



# News Sentiment in Volatility predictions

*Exploring the effect of news sentiment on stock volatility using machine learning regression models*

**Jacob Hagan, Rolf Tynning Henriksen**

**Supervisor: Sondre Nedreås Hølleland**

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

This thesis was written as a part of the Master of Science in Economics and Business Administration at NHH. Please note that neither the institution nor the examiners are responsible – through the approval of this thesis – for the theories and methods used, or results and conclusions drawn in this work.



# Acknowledgements

This thesis was written in the fall of 2022 as a part of our Masters of Science and Business Administration at NHH with specialization in business analytics. First and foremost, we would like to thank our supervisor Sondre Hølleland for his excellent support. Always available with skillful advice and knowledgeable guidance. We would also like to thank our friends and family for all the support the last months.

Norwegian School of Economics

Bergen, December 2022

---

Jacob Hagan

---

Rolf Tynning Henriksen

# Abstract

In this thesis, we explore how sentiment from financial news could affect stock volatility. Using financial data from the S&P100, volatility data from the Volatility Index (VIX) and sentiment data collected with web scraping we make five different machine learning models with different covariates. Analyzing the effect in both individual sectors and a combination of all sectors, with a total of 240 different models.

In order to isolate the effect of sentiment, we create datasets with and without the information and look at how the results differ. We found little proof that the additional information from the news sentiment affects the result significantly. The reason for this is complex, but we believe that using sentiment would be better suited for classification of volatility direction. Our best attempts to predict volatility on index level came from the LSTM model that got an  $R^2$  score of 43,6% using sentiment as a covariate. The best result on an individual sector came from the random forest model that got an  $R^2$  score of 62.5% using sentiment to predict volatility in the energy sector. Although these scores isolated are acceptable, for the majority of the models, those without sentiment data performed as well, if not better.

**Keywords** – Machine learning, Sentiment Analysis, Volatility

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Volatility forecasting . . . . .	3
2.2	Forecasting with sentiment . . . . .	4
2.3	Volatility prediction with sentiment . . . . .	5
<b>3</b>	<b>Sentiment Analysis</b>	<b>6</b>
3.1	Scraping data . . . . .	6
3.2	Natural Language Processing . . . . .	6
3.3	Sentiment analysis . . . . .	7
3.3.1	Pre-processing . . . . .	7
3.3.2	Tokenization . . . . .	7
3.3.3	Stopwords . . . . .	7
3.3.4	Lemmatization . . . . .	8
3.3.5	Lexicon Based Sentiment Analysis . . . . .	8
3.3.6	Vader Sentiment Analysis . . . . .	8
<b>4</b>	<b>Data</b>	<b>9</b>
4.1	Period of study . . . . .	9
4.2	Sentiment data . . . . .	10
4.2.1	Pre-processing news headlines . . . . .	10
4.2.2	Descriptive statistics . . . . .	10
4.2.3	Sentiment score . . . . .	11
4.3	Financial data . . . . .	12
4.3.1	Volatility Measure . . . . .	13
4.3.2	Pre-processing decisions . . . . .	15
<b>5</b>	<b>Machine Learning Methodology</b>	<b>17</b>
5.1	Supervised vs unsupervised machine learning . . . . .	17
5.2	Regression vs classification . . . . .	17
5.3	Train-test split . . . . .	18
5.4	Overfitting . . . . .	18
5.5	Scaling data . . . . .	19
5.6	Validation set . . . . .	20
5.7	Walk Forward Validation . . . . .	20
5.8	Assessing performance . . . . .	20
5.9	Ridge Regression . . . . .	21
5.10	Support Vector Regression . . . . .	22
5.11	Random Forest regression . . . . .	23
5.12	Extreme gradient boosted trees . . . . .	25
5.13	Recurrent Neural Networks . . . . .	26
5.13.1	Long Short Term Memory models . . . . .	26
<b>6</b>	<b>Analysis</b>	<b>29</b>
6.1	Ridge regression . . . . .	29

---

6.2	Support vector regression . . . . .	30
6.3	Random Forest . . . . .	32
6.4	XgBoost . . . . .	33
6.5	LSTM . . . . .	35
6.6	Results Summary . . . . .	36
<b>7</b>	<b>Discussion</b>	<b>38</b>
7.1	Predicting power of sentiment . . . . .	38
7.2	Economic application of volatility . . . . .	41
7.3	Limitations . . . . .	41
<b>8</b>	<b>Conclusion</b>	<b>43</b>
8.1	Further work . . . . .	43
	<b>References</b>	<b>45</b>
	<b>Appendix</b>	<b>49</b>
<b>A</b>	<b>Additional figures</b>	<b>49</b>

# List of Figures

4.1	Flowchart of pre-processing for sentiment analysis . . . . .	10
4.2	Word cloud of the 100 most used words in the collected news headlines . . . . .	11
4.3	Histogram of sentiment scores, without neutral scores. . . . .	12
4.4	Time series of the daily sentiment scores . . . . .	12
4.5	Time series of daily volatility aggregated on all companies . . . . .	15
5.1	Illustration of Random Forest structure (Amanoul et al., 2021) . . . . .	24
5.2	Illustration of Extreme gradient boosting structure (Wang et al., 2019) . . . . .	26
5.3	Basic illustration of Long short term memory cell structure (Zhou et al., 2022) . . . . .	28
6.1	Volatility predictions for all sectors using news sentiment . . . . .	36
6.2	Time series of predicted and true value, using a LSTM model on the financial sector . . . . .	37
6.3	R <sup>2</sup> scores on all sectors grouped by model . . . . .	37
A.1	AI generated introduction . . . . .	49
A.2	Correlation of predictions with all covariates . . . . .	50
A.3	Time series plot of predictions from all models with all covariates . . . . .	50
A.4	Correlation of predictions with new sentiment only . . . . .	51
A.5	Time series plot of predictions from all models with news sentiment only . . . . .	51
A.6	Correlation of predictions with VIX index only . . . . .	52
A.7	Time series plot of predictions from all models with VIX index only . . . . .	52
A.8	Correlation of predictions without VIX index and sentiment . . . . .	53
A.9	Time series plot of predictions from all models without VIX index and sentiment . . . . .	53
A.10	Correlation matrix of individual companies in the information technology sector . . . . .	54

# 1 Introduction

Sentiment analysis is a task of natural language processing (NLP) that deals with the detection and classification of emotions in text. The goal is to identify the polarity of a given sentence (positive, negative, or neutral), and sometimes also the intensity of the sentiment. The application of sentiment analysis has grown in recent years with the increasing popularity of machine learning. One such application is in detecting stock volatility. Market volatility can be affected by a number of factors, including public sentiment around a company or product. By analyzing sentiment data, it may be possible to predict future market movements more accurately.

As an example of how far NLP and machine learning has come, the paragraph above was in its entirety generated by an AI from Writerly-AI (2022). Given only a selection of relevant keywords (seen in figure A.1). Although this is an amusing use case, it speaks to the multitude of different applications sentiment analysis can have. One of these applications, as the AI mentioned, could be predicting stock volatility.

Vast amounts of news are published to the internet every day from a myriad of sources. Processing information is a key element in our everyday life, but analyzing this amount of data is impossible for any human to do alone. This is where sentiment analysis can be applied. By understanding the emotional direction of thousands of articles automatically, traders could use this information for economic gain. As of now, there exists a fair amount of research on the relation between media sentiment and stock return (Schumaker et al., 2012; Li et al., 2014b,a). A more limited area of research however, is on the relation between media sentiment and market volatility. This is an interesting topic because of the complexity in modern financial markets and the potential implications of automated volatility predictions. With this brief introduction we present the research question of this thesis.

*Explore the relationship between volatility and news sentiment and evaluate its predicting powers.*

This thesis will look at 100,000 news headlines and 4 years of financial data in order to explore this relationship between news sentiment and stock volatility. We will construct a



daily sentiment score and use this information in conjunction with financial data and a volatility index (VIX), in an attempt to make predictions of the volatility of the S&P100. Additionally, we will look at how sentiment score affect volatility predictions, compared to using the volatility index in an attempt to isolate the effect of sentiment. We will try to make predictions of market volatility, in addition to the individual sectors.

We will first present previous work within the field of sentiment analysis and volatility prediction. Secondly, we describe the theoretical foundation of sentiment analysis required to analyze the effect of sentiment on stock volatility. Followed by the data used in our analysis. The theory behind the machine learning models is separated in its own section. This order of sections is chosen to reflect the step-wise approach of our study. Finally, we present our findings, a discussion of relevant observations, limitations, and suggested avenues for further work.

## 2 Background

In this section we aim to cover the previous relevant literature within sentiment analysis, volatility forecasting and the combination of the two. There is an abundance of research conducted on both sentiment and volatility forecasting, but literature on the combination is somewhat limited.

### 2.1 Volatility forecasting

An early, but fairly important paper was written by Brailsford and Faff (1996) comparing different volatility forecasting techniques. They tested ten different forecasting models to predict monthly stock market volatility in Australia. They found that no one model could be considered the best, and that it was largely dependent on the choice of error statistic. They favored a simple regression model, but suggested that an Auto Regressive Conditional Heteroskedasticity (ARCH) class model (Engle, 1982) could be equally good, if not slightly superior. Their final remark was however that "volatility forecasting is a notoriously difficult task."

Another study by Brooks and Persaud (2003) compared a set of different models for forecasting volatility in order to re-examine the existing theories. The set included a generalized ARCH (GARCH) model (Bollerslev, 1986), random walk (see e.g. Lawler and Limic (2010)), Long Short Term Memory (Hochreiter and Schmidhuber, 1997) model and different multivariate models. These were used to forecast volatility in the UK financial market. They argued that the loss-function could have an over-riding effect on the accuracy of the forecast. Illustrating that there is more to volatility forecasting than the choice of model and error statistic.

In contrast to testing different models, Mittnik et al. (2015) looked at what variables that affects the S&P500s volatility. Using variables from equity markets, macroeconomics, foreign exchange markets and other risk measurements they selected 84 different predictors. The selection included the Volatility Index (VIX) from the Chicago Board Options Exchange, interest rates, inflation rates, bond rates and S&P500 returns. They identified the VIX as being the best predictor for volatility.

## 2.2 Forecasting with sentiment

The choice of model will affect the results, but another important part is how to best incorporate the available information. In 2011, Joseph et al. tried to predict stock return using investor sentiment. Utilizing Google search frequency of stock tickers as a proxy from sentiment. They found that increased search intensity foreshadowed abnormal returns and trading volumes (Joseph et al., 2011).

Further, Schumaker et al. (2012) investigated the relationship between the sentiment in financial news articles and stock price movement using a combination of minute based stock quotes and articles from Yahoo! Finance. They found that using sentiment in predicting price direction gave a correct result in more than 50% of the time, showing that sentiment analysis could be beneficial in stock predictions.

Several other studies have been conducted on this topic. Notably, a study done by Li et al. (2014a) found using a combination of lexicon-based sentiment analysis (see chapter 3.3.5) and Support Vector Regression (SVR) model (Vapnik, 1999a) that investors are affected by public sentiment and firm-specific news articles. Giving them increased knowledge that affect their investment decision making. Using data from news articles and financial discussion forum posts, they concluded that stock markets are affected by public news.

Li et al. (2014b) tried to predict Hong Kong stock prices using a combination of sentiment analysis and support vector machines. They used two different lexicons for their sentiment analysis. Comparing their results with a Bag-of-words model which analyzed the relationship between statistical patterns in headlines and stock prices. Their result showed that the lexicon-based sentiment analysis outperformed the bag-of-words model at individual stock, sector, and index level. Further, they found that models that used polarity from headlines could not make useful stock predictions. Lastly, they found that there was a minor difference when using different lexicons for their sentiment analysis.

## 2.3 Volatility prediction with sentiment

One study by Antweiler and Frank (2004) analyzed the relationship between 1.5 million posts from stock message boards and the 45 companies from Dow Jones Industrial Average. They found that the stock messages helped predict the market volatility, and that the effect on stock returns was statistically significant.

On the topic of predicting volatility with the added information of sentiment analysis, Liu et al. (2017) compared a Recurring Neural Network (RNN) model (Rumelhart et al., 1985) with and without the use of sentiment analysis. They used data from an online stock forum to predict volatility in a Chinese market. They found that using sentiment in the model boosted the accuracy of their classifications from 61.1% to 65.5%. They mention that one drawback of their method is the trustability of messages from a stock forum since there are no mechanisms to ensure the validity of the messages.

A more recent analysis was done by Deveikyte et al. (2020). They used data from almost 1 million financial news and tweets to predict stock volatility on the index of the 100 largest companies on the London stock exchange. Using a lexicon-based model they extracted the daily sentiment from the news and found that there was a strong and significant negative correlation between positive news and volatility the following day. Further, they achieved a classification accuracy of 63% when trying to predict volatility.

## 3 Sentiment Analysis

In this chapter we will explain our choice of methodology regarding sentiment analysis. First we explain how the sentiment data is collected. Followed by an explanation of the applied steps in our sentiment analysis.

### 3.1 Scraping data

Web scraping is a technique for extracting data from the World Wide Web to save it for later retrieval or further analysis (Zhao, 2017). The most common implementation of a web scraper is the use of automated scripts that queries a web server for data. Usually in the form of HTML (Mitchell, 2018). Automatic web scraping allows us to collect thousands of news headlines automatically and save them to a desired format for further use. We achieve this by sending a request to the page containing the news headlines. A GET request uses the URL to specify the search parameters. This replicates what a physical user would type in the URL bar in a browser. The web server returns the HTML of the requested page, and we can use the Python package Beautiful Soup (Richardson, 2007) to structure the HTML in a way that enables extraction of the relevant data.

### 3.2 Natural Language Processing

Natural Language Processing (NLP) uses computer science in order to analyze, understand and produce human language content (Hirschberg and Manning, 2015). It started as an combination of artificial intelligence and linguistics in the 1950s (Nadkarni et al., 2011), but has since evolved greatly with the modern advances within neural networks and large scale data collection (Anastasopoulos et al., 2020). Sentiment analysis, a field within NLP, is the task of extracting the opinions of authors about specific entities within a text (Feldman, 2013).

## 3.3 Sentiment analysis

An important activity for humans is to understand the emotions that are communicated to us. It may be news, a report, a political speech, or a normal conversation. However, when the volume of information increases it becomes challenging for us to process it all (Appel et al., 2015). This is where we see a need of an automated process for extracting sentiment. Further, Cambria et al. (2017) argues that the ability to automatically capture the sentiments of the general public has become of increasing interest in the business world because of the usability in financial market prediction. In order to extract the sentiment through automated analysis, several techniques have been developed. The two most prominent methods being lexicon-based and machine learning based (Taboada et al., 2011). Our preliminary analysis on the accuracy of the two different models showed us that the lexicon-based approach was advantageous based on the data we have.

### 3.3.1 Pre-processing

An essential step in preparing the data for modeling is textual preprocessing (Chai, 2022). This is considered crucial in order to extract the most accurate results from the sentiment analysis (Krouska et al., 2016). There are several different methods that need to be considered, but the most important techniques are tokenization, text normalization, punctuation handling, removing stopwords, stemming and lemmatization (Chai, 2022).

### 3.3.2 Tokenization

Tokenization, in NLP, is the process of splitting sentences into pieces called tokens. These are most often words, but can be any grouping of letters that form a useful semantic unit for further processing (Christopher et al., 2008). For a computer, a sentence is just a long string of characters. Tokenization splits this up into subunits that is suited for analysis. This is one of the earliest steps in making text ready for text analysis (Grefenstette, 1999).

### 3.3.3 Stopwords

In order to reduce the noise in textual data, a list of predefined stopwords (e.g "the", "because", "of") is filtered out. This is a popular procedure (Saif et al., 2014) and a

standard component (Sarica and Luo, 2021) in preprocessing data for sentiment analysis.

### 3.3.4 Lemmatization

Lemmatization is the process of finding the normalized form of a word, and is an important preprocessing step in NLP (Plisson et al., 2004). The method of lemmatization is simply put to reverse a word to its base form. For example, the words "Driving", "Drives", "Drove" should all be transformed to the normalized verb "Drive".

For all our preprocessing requirements we used the Python module Natural Language Toolkit (Bird, 2006). This module contains functions for lemmatization, tokenization and stop word removal.

### 3.3.5 Lexicon Based Sentiment Analysis

The lexicon-based approach is making use of a predefined dictionary where words are given a semantic orientation (i.e, positive or negative) in the form of a numeric score (Taboada et al., 2011). The main idea is that opinion words express positive (e.g., "perfect" and "good") or negative sentiments (e.g., "awful" and "terrible"). Although most opinionated words are adjectives or adverbs, there are also verbs and nouns that have their own sentiment (Zhang et al., 2011). The sentiment of a sentence is calculated in the following way. First the sentence is preprocessed and split into tokens. Then each token has its sentiment score checked in the lexicon and given a sentiment score. When this is done for all tokens, typically the average or sum of sentiment scores will produce the overall sentiment for the sentence (Jurek et al., 2015).

### 3.3.6 Vader Sentiment Analysis

Valence Aware Dictionary and sEntiment Reasoner (VADER) is a lexicon and rule-based analysis tool for determining the sentiment of text. In a benchmark comparison of lexicons VADER was considered one of the best on multiple datasets with sentence level data (Ribeiro et al., 2016). Additionally, Viegas et al. (2020) noted the VADER lexicon as the best available off-the-shelf choice for sentiment analysis.

---

## 4 Data

In this section we will explain our choice of data, data collection process, the necessary pre-processing and present the final data. We have included essential graphs and figures and additional plots can be found in the appendix.

### 4.1 Period of study

Our dataset contains observations from 1. January 2018 to 31. December 2021. The period is split into 800 days of training from 01.01.2018 to 15.03.2021 and 200 days of testing from 16.03.2021 to 31.12.2021. News sentiment is collected from web scraping news headlines from the same time period. The choice of time span was motivated by several factors. Firstly, we wanted data over several years to account for yearly cycles and to reduce the impact of years with one-off events. Secondly, we wanted to include known volatile periods to highlight changes in volatility. The selected time span includes both large positive and large negative changes in the stock market. The largest are particularly found during the Covid-19 pandemic. This might highlight changes in volatility but also challenge the robustness of the models. Despite potential challenges, we chose to keep the volatile periods because there are volatility shocks present in both training and testing sets. Therefore, we believe these shocks will not compromise the results but rather be a realistic representation of volatility. It is also worth mentioning that computational cost was central in our decision not to increase the time span further than four years.

For the selected time span, we aim to aggregate data by day. This will be done for each of the eleven sectors contained in the S&P100. Additionally, we will include one aggregation across all sectors, resulting in a total of twelve datasets. This will ensure all our data sources are combined on the same daily level. Another argument for this aggregation, is to group the companies one might expect to have similar dependencies together. This might create a less noisy volatility development. When presenting the volatility data, we will explain how this is calculated.

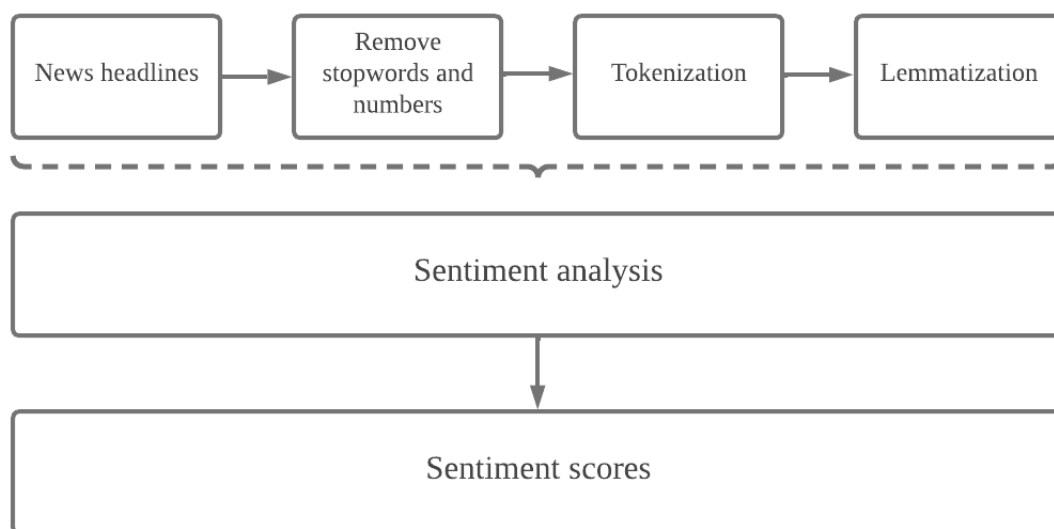


## 4.2 Sentiment data

This section looks at the data collection, characteristics, and sentiment of the news headlines used in this paper. The data for the sentiment analysis is gathered from Investing.com. This is an investing website that collects news from several sources, mainly Reuters in order to provide a collective of financial news to the public. We created an automated web scraper in order to collect the needed data efficiently.

### 4.2.1 Pre-processing news headlines

In order to reduce noise and improve clarity in the headlines, the data needs to be pre-processed. The steps include removing numbers and special characters (e.g., "3%", "!", "?"), removing stopwords (e.g., "and", "that", "for"), tokenization and lemmatization. The individual processes are described chapter 3.



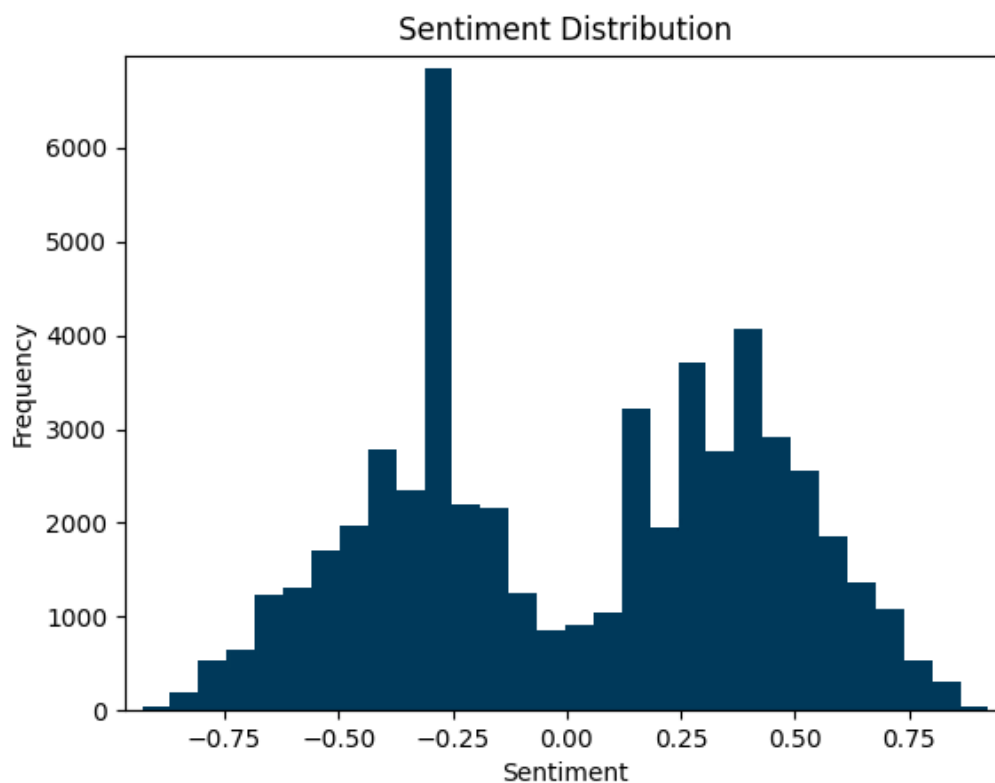
**Figure 4.1:** Flowchart of pre-processing for sentiment analysis

### 4.2.2 Descriptive statistics

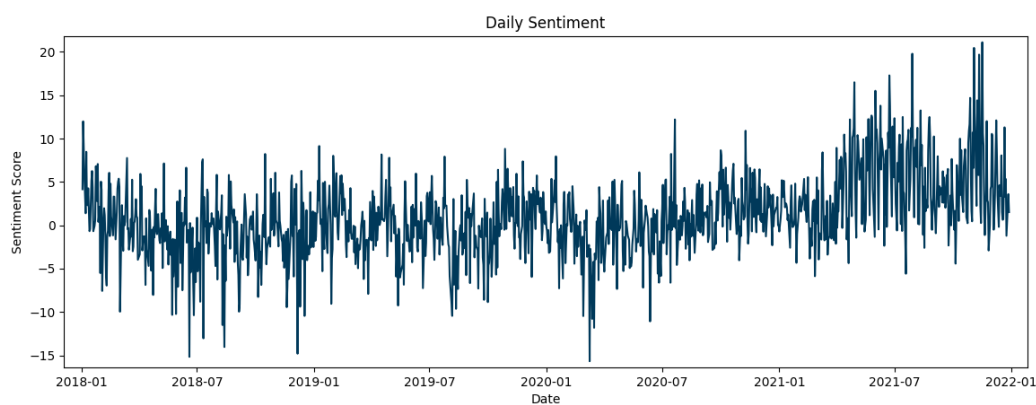
The dataset consist of 92 201 unique observations with an average length of 9.9 words per headline.

We can look at a word cloud for the headlines. This represents the word frequency of the top 100 most used words for our sentiment dataset. Some prominent words are large





**Figure 4.3:** Histogram of sentiment scores, without neutral scores.



**Figure 4.4:** Time series of the daily sentiment scores

## 4.3 Financial data

There are arguably a large number of unknown and sometimes un-quantifiable covariates affecting changes in volatility. In general, our main motivation behind the selected financial covariates are their assumed predicting power and direct relevance to stock volatility forecasting. Another important factor in our choice is daily availability. The combination of these two criteria on a company level naturally limits the pool of variables to choose

from. The financial data was extracted from Wharton Research Data Service as a merged dataset from the Center of Research in Security Price (CRSP) and Compustat database. Where the latter is Standard and Poor's own database on financial and market statistics. In the merged dataset *Trading volume*, *open*, *close*, *high* and *low* prices are retrieved from CRSP. The remaining financial variables, *Indicated annual* dividend and *Earnings per share* are retrieved from Compustat.

*Open*, *close*, *high*, and *low* prices for the stock are included to provide measures of the changes in price during the day. *High* and *low* covariates provide information about the range while *open* and *close* indicates the daily total change. To highlight the daily development in price further we also calculate *returns* by subtracting the previous *close* price from the current day's *close* price.

*Indicated annual dividend* is an estimated dividend an investor can expect to receive from the stock the coming year. Changes in this variable could change investors' view of the stock and thus the variable could be relevant to volatility predictions. *Earnings per share* and *trade volume daily* can also be considered relevant to volatility as they indicate the development in company performance as well as how the trade activity changes daily. *Earnings per share* is calculated quarterly for each company and changes could be a signal for investors to buy or sell shares.

Li et al. (2022) and Mittnik et al. (2015) both identified the VIX (Volatility Index) as a good predictor for volatility. Therefore, we have chosen to include it, as it is also an alternative measure of sentiment in the market. The index is constructed from real-time, mid-quote prices in S&P500 put and call options. This measures an indication of the investors view of the future. To represent the daily change in the VIX index we calculate the difference between high and low. Our choice is based on careful testing in preliminary work.

### 4.3.1 Volatility Measure

To predict volatility, we need to calculate a target variable. A traditional way of calculating volatility is by using a GARCH model (Chong et al., 1999). To reduce complexity in our study, we instead chose to calculate the volatility variable as the squared standard deviation of the five-day previous returns. Despite a simpler calculation, one can argue

that we will still be able to investigate news sentiment's effect on volatility, as standard deviation captures the variance in returns. This is similar to (Deveikyte et al., 2020), who also chose to use a similar measure for volatility. Regardless of alternative ways to calculate volatility, we will not pursue this further. The focus of this study is not to find the best volatility measure, but to explore the relationship between sentiment and volatility changes.

As mentioned earlier, data is divided into different datasets containing individual sectors or a combination of all sectors. In a dataset, volatility is therefore the aggregated volatility for the current sector and calculations are done separately for each dataset. Aggregating the volatility is done for several reasons. First, the data leakage between covariates and target variable is likely reduced. This potential data leakage is assumed to come from *returns*, as it is used as a lagged covariate but also in the calculation of volatility. By first calculating volatility for individual companies and then aggregating across the sector we are reducing the potential data leakage. Secondly, we have several news articles for each day. Therefore, the sentiment score is a daily aggregation, and not filtered by company. This implies the news sentiment is mostly relevant to aggregated financial and volatility data, and would maybe not be as relevant to individual companies. Further implications this could entail is discussed in section 7.

When calculating volatility, we start by defining returns on day  $t$  for company  $s$  as  $y_{st}$ . Returns aggregated across all companies in a sector are adjusted for company market share which is denoted as  $w_s$ . The applied weights are calculated as the company's fraction of the total market capitalization. Where the latter is calculated as a company's stock price multiplied with number of shares. This calculation is done individually for each dataset. Therefore, a company can have a different  $w_s$  for its individual sector and the dataset containing all sectors. Then, the sum of all returns for a single day  $t$ , adjusted for market share is denoted as

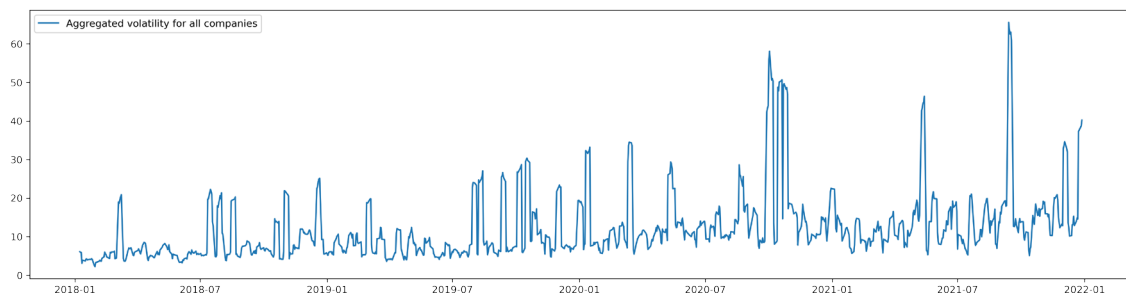
$$y_t = \sum_s w_s y_{st}.$$

Volatility for each company is given as  $v(y_{st})$ . It is calculated as the empirical variance of the five previous daily returns. This rolling window is moved throughout the time span.

Finally,  $V(y_t)$  is denoting the aggregated volatility from the sum of individual company volatility  $v(y_{st})$  in time  $t$ . After calculating  $V(y_t)$ , the values are annualized. When combining all steps, the volatility target variable is calculated as

$$\sqrt{V(y_t)} = \sqrt{\sum_s w_s^2 v(y_{st})}.$$

This calculation is not accounting for co-variation between the different companies. One could argue companies within the same sectors have similar development in returns, which needs to be accounted for. When creating correlation matrices of returns from different companies, we find little to no correlation. An example of this can be seen in figure A.10 in the appendix. Thus, we assume there is no need to incorporate a co-variation term in our volatility calculations. Despite this assumption, we are aware this is a source of error in our study and one could argue the co-variance in asset returns could become larger during large systematic risk events like covid-19. The final daily volatility development can be seen in figure 4.5.



**Figure 4.5:** Time series of daily volatility aggregated on all companies

### 4.3.2 Pre-processing decisions

To further prepare the data for machine learning the VIX, financial data and news sentiment are combined by date. This removes all news sentiment from non-trading days, as we have news sentiment for every day of the year but only financial data from trading days. Therefore, an important assumption is that information from important news will be represented over several days. While news mentioned only once will not effect the volatility significantly. Based on this we consider the reduction in data as justified, but will discuss implications this might have in section 7.

Throughout the data there are some companies that has stock returns significantly higher

than the rest. This creates shocks for certain days, even after aggregating across sectors. This can be caused by stock rallies or organizational changes. To reduce the magnitude of these shocks one could consider windzoring the largest values, exchanging the largest values with the 95th percentile. After careful testing we found the data best to be unchanged. It seems the rest of the financial data is reflecting the changes to some degree and the data is arguably most realistic when unmanipulated.

Several of the machine learning models used in this thesis are not able to incorporate the sequential nature of time series and thus we are converting the time series into a dataset suited for supervised learning. This is done by lagging the covariates in relation to the volatility measure. Volatility at present time will therefore be matched with observations of covariates at least one step earlier. All variables are lagged by one day, but returns, news sentiment and VIX index are lagged by a whole week. This is to account for any significantly lagged relationships between Volatility and news sentiment or VIX. The final data contains a total of 34 variables in each of our 12 datasets. After pre-processing we are left with the following base covariates.

Variable name	Data type	Description
datadate	datetime64	Daily trading dates
sentiment_score	float64	News Sentiment score
dvi	float64	Indicated annual dividend
eps	float64	Earnings per share
cshtd	float64	Trade volume daily
prcd	float64	Close price for company
prchd	float64	Highest price for company
prcl	float64	Lowest price for company
prcod	float64	Open price for company
returns	float64	Daily returns for company
volatility	float64	Volatility for company
tic	Category	Company tic identification
industry	Category	Industry identification
vix_change	float64	Daily difference for VIX index

**Table 4.1:** Table showing all variables used in the analysis

## 5 Machine Learning Methodology

In this section we will present the relevant machine learning theory used in this thesis. First, we explain several key methodology considerations, followed by how we assess prediction performance. Lastly, we briefly present the different models.

### 5.1 Supervised vs unsupervised machine learning

Supervised machine learning is using labeled data to try to find a relationship between the dependent and independent variables, and then to make predictions based on new data (Burkov, 2019). Unsupervised learning does not need labeled data and tries to find unknown structure and relationships in the data, typically by clustering observations. Supervised learning is known for providing more accurate results but has a higher computational cost than unsupervised learning.

We have chosen to use only supervised learning models when predicting volatility. This is because our goal is to make predictions based on new data, and not to cluster or structure the dataset. In addition, time series and sequential data is not suited for clustering (Burkov, 2019). Our goal is also well defined and needs accurate results, indicating that supervised learning is the correct approach to take in this study.

### 5.2 Regression vs classification

When creating supervised machine learning models, we need to decide if we want to solve a classification problem or a regression problem. This is often decided by the type of data available. Our data indicate that regression models are best suited, as we have almost exclusively continuous data (James et al., 2013). It is still important to reflect on the implications of using a regression model to predict volatility. A classification model is classifying data into predefined categories while regression models are trying to predict an exact value (James et al., 2013). This might make the model less likely to make very accurate predictions as volatility is affected by several unknown factors. Despite this potential implication, we want to explore news sentiment's relationship to volatility as a regression problem and will treat the data as time series.



## 5.3 Train-test split

In supervised machine learning, models need data to be trained upon. To be able to determine if a model is memorizing the data or learning trends and structures, we need separate and unseen data to verify if our predictions are accurate (James et al., 2013). Therefore, a separation of training and testing data is essential when building supervised learning models. The main motivation for the data split is to prevent overfitting and to be able to evaluate predictions with an unbiased true value (James et al., 2013). We have chosen to split our data into 80% training data and 20% test data, as this is widely considered a good practice (Hastie et al., 2009).

## 5.4 Overfitting

Overfitting is the concept of making a statistical model fit the training data too well. (Burkov, 2019) As a result, it will be unable to make good prediction on data that is not the training data. Thus, defeating its purpose. This means that the model has mistakenly captured too much statistical noise from the dataset as part of the underlying data structure. This creates a more complex model than needed and the model is less generalized when presented with unseen data (Burkov, 2019). To counter overfitting it is possible to apply several different techniques.

Adding more training data can give the statistical model more opportunities to capture the true underlying structure of the data. This technique is heavily dependent on the quality of the data added and has to be applied with caution (Hastie et al., 2009). If the additional data is not relevant, the new data could instead provide more noise to the model and increase overfitting further (James et al., 2013).

Regularization is a technique for reducing the impact each feature has on the model and thus moderate learning. It applies penalties to the input values which is regulating how much influence individual features can have. This limits the variance in the model and reduces the impact of irrelevant features on the model (Burkov, 2019). Some regularization techniques apply penalties that can shrink the feature value to zero, thus removing the feature from the model. The lasso (L1) regularization technique is an example of this. Alternatively, Ridge regularization can shrink the value down, but never entirely remove

the feature from impacting the model (Burkov, 2019). Ridge (L2) regularization is mainly used in this study.

Feature Selection is the process of reducing the number of features in a model. This is done to reduce the complexity of the model and to allow the model to learn a more dominant trend in the data. The training data is used to find the most impactful features and remove the irrelevant or insignificant features. This method highly impacts the variance – bias trade off when evaluating the model. Which can be described as the trade off between the variation in predictions and the accuracy of predictions. (Burkov, 2019).

Ensemble algorithms create a set of the training data and train individual models with individual predictions. The predictions are then aggregated to find the average outcome. This aggregated result often provides a better reflection of the trends and structures in the data. (Géron, 2019). Bagging, boosting, and decision trees are some common types of ensemble methods, and these are often used to reduce variance in datasets which contain significant noise

Stopping the model early is a way to reduce overfitting (Géron, 2019). This method aims to reduce the amount of training done to prevent the model from modeling noise in the dataset. The reduction in training time will also increase the risk of underfitting the model. Thus, to make good predictions it is important to find a balance between giving the model enough time to learn trends and structure, while not overfitting the data.

## 5.5 Scaling data

Hastie et al. (2009) explains how several machine learning models uses distance based algorithms during their fitting process. Therefore, we need to scale the data and we apply a min-max scaler to all our covariates. This is also known as normalizing the data. Another key step in scaling is to keep the training and test scales separate. This prevents data leakage between training and out of sample data (Hastie et al., 2009). Lastly, to have a real-world interpretation of our results we are also performing descaling of predictions with the same scaler.

## 5.6 Validation set

Validation data is a subset of the training data. It is used to tune hyperparameters in models and to choose a learning algorithm (James et al., 2013). This is done by using the validation set to evaluate predictions and then adjust the model configuration. Gradually the search can find the hyperparameters and learning algorithm that produces the best predictions. This allows us to keep the test data separate and unseen while also creating a better model (Géron, 2019).

## 5.7 Walk Forward Validation

For time series data, the current value may be dependent on previous values. Walk forward validation can be used to account for the order and dependencies between the observations and also ensure no data leakage from future observations. This means the models are trained several times with new data as it becomes available. For the current time-step, the model is provided with previous values when making the prediction for the next time-step. The method used is an anchored walk forward validation (Carta et al., 2021). This provides us an increasing amount of data in the training set, as we move through time. For each iteration in the training stage the model is validated to find the best hyperparameters. Overall, this form of validation creates tuned models while preventing data leakage from future observations.

## 5.8 Assessing performance

During the training of the model, the model is usually given a performance measure to use as the optimization metric. In regression models the same metric is then used when calculating the difference between the predicted and the actual value in the test set. Mean squared error (MSE), mean absolute error (MAE) and  $R^2$  score are common metrics when calculating the performance of regression models (James et al., 2013). However, MSE and MAE will return scale dependent values. In order to compare forecast accuracy on different scales Mean Absolute Scaled Error (MASE) is a preferred option. Hyndman and Koehler (2006) suggested that MASE should be used as the standard way to compare the accuracy of different time-series forecasts and was further supported by Franses (2016).

For predictions one step ahead the MASE is often  $<1$ , but increasing the prediction horizon will lead to MASE values generally  $>1$ . Hyndman (2006) proposed the following method for calculating MASE. A scaled error at time  $t$  is defined as:

$$q_t = \frac{e_t}{\frac{1}{n-1} \sum_{i=2}^n |Y_i - Y_{i-1}|},$$

where  $Y_i$  is the naïve predicted value. The error term  $e$  from the chosen prediction method is scaled based on the mean absolute error of a naïve method applied to the testing data (the denominator in the equation). The MASE is then simply

$$MASE = \text{mean}(|q_t|).$$

Another metric used to assess the performance in prediction models is the  $R^2$  score. The metric describe how much of the variance in the predicted result is derived from the input variables (Miles, 2005). The scale is generally from 0 to 1 where an  $R^2$  score of 1 would indicate that all variability could be explained by the input variables. It is often presented as a percentage.

## 5.9 Ridge Regression

Ridge regression is a type of multiple linear regression model that is well suited to handle multicollinearity in data (McDonald, 2009). It is developed from a standard least squared error model but includes a regularization term. This enables the model to apply penalty weights to the covariates and thus also function as a feature selection (James et al., 2013). Ridge regression is broadly studied and is a model that is well suited to handle biased data, because of its self-regulating term (McDonald, 2009).

The penalty term,  $\lambda$ , must be tuned and set manually. When  $\lambda$  increases the magnitude of the coefficients is reduced and further reducing the impact the coefficients have on the model. In the ridge regression fitting procedure, the model aims to minimize (James et al., 2013)

$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2,$$

where, the first term is identical to ordinary least square, while the last term is the L2 regularizing term.  $y_i$  is the  $i$ -th row for the dependent variable and  $x_{ij}$  denotes the  $i$ -th row and  $j$ -th column of the input matrix  $X$ .  $\beta$  is estimated from the sample during the minimization process.

## 5.10 Support Vector Regression

Support vector regression (SVR) is based on the principle of separating observations with a hyperplane, and is built from the classification model support vector machines (Burkov, 2019). Regression problems are not necessarily linearly separable and to handle this, the model creates linear regressions in a higher dimensional space. This is done by implementing kernel-functions to the cost-function optimization (Vapnik, 1999b). This dimensional transformation is called the kernel trick and is what allows the model to learn nonlinear relationships.

The fitting procedure of SVR is minimizing the error from the observations to a regression line, similar to normal linear regression. The difference comes from the margin, also called support vector, which is added. Observations within the support vectors are acceptable errors and the linear optimization problem is given by Vapnik (1999b) as

$$\text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*),$$

subject to:

$$y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i,$$

$$\langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^*,$$

$$\xi_i, \xi_i^* \geq 0$$

where,  $\xi_i$  and  $\xi_i^*$  are slack variables indicating the maximum and minimum of slack from the acceptable error limit. These makes the constraints less strict and prevents infeasible constraints when no perfect linear line exist.  $\langle w, x_i \rangle$  denotes the dot product in the space of input patterns and  $\epsilon$  denotes the limit for acceptable error. The constant  $C$  reflects the relationship between the flatness of the line and what amount of deviation larger than  $\epsilon$  is

tolerated. In other words, it determines the distance between  $\epsilon$  and  $\xi_i, \xi_i^*$ .

To be able to make linear regressions of non-linear relationships we need to do a dimension transformation of the input features. This is done by applying a kernel-function  $K(x_i, x_j)$ . Based on linear regression  $y = ax + b$ , the non-linear support vector regression is given by Vapnik (1999b) as

$$y = \sum_{i=1}^n (a_i - a_i^*) \cdot K(x_i, x_j) + b,$$

where,  $K(x_i, x_j)$  is the kernel-function which takes the dot product  $\langle x_i, x_j \rangle$  as input. Because the problem is converted to a dual problem,  $(a_i - a_i^*)$  is included as a lagrangian multiplier.  $b$  denotes the constant in the regression. There are several types of kernel-functions but the Gaussian radial bias function is widely used (James et al., 2013) and is also the one used in our SVR models. This is largely because of its applicability and because it creates an output similar to a Gaussian distribution (Ko and Lee, 2013). The Gaussian radial bias function can be mathematically expressed as

$$K(x_i, x_j) = \exp\left\{-\frac{\|x_i, x_j\|^2}{\alpha^2}\right\},$$

where,  $\alpha^2$  is a manually selected smoothing parameter affecting the distribution of the data in the kernel space.  $x_i, x_j$  is denoting the dot product of space of input patterns.

## 5.11 Random Forest regression

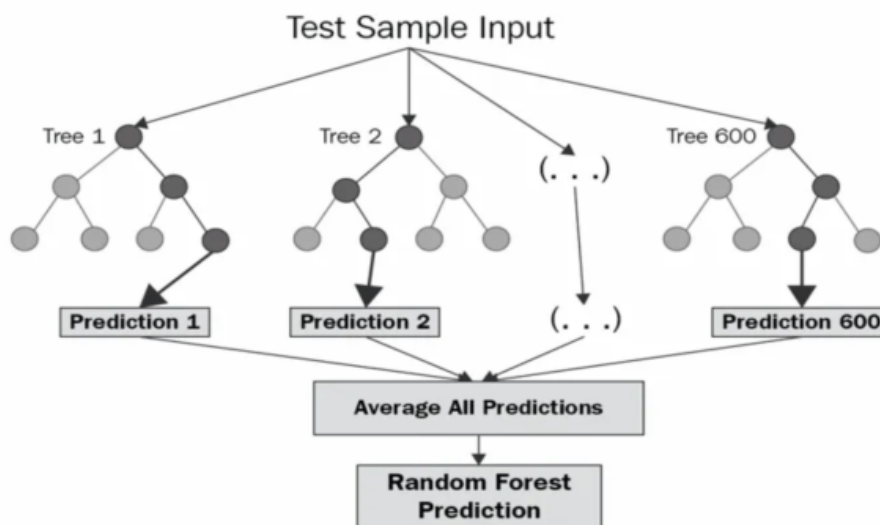
Random Forest regression is an ensemble method inspired by bagging in statistics. It combines several models created from equal sized samples of the training data, with a random sample of features included in each tree (Breiman, 2001). The combination of these trees make up the "forest" of predictions, where the average prediction is the final output. This is creating a self-regularizing model and thus limits overfitting on the training data (Segal, 2004). The inherent random nature of the Random Forest algorithm prevents a simple mathematical formulation but the algorithm can be described by the following steps (Hastie et al., 2009):

For all values from  $b=1$  to  $B$ , where  $B$  is the total number of trees, we draw a bootstrap sample of size  $N$  from the training data. From this sample, we grow a random forest tree

$T_b$ , by recursively repeating the following three steps for each of the nodes in the tree. i) Select  $m$  variables at random from total of  $p$  variables. This is done independently at each node. ii) Find the split-point among the  $m$  variables that will produce the best prediction. iii) Finally, we split the node into two daughter nodes. The output from the forest of trees can be noted as  $T_{b_1}^B$ . When the model is trained on the training data, the prediction  $\hat{f}_{rf}^B(x)$  for a new observation  $x$  is given as

$$\hat{f}_{rf}^B(x) = \frac{1}{B} \sum_{b=1}^B T_b(x),$$

where,  $B$  is the total number of trees and  $T_b(x)$  is an individual prediction from a single tree. Random Forest regression always creates average predictions and therefore struggles with extrapolation, meaning the method is not able to make predictions outside the value range found in the training observations (Zhang et al., 2019). This does not necessarily indicate poor performance in all cases but reduces the performance for problems involving data with expanding value range.



**Figure 5.1:** Illustration of Random Forest structure (Amanoul et al., 2021)

## 5.12 Extreme gradient boosted trees

Extreme gradient boosting (XgBoost) is an ensemble tree algorithm formalized by Chen and Guestrin (2016). It was quickly known for its good performance in forecasting competitions. The idea is to combine several weak learners in a decision tree to make a stronger collection. Decision trees are built sequentially on the residuals of the previous trees. Therefore, XgBoost does not explore every tree combination but instead creates trees greedily. The performance of these trees are purposely boosted and then regularized afterwards. This enables the model to identify variance not picked up by the previous tree. Chen and Guestrin (2016) formulates the predictions  $\hat{y}_i$  as  $K$  additive functions

$$\hat{y}_i = \varphi(x_i) = \sum_{k=1}^K f_k(x_i) \quad f_k \in F,$$

where  $f_k$  corresponds to an individual tree  $n$  in the regression tree space  $F$ .  $\hat{y}_i$  is the overall prediction across the entire set of trees.  $x_i$  denotes the input matrix. To find the function  $f_k$  to create a tree, the following expression is minimized

$$L(\varphi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k),$$

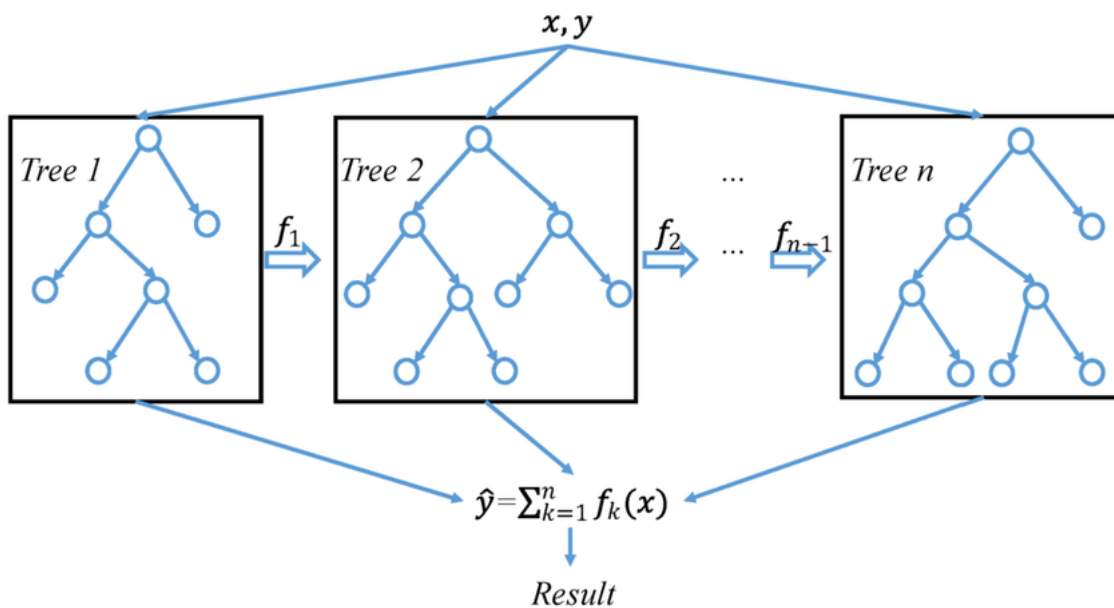
where, the first function  $l(\hat{y}_i, y_i)$  is the loss function calculating the difference between the predicted ( $\hat{y}_i$ ) and the actual ( $y_i$ ) value. The  $\Omega(f_k)$  term at the end is a regularization term and can, for a single tree  $f$ , be expressed as

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2,$$

where,  $T$  is the total number of leaves and  $\lambda$  and  $\gamma$  are manually set. Like in ridge regression,  $\lambda$  is part of the L2 regularization term and will apply penalties that effect the weights of the covariates.  $\gamma$  is controlling the minimum loss function value required at a node, to allow the tree to make a split. Further there are several hyperparameters that can be tuned and *learning rate*, *minimum child weight*, *tree dept* and *number of trees* will be tuned as part of a grid search. This will explore the most influential hyperparameters and find the best model setup for our data. (Chen and Guestrin, 2016) A prediction of an



XgBoost model is illustrated in figure 5.2.



**Figure 5.2:** Illustration of Extreme gradient boosting structure (Wang et al., 2019)

## 5.13 Recurrent Neural Networks

Recurrent neural networks (RNN) is a collective term for neural network models well suited for time series and sequential data. They are different from other neural networks as the flow of information is not only moving forward, but also back through time. This provides the model with a memory where values from observations in previous time steps can be remembered. This is a form of short-term memory in contrast to slowly changing weights, which can be viewed as a form of long-term memory (Goodfellow et al., 2016).

A known challenge with RNN is the problem of vanishing- and exploding gradient (Hochreiter, 1998). This is where the flow of error back through time moves exponentially fast to zero or to infinity. One can interpret this as the previous information is either emphasized too greatly or the impact of the previous information is almost non-existing. This can either make learning unstable, or make cost function improvements very hard (Hochreiter and Schmidhuber, 1997).

### 5.13.1 Long Short Term Memory models

Long short-term memory (LSTM) method was created by Hochreiter and Schmidhuber (1997) to improve the computational inefficiencies when performing back propagation over

considerable time intervals. It also handles the problem with vanishing and exploding gradients though time. This is done by proposing three gated cells to the neuron and Hochreiter and Schmidhuber (1997) formulates the different gated cells in an LSTM neuron as

$$\begin{pmatrix} f_t \\ i_t \\ g_t \\ o_t \end{pmatrix} = \begin{pmatrix} \sigma(W_f [x_t, h_{t-1}] + b_f) \\ \sigma(W_i [x_t, h_{t-1}] + b_i) \\ \tanh(W_g [x_t, h_{t-1}] + b_g) \\ \sigma(W_o [x_t, h_{t-1}] + b_o) \end{pmatrix},$$

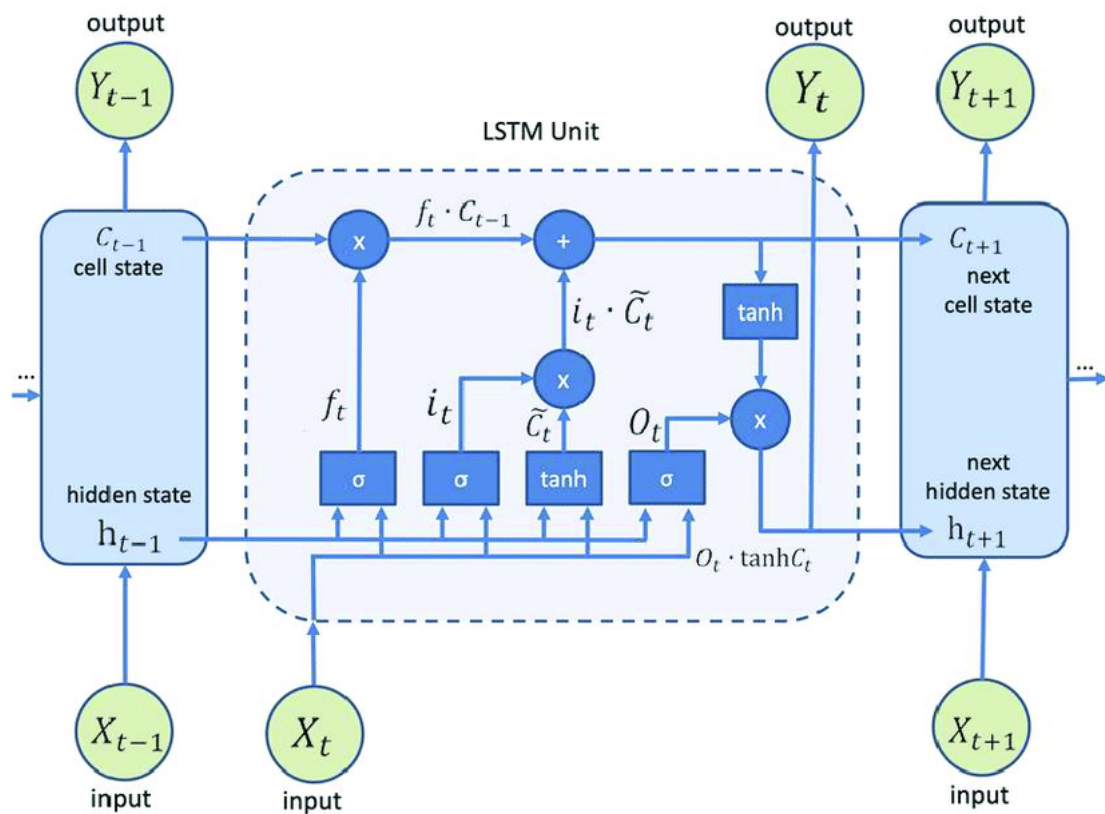
where the forget gate  $f_t$  decides what information should be used in the cell by having the possibility to set old states to zero. This is done by a sigmoid function applying weights to the previous cell state of either one or zero. Similarly, the input gate  $i_t$  controls input values and is combined with the vector of previous cell updates  $g_t$ , when updating the internal state of the cell. Then, the state of a cell  $c$  in time  $t$  can be expressed as

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t.$$

The output gate  $o_t$  contains what information is passed on and is combined with a  $\tanh$  layer to control the flow of information to the next cell (Hochreiter and Schmidhuber, 1997). Thus, the output values to the next cell can be formulated as:

$$h_t = o_t \cdot \tanh(c_t).$$

Neural networks are known to be prone to overfitting during the training phase (Burkov, 2019). To handle this, we are exploring L1(Lasso) and L2(Ridge) regularization for the the neuron layer kernel, bias, or output value. Adaptive Moment Estimation optimizer (*adam*) will be used throughout all model combinations. Chandriah and Naraganahalli (2021) found *adam* to work well with LSTM models. In addition, dropout rates between 10% and 50% are applied to further explore model regularization. This is done through a grid search, where we also explore different combinations of learning rates and number of neurons. The overall information flow in an LSTM cell is illustrated in Figure 5.3.



**Figure 5.3:** Basic illustration of Long short term memory cell structure (Zhou et al., 2022)

## 6 Analysis

In this chapter we will look at the results from the analysis. Five different models have been applied to the dataset, both on all sectors and on individual sector level. Further, to highlight and isolate the effect of news sentiment and VIX index, we will look at the model performance with and without these variables. All models have been given 800 days of training data from 01.01.2018 to 15.03.2021 and 200 days of test data from 16.03.2021 to 31.12.2021. To present our findings, all models will be presented independently.

### 6.1 Ridge regression

Illustrated in table 6.1 are the combined results from the Ridge regression on all sectors. The scoring metric is the  $R^2$  of the Ridge model predictions. It shows the score from the different datasets. We observe a minor difference in the respective scenarios, where the best performer is the one using VIX.

	<b>VIX&amp;Sentiment</b>	<b>w/o Sentiment&amp;VIX</b>	<b>w/Vix</b>	<b>w/Sentiment</b>
<i>Total</i>	40.5%	38.7%	41.4%	38.7%

**Table 6.1:** R-squared values for all sectors using Ridge regression

Further, we can look at how the model performs in predicting volatility in the different sectors. Table 6.2 shows the  $R^2$  scores. We observe a major difference in the different sectors. In order for an  $R^2$  score to be below zero for non-linear models, it needs to perform poorer than a straight line. This is the case for the Financial, the Consumer Staples and the Industrial sector. We see that the best results are from the Consumer Discretionary and the Communication Services sector. Another observation is that inclusion or exclusion of sentiment scores and the VIX index is a lot less impactful on the model performance, than the choice of sector.

Looking at the mean absolute scaled error (MASE) from the model, the consumer staples sector is, as expected, an outlier in terms of error margin. The ridge model is not able to make any valuable predictions on this sector. We can also see that the variance of the errors are low within each sector, but higher when comparing different sectors. We observe that overall the MASE scores are consistent with the  $R^2$  scores.

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	-101,6 %	-104,0 %	-105,5 %	-107,0 %
<i>Communication Services</i>	40,5 %	40,8 %	41,3 %	40,1 %
<i>Consumer Discretionary</i>	48,3 %	51,4 %	51,5 %	48,9 %
<i>Consumer Staples</i>	-281,9 %	-304,3 %	-293,3 %	-298,7 %
<i>Energy</i>	27,2 %	33,8 %	27,5 %	37,1 %
<i>Health Care</i>	2,3 %	-1,0 %	-6,0 %	8,8 %
<i>Industrials</i>	-40,8 %	-39,8 %	-46,4 %	-34,0 %
<i>Information Technology</i>	34,5 %	30,6 %	33,6 %	30,7 %
<i>Materials</i>	-12,2 %	30,7 %	12,0 %	24,3 %
<i>Real Estate</i>	38,2 %	39,1 %	38,9 %	37,1 %
<i>Utilities</i>	38,4 %	-5,4 %	38,5 %	-14,6 %

**Table 6.2:** R-squared values for each sector using Ridge regression

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	4,92	4,95	4,95	5,01
<i>Communication Services</i>	1,79	1,89	1,86	1,82
<i>Consumer Discretionary</i>	1,52	1,43	1,43	1,50
<i>Consumer Staples</i>	5,22	5,44	5,34	5,38
<i>Energy</i>	2,29	2,14	2,31	2,05
<i>Health Care</i>	2,93	2,98	3,09	2,81
<i>Industrials</i>	3,33	3,32	3,42	3,24
<i>Information Technology</i>	1,89	2,02	1,91	2,02
<i>Materials</i>	3,51	2,51	2,96	2,72
<i>Real Estate</i>	1,78	1,72	1,63	1,86
<i>Utilities</i>	1,70	2,75	1,70	2,90

**Table 6.3:** MASE for each sector using Ridge regression

## 6.2 Support vector regression

In this section we will look at the results from the support vector regression (SVR). Running the model on all sectors yields the result illustrated in table 6.4.

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
Total	34.6%	35.5%	36.1%	32.1%

**Table 6.4:** R-squared values for all sectors using SVR

The main observation here is that the model using sentiment as the only external predictor performs slightly worse than its counterparts. The model using the VIX performs best, followed by the model where both the VIX and news sentiment is removed. This could indicate that SVR is not able to use news sentiment to make good predictions, but it is able to find some useful information in the VIX index.

However, if we look at the individual sectors in table 6.5, we see a slightly different pattern. We observe some cases where the best result are from the models that have excluded

the VIX and sentiment scores, and other models that find both VIX index and news sentiment useful. This indicates again that the choice of sector is more impactful than news sentiment.

Further, the model still achieves the highest  $R^2$  score for the Energy sector and the scores from Communication Services and utilities are somewhat solid. There are still several models that get a negative  $R^2$  and subsequently does not satisfy any purpose. This reduces the overall credibility of the SVR predictions.

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	-299,9 %	-144,3 %	-204,7 %	-322,1 %
<i>Communication Services</i>	22,0 %	50,7 %	23,5 %	34,1 %
<i>Consumer Discretionary</i>	-17,1 %	19,8 %	36,6 %	-14,7 %
<i>Consumer Staples</i>	-173,1 %	-86,3 %	-66,4 %	-164,2 %
<i>Energy</i>	45,4 %	60,9 %	52,2 %	54,4 %
<i>Health Care</i>	18,7 %	-7,6 %	-4,5 %	3,7 %
<i>Industrials</i>	-28,1 %	-30,7 %	-12,8 %	-29,8 %
<i>Information Technology</i>	12,8 %	5,5 %	15,8 %	3,5 %
<i>Materials</i>	-47,5 %	15,3 %	-3,0 %	-3,4 %
<i>Real Estate</i>	0,8 %	1,1 %	0,2 %	1,6 %
<i>Utilities</i>	20,6 %	42,9 %	30,8 %	34,4 %
<i>Total</i>	34,6 %	35,5 %	36,1 %	32,1 %

**Table 6.5:** R-squared values for each sector using SVR

Table 6.6 shows the MASE for each sector for each variant of the model. We observe the same pattern as with ridge, namely that the model variants that got a low  $R^2$  score have higher margins of error. Overall, it seems like SVR achieves reasonable predictions on the Communications Service, Energy, and Utilities. The consistent performance of sectors across models might indicate the covariates used are representing changes in volatility better in these sectors.

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	7,25	5,58	6,32	7,42
<i>Communication Services</i>	1,90	1,90	1,89	2,20
<i>Consumer Discretionary</i>	2,02	1,95	1,87	2,00
<i>Consumer Staples</i>	4,37	3,25	3,09	4,14
<i>Energy</i>	2,01	1,56	1,80	1,75
<i>Health Care</i>	2,67	3,11	3,05	2,92
<i>Industrials</i>	3,18	3,23	2,91	3,23
<i>Information Technology</i>	2,43	2,68	2,36	2,63
<i>Materials</i>	4,41	3,02	3,37	3,62
<i>Real Estate</i>	1,74	1,77	1,79	1,72
<i>Utilities</i>	2,33	1,91	2,16	2,13
<i>Total</i>	2,07	2,13	2,13	2,19

**Table 6.6:** MASE for each sector using SVR

## 6.3 Random Forest

In this section we will presents the results from the Random Forest regression. When running the model on all the sectors as a whole, we get the results in table 6.7.

	VIX&Sentiment	w/o VIX&Sentiment	w/VIX	w/Sentiment
Total	36.2%	40.6%	39.5%	36.8%

**Table 6.7:** R-squared values for all sectors using Random Forest regression

We see that the Random Forest model attain the best  $R^2$  score when neither the VIX index nor the news sentiment is included. However, we observe in table 6.8 than the Random Forest gets an  $R^2$  score of over 55% in the Communication Services, Energy, Materials and Utilities sector for all variants. A significant improvement pointing in the direction of Random Forestt outperforming ridge and SVM on individual sector level, but not on the data as a whole. There is again some consistency in which sectors are the best performers, but there are inconsistencies when the VIX index and news sentiment score contributes to the best model.

The main observation when looking at the MASE scores from Random Forest is that it generally does better on individual sector level than for the dataset in total. For several sectors, the model seems to be able to make better predictions than SVR and ridge. There is again no clear pattern in the performance of news sentiment and VIX index. Overall, the results indicate that while the Random Forest model performed subpar on the total dataset, its strength lies in predicting volatility in the individual sectors.

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	4,0 %	6,6 %	1,6 %	6,1 %
<i>Communication Services</i>	57,2 %	58,2 %	58,1 %	57,1 %
<i>Consumer Discretionary</i>	36,8 %	49,1 %	41,5 %	42,1 %
<i>Consumer Staples</i>	-43,4 %	-39,9 %	-45,5 %	-41,8 %
<i>Energy</i>	62,5 %	62,3 %	62,4 %	62,5 %
<i>Health Care</i>	33,1 %	36,5 %	32,1 %	36,4 %
<i>Industrials</i>	35,4 %	37,4 %	35,9 %	36,8 %
<i>Information Technology</i>	41,9 %	48,4 %	41,5 %	43,6 %
<i>Materials</i>	61,7 %	61,7 %	61,4 %	61,6 %
<i>Real Estate</i>	34,2 %	37,7 %	33,7 %	36,2 %
<i>Utilities</i>	57,5 %	61,7 %	57,1 %	61,9 %
<i>Total</i>	36,2 %	40,6 %	39,5 %	36,8 %

**Table 6.8:** R-squared values for each sector using Random Forest regression

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	3,12	3,07	3,15	3,07
<i>Communication Services</i>	1,41	1,40	1,41	1,42
<i>Consumer Discretionary</i>	1,80	1,60	1,75	1,72
<i>Consumer Staples</i>	3,16	3,11	3,18	3,14
<i>Energy</i>	1,23	1,20	1,23	1,20
<i>Health Care</i>	2,30	2,23	2,32	2,24
<i>Industrials</i>	2,19	2,15	2,17	2,17
<i>Information Technology</i>	1,81	1,68	1,84	1,77
<i>Materials</i>	1,53	1,57	1,57	1,53
<i>Real Estate</i>	1,65	1,62	1,67	1,64
<i>Utilities</i>	1,59	1,39	1,59	1,42
<i>Total</i>	2,03	1,91	1,93	2,02

**Table 6.9:** MASE for each sector using Random Forest regression

## 6.4 XgBoost

Like the Random Forest model, XgBoost is tree-based. The similarity of these two models should indicate that with optimal parameter tuning, the XgBoost could outperform the Random Forest model. This does not seem to be the case and indicates that our choice of tuning parameters are not optimal, reducing the credibility of XgBoost's results. In some cases we see that XgBoost, and Random Forest perform about the same, with the Random Forest having a slightly better result. Most notably the two models get different outcomes with regards to the predicting power of the different covariate combinations. For the XgBoost model the variant with VIX gets the highest score with 34%, whereas the Random Forest is able to produce slightly better overall results.



	VIX&Sentiment	w/o VIX&Sentiment	w/VIX	w/Sentiment
Total	28,8%	27,3%	34,0%	30,6%

**Table 6.10:** R-squared values for all sectors using XgBoost

When looking at the individual sectors for XgBoost, we see observations in line with the previous results. The same sectors seem to be more predictable than the others. It is still the Energy sector with the exclusion of sentiment scores and the VIX that gets the highest  $R^2$  score and the lowest MASE. However, when we look at the results on the total dataset we see that the model using the VIX has the highest  $R^2$  score and the lowest MASE. This could suggest that some models are better at utilizing news sentiment and VIX index when predicting the market as a whole, rather than specific sectors.

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	-72,1%	-83,5%	-73,3%	-89,9%
<i>Communication Services</i>	33,1%	28,3%	32,3%	30,2%
<i>Consumer Discretionary</i>	14,7%	33,7%	31,9%	17,5%
<i>Consumer Staples</i>	-92,8%	-149,0%	-94,7%	-87,4%
<i>Energy</i>	59,1%	62,3%	60,4%	60,8%
<i>Health Care</i>	14,7%	1,7%	11,9%	13,7%
<i>Industrials</i>	10,7%	5,5%	-2,0%	-1,4%
<i>Information Technology</i>	20,7%	37,7%	27,3%	26,2%
<i>Materials</i>	42,6%	60,0%	54,3%	39,2%
<i>Real Estate</i>	24,7%	26,1%	30,6%	24,1%
<i>Utilities</i>	44,7%	41,8%	33,8%	44,2%
<i>Total</i>	28,8%	27,3%	34,0%	30,6%

**Table 6.11:** R-squared values for each sector using XgBoost

	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	4.42	4.66	4.47	4.80
<i>Communication Services</i>	1.85	1.97	1.85	1.88
<i>Consumer Discretionary</i>	1.98	1.81	1.83	1.99
<i>Consumer Staples</i>	3.68	4.18	3.70	3.64
<i>Energy</i>	1.35	1.33	1.33	1.33
<i>Health Care</i>	2.69	2.86	2.72	2.70
<i>Industrials</i>	2.65	2.72	2.83	2.82
<i>Information Technology</i>	2.28	1.98	2.18	2.16
<i>Materials</i>	2.04	1.48	1.65	2.19
<i>Real Estate</i>	2.07	1.96	1.85	2.03
<i>Utilities</i>	1.95	1.97	2.13	1.95
<i>Total</i>	2.31	2.27	2.10	2.24

**Table 6.12:** MASE for each sector using XgBoost

## 6.5 LSTM

The LSTM model produces the best overall scores. The results are fairly similar for the different model variants, but the model with sentiment, seen in table 6.13, stands out as the one with the highest  $R^2$  score.

	<b>VIX&amp;Sentiment</b>	<b>w/o VIX&amp;Sentiment</b>	<b>w/VIX</b>	<b>w/Sentiment</b>
Total	33.8%	40.9%	38.6%	43.6%

**Table 6.13:** R-squared values for all sectors using LSTM

If we compare the results from LSTM in table 6.14 with the other models, we see that the main improvement lies in the results using sentiment as a covariate. For some cases the model using sentiment achieves the best score, but it is not a consistent trait. Further, for the same dataset, the model variants with both VIX and news sentiment performs worse than the covariates individually. This might indicate they create noise in the model when combined together. Again, we find inconclusive results, but indications that the use of sentiment has an effect on the predicted volatility.

	<b>VIX&amp;Sentiment</b>	<b>w/o Sentiment&amp;VIX</b>	<b>w/VIX</b>	<b>w/Sentiment</b>
<i>Financials</i>	-13,7 %	-81,0 %	-77,4 %	-53,0 %
<i>Communication Services</i>	36,8 %	46,2 %	45,5 %	38,5 %
<i>Consumer Discretionary</i>	38,9 %	47,1 %	52,1 %	42,4 %
<i>Consumer Staples</i>	-173,8 %	-400,2 %	-382,2 %	-173,2 %
<i>Energy</i>	48,7 %	47,9 %	47,6 %	49,6 %
<i>Health Care</i>	29,0 %	22,3 %	31,6 %	10,5 %
<i>Industrials</i>	-9,7 %	-64,7 %	-61,8 %	-42,5 %
<i>Information Technology</i>	27,1 %	35,1 %	32,3 %	35,6 %
<i>Materials</i>	43,4 %	43,2 %	38,0 %	39,5 %
<i>Real Estate</i>	44,9 %	37,3 %	45,8 %	40,7 %
<i>Utilities</i>	41,2 %	24,2 %	37,6 %	29,2 %
<i>Total</i>	33,8 %	40,9 %	38,6 %	43,6 %

**Table 6.14:** R-squared values for each sector using LSTM

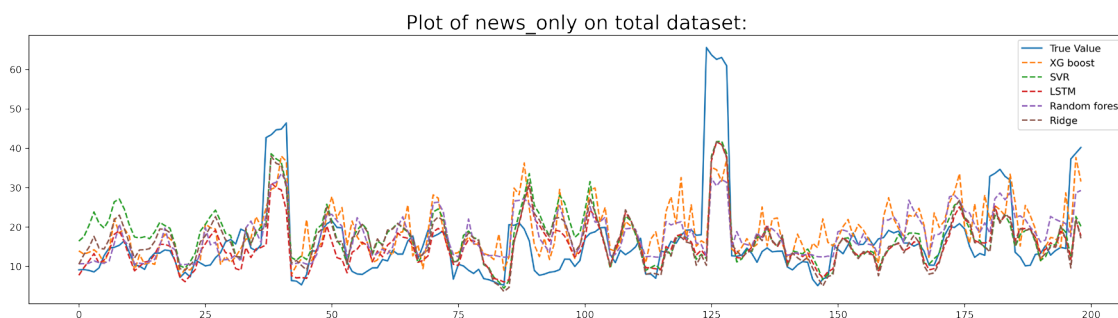
	VIX&Sentiment	w/o Sentiment&VIX	w/VIX	w/Sentiment
<i>Financials</i>	3,62	4,65	4,59	4,18
<i>Communication Services</i>	1,87	1,78	1,76	1,85
<i>Consumer Discretionary</i>	1,77	1,61	1,49	1,70
<i>Consumer Staples</i>	4,11	5,90	5,86	4,25
<i>Energy</i>	1,82	1,79	1,79	1,79
<i>Health Care</i>	2,46	2,55	2,37	2,82
<i>Industrials</i>	2,88	3,61	3,55	3,31
<i>Information Technology</i>	2,10	1,87	1,94	1,83
<i>Materials</i>	2,11	2,11	2,33	2,31
<i>Real Estate</i>	1,67	1,92	1,63	1,77
<i>Utilities</i>	1,95	2,28	1,98	2,19
<i>Total</i>	1,81	1,82	1,92	1,74

**Table 6.15:** MASE for each sector using LSTM

## 6.6 Results Summary

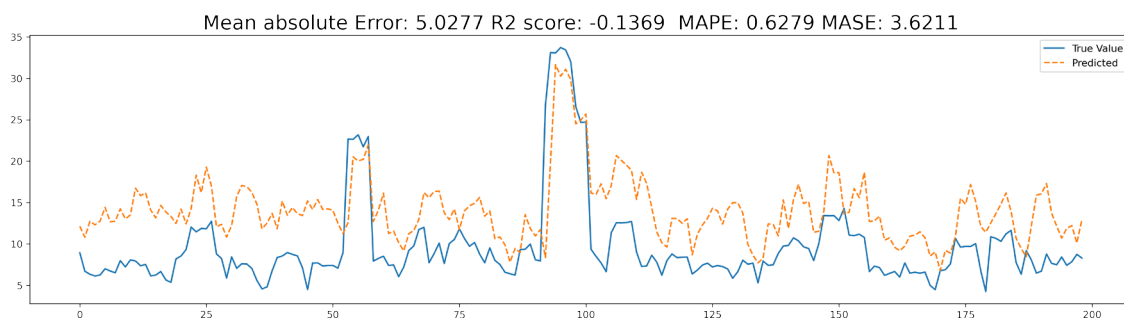
Throughout the different configurations, there is little evidence news sentiment or VIX index have better predicting power when predicting volatility. There are models where the presence of news sentiment and VIX index improves the predictions and other models where the predictions are worsened. Our results indicate model choice and sector filtration has a significant impact on prediction performance and that the effect of VIX index and news sentiment is ambiguous.

Figure 6.1 shows the prediction for each of the five models together with the volatility target value. There are some similarities between the models and we observe that none of them are able to make good predictions for the largest spike in volatility. Several models are also predicting too large values for the less volatile days. Despite this, we observe a high degree of correlation between the true value and our predictions. Figure A.2 to A.9 in the appendix illustrates this correlation.



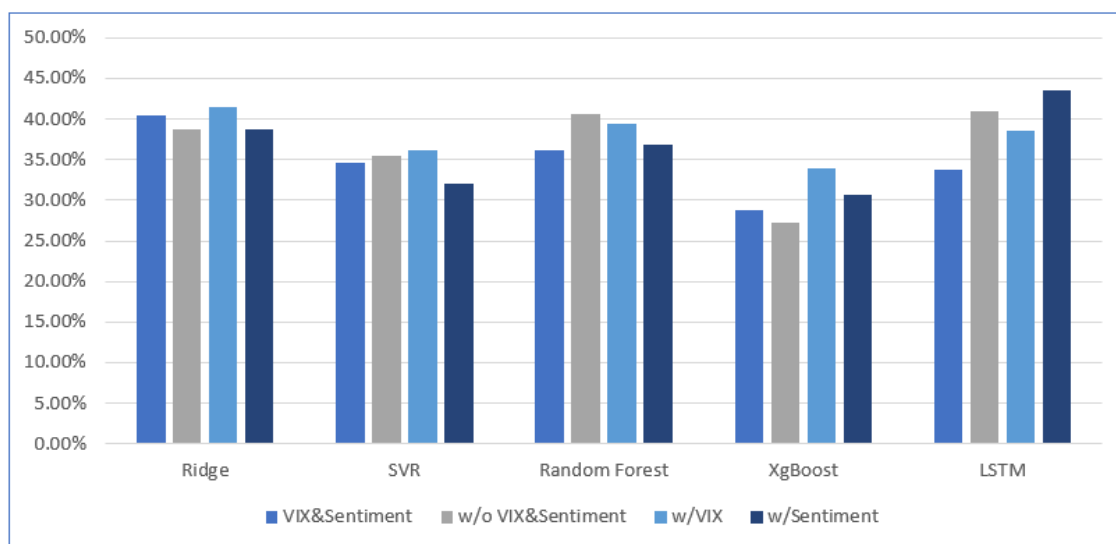
**Figure 6.1:** Volatility predictions for all sectors using news sentiment

An interesting observation is how close the predicted and true value correlate, even for the worst models. 6.2 shows the predicted values and true values from an LSTM model with both VIX index and sentiment score. For the most part, the graphs correlate fairly well, but the predictions are shifted up. The predictions are made for the Financial sector.  $R^2$  score is negative 13,7% but we can observe from the graph there is a fair degree of correlation between the predicted and true value.



**Figure 6.2:** Time series of predicted and true value, using a LSTM model on the financial sector

To summarize the model performance, we have plotted  $R^2$  score for all sectors from the different modes. We observe that the highest single score came from using the LSTM model with sentiment.



**Figure 6.3:**  $R^2$  scores on all sectors grouped by model

## 7 Discussion

Throughout this thesis we have examined five different regression models that used financial data, sentiment scores and VIX index in order to analyze how they predict market volatility. For this next section, we will discuss the results with emphasis on evaluating the relationships between volatility and news sentiment and its overall predicting powers. Lastly, we will discuss the limitations of our approach.

### 7.1 Predicting power of sentiment

In order to evaluate the effect of sentiment on our volatility predictions we will need to look at the difference between the models with exclusively financial data and the models that also applied sentiment scores. The main findings from the analysis when looking at the predicting power of sentiment alone indicate that there is a measurable relation between volatility and market sentiment. However, we saw that the difference in predicting volatility with or without sentiment was marginal overall. On one hand, some of the models with sentiment as a covariate did yield the best results, and shows how sentiment could be useful information in predicting volatility. On the other hand the results were incongruous for different models. Underlining the fact that the results was heavily dependent on the model used. It also shows that using sentiment scores is challenging if the relation between market sentiment and volatility is not strong enough for the models to use.

The analysis showed that the LSTM model attained the best results for predicting the total market volatility. Therefore, we believe the results from this model will have slightly higher credibility when evaluating the effect of sentiment. Using all sectors and adding the sentiment variable returned a 43.6%  $R^2$  score. Comparing this with the other models we saw that this was slightly better. This means the best model could use sentiment in order to make better predictions. Nevertheless, it is still a single observation, and it is difficult to claim a final conclusion based on this result alone. On the basis of this we would argue that sentiment does affect the volatility of the market somewhat, but the total effect is challenging to isolate, and a direct and causal link is hard to prove.

Given that our best results were achieved on individual sectors, it is befitting that we

discuss the effects of sentiment on this level as well. The findings on sector level are more varying and has a greater results span. This indicate that each sector has independent patterns in volatility and consequently have varying prediction results. On one side, Financial and Consumer Staples are two sectors that got unusable results on all models regardless of covariates. So, for these sectors it is hard to tell if sentiment had any effect. On the other side we had the energy sector proving to be the one with the most promising results. Attaining at best an  $R^2$  score of around 60% and higher with Random Forest and XgBoost for covariate combinations. This could be the result from a well build model, or it could be from the volatility in the energy sector being easier to predict, regardless of the model. An interesting point nonetheless is how different the results are based on the model applied. Despite what Li et al. (2014a) found about how the stock market is affected by news, our results indicate the amount of influence news has is likely highly dependent on the type of sector it is used in.

In several model combinations the worst results were achieved when including both VIX index and news sentiment. This is interesting and raises the question on how quantifiable sentiment is and whether one can include different measures of sentiment. The calculation of the sentiment score stems for a sentiment analysis, while the VIX index is calculated based on put and call options in the market. The latter is arguably an indirect measure of sentiment, and one could expect these two variables to be highly correlated. Nonetheless, our results indicate this might not be the case. One explanation could be that investors in the option market are motivated by other factors than the journalists producing the sentiment in news articles. Therefore, these covariates might create noise when combined. Another explanation could be that there is some lag associated with their relationship and only some models are able to detect this lagged structure. Lastly, there might be some sectors where options or news sentiment is not as important as other factors, thus creating noise when included.

The calculation of volatility could be a source of error in itself. It is hard to determine which type of calculation is best suited as there is no universal way to calculate volatility. GARCH models and other alternative ways to model volatility could be better suited for this task. We argue our fairly simple calculation could reduce the model complexity and therefore make it easier to isolate the effect of news sentiment, but this assumption might

not hold. Either way, it would be desirable to investigate the effect choice of volatility calculation has on the predicting power of news sentiment.

It is hard to say which way the relationship between news sentiment and volatility is moving. We cannot determine if the volatility is dependent on the news sentiment, or if the sentiment is dependent on the volatility. Our observations are on a daily level but there might be additional insights accessible if intraday data was available. Regardless, it is likely there exists a dual relationship where sentiment can affect volatility, but also where changes in volatility can affect or at least reinforce the changes in sentiment. Nonetheless, our results suggest sentiment from news headlines hardly is the driving factor for changes in volatility.

Another important discussion topic is how relevant and accurate the news are to the investors. Our news articles are collected through web-scraping from American financial papers and we are thus indirectly assuming this will cover the information sources of the investors owning shares in S&P100 companies. This might not be the case and these investors might get their information from other sources in other languages. One could argue the selected news sources are covering the majority of investors because S&P100 are American companies. If this is not the case, then our prediction might benefit from news from a broader array of financial papers. This is again a trade-off between including a broad specter of news and the trustworthiness of the news, where we chose to prioritize the latter. Liu et al. (2017) and Deveikyte et al. (2020) both found that sentiment improved their predictions of volatility but both studies are conducted with sentiment from social media in addition to news sentiment. This could indicate it might be beneficial to include a broader specter of news sentiment, as well as sentiment from alternative sources like social media.

The relevance of our news to specific companies is likely to be highly variable. We have prioritized to include all the financial news available during the time period, and have not verified or filtered the data based on relevance to the S&P100 companies. This is also the reason we are aggregating data to individual sectors and a combination of all sectors. The assumption is based on our sentiment being most relevant to market level changes. Despite these precautions, the question of whether the news are relevant is still present. Irrelevant news might create significant noise in our data and filtering news based on

relevance might be an key step when using sentiment to predicting stock volatility.

## 7.2 Economic application of volatility

Regardless of the effect of news *sentiment* on volatility, the models' ability to predict volatility can still be of use. Investment strategies such as At-The-Money straddle (Goltz and Lai, 2009), which is profitable when the volatility is high regardless of price direction, could be utilized with a good prediction model. Another opportunity is being able to predict risk. In contrast to Antweiler and Frank (2004) and Schumaker et al. (2012), we are trying to predict the volatility value directly and not classifying direction or categories of volatility. It might be the case that sentiment in volatility predictions is best utilized when provided with some leeway within a category. Investors does not necessary need a specific volatility but might benefit from knowing direction or what interval the prediction falls within. Specifically, investors could hedge investments or adjust their portfolio based on this information.

Some of the error in our predictions originates from the models overshooting the target value. When plotting predictions and true value together we observe two similar graphs. The correlation is between 60%-70% (seen in A.2 to A.9 for all models, but the predictions are usually shifted compared to the target value. This could once again indicate the predictions should be classified into bins or categories if they are going to have any practical value for an investor.

## 7.3 Limitations

The main limiting factor when working with sentiment is the accuracy one can obtain when working with large amount of data. Noise in the data will have an effect on the accuracy. Since not all news articles are going to be equally important in measuring market sentiment. The accuracy will vary, and the overall sentiment for one trading day has the potential to be skewed by headlines that are less relevant for the true sentiment.

Another drawback is the fact that not all sentiment data is incorporated in the models since the markets are closed on weekends and holidays. This causes all headlines that happen on those days to be exempted from the model and the information they hold will



not have any effect on the predictions. As mentioned above, the sentiment scores are not filtered on company or sector, but an overall representation of the market sentiment. This would indicate that sectors that follow the overall market closer will have better prediction models associated with them.

An additional limiting factor might be the use of data during the Covid-19 pandemic. Our assumption might not hold, and these volatile periods could maybe compromise the results despite volatile shocks in both train and test data. One could maybe argue these volatile periods are abnormal systematic risk events, and no model will be able to predict these accurately. Therefore, it might be beneficial to use a time span without known systematic risk events, creating shocks in the market.

In general, we found it challenging to determine which model was best at predicting volatility. This is in line with Brailsford and Faff (1996) who could not firmly decide on a model being superior to the others. Because of this, it is challenging to test the relevance of sentiment when there are existing issues related to predicting volatility with only financial data. Therefore, our ambiguous results present a need to explore additional volatility predictors. Poon and Granger (2003) suggests dollar rate as a potential predictor while Christiansen et al. (2012) suggests that all financial variables which has a sensible economic interpretation could be good predictors. Further, he is especially emphasizing variables capturing time-varying risk. Thus, interest rate differentials in foreign exchange and valuation ratios for equities are potential good candidates. These additional variables might contribute to build more consistent and robust models, on which the effect of news sentiment can better be observed.

From our results we observe XgBoost returning weak results compared to the other models. This is likely because of sub-optimal hyperparameter tuning, which can lead to overfitting and poor generalization. In line with the guidelines of Chen and Guestrin (2016) we explored tuning of *learning rate*, *minimum child weight*, *tree dept* and *number of trees* through a grid search. Despite this, our results indicate it could be beneficial to tune additional parameters. One could also argue some of this inferior performance originates from XgBoost's problems with extrapolation. This is still not likely, as Random Forest has the same issues but is without the significant drop in performance.

## 8 Conclusion

In this thesis we have analyzed the relationship between news sentiment and market volatility. Using five different machine learning models to make volatility predictions. By using different covariates, we attempted to isolate the effect of news sentiment on these predictions.

Our results indicate that news sentiment shows tendencies to contribute positively when predicting volatility. There are some contradictory results, and we cannot give a conclusive verdict. Model selection and type of industry seems to have a larger effect than including news sentiment or VIX index. Based on our analysis we would claim that isolating the effect of sentiment on volatility is especially hard because of the complex factors it is dependent on.

Lastly, we would say that although we were not able to replicate the result of previous studies, or make consistent improved models by including sentiment, we are convinced that market sentiment has an effect on volatility.

### 8.1 Further work

For further analysis on this subject, it might be beneficial to take a more granulated approach. Looking at the intraday volatility and collecting sentiment from more specific or filtered news headlines could be interesting. We think that this could give a more nuanced view on the relationship between news sentiment and stock volatility. Further, collecting data from a broader range of news and social media sources could also make the sentiment score more representative.

Another way to improve the sentiment analysis would be to create a custom financial lexicon that would have more focus on words that could affect volatility. This is a major task but could definitely improve the accuracy of the sentiment.

To make more consistent and robust models it might be advantageous to include macro economic variables to count for external market factors. Interest rates, inflation rates and employment rates could all potentially affect the volatility of a market.

Lastly, we would also urge future work to consider news bias in their sentiment analysis.

The internet provides an overload of information and to be able to find a combination of representative and un-biased news will be important. A potential solution could be to collect information from sources with different political viewpoints or sources from other countries.

## References

- Amanoul, S. V., Abdulazeez, A. M., Zeebare, D. Q., and Ahmed, F. Y. (2021). Intrusion detection systems based on machine learning algorithms. In *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)*, pages 282–287. IEEE.
- Anastasopoulos, A., Cox, C., Neubig, G., and Cruz, H. (2020). Endangered languages meet modern nlp. In *Proceedings of the 28th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 39–45.
- Antweiler, W. and Frank, M. Z. (2004). Is all that talk just noise? the information content of internet stock message boards. *The Journal of finance*, 59(3):1259–1294.
- Appel, O., Chiclana, F., and Carter, J. (2015). Main concepts, state of the art and future research questions in sentiment analysis. *Acta Polytechnica Hungarica*, 12(3):87–108.
- Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327.
- Brailsford, T. J. and Faff, R. W. (1996). An evaluation of volatility forecasting techniques. *Journal of Banking & Finance*, 20(3):419–438.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- Brooks, C. and Persaud, G. (2003). Volatility forecasting for risk management. *Journal of forecasting*, 22(1):1–22.
- Burkov, A. (2019). *The hundred-page machine learning book*, volume 1. Andriy Burkov Quebec City, QC, Canada.
- Cambria, E., Das, D., Bandyopadhyay, S., and Feraco, A. (2017). Affective computing and sentiment analysis. In *A practical guide to sentiment analysis*, pages 1–10. Springer.
- Carta, S. M., Consoli, S., Piras, L., Podda, A. S., and Recuperio, D. R. (2021). Explainable machine learning exploiting news and domain-specific lexicon for stock market forecasting. *IEEE Access*, 9:30193–30205.
- Chai, C. P. (2022). Comparison of text preprocessing methods. *Natural Language Engineering*, pages 1–45.
- Chandriah, K. K. and Naraganahalli, R. V. (2021). Rnn/lstm with modified adam optimizer in deep learning approach for automobile spare parts demand forecasting. *Multimedia Tools and Applications*, 80(17):26145–26159.
- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Chong, C. W., Ahmad, M. I., and Abdullah, M. Y. (1999). Performance of garch models in forecasting stock market volatility. *Journal of forecasting*, 18(5):333–343.

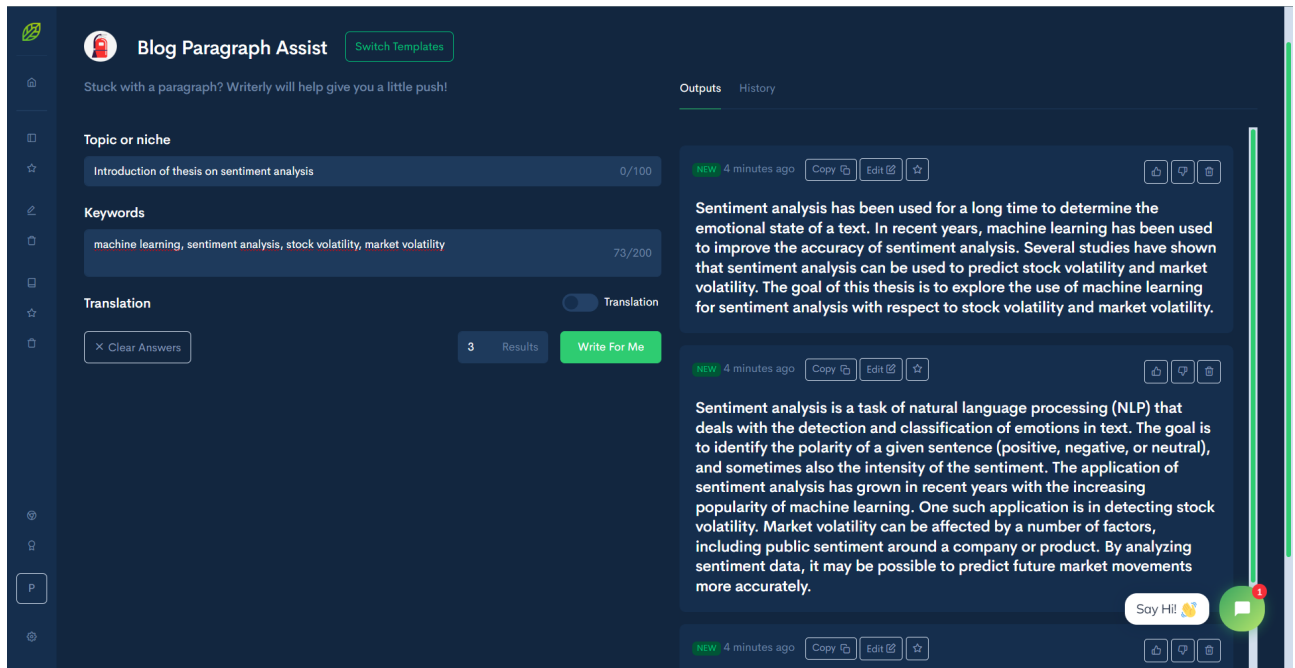
- Christiansen, C., Schmeling, M., and Schrimpf, A. (2012). A comprehensive look at financial volatility prediction by economic variables. *Journal of Applied Econometrics*, 27(6):956–977.
- Christopher, D. M., Prabhakar, R., and Hinrich, S. (2008). Introduction to information retrieval.
- Deveikyte, J., Geman, H., Piccari, C., and Provetti, A. (2020). A sentiment analysis approach to the prediction of market volatility. *arXiv preprint arXiv:2012.05906*.
- Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007.
- Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.
- Franses, P. H. (2016). A note on the mean absolute scaled error. *International Journal of Forecasting*, 32(1):20–22.
- 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."
- Goltz, F. and Lai, W. N. (2009). Empirical properties of straddle returns. *The Journal of Derivatives*, 17(1):38–48.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Grefenstette, G. (1999). Tokenization. In *Syntactic Wordclass Tagging*, pages 117–133. Springer.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.
- Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.
- Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.
- Hyndman, R. J. (2006). Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4(4):43–46.
- Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688.

- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.
- Joseph, K., Wintoki, M. B., and Zhang, Z. (2011). Forecasting abnormal stock returns and trading volume using investor sentiment: Evidence from online search. *International Journal of Forecasting*, 27(4):1116–1127.
- Jurek, A., Mulvena, M. D., and Bi, Y. (2015). Improved lexicon-based sentiment analysis for social media analytics. *Security Informatics*, 4(1):1–13.
- Ko, C.-N. and Lee, C.-M. (2013). Short-term load forecasting using svr (support vector regression)-based radial basis function neural network with dual extended kalman filter. *Energy*, 49:413–422.
- Krouska, A., Troussas, C., and Virvou, M. (2016). The effect of preprocessing techniques on twitter sentiment analysis. In *2016 7th international conference on information, intelligence, systems & applications (IISA)*, pages 1–5. IEEE.
- Lawler, G. F. and Limic, V. (2010). *Random walk: a modern introduction*, volume 123. Cambridge University Press.
- Li, Q., Wang, T., Li, P., Liu, L., Gong, Q., and Chen, Y. (2014a). The effect of news and public mood on stock movements. *Information Sciences*, 278:826–840.
- Li, X., Liang, C., and Ma, F. (2022). Forecasting stock market volatility with a large number of predictors: New evidence from the ms-midas-lasso model. *Annals of Operations Research*, pages 1–40.
- Li, X., Xie, H., Chen, L., Wang, J., and Deng, X. (2014b). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23.
- Liu, Y., Qin, Z., Li, P., and Wan, T. (2017). Stock volatility prediction using recurrent neural networks with sentiment analysis. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 192–201. Springer.
- McDonald, G. C. (2009). Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1):93–100.
- Miles, J. (2005). R-squared, adjusted r-squared. *Encyclopedia of statistics in behavioral science*.
- Mitchell, R. (2018). *Web scraping with Python: Collecting more data from the modern web*. " O'Reilly Media, Inc."
- Mittnik, S., Robinzonov, N., and Spindler, M. (2015). Stock market volatility: Identifying major drivers and the nature of their impact. *Journal of banking & Finance*, 58:1–14.
- Nadkarni, P. M., Ohno-Machado, L., and Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551.
- Plisson, J., Lavrac, N., Mladenic, D., et al. (2004). A rule based approach to word lemmatization. In *Proceedings of IS*, volume 3, pages 83–86.

- Poon, S.-H. and Granger, C. W. (2003). Forecasting volatility in financial markets: A review. *Journal of economic literature*, 41(2):478–539.
- Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., and Benevenuto, F. (2016). Sentibench—a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29.
- Richardson, L. (2007). Beautiful soup documentation. *Dosegljivo*: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. [Dostopano: 7. 7. 2018].
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.
- Saif, H., Fernández, M., He, Y., and Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of twitter.
- Sarica, S. and Luo, J. (2021). Stopwords in technical language processing. *Plos one*, 16(8):e0254937.
- Schumaker, R. P., Zhang, Y., Huang, C.-N., and Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems*, 53(3):458–464.
- Segal, M. R. (2004). Machine learning benchmarks and random forest regression.
- Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.
- Vapnik, V. (1999a). *The nature of statistical learning theory*. Springer science & business media.
- Vapnik, V. N. (1999b). An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999.
- Viegas, F., Alvim, M. S., Canuto, S., Rosa, T., Gonçalves, M. A., and Rocha, L. (2020). Exploiting semantic relationships for unsupervised expansion of sentiment lexicons. *Information Systems*, 94:101606.
- Wang, Y., Pan, Z., Zheng, J., Qian, L., and Li, M. (2019). A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science*, 364(8):1–13.
- Writerly-AI (2022). URL: <https://writerly.ai>.
- Zhang, H., Nettleton, D., and Zhu, Z. (2019). Regression-enhanced random forests. *arXiv preprint arXiv:1904.10416*.
- Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., and Liu, B. (2011). Combining lexicon-based and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89:1–8.
- Zhao, B. (2017). Web scraping. *Encyclopedia of big data*, pages 1–3.
- Zhou, D., Zuo, X., and Zhao, Z. (2022). Constructing a large-scale urban land subsidence prediction method based on neural network algorithm from the perspective of multiple factors. *Remote Sensing*, 14(8):1803.

# Appendix

## A Additional figures



The screenshot displays the 'Blog Paragraph Assist' interface. On the left, the 'Topic or niche' is 'Introduction of thesis on sentiment analysis' (0/100) and 'Keywords' are 'machine learning, sentiment analysis, stock volatility, market volatility' (73/200). A 'Write For Me' button is visible. The main area shows two generated paragraphs. The first paragraph reads: 'Sentiment analysis has been used for a long time to determine the emotional state of a text. In recent years, machine learning has been used to improve the accuracy of sentiment analysis. Several studies have shown that sentiment analysis can be used to predict stock volatility and market volatility. The goal of this thesis is to explore the use of machine learning for sentiment analysis with respect to stock volatility and market volatility.' The second paragraph reads: 'Sentiment analysis is a task of natural language processing (NLP) that deals with the detection and classification of emotions in text. The goal is to identify the polarity of a given sentence (positive, negative, or neutral), and sometimes also the intensity of the sentiment. The application of sentiment analysis has grown in recent years with the increasing popularity of machine learning. One such application is in detecting stock volatility. Market volatility can be affected by a number of factors, including public sentiment around a company or product. By analyzing sentiment data, it may be possible to predict future market movements more accurately.'

Figure A.1: AI generated introduction



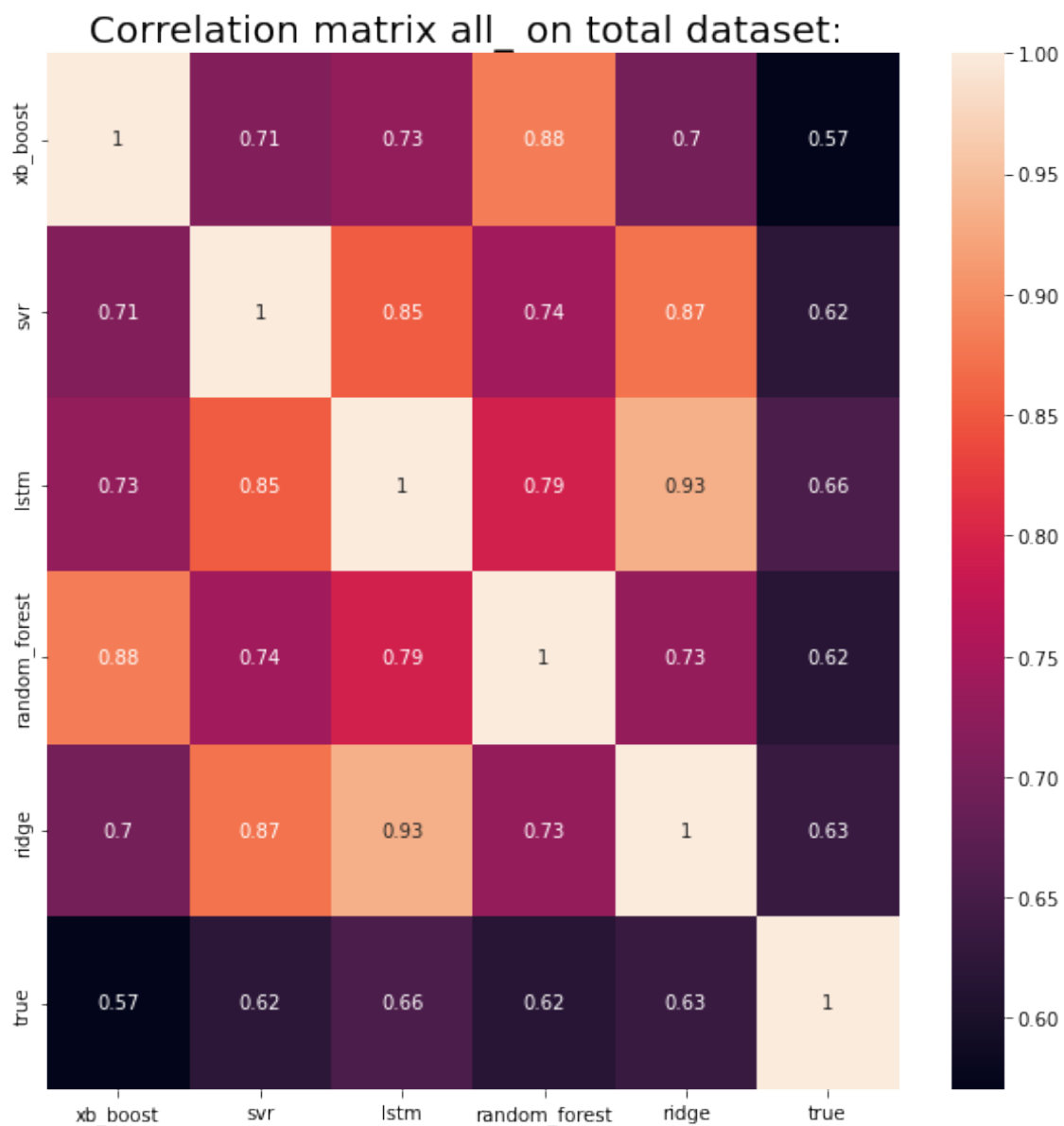


Figure A.2: Correlation of predictions with all covariates

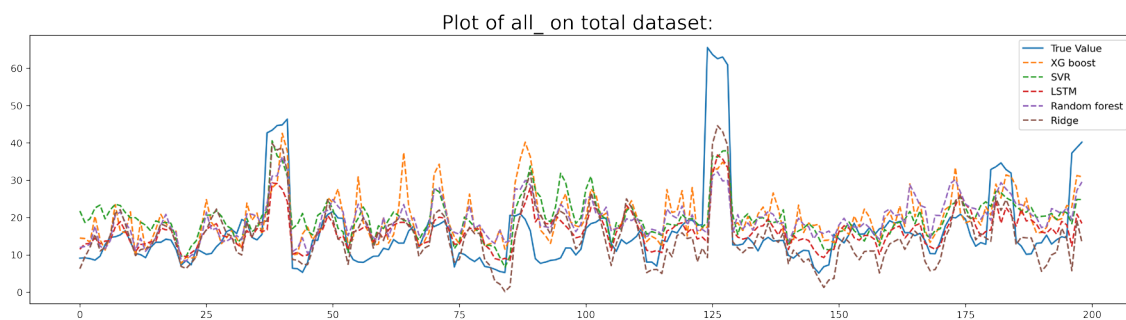
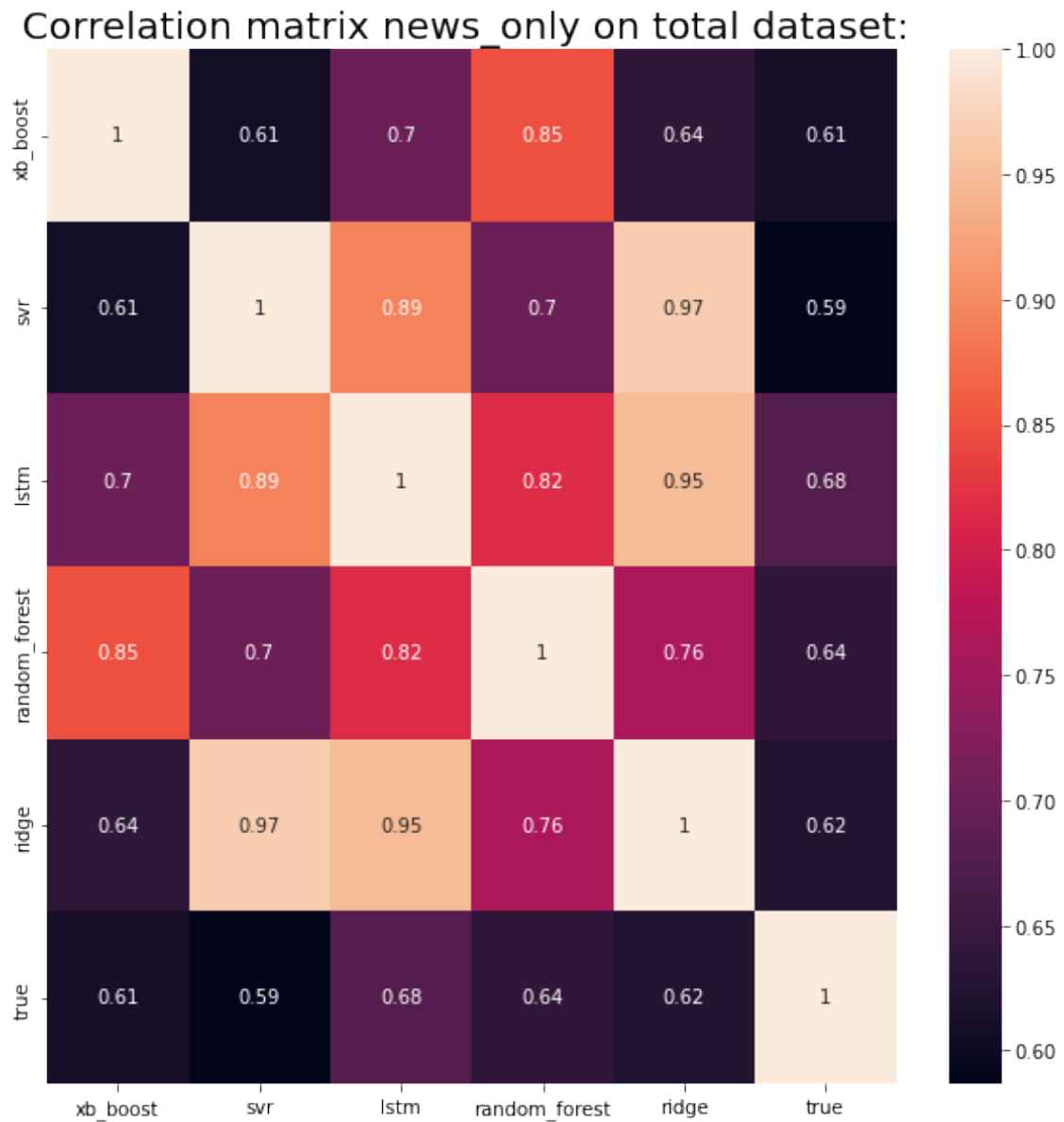
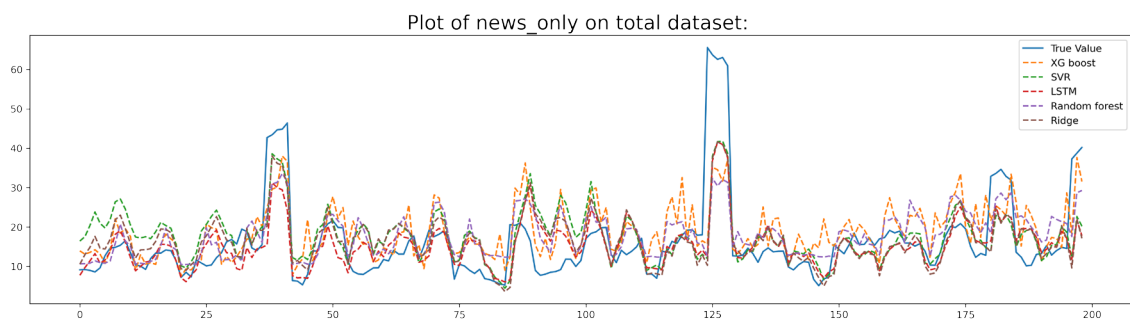


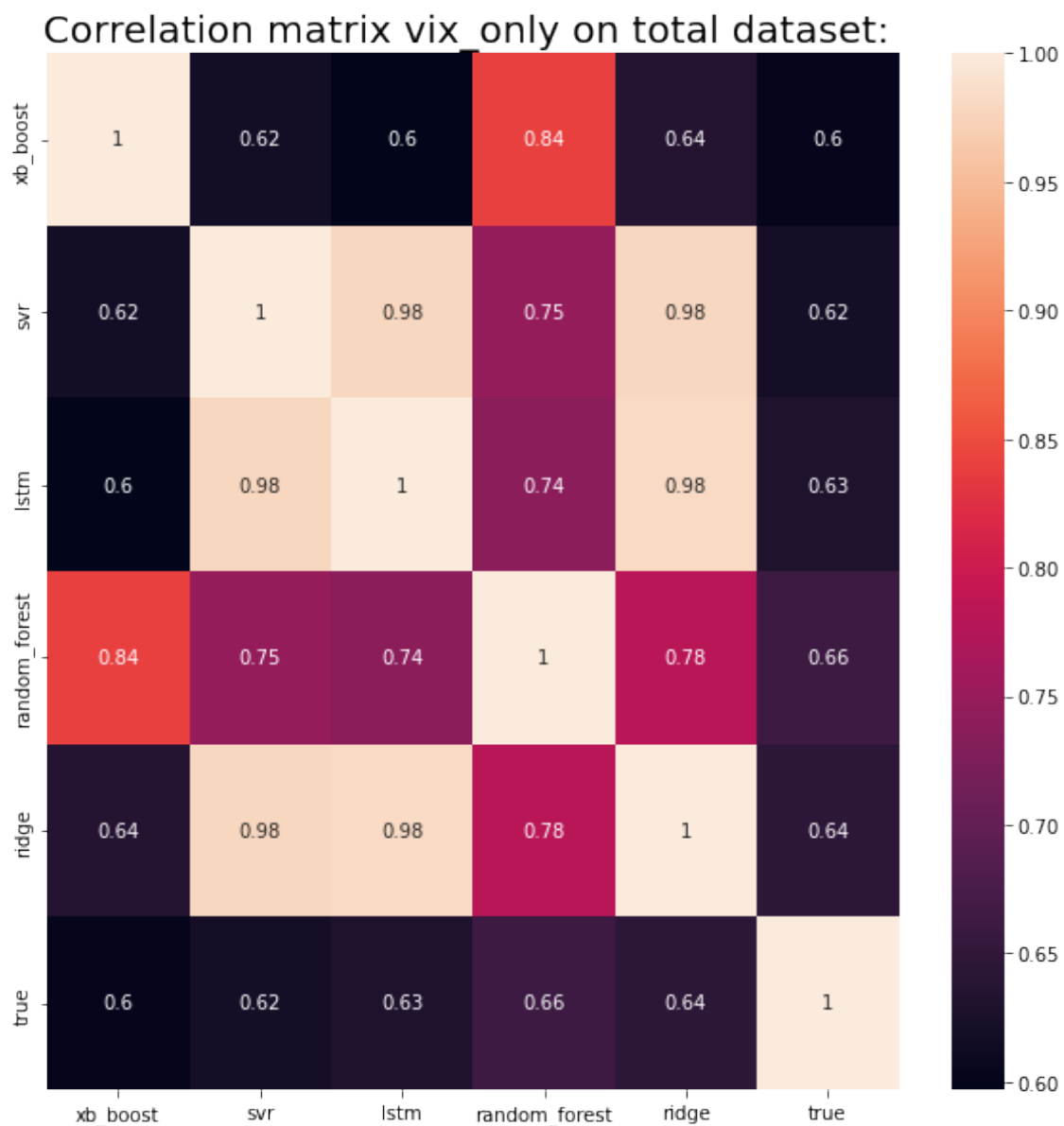
Figure A.3: Time series plot of predictions from all models with all covariates



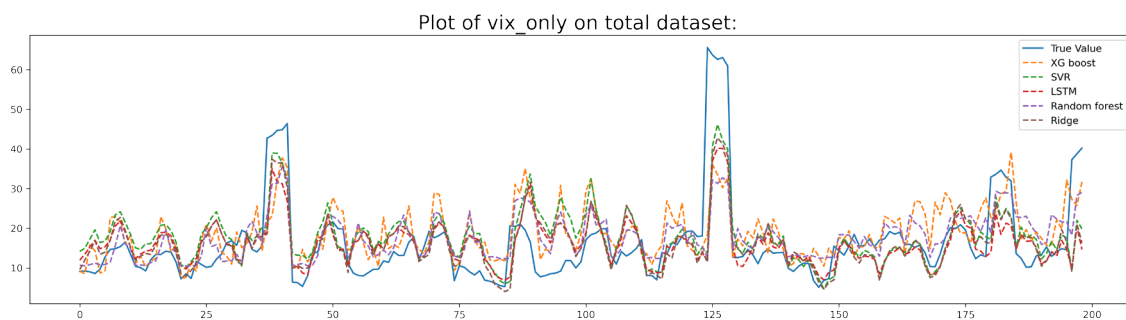
**Figure A.4:** Correlation of predictions with new sentiment only



**Figure A.5:** Time series plot of predictions from all models with news sentiment only

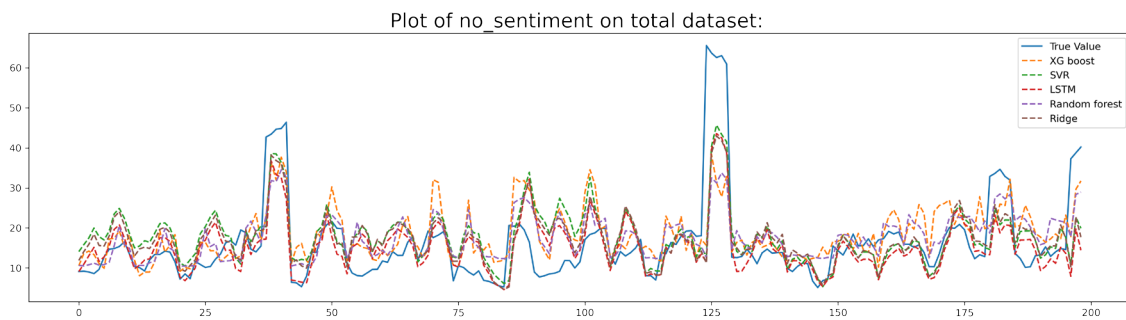


**Figure A.6:** Correlation of predictions with VIX index only



**Figure A.7:** Time series plot of predictions from all models with VIX index only

Correlation matrix no\_sentiment on total dataset:

**Figure A.8:** Correlation of predictions without VIX index and sentiment**Figure A.9:** Time series plot of predictions from all models without VIX index and sentiment

<i>TIC</i>	AVGO	MSFT	MA	ADBE	QCOM	IBM	NVDA	V	CSCO	TXN	AMD	ORCL	INTC	AAPL
AVGO	100%	-1%	1%	0%	7%	-3%	4%	3%	-5%	-2%	7%	4%	-5%	-7%
MSFT	-1%	100%	-2%	0%	5%	1%	1%	5%	-6%	-2%	7%	-2%	-4%	-3%
MA	1%	-2%	100%	3%	-1%	-1%	3%	-1%	0%	5%	1%	0%	-2%	-6%
ADBE	0%	0%	3%	100%	4%	-3%	-2%	3%	5%	-1%	3%	8%	-4%	-3%
QCOM	7%	5%	-1%	4%	100%	-5%	3%	0%	2%	1%	11%	0%	-3%	-7%
IBM	-3%	1%	-1%	-3%	-5%	100%	-2%	0%	5%	-1%	-2%	3%	-2%	0%
NVDA	4%	1%	3%	-2%	3%	-2%	100%	0%	3%	2%	1%	-2%	7%	-2%
V	3%	5%	-1%	3%	0%	0%	0%	100%	-4%	2%	1%	3%	0%	2%
CSCO	-5%	-6%	0%	5%	2%	5%	3%	-4%	100%	0%	2%	2%	-4%	5%
TXN	-2%	-2%	5%	-1%	1%	-1%	2%	2%	0%	100%	3%	-2%	3%	-1%
AMD	7%	7%	1%	3%	11%	-2%	1%	1%	2%	3%	100%	-3%	-1%	-4%
ORCL	4%	-2%	0%	8%	0%	3%	-2%	3%	2%	-2%	-3%	100%	-3%	-2%
INTC	-5%	-4%	-2%	-4%	-3%	-2%	7%	0%	-4%	3%	-1%	-3%	100%	6%
AAPL	-7%	-3%	-6%	-3%	-7%	0%	-2%	2%	5%	-1%	-4%	-2%	6%	100%

**Figure A.10:** Correlation matrix of individual companies in the information technology sector