401.561.050,00	40,83	100,00
MMM	1,58	60,39	4,54	9,82	5,66	20,69	0,00	205,63	3.156.825,00	78,07	80,24	53,67	104,54	374.763.550,00	76,67	90,83
MRK	3,85	100,37	8,92	16,39	13,63	37,30	0,01	81,66	13.263.300,00	74,64	77,08	60,09	125,14	361.172.175,00	63,33	96,67
MSFT	8,63	312,12	27,92	69,36	41,39	113,94	0,09	127,93	33.532.025,00	80,81	83,62	61,53	116,69	-2.126.859.825,00	63,33	96,67
NKE	2,57	76,28	6,80	12,38	7,70	26,92	0,01	85,71	7.761.650,00	77,96	79,15	57,52	107,17	1.335.181.000,00	76,67	93,33
PFE	3,53	106,68	9,14	20,01	14,03	41,71	0,01	43,14	26.951.050,00	71,49	72,43	52,52	71,63	-3.195.659.725,00	73,33	86,67
PG	2,39	83,67	7,93	18,67	13,83	35,95	0,01	106,69	9.826.125,00	80,88	82,41	62,82	131,87	570.359.475,00	43,33	100,00
TRV	0,55	20,37	1,77	3,70	1,92	5,75	0,00	147,06	1.680.300,00	80,33	81,99	62,39	125,82	358.635.075,00	46,67	96,67
UNH	1,93	74,33	6,44	15,57	8,09	24,84	0,00	264,73	4.993.850,00	74,33	77,37	52,99	75,40	730.614.900,00	70,00	84,17
UTX	1,28	46,27	8,88	22,81	4,95	16,02	0,00	132,28	5.101.975,00	79,35	81,83	56,26	113,52	186.777.000,00	73,33	90,00
V	8,15	175,00	14,06	24,05	15,91	56,46	0,01	163,75	9.734.775,00	81,27	85,36	63,41	128,00	2.987.899.900,00	56,67	100,00
VZ	2,69	93,86	8,54	19,46	11,83	34,56	0,00	58,27	16.635.975,00	69,24	71,46	54,58	63,42	152.887.075,00	76,67	76,67
WBA	1,21	37,25	2,34	4,32	3,13	10,26	0,00	71,71	6.598.825,00	71,42	72,13	50,22	51,10	35.561.975,00	90,00	84,17
WMT	4,18	126,55	11,50	24,19	14,94	44,79	0,02	103,38	8.371.950,00	78,17	80,47	61,12	115,67	229.572.325,00	63,33	96,67
XOM	3,85	120,26	10,31	25,36	11,52	39,94	0,01	79,79	14.352.450,00	74,86	74,68	54,71	90,18	346.126.125,00	84,17	90,00
Overall Average	5,32	148,23	14,57	29,91	17,97	55,00	0,02	132,23	11.675.486,67	76,12	77,99	57,58	102,53	1.221.226.269,17	68,44	90,19

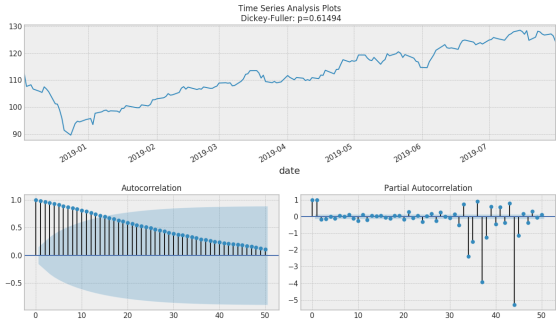

Figure 53: American Express EDA

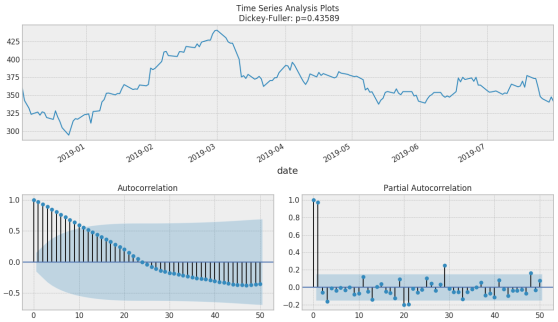

Figure 54: Boeing EDA

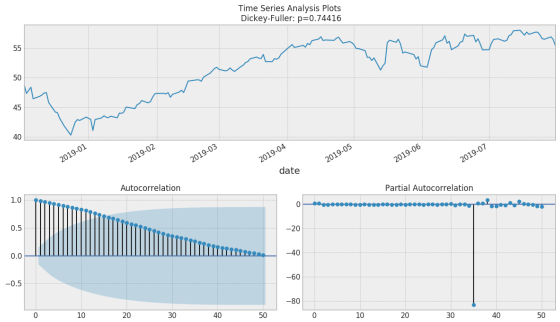

Figure 55: Cisco EDA

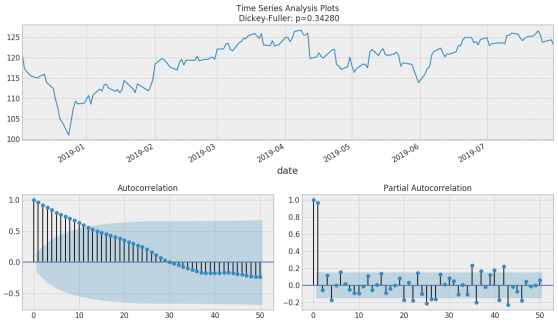

Figure 56: CVS Health EDA

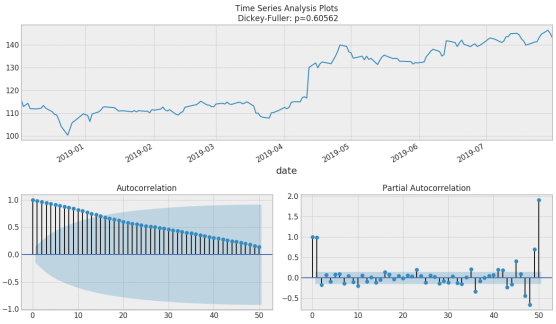

Figure 57: Walt Disney EDA

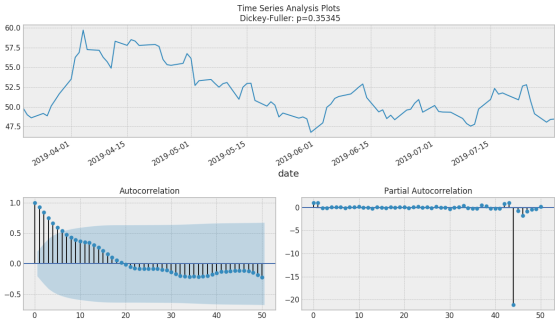

Figure 58: Dow EDA

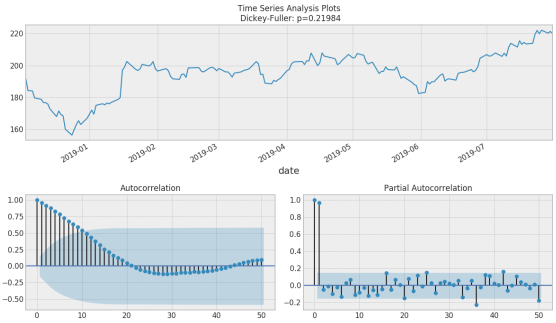

Figure 59: Goldman Sachs EDA

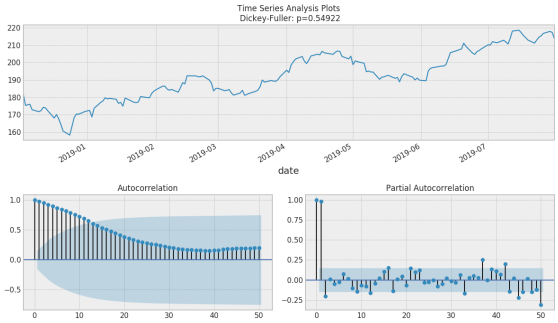

Figure 60: Home Dipot EDA

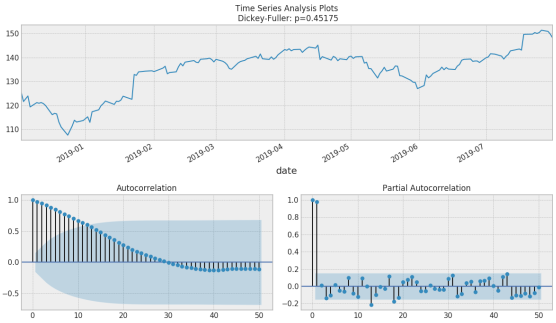

Figure 61: IBM EDA

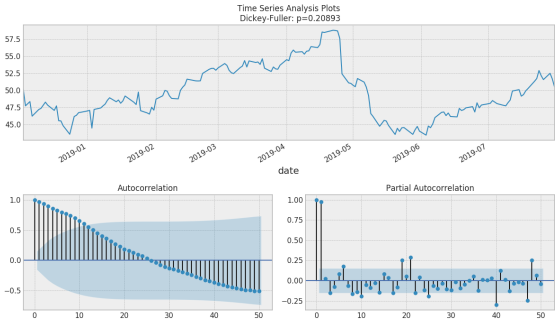

Figure 62: Intel EDA

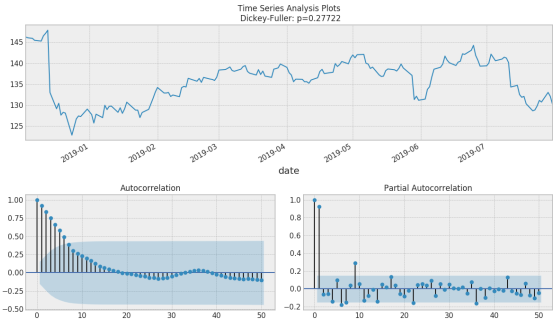

Figure 63: Johnson & Johnson EDA

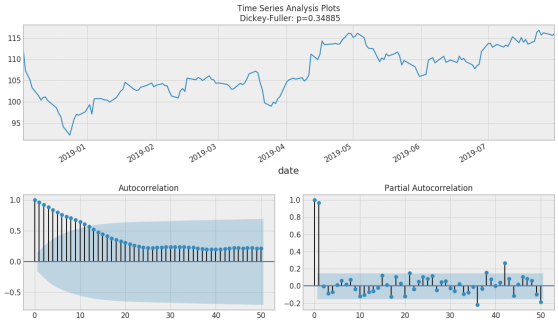

Figure 64: JPMorgan Chase EDA

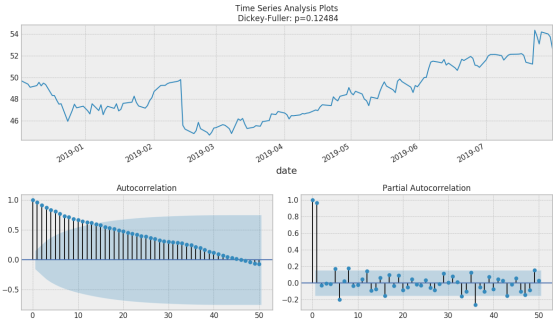

Figure 65: Coca-Cola EDA

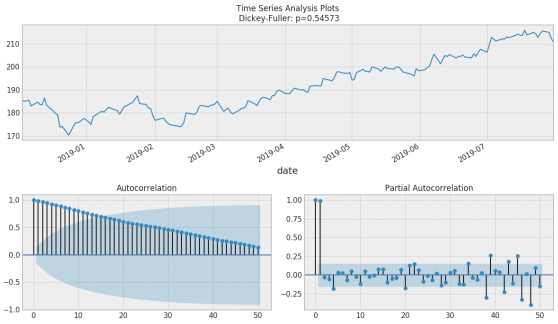

Figure 66: Mcdonald's EDA

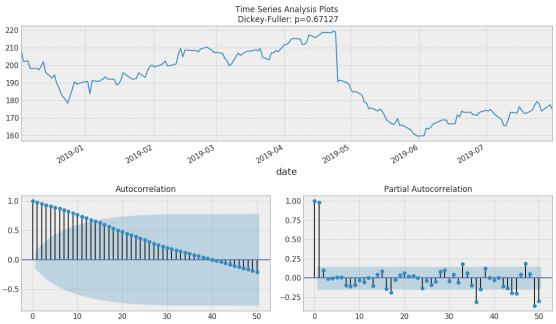

Figure 67: 3M EDA

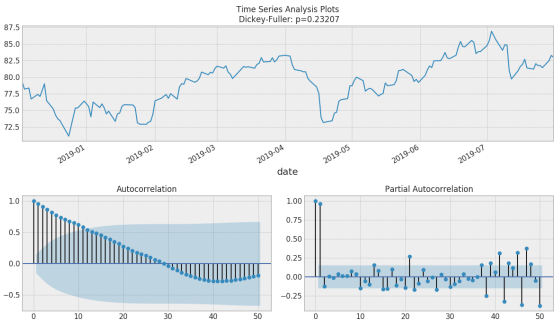

Figure 68: Merck & Co EDA

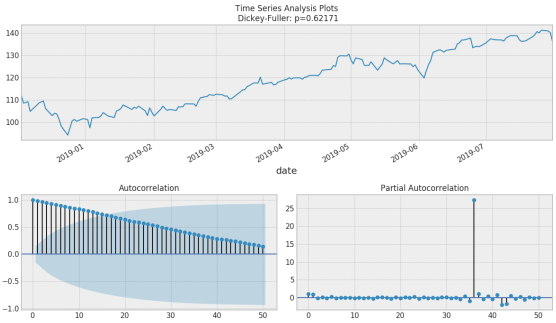

Figure 69: Microsoft EDA

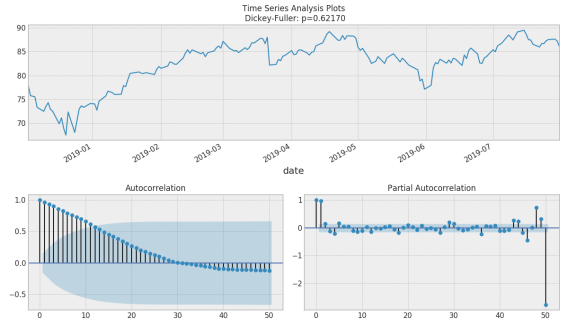

Figure 70: Nike EDA

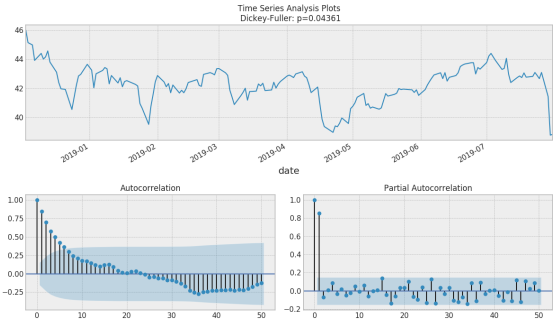

Figure 71: Pfizer EDA

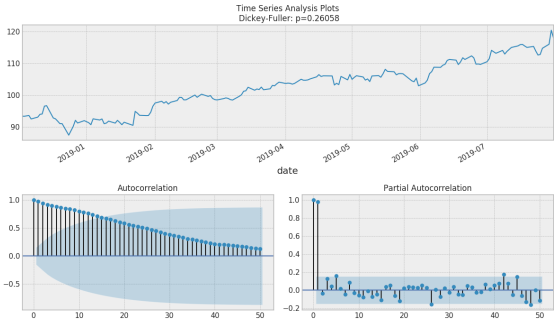

Figure 72: Procter & Gamble EDA

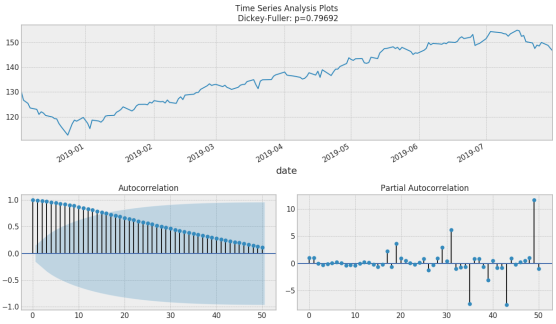

Figure 73: Travelers Companies EDA

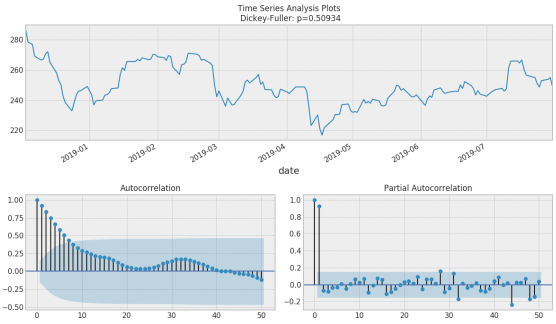

Figure 74: UnitedHealth Group EDA

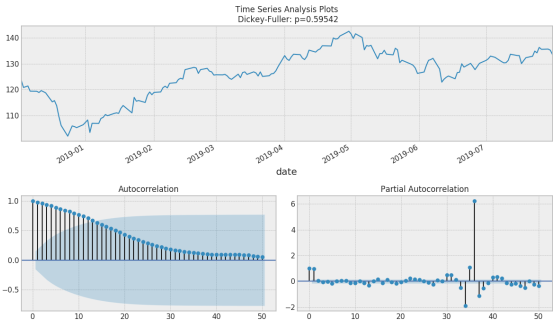

Figure 75: United Technologies EDA

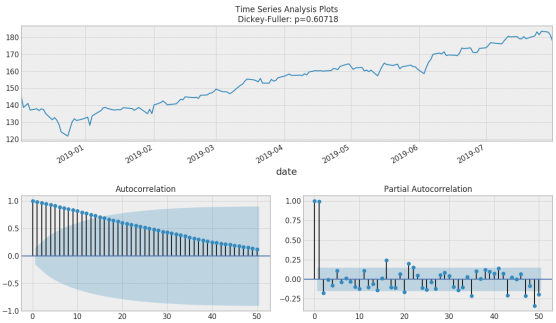

Figure 76: Visa EDA

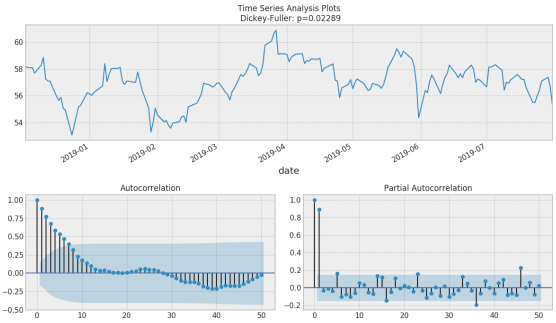

Figure 77: Verizon Communications EDA

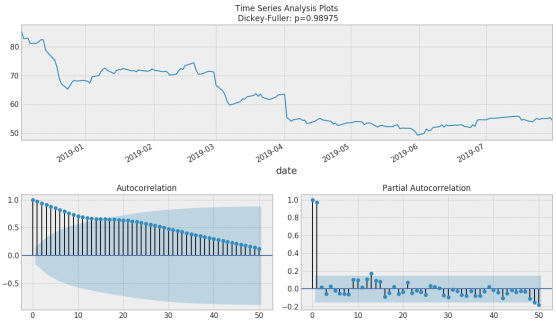

Figure 78: Walgreens Boots Alliance EDA

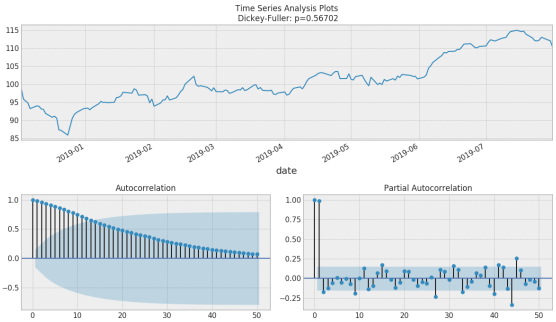

Figure 79: Walmart EDA

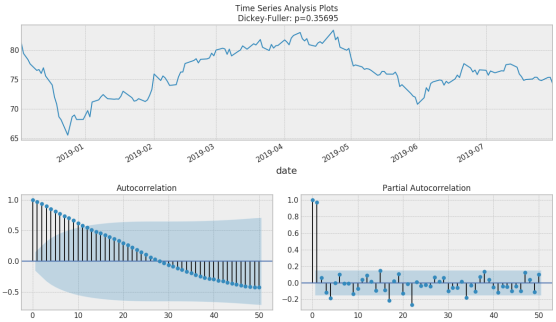

Figure 80: Exxon Mobil EDA