UNIVERSITEIT STELLENBOSCH UNIVERSITY

DEPARTMENT OF ECONOMICS

UNIVERSITY OF STELLENBOSCH

Correlating Factors of U.S. Presidential Speeches with Stock Market Movements – a Deep Learning Approach

Draft 1

By Pablo Rees

[19119461]

First draft formally presented with intention of fulfilment of the requirements for completion of the MCom Economics Programme (full thesis only) at Stellenbosch University.

With supervision from Dawie van Lill

14 June 2022

1. Introduction

2. Literature review

The aim of this project is to determine whether U.S Presidential speeches have predictive power over stock market movements. A positive result would be powerful and could add to the ability of traders to accurately predict stock market movements. Broadly, machine learning has been selected as the method of modelling and the S&P 500 index has been chosen as representative of the 'stock market'.

The undertaking of this project assumes 2 important conjectures. That there is a correlation between the linguistic factors[1] of speech employed by U.S. presidents in their speeches and U.S. stock market movements; and that the prominence of speech factors can be quantified by using machine learning algorithms. The following literature review defends the two conjectures, confirms the novelty of the project, surveys methods of data cleaning that may be applicable and to discover which machine learning methods are most appropriate for this project.

2.1. Conjecture defence

The following section references the findings of four peer-reviewed articles in defence of the conjectures made in section 1. More evidence exists and is reviewed in later sections but is not necessary to defend the conjectures made here.

2.1.1. FOMC speeches and U.S. financial market reactions

Hayo, Kutan & Neuenkirch (2008) performed a generalized autoregressive conditionally heteroscedastic (GARCH) analysis of the relationship between Federal Open Market Committee (FOMC) speeches and the U.S. financial market. The analysis was quantitative on the market side and qualitative on the speech side. They found that FOMC speeches influence trader behaviour, but that this effect is both asymmetrical (negative impacts were larger than positive impacts) and non-uniform across trader type (bond markets were affected far more than financial and forex markets). Further, more formal modes of communication have a larger impact on both returns and conditional variance and more prominent speakers have a greater impact on bond markets. Finally, they found that volatility in 3- and 6-month T-bills was reduced on the day of a speech.

It was commented that heteroskedasticity left something to be desired when assessing the effects of monetary shocks. Further, it was found that speeches alone were not sufficient to create significant effects for financial markets. It is important that news agencies propagate the news for market repercussions to occur. In a brief, informal interview with a few bond traders it was discovered that they tended to "read monetary policy statements and listen to speeches by Greenspan (Bernanke) themselves. Other types of communications are rather neglected and the traders tend to rely on newswire information" (Hayo et al., 2008:27). Further, it was noted that news articles fail to take a neutral stance on the contents of speeches implying that the market effect may be distorted by the sentiment of news agencies.

[1] 'Linguistic factors' in this sense is intended to mean any and all patterns that can be detected in spoken language including verbiage, lexicon, tone, register, sentence length, word combination etc.

This article shows that the sentiments of communications influence the effect on markets. Thus, analysing sentiment is an important factor in an accurate appraisal of the relationship between speeches and markets. The finding that speeches need to be propagated by news agencies in order to have significant impacts on financial markets is counter to the hypothesis of this project but the FOMC is less publicly scrutinized than the U.S. president which may render this finding inconsequential to this project.

2.1.2. Political speeches and stock market outcomes

Maligkris (2017) demonstrates that the speeches given by U.S. presidential candidates directly influence the stock market, particularly during the early months of their campaigns. These speeches often contain information about potential presidents' positions on policy changes and public issues. Thus, they can affect investor sentiment and in turn the stock market. The employed methodology was to analyse transcripts of presidential candidate speeches from the American Presidency Project archives and the U.S. Government Publishing Project from the 2004-2016 period according to the index developed in Baker, Bloom & Davis (2016) (explained in section 2.3). He shows, using regression analysis that there is an increase in excess market returns of 26 basis points following candidate speeches, however the direction and magnitude of this effect varies between candidates. He goes on to examine whether the difference in effect is due to heterogenous speech content. Finally, it was demonstrated that speeches laden with economic information tend to boost stock returns while also reducing volatility. Speeches with a negative *tone* have the opposite effect. The long run effect of speeches is dependent on market conditions.

This paper indicates that there is a correlation between presidential candidates' speeches and stock movements. It is then reasonable to believe that there is a correlation between presidential speeches and stock market reactions. It also highlights that tone affects the relationship and thus that the sentiment of a speech is important. Further, the archives of the American Presidency Project and the U.S. Government Publishing Project are good sources for U.S. political speech transcripts – the potential predictive data.

2.1.3. Measuring economic policy uncertainty

Baker, Bloom & Davis (2016) develop an Economic Policy Uncertainty Index based on the frequency of articles containing a trio of terms in 10 leading U.S. newspapers. These terms were: ' "economic" or "economy" ; "uncertain" or "uncertainty"; and one or more of "congress", "deficit", "Federal Reserve", 'legislation", "regulation", or "White House" ' (Baker et al., 2016:1594). The terms were selected over a period of 24 months during which more than 15 000 news articles were read by humans in an auditing process. The index proved to be quite accurate, spiking near the expected events, including wars, tight presidential elections, fiscal policy battles and terrorist activity.

They went on to show that their index had a strong relationship with other economic uncertainty measures and policy uncertainty measures. Further, congruence in uncertainty prediction was found between left- and right- leaning newspapers.

This article shows that language processing can be used to predict economic events, particularly economic uncertainty and economic policy uncertainty. However, the type of language processing proposed for this project differs significantly. While Baker et al. (2016) used man hours this project will use deep learning – the result of which is likely to be greater accuracy and reduced man hour expenditure, if it is correctly applied.

2.1.4. Hope, change and financial markets: can Obama's words drive the market?

Sazedj & Tavares (2011) asked whether the speeches made by former U.S. President Barack Obama affected stock market prices. By regressing the event of a speech during Obama's first 11 months in office on the daily excess returns of the Dow Jones, the S&P 500 and the NASDAQ they found that the event of a speech had a generally insignificant effect on daily excesses. However, by regressing key terms contained in 43 speeches given during the first 11 months of Obama's presidency it was found that the content of speeches can significantly affect daily excess returns nearly uniformly across all three indices. Notably, the NASDAQ's correlation to the content of speeches was weaker – indicating that technology markets may be less susceptible to presidential rhetoric.

This paper highlights the relationship between the content of a presidential speech and stock market returns. This study was correlational rather than predictive in nature.

2.1.5. Conjecture defence summary

As seen in section 2.1.2. and section 2.1.4. it is true that there is a correlation between the linguistic factors of speech employed by U.S. presidents in their speeches and U.S. stock market movements (Sazedj & Tavares, 2011; Maligkris, 2017). Further, section 2.1. and section 2.3. show that sentiment and other linguistic factors of speech can be quantified using machine learning methods (Hayo, Kutan & Neuenkirch, 2008; Baker, Bloom & Davis, 2016). Thus, both conjectures hold and further investigation is warranted. This is not the total of all evidence supporting these conjectures but is sufficient. Other articles reviewed here can be seen for further evidence.

2.2. Novelty

Searching 'sentiment analysis stock market' on Google Scholar yields mostly articles linking Twitter data and the stock market. Searching 'presidential speeches affect stock market' on Google Scholar yields articles on the relationship between presidential speeches and the stock market but none using Machine Learning techniques. Khedr, Salama & Yaseen (2017) describe related work as including relating news or twitter data to stock market behaviour and prices and relating financial news to stock prices, but do not mention presidential speeches. Searching 'machine learning S&P 500' on Google Scholar yields an array of articles using machine learning as a technical analysis tool but most of these use other financial indicators and are not NLP based – bar one article analysing the effects of former U.S. president, Donald Trump's tweets on the S&P 500 and the DJIA. Searching 'political speech machine learning' on Google Scholar yields articles that focus on political speeches but have no link to stock markets. Other searches yielded similar results, thus as far as can be told – this research is novel in nature.

2.3. Data cleaning methods for NLP

Katre (2019), in his analysis of Indian political speeches, uses Natural language Toolkit (NLTK) and string methods to remove punctuation, HTML tags and English stopwords[2], as well as converting speeches to lowercase and tokenizing[3] them. Zubair & Cios (2015), before

[2] 'Stopwords' are words that commonly occur across all speech and therefore only create noise in the data. Some examples are 'the', 'it', 'they' and 'and'.

[3] 'Tokenizing' refers to the splitting of words into tokens that have linguistic importance, for example the words 'terrorism', 'terrorist' and 'terror' may all be tokenized to 'terror'. Thus, the core concept of the word is captured while also simplifying the dataset.

correlating the sentiment in Reuters articles with S&P 500 movements, clean their text data by tokenizing it with NLTK. Kinyua et al. (2021) clean their twitter data (tweets from then U.S. president, Donald Trump) by deleting all tweets on days when the stock trading was closed, deleting all tweets that only contained standard stopwords and deleting all tweets that only contained URLs. Khedr, Salama & Yaseen (2017) tokenize, standardize by converting to lowercase, remove stopwords from and stem[4] their textual data before processing the abbreviations (replacing abbreviations with the full phrase) and filtering out words that consist of two or less characters.

2.4. Machine learning methods

The following section looks first at the literature informing the sentiment analysis space and then at the literature around stock market prediction in order to determine methods that suit the intersection between the two.

2.4.1. Sentiment analysis methods

Ren, Wu & Liu (2019) analyse news articles at the sentence level by assigning a sentiment polarity (using software designed for the Chinese language) to each word followed by a sentiment score for each sentence in a document. Each document is then categorized and a sentiment score between -1 and 1 for all news for that day is generated. Zubair & Cios (2015) use the positive and negative valence categories from the Harvard General Enquirer (HGI) to assign each word in a Reuters news article a positive or negative label. They then sum the positives and negatives into a tuple and divide that tuple by the number of words in an article in order to create a vector that represents each news article. The vectors are organized into time series, normalized by dividing all vectors by the first vector, parsed through a Kalman filter and then correlated to S&P 500 returns using Pearson correlation (for both the positive and negative scalar in the vector). Kinyua et al., (2021) use the Valence Aware Dictionary for Sentiment Reasoning (VADER) to create a sentiment feature for former U.S. president Donald Trump's tweets which was then used as a regression feature in linear, decision tree and random forest regressions. Khedr, Salama & Yaseen (2017) use N-gram (n=2) to extract key phrases from their corpus of news text data, then term-frequency inverse-document-frequency is used to determine the importance of those phrases within the corpus, and finally use a naïve-Bayes classifier to assign positive and negative labels to each news document. Purevdagva et al. (2020) use a variety of features present in both data and metadata to predict fake political speech. Two features relevant to this project were 'speaker job' and 'context' (press, direct or social) which were labelled using universal serial encoders. For the actual sentiment analysis they used the linguistic inquiry and word count (LIWC) tool to categorize and count words into emotional, cognitive and structural text components. Various further attempts to extract sentiment from the text did not yield increased prediction accuracies. They go on to use an extra tree classifier for feature selection and then support vector machine (SVM), multilayer perceptron, convolutional neural network, decision trees, fasttext and bidirectional encoder representations from transformers (BERT) for prediction with highest accuracy coming from the SVM. Dilai, Onukevych & Dilay, (2018) use SentiStrength – an automatic sentiment analysis tool - to compare the sentiment in speeches between former U.S. president Donald Trump and former Ukrainian president Petro Poroshenko.

2.4.2. Stock market prediction methods

[4] 'Stemming' refers to the removal of suffixes to reduce the complexity of a dataset.

Ren, Wu & Liu (2019) use an SVM and fivefold cross validation approach to achieve a prediction accuracy of 98% when predicting fake news in political speech. They combined sentiment data and market indicators as their input data. Kinyua et al. (2021) use linear, decision tree and random forest regressions to predict S&P 500 and DJIA directional changes. Random forest regression performed best for both datasets. Khedr, Salama & Yaseen (2017) use open, high, low and close prices from their stock market data as features after labelling the change from the previous day as positive, negative or equal. Jiao & Jakubowicz (2017) extracted lag and window features from the S&P 500 and the global 8 index before running time series random forest, neural network and gradient boosted trees to predict movements of individual stocks in the S&P 500. Liu et al. (2016) used forward search feature selection to select features for SVM, naïve-Bayes, Gaussian discriminant analysis and logistic regression from a set of economic features including the crude oil daily return, currency exchange rates and major stock indices daily returns in order to forecast the S&P 500 movement.

2.5. Summary of literature review

Table 1 represents the literature review in a condensed format which allows for easy comparison of the data and methods used and resulting accuracies. Table 2 represents the metadata, linked through 'Paper number' for Table 1.

Paper number	ML method	Cleaning method	Data type	Index predicted	Accuracy
1	Support vector machine with fivefold cross validation		Daily online stock reviews relating to the SSE 50	SSE 50	0.7996 – 0.9773
1	Support vector machine with rolling windows		Daily online stock reviews relating to the SSE 50	SSE 50	0.7133 – 0.8993
1	Logistic regression with fivefold cross validation		Daily online stock reviews relating to the SSE 50	SSE 50	0.7096 – 0.8656
2	Non-ML:	Tokenized using NLTK and mined for sentiment using the Harvard General Inquirer dictionary	Reuters textual data	S&P 500	Correlation: up to -0.908
3	Random forest	Tweets from non-trading days were removed, standard stop words and URLs were removed.	@theRealDonaldTrump tweets	INDU	0.94-0.98
3	Decision tree				0.93 – 0.97
3	Logistic regression				0.7-0.81
3	Random forest			S&P 500	0.91-0.92
3	Decision tree				0.83-0.88

3	Logistic regression				0.64-0.77
4	N-gram, TF-IDF, Naïve Bayes, K-NN	Tokenize, to lowercase, stopwords, stemming, abbreviation processing, filtering words with two or less characters	News articles and financial reports from Nasdaq.com, Reuters, wall street journal, marketwatch.com, zacks.com, yahoo finance, Google finance and economics.com.	Yahoo Inc, Microsoft Corporation MSFT and Facebook Inc.	0.898
5	Time series logistic regression	Feature extraction: lags, window features (technical indicators)	Non-text data: numerical indicator data – S&P 500 historical data and the global 8 index	Individual S&P 500 stocks movements	0.7861
5	Time series random forest				0.7797
5	Time series neural network				0.7775
5	Time series gradient boosted trees				0.7798
6	Logistic regression	Transform daily prices into daily returns, exclude data from days when markets were closed, aligning data from markets in different time zones, feature selection using the forward search method	Non-text data: numerical indicator data - global financial market indices, currency exchange rates, S&P 500 historical data	S&P 500 index future market trend	0.6062
6	Gaussian discriminant analysis				0.6062
6	Naïve Bayes				0.6038
6	Linear SVM				0.5979
6	Radial Basis Function SVM				0.6251
6	Polynomial SVM				0.5943
7	SVM	Feature extraction: speech subject, location, speaker profile, speaker credibility, context. Feature selection: extra tree classifier	Political speeches and metadata	Liar dataset	73.8
7	Multilayer perceptron				0.557
7	Convolutional neural network				0.614
7	Fasttext				0.662
7	BERT				0.66

Paper number	Title	Citation	DOI
1	Forecasting Stock Market Movement	(Ren, Wu & Liu, 2019)	10.1109/JSYST.2018.2794462

	Direction Using Sentiment Analysis and Support Vector Machine		
2	Extracting News Sentiment and Establishing its Relationship with the S&P 500 Index	(Zubair & Cios, 2015)	10.1109/JSYST.2018.2794462
3	An analysis of the impact of President Trump's tweets on the DJIA and S&P 500 using machine learning and sentiment analysis	(Kinyua et al., 2021)	10.1016/j.jbef.2020.100447
4	Predicting Stock Market Behaviour using Data Mining Technique and News Sentiment Analysis	(Khedr, Salama & Yaseen, 2017)	10.5815/ijisa.2017.07.03
5	Predicting Stock Movement Direction with Machine Learning: an Extensive Study on S&P 500 Stocks	(Jiao & Jakubowicz, 2017	10.1109/BigData.2017.8258518
6	Forecasting S&P 500 Stock Index Using Statistical Learning Models	(Liu et al., 2016)	10.4236/ojs.2016.66086
7	A machine-learning based framework for detection of fake political speech	(Purevdagva et al., 2020)	10.1109/BigDataSE50710.2020.00019

3. Data collection and cleaning process

The following is a description of the data collection and cleaning process and the feature extraction process. It begins by explaining the three datasets collected for this study, namely control, meta and test data. Control refers to autoregressive features extracted from the S&P 500, meta refers to S&P 500 adjacent financial data and test refers to vectorised presidential speeches which are the focus of this study. The section begins by elaborating on the collection and cleaning steps for each of these datasets. The next subsection describes the feature engineering methods (sentiment analysis and word vectorization) that were used to draw variables with potentially strong signals from the text data. The final subsection describes the reasoning, methods and results used in a time series analysis of the financial time series (S&P 500) data in order to extract useable autoregressive features from it.

3.1. Data

Three types of data were gathered for this project, namely: presidential speeches/text data (all the transcripts from the Presidency Project website including formal and informal and written and verbal addresses), a history of the S&P 500 index and a history of 6 S&P 500 adjacent price histories – 2 sets of financial data.

The S&P 500 index and the metadata was downloaded from Yahoo Finance while the presidential speeches were scraped from the American Presidency Project (Yahoo Finance, n.d.; Woolley & Peters, n.d.). The S&P 500 is downloaded using the 'fin_data_downloader' Python module and will always download the entire history of the S&P 500 index at a daily interval. The same goes for the metadata. The presidential speeches on the other hand were scraped using the 'WebScrapeAndClean' Python module which will also always scrape the entire corpus available on the American Presidency Project website. This allows for perfectly updated data to be collected at any time.

The S&P 500 data contains seven variables, namely: 'Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', and 'Volume'. 'Open' records the opening price for each day, while 'Close' records the closing price. 'High' records the daily high and 'Low' the daily low. 'Volume' records the dollar amount of stock traded in the S&P 500 on each day and 'Adj Close' is irrelevant because it never differs from 'Close'. Notably, opening and closing prices are only differentiated between after April 20th 1982 while volume was only recorded after 1950.

After scraping, the Presidency Project data contains five variables. These are 'Type' which records the category and sub-category of each speech, 'Name' which records the title and name of the main speaker, 'Date' which records the date the speech occurred on, 'Title' which records the title of each speech and 'Transcript' which contains the raw HTML transcript of the speech.

3.2. Cleaning

The S&P 500 index did not require any cleaning after download, besides the removal of the redundant 'Adj Close' variable. Conversely, the presidential speeches required extensive cleaning. Text that has been web-scraped contains HTML tags[5], thus the first step was to remove the HTML tags from the text. Similarly, the test contained reactions from the crowds

[5] HTML tags include paragraphing and spacing indicators for computers to interpret such as ' ' and ' '.

listening to the speeches[6], which were also removed. Next, the transcripts were converted to lower case and the question sections removed. Next a 'No Stops Transcript' variable was created by removing the stopwords[7] in the Natural Language Tool Kit (NLTK) stop words dictionary from the clean transcript. The original transcripts were also kept. Table A1 in Appendix A shows the shape of the data at this point. The cleaning of both the speech data and both financial datasets was done in their original collection scripts, i.e. the 'fin_data_downloader' Python script and the 'WebScrapeAndClean' Python script respectively.

3.3. Text feature engineering

To increase the strength of the signals coming out of the speech data and reduce the computational power required to run the machine learning algorithms – feature engineering is required. Feature engineering refers to the emphasis of certain signals within the available data and creation of new variables which capture these signals. It results in the addition of extra features (variables) to the dataset. There are three broad methods of feature engineering: feature selection, feature extraction and the creation of new features (Géron, 2019:27). In this section the creation of new features occurred and took the form of sentiment analysis and word vectorization for the text data.

3.3.1. Sentiment analysis

Two versions of sentiment analysis were carried out. First, NLTK's Valence Aware Dictionary for Sentiment Reasoning (VADER) was used to extract a sentiment analysis tuple, in this instance containing four scores, namely, negativity, neutrality, positivity and compound. VADER is a lexicon based system of sentiment analysis (Sohangir, Petty & Wang, 2018). Each of negativity, neutrality and positivity describe a transcript independently of the other scores while compound describes a transcript comprehensively (combining the other three scores). VADER, when compared with alternative NLP feature extraction techniques has performed better on social media transcripts and generalized better to other areas (Hutto & Gilbert, 2014; Elbagir & Yang, 2019). VADER has been used in vectorizing text relative to financial data in Pano & Kashef (2020) for Bitcoin price predictions, Agarwal (2020) which found a strong correlation between VADER sentiment scores and stock price changes and Sohangir et al. (2018) which shows the superiority of lexicon based approaches (specifically VADER) over ML approaches for sentiment classification.

Next, the TextBlob package's sentiment analysis tool was used. This yields two scores (in a tuple) describing the sentiment of a transcript, namely, a polarity score (ranging from -1 to 1) and a subjectivity score (ranging from 0 to 1). Polarity describes whether the emotions expressed are negative or positive, with lower scores indicating negativity while subjectivity indicates the extent of the usage of subjective words (Chudy, n.d.). Biswas, Sarkar, Das, Bose & Roy (2020) use Textblob sentiment scores in their analysis of the effects of Covid-19 on stock markets. Textblob was also tested in Sohangir et al. (2018) but did not perform as well as VADER although it did outperform ML methods in terms of area under the curve (AUC) scores. It is expected that the VADER sentiment tuple will outperform Textblob's sentiment tuple as a predictor of the S&P 500 data.

---

[6] These were tags such as 'Laughter' and 'Applause'.

[7] Stopwords are words that commonly occur in the English language and are therefore unlikely to contain any sort of signal and thus constitute only noise.

3.3.2. Speech vectorization

Converting human readable text into machine readable data requires the conversion from words to numbers. This vectorization can be done in various ways but in order to preserve the meaning of the texts the Word2Vec and Doc2Vec Python packages provided by Gensim were used (Řeh uřek & Sojka, 2010). However, Word2Vec was originally published in two papers by Mikolov, Chen, Corrado & Dean (2013a,b) while Doc2Vec was suggested by Le & Mikolov (2014).

3.3.2.1. Word2Vec

Gensim's Word2Vec Python package's skip-gram model is used for Word2Vec vectorization. Using the full vocabulary of words in a corpus of speeches and one-hot encoding for word vectors, Word2Vec trains a single hidden layer neural network to predict words based on the words around them. The parameters used for training are available in Appendix A.

The model contextually embeds each word in the entire corpus of speeches by running the speeches through the single hidden layer predictive neural network (NN). The NN is provided with the pseudo-task of predicting the word of focus from the words surrounding it each time it occurs in the corpus. Thus, a hidden layer is trained to contain the information that contextually embeds each word in the corpus. These hidden layer vectors (rather than the predictions) are the real output of the Word2Vec model. Words that appear in similar contexts throughout the corpus will have similar representational vectors (hidden layers) and thus can be said to have similar meanings in the corpus(Mikolov et al., 2013a,b).

An example phrase might be 'The quick brown fox jumps'. If the word 'brown' is the focus word, the words 'quick' and 'fox' would be fed into the neural network which would then be trained to map the input to the word 'brown'. Doing this for every instance of 'brown' in a corpus creates a hidden layer that contains all the contextual information required to predict the word 'brown' in the given corpus. Figure 1 depicts this process.

The model is extremely good at relating words that appear in similar contexts to each other. For example, when asked for the three words most similar to 'oil' the model trained on the presidential speeches corpus returns 'crude', 'gas' and 'petroleum'. The input 'gold' returns 'silver', 'bullion' and 'coin'; whilst 'virus' returns 'covid19', 'infection' and 'pandemic'.
In this study the representational vectors each contain 200 elements (because the hidden layers were set to contain 200 elements). Thus each word is described by a vector containing 200 elements. In order to create a similarly sized vector for each speech, the vectors describing all the words in each speech were averaged. Thus each speech has been reduced to a 200 element vector averaging the contextual embeddings of each word contained therein. This averaging technique was also used in Vargas, de Lima & Evsukoff (2017) and Qin & Ji (2018). However, this method fails to preserve word order in a document vector (Le & Mikolov, 2014).

Notably, the algorithm also makes room for phrases such as 'asset backed' or 'short selling' – which are considered as 'assetbacked' and 'shortselling'. When two words that occur irregularly in the vocabulary occur together frequently the algorithm interprets them as a phrase and treats them as such. There is, however, only room for two words in each phrase if the algorithm has only been executed once – which is the case here.

Word2vec is used by (Shi, Zhao & Xu, 2019) for the improvement of sentiment classification which implies that the final vectors in this study may contain sentiment information. It is also used by Vargas et al. (2017) and Qin & Ji (2018)in their predictive modelling of S&P 500 changes based on twitter data. Vargas et al. (2017) also used 200 element vectors while Qin & Ji (2018) used 300 element vectors. Both studies achieved prediction accuracy around 65%.

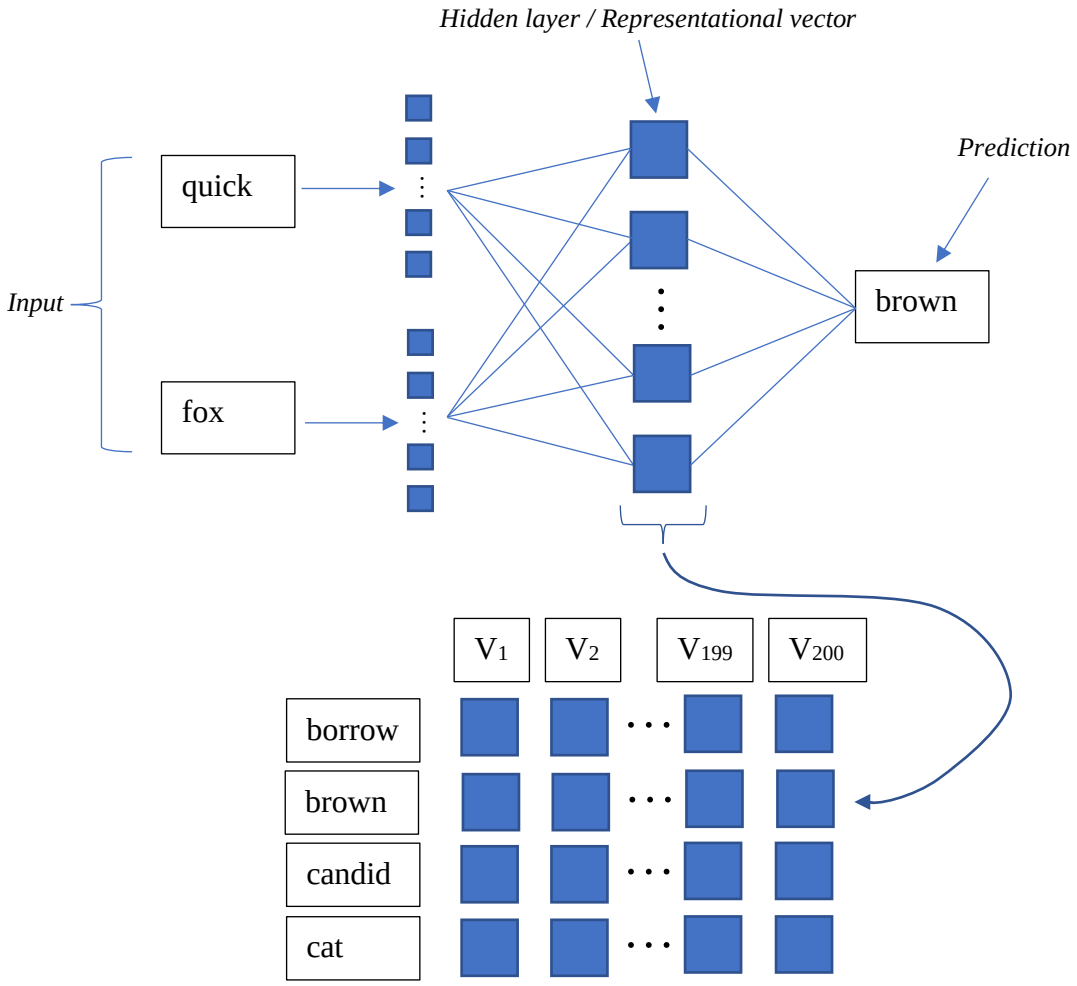

Figure 1: Word2Vec model

3.3.2.2. Doc2Vec

An alternative method to creating a vector representation of a sentence or document is Doc2Vec. This method is similar to the Word2Vec method described above but includes an additional floating vector when performing its pseudo-task. This vector is maintained across every word prediction task within a document and subjected to training for each instance of every word. Thus each document in a corpus is assigned a single comprehensive vector that embeds it in the corpus. Embedding within the broader corpus is maintained by tagging each document with a unique tagging phrase. This method outperforms other methods such as bag-of-words for text representations (Le & Mikolov, 2014).

There are two implementations of Doc2Vec, namely Distributed Bag of Words (DBOW) and Distributed Memory (DM) (Sohangir et al., 2018). Both have been used in this study. In both

cases each document is assigned a paragraph tag which represents the paragraph to the Doc2Vec algorithm. DM Doc2vec does this in the same way that a word represents itself to the Word2Vec algorithm. For every word in a document, its paragraph tag is passed to the Doc2Vec algorithm along with the words relevant to the current prediction pseudo-task. Back propagation is employed in the same manner as in Word2Vec except that the document tag vector is optimized for separately from the word vectors. This document tag vector is the relevant output in this case. Because a single vector of weights is created for each document, this vector should constitute a vectorized representation of the document as a whole. Figure 2 depicts the architecture of DM Doc2Vec algorithms. Alternatively, DBOW Doc2Vec algorithms ignore the context of a word and use random sampling to predict words from a paragraph. Figure 3 depicts the architecture of a DBOW Doc2Vec algorithm.

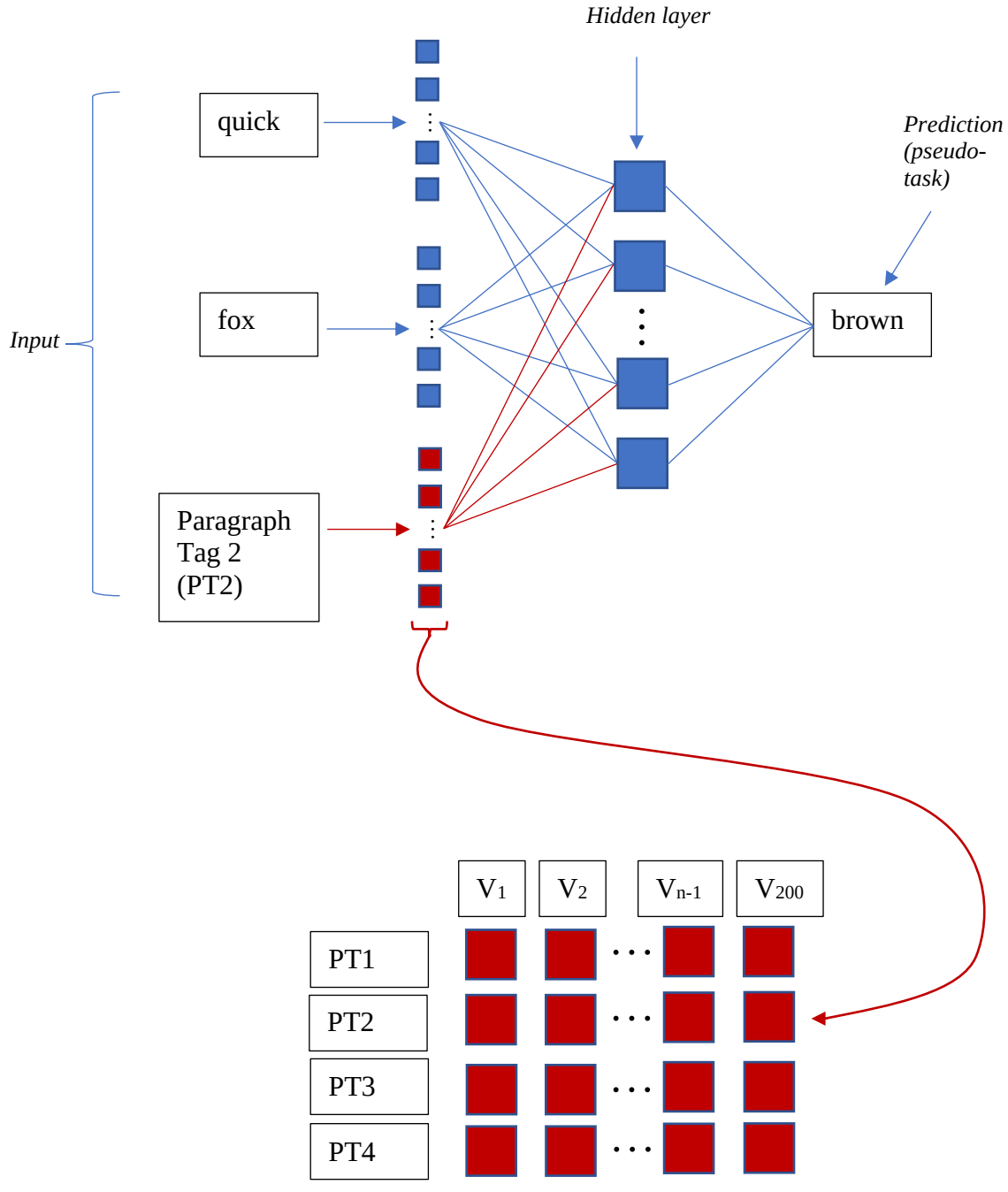

Figure 2: Doc2Vec model – distributed memory architecture - dm=1

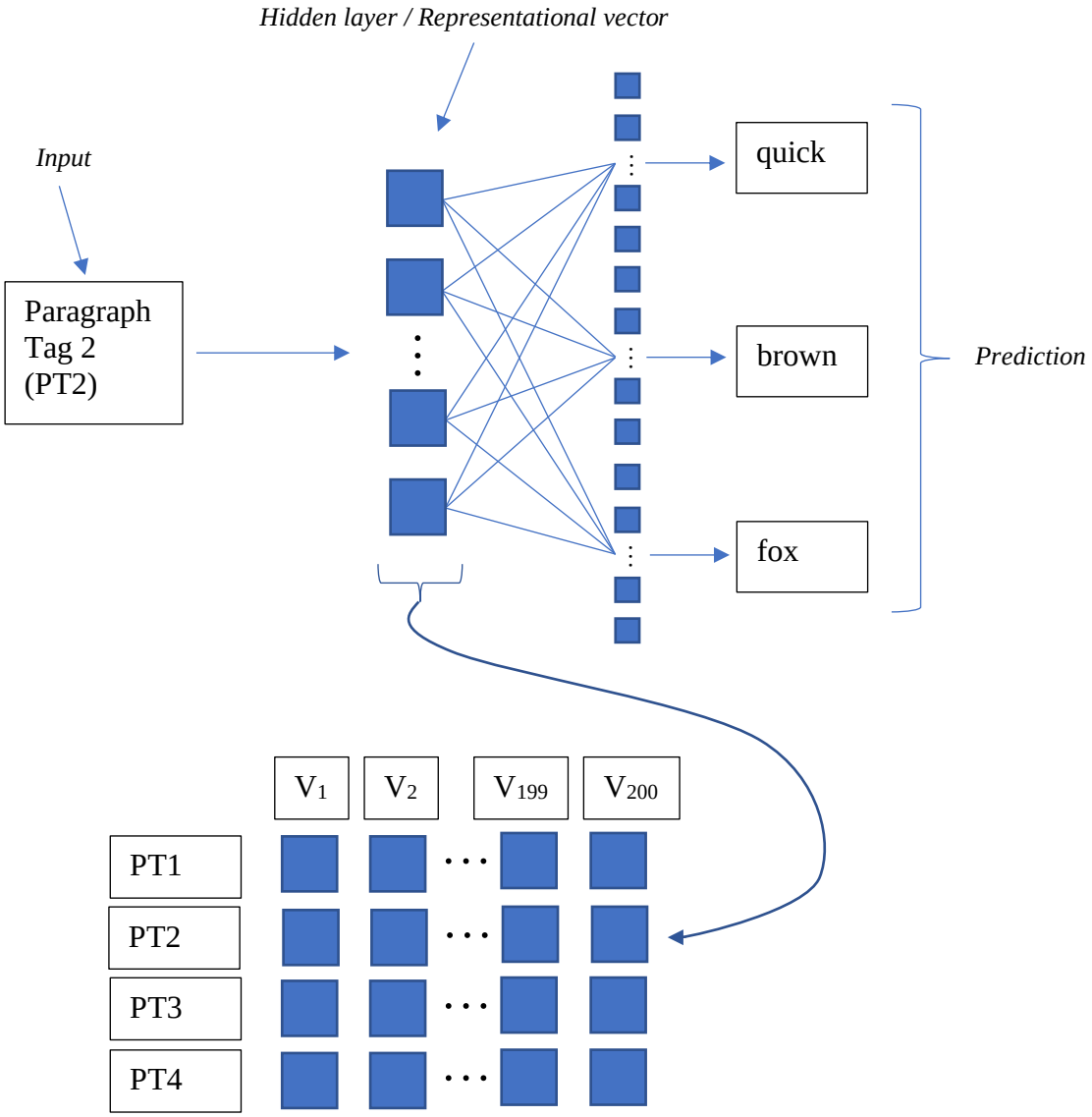

Figure 3: Doc2Vec model - distributed bag of words architecture - dm=0

WHERE HAS DOC2VEC BEEN USEFUL FOR FINANCIAL PREDICTIONS

3.4. S&P 500 time series analysis

As part of feature extraction - econometric time series analysis has been run on the S&P 500 data. The aim of this analysis was to find the linear model of best fit to the S&P 500 data and then include relevant autoregressive variables in the final dataset under the assumption that they will be relevant in the highly non-linear ML models. An initial analysis was done on the S&P 500 data after April 20th 1982 because opening and closing prices are differentiated between from that date onwards. This analysis found a large array of significant variables – more than half the days of the month, the month of September, a few specific years, 10 non-consecutive lags and the first lag of volume of trade. The significance of these variables,

particularly the days of the month were difficult to explain rationally. However, they did indicate persistence of volatility. This volatility persistence and the fact that volume is a strong indicator of absolute price change encouraged a second round of analysis that was done on all data following the initial recording of volume in 1950. This second round of analysis was focussed on finding an autoregressive conditional heteroskedasticity (ARCH) model that fit the data well. The analysis concluded with the selection of an ARMA (1,0) GARCH (1,1) model. Thus, the core variables selected for inclusion in the final data set are the first lag of standardized volume, the first lag of daily percentage change, the first residual of daily percentage change (DPC) predicted by an ARMA (1,0) model and the first lag of variance of the DPC.

3.4.1. 1982 onwards

Running time series analysis on the daily percentage change in the S&P 500 after April 20th 1982 has revealed 10 significant non-consecutive lags. The 'Daily percentage change 'variable was created by taking the percentage difference between the 'Close' and 'Open' variables for each day of the data. Before April 20th 1982 the 'Open' and 'Close' variables hold identical values, hence the time series analysis only being run after that date. As can be seen in Figure 1, 'Daily percentage change' is naturally stationary. This is supported by the results of an Augmented Dickey-Fuller unit root test which indicated that no unit root is present in the data. Running a partial autocorrelation function revealed 10 significant lags (these were lagged by 1, 2, 4, 12, 15, 16, 18, 27, 32, and 34 periods). Regressing 'Daily percentage change' on all 10 significant lags reinforced the finding by yielding significance above the 95% confidence interval for all 10 lags. Further, regressing the daily percentage change on weekday, monthday, month and year categorical variables yielded the significant correlations depicted in Table 1 (none of the weekdays were significant). Regressing on the categorical variables and the lags simultaneously yielded similar results with increased significance for 2002 (to the 99% level) and 2008 (to the 99,9% level) and the addition of 2018 (significant at the 95% level), further an extra 4 monthdays were deemed significant - bringing the total to 21 (out of 31) significant monthdays, finally, some of the significance levels on the lags changed. These statistics are depicted in Table 2. Adding a normalized (distributed standard normal) volume variable lagged by one period to the regression yields a significance at the 99% level on the lagged volume variable and alters the significance on the year variables as reported in Table 3.

While the years 2002 and 2008 are justifiable as significant because of the financial crashes that happened in each of those years (2002 – dot com bubble and 2008 housing bubble) and September is also relatable to the housing bubble; it is difficult to rationally justify the monthday variables as significant regardless of their quantitative significance. The high number of lags is also difficult to justify and implies rather that there may be persistence of volatility. Thus the following section focusses on the modelling of volatility over a time period that maximises the inclusion of volume statistics in the data. All of the analysis reported in this section was done in R using the packages 'stats', 'dplyr', 'urca', 'tidyverse', 'ggplot2' and 'fixest' (Marais, 2022).

Monthday	Month	Year	Lags
03 *	September *	2002 .	1 ***
04 *		2008 *	2 ***
07 **			4 **
09 ***			12 ***

10 *			15 ***
11 *			16 ***
12 .			18 *
15 *			27 **
17 *			32 *
19 ***			34 ***
20 **			
22 **			
23 **			
24 *			
25 .			
27 **			
30*			

Table 1: Significant variables to S&P 500 daily change – separate regressions

*Significance codes: 0 '***', 0.001 '**', 0.01 '*', 0.05 '.'*

Monthday	Month	Year	Lags
03 *	September *	2002 *	1 ***
04 *		2008 **	2 ***
06 .		2018 .	4 **
07 ***			12 ***
08 .			15 ***
09 ***			16 ***
10 .			18 *
11 *			27 **
12 .			32 .
14 .			34 ***
15 *			
17 *			
19 ***			
20 **			
21 .			
22 **			
23 **			
24 **			
25 .			
27 **			
30*			

Table 2: Significant variables to S&P 500 daily change – combined regression

*Significance codes: 0 '***', 0.001 '**', 0.01 '*', 0.05 '.'*

Monthday	Month	Year	% change Lags	Standardized volume lag
03 *	September *	2001 .	1 ***	1 **
04 *		2002 *	2 ***	
06 .		2008 *	4 **	

07 ***			12 ***	
08 .			15 ***	
09 ***			16 ***	
10 .			18 *	
11 *			27 **	
12 .			32 .	
14 .			34 ***	
15 *				
17 *				
19 ***				
20 **				
21 .				
22 **				
23 **				
24 **				
25 .				
27 **				
30*				

Table 3: Significant variables to S&P 500 daily change including stdVol

3.4.2. 1950 onwards – ARCH model

A second round of time series analysis was performed on all the data after Jan 1st 1950 which revealed that an ARMA(1,0) GARCH(1,1) model might be the most appropriate fit to the data and that the first lag on a date controlled version of standardized volume is a good predictor of DPC. The secondary analysis was performed because the number (and non-consecutiveness) of significant lags in the first model was out of the ordinary. The choice was made to extend the dataset back to 1950 because lagged volume appeared to be significant and volume data is available from that time onwards.

The extended dataset required a different dependent variable because open and closing prices were only differentiated between after April 20th 1982. Thus, the inter-day percentage change in closing price constituted the dependent variable in the second round of analysis. Figure 4 depicts the Daily Percentage Change (DPC) in the S&P 500 from 1950 until present – the persistence of volatility is again apparent. Figure 5 illustrates the correlation between the DPC and standardized volume which justifies the extension of the dataset.

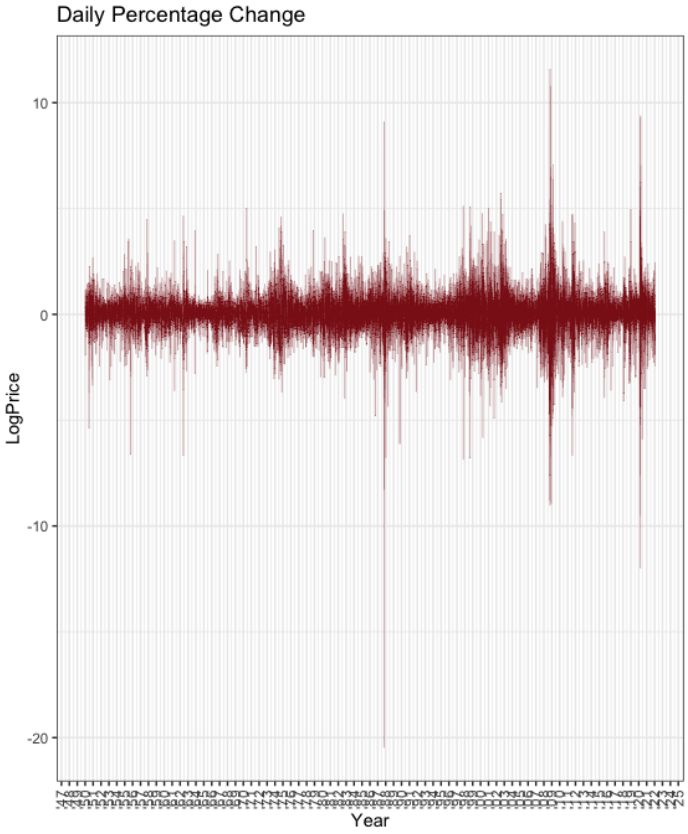

Figure 4: S&P 500 Daily Percentage Change 1950 – Present

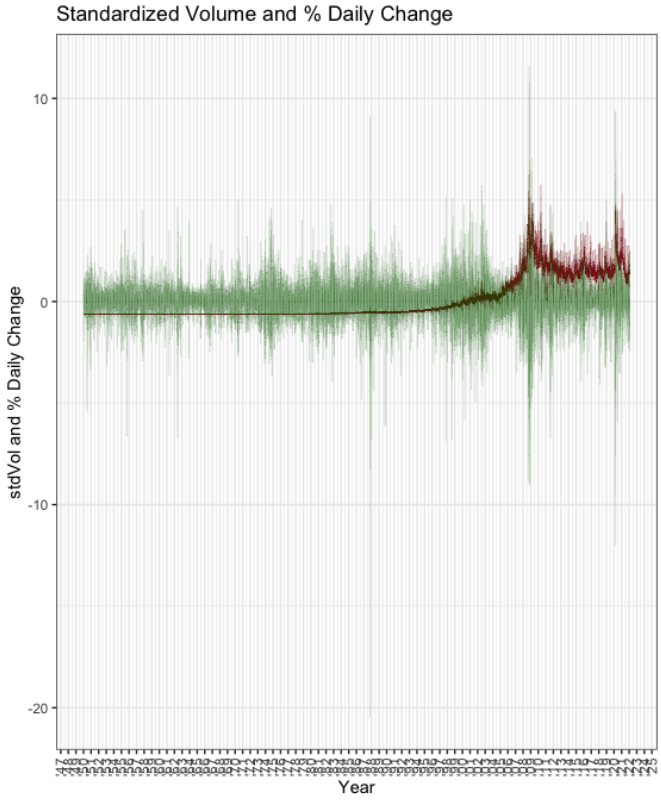

Figure 5: Standardized volume (red) and DPC (green)

3.4.2.1. Volume

Regressing the DPC on the first lag of standardized volume yielded insignificant results. However, regressing absolute DPC (pseudo-volatility) on the first lag of standardized volume (stdVol_1) yielded a coefficient of 0,148 significant above the 99,9th percentile. Figure 6 shows this relationship. Figure 6 also shows that standardized volume (stdVol) trends upwards over time. Further, regressing stdVol_1 on Date (regression 6) yields a positive coefficient significant above the 99,9th percentile and an adjusted R^2 of 0,6. This shows that a large percentage of the variation in stdVol is explained by variation in Date. Thus to avoid implicitly including a Date[8] proxy as a predictor of S&P 500 returns the included volume variable must be adjusted to be stationary over time. The residuals from regression 6 make up the variation in stdVol that cannot be explained by the variation in Date – thus they are a Date neutral measure of stdVol. Figure 7 shows the difference between stdVol and stdVol controlled for variance due to Date (stdVolControlled). Regressing the residuals of regression 6 on Date (regression 7) shows no significant relationship between the two while regressing absolute DPC on the residuals from regression 6 (regression 8) yields a coefficient of 0,164 significant above the 99,9th percentile. Figure 8 shows the relationship between absolute DPC and stdVolControlled. Thus these residuals (rather than stdVol) should be included as a variable in the final prediction data.

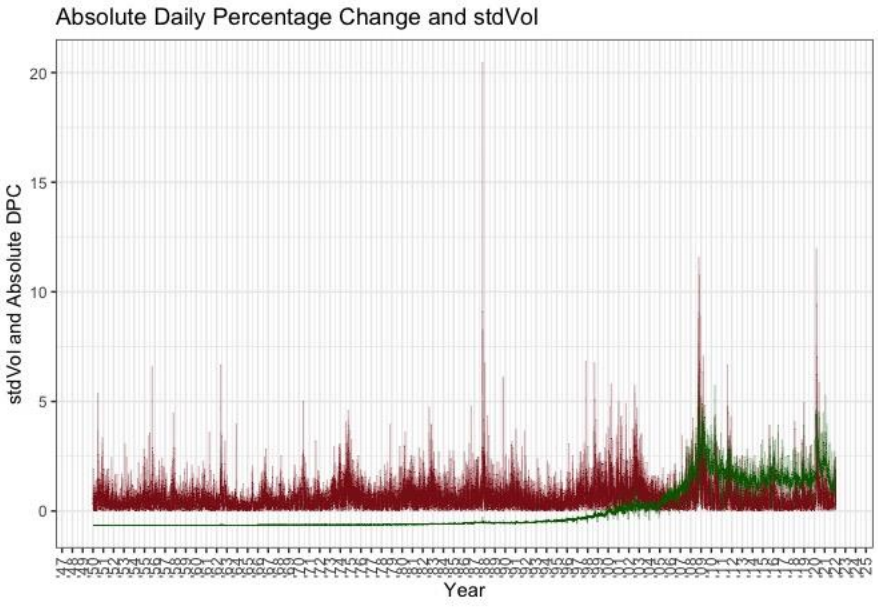

Figure 6:Absolute DPC and stdVol

[8] The goal of this project is to create a model that can predict future S&P 500 returns. Thus, to include a proxy for date (in this case in the date trending stdVol) would cause overfitting since dates do not hold any information relevant to the S&P 500 returns besides price or volatility trends.

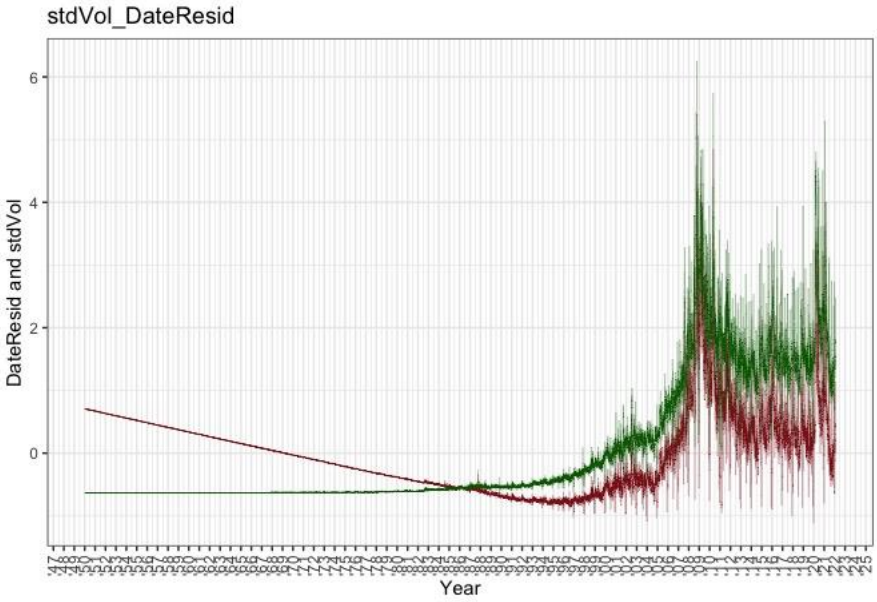

Figure 7: DateResid of StdVo (Red)l and stdVol (Green)

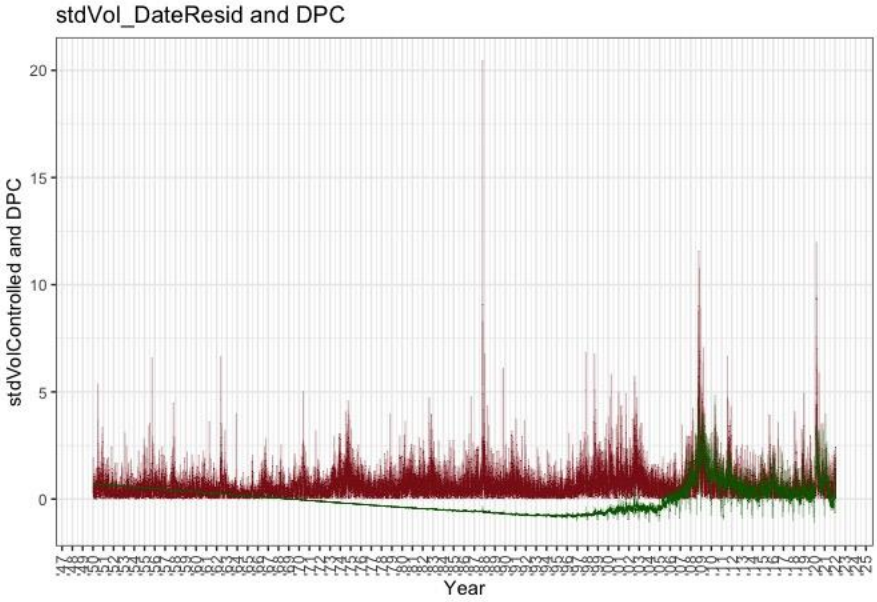

Figure 8: stdVolControlled and DPC

3.4.2.2. ARCH and GARCH testing

The LJung-Box test is a standard inference test in time series analysis. It tests the null-hypothesis that a series is a white noise series (Hassani & Yeganegi, 2020). Selection of the number of lags to be included in the Ljung-Box test is discussed by Hassani & Yenanegi (2020). Critical to this decision is the ratio of lags (H) to the number of observations (T) in the data set. If the size of H is too large the actual size of the test exceeds the nominal size of the test and the integrity of the test is violated. The dataset for this set of tests contains 18143 observations. The smallest (most difficult to satisfy) ratio permitted in the discussion is 0,001 suggested by Hyndman & Athanasopoulos, 2018). Thus the maximum number of lags that satisfy the smallest H/T ratio is 18 and any number of lags less than 18 is permissible.

Running a 10-Lag LJung Box test on the DPC yields a p-value of 0.0002 which indicates that the null hypothesis should be rejected and that the series is not a white noise series. I.e. ARCH effects are present in the series. Running autocorrelation and partial-autocorrelation tests on the absolute DPC indicates that there is a degree of persistence of volatility (Kotzé, 2021). This can be seen in Figure 9. A t-test indicates that the mean of the population is significantly different from zero (0.036). Thus, demeaning the equation constitutes the construction of a mean equation (essentially the demeaned DPC) represented in the series – a demeaned daily percentage change(DDPC) (Kotzé, 2020). A 10 lag Ljung-Box test again indicates the presence of ARCH effects specifically in the 1st,2nd, 5th, 6th, 8th, and 9th lags. However, running a multiple linear regression (OLS) of DDPC on its lags reveals only the 2nd, 6th, 8th and 9th lags as significant above the 95th percentile.

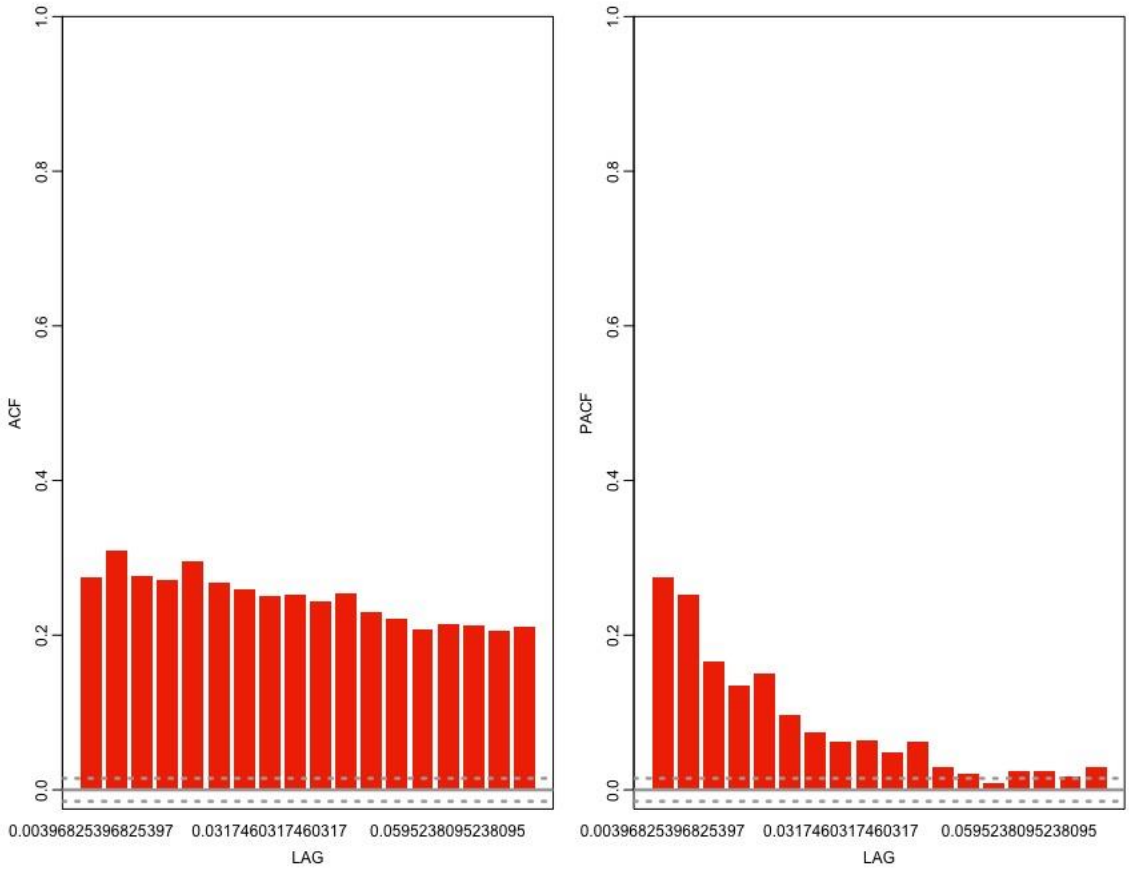

Figure 9: Autocorrelation and partial-autocorrelation functions of DPC

Seven possible models were checked for performance considering the above results. Model 1 is an ARIMA(0,0,1) , model 2 is an ARIMA(1,0,0), model 3 is an ARMA(1,0) GARCH(1,1), model 4 is a GARCH(1,1), model 5 is an EGARCH(1,1), model 6 is an ARMA(1,0) ARCH(1,0) and model 7 is an ARCH(1,0). The inclusion of an EGARCH model is motivated by its use in Nelson (1991), Blazsek & Mendoza (2016) and Tiwari, Raheem & Kang (2019) in their time series analyses of the S&P 500. Table 4 compares the coefficients from the 5 ARCH model variations (models 3, 4 and 5). Figure 10, 11 and 12, 13 and 14 show the results of autocorrelation and partial-autocorrelation functions of the residuals of the 5 ARCH variation models. Notably, all three of the models that do not contain an ARMA(1,0) component (models 4,5 and 7) yield a significant first residual indicating that inclusion of the ARMA(1,0) component is important. Further, model 6 – the non-generalized ARCH model –

has intermittent significant residuals indicating that volatility is clustered and persistent. Persistent volatility is one of the oversights of the ARCH model that is addressed in the GARCH model . As can be seen, Model 3 (Figure 10) yields the least significant residuals and as such has been selected as the model that best fits this time series. The model linearly predicts the dependent variable using the first lag of the dependent variable, the first residual of an ARMA(6,0) model and the first lag of variance in the time series. As such the 1st , 2nd and 6th lags of DDPC and the first lag of variance of DPC will be included in the final dataset (Kotzé, 2020). Additionally, as illustrated in Aras (2021) the inclusion of the GARCH model prediction can be beneficial to ML model training, thus the predictions of model 3 will also be included in the dataset.

	Model 3	Model 4	Model 5	Model 6	Model 7
μ	1	1	1	1	1
ar1	<2e-16 ***			6.03e-05 ***	
ω	<2e-16 ***	4.86e-14 ***	0.21316	< 2e-16 ***	< 2e-16 ***
α1	<2e-16 ***	<2e-16 ***	~ 0 ***	< 2e-16 ***	< 2e-16 ***
β1	<2e-16 ***	<2e-16 ***	~ 0 ***		
γ1			~ 0 ***		

Table 4: Significance levels of variations of ARCH models from TS analysis 2

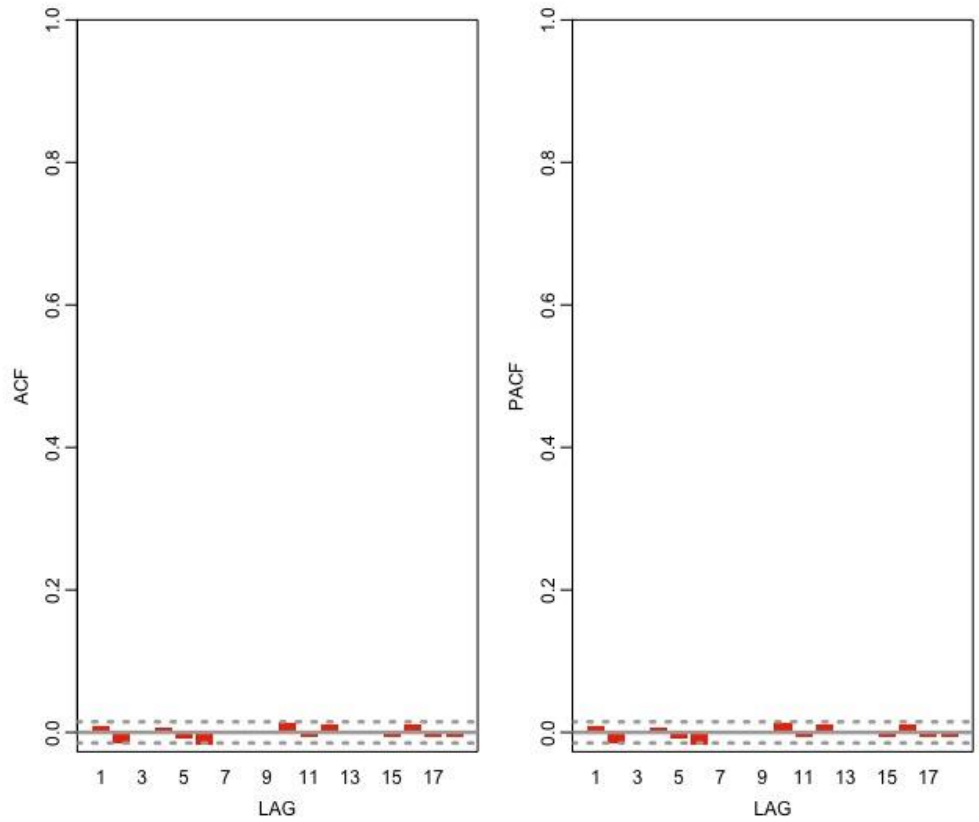

Figure 10:Model 3 residual ACF and PACF - ARMA(1,0) GARCH(1,1)

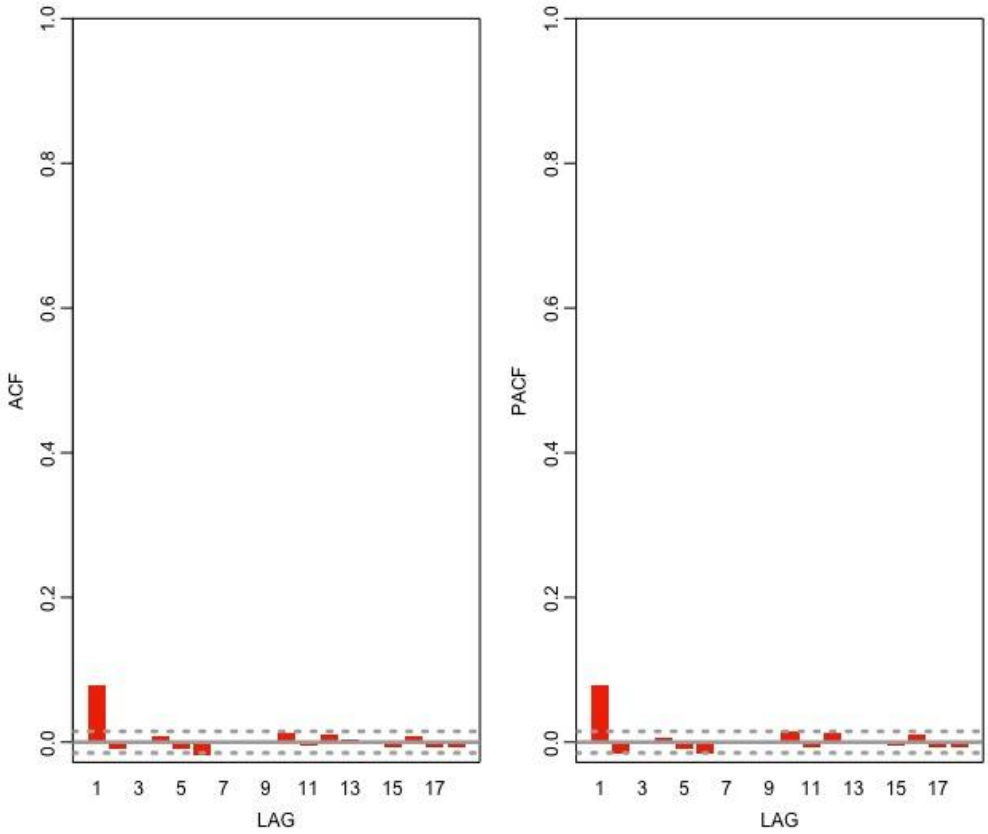

Figure 11: Model 4 residual ACF and PACF - GARCH(1,1)

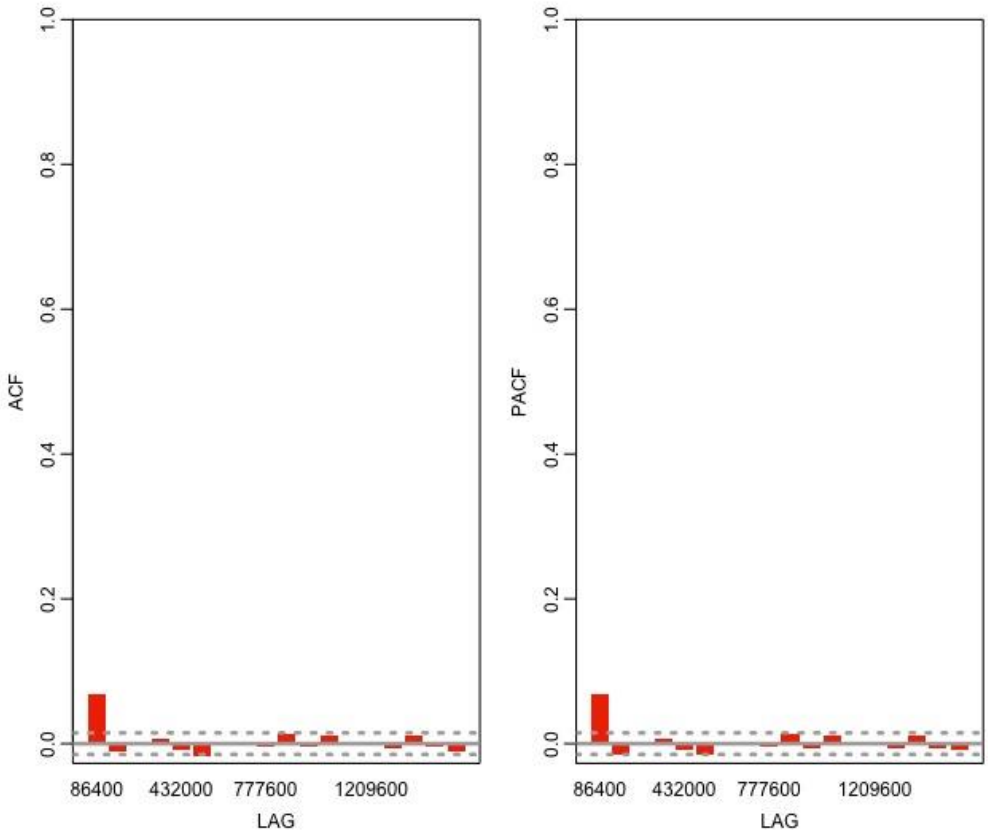

Figure 12: Model 5 residual ACF and PACF - EGARCH(1,1)

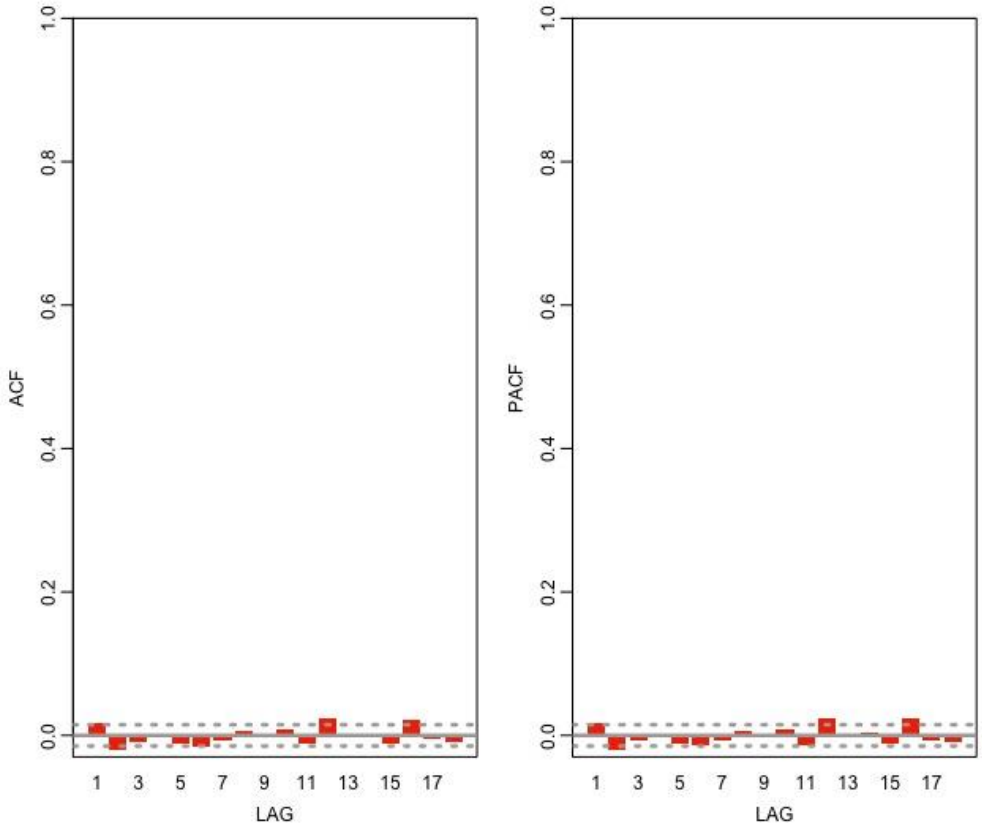

Figure 13: Model 6 residual ACF and PACF - ARMA(1,0) ARCH(1,0)

3.4.2.3. Additional autoregressive variables

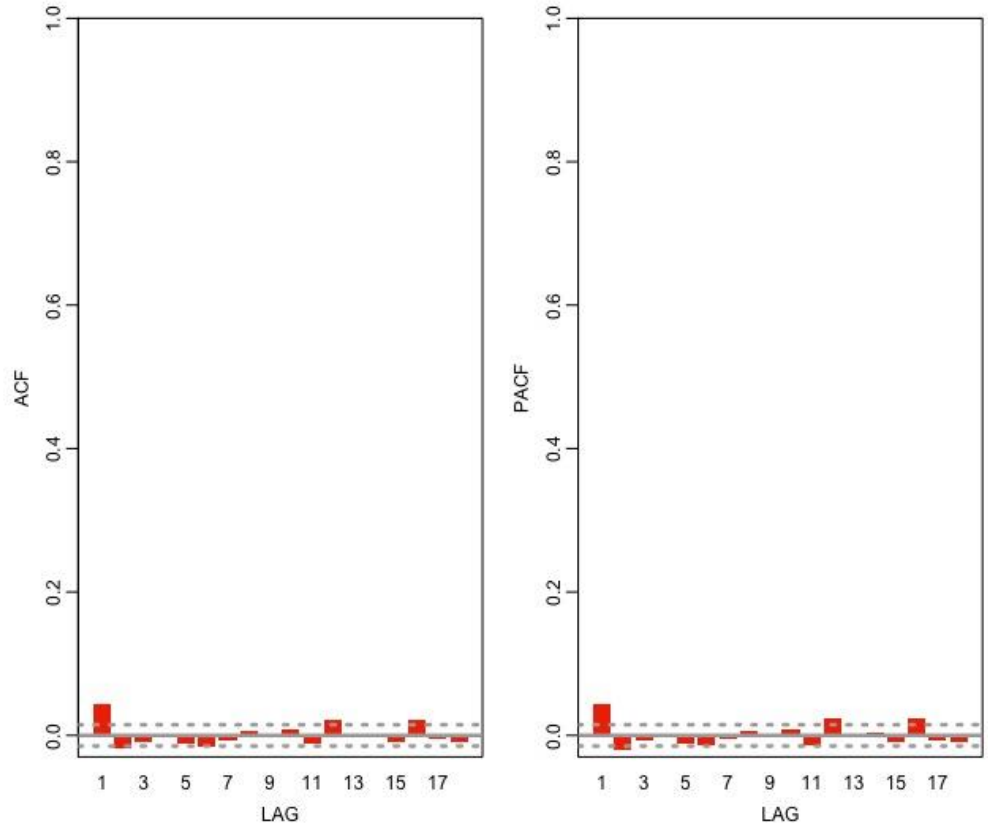

Figure14: Model 7 residual ACF and PACF - ARCH(1,0)

3.5. Meta financial data

Six S&P 500 adjacent indicators were selected for this study. Notably they were only included in the speech centred dataset. The six series included were Bitcoin prices , the NASDAQ composite index, the oil price, the Shanghai Stock Exchange composite index (SSE), the Dollar strength index and the CBOE volatility index (VIX).

Include reasons for these selections.

3.6. Summary of data collection and preparation

The data preparation phase of this project consisted of four sections. These were, data collection, data cleaning, text data feature engineering and financial data feature engineering. Data collection and cleaning consisted of web scraping, removing HTML tags, converting to lowercase, stopword removal and lemmatization for the text data. Collection of the financial data is a done via a Python script that will always download the entire history of the S&P 500. Cleaning requires only the removal of the adjusted close variable. Text data feature engineering took the form of Textblob sentiment analysis, VADER sentiment analysis, Word2Vec vectorization and Doc2Vec vectorization. The results of which were 2 sentiment descriptors from Textblob, 4 sentiment descriptors from VADER, a 200 element average descriptor vector from Word2Vec, a 200 and a 20 element vector speech descriptor from Doc2Vec for the speech centred data, and two 21 element vector speech descriptors from Doc2Vec for the date centred data for a total of 426 variables (features) engineered from the text data for the speech centred data and 42 variables engineered from the text data for the

date centred data. The financial data analysis took the form of time series econometrics. This analysis resulted in the fitting of an ARMA(1,0) GARCH(1,1) model to the engineered daily percentage change variable. The model linearly predicts the dependent variable using the first lag of the dependent variable, the first residual and the first lag of variance in the time series. Thus the first lag of daily percentage change and the first lag of variance of daily percentage change are included in the data. Additionally, the prediction of the ARMA(1,0) GARCH(1,1) model and the first lag of standardized volume of trade are included. Thus 4 variables (features) have been engineered using time series analysis of the S&P 500 data.

4. Experiment Design

Two sets of experiments were run on two categories of data design. The first is a speech centred design. For every speech in this dataset there is a row of data attached. Thus there is no data for dates when no speeches occurred and there are duplicate dates because there are days when more than one speech occurred. This dataset is the larger of the two datasets and contains 35 251 rows of data – one for each unique speech since 1998-01-01.

Based on the assumption that every speech that occurs on a day affects the closing price of the S&P 500 on that day - he second dataset uses a date centred design. For every date (trading day) there is one row in the dataset. Thus on dates when more than one speech occurred the speech data has been aggregated into a single vector (see the Doc2Vec methods section for an explanation of this method). The date centred dataset contains 25 383 rows of data – one for each trading day between 1950-01-01 and 2022-03-22. The speech centred and date centred datasets are exemplified in Table 1 and 2, respectively.

Date	Speech	LD_Date_Resid_1	V1	V2	V3
1950-01-02	Good morning…	0,02	3,6	2,6	1
1950-01-06	Ladies and…	-0,05	2,1	2	0,36
1950-01-06	Congress…	-0,05	0,12	3,4	2,8
…					

Table 5:Example of the speech centred data design – Note that there is no data for dates that no speeches occurred and that there are duplicate dates. Note also that on dates where more than one speech occurred there is duplicate data in the LD_Date_Resid_1 column which is representative of the autoregressive data in this example. Finally, note that the V1 and V2 columns (representative of the vectorized speech data) is different for every row of data.

Date	Speech	LD_Date_Resid_1	Days since last speech	V1	V2
1950-01-02	Good morning…	0,02	0	1,2	1,8
1950-01-03	N/A	0,025	1	1,2	1,8
1950-01-04	N/A	-0,03	2	1,2	1,8
1950-01-05	(Multiple speeches)	0,05	0	2,1	-0,6
1950-01-06	Today marks…	-0,03	0	5,3	-0,86
1950-01-09	N/A	0,04	3	5,3	-0,86
...					

Table 6:Example of the date centred data design – Note that every consecutive trading day is included in the data. Note also that on days when no speeches occurred the V1 and V2 (representing vectorized speech data) from the last date that a speech occurred are duplicated and the number of days since the last speech is recorded. Finally, note that on days when multiple speeches occurred there is only one row of data – i.e. the speeches are amalgamated into a single vector.

For the speech centred dataset a total of 1152 models were tested. These consisted of 384 regression tasks and 768 classification tasks. While for the date centred data 336 models were tested. These consisted of 112 regression models and 224 classification models. The models were selected by incrementally altering a single hyperparameter across 5 fields for regression and 6 fields for classification. These fields are detailed in the following sections.

4.1. Regression fields

The 5 fields of hyperparameters for regression were StartDates, Remove_duplicates, Reg_Types, Reg_algos and Datasets. StartDates refers to the date from which the dataset began: 1998-01-01, 2000-01-01 and 2010-01-01. The 1998 dataset runs from 1998-01-01 until 2022-03-22, the 2000 dataset runs from 2000-01-01 until 2022-03-22 etc.

The Remove_duplicates field divides the speech centred data into two subsets – one with all the duplicate dates removed and one including the duplicate dates. Duplicate dates occurred because some speeches occurred on the same date. This hyperparameter is necessary because the Meta and Auto datasets only contain one value in each field – thus duplicating dates duplicates Meta and Auto data which creates the risk of data contamination between the train and test sets. This field is not valid for the dates centred dataset and is set to 'False' for all the models using it.

The Reg_Types field contains two options, TS_Regressor and CS_Regressor. These refer to the manner in which the data is split for cross validation. The TS_Regressor cross validation method splits the data into 5 time consecutive subsets using the 'TimeSeriesSplit' out of sample (OOS) method from Sci Kit Learn. Model training is done on the first subset, then the first and second subset etc. up until the fourth subset. Training scores are calculated for each of the four training subsets and the maximum score is passed forward as the final representation of the training score. Test scores are calculated for only the fifth (and latest) split. This method is employed to force the algorithm to perform prediction instead of interpolation. It avoids giving the models access to the future of a data trend when making predictions for a data point. However, the method is not strictly necessary in this case because the dependent variable has been centred and is close to normally distributed – i.e. it does not have a trend (Bergmeir, Hyndman & Koo, 2018). The CS_Regressor option performs regular 5-fold cross validation.

The Reg_algos field refers to the four different ML algorithms available for training. These are stochastic gradient descent, a 3 hidden layer neural network, multiple linear regression and gradient boosting. The hyper-parameters for each of these fields can be found in Appendix B.1.

Datasets is a complicated field. There are three groups of prediction sub-datasets in the speech centred dataset – the control subset: X_control ; the meta subset: X_meta ; and the test subset: X_test. X_control contains the autoregressive variables described in section 3.4.2.3 and 3.4.2.4, X_meta contains the meta variables (other possibly relevant financial variables, such as the dollar strength index) and X_test contains the variables of focus – the NLP variables. However, in total there are 458 variables across these three subsets. The X_test

subset makes up the majority of this. There are 4 variables derived from the VADER sentiment analysis, 2 variables from the textBlob sentiment analysis, 200 variables from Word2Vec, 200 variables from Doc2Vec_200 and 20 variables from Doc2Vec_20. In order to minimize training times and avoid multi-collinearity the X_control dataset was reduced to contain only 8 variables, X_meta was reduced to contain 5 variables and X_test was reduced to contain 26 variables. (Should run an elastic net or lasso to deduce the datasets selected for this section).

The date centred dataset also contains three groups of prediction subsets. These are the same X_control dataset as in the speech centred data, and two versions of the vectorised speech data – the distributed bag of words (DBOW) subset and the distributed memory (DM) subset – both detailed in section 3.3.2.2. A detailed description of the variables available and tested is available in Appendix B.3.

The final X_control variables selected for both the speech centred and the date centred datasets were DlogDif_1, DlogDif_2, absDlogDif_1, blackSwan_SD3_1, blackSwan_SD4_1, blackSwan_SD5_1, stdVol_1DateResid and pos_neg_transform. The final X_meta variables selected were Nasdaq_ld_1, Oil_ld_1, SSE_ld_1, USDX_ld_1 and VIX_ld_1. The final X_test variables selected were the VADER scores, the TextBlob scores and the set of 20 Doc2Vec variables. While the DBOW and DM datasets each contained all 20 of their variables and the number of days since the last speech.

The datasets field provided the option for any combination of each data designs' three subsets as training variables for a sum of 7 combinations each. Additionally a PossibleBest set of variables was selected for the speech centred dataset using an Elastic Net algorithm from SciKitLearn (hyperparameters available in appendix B). This PossibleBest dataset consisted of DlogDif_1, DlogDif_2, pos_neg_transform, Nasdaq_ld_1, Oil_ld_1, VIX_ld_1, DV_20_6, DV_20_8, DV_20_13, DV_20_15.

4.2. Classification fields

The 5 fields of hyperparameters for classification were StartDates, Remove_duplicates, Binary, Clf_Types, Clf_algos and Datasets. StartDates, Remove_duplicates and Datasets are identical to their equivalents in the regression section above.

Binary refers to the Y-variable (classification fields) being predicted. The options for Binary are True and False. If False is selected the continuous Y-variable is split into 8 categories denoted by the numbers 1-8. These categories represent a number of standard deviations from the mean of the input Y-variable. See Equation 1. If Binary is set to True then the Y-variable is split into 2 categories denoted by 1 and 0 which indicate whether the entry is above or below the mean of the continuous Y variable. See Equation 2.

$$Y = \begin{cases} 1\ where\ \mu - 2\sigma > Ycont \\ 2\ where\ \mu - \sigma > Ycont > \mu - 2\sigma \\ 3\ where\ \mu - 0{,}5\sigma > Ycont > \mu - \sigma \\ 4\ where\ \mu > Ycont > \mu - 0{,}5\sigma \\ 5\ where\ \mu < Ycont < \mu + 0{,}5\sigma \\ 6\ where\ \mu + 0{,}5\sigma < Ycont < \mu + \sigma \\ 7\ where\ \mu + \sigma < Ycont < \mu + 2\sigma \\ 8\ where\ \mu + 2\sigma < Ycont \end{cases}$$

$$; where\ \sigma = standard\ deviation\ of\ Ycont \\ and\ \mu = mean\ of\ Ycont$$

Equation 1.: Conversion of continuous Y variable to a non-binary categorical variable

$$Y = \begin{cases} 1\ where\ Ycont > \mu \\ 0\ where\ Ycont < \mu \end{cases}$$

$$; where\ \sigma = standard\ deviation\ of\ Ycont \\ and\ \mu = mean\ of\ Ycont$$

Equation 2: Conversion of continuous Y variable to a binary categorical variable

Clf_Types refers to the Time Series cross validation and Cross-section cross validation methods as described in the regression section. The options in this category are CS_Classifier and TS_Classifier. Clf_algos is very similar to the Reg_algos field described in the previous section. The algorithms available for classification are stochastic gradient descent, a 3 hidden layer neural network, logistic regression and gradient boosting. The hyper-parameters for each of these fields can be found in Appendix B.2.

5. Results and discussion

The results of the models indicated that U.S. Presidential Speeches hold at least a small amount of predictive power over movements in the S&P 500 stock index. This conclusion was deduced from the fact that the best performing datasets in both the date centred classification analysis and the date centred regression analysis included vectorized speech data. It is also corroborated by the consistent appearance of speech subset inclusive data in the top ten performing models across both data design types, and both classification and regression tasks.

The date centred dataset outperformed the speech centred dataset by about 2 percentage points (~0,58 vs ~0,6) on the test set accuracy score for binary classification analysis. Models using the date centred dataset also achieved lower MAE's than models trained and tested on the speech centred dataset. Interestingly, an implication of the superior performance of the (smaller) date centred dataset is that data with high quality has outperformed data with high quantity. The section below explains the results in more detail.

5.1. Speech centred classification analysis

Analysing the best performing classification models, in terms of test set accuracy, across all categories of the speech centred data reveals a very high likelihood of data contamination in the data containing duplicate dates. All ten of the top performers are on data containing duplicates and the accuracies range from 0,72 to 0,80. It is highly unlikely that these models have managed to achieve 80% accuracy in prediction of detrended S&P 500 on unseen data over a 52 year period. Thus results of models trained and tested on duplicate date containing data is discarded henceforth.

Of the duplicate removed results the top ten performers contain seven models trained and tested on the 2010 dataset – including the top four. Further, eight of the top ten models performed binary classification. Finally seven of the ten were trained on cross-sectional cross validation. Thus further analysis will take place within the binary, cross-sectional and 2010 categories. The test set accuracy ranges from 0,54 to 0,58. The mean of the target variable is ~ 0,5 and thus every model in the top ten models outperforms chance (recall that the target variable is binary). NLP data is included in six of the ten best models, while autoregressive data is included in five and meta data is included in eight of them. Notably an NLP only (gradient boosting) model ranks fourth with a test set accuracy of 0,56. This constitutes evidence of the predictive power of US Presidential speeches over S&P 500 movements and lends credibility to the hypothesis of this project.

5.2. Date centred classification analysis

The means of both the full set of 1950-01-01 binary Y's and the test set of 1950-01-01 binary Y's are maintained at ~ 0,5. This implies that any accuracy above 0,5 beats chance.

In the date centred classification results analysis, the top ten models in terms of test set accuracies contain ten 1950 subsets, ten TS subsets and eight binary subsets. All of the top ten models utilised autoregressive data, five positions included the DBOW vector set and two included the DM vector set. The range of the top ten models test set accuracies is 0,57 to 0,59. Thus the date centred dataset initially outperforms the (duplicate dates removed) speech centred dataset by 2 percentage points at the bottom of the range and 1 percentage point at the top of the range in terms of test set classification accuracy. Because of this the models trained on the date centred dataset were optimised for maximum performance. Log-regression is initially the best performing classifier in the TS section generally taking the top two and the seventh positions. However, AutoDBOW outperforms Auto in the Binary NN and the Binary Gradient Boosting categories. Thus, these three models were selected for optimisation.

5.2.1. Optimization of classification models for TS classification of date centred data

The training set accuracy on the 1950 binary AutoDBOW for the gradient boosting classifier was initially 77,5% indicating that it may have been slightly overfitting the training data. For the NN the training accuracy is 59% which is similar to the test set accuracy and indicates a good fit for the data. Grid search was manually performed to optimize NN and Gradient Boosting for the binary 1950 AutoDBOW and Auto datasets.

After closer optimization of hyper-parameters it becomes clear that the AutoDBOW dataset outperforms the Auto dataset on test score both overall and across both the Stochastic Gradient Descent and Log Regression algorithms. The best performing algorithm for the AutoDBOW dataset was the LogReg_4 algorithm (please see appendix B for hyperparameters) which achieved a test set accuracy of 0,601 while the best performing algorithm for the Auto dataset was the NN_7 algorithm which achieved a test set accuracy of

0,599. This difference indicates that there is at least a slight benefit to including the DBOW dataset in predictive data and corroborates the evidence presented in section 2.1.

5.3. Speech centred regression analysis

The average Mean Absolute Error (MAE) recorded across all 384 regression models run was 0,87 for the training data, 0,95 for the test data and 0,8 for the validation data. Comparing these with a standard deviation of 1,26 for the *logDif_date_resid* variable in the 1998 subset shows that the average absolute error across all the regressions run lies within one standard deviation of the dependent variable indicating that the regressions are better predictors of the series than the mean of the series. This holds across each of the three date subsets (1998, 2000, and 2010).

The top ten performing models in regression tasks were all trained on the cross-sectional 2010 dataset and achieved MAE's between 0,68 and 0,7. Interestingly, there was an even split between datasets with duplicates removed and without. The best performer did not have duplicates removed whilst the second, third and fourth best performers did. Analysis of only data with duplicates not removed gave the top four places to gradient boosting models trained on the AutoMeta, All, Meta and Auto datasets (in that order). The MAEs for the top four performers ranged from 0,68 to 0,7. NLP inclusive datasets ranked in five of the top ten places. Analysis on the duplicates removed top ten performers shows the NN taking 8 of the top ten places – including the top seven spots. NLP inclusive datasets ranked well taking the second, third, fourth, fifth and seventh spots but not beating the purely autoregressive data.

It is difficult to draw a solid conjecture about the predictive power of the NLP data from these mixed results. However, data contamination in the duplicate inclusive dataset is likely and the results should probably be discarded. Given the fair performance of the NLP inclusive datasets in the duplicates removed dataset, at this point, it remains likely that the NLP data holds predictive power.

A final analysis of all the regression models trained on only 1998 NLP data from the speech centred dataset shows that the top ten performing models all achieved a MAE below 1,06. This is still below the standard deviation of 1,26 for the 1998 Y variable (*logDif_date_resid*) and thus indicates that regression models trained only on the NLP data outperform the mean of the series as a predictor. This again corroborates evidence that U.S. Presidential Speeches having predictive power over S&P 500 movements.

5.4. Date centred regression analysis

The initial ten lowest test set MAE's across all the regression categories were in the 1950 and cross section categories. The standard deviation for the 1950 test set is 0.96. All ten of these MAEs clustered around 0,66 so the regressions are better predictors than the mean. Again the date centred dataset has outperformed the speech centred dataset. The best score was achieved for the DBOW dataset by the SGD algorithm. The subsets including NLP data were included in seven of the top ten performing models and three of them only contained NLP data. Notably, the AutoDM subset outperformed the Auto subset in the NN model - undeniably indicating the predictive power of the NLP data.

NN algorithms were responsible for six of the lowest ten MAEs and SGD models were responsible for a further 3. The DBOW dataset appeared twice, the Auto dataset 3 times, the DM dataset once, AutoDBOW twice, AutoDM once and AutoBoth once. In total the Auto dataset appeared in 7 of the top ten performers, the DBOW appeared in 5 and the DM dataset

appeared 3 times. Given this and the predictive ability of the DBOW shown in the classification section above, DBOW and Auto data sets were compared across attempted optimizations of the NN and SGD algorithms. However, no significant improvement could be engineered and almost all models performed worse than the initial models. Given that the best performing dataset in terms of test MAE was the DBOW dataset, further evidence is presented that U.S. presidential Speeches hold predictive power over the S&P 500.

6. Conclusion

7. Bibliography

Agarwal, A. 2020. Sentiment Analysis of Financial News. In *12th International Conference on Computational Intelligence and Communication Networks*.

Aras, S. 2021. Stacking hybrid GARCH models for forecasting Bitcoin volatility. *Expert Systems with Applications*. 174. DOI: 10.1016/j.eswa.2021.114747.

Baker, S.R., Bloom, N. & Davis, S.J. 2016. Measuring economic policy uncertainty. *Quarterly Journal of Economics*. 131(4):1593–1636. DOI: 10.1093/qje/qjw024.

Bergmeir, C., Hyndman, R.J. & Koo, B. 2018. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*. 120:70–83. DOI: 10.1016/j.csda.2017.11.003.

Biswas, S., Sarkar, I., Das, P., Bose, R. & Roy, S. 2020. Examining the Effects of Pandemics on Stock Market Trends through Sentiment Analysis. *Journal of Xidian University*. 14(6). Available: https://www.researchgate.net/publication/342083661.

Blazsek, S. & Mendoza, V. 2016. QARMA-Beta-t-EGARCH versus ARMA-GARCH: an application to S&P 500. *Applied Economics*. 48(12):1119–1129. DOI: 10.1080/00036846.2015.1093086.

Chudy, A. n.d. *Text sentiment analysis with textblob*. Available: https://deepnote.com/@andrej/Text-sentiment-analysis-with-textblob-n34b1QiTQ1uyVLvOMLTGiw [2022, January 07].

Dilai, M., Onukevych, Y. & Dilay, I. 2018. Sentiment Analysis of the US and Ukrainian Presidential Speeches. In *Computational linguistics and intelligent systems (2)*. V. II: Workshop. Lviv. Lviv, Ukraine: Lviv Polytechnic National University. 60–70. DOI: http://ena.lp.edu.ua:8080/handle/ntb/42572.

Elbagir, S. & Yang, J. 2019. Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2019 IMECS 2019, March 13-15, 2019, Hong Kong*. 575.

Géron, A. 2019. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc.

Hassani, H. & Yeganegi, M.R. 2020. Selecting optimal lag order in Ljung–Box test. *Physica A: Statistical Mechanics and its Applications*. 541. DOI: 10.1016/j.physa.2019.123700.

Hayo, B.;, Kutan, A.M.; & Neuenkirch, M. 2008. *Communicating with many tongues: FOMC speeches and US financial market reaction*. Available: http://hdl.handle.net/10419/30104www.econstor.eu.

Hutto, C.J. & Gilbert, E. 2014. *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. Available: http://sentic.net/.

Hyndman, R.J. & Athanasopoulos, G. 2018. *Forecasting- principles and practice*. OTexts.

Jiao, Y. & Jakubowicz, J. 2017. *Predicting Stock Movement Direction with Machine Learning: an Extensive Study on S&P 500 Stocks*. Institute of Electrical and Electronics Engineers.

Katre, P.D. 2019. NLP based text analytics and visualization of political speeches. *International Journal of Recent Technology and Engineering*. 8(3):8574–8579. DOI: 10.35940/ijrte.C6503.098319.

Khedr, A.E., Salama, S.E. & Yaseen, N. 2017. Predicting stock market behavior using data mining technique and news sentiment analysis. *International Journal of Intelligent Systems and Applications*. 9(7):22–30. DOI: 10.5815/ijisa.2017.07.03.

Kinyua, J., Mutigwe, C., Cushing, D. & Poggi, M. 2021. An analysis of the impact of President Trump's tweets on the DJIA and S&P 500 using machine learning and sentiment analysis. *Journal of Behavioural and Experimental Finance*. DOI: 10.1016/j.jbef.2020.100447.

Kotzé, K. 2020. Available: https://kevin-kotze.gitlab.io/tsm/ts-8-note/ [2022, February 23].
Kotzé, K. 2021. Available: https://kevinkotze.github.io/ts-12-tut/ [2022, February 09].
Le, Q. & Mikolov, T. 2014. *Distributed Representations of Sentences and Documents*.
Liu, C., Wang, J., Xiao, D. & Liang, Q. 2016. Forecasting S&P 500 Stock Index Using Statistical Learning Models. *Open Journal of Statistics*. 06(06):1067–1075. DOI: 10.4236/ojs.2016.66086.
Maligkris, A. 2017. *Political Speeches and Stock Market Outcomes*. Miami.
Marais, W. 2022. *Introduction to R.* Available: https://github.com/WihanZA/Intro_to_R_2022 [2022, February 02].
Mikolov, T., Chen, K., Corrado, G. & Dean, J. 2013a. *Distributed Representations of Words and Phrases and their Compositionality*. Available: https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf [2022, May 31].
Mikolov, T., Chen, K., Corrado, G. & Dean, J. 2013b. *Efficient Estimation of Word Representations in Vector Space*. Available: http://ronan.collobert.com/senna/.
Nelson, D.B. 1991. *Conditional Heteroskedasticity in Asset Returns: A New Approach*.
Pano, T. & Kashef, R. 2020. A complete vader-based sentiment analysis of bitcoin (BTC) tweets during the ERA of COVID-19. *Big Data and Cognitive Computing*. 4(4):1–17. DOI: 10.3390/bdcc4040033.
Purevdagva, C., Zhao, R., Huang, P. & Mahoney, W. 2020. A machine-learning based framework for detection of fake political speech. *IEEE 14th International Conference on Big Data Science and Engineering*. DOI: 10.1109/BigDataSE50710.2020.00019.
Qin, C. & Ji J. 2018. *Natural Language Processing and Event-driven Stock Prediction*. Available: http://cs230.stanford.edu/projects_spring_2018/reports/8290001.pdf [2022, February 28].
Řeh uřek, R. & Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA. 45–50.
Ren, R., Wu, D.D. & Liu, T. 2019. Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Systems Journal*. 13(1):760–770. DOI: 10.1109/JSYST.2018.2794462.
Sazedj, S. & Tavares, J. 2011. HOPE, CHANGE, AND FINANCIAL MARKETS: CAN OBAMA'S WORDS DRIVE THE MARKET? *Centre for Economic Policy Research: Financial Economics and Public Policy Discussion Paper Series*. (8713). Available: www.cepr.org.
Shelar, A. & Huang, C.Y. 2018. Sentiment analysis of twitter data. In *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*. Institute of Electrical and Electronics Engineers Inc. 1301–1302. DOI: 10.1109/CSCI46756.2018.00252.
Shi, B., Zhao, J. & Xu, K. 2019. A Word2vec model for sentiment analysis of Weibo. In *16th International Conference on Service Systems and Service Management (ICSSSM)*. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8887652&casa_token=2CYAr-GgtaMAAAAA:QI03O84FaMMLu2hT03Ysq8IWROXj5OcQu0zFcbAXfiFnDCNvcKBdGTOQp3Xc1gkO3zmPKk7bock_ [2022, February 28].
Sohangir, S., Petty, N. & Wang, Di. 2018. Financial Sentiment Lexicon Analysis. In *Proceedings - 12th IEEE International Conference on Semantic Computing, ICSC 2018*. V. 2018-January. Institute of Electrical and Electronics Engineers Inc. 286–289. DOI: 10.1109/ICSC.2018.00052.

Sohangir, S., Wang, D., Pomeranets, A. & Khoshgoftaar, T.M. 2018. Big Data: Deep Learning for financial sentiment analysis. *Journal of Big Data*. 5(1). DOI: 10.1186/s40537-017-0111-6.
Tiwari, A.K., Raheem, I.D. & Kang, S.H. 2019. Time-varying dynamic conditional correlation between stock and cryptocurrency markets using the copula-ADCC-EGARCH model. *Physica A: Statistical Mechanics and its Applications*. 535. DOI: 10.1016/j.physa.2019.122295.
Vargas, M.R., de Lima, B.S.L.P. & Evsukoff, A.G. 2017. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, CIVEMSA 2017 - Proceedings*. Institute of Electrical and Electronics Engineers Inc. 60–65. DOI: 10.1109/CIVEMSA.2017.7995302.
Woolley, J. & Peters, G. n.d. *The American Presidency Project*. Available: https://www.presidency.ucsb.edu/ [2022, January 05].
Yahoo Finance. n.d. *Yahoo Finance: S&P 500*. Available: https://finance.yahoo.com/quote/%5EGSPC/history?period1=-1325635200&period2=1641340800&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true [2022, January 05].
Zubair, S. & Cios, K.J. 2015. Extracting news sentiment and establishing its relationship with the S&P 500 index. In *Proceedings of the Annual Hawaii International Conference on System Sciences*. V. 2015-March. IEEE Computer Society. 969–975. DOI: 10.1109/HICSS.2015.120.