19738

Bsc Computer Science & Artificial Intelligence (GG47)

Department of Informatics

On the effect of various NLP techniques, data and ML models in stock price movement prediction

---

Dr John Carroll

May 2021

Statement of Originality

This report is submitted as part requirement for the degree of Computer Science & Artificial Intelligence at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

Acknowledgements

I would like to thank Dr/Professor John Carroll for his immeasurable contribution, patience and guidance throughout this research. His support has been invaluable, even through the exceptionally difficult times the world has been through during the pandemia of the Coronavirus. His willingness to have regular meetings and discuss ideas while providing very useful insights has been crucial to my detailed understanding of the problem in hand.

Professional considerations

This project meets all the ethical standards governing the conduct of computing professionals in the UK set by the *BCS -The Chartered Institute for IT*[1]. The four key principles rightfully addressed are sections 1 (Public Interest), 2 (Professional Competence and Integrity), 3 (Duty to Relevant Authority) and 4 (Duty to the Profession).

The project heavily relies on input secondary data in order to create representative models using AI techniques. The intended dataset is a combination of a prepared dataset from REFERENCE TO ORIGINAL PAPER and extrapolated with data coming from twitter.

The work undertaken in this project is within the profession of the author as it utilised an array of expertise provided by the author's Computer Science and Artificial Intelligence undergraduate degree. The author also makes it a priority to independently find answers to fill gaps in their knowledge and ensures that any further research and references provided in the report are up to date.

In addition, the value of the extensive insights possessed by the author's supervisor contributes to maintaining awareness of different technological updates, developments and standards that may be relevant and important to the investigation. The author obtains other views and criticisms from the supervisor, which are respected and always considered.

Ethical Review

Since this project involves scientific research based around literature and computer modelling, there are no major ethical issues to be considered.

Although this project uses twitter data, the users personal identity is not collected. It is important to note that the final models can be validated and tested via *machine learning* techniques without human participants. Given that no personal or confidential data is used, an ethical review is not required.

[1] Full code of conduct available at: https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/

Abstract

This project documents the detailed exploration, design, development and implementation of natural language processing techniques as well as machine learning models used to predict stock price movements (UP, STAY, DOWN) 2-days after the release of an 8K report (reports which U.S companies must file with the SEC to announce major events that shareholders should know about). The project analyses the impact of using different types of sentiment data classified as speculative (where opinions rather than facts tend to be shared, e.g. twitter) and factual (where proven information, generally unbiased is shared, e.g. 8K reports). In addition, the study also looks at the effect of using different feature representations of the sentiment data. Several architectures (pipelines) are designed with the purpose of comparing between different approaches in feature extraction and model selection. The dataset used is built on top of an already provided dataset by Heeyoung Lee et al. (2014). The final accuracy obtained by the models is compared against the best performing model of the latter paper.

Ultimately, this research's objective is to improve the predictive accuracy score of a random walk (50%) and the baseline model by testing different models, architectures, feature representations and even data selection.

Ultimately, the results obtained indicated that using 8K polarity (negative or positive) scores through a hybrid method of a domain-specific word bank and an open-domain sentiment lexicon bank was together with other standard numerical data more powerful to create a more accurate ensemble model (56%) than the baseline (55.5%). The combination of using speculative and factual sentiment data resulted in the model with the lowest accuracy 49%, however, the results were potentially poor due to a great reduction in training data as a consequence of inconsistencies between the prepared dataset and the extended corpus obtained from twitter.

Table of Contents

Introduction

Keywords: 8K reports, ensemble methods, tf-idf values, polarity scores, factual sentiment data, non-factual sentiment data.

Background

Over the last 10 years, artificial intelligence has become the go-to field to solve all problems associated with predictions and data. The wide range of algorithms and techniques being developed have enabled computers' capacity to capture meaning and trends within structured and unstructured data. Such advancements in the field are opening up a broad range of opportunities for everyone whether it's big corporations or professional individuals to use these tools for their own benefit.

The principal idea is that AI is fundamentally implemented to make as accurate as possible predictions of a future event, for which users can be prepared to react to. This paper tackles the widely yet unresolved problem of stock trends predictions. Stock price and trend predictions are complex problems to solve due to the highly dynamic and fast changing patterns which seem random across all stock price movements in the world. Most of the models are incapable of yielding very accurate results since stock price movement is a highly non-linear process.

Prediction of stock trends has long been an intriguing topic and is extensively studied by researchers from different fields. However, the general lack of high accuracy reflecting a random walk, is the main drawback of most of these studies. The majority of the deployed models rely on quantitative numerical data only (prices and technical indicators) ignoring sentiments.

At the same time, the scarce number of studies that use a combination of numerical and sentiment data tend to not differentiate between speculative (e.g. non-factual statements such as opinions made via twitter) and factual data (e.g. 8K company reports) and use only one of both sources. Ultimately, we find that the existing solutions either lack conclusive data, quantity of data, appropriate models or all of these combined for the problem set, leaving a big gap of research unexplored.

The complexity of stock behaviour is ultimately driven by supply and demand which derives from many key factors such as latest earnings statements, announcement of dividends by a company, information about new products, prediction by financial experts, social, economic and political

conditions, media hype, speculations, etc. Professionals have used this influential market knowledge to research methods to implement artificial intelligence in this problem.

Most of the research done has been primarily focused on machine and deep learning, subfields of artificial intelligence, with an average accuracy below 51%, close to the probability of a human guess. However, it is important to note that minor increases, as little as 0.1% in the predictability accuracy, can be exploited to generate major profits using the law of large numbers[2]. Hence, it is of great interest to continue investigating other approaches to AI's implementation in the stock market.

Data in the stock market

The type of data required to train machine learning models to predict different events in the stock market depends on the strategy applied to make profits. If the strategy is to identify a future price of a stock given its price history, then a sequential set of data may be deemed as required in order to understand the markets' behaviour, however if the strategy is to anticipate the average number of traders after an important event occurs in a company, the data required might not necessarily be sequential, as important events do not occur on a constant basis. Ultimately, this leads to the selection of different algorithms depending on the type of data obtained, for example a LSTM Recurrent Neural Network is usually more convenient for sequential data than disordered data.

The vital part of machine learning is the dataset used. The dataset should be as concrete as possible because a little change in the data can perpetuate massive changes in the outcome. Clearly, this is an era in which text and data mining tools and techniques can be employed. With the advent of online news sites such as Google Finance, Yahoo Finance and SEC, financial news is delivered in real-time, streaming format.

Early research on stock market prediction was created on the Efficient Market Hypothesis and the random walk theory. These early models show that stock prices cannot be predicted because they are driven by new information rather than current/past prices. Therefore, the stock market price will follow random fluctuations, and its prediction accuracy cannot exceed 50%, which is obviously always present in any random event (including coin toss). Previous research suggests that sentiment index has an impact on securities and that emotions enhance the predictability of candidate factors of returns.

[2] Law of large numbers: "Law of Large Numbers." *SpringerReference*, doi:10.1007/springerreference_60988.

A real life example of sentiment data impacting the stock market occurred in the beginning of 2021 when the WallStreetBets, a thriving group on the popular social-media discussion forum site Reddit, executed a plan to go against sizable hedge funds which seemed to be shorting GameStop Inc. stock (borrowing the stock in order to sell it with the view that its price would fall, at which point they would buy the stock and lock in healthy profits). The price of the stock rose from 20$ at the end of 2020 to almost double its value ($40) in only three weeks, five days later to $76 and on the morning of the 28th to a staggering peak of $483. This unusual market behaviour was caused as a result of a group of individual investors coming together through the platform of Reddit and deciding to go against usual market logic, and ultimately causing a domino effect on other investors outside the platform that used the sentiment data to analyse the situation and be attracted to the opportunity of making profits as the price was rapidly rising.

In addition, a study on the 'Importance of Text Analysis for Stock Prediction' from Heeyoung Lee et al. (2014) demonstrated that using text coming from financial events reported in 8K documents (reports which U.S companies must file with the SEC to announce major events that shareholders should know about) helped to boost prediction accuracy over 10% (relative) over their baseline models to forecast companies' stock price changes.

The latter study provided a link where the prepared dataset could be found. The authors established a corpus that can be used to investigate the importance of text analysis to stock price changes. The corpus aligns descriptions of financial events reported in 8K documents with corresponding stock prices, which helps to develop a stock price prediction system that combines financial and textual information. The corpus is publicly available. Using this corpus, the author's proved that merging text information is indeed very important, especially in the short term (two days after the event).

Structure of the report

The rest of the paper is organized as follows. The next subsections of the 'Introduction' provide a brief review of the related work on stock price movement prediction using NLP and ML and explicit definitions of the project's aim and proposed solution. The next section is the 'Methodology' which describes step by step the architectures, models and data used in the investigation. Next, is the 'Experimental Results' which discusses the results obtained from each model and discusses the conclusions derived from these ones. Finally, the last sections of the paper evaluate the work done in this research and conclude the paper with a personal view on the overall project and further work on the problem.

Related work

Over the last decade substantial studies have been carried out on the topic of machine learning and natural language processing in the stock market. Sidra Mehtab and Jaydip Sen (2019) augmented their predictive models by integrating a sentiment analysis module on twitter data to correlate the public sentiment of stock prices with the market sentiment, obtaining final and proved that public sentiments in the social media serve as a very significant input in predictive model building for stock price movement. Zhou et al. (2016) demonstrated a correlation between emotions expressed in tweets and the stock market. Michael Hahsler et al. (2015) proposed a sentiment metric, called NewsSentiment, utilizing the count of positive and negative polarity words as a measure of the sentiment of the overall news corpus to show that the time variation of the tool presented a very strong correlation with the actual stock price movement. Yauheniya Shynkevich et al. (2015) studied how the results of financial forecasting can be improved when news articles with different levels of relevance to the target stock are used simultaneously. Xie et al. (2013) introduced tree representations of information in news, Bar-Haim et al. (2011) focused on identifying better expert investors, Leinweber and Sisk (2011) studied the effect of news and the time needed to process the news in event-driven trading. Kogan et al. (2009) proposed a method that predicts risk based on financial reports. Engelberg (2008) shows that linguistic information, perhaps because of cognitive load in processing, has greater long-term predictability for asset prices than quantitative information. While this literature provides an important background, few previous results show improvements from textual information on predicting the impact of financial events on top of quantitative features like earnings surprise, which are known to be very predictive.

Project Aim & Proposed Solution

Following the previous assumptions with regards to the implementation of AI in the stock market, this investigation attempts to tap into the almost unexplored gap in the literature by approaching the problem through different subfields of AI and Computer Science such as Data Science, Machine Learning, Deep Learning and Natural Language Processing.

The project aims to build systems and models for short term (daily) traders where the stock price change can be predicted within 2 days horizon after the release of an 8K document with and without taking the combination of numerical and sentiment, specifically speculative data (tweets) into account. The 2-day prediction horizon is selected in order to be able to compare against the baseline models from Heeyoung Lee et al. (2014) which stated that predicting price change movements 2 days after the release of 8K documents resulted in the best performing models as newspapers and articles reflect not only new information about companies, but also the perspectives and opinions of third parties, which may require more time for the market to digest.

The study also intends to investigate and collect appropriate sentiment data from online sources ranging from factual to speculative (e.g. twitter) that can be used along with the prepared dataset in order to form a larger and more conclusive dataset. In addition, several machine learning and natural language processing techniques are used in an attempt to obtain an optimal performing system. Some of the algorithms considered are: *naïve bayes classifier (NB), random forest (RF- ensemble method), support vector machine (SVM)* and *sentiment analysis using lexicons*. Furthermore, some of the weaknesses found in previous research studies as well as their 'future research' suggested are discussed and considered throughout the paper for implementation purposes.

The investigation uses various different pipeline architectures in order to demonstrate the impact of using different features and data quantity in the accuracy of the models. The results obtained by Heeyoung Lee et al. (2014) are used as the baseline to compare against the results obtained from the models used in this study. In order to obtain features from the 8K documents, tf-idf values (a statistical measure that evaluates how relevant a word is to a document in a collection of documents) as well as polarity scores are used as features in some models to evaluate their importance. In addition, all the architectures also account for negation in semantics, since many expressions could be mistakenly represented by opposing meanings as a result of semantic opposition. For example, the phrase 'The company did not perform exceptionally well this year' can potentially be interpreted as neutral or positive if the words 'exceptionally well' are considered without their negated sense.

	Architecture 1	Architecture 2	Architecture 4	Architecture 3
Non-linguistic features	✓	✓	✓	✓
tf-idf values from 8K docs	✓	□	✓	✓
Polarity scores from 8K docs	□	✓	✓	✓
Polarity scores from tweets	□	□	□	✓

Table 1: Different features used for each architecture in the study.

The list of features displayed above, represent the variables that are changed and used for comparison purposes throughout the experiment, however they do not represent all the features used for each model.

Methodology

Architectures

In order to evaluate the impact of using different features with different representations on the overall prediction accuracy, different models and system (pipelines) variants within each architecture are tested. The main differences between the architectures are the feature extraction methods of the 8K reports and tweets as well as their different representations. The purpose of using different variants of System 1 (shown in **Appendix System 1 ~ Variant 1, Variant 2 and Variant 3**) is to select the method that uses the most meaningful features to the final model when predicting price change movements before taking tweets into account.

Ultimately, the end goal is to create three systems. The first system creates a set of changing features representing the 8K reports. The seconds system creates features that remain the same throughout the investigation (**Appendix System 2**) which are derived from the non-linguistic (numerical) data and adds the event categories extracted from the 8K reports. The third system (**Appendix System 3**) implements polarity scores extracted from tweets aligned and related to companies during the first day of the 8K report release. Finally, the features generated by all Systems are combined to be used to predict the price change movements.

There are 4 architectures. The first 3, use different variants of System 1, and the final one uses the System 1 variant that performed the best in the previous 3 architectures, with an additional System 3.

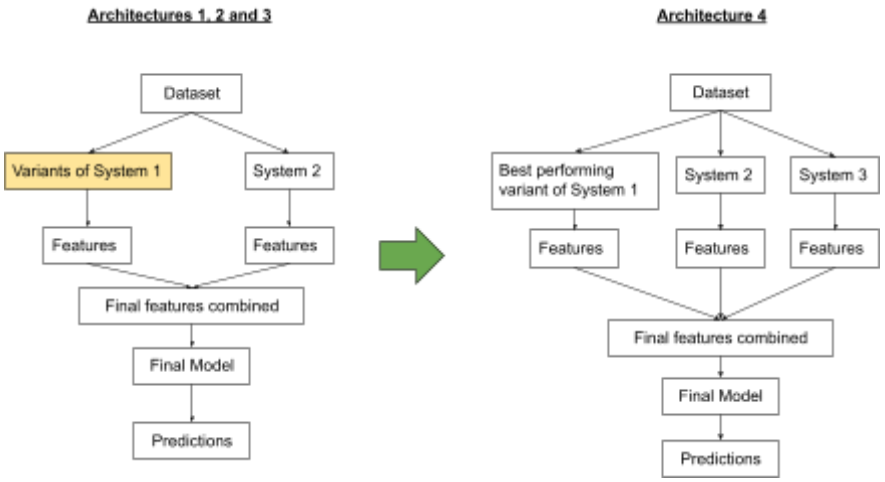


Figure 1: Architecture model used to generate features for a final predictive model. The first 3 architectures are all the same except for the variants used in System 1.

The reason why different variants of System 1 are used is because we want to understand what 8K report representation is more suitable for the problem. In previous studies, polarity scores alone have demonstrated to perform well in predictive models, however in other cases, the use of tf-idf values have also shown to be promising features. These variants help to decide which and if not both combined linguistic feature representations are better to take into account on the final model.

Data description

Corpus

A modified version of the prepared dataset is used for the supervised ML algorithms. As previously mentioned, this dataset contains descriptions of financial events reported in 8K documents from the year 2002 to 2012 with corresponding stock prices.

Evidently, this dataset is based on publicly U.S listed companies, specifically those in the S&P 1500 index, composed of some of the 1500 largest companies listed on stock exchanges in the United States.

The working dataset consists of:

- Dataset from Heeyoung Lee et al. (2014) containing historical prices for all companies in S&P 1500 index with corresponding 8K documents released as well as historical prices and volatility index of the S&P 500 index.
- Twitter dataset, containing tweets making a direct or indirect reference to any of the companies used in the previous dataset at the time of releasing 8K documents.

In order to avoid and reduce missing values, the historical prices and volatility index of the S&P 500 index is used instead of the S&P 1500 index, as the dataset of the latter one is generally provided from the year 2012 onwards (may be costly to obtain data from earlier years). At the same time, the S&P 500 index still provides a good notion of how the market is performing overall (as it includes many of the companies from the dataset), thus can be used to normalize certain features (which will be explained in the following section). The data is split into 80% training and 20% testing. A great care for the quality of the input data used for training and testing is applied by preprocessing the 8K documents and obtaining the necessary features from the initial 20,000 8K reports that were available in the dataset, 1,070 of those were removed (resulting in 18,930 reports left) for having missing values such as moving averages.

Dataset	# of 8K documents
Train	15,144
Test	3,786

Table 2: The training and testing data quantities.

The numbers in **Table 2** are true for architectures 1, 2 and 3. However, the quantity of training and testing data decreases dramatically (almost 10 times less) on the last architecture. This is due to a matching problem between datasets. The original dataset that this research works with contains 8K reports released between the year 2002 to 2012, however, because Twitter data is used, aligned with reports, this implies that a large part of reports is given up, since Twitter was first founded the 21st of March in 2006. In addition, given the fact that the last architecture accounts for the importance of speculative data found in Twitter, it is imperative that only those reports published after 2006 are considered (**Table 3** shows the updated dataset for Architecture 4).

Dataset	# of 8K documents
Train	1,535
Test	392

Table 3: The training and testing data quantities after using Twitter data.

Corpus extension

In order to account for additional linguistic data coming from twitter, a new dataset must be created and appended to the existing one. For each 8K document, the company's symbol and release date is used in order to scrape a maximum of 50 tweets from the same date until the next day. As each tweet is scraped, this one is immediately cleaned by removing links, hashtags, mentions, punctuations and emojis.

After the list of clean tweets obtained per report is completed, the polarity and confidence (probability) of these ones is obtained by using the pre-trained text classifier from the *flair* package. In order to rule possible tweets that are collected by chance, those reports with less than related 3 tweets found are ignored.

Data Analysis

From the updated data (after removing missing values) a matrix of 18,930 (8K reports) rows x 37 (number of features) columns is made. In order to understand the composition of the dataset, the univariate distribution of the target is displayed in **Figure 2**.

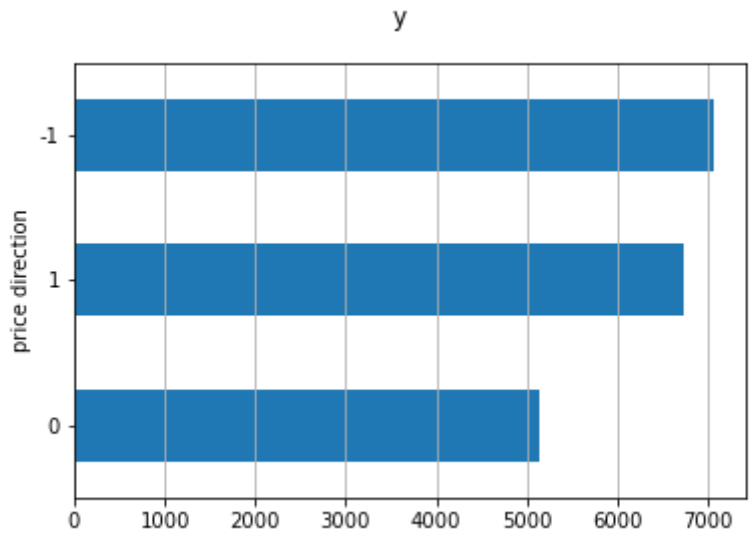


Figure 2: Univariate distribution of the target labels frequency with a bar plot (entire dataset).

Based on the plot from above, we can say that the dataset is relatively balanced as the 8K reports in the entire dataset resulted in a similar distributed percent of prices going up, remaining the same and going down. Effectively, this is helpful for the predictive models as imbalanced data could potentially make it harder for the models to classify the features due to certain classes appearing noticeably less times during training, affecting the precision and recall of the models which could learn to predict some classes well but not others.

After performing some analysis on the data, it was found that 'Other events, Regulation FD Disclosure and Results of Operations and Financial Condition' were the 3 categories that appeared more often in the reports (43% of the times). The nature of these events have a tendency to impact the stock prices more than categories such as 'Unregistered Sales of Equity Securities'. Therefore, the slight data imbalance (relatively minimal) between categories can be justified by the fact that more reports were released with a category that was more likely to change the price of the stock than remaining the same. The data is ensured to remain balanced after splitting it (**Appendix Figure 1-2**).

Features

The features used vary on the models implemented. The numerical/non-linguistic features remain the same throughout the entire experiment.

Type	Features	Linguistic features
Non-linguistic	Moving Averages	1 week, 1 month, 1 quarter and 1 year price change before the 8K document is released. Also normalized using the S&P 500 index.
	Volatility index	Volatility index of the S&P 500 at the time of the 8K document release. Volatility is a statistical measure of the variability of returns for a given security or market index, typically defined as the standard deviation of returns over some finite period.
Linguistic	Event category	The event type of 8K reports shown in Appendix 1. An 8K report with multiple events has multiple event category features (One-hot-encoded vector).
	8K Report Polarity score	Positive or Negative score of 8K reports.
	Tweets Polarity and confidence score	Positive or Negative polarity (one-hot-encoded) with confidence probability of Tweets.
	Tf-idf 8K report values	Vector representing word relevancy of a document

Table 4: The list of features. We have 4 moving average features, 1 volatility index feature, 39 event category features, 1 or 3 polarity scores (depending on the model used accounting for 8k reports and tweets) and 1 or none (representing 8K docs) tf-idf values after feature selection.

For each 8K report, the difference in the company's stock price before and after the report is released is calculated. For example, if the 8-K report is published before market opens, this difference is computed between the closing price on the day of the 8K release and the opening price 2 days after. This is then normalized by subtracting the same difference computed for the entire S&P 500 index (stock index GSPC) for the same period. For example, if a company's stock price goes up 3% 2 days after the event and the S&P 500 index goes up 1% in the same period, then the normalized change is 2%. This normalization is needed to isolate the company-specific change from the overall market trend, in the hope that the investor using this tool can outperform overall market trends. The normalized price change rate is binned into one of three labels: UP (the price goes up more than 1%), DOWN (the price goes down more than 1%), STAY (the price change is within 1%).

Even though Heeyoung Lee et al. (2014) results' are used as a baseline, it must be noted that they solely used 8K reports which contained information related to Earnings Surprise, as opposed to this study, which completely disregards this factor, as this study intends to predict price movements given any type of events reported and not just those that are directly or somehow related to the Earnings Surprise factor of a company. This factor may limit the ability of the models to generalize and comprehend different driving elements of price changes. In addition, given that Earnings Surprise is a highly informative feature of how well or bad a company performs, it may also limit the ability to evaluate how effective the NLP and ML algorithms are.

Non-linguistic features

This section shows how the feature extraction and labeling is done for the non-linguistic features using a real example. Below is a snippet from an 8K report of *Fidelity National Information Services Inc*. on July 24th 2007.

On July 3, 2007, Fidelity National Information Services, Inc. ("FIS") announced that its subsidiary Certegy Check Services, Inc., had learned of the misappropriation of consumer information by a former employee...

There are two event types in this 8K report: *Regulation FD Disclosure, Financial Statements and Exhibits.*

Close price of FIS Inc. on July 24, 2007	55.84
Open price of FIS Inc. on July 25, 2007	54.00
Close price of S&P index on July 24, 2007	1511.04
Open price of S&P index on July 25, 2007	1518.09
Close price of VIX on July 24, 2007	24.17

Table 5: Various financial information about FIS Inc. or other market index (VIX: volatility index)

Table 3 shows the price and other information about the company at the time of the event. Given this document and stock prices of FIS Inc., the price change is calculated to be -3.3%, and normalized by the subtracting change of the S&P 500 index in the same time period, which was -0.5%. Note, that with and without normalization, the original change stays binned into STAY. This label is paired with a set of features extracted automatically. The first group of features is financial. A separate feature is

constructed based on the close price of the volatility index on July 24, 2007, which was 24.17. This report contains two event types: Regulation FD Disclosure and Financial Statements and Exhibits, which generate another 39 features (one-hot-encoded) where the presence of these two events is represented by a 1 and the absence of other categories is represented by 0s in a single vector. To calculate recent price movements, a 5-day moving average (MA) for the 1-month price change feature is used, 10-day MA for the 1-quarter change, and 20-day MA for 1 year. These are also normalized using the S&P 500 index. For the stock in this example, these values were: 9.5% for 1-week change, 11.7% for 1-month change, 19.3% for 1-quarter change and 46.2% for 1-year change.

Linguistic features

Event Categories

Each report can be labeled by 1 or more categories. These categories are extracted from the metadata in the 8K reports. Prior to August 23, 2004, 8K items were filed under different item numbers and names. Therefore, the dataset contains a mix of reports which are categorized using the old and new categories, hence the large list in **Appendix 1**.

In order to be able to extract some of the linguistic features, the 8K documents must be cleaned (preprocessed) by tokenizing , lemmatizing, removing stop words and applying a negation heuristic to the documents. Below you can find an illustration of this process.

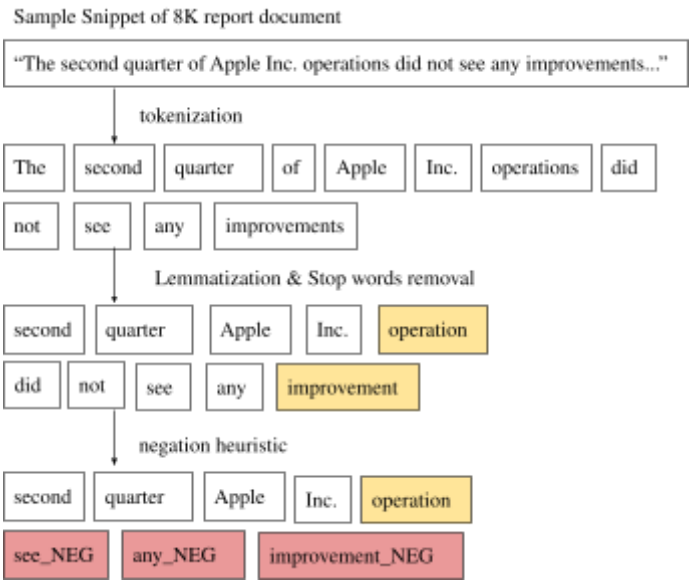


Figure 3: Preprocessing pipeline of 8K documents

Tf-idf Values

Sentiment classification can be done via different methods. One of the selected methods is tf-idf values. An intuitive way to classify text is to create a matrix with a column for each unique word in the corpus, create a row for each 8K document, and then fill in the values in each column for how many times the word appeared in the corpus. However, this bag-of-words method (count vectorization) is likely to produce a huge sparse matrix. This method also misleads the models by using common words with high frequencies but little predictive power over the target variable. Tf-idf values address this problem by rejecting the simple idea of counting, and instead, increase the value of a word proportionally to its count while remaining inversely proportional to the frequency of the word in the corpus.

The initial feature matrix of the training data has a shape of 15,144 (number of 8K documents in training) x 40,000 (maximum length of vocabulary), evidently being very sparse.

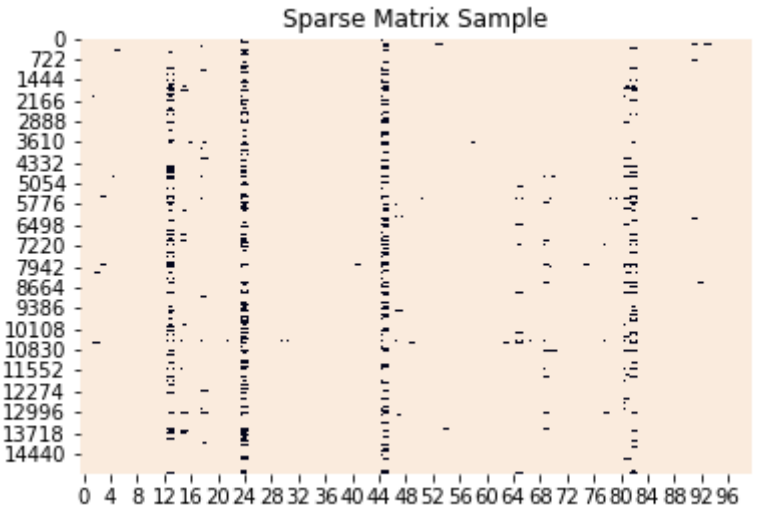


Figure 4: Sparse matrix sample of the training data representing tf-idf values

In order to drop some columns and reduce the matrix dimensionality, some feature selection is carried out.

1) Each category is treated as binary (for example, the 'UP' category is 1 for the 8K documents that led to the price going Up and 0 for the others).
2) Chi-Square test is performed to determine whether a feature (word in the vocabulary) and the (binary) target are independent.
3) Only the features with a certain p-value from the Chi-Square test (used to identify relationships between categorical variables) are kept.

The p-value is the probability that the null hypothesis is true. The p-values tell us whether an observation is as a result of a change that was made or is a result of random occurrences. In order to accept a test result we want the p-value to be low, that way we are able to identify if there is a strong correlation or dependency between words present in the 8K documents and the price of this correspondent stock going up, staying the same or down.

After performing dimensionality reduction on the document to the tf-idf matrix from the 15,144 training files with two different p-values, the Chi-Square test demonstrated that there was no strong dependent association between words and any of the targets.

P-value	**0.20**	**0.05**	**Some selected features (words)**
up	67	7	Ford, lone, apollo, medici, alpha, gap
stay	117	8	Quarter, million, sale, compared, gambl
down	48	2	Gap, riverbed, million, orbital
TOTAL	**210**	**15**	

Table 6: Number and example of words selected after using different p-values for the Chi-Square test

From the results displayed in **Table 6**, it is evident that using the Chi-Square test to determine the new dimensions of the tf-idf matrix is not feasible as most of the words (+90%) in the vocabulary would be discarded, and the words selected to do not show a strong relationship which the potential causes of price changes (e.g. words that could affect the price going up such as overperformed, growth, etc). No word on its own is strongly related to the result of price going up,staying the same or down, however the combination of many different words could potentially have more power in classifying a document as Up, Stay or Down.

Fortunately, the dimensions of this matrix can be further reduced, by removing the columns (representing words) that do appear less than 12 times in the entire corpus. After completing the dimensionality reduction process, the matrix results in a size of 15,144 x 17,434 (approx. a quarter of what it was - 40,000).

The tf-idf values were trained with training data using a Naive Bayes algorithm: a probabilistic classifier that makes use of Bayes' Theorem, a rule that uses probability to make predictions based on prior knowledge of conditions that might be related. This algorithm is generally the most suitable for large datasets as it considers each feature independently, calculates the probability of each category, and then predicts the category with the highest probability.

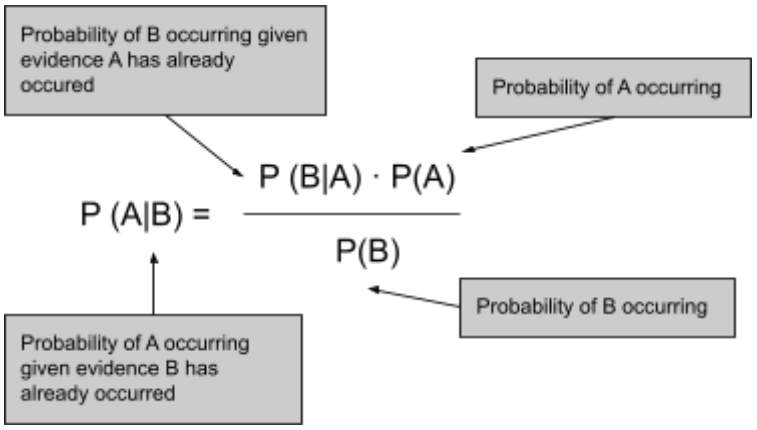


Figure 5: Naive Bayes equation

Apart from predicting and classifying values, the Naive Bayes model can also provide the probability of each class. Finally, the predictions obtained from the Naive Bayes model, are multiplied by the highest probability class (corresponding to the predicted class itself) in order to normalize it and store that value as a confidence value rather than a discrete one (1, 0, -1). For instance, if the predicted class is 1 (Up) out of 0 and -1, and the probability of having predicted a 1 is 0.63, the final feature would be 1 x 0.63 = 0.63. Using the confidence scores will allow the final model to account for reports that have already shown a higher probability of obtaining a given class.

8K docs - Polarity scores

The 8K docs polarity score feature attempts to provide a scoring measure of how negative or positive an 8K document is. In order to do this, the SentiWordNet, an open-domain sentiment lexicon along with a list of domain-specific words from Loughran & McDonald (L&M) REFERENCE are used. The purpose of using the latter list is to minimize the issue faced by ORIGINAL PAPER REFERENCE where the SentiWordNet did not model well the financial domain thus polarity scores obtained from the reports seemed to be misleading. The L&M list is a financial domain-specific list of words classified as positive and negative.

All tokenized words are searched in the L&M first to obtain a polarity (all positive words are given a 0.5 score and all negative scores are given a -0.5 score), however if the word is not present in the list, the polarity score of this one is obtained by using the SentiWordNet (words with a positive score

greater than 0.5 and a negative score lower than 0.4 are given a positive score of 0.5. Words with a negative score greater than 0.5 and a positive score lower than 0.4 are considered negative with a final score of -0.5). Furthermore, given that a negation heuristic is used to preprocess the 8K documents (all negated words contain '_NEG' at the end, e.g 'grow_NEG') the polarity score of all words is multiplied by -1 to account for the opposing polarity of the word when negated.

The scores of all words are summed up, ultimately providing a single final polarity score for the entire 8K document. Scores below 0 represent negative polarity and scores above 0 represent positive polarity.

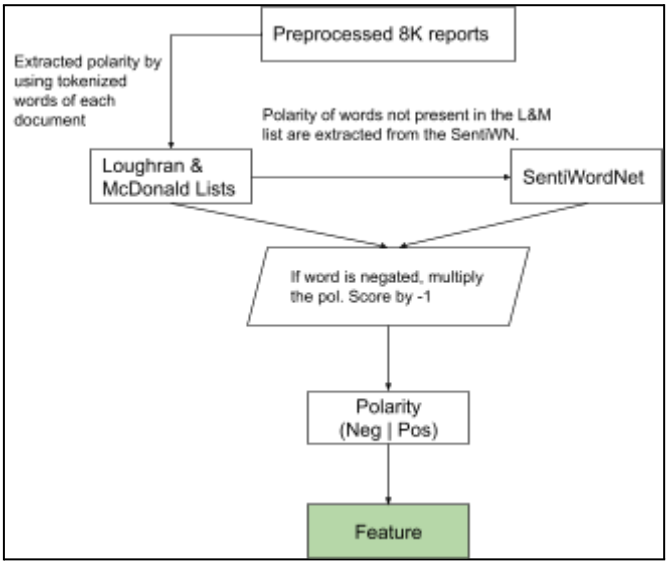


Figure 6: Process of obtaining polarity scores from 8K documents

Twitter - Polarity scores

The twitter polarity scores are obtained through a pre-trained sentiment analysis model from the *flair* library (a state-of-he-art model). This model splits the text into character-level tokens and uses the DistilBert model to make predictions. The advantage of working at the character-level as opposed to word-level is that words that the network has never seen before can still be assigned a sentiment. The model predicts the polarity of a given sentence (tweet) as well as the corresponding probability of being of that class (positive or negative). The process of obtaining these features is displayed in **Appendix System 3.**

Final model

The final model in which all architectures rely to make final predictions in this investigation is an ensemble model. Ensemble models combine several base algorithms (often called 'weak learners') to construct better predictive performance than a single tree base algorithm. These weak learners can be the same (homogeneous) or different (heterogeneous). The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner, thus increasing the accuracy of the model. When we try to predict the target variable using any machine learning technique, the main causes of difference in actual and predicted values are noise, variance, and bias. Ensemble helps to reduce these factors (except noise, which is irreducible error).

The model selection depends on many variables of the problem: quantity of data, dimensionality of the space, distribution hypothesis. A low bias and a low variance, although they most often vary in opposite directions, are the two most fundamental features expected for a model. Indeed, to be able to solve a problem, we want our model to have enough degrees of freedom to resolve the underlying complexity of the data we are working with, but we also want it to not have too many degrees of freedom to avoid high variance and be more robust. This is well known as the bias-variance tradeoff.

In order to set up an ensemble learning method, we first need to select our base models to be aggregated. In this case, we use different types of base learning algorithms, as previously mentioned, this is a heterogeneous ensemble model, specifically a *stacking* ensemble.

The stacking ensemble trains the weak learners in parallel and combines them by training a meta-model to output a prediction based on the different weak models predictions. Stacking will mainly try to produce strong models less biased than their components (even if variance can also be reduced).

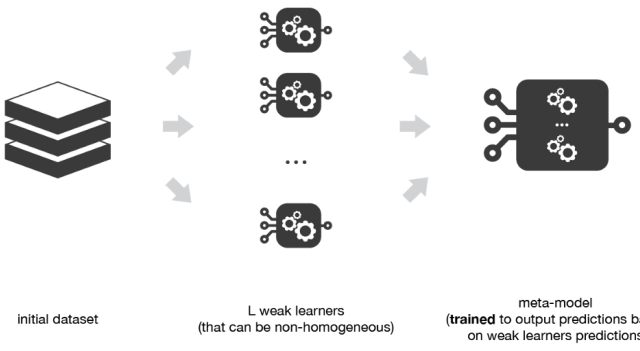


Figure 7: Example of architecture of an ensemble method

In order to evaluate the performance of the final model on each of the selected architectures, this one is compared against the individual performance of each of the selected weak learners, which are logistic regression classifier, k-nearest neighbours (k-NN) classifier, decision tree classifier, support vector machine and a gaussian naive bayes classifier. Each model is evaluated using repeated k-fold cross-validation. The mean performance of each algorithm is computed and also a box and whisker plot is created to compare the distribution of accuracy scores for each algorithm.

Experimental Results

Naive bayes performance on tf-idf values

The Naive Bayes classifier used for the tf-idf values is trained and then tested on the transformed test set. In order to evaluate the performance of the model, the following metrics were used: accuracy, confusion matrix, ROC, precision, recall and f1-score.

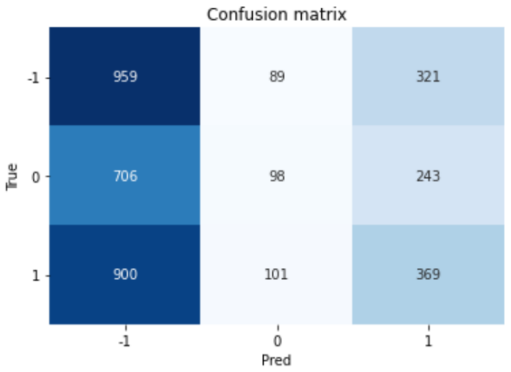


Figure 8: Confusion matrix of tf-idf predicted classes

	Precision	**Recall**	**f1-score**
-1	37%	70%	49%
0	34%	9%	15%
1	40%	37%	32%

Table 7: Metrics used to evaluate the performance of the Naive Bayes model on the test set

The model obtained an accuracy of 38% on the test set, suggesting a low predictability power and inability to accurately categorize documents as Up (1), Stay (0) or Down (-1). Moreover, the confusion matrix displays a summary breaking down the correct and incorrect predictions by each class. The table suggests that the model had a tendency to predict documents to be Down (-1) most of the time. Given that the training and testing dataset are balanced, it can be assumed that the model created more relationships between words and the label -1 than with the rest of labels, essentially classifying the documents as Down as a result of the higher number of relevant words found in negative events within corporations appearing in the documents.

From this table we are also able to extract the precision, recall and f1-score (**Table 7**). The precision indicates the fraction of relevant instances among the retrieved instances. In other words, the precision

allows us to see how many correct positive predictions the model made from a given class. For instance, the model predicted 933 8K documents to have a price change of Up (1) and 369 of those (a 40%) were actually correct, meaning that overall the model is more likely to be right about a prediction if this one is 1. Similarly, the recall displays the fraction of the total amount of relevant instances that were actually retrieved. For example, everytime the model made a prediction 70% of the time it would correctly predict the category Down (-1). Finally, the f1-score combines the precision and recall of the model into a single metric by taking the weighted average. Evidently, this last metric suggests that the model performs the best when it is faced with an 8K document that may potentially lead to the price of the correspondent stock going Down (-1) over the next 2 days.

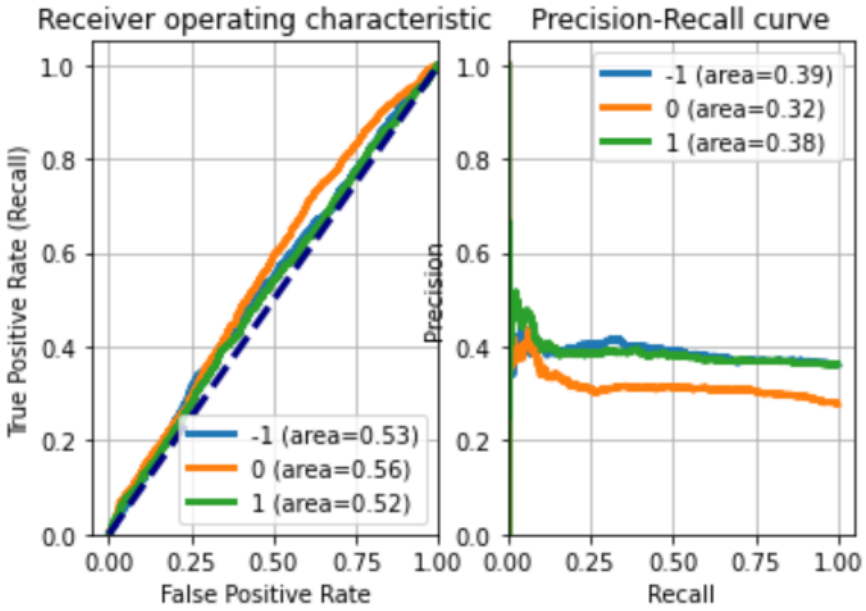


Figure 9: ROC and AUC graphs of the Naive Bayes classifier

The first graph, Receiver Operating Characteristic (ROC), is a plot illustrating true positive rate against the false positive rate at various threshold settings. The area under the curve (AUC) indicates the probability that the classifier will rank a randomly chosen positive observation higher than a randomly chosen negative one, making it even more clear that the classifier is not able to distinguish well between classes. The second graph summarizes the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds. A model with good skill is represented by a curve that bows towards (1,1) above the flat line of no skill.

Architecture 1

The first architecture contains the Variant 1 of System 1, in which only the polarity (positive or negative) scores of the 8K reports are considered, apart from event categories and the non-linguistic

values (moving averages and volatility). The plot below makes the previous hypothesis true for this problem (ensemble methods tend to improve overall accuracy), as the stacking ensemble model performed better than any of the individual weak learners (very close to the SVM and LR).

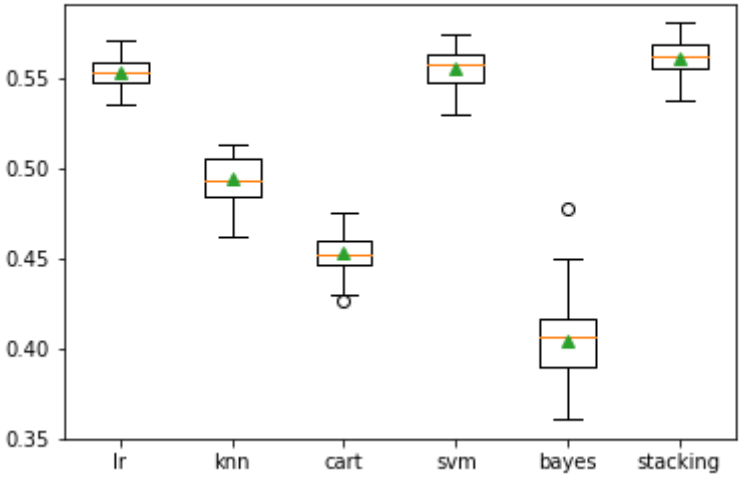


Figure 10: Box and whisker showing the mean performance and comparison of the distribution of accuracy scores for each algorithm when using polarity scores as the linguistic feature variable.

Furthermore, the following table displays the performance of each base model alone and the stacking model using k-fold validation on the training data.

	Accuracy % on training data (K-fold validation)					
	Logistic regression	KNN	Decision Tree	SVM	NB	Stacking
Accuracy using 10 k-fold validation	55.2	49.5	45.7	55.6	41.4	56.2

Table 8: Accuracy of base and stacking models on K-fold validation of training data

Architecture 2

The second system contains the Variant 2, which uses the same features used in the first variant but replacing 8K reports polarity scores for the predicted values obtained from the tf-idf values (after normalization, multiplying by the class probability). The performance of the weak learners and the Stacking algorithm with the given set of features is shown in **Figure 11**.

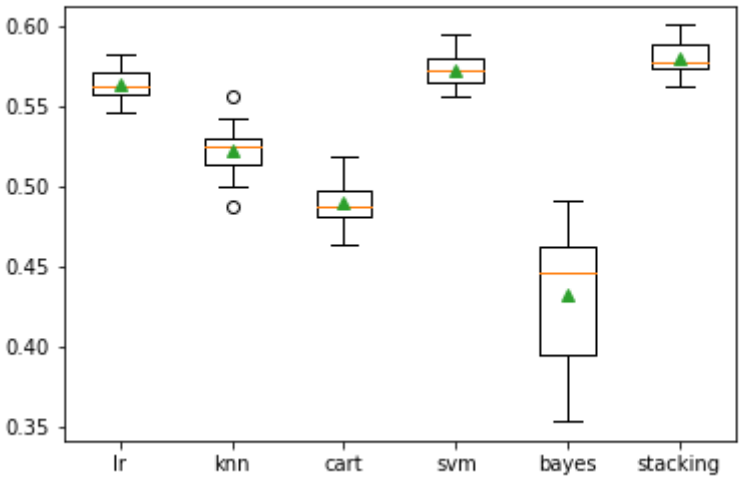


Figure 11: Box and whisker showing the mean performance and comparison of the distribution of accuracy scores for each algorithm when using tf-idf normalised values as the linguistic feature variable.

	Accuracy % on training data (K-fold validation)					
	Logistic regression	KNN	Decision Tree	SVM	NB	Stacking
Accuracy using 10 k-fold validation	56.4	52.3	49.0	57.3	43.2	58.0

Figure 9: Accuracy of base and stacking models on K-fold validation of training data

Once again, the Stacking ensemble method was the best performer with a 58% accuracy on the K-fold validation. However, it must be noted that all scores improved from the previous system, suggesting that the tf-idf normalized values could potentially be more meaningful to the system than the polarity scores, unless the models overfitted the training data.

Architecture 3

The third system combines the variable features used in the first and second variants (polarity scores and tf-idf normalized values). The performance of the weak learners and the Stacking algorithm follow the same trend as seen on the previous architectures (**Figure 12**). The previous two and the below plots demonstrate that even though the base models do not perform the best independently, they

are able to provide some notion of the given data and combined, increase accuracy and lower bias and variance.

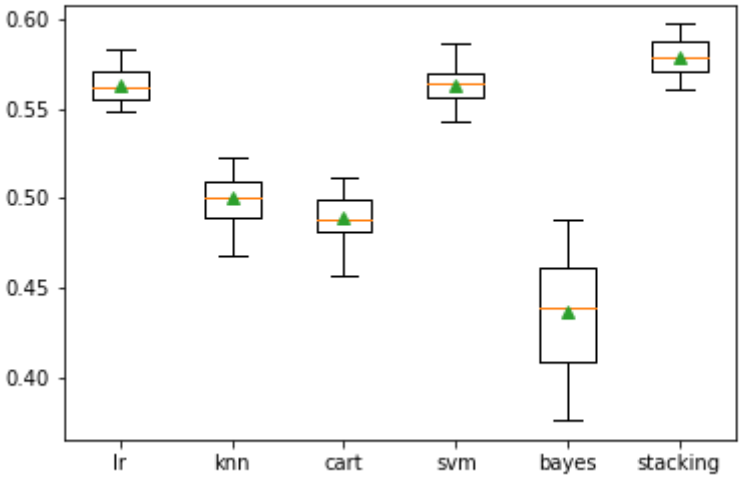


Figure 12: Box and whisker showing the mean performance and comparison of the distribution of accuracy scores for each algorithm when using tf-idf normalised values and polarity scores as the linguistic feature variables.

	Accuracy % on training data (K-fold validation)					
	Logistic regression	KNN	Decision Tree	SVM	NB	Stacking
Accuracy using 10 k-fold validation	56.3	50.0	48.9	56.3	43.6	57.8

Table 10: Accuracy of base and stacking models on K-fold validation of training data

Architectures 1, 2, 3 on testing data

Even though each system is evaluated using the K-fold method, the accuracy obtained does not actually explain how well the model would perform when faced with unseen data. In order to test the actual predictability power of the Stacking ensembles, these ones are used to predict testing data (**Table 11**).

	Stacking Ensemble Accuracy % on training data	Stacking Ensemble Accuracy % on testing data
Architecture 1	56.2	56.0
Architecture 2	58.0	54.0
Architecture 3	57.8	53.0

Table 11: Accuracy of stacking ensembles in Systems 1, 2 & 3 using training and testing data.

As expected, the accuracy on the test set decreased in comparison to the results obtained with the training data. However, even though the decrease was not relatively large, it was significant enough in some architectures to indicate which model overfitted the data more and performed worse with testing data (generalizing). The most promising architecture was the second one, however when testing data was used, the first one proved to be a more robust and accurate model. Given that as little as 0.5% can make a big impact in financial investments, the difference between architecture 1 and 2 is considered a large one (2%).

Results on architecture 3 also suggest that the combination of speculative and factual data as already stated, does not bring major benefits to the model's accuracy as it was expected to do.

Architecture 4

From the previous results obtained, the last architecture uses the feature that resulted in the best performance using testing data (Variant 1 - polarity scores). The box and whisker in **Figure 13** still show the same pattern between the weak learners and the stacking model.

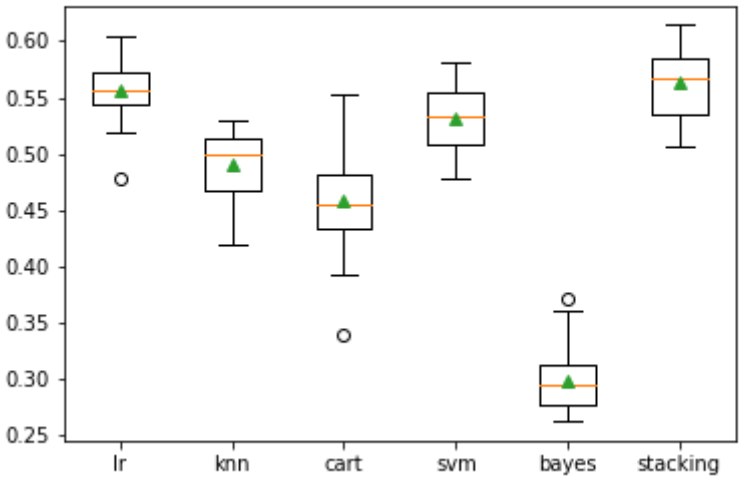


Figure 13: Box and whisker showing the mean performance and comparison of the distribution of accuracy scores for each algorithm when using polarity scores and tweets.

The smaller boxes reflect the lower deviation in data given the reduced dataset for this architecture.

	Accuracy % on training data (K-fold validation)					
	Logistic regression	KNN	Decision Tree	SVM	NB	Stacking
Accuracy using 10 k-fold validation	55.6	49.1	45.7	53.1	29.8	56.3

Table 12: Accuracy of base and stacking models on K-fold validation of training data

The overall results from the evaluation of this architecture demonstrate signs of potential predictability power, as the training data used was 1/10 of the amount used for the previous architectures.

Comparisons with baseline results

Ultimately, the forth architecture failed to perform as expected, by combining the speculative data coming from twitter along with the most significant factual feature (linguistic) from Systems 1 and 2. However, the other architectures successfully met the aim of the research by obtaining a higher accuracy percentage than a random walk with a 50%.

Moreover, the first architecture proved to perform better than the baseline model by 1%, a significant improvement in the accuracy of the predictive model (in the financial context).

	Accuracy % on testing data	Predominant linguistic features
Architecture 1	56.0	8K polarity scores
Architecture 2	54.0	Tf-idf 8k reports normalized predicted scores
Architecture 3	53.0	Combination of Arch.1 & 2
Architecture 4	49.0	8K polarity scores + twitter polarity scores
Baseline	55.5	8K reports unigram

Table 13: Accuracy comparison between the models used in this investigation and the baseline model from the paper providing the prepared dataset.

Evaluation

Even though the experiment has successfully met the project aim by applying NLP and ML techniques into the problem in hand, outperforming the baseline models through the exploration of different approaches and techniques to treat the data and use convenient models, the potential improvements and findings have been limited due to time and computational resources. The initial number of 8K reports available was in fact 195,000. However, the preprocessing and feature extraction time of only 20,000 reports was approximately 4 hours, requiring an intense memory load and space to compute. Unfortunately, the conditions under which this research has been carried out, did not allow for more resources to be used. Having used more reports, could have potentially improved the models (more training data), especially the model on the 4th architecture, as more reports would have implied a higher probability of retrieving reports that were released after 2006 (Twitter founded) which could have been used along with more twitter data.

It must be noted that even though Heeyoung Lee et al. (2014) used company filings (8K reports) too to make predictive models, they assisted the models by including data regarding earnings per share (EPS) which is a highly conclusive and meaningful information when determining the price change of a stock. Having this in mind, it could be stated that the feature extraction and preprocessing approaches of this investigation along with the models, have more predictive power than the methods and approaches proposed in baseline model paper since the models were able to obtain similar and even better accuracies with less deterministic data.

Instead of using the old school approach for sentiment analysis in the 8K reports, obtaining the tf-idf values, more popular techniques could have been applied such as Word embeddings (used with deep learning neural networks) or the state of the art language models (used with transfer learning from attention-based transformers) to obtain improved representations and potential increase accuracies.

As already stated, the best performing model was the one that used polarity scores from the 8K reports derived from the Loughran & McDonald (L&M) list and the SentiWordNet. Despite the fact that the L&M list is derived from 10X documents (which share similarities with 8K reports in terms of financial context) and uses categorization for a specific-domain (financial), the list contains a scarce number of positive (354) and negative (2,337) words relative to the number of vocabulary words found in the corpus, which ultimately lead to most of the polarity scores being obtained through the SentiWordNet since most of the words are likely to not be on the L&M list.

The task of creating a new dataset from tweets corresponding to specific companies and the dates in which their 8K reports were released proved to be a challenging task for the following reasons: keyword search and tweet cleaning. The lack of time, limited this research to only attempt a single method for each task and thus encounter several drawbacks. Using company tickers followed by the word 'stock' resulted in the majority of tweets being relevant to the search and restricted the number of possible tweets found, avoiding the collection of unrelated tweets containing the same two words by chance. For instance, the tweet 'there are no more aapls in stock, they are always late' could be retrieved following the explicit rules set, and ultimately contributing with a negative score to that specific company on the date of the 8K release. It was also found that many people tweeted the same thing over several times, which increased the polarity score of the collection of tweets for a given report. These repeated tweets also counted towards the Twitter API quantities requests and time limits, slowing down the process. A possible way to improve tweet selection could be based on a ranking system which considers other factors such as likes, shares and retweets.

While this work suggests that text analysis of company filings and tweets directly related to these can improve predictions of short term movements in stock prices, especifically with a 2 day horizon, this study does not claim that these techniques are the basis of a feasible trading strategy due other factors influencing the final outcome such as transaction costs, slippage, and borrowing fees (when taking short positions).

The study on the impact of using more factual and speculative information could be significantly improved if the dataset was updated. More sentiment and numerical data may be available at a cheaper cost. As part of the speculative sentiment data, this investigation planned to include historical financial news that were aligned to the release of the 8K reports (e.g. Reuters Financial News and Bloombers), however the only reliable sources claim to provide this structured historical data to big institutions at a very high cost to the average user.

The technological advancements over the last decade have pushed most of the companies to go digital and the potential of individual users affecting the market simply through sentiment is becoming more popular.

Conclusion

In this work we used an already prepared corpus from Heeyoung Lee et a. (2014) and extended it to investigate the importance of text analytics for stock price movement using different NLP and ML approaches. We also tested the impact of using speculative text data and factual text data, in spite of resulting in a significantly small dataset due to alignment inconsistencies between the prepared and the extended corpus. The results obtained reaffirmed the already stated hypothesis from other researchers that text does indeed carry predictive power for stock price movement.

This study also explored the various NLP approaches for sentiment analysis, utilizing tf-idf values and polarity scores. Each system demonstrated its limitations such as high dimensionality in tf-idf values and the short list available for domain specific words. On the other hand, the advantages of these methods were also explored and reflected on the positive results obtained from the final models. In addition, the importance of model selection was depicted by the comparison between the weak learners of the ensemble method and the ensemble method itself.

Further work can be done on the hyper parameter tuning of the models optimizing each weak learner, as well as improving the tweet selection rules/methods in order to extract more relevant and significant tweets related to the target stock. In addition, apart from attempting to use the state-of-the-art and popular techniques for sentiment analysis, the current approach of using tf-idf values could be enhanced by using other dimensionality reduction methods. Finally, there is also further work to be explored in the augmentation of the textual corpus by collecting data not only from Twitter but from recent, and new financial forums with high user engagements. Overall, the project explored different areas, advantages and limitations of different NLP and ML techniques used to solve the problem set, and met its aim by successfully obtaining positive results from its proposed solution.

Appendix

New categorisation

-Entry into a Material Definitive Agreement

-Termination of a Material Definitive Agreement

-Mine Safety - Reporting of Shutdowns and Patterns of Violations

-Completion of Acquisition or Disposition of Assets

-Results of Operations and Financial Condition

-Creation of a Direct Financial Obligation or an Obligation under an Off-Balance Sheet Arrangement of a Registrant

-Triggering Events That Accelerate or Increase a Direct Financial Obligation or an Obligation under an Off-Balance Sheet Arrangement

-Costs Associated with Exit or Disposal Activities

-Material Impairments

-Notice of Delisting or Failure to Satisfy a Continued Listing Rule or Standard; Transfer of Listing

-Unregistered Sales of Equity Securities

-Material Modification to Rights of Security Holders

-Non-Reliance on Previously Issued Financial Statements or a Related Audit Report or Completed Interim Review

-Departure of Directors or Certain Officers; Election of Directors; Appointment of Certain Officers; Compensatory Arrangements of Certain Officers

-Departure of Directors or Principal Officers; Election of Directors; Appointment of Principal Officers

-Amendments to Articles of Incorporation or Bylaws; Change in Fiscal Year

-Amendment to Registrant's Code of Ethics, or Waiver of a Provision of the Code of Ethics

-Change in Shell Company Status

-Submission of Matters to a Vote of Security Holders

-Shareholder Director Nominations

-ABS Informational and Computational Material

-Change of Servicer or Trustee

-Change in Credit Enhancement or Other External Support

-Failure to Make a Required Distribution

-Securities Act Updating Disclosure

-Regulation FD Disclosure

Other Events (The registrant can use this Item to report events that are not specifically called for by Form 8-K, that the registrant considers to be of importance to security holders.)

Financial Statements and Exhibits

Old categorisation

Changes in Control of Registrant

Acquisition or Disposition of Assets

Bankruptcy or Receivership

Changes in Registrant's Certifying Accountant

Other Events

Other events

Resignation of Registrant's Director

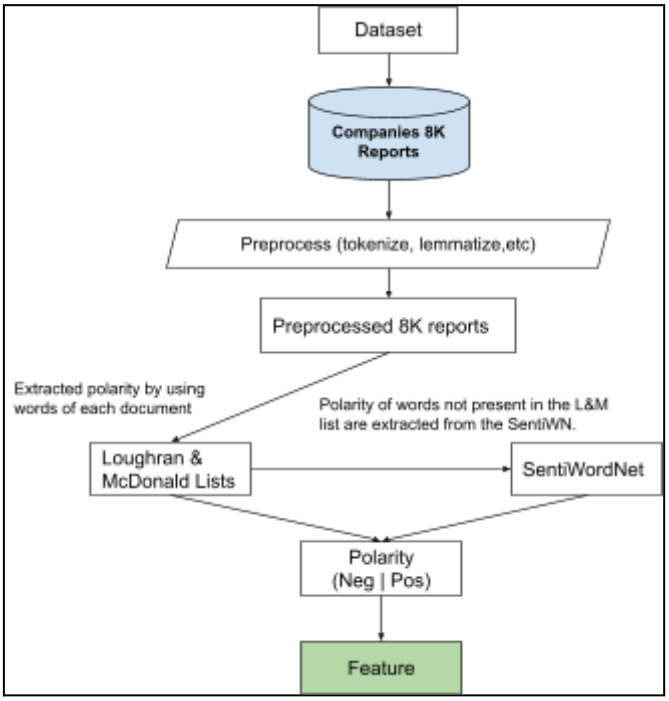
Change in Fiscal Year

Amendments to the Registrant's Code of Ethics

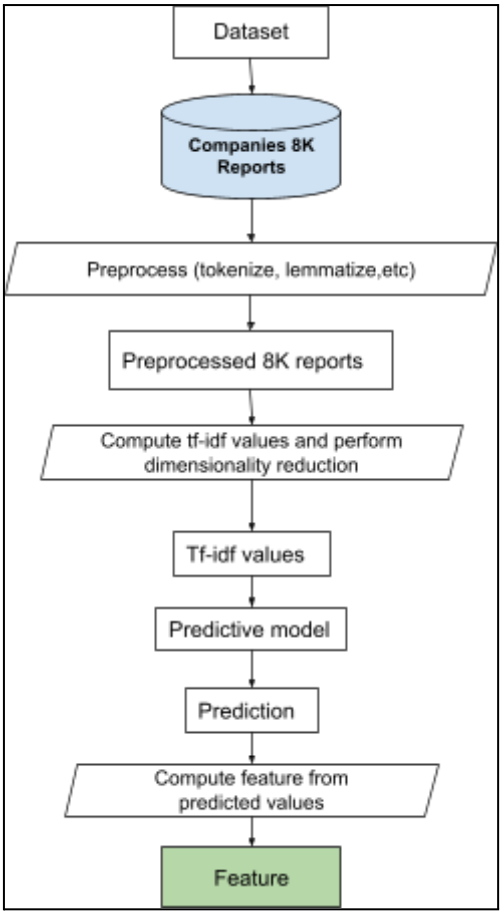
Temporary Suspension of Trading Under Registrant's Employee Benefit Plans

---

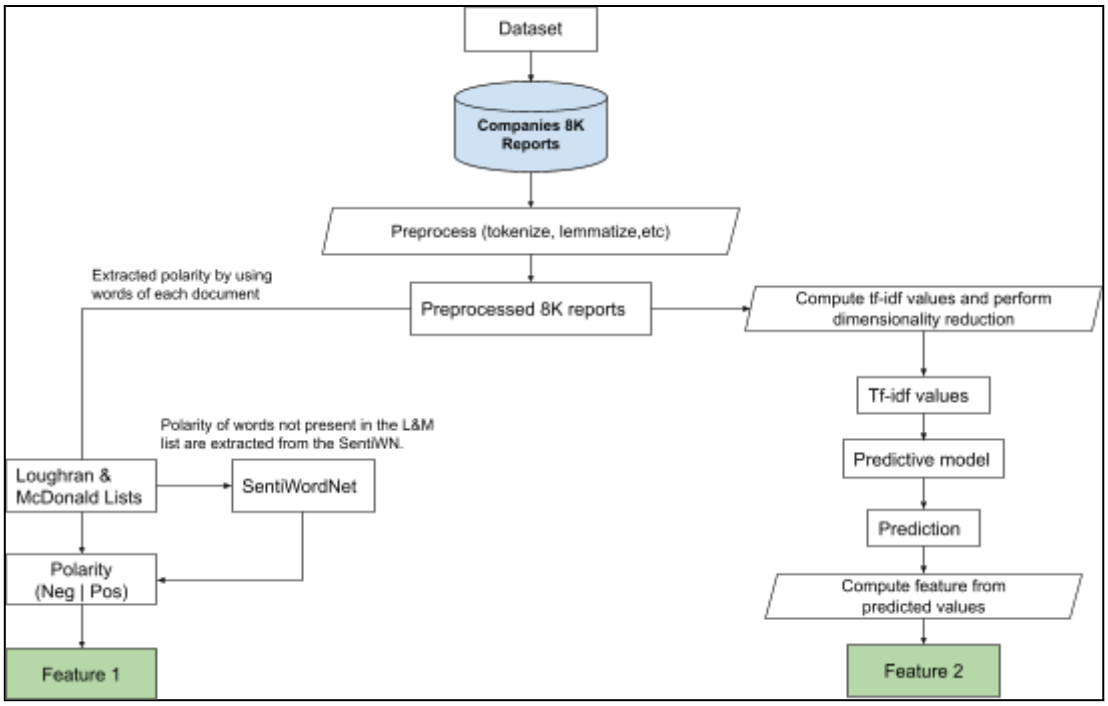
Appendix 1: Old and New list of financial event types in 8K reports from http://en.wikipedia.org/wiki/Form_8-K



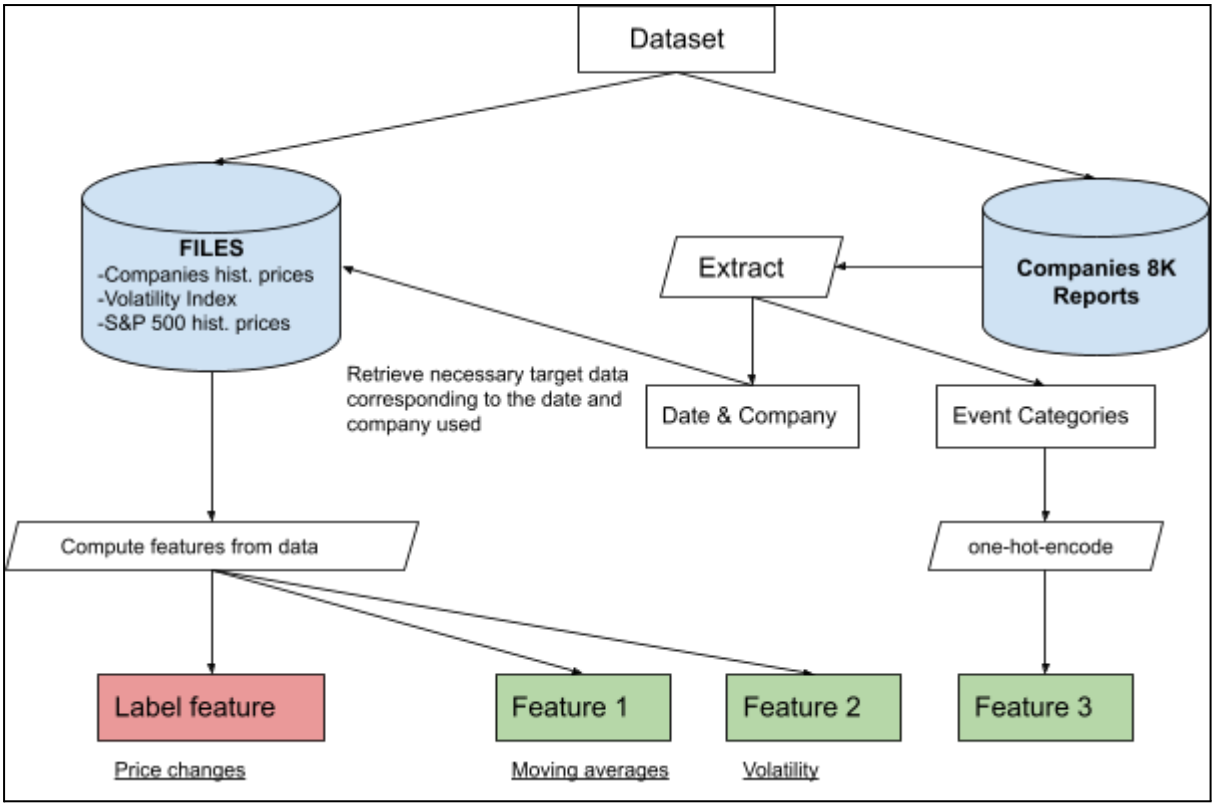
Appendix System 1 - Variant 1: Feature process extraction of Polarity of 8K reports



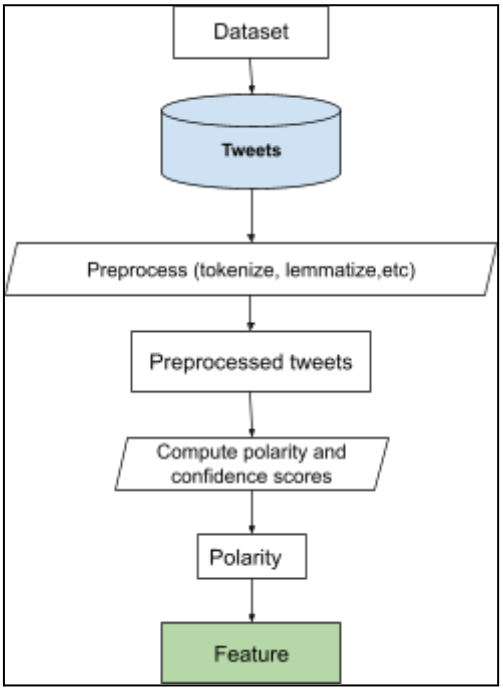
Appendix System 1 - Variant 2: Feature process extraction using tf-idf values of 8K reports



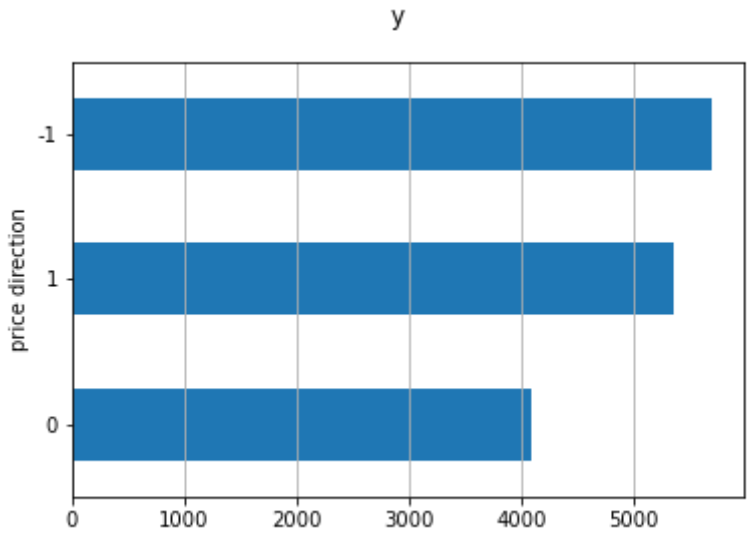
Appendix System 1 - Variant 3: Feature process extraction combining Variants 1 and 2.



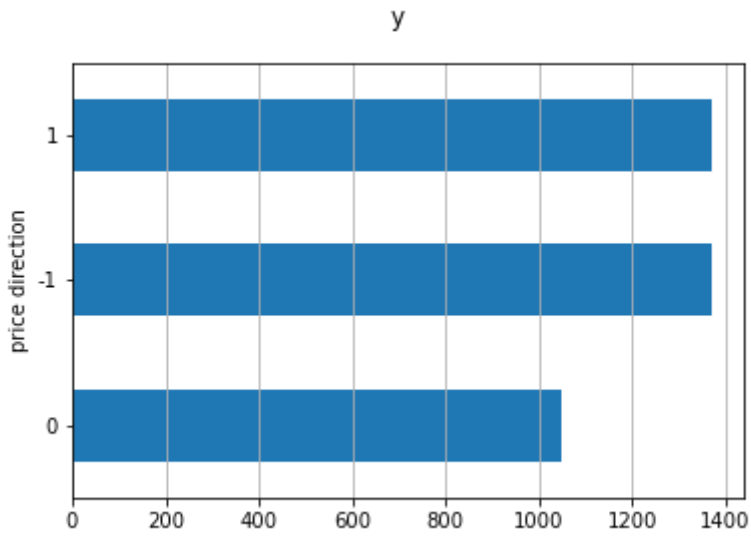
Appendix System 2: Feature process extraction of moving averages, volatility and event categories



Appendix System 3: Feature process extraction of tweets polarity.



Appendix Figure 1: Univariate distribution of the target labels frequency with a bar plot (training data).



Appendix Figure 2: Univariate distribution of the target labels frequency with a bar plot (testing data).

Bibliography

Khan, W., Malik, U., Ghazanfar, M.A., Azam, M.A., Alyoubi, K.H. and Alfakeeh, A.S. (2019). Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis. *Soft Computing*.

Schumaker, R.P. and Chen, H. (2009). A quantitative stock prediction system based on financial news. Information Processing & Management, [online] 45(5), pp.571–583. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0306457309000478 [Accessed 20 Apr. 2020].

Appati, J.K., Denwar, I.W., Owusu, E. and Soli, M.A.T. (2021). Construction of an Ensemble Scheme for Stock Price Prediction Using Deep Learning Techniques. International Journal of Intelligent Information Technologies, 17(2), pp.72–95.

Hiyama, Y. and Yanagimoto, H. (2018). Word polarity attention in sentiment analysis. Artificial Life and Robotics, 23(3), pp.311–315.

Jiang, H. and Li, W.Q. (2011). Improved Algorithm Based on TFIDF in Text Classification. Advanced Materials Research, 403-408, pp.1791–1794.

Stock Price prediction and Performance analysis using Machine Learning. (2020). Journal of Xidian University, 14(6).

Nam, K. and Seong, N. (2019). Financial news-based stock movement prediction using causality analysis of influence in the Korean stock market. Decision Support Systems, 117, pp.100–112.

Schumaker, R.P. and Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news. ACM Transactions on Information Systems, 27(2), pp.1–19.

Mehtab, S. and Sen, J. (2019). A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing. SSRN Electronic Journal.

Iyinoluwa, O. (2019). Stock Market Trend Prediction Model Using Data Mining Techniques. Current Trends in Computer Sciences & Applications, 1(5).

Paul, P.V. and N, D. (2020). Stock Market prediction based on Technical –Deviation-ROC indicators using stock and Feeds data. Recent Advances in Computer Science and Communications, 13.

Goshima, K. and Takahashi, H. (2017). Building a Sentiment Dictionary for News Analytics using Stock Prices. Journal of Natural Language Processing, 24(4), pp.547–577.

Penka, D. and Zeijlstra, H. (2010). Negation and polarity: an introduction. Natural Language & Linguistic Theory, 28(4), pp.771–786.

Jarrah, M. and Salim, N. (2016). Stock Market Prediction Based on Term Frequency-Inverse Document Frequency. Journal of Economics, Business and Management, 4(3), pp.183–187.