Norwegian School of Economics,

Bergen, Fall 2022

Prediction of Stock Market Volatility Utilizing Sentiment from News and Social Media Texts

A study on the practical implementation of sentiment analysis and deep learning models for predicting day-ahead volatility

Martin Berge & Lars R. Bessesen

Supervisor: Christian Langerfeld

Master thesis, Economics and Business Administration
Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Preface

This thesis is submitted in partial fulfillment of the requirements for the master's degree in Business and Administration with a Business Analytics major at The Norwegian School of Economics. It is written during the fall of 2022.

We would like to express our gratitude to our supervisor Christian Langerfeld for stellar guidance throughout working on this thesis. Additionally, we would like to highlight the course *"BAN432: Applied Textual Data Analysis for Business and Finance"* for providing the necessary skillset and further motivation to explore the field of natural language processing. We extend our gratitude to the course responsible and Norwegian School of Economics.

Lastly, we would like to thank family and friends for their support throughout the process of writing this thesis.

Norwegian School of Economics

Bergen, December 2022

X **X**

Lars Røed Bessesen Martin Berge

Abstract

This thesis studies the impact of sentiment on the prediction of volatility for 100 of the largest stocks in the S&P500 index. The purpose is to find out if sentiment can improve the forecast of day-ahead volatility wherein volatility is measured as the realized volatility of intraday returns.

The textual data has been gathered from three different sources: Eikon, Twitter, and Reddit. The data consists of respectively 397 564 headlines from Eikon, 35 811 098 tweets, and 4 109 008 comments from Reddit. These numbers represent the uncleaned data before filtration. The data has been collected for the period between 01.08.2021 and 31.08.2022.

Sentiment is calculated by the FinBERT model, an NLP model created by further pre-training of the BERT model on financial text. To predict volatility with the sentiment from FinBERT, three different deep learning models have been applied: A feed forward neural network, a recurrent neural network, and a long short-term memory model. They are used to solve both regression and classification problems.

The inference analysis shows significant effects from the computed sentiment variables, and it implies that there exists a correlation between the number of text items and volatility. This is in line with previous literature on sentiment and volatility. The results from the deep learning models show that sentiment has an impact on the prediction of volatility. Both in terms of lower MSE and MAE for the regression problem and higher accuracy for the classification problem.

Moreover, this thesis looks at potential weaknesses that could influence the validity of the results. Potential weaknesses include how sentiment is represented, noise in the data, and the fact that the FinBERT model is not trained on financial oriented text from social media.

Contents

Figures

Tables

Equations

1 Introduction

The amount of unstructured information available to investors is ever increasing. It is a combination of news media, alternative sources of information, voluntary filings, and regulatory filings. These large quantities of data are impossible to cover for a single person or analyst. The result of this has sent both institutional and retail investors searching for new tools to gain an edge in a market teeming with information. Because of this, the field of natural language processing (NLP) has seen a massive influx of users wanting to adapt NLP to financial markets. NLP originated in the 1940's, after World War II, to create a machine that could translate language automatically. However, with the introduction of the Internet, NLP has become more focused on extraction of information (Gallagher et al., 2004) . Within the space of finance, complex models and methods developed by data scientists and linguists have allowed investors to consume and analyze information from large amounts of textual data.

With the introduction of NLP to finance, the main research area has revolved around the prediction of stock price movement. The key technique for this area has been analyzing text to identify and categorize the writers' attitude towards the topic in question, also known as sentiment analysis. Early research was based on dictionaries and some machine learning. In recent years, deep learning has enabled models to capture more of the meaning in text.

The prediction of stock price movement by the use of sentiment analysis has mostly focused on classifying the direction of the asset as either higher or lower than the day before. Another, less explored way to make use of NLP in finance, is to predict volatility. Forecasting volatility is a fundamental part of financial markets. Volatility can be interpreted as uncertainty, and it is divided primarily into two types of volatility, historical volatility and implied volatility (Hayes, 2022).

This thesis utilizes NLP in order to predict the historical, realized volatility. It predicts the volatility of individual stocks through day-ahead forecasts and by classifying the next day's volatility as higher or lower than for the previous day. The thesis introduces sentiment as predictors, calculated by analyzing sentiment of news and social media text with a state-of-the-art deep learning model called FinBERT.

1.1 Motivation

The question of whether news impacts market volatility or market volatility generates news resembles an apparent causality problem like the chicken or the egg problem. The paper *"Language, news and volatility"* by Byström (2016) studies the relationship between news and volatility. The study finds that there is a significant relationship between news leading to volatility rather than the opposite.

In today's age investors utilize other sources of information than just traditional news. Social media platforms act as a public place for retail investors to communicate and share information. Such information can influence stock movement and an extreme example of this is the recent GameStop case. This thesis investigates if the sentiment of news and social media text could predict day-ahead volatility. It is based on the findings of Hans Byström, that news leads to volatility.

Whereas studies often focus on indices for measuring the effect of sentiment, this thesis will investigate the effect on firm-level movement of multiple assets. This decision ensures sufficient amount of data, and it provides a different prediction objective than prior studies. While the movement of indices such as DJIA and S&P500 can represent investors' confidence in the market, the individual stocks capture firm-level movements. Firm-level movement can be used in portfolio management, value at risk models, or when developing pairs trading strategies including multiple pairs of securities. This leads to the need for a model able to capture firm-level movement from both traditional news sources and social media.

This thesis proposes the idea of constructing a general model which analyzes volatility for the 100 largest stocks in the S&P500 index with sentiment as explanatory variable. The models are deep learning (DL) models. This ensures that potential non-linear relations and interactions in the data are captured.

1.2 Research Question

"Can the sentiment of public available information in news and social media aid in prediction of stock market volatility?"

2 Literature Review

In recent days, sentiment analysis has been one of the most popular ways of applying natural language processing (NLP) in finance. Utilizing textual data from corporate filings, reports, news, social media, earnings calls, and forum posts. Sentiment analysis of textual analysis is divided into two main approaches, lexicon-based, and machine learning-based (Li, et al., 2016). In Kearny and Liu (2013), models and approaches to sentiment in textual analysis prior to 2013 have been reviewed. The work has primarily been lexicon-based and machine learning-based (ML). After the review study by Kearny and Liu, a wave of deep learning models and frameworks have further enhanced the capabilities of sentiment analysis. This literature review presents the different approaches to sentiment analysis in finance. It includes an extensive but not exhaustive review of important papers and developments.

2.1 An Introduction to Sentiment

Definition of Sentiment Analysis

Sentiment analysis can be understood as analysis performed in order to extract opinions. Most often related to polarity. In this thesis, sentiment itself is defined as the collective representation of polarity in the market, represented by news and social media text. With the vast amounts of text generated online every day, sentiment analysis can leverage this text data and generate insights and subsequently better decision making.

Lexicon-Based Approaches

Lexicon-based also known as dictionary-based methods rely on dictionaries or wordlists (Boghe, 2020). Each word or feature has its own corresponding polarity. Creating dictionaries is time consuming work and since words can have different meanings in different settings or fields. The sentiment value or label can therefore differ a lot from one setting to another. The process of coding the features is often performed by linguists or specialists.

Lexicon-based sentiment analysis is performed on a preprocessed text. The corpus has been tokenized, the tokens transformed to their lemma and the stop words removed. The tokens are then given their corresponding polarity-score. The sentiment is often calculated as the average sentiment of the tokens in the text. The models are categorized as bag-of-words models as the linear ordering of words within the context is ignored (Kearny & Liu, 2013). Bag-of-words refers to an orderless representation of a document where the words are represented as frequencies (Niebles & Krishna, 2017). To capture the meaning of word sequences, a technique call N-grams can be applied (Jurafsky & Martin, 2021). N refers to the number of words in the expression. The segmentation into N-grams is performed since some words can have different meanings when put next to each other, like "very good" and "not very good". Another lexicon-based approach is to have rule-based lexicons. This can have a significant impact as these rules can apply different sentiment if the occurrence of words together changes the meaning or degree. The VADER model is an example of such a model (Yalçın, 2020).

Approaches Based on Word Embedding Models

Word embedding models follow the distributional hypothesis, that the context is important for the meaning of a word (Young et al., 2017). This is used to capture similarity between words. Word embedding models have been a key factor and precursor to the increased use of machine learning and deep learning approaches in NLP, and the improved results from prior word vectors. Mikolov et al. (2013) introduced the Word2Vec model. Which combines a continuous bag of words model (CBOW) and a skip-gram model (Mikolov, et al., 2013). Respectively, the CBOW predicts the conditional probability of the center word given the target words, and the skip-gram predicts the conditional probability of the context words given the center word. Other popular word embedding models are the GloVe (Pennington, et al., 2014) and ELMo (Peters, et al., 2018).

Machine Learning Approaches

Machine learning is another approach that together with dictionaries, dominated the field of textual analysis until the early 2010's. The models learn to classify text and then the sentiment is calculated by combinations of the classifications. Machine learning models can in general be divided into two categories: Unsupervised and supervised. There is a third type of model, reinforcement learning, but this type of learning will be disregarded as it is not applicable for the textual analysis in this thesis. Supervised models are based on a training set of labeled data. The ML model is then trained by applying a classification algorithm. An unsupervised model will train without labeled data and is commonly used to classify unstructured data. Popular classification algorithms include Naïve Bayes and Support Vector Machines (SVM). Naïve Bayes is a simplistic model used for classification, and it is based on Bayes' theorem which provides an expression for the posterior probability as a combination of the prior and the density function (James, et al., 2021). The SVM is based on support vectors, the observations used to create the separating hyperplane. Both methods are described in detail in James et al. (2021).

Deep Learning Approaches

Christopher Manning writes that the deep learning tsunami hit the NLP conferences in 2015 (Manning C. , 2015). From 2015 and until now, tremendous progress has been made. Several deep neural network models have been popular. This includes CNNs[1], RNNs[2], LSTM[3], BERT[4] and Open AIs GPT[5] models. Deep learning models are highly flexible and allow models to capture and understand key information and semantics. RNNs, LSTM and BERT will be explained in detail later.

2.2 Literature on Sentiment Analysis for Prediction in Finance

One of the first applications of textual analysis was introduced by Frazier et al. (1980). Robert W. Ingram and Katherine Beal Fraizer looked at social responsibility disclosures, categorized the content and looked at the correlation between the contents and indices from the Council on Economic Priorities(CEP). When the Sarbanes-Oxley Act was introduced in 2002, social responsibility filings increased. The federal act was passed to improve auditing and public disclosure, in order to avoid scandals like Enron (Wex Definitions Team, 2021). This act has led to a greater degree of disclosure and ever-increasing amounts of textual data. In recent times, the analysis of social media has also increased in importance ever since the SEC announced that public disclosures could be made through the media platform Twitter (SEC, 2013).

In the 2000's, several impactful studies were conducted on NLP in finance. Coval & Shumway (2001) had some interesting findings in *"Is Sound Just Noise?"*. The study looked at the ambient noise level in the Chicago Board of Trade's 30-year Treasury Bond futures trading pit. Both volume and volatility were linked with noise levels, but not returns. While noise is in the periphery of what could be defined as NLP, the study investigates a similar dynamic as stock mentions in social media. Comparing pit traders to social media users.

[1] Convolutional Neural Network - CNN

[2] Recurrent Neural Network - RNN

[3] Long Short Term Memory - LSTM

[4] Bidirectional Encoder Representations from Transformers - BERT

[5] Generative Pre-trained Transformer - GPT

Later, Antweiler & Frank (2002) studied messages of online chat boards. The messages were limited to the Dow Jones Industrial Average and the Dow Jones Internet Commerce Index. The methods used to classify messages as “buy,” “sell”, or “hold” were Naïve Bayes and Support Vector Machine. An interesting discovery was that an above average number of messages could forecast high levels of volatility. In addition, they found that the chat board messages were predictive even when controlling for impact from news media reporting. To some extent message boards could be an even earlier source of information than news media.

Another impactful paper is *“Giving Content to Investor Sentiment: The Role of Media in the Stock Market”* by Tetlock (2005). He utilizes the General Inquirer’s Harvard IV-4 psychosocial dictionary. The dictionary is used to categorize sentiment in the Wall Street Journal column “Abreast of the Market”. He found that high values of pessimism, negative sentiment, is related to negative returns. Furthermore, very high or low values leads to higher volume. Regarding the prediction of volatility, he discovered a weak link between pessimism and volatility.

In Kothari et al. (2009) the GI/Harvard IV-4 dictionary was also used when they analyzed the effects of disclosures by management, analysts, and financial press in 2009. Among several discoveries, they found that negative disclosures by financial press resulted in increased cost of capital and volatility.

In finance the four most commonly used dictionaries have been GI/Harvard IV-4 Dictionary, Diction, Henry, and the Loughran–McDonald Financial Dictionary (Loughran & McDonald, 2016). A survey of methods and models financial researchers have used was presented in Kearny & Liu (2013). Both the GI/Harvard IV-4 Dictionary, and Diction are examples of dictionaries not designed for financial text, but they have still been used for texts like news articles, earnings calls, and SEC filings (Loughran & McDonald, 2016). A known weakness of the GI/Harvard IV-4 is the misclassification of words in financial context.

Henry was the first wordlist created especially for financial text and was created from earnings press releases for the telecommunications and computer services industries (Loughran & McDonald, 2016). A weakness being few negative words, only 85. It was used in “*Earnings Conference Call Content and Stock Price: The Case of REITs*” by Doran et al. (2012) where they measured the tone of REIT earnings calls and found that a positive tone could almost offset the negative impact of a negative earnings surprise.

In “*When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks*” by Loughran & McDonald (2010), they found that 75% of the negative words in the Harvard GI dictionary is not pessimistic when used in a financial context. They created their own word list, here referred to as Loughran–McDonald Financial Dictionary. It contains six word lists: Negative, positive, uncertainty, litigious, strong modal, and weak modal. The words are gathered from 10-K filings in the period from 1994-2008. Making it a domain specific dictionary in addition to being more extensive than Henry. Both the Loughran–McDonald Financial Dictionary and Henry are dictionaries specifically designed for financial data.

According to behavior economics, emotions will impact behavior and decisions. Bollen et al. (2011) investigated if mood measured from Twitter feeds are correlated with the value of Dow Jones Industrial Average. The use of textual data from social media to measure public mood is interesting and closely linked to measuring sentiment. They found that up or down predictions of DJIA returns could be improved by some of the mood predicators. By utilizing a Self-Organizing Fuzzy Neural Network, they achieved an accuracy of 86.7%.

In “*Decision support from financial disclosures with deep neural networks and transfer learning”* by Kraus & Feuerriegel (2017) they looked at how deep learning could enhance financial decision making. They achieved higher accuracy while predicting stock price movement than traditional machine learning methods. From this, it can be seen that more flexible methods like RNN and LSTM can capture dependencies in text that simpler machine learning methods based on bag of word representation like the SVM cannot. Capturing these dependencies and non-linear relationships are essential in order to achieve superior predictive performance.

In 2018, a team from Google introduced the BERT model (Devlin, et al., 2018). Deep learning models such as BERT vastly improved the sentiment analysis by allowing for deeper semantic understanding of text. What makes the BERT model even more powerful is the ability to enhance results by further pre-train models on domain specific text. For instance, financial text. In this thesis, the FinBERT[6] model developed by Prosus, is used to perform sentiment analysis. The model is presented in the work by Araci (2019).

[6] Financial Bidirectional Encoder Representations from Transformers - FinBERT

3 Data

In prior research on textual data for financial forecasting, the textual data have consisted of annual reports, earnings press releases, earnings conference calls, news articles, analyst reports, and social media (Kearny & Liu, 2013). The data in this thesis is a combination of text from both news headlines, tweets, and Reddit comments. As described in the literature review, social media serves as a communication channel for financial markets in addition to being an early purveyor of information, hence it includes important data for sentiment prediction. This chapter covers the process of collecting and pre-processing textual data from Eikon, Twitter, and Reddit.

3.1 Data Sources

The data in this thesis consists of both textual data and price data. The textual data is collected from the social media platforms Reddit and Twitter in addition to Eikon, a data service delivered by Refinitiv. The price data is collected from AlphaVantage.

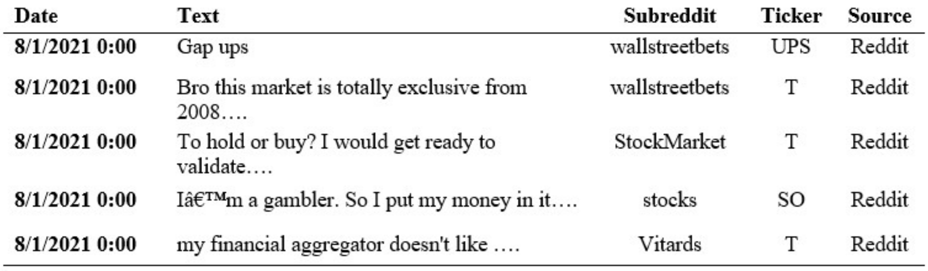
The data contains text and intra-day trading prices for the 100 largest stocks in the S&P500 index by market capitalization. The list of tickers can be seen in Appendix A 1. The retrieval process from each source is covered in the following subsections.

Reddit

Reddit is a social media forum where users interact through a message board. The site enables discussion among communities known as subreddits where specific subreddits are dedicated towards the overall category of the topics. Therefore, the data collected from Reddit is specific to subreddits dedicated to finance and stocks, the list of subreddits can be found in Appendix A 2.

In order to retrieve relevant comments containing the tickers of interest, every comment made on Reddit for the period 01.08.2021 to 31.08.2022 has been downloaded. These comments were collected through the *pushshift.io* Reddit API and compressed to ".zst" files. *"The Pushshift Reddit Dataset"* by Baumgartner et al. (2020) reviews the API in-depth.

A multi-processing script[7] was used to extract the comments in the specified subreddits. After the relevant subreddits were extracted, a modified version of the multi-processing script was applied to parse the comments for each of the tickers. Table 1 exemplifies the format of a retrieved object.



Date	Text	Subreddit	Ticker	Source
8/1/2021 0:00	Gap ups	wallstreetbets	UPS	Reddit
8/1/2021 0:00	Bro this market is totally exclusive from 2008….	wallstreetbets	T	Reddit
8/1/2021 0:00	To hold or buy? I would get ready to validate….	StockMarket	T	Reddit
8/1/2021 0:00	Iâ€™m a gambler. So I put my money in it….	stocks	SO	Reddit
8/1/2021 0:00	my financial aggregator doesn't like ….	Vitards	T	Reddit

Table 1: Retrieved object from Reddit

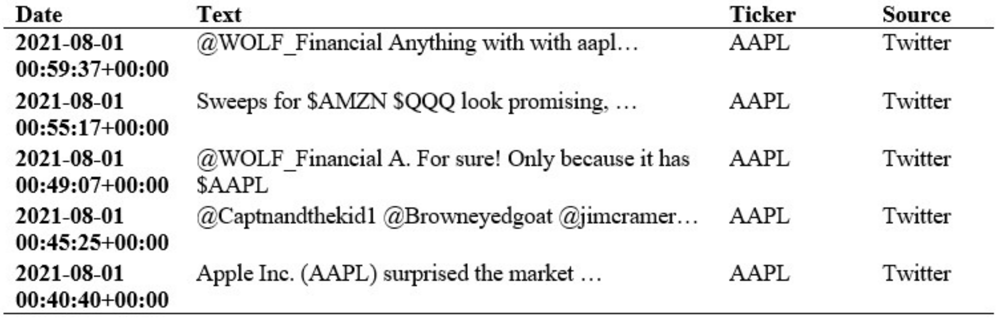
Twitter

The data collection process of gathering tweets differs from the extraction of Reddit comments. The Twitter data was not collected through an API due to limitations on extraction of data. The *SNScrape* web scraper has been utilized instead. The scraper utilizes the Twitter search field to look up the specified query, it then proceeds to collect tweets for the prompted period.

Due to the high volume of text that could be retrieved from Twitter, a limit of tweets per hour had to be implemented. The scraper was set to collect a maximum of 60 tweets per hour across the time range instead of a specified number of tweets between 01.08.2021 and 31.08.2022. This ensures an even collection of text throughout the time range instead of collecting a fixed amount of the latest posted tweets. The computational cost of not implementing this restriction would be very high and not feasible for this thesis.

After the object was retrieved, the keys and values relevant to the analysis were saved. This includes date, text, ticker, and source. Table 2 visualizes the structured object after retrieval.

[7] https://github.com/Watchful1/PushshiftDumps/blob/master/scripts/combine_folder_multiprocess.py#L156 created by *Watchful1*

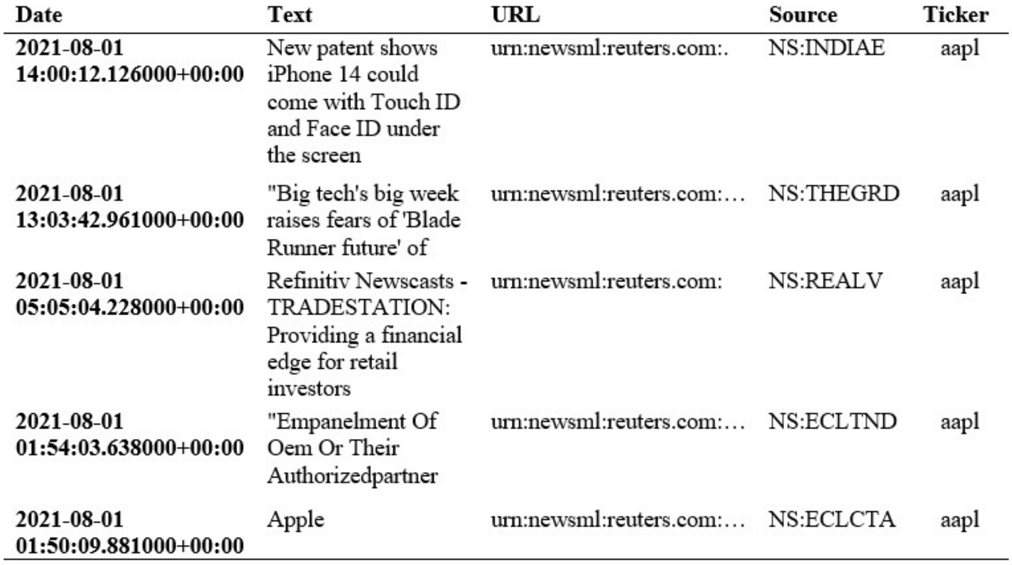


Date	Text	Ticker	Source
2021-08-01 00:59:37+00:00	@WOLF_Financial Anything with with aapl…	AAPL	Twitter
2021-08-01 00:55:17+00:00	Sweeps for $AMZN $QQQ look promising, …	AAPL	Twitter
2021-08-01 00:49:07+00:00	@WOLF_Financial A. For sure! Only because it has $AAPL	AAPL	Twitter
2021-08-01 00:45:25+00:00	@Captnandthekid1 @Browneyedgoat @jimcramer…	AAPL	Twitter
2021-08-01 00:40:40+00:00	Apple Inc. (AAPL) surprised the market …	AAPL	Twitter

Table 2: Retrieved object from Twitter

Eikon

The news headlines were retrieved from the Refinitiv Eikon API. The platform provides various data on financial markets including text from Reuters and other various news sources. The data is collected with the “get_news_headline” function. Due to restrictions described in the API documentation[8], the number of text items were limited to 100 for each specified time window and ticker. Each time window represents a 24 hour collection period. In addition, there were limitations on the total extraction of data. In most cases this extracted all available data. Table 3 presents raw data from the retrieved object and an overview of the newspaper ID’s can be seen in Appendix A 3 and Appendix A 4.



Date	Text	URL	Source	Ticker
2021-08-01 14:00:12.126000+00:00	New patent shows iPhone 14 could come with Touch ID and Face ID under the screen	urn:newsml:reuters.com:.	NS:INDIAE	aapl
2021-08-01 13:03:42.961000+00:00	"Big tech's big week raises fears of 'Blade Runner future' of	urn:newsml:reuters.com:…	NS:THEGRD	aapl
2021-08-01 05:05:04.228000+00:00	Refinitiv Newscasts - TRADESTATION: Providing a financial edge for retail investors	urn:newsml:reuters.com:	NS:REALV	aapl
2021-08-01 01:54:03.638000+00:00	"Empanelment Of Oem Or Their Authorizedpartner	urn:newsml:reuters.com:…	NS:ECLTND	aapl
2021-08-01 01:50:09.881000+00:00	Apple	urn:newsml:reuters.com:…	NS:ECLCTA	aapl

Table 3: Retrieved object from Eikon

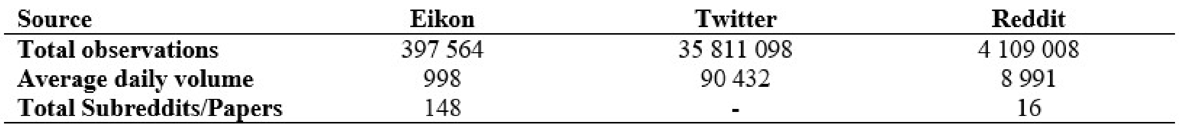
[8] https://developers.refinitiv.com/en/api-catalog/eikon/eikon-data-api/documentation

Price Data

In this thesis, the aim is to predict day-ahead volatility. This is measured as the intraday volatility from 15-minute tick data. AlphaVantage API[9] has been used to gather the historical intra-day data from 01.08.2021 to 31.08.2022. The choice of price data used to measure the volatility is an essential part of this thesis and is described in chapter 4.4 on volatility.

3.2 Overview of Data

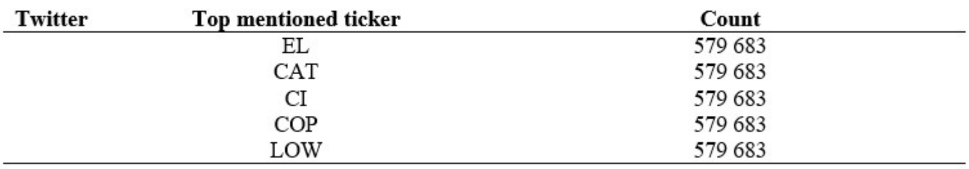
Table 4 summarizes the volume of raw text data before pre-processing. The total amount of data from Twitter is roughly one order of magnitude higher than Reddit and two orders of magnitude for Eikon. The daily limit implemented for the collection of tweets amounts to a total of 144 000 tweets[10]. With the daily average volume being 90 432 the utilization of allowed collection is 62.8 %.



Source	Eikon	Twitter	Reddit
Total observations	397 564	35 811 098	4 109 008
Average daily volume	998	90 432	8 991
Total Subreddits/Papers	148	-	16

Table 4: Raw data by source

Further insight can be found in Table 5. It is unexpected that popular stocks like Tesla or Apple are not among the top mentioned tickers from Twitter. These are two of the favorites amongst retail investors (Williams, 2022).



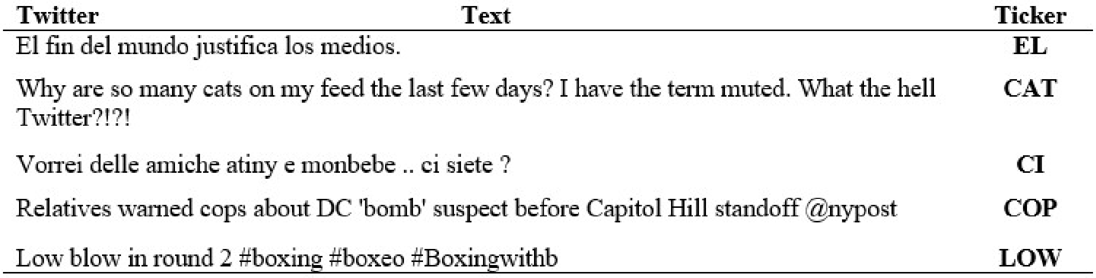
Twitter	Top mentioned ticker	Count
	EL	579 683
	CAT	579 683
	CI	579 683
	COP	579 683
	LOW	579 683

Table 5: Top mentioned tickers from Twitter

The maximum number of tweets per hour is collected for all of the top mentioned tickers. The total collection of tweets combined with the relative popularity of these companies are contradictory to the article on retail favorites. This needs further exploration. Table 6 previews text collected for the tickers in the previous table.

[9] Documentation for the API: https://www.alphavantage.co/documentation/#intraday-extended

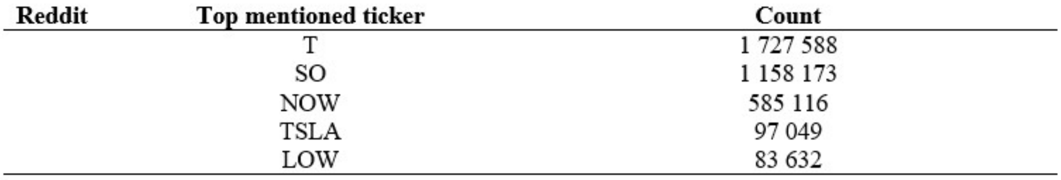
[10] $60\ tweets\ per\ hour * 24\ hours * 100\ stocks$



Twitter	Text	Ticker
	El fin del mundo justifica los medios.	**EL**
	Why are so many cats on my feed the last few days? I have the term muted. What the hell Twitter?!?!	**CAT**
	Vorrei delle amiche atiny e monbebe .. ci siete ?	**CI**
	Relatives warned cops about DC 'bomb' suspect before Capitol Hill standoff @nypost	**COP**
	Low blow in round 2 #boxing #boxeo #Boxingwithb	**LOW**

Table 6: Sample text from Twitter

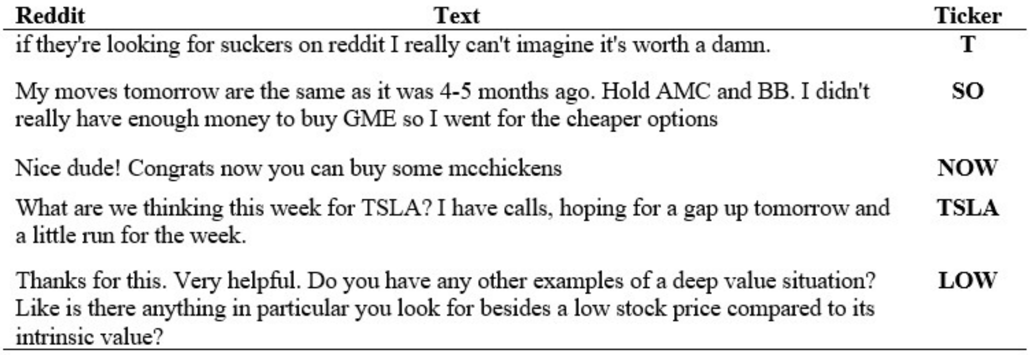
After an in-depth review of the data, it is revealed that the text collected for these tickers consists primarily of noise, and that the content has nothing to do with the company in question. This warrants further text filtering before any pre-processing methods can be applied to make the data ready for analysis. The problems in the Twitter data may also be a problem for the other sources and it needs to be investigated. Table 7 presents the results for the Reddit data.



Reddit	Top mentioned ticker	Count
	T	1 727 588
	SO	1 158 173
	NOW	585 116
	TSLA	97 049
	LOW	83 632

Table 7: Top mentioned tickers from Reddit

There is not a daily restriction on comments collected from Reddit. This results in different counts for each of the top mentioned tickers. The tickers themselves also differ from Twitter, except “LOW”. The Reddit data does not indicate any abnormalities at first glance. However, when looking through the collected comments, symptoms of noise can be observed.

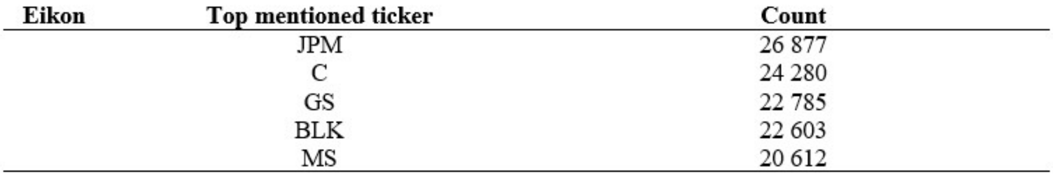


Reddit	Text	Ticker
	if they're looking for suckers on reddit I really can't imagine it's worth a damn.	**T**
	My moves tomorrow are the same as it was 4-5 months ago. Hold AMC and BB. I didn't really have enough money to buy GME so I went for the cheaper options	**SO**
	Nice dude! Congrats now you can buy some mcchickens	**NOW**
	What are we thinking this week for TSLA? I have calls, hoping for a gap up tomorrow and a little run for the week.	**TSLA**
	Thanks for this. Very helpful. Do you have any other examples of a deep value situation? Like is there anything in particular you look for besides a low stock price compared to its intrinsic value?	**LOW**

Table 8: Sample text from Reddit

Table 8 previews the Reddit data. None of the text is actual chatter about the company ticker, except for TSLA. This needs further examination, and a more in-depth review of the data is warranted. It is observed that other short ticker names suffer from the same kind of issues as the text shown in Table 8. Most likely due to the fact that the ticker symbols can be words or characters used in different settings other than chatter about the company.

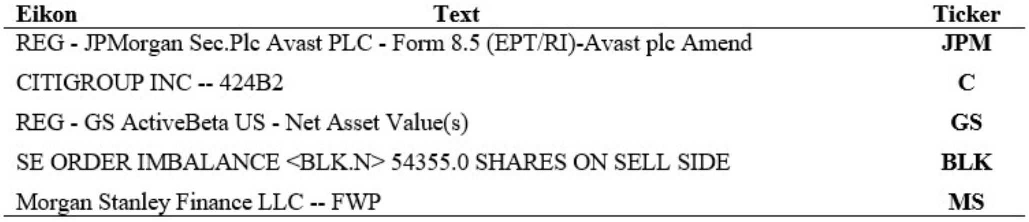
Lastly, a look at mentions from the Eikon data can be seen in Table 9. This data should be of higher quality than the previous two sources, as it is collected from a professional data vendor.



Eikon	Top mentioned ticker	Count
	JPM	26 877
	C	24 280
	GS	22 785
	BLK	22 603
	MS	20 612

Table 9: Top mentioned tickers from Eikon

The volume is lower, as expected. Again, it is strange that mediacentric companies such as Tesla, Apple, and Amazon, are not amongst the top mentioned tickers. Instead, the list is populated by bank and asset management firms.



Eikon	Text	Ticker
	REG - JPMorgan Sec.Plc Avast PLC - Form 8.5 (EPT/RI)-Avast plc Amend	**JPM**
	CITIGROUP INC -- 424B2	**C**
	REG - GS ActiveBeta US - Net Asset Value(s)	**GS**
	SE ORDER IMBALANCE <BLK.N> 54355.0 SHARES ON SELL SIDE	**BLK**
	Morgan Stanley Finance LLC -- FWP	**MS**

Table 10: Sample text from Eikon

The headlines consist primarily of various financial filings that would not contribute to the sentiment analysis of the companies. After a closer look at the data, headlines like the examples in Table 10 are not unique to the top mentioned tickers. Nearly all of the tickers used for the analysis contain some noise in the form of filings that should not impact sentiment. Therefore, the Eikon data also require further filtering.

Summary of Data Overview

This chapter has covered the exploration of text and presented representative text samples of noisy data. All of the data sources have indications of containing large amounts of noise. An in-depth review of the data revealed several specific issues that could impact the results. Conducting an in-depth review of the data ensures quality data, reduces unnecessary computational costs, and increases the validity of the results. The findings in this section should have a significant impact on the data quality. Thus, improving the sentiment classification. In the methodology chapter, techniques to reduce the possibility of noisy text affecting the sentiment analysis are applied.

4 Methodology

The following chapter covers methods used in this thesis. It includes data preparation, sentiment scoring with FinBERT, calculation of volatility, and it covers the models used to predict volatility. This chapter begins with the preparation of data followed by an introduction to the BERT and FinBERT models before volatility is explained. In the end follows a description of the deep learning models and the implementation.

4.1 Data Preparation

Pre-processing of the raw text gathered in the last section is essential for the validity of the sentiment scores. It can heavily influence the outcome of the analysis. The overall goal of pre-processing is to clean and prepare the text for further analysis. The necessary steps included in the pre-processing depend on the data source from which the text is obtained. The text from Twitter and Reddit requires different cleaning and filtering than news headlines from Eikon. Social media texts often contain hyperlinks, emojis and pictures, and this is not common in editorial newspaper headlines. While the Eikon data contains generic filings the company has made, such as 10-K's and 10-Q's, which should have zero effect on the sentiment of the company as the statements themselves are not analyzed.

The following section covers the various pre-processing and wrangling performed on the data to filter out perceived noise and otherwise increase the quality of the data.

Data Wrangling

The overview of the raw text demonstrated that all sources contained noisy data, which worsens the validity of the sentiment analysis. Thereby the prediction of volatility. To combat this, a general ruleset for filtering out unnecessary tweets, comments, and headlines, has been applied. In addition, the dates had to be normalized by adjusting for time zones and trading hours.

Fixing Datetime

To ensure that the various datetimes from Reddit, Twitter and Eikon are on the same format, the dates have been normalized to UTC and then stripped of the time zone value from the date object. The reason for this is to ensure that text is categorized to the correct day before the sentiment is calculated. The price data did not need any correction as the calculation of the daily volatility is stored as a date stamp without hours and minutes.

The text data has also been adjusted to the trading hours. Like text collected during weekends, holidays or after the markets had closed. The reason for this is that sentiment in text provided outside of trading hours could have an impact on the next days' volatility as both positive and negative news regarding companies are announced at times where markets are unable to react. While some traders do have access to pre-market and after-hours trading, most do not. To adjust for this, the text collected during weekends, US holidays or after close, are all shifted to the next available trading day. Manipulating the dates according to this assumption is performed by implementing a datetime checking algorithm. The algorithm checks whether the time is past close, 9 pm, if so, it adds one hour until the next day. If the current or next day is a weekend or holiday, it pushes the date to the next available trading day.

Filtering Noise

The overview of the text data highlighted large amounts of noise, especially from the social media sources. Since the various sources provide a different set of challenges, the filtration for each source is explained. Overall, the filtration results in higher quality data and fewer observations. The result of the filtration is summarized in the next section.

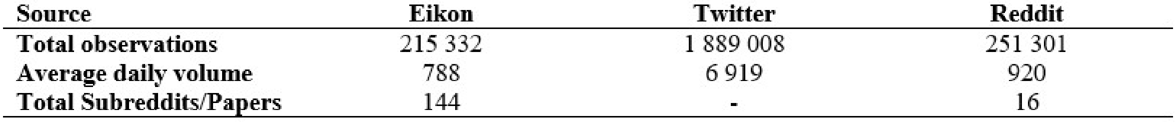
First, the volume of tweets collected from Twitter is not feasible to process in this thesis due to the computational costs of analyzing 36 million rows of text. To reduce the number of observations, the tweets are matched to "$TICKER" instead of "TICKER". This has the added benefit of removing some of the text related to troublesome tickers. Next, the tweets are checked for non-English language through an NLP language detector. The language detector used for this task is from the package spacey in Python. The language detector is pre-trained and utilizes neural network models for tagging and text classification (Eden Ai, 2022). The length of the tweets were checked and if they contained less than 3 words they were dropped. In addition, duplicate texts have been dropped. To filter out spam from bot accounts, a maximum limit on the number of tickers in one tweet were applied. This assumes that if a tweet contains more than five tickers, it is most likely spam, with the intention of appearing in a high number of feeds or searches.

The filtrations needed for the Reddit data were not as extensive as with the Twitter data. This is because each subreddit is maintained by moderators that curate the content for human interaction and prevents spam from bots. The overall volume is lower, but the comments are heavily impacted by the same troublesome tickers as the Twitter data. Therefore, a less strict matching was introduced. If the ticker length is less than 3 characters, it requires a “$” in front. The list was extended to include these tickers: “LOW”, “COST”, “NOW”, “CAT”, and “UPS”. Since these can be used as words in sentences with no connection to the company ticker. Duplicates were removed too, as with the Twitter data.

The data from Eikon is collected through a professional API by a reputable data provider. That ensures higher quality data and a match on the correct company. However, this source also contained noisy headlines such as certain types of journal entries, generic financial filings, and non-relevant market reports. To filter out these types of headlines, a comprehensive regex pattern was deployed. It was made to match the aforementioned type of headlines. Duplicates are also removed in the same manner as per the previous sources.

Overview of Filtered Data

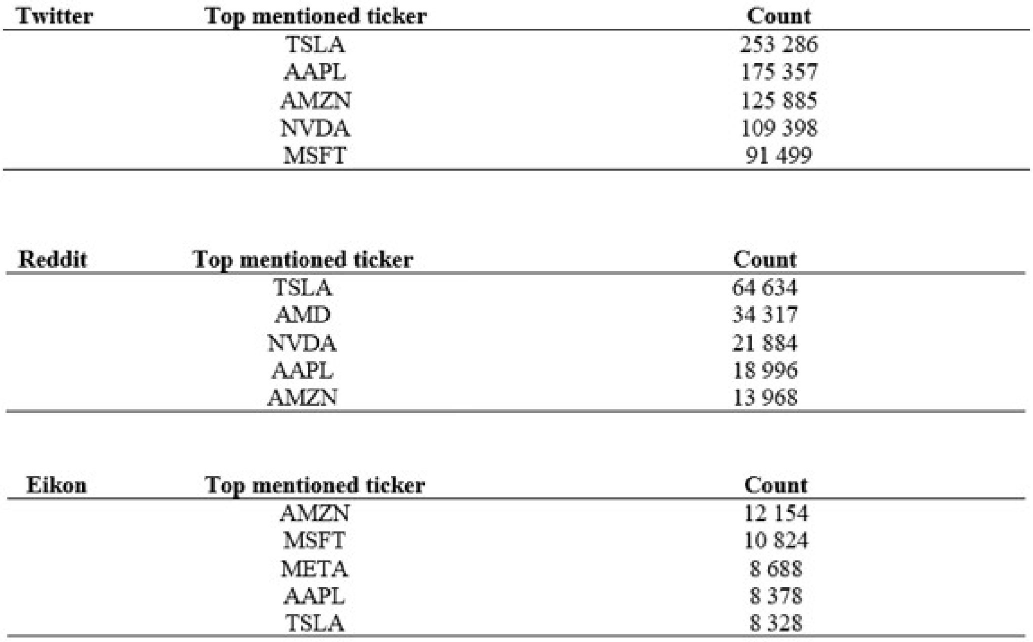
Table 11 presents the results for the cleaned data. This section reviews the results of the filtering techniques applied to the data.



Source	Eikon	Twitter	Reddit
Total observations	215 332	1 889 008	251 301
Average daily volume	788	6 919	920
Total Subreddits/Papers	144	-	16

Table 11: Cleaned data by source

The filtrations have been quite successful. The impact on the Twitter data was substantial. It filtered out a total of 94.7 % of observations. The filtration of Reddit and Eikon data amounted to respectively a 93.9 % and 45.8 % reduction in observations. An added benefit is that the computational costs were greatly reduced, in addition to an increase in quality.



Twitter	Top mentioned ticker	Count
	TSLA	253 286
	AAPL	175 357
	AMZN	125 885
	NVDA	109 398
	MSFT	91 499

Reddit	Top mentioned ticker	Count
	TSLA	64 634
	AMD	34 317
	NVDA	21 884
	AAPL	18 996
	AMZN	13 968

Eikon	Top mentioned ticker	Count
	AMZN	12 154
	MSFT	10 824
	META	8 688
	AAPL	8 378
	TSLA	8 328

Table 12: Top mentioned tickers by all sources for cleaned data

The top mentioned tickers after the filtration shown in Table 12 are probably a better representation of the most frequently mentioned tickers. In the context of Williams (2022) these companies are more likely to attract Internet chatter or headlines. There is now more overlap between the different sources than before the filtering.

The dataset has now been successfully reduced. Insights into text collection for the cleaned data is presented in Figure 1. It visualizes the top 20 tickers by mentions, for all sources of

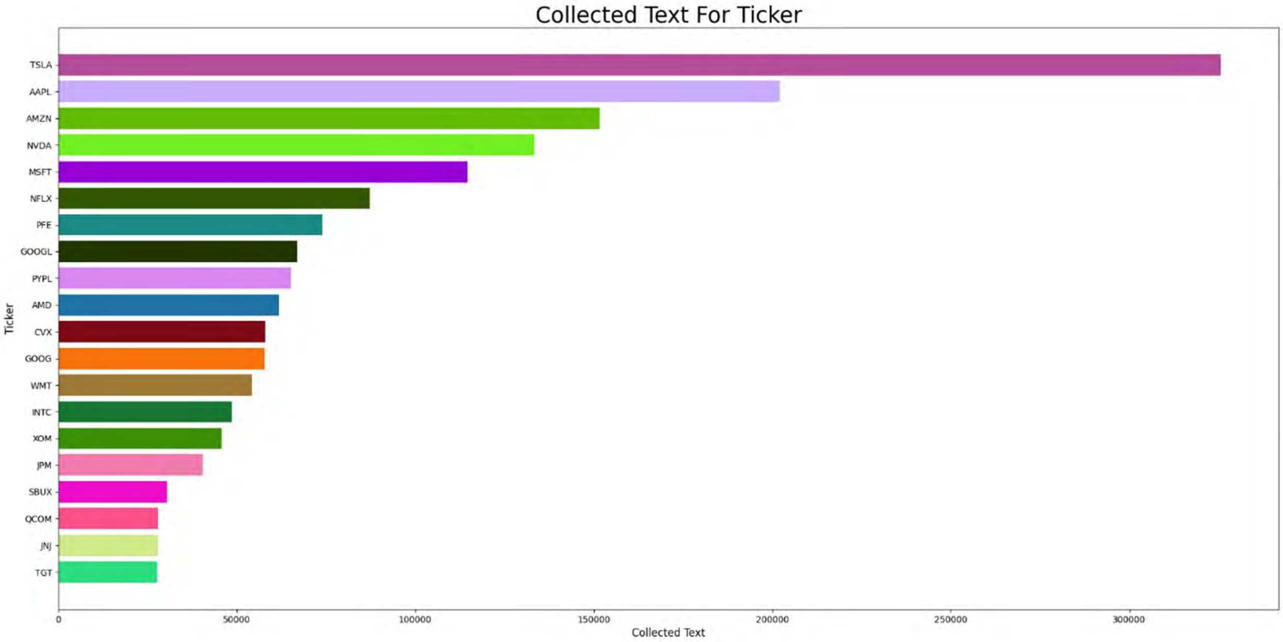


Figure 1: Collected text for top 20 mentioned tickers in the dataset

text. Furthermore, the collection of text from the sources across the time period is displayed in Figure 2. The secondary y-axis for Twitter is necessary as this source collects far more text than the others. The timeseries for collected tweets illustrates the effect of manipulating the datetime for the text. The pattern is most likely the result of tweets collected after close on Fridays and weekends being pushed to Mondays.

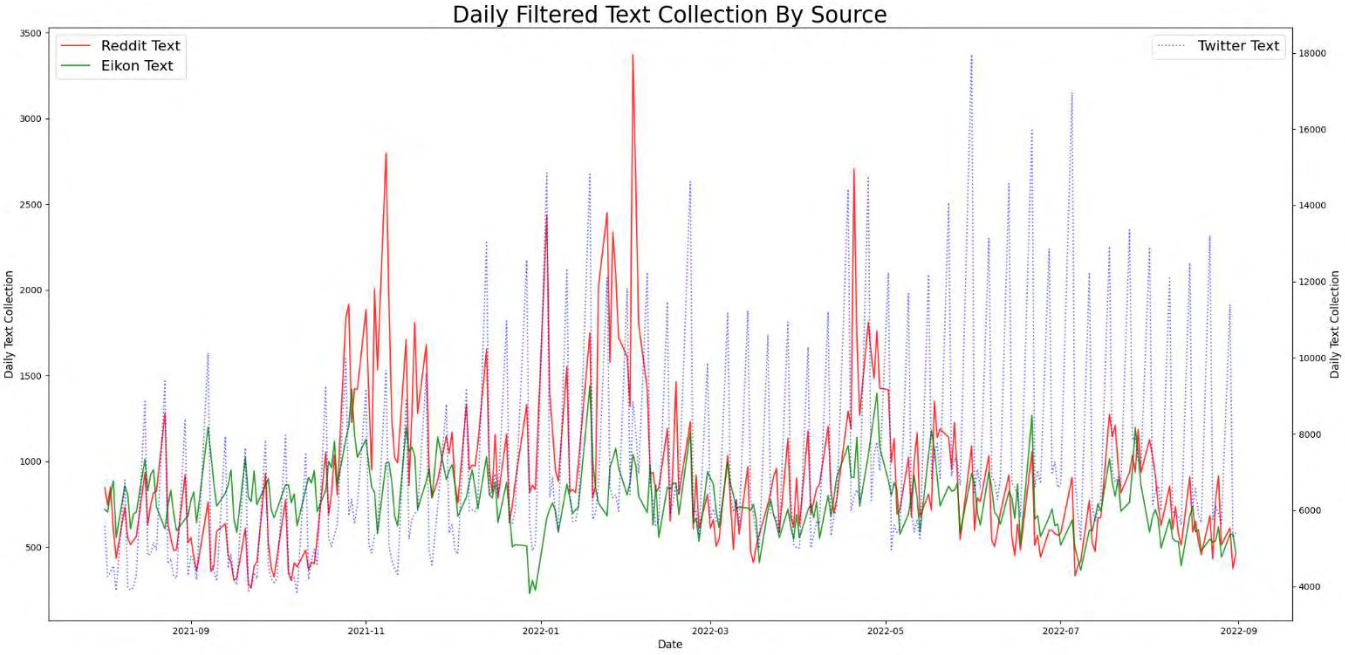


Figure 2: Daily collection of filtered text by source

Text Pre-processing – Method and Implementation

Text preprocessing is a fundamental part of NLP. It is the process of turning text into machine readable data. To get a processed corpus, techniques such as tokenization and lemmatization are performed. Tokenization is the process of separating a corpus into smaller units called tokens. The process can range from simple approaches as splitting and whitespace and removing punctuation to more complex standardized methods (Manning, et al., 2009). Different languages often require different rules as of how to tokenize the text. Stemming is another text pre-processing technique that tries to reduce a word to the stem, the base part of the word. It is done by removing the derivational affixes, by a not always precise heuristic. Lemmatization is a more refined option of pre-processing text than stemming. The aim is to remove inflectional endings and to return the dictionary form of a word, the lemma. Both lemmatization and stemming are performed in order to generalize across similar terms. This facilitates learning in the models (Eisenstein, 2019).

In order to prepare the data for sentiment analysis, the data needs to be preprocessed. This process depends on the selected model. The following paragraphs explains the pre-processing of the filtered data which differs from the description above due to the self-attention mechanism in the FinBERT model. The theory behind the FinBERT model is explained in the next section.

The classic NLP approach of cleaning the text for punctuation, digits and emojis is not needed for FinBERT. The model is capable of classifying texts where these objects occur. It is argued that such characters should not be removed as they can provide meaningful context (Bricken, 2021). Stemming or lemmatizing the tokens after tokenization is also not necessary since the model is capable of understanding context and performing it as part of the pre-processing will reduce the quality of the input. This differs from traditional machine learning methods where these processes often are utilized. However, html-links are removed. The overall process thereby differs as to how the text would have been handled in a machine learning approach where such steps are crucial in order to improve the quality of the input.

Furthermore, the text is tokenized by the AutoTokenizer from the Transformers library. The Hugging Face-API[11] provides a quick implementation of the sentiment model and eliminates the need to store the tokens locally. The pre-trained tokenizer is specified as *ProsusAI/finbert*

[11] Hugging Face is a platform used to build, train and deploy DL/ML models - https://huggingface.co/

when preparing the pipeline for sentiment analysis. The pre-trained tokenizer handles the process of splitting the text and making it machine readable through embedded encodings. There are two choices of encodings, encode and encode+. Both provides output in the form of token ID tensors, but the latter provides more information (Briggs, 2021). The sentiment scoring in this thesis is based on the encode+ method as it provides attention mask tensors. In short, attention mask tensors provide information to the sentiment model in the form of batching input sequences together and indicating if the token should be attended to by the model or not. The outputs are 0s and 1s where 0 signalizes the token to be ignored and 1s for the tokens to be important and utilized for further processing (Sharma N. , 2022).

4.2 Sentiment Models

As mentioned previously, the sentiment analysis in this thesis is performed by a BERT model, specifically a BERT model that is pre-trained on financial text, FinBERT. The following section introduces the theory behind the BERT and FinBERT models. In addition, it includes implementation of the model and an analysis of the sentiment output.

BERT – Bidirectional Encoder Representations from Transformers

In 2018 the BERT model was introduced by Google, and it achieved state of the art results (Devlin, et al., 2018). In the paper, two BERT models were presented. A base model and a large model. The base model was used to compare the performance to the previous state of the art model, the OpenAI GPT model presented by Radford et al. (2018). The large model achieved state of the art results. Today there are models that are trained on even larger amounts of data. The training data used to pre-train BERT is a combination of the BooksCorpus and English Wikipedia. The motivation behind the invention of BERT was to improve performance, compared to previous models like ELMo[12] and GPT. Both of those models had approaches that were based on unidirectional language models to learn language representations (Devlin, et al., 2018). BERT is a bidirectional model. This allows the model to learn left and right context. This was applied by using a masked language model in the pre-training phase. Input in the BERT model is based on the WordPiece embeddings, with a vocabulary size of 30 000 (Wu, et al., 2016). At the beginning of every BERT sequence is a special classification token, CLS, it represents the aggregated sequence and can be used for classification tasks such as sentiment scoring.

[12] Embeddings from Language Model - ELMo

The BERT model has a multi-layer bidirectional transformer encoder architecture. The large BERT model is made up of 12 layers, also referred to as transformer blocks or encoder layers, the hidden size is 1024 and the number of self-attention heads is 16. Each encoder consists of a multi-head self-attention layer and a feed forward neural network[13]. In addition, there is one "add and norm" layer after each sublayer. The "add and norm" layer is described in Vaswani et al. (2017). This presents a few important concepts needed to understand the BERT model, the transformer, and self-attention. They are prerequisites to understand the BERT model and are explained briefly.

The transformer, as introduced in Vaswani et al. (2017), consist of two parts. An encoder and a decoder. Or of stacks of encoders and decoders, known as blocks. Both the encoders and decoders have attention layers. Attention is a function mapping a query and a set of key-value pairs to an output, where the output is a weighted sum of the values (Vaswani, et al., 2017). It should be noted that BERT only uses the encoder stack. Conceptually, the encoder processes the input and compiles information to a vector, called context. The context vector that is passed to the decoder is the last hidden state of the encoder. The encoders are often layered. The decoder produces output based on the received context. For further information on the transformer, see *"Attention Is All You Need"* by Vaswani et al. (2017).

Self-attention is the other essential building block in the transformer-based BERT model. It enables the model to understand the interactions between input values, and this differentiates self-attention from attention (Alammar, 2018). This understanding stems from a set of vector and matrix multiplications and SoftMax transformations.

In BERT, the self-attention is refined to a multi-headed self-attention (Alammar, 2018). It provides more representation subspaces to the attention layer, and it improves the model's ability to focus on different parts of the input. Capturing the fact that different words relates to each other by different relations. Multi-headed self-attention layer is described in Alammar (2018). For each encoder, the output from the multi-head self-attention layer is passed to a feed forward neural network. FFNN models will be described in chapter 4.6. The output from the encoder stack is sent as input to a new classifier that provides the final output.

[13] Feed forward neural network - FFNN

FinBERT

This thesis uses textual data related to finance. The regular BERT model would therefore have had the same problem as the non-domain-specific dictionaries, it does not understand that words in a financial context could vastly change the sentiment. It needed to "talk like a trader" to quote Dogu Araci (Araci & Genc, 2020). The description of FinBERT is based on the paper introducing the model, *"FinBERT: Financial Sentiment Analysis with Pre-trained Language Models"* (Araci, 2019).

The new addition to the FinBERT model, compared to the regular BERT, was more pre-training. This domain adaptation was performed by using the Reuters TRC2 corpus (Araci & Genc, 2020). To fine-tune the model for the sentiment analysis task, they used a dataset from the Financial Phrasebank. It contains 4500 sentences from news articles related to finance, each labeled by field experts. The FinBERT model achieved an accuracy of 97% on the test set. Beating the comparable models used in the study. Those models included an LSTM, an LSTM with ELMo and a ULMFit[14] model.

Sentiment Scoring with FinBERT

The last paragraphs introduced the FinBERT model. Since it is a pre-trained model, it requires little to no modification to perform the sentiment analysis. The advantage of a pre-trained model is that it can be deployed quickly. However, it might perform worse than a model specifically designed for this task.

The model was implemented utilizing the Hugging Face API to import the pre-trained model. The model looped through each row in the dataset of collected texts. Then it outputs the probability scores for each of the three categories: Negative, neutral, and positive. After the model was finished, the sentiment scores were used to create the sentiment predictors.

Explorative Data Analysis of the Output

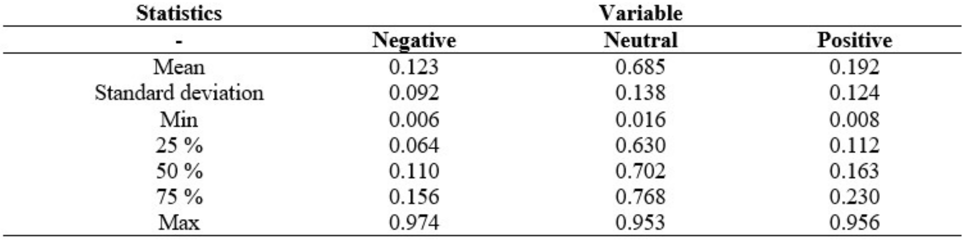
Conducting explorative analysis of the data is essential when creating predictive models. Exploratory data analysis (EDA) is the process of exploring data, generating insights, testing hypotheses, checking assumptions, and revealing underlying hidden patterns in the data (Sharma A. , 2022). This thesis utilizes predictors based on the sentiment score provided by the FinBERT model as input for the models that predict and classify market volatility. One

[14] Universal Language Model Fine-tuning - ULMFit

assumption is that the pre-trained model, FinBERT, provides a viable score with predictive capabilities. The basis for the sentiment scores is thereby the text given as input to the model. To ensure the validity of the data, the sentiment scores are analyzed.

Summary Statistics for the Sentiment Scores

Table 13 presents descriptive statistics for the three different sentiment classes generated by FinBERT. The numbers represent probabilities.



Statistics		Variable	
-	Negative	Neutral	Positive
Mean	0.123	0.685	0.192
Standard deviation	0.092	0.138	0.124
Min	0.006	0.016	0.008
25 %	0.064	0.630	0.112
50 %	0.110	0.702	0.163
75 %	0.156	0.768	0.230
Max	0.974	0.953	0.956

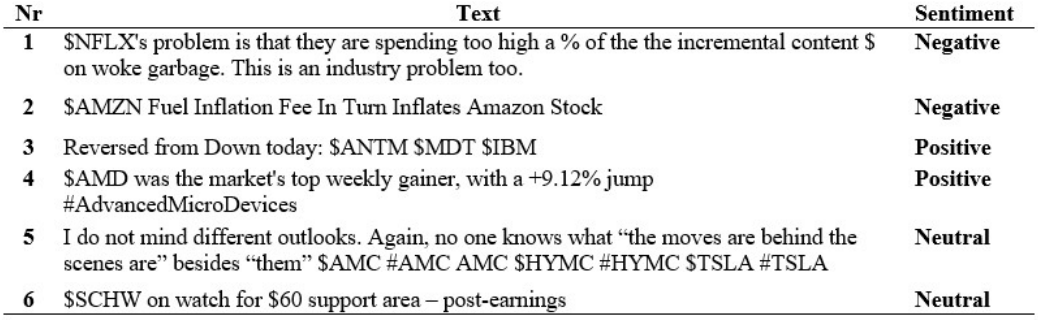
Table 13: Descriptive statistics for sentiment probabilities

This provides useful insights in the dataset, for instance, the average observation has a 68.5 % probability of being neutral. Positive classifications have a high standard deviation compared to the mean, representing large fluctuations for the probability of a positive sentiment. The median observation at 50 % percentile reveals that half of the observations are above and half are below the respective value. The 75 % percentile reveals that only 25 % of observations are above this value while the rest fall below it. Lastly, the maximum observed value for the scores is relatively close to 100 % probability of being the respective sentiment. Indicating that the model is capable of classifying with high certainty whether the text has negative, neutral or positive sentiment.

Keep in mind that FinBERT assigns probabilities for each of the three sentiment categories. Therefore, the strictly classified sentiment of the observations is provided by the sentiment category with the highest probability score. This generates data with strict classifications that can be used for analysis and visualization.

Negative, Neutral and Positive Text

Table 14 presents a sample of observations that illustrates the sentiment classification. Observation one is overwhelmingly negative about Netflix and their spending. Observation two is not clear cut and probably a misclassification. A “fuel inflation fee” is likely negative, however, “inflates” is probably a positive word in this context. Given that the Amazon stock price was inflated i.e., went higher. The neutral classifications both seem neutral in regard to language. However, observation six should probably be interpreted as negative as it relates to a support area. A common terminology among traders and it refers to an area where there are traders willing to buy. The two positive classifications are both correct. The positive sentiment is implied respectively by a reversal of a downward movement and a “top weekly gainer”.



Nr	Text	Sentiment
1	$NFLX's problem is that they are spending too high a % of the the incremental content $ on woke garbage. This is an industry problem too.	**Negative**
2	$AMZN Fuel Inflation Fee In Turn Inflates Amazon Stock	**Negative**
3	Reversed from Down today: $ANTM $MDT $IBM	**Positive**
4	$AMD was the market's top weekly gainer, with a +9.12% jump #AdvancedMicroDevices	**Positive**
5	I do not mind different outlooks. Again, no one knows what “the moves are behind the scenes are” besides “them” $AMC #AMC AMC $HYMC #HYMC $TSLA #TSLA	**Neutral**
6	$SCHW on watch for $60 support area – post-earnings	**Neutral**

Table 14: Sample of strictly classified texts

Overall the model performance is fine. However, some of the misclassifications might be casued by the fact that FinBERT is trained on corporate financial language. As the FinBERT pre-training consisted of a number of curated articles and company produced reports and not on financial, social media text. This can be observed by looking at larger volum of text and their respective classifications.

The Distribution of Sentiment

This section covers the distribution of sentiment scores. Observed in Figure 3 is a plot of the distribution of all sentiment scores. Most observations are classified with a high probability of being either neutral, negative, or positive, whereas neutral classification has the most observations. Inferred from this is the fact that the majority of observations above 0.8 probability are classified as neutral. The duality of the plot is a result of FinBERT assigning probabilities for each class of sentiment. Meaning that if an observation is classified as neutral, it also contains additional probabilities for positive or negative sentiment.

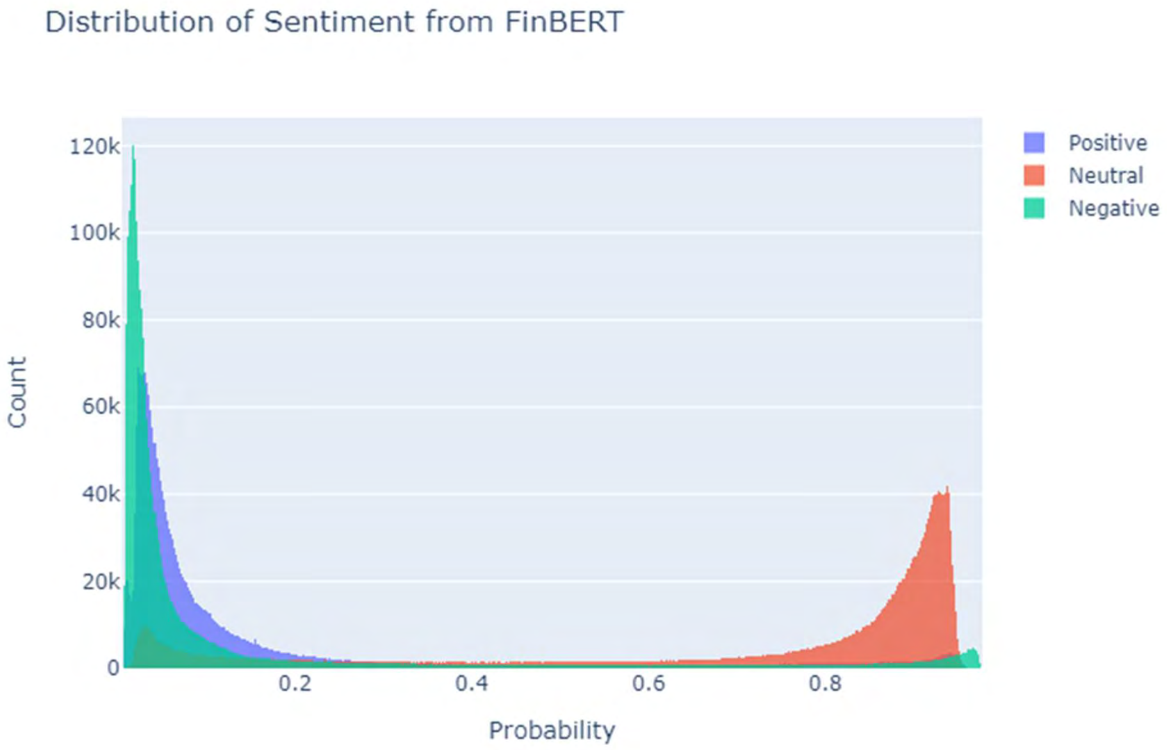


Figure 3: Distribution of Sentiment probability from FinBERT

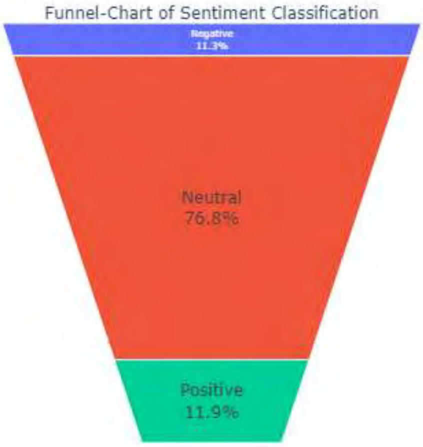


Figure 4: Funnel-Chart of sentiment classification for the dataset

The strict distribution of classifications is more clearly observed in Figure 4. 76.8 % of the observations have the highest probability of conveying a neutral sentiment, while positive or negative sentiment has about the same observations at 11.9% and 11.3% respectively. The difficulties with text from social media are best observed when breaking down the funnel chart by source. Figure 5 presents the individual funnel charts for the data sources.

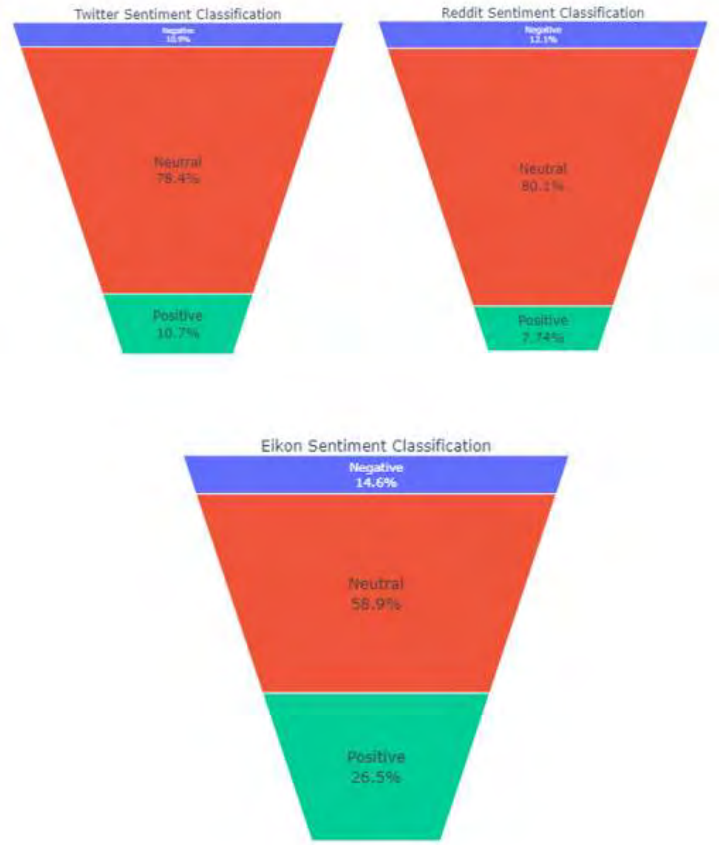


Figure 5: Funnel-Chart of sentiment classification by source

FinBERT interprets and classifies sentiment for both of the social media sources in roughly the same manner. Nearly 80% of observations for both sources end up being classified as neutral. Note that the argument of less relevant training data is speculative and comes from reviewing the data to validate the classifications. The problem might exist due to other causes. Another potential reason might be that social media data has an inherently different distribution of sentiment in the data.

Summary of the Explorative Data Analysis

The EDA reveals that the sentiment model might have a few issues handling text from Twitter and Reddit. However, it does an adequate job of classifying texts that with a clear and obvious polarity. It struggles to interpret social media texts with ambiguous language, and it would require further understanding of typical social media speech connected to financial markets. Since the sentiment scores impact the explanatory variables used in the predictive analysis this may have a negative effect on the predictive capabilities of the deep learning models.

4.3 Representation of Sentiment

There are many ways to represent sentiment. The representation of sentiment that best captures the real, underlying sentiment would be the favored representation. This section introduces the two types of representation of sentiment used in this thesis. It is based on the output from the FinBERT model and number of articles. Variable descriptions can be found in Appendix A 5.

Sentiment Average

The sentiment average variables consist of *"Negative"*, *"Neutral"*, and *"Positive"*. Each represents the average sentiment score from the FinBERT model. The average is calculated daily for each ticker. This results in three columns of data with the same length. Thereby avoiding the problem of different lengths on the input to the model. The averages are not weighed by text source. All observations are treated equally.

Sentiment Count

The sentiment count predictor is based on a combination of sentiment scores and the number of texts. The distribution of positive, negative, and neutral sentiment could provide predictive capabilities for volatility as it is established in literature that negative text is linked with higher volatility. The sentiment count variable is calculated by summarizing all of the observations for each category, for which the sentiment probability is the highest. It is aggregated by day and ticker. This yields three new variables called *"Sentiment_pos"*, *"Sentiment_neg"* and *"Sentiment_neu"*. The count variables are all normalized to values between 0 and 1 with sklearn min-max scaler in Python.

Total Ticker Count

In the literature review, the paper by Antweiler & Frank (2002) was introduced. They looked at the relation between number of messages on financial message boards and volatility and found that they corresponded. This is represented in a similar way in this thesis through the predictor called *"Total_Count"*. It summarizes the total number of times the ticker has been mentioned that day across all of the different sources. Figure 6 shows a similar correspondence between volatility and text as the Antweiler & Frank study. In addition, the daily stock mentions are comparable to the noise level studied in *"Is Sound Just Noise"* by Coval & Shumway (2001).

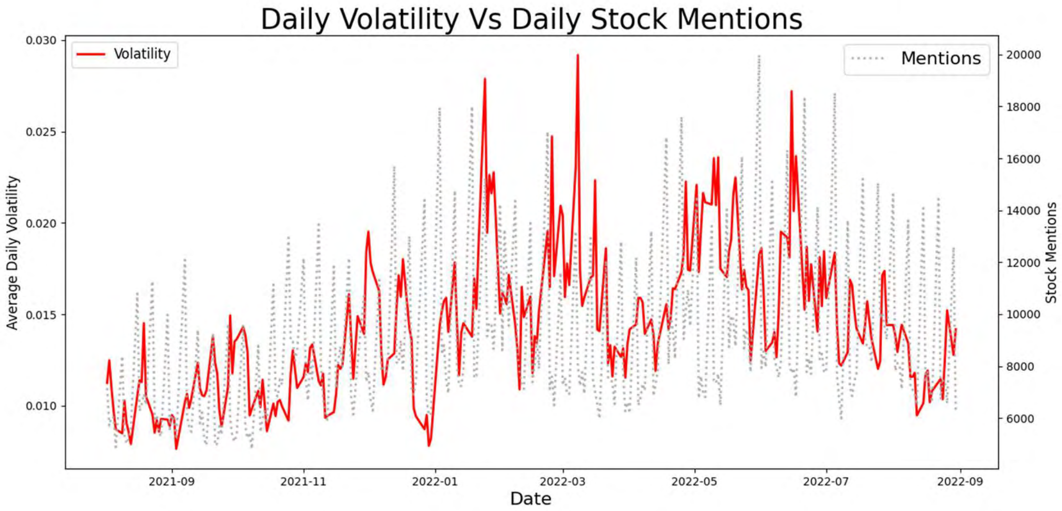


Figure 6: Plot of daily stock mentions and daily average volatility

4.4 Volatility

In order to predict volatility in financial markets, it is important to understand how and why it is calculated. Forecasting volatility is an important task in financial markets. Volatility is not equivalent to risk (Poon & Granger, 2003). It should be interpreted as uncertainty. Volatility is used by investors and portfolio managers to manage risk, price assets, and much more. Since the goal of this thesis is to better predict volatility, a proper introduction to the subject is needed. There are two main types of volatility, historical volatility, and implied volatility. The first is commonly used in value at risk (VAR) models and the second in options pricing. This thesis aims to predict historical volatility.

Volatility – Method and Implementation

Historical volatility, hereby interchangeably referred to as volatility, is often calculated as the sample standard deviation of returns. Although, for a long time daily squared returns have been used as a proxy for volatility. This has changed with the introduction of high frequency data and intraday prices (Poon & Granger, 2003). Daily volatility can now be derived from intraday returns. Equation 1 is the formula for realized volatility and it is how volatility have been calculated in this thesis. The returns used to calculate volatility are important. When calculating actual historical volatility, it is assumed that it is the natural logarithm of stock returns that follows a normal distribution (Hull, 2017). Hence the use of logarithmic returns as formulated in Equation 2.

$$Logarithmic\ Returns = r_t = \frac{\log{(P_t)}}{\log{(P_{t-1})}}$$

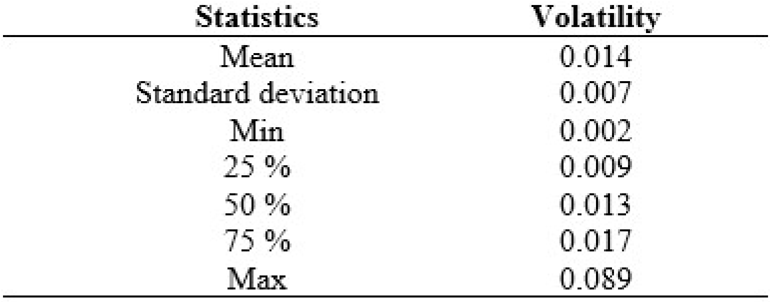
Equation 1: Formula for logarithmic returns

$$Realized\ Volatility = \sigma = \sqrt{\sum_{i=1}^{T} r_t^2}$$

Equation 2: Formula for realized volatility

Instead of calculating the representation of volatility from daily closing prices, volatility is calculated from the collected intraday price data. The calculation of volatility in this thesis is performed in Python with the use of the Numpy and Pandas libraries. The calculated volatility will be treated as an actual representation of volatility that day for each security. The values are then used both as predictors and response variables in the prediction models.

When implementing the realized volatility, the goal is an output that represents the volatility of intraday returns for each stock. This is achieved by first creating a function that calculates the realized volatility of the log returns. Secondly the data is transformed to the correct shape by grouping the data by both ticker and day. The volatility function is then applied to the data by an aggregation function. Descriptive statistics for volatility can be seen in Table 15.



Statistics	Volatility
Mean	0.014
Standard deviation	0.007
Min	0.002
25 %	0.009
50 %	0.013
75 %	0.017
Max	0.089

Table 15: Descriptive statistics for volatility

4.5 Inference Models

This section explains the prediction and inference models used in this thesis. Predictive and inference analysis are two different analysis frameworks and have distinctive differences in their objectives. The key reason behind the models is to estimate *"f"*, a function representing the link between predictors and response variables (James et al., 2021). Inference is the task of understanding how the dependent variable is associated with independent variables in order to describe the relationship. For instance, whether there is a linear relationship or not. On the other hand, prediction aims to minimize the reducible error. In prediction there will always be some irreducible error.

Linear Regression

Linear regression is one of the most widely applied methods in statistical analysis. This is due to ease of implementation and interpretation. It is commonly used in inference analysis with the goal of evaluating if there exists a linear relationship between dependent and independent variables.

The relationship between the dependent variable and the independent variable is formulated mathematically in Equation 3. It consists of an intercept, a slope, and the parameter value. The model's objective is to find the values of the intercept and slope that best fits the data. This is performed by minimizing the least squares criterion (James et al., 2021).

$$Linear\ Regression = Y = \beta_0 + \beta_1 X$$

Equation 3: Formula for linear regression

In order to perform a linear regression, 80% of the observations were sampled randomly. These observations were then used to fit the regression models. Due to high correlations between several of the independent variables, separate regressions for each of the independent variables were performed. This avoids the problem of multicollinearity. The dependent variable is *"Volatility"* for all predictors. The predictions from the linear regression are used as a benchmark to the other more flexible methods.

Logistic Regression

Where a linear regression aims to predict a quantitative response variable, the logistic regression predicts a qualitative response variable (James et al., 2021). Thus, logistic regression is a method for classification.

A logistic regression models the probability that the response variable belongs to a category. The multiple logistic function is formulated mathematically in Equation 4. The function is used to get outputs ranging from zero to one. Maximum likelihood is then used to fit the model (James et al., 2021). The interpretation differs from the linear regression. An increase in X changes the log odds.

$$Multiple\ Logistic\ Regression = Y = \ p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X+\cdots+\beta_p X_p}}$$

Equation 4: Formula for multiple logistic regression

As with the linear regression, the training data is a random sample of 80% percent of the observations. These observations are then used to fit the regression in R. The aim of this logistic model is to be a baseline comparison to the other models. Therefore, it is fitted using all of the variables.

4.6 Prediction Models: Neural Networks

Neural networks are the cornerstone of deep learning and have since 2010 had a great impact on many niche problems. Such problems range from image classification to time series forecasting. The models in this section aim to forecast the level of volatility and classify the movement of the next day's volatility as up or down. The models are developed with Keras and TensorFlow.

Training, Hyperparameters and Measurements

The following section provides a brief introduction to some of the key building blocks in neural network models. This includes data partitioning, optimizers, and activation functions.

In order to avoid overfitting, the data is partitioned in training, validation, and test data. The validation data ensures that the test data is entirely unseen by the model, and it has an important role in finding the best possible model. The DL models in this thesis are developed using early stopping and saving the best model, with patience set to 10 and number of epochs to 60. Subsequently, the best model is not likely to overfit the data. The model architecture and activation functions are found with a combination of a hyperparameter tuner provided by Keras and rigorous testing.

The optimizers, loss function and metrics differ from the DL regression models to the DL classification models. The *Adam optimizer* (Kingma & Ba, 2014) have been applied for the regression models and *rmsprop* for the classification models (Hinton, 2014).The loss functions are respectively mean squared error (MSE) and binary cross entropy. The evaluation metrics are set to mean squared error (MSE), mean absolute error (MAE), and accuracy.

Activation functions are also an important part of neural nets. An activation function refers to a function, usually non-linear and specified in advance, that transforms the input to the function. Activation functions are key part of neural nets. "ReLU", "Tanh" and "Sigmoid" are common and have different properties. These activations are presented below in Equation 5, Equation 6, and Equation 7. An activation function is applied to each unit in the network. One of the benefits of ReLU is that it avoids the vanishing gradient problem (Agarap, 2018). This is because the gradient of the ReLU is one if the output is larger than zero and zero otherwise. The vanishing gradient problem will be outlined when describing the RNN model. Below follows the activation functions used in this thesis.

$$ReLU\ Activation: g(z) = R(z) = max(0, z)$$

Equation 5: Formula for ReLU activation

$$Sigmoid\ Activation: g(z) = \sigma(z) = \frac{1}{1 +\ e^{-z}}$$

Equation 6: Formula for Sigmoid activation

$$Tanh\ Activation: g(z) = f(z) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

Equation 7: Formula for Tanh activation

FFNN – Feed Forward Neural Network

One of the simplest neural net models is the Feed Forward Neural Network, FFNN. Created to resemble how the brain works (James et al., 2021). They are highly flexible and have had great results in classification tasks.

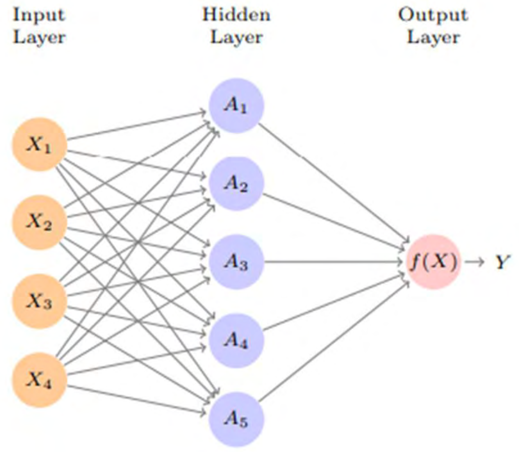


Figure 7: Illustration of Feed Forward Neural Network. Source: (James et al., 2021)

The structure of a FFNN consists of an input vector, one or more hidden layers, and an output layer (James et al., 2021). This can be seen in Figure 7. The hidden layers produce a function to predict the response variable, as seen in Equation 8. Each hidden layer is made up of hidden units. For the first hidden layer, each hidden unit or activation in that layer is a combination of the input vector and an activation function. This is presented in Equation 10. Later layers take the prior activations as input and compute new activations, as seen in Equation 11. The activations from the last hidden layer are then fed forward to the output layer. The output layer can consist of one or more units, and it outputs the prediction of the response variable, seen in Equation 9. The output from the output layer is determined by what the outputs represent. Ranging from a linear representation of the variables, to fitting a SoftMax function in order to transform the outputs to probabilities, or possibly a sigmoid activation, commonly used for binary classification. A more detailed explanation of a FFNN can be found in *"An Introduction to Statistical Learning - with Applications in R"* by James et al. (2021). To find the optimal values for the units in the neural net, the values get updated by a combination of gradient decent and backpropagation (Graves, 2008).

$$Input\ Vector: X = (\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_\mathrm{p})$$

Equation 8: FFNN Input vector

$$Output: f(X) = \beta_0 + \sum_{l=1}^{K2} \beta_l A_l^{(2)}$$

Equation 9: FFNN formula for output

$$Activations\ in\ the\ First\ layer: A_k^{(1)} = h_k^{(1)}(X) = g(w_{k0}^{(1)} + \sum_{j=1}^{p} w_{kj}^{(1)} X_j)$$

Equation 10: FFNN Activation function first layer

$$Activations\ in\ the\ Second\ layer: A_l^{(2)} = h_l^{(2)}(X) = g(w_{l0}^{(2)} + \sum_{k=1}^{K_1} w_{lk}^{(2)} A_k^{(1)})$$

Equation 11: FFNN Activation function second layer

TensorFlow does not handle the input shape of a pandas dataframe. Because of this, the values are converted to a Numpy array, resulting in a three-dimensional tensor. The FFNN models use either all predictors or only prior volatility, and the time window length is 1 or 5 trading days. The first hidden layer is a dense layer, with 128 units and ReLU as the activation function. It is followed by a dropout layer with a rate of 0.2. The dropout layer aims to avoid overfitting by randomly setting input units to zero while scaling up the units larger than zero (Keras, 2022). The dropout layer is followed by a new dense layer with 64 units and ReLU activation. The output layer for the regression models is a dense layer with one unit and linear activation. For the classification problem, the last hidden layer is flattened before it is sent to the output layer. There the output layer has one unit and a sigmoid activation function. The sigmoid activation function is applied to create a binary outcome.

RNN – Recurrent Neural Networks

RNNs are designed to handle sequential data like time series. In standard neural networks, there is an assumption of independence among the training and test examples (Lipton & Berkowitz, 2015). This leads to a model that loses the state of the network after each new input. This is fine for independent data. However, time series are most likely not independent and such a structure will therefore be unacceptable. The benefit of a RNN compared to a FFNN is the addition of a feedback loop. Where the feedback allows the model to have an understanding of time in the sequential data.

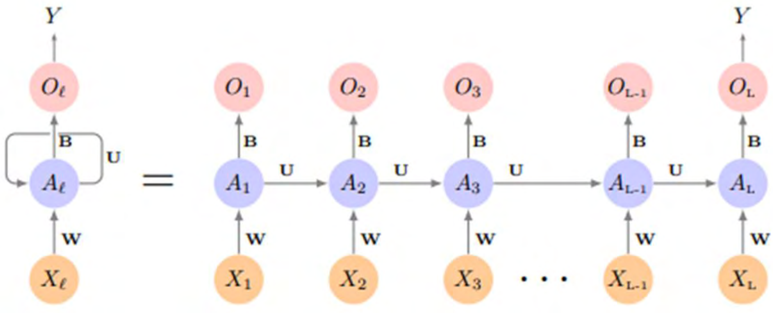


Figure 8: Illustration of RNN activation process. Source: (James et al., 2021)

The input of an RNN is a sequence (James et al., 2021). It consists of an input layer, one or more hidden layer(s) and an output layer. The activation in the hidden layer consists of both the new input, the new value in the sequence, and the hidden layer activations of the previous timestep, the earlier input value (Graves, 2008). A visualization of this is seen in Figure 8, displayed as an unfolded recurrent network. The output is calculated as seen in Equation 15. Equation 16 represents the final output which is used for the predictions. The activations are calculated per Equation 14. Combining the shared weights, input as seen in Equation 12, and the previous, hidden layer seen in Equation 13. A more detailed explanation of a RNN can be found in *"An Introduction to Statistical Learning - with Applications in R"* by James et al. (2021).

A known problem with RNNs is the vanishing or exploding gradient problems (Nielsen, 2019). Due to parameter sharing, the weights in a RNN are shared and thereby have the same value (Goodfellow et al., 2016). This causes the gradient, a learning parameter in neural net model, to become very large or very small over time, for values not equal to 1. This makes training the RNN impossible due to two reasons. One, values below 1 lead to a vanishing gradient. When finding parameter values by optimizing the loss function, the model will hit a limit on the number of steps it takes. Lowering the chances of finding the optimum. Secondly, in the case of an exploding gradient, the steps are large. In the optimization, this leads to the parameter value bouncing around, not finding the optimum either. This is important to note as it makes long sequences of data harder for a RNN to handle.

$$\textit{Input Sequence}: \mathrm{X} = \{\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_\mathrm{L}\}$$

Equation 12: RNN Input sequence

$$\textit{Components in the Input Sequence}: X_l^T = (\mathrm{X}_{l1}, \mathrm{X}_{l2}, \dots, \mathrm{X}_{lp})$$

Equation 13: RNN Components in input sequence

$$\textit{Activations in the RNN}: \mathrm{A}_{lk} = \mathrm{g}(\,\mathrm{w}_{k0} + \sum_{j=1}^{p} \mathrm{w}_{kj}\mathrm{X}_{lj} + \sum_{s=1}^{K} \mathrm{u}_{ks}\mathrm{A}_{l-1,\ s})$$

Equation 14: RNN Function for activation

$$\textit{Output Intermediary Layers}: \mathrm{O}_l = \beta_0 + \sum_{k=1}^{K} \beta_k \mathrm{A}_{lk}$$

Equation 15: RNN Function for intermediary layers

$$\textit{Final Output}: \mathrm{O}_L = \beta_0 + \sum_{k=1}^{K} \beta_k \mathrm{A}_{Lk}$$

Equation 16: RNN Formula for output

The input to the RNN model is the same as for the FFNN models. Note that the window length equals the length of the vector provided as input to the RNN. The model is defined as sequential, and the input is then fed forward to the first hidden layer. This is a SimpleRNN layer of 128 units and ReLU as the activation function. Again, as in the FFNN, a dropout layer with rate 0.2 is fitted. Then there is an additional SimpleRNN layer, now with 64 units and ReLU activation. For the regression problem, the nest layer is the final dense layer of one unit and linear activation, and for the classification problem there is first a flattening and then a dense layer of one unit with a sigmoid activation.

LSTM – Long Short Term Memory

The LSTM was introduced in 1997 as a solution to the vanishing gradient problem (Schmidhuber & Hochreiter, 1997). This has allowed the models to better capture long run dependencies in data, like in text or time series data.

The structure of the LSTM consists of a set of memory blocks (Graves, 2008). A memory block is made up of one or more self-connected memory cells and three multiplicative units. The multiplicative units are input, output, and forget gates. A series of memory cells can be seen in Figure 9. Each gate is constructed by summations of activations from inside and outside the memory block. The vanishing gradient problem is partly solved by having separate paths for long term and short term memories. Represented as the top and bottom path in Figure 9. This allows the LSTM models to handle longer sequences of input data than the RNN. A more detailed explanation of a LSTM can be found in Graves (2008).

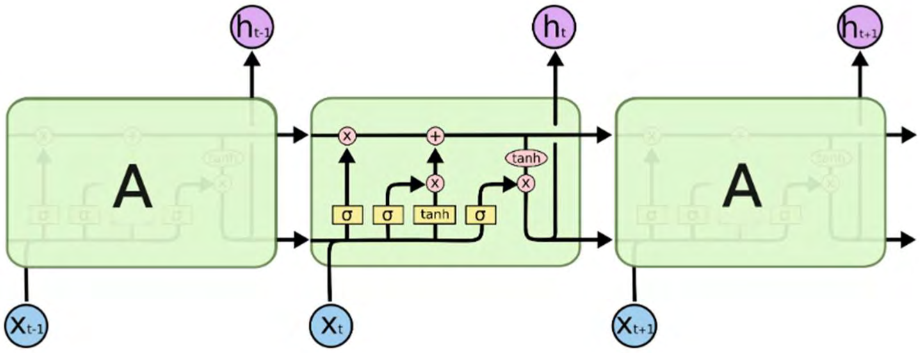


Figure 9: Illustration of LSTM memory cells. Source: (Olah, 2015)

The LSTM model is implemented by stacking three layers of an LSTM layer, each followed by a dropout layer. Each of the LSTM cells have 128 units and a ReLU activation function. The dropout rate is 0.2. As with the previous models the last layers are different for the regression and for the classification problem.

4.7 Performance Metrics

Performance metrics are used to evaluate and compare predictive models. These metrics vary depending on the prediction problem. This section introduces the performance metrics used in this thesis for both the regression and the classification problem. It includes MSE, MAE, accuracy, and AUC-ROC.

MSE and MAE

Evaluating point forecast accuracy is important in order to compare the fit of multiple models. A forecast error is the difference between the predicted and real value (Hyndman & Athanasopoulos, 2021). Two popular methods of evaluating the forecasts are by scale dependent errors: Mean squared error (MSE) and mean absolute error (MAE). The formula for calculating the metrics can be seen in Equation 17 and Equation 18. Minimizing MAE leads to a forecast of the mean, while minimizing MSE leads to a forecast of the mean, as noted by Hyndman and Athanasopoulos. MAE is more robust to outliers compared to the MSE as it takes the absolute value of the errors and not the square value that makes the MSE more sensitive to outliers. Note that for the regression DL models, the models are trained to minimize the MSE. The evaluation of the models afterwards is however based on both measures. Below follows a mathematical description of the two accuracy measures.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{pred})^2$$

Equation 17: Formula for MSE

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^{pred}|$$

Equation 18: Formula for MAE

Accuracy

The accuracy metric refers to the fraction of predictions the model got right. Equation 19 is the formal definition and Equation 20 is used when dealing with binary classification (Google Developers, 2022).

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Equation 19: Formula for accuracy

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives}$$

Equation 20: Formula for binary accuracy

The binary equation has been used when dealing with the accuracy metric in this thesis. The accuracy of a model is easy to interpret as it outputs percentage value of correct predictions. The simplicity of the metric is also its downfall as high accuracy does not necessarily mean that the model is performing educated predictions, only that it often predicts the right outcome.

AUC-ROC

The AUC – ROC curve is a performance measurement for classification problems at various threshold settings. The acronym ROC stands for “Receiver Operating Characteristics” and AUC for “Area Under the Curve”. Where ROC represents the probability curve and AUC the measure of separability (Narkhede, 2018). To generate the AUC – ROC curve Equation 21 and Equation 22 are used for calculating the sensitivity and specificity of the model. Each point on the curve represents a sensitivity/specificity pair corresponding to a particular decision threshold (Schoonjans, 2017). The sensitivity is plotted on the y-axis while the false positive rate calculated by Equation 23 is plotted on the x-axis. In Figure 12 an illustration of the ROC curve can be seen.

$$Sensitivity = True\ Positive\ Rate = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Equation 21: AUC-ROC Formula for sensitivity

$$Specificity = True\ Negative\ Rate = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

Equation 22: AUC-ROC Formula for specificity

$$False\ Positive\ Rate = 1 - Specificity = \frac{False\ Positives}{True\ Negatives + False\ Positives}$$

Equation 23: AUC-ROC Formula for False Positive Rate

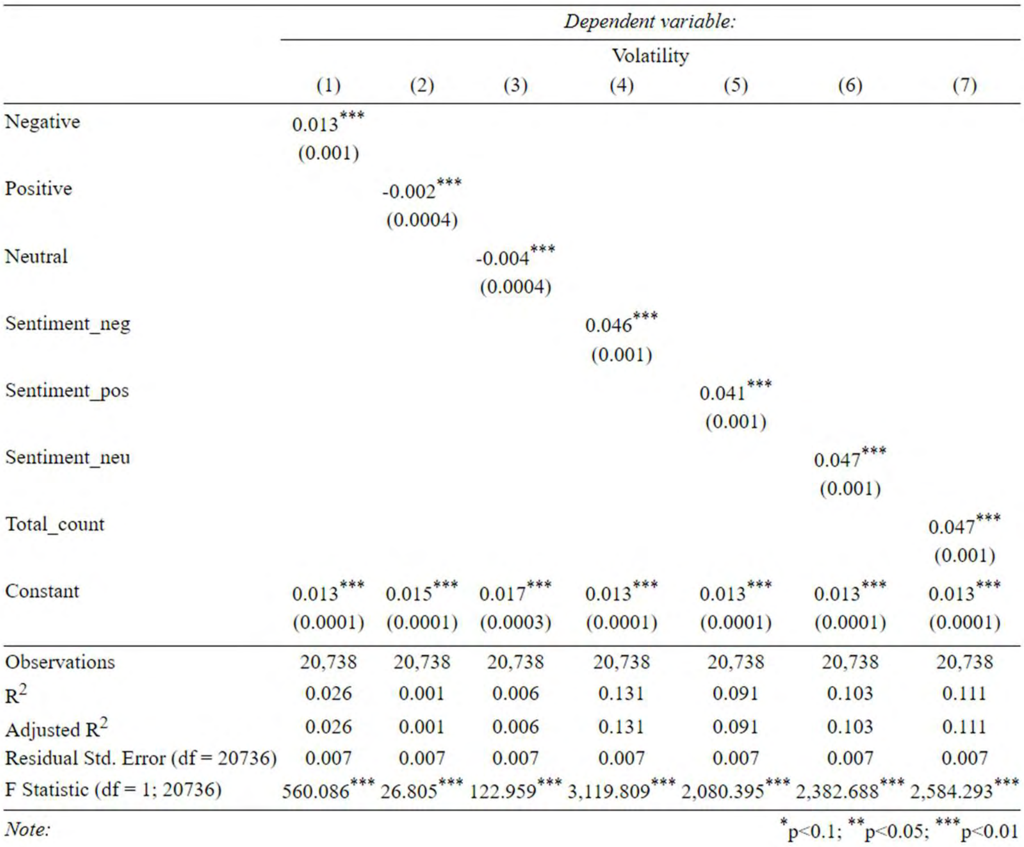
The ROC curve is used to display the models’ classification capabilities at various decision thresholds, while the numeric AUC score is used for easier comparison of models. The AUC is equivalent to the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance, which is equivalent to a two sample Wilcoxon rank-sum statistic (Chan, 2018). In this thesis the AUC score will be used to measure the multiple models’ capability of distinguishing between day-ahead higher and lower volatility days.

5 Results

This chapter presents the results of the deep learning and baseline models. First, a look at the linear regression and the linear relation between the predictors and volatility. Then, the results of the DL models for the regression problem are presented and the best model identified. This will be based on the standardized measures, MSE and MAE. Finally, a look at the classification results. The results are summarized in confusion matrices, while accuracy and AUC-ROC are used to evaluate and rank the models.

5.1 Linear Regression Inference Analysis

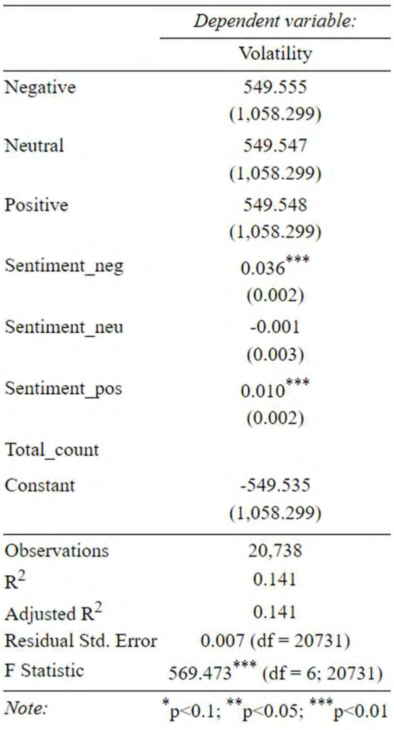
The linear regression model was introduced to make inference on the relationship between the variables used for prediction. This is due to the ease of interpretability that the linear regression model offers compared to the deep learning models. The linear regression model in this thesis will be used as a method to understand how the realized volatility is influenced by the sentiment variables developed in the methodology. This is known as inference analysis.



	Dependent variable:						
	Volatility						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Negative	0.013^{***} (0.001)						
Positive		-0.002^{***} (0.0004)					
Neutral			-0.004^{***} (0.0004)				
Sentiment_neg				0.046^{***} (0.001)			
Sentiment_pos					0.041^{***} (0.001)		
Sentiment_neu						0.047^{***} (0.001)	
Total_count							0.047^{***} (0.001)
Constant	0.013^{***} (0.0001)	0.015^{***} (0.0001)	0.017^{***} (0.0003)	0.013^{***} (0.0001)	0.013^{***} (0.0001)	0.013^{***} (0.0001)	0.013^{***} (0.0001)
Observations	20,738	20,738	20,738	20,738	20,738	20,738	20,738
R^2	0.026	0.001	0.006	0.131	0.091	0.103	0.111
Adjusted R^2	0.026	0.001	0.006	0.131	0.091	0.103	0.111
Residual Std. Error (df = 20736)	0.007	0.007	0.007	0.007	0.007	0.007	0.007
F Statistic (df = 1; 20736)	560.086^{***}	26.805^{***}	122.959^{***}	$3,119.809^{***}$	$2,080.395^{***}$	$2,382.688^{***}$	$2,584.293^{***}$
Note:							*p<0.1; **p<0.05; ***p<0.01

Table 16: Summary of single variable regression

Table 16 depicts all of the predictors regressed against the response variable, *"Volatility"*. All of the predictors are significant. However, the variables related to number of articles, *"Total_count"*, *"Sentiment_neg"*, *"Sentiment_neu"*, and *"Sentiment_pos"* have the highest R-Squared. This implies that the sheer volume of text items is important. Another noteworthy piece of information is that while negative sentiment increases volatility, both neutral and positive news seems to have a negligible positive impact. The next part is to look at the relationship between volatility and all of the predictors. The summary of the multiple linear regression model can be seen in Table 17.



	Dependent variable:
	Volatility
Negative	549.555 (1,058.299)
Neutral	549.547 (1,058.299)
Positive	549.548 (1,058.299)
Sentiment_neg	0.036^{***} (0.002)
Sentiment_neu	-0.001 (0.003)
Sentiment_pos	0.010^{***} (0.002)
Total_count	
Constant	-549.535 (1,058.299)
Observations	20,738
R^2	0.141
Adjusted R^2	0.141
Residual Std. Error	0.007 (df = 20731)
F Statistic	569.473^{***} (df = 6; 20731)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 17: Summary of multiple regression

When implementing the multiple linear regression model, an adjusted R-squared of 14.1% was achieved. An overall higher explanatory power than any of the simple regression models. However, the less impactful explanatory variables such as the sentiment averages lose their significance. The volume of neutral texts also sees a reduction in significance while the two others retain their significance. This seems to suggest that the volume of neutral texts becomes less important to the observed volatility when measures such as positive and negative volume are present. Note that the issue of multicollinearity leaves the values for total count blank. The multiple linear regression model is used as a baseline model for the other prediction models.

Summary of the Inference Analysis

In the simple regression models, all of the variables are significant. Albeit they do not explain much of the variation, resulting in a low R-squared. The count predictors explained the most while the sentiment averages had a minuscule contribution. This was reinforced by the multiple linear regression model.

However, these models do not capture non-linear relationships that could be important to explain the observed volatility. Overall, the result of the analysis is meant to be explorative to intuitively understand what might be important between the variables and volatility before the review of the deep learning models, which are less interpretable.

5.2 Prediction Models

The results from the prediction models are split in two. First, the results from the regression models, followed by the results from the classification models.

Deep Learning Models - Regression Results

In this section, the results for the regression problem are presented. The models used for this task have been FFNNs, RNNs, and LSTMs. The predictors used as input have been either previous volatility or previous volatility in addition to the sentiment predictors presented earlier. Some of the model types, the RNNs and the LSTMs, are designed for sequential data like time series. To take advantage of this, the time window has a value of 1, 5 or 21, representing trading days. This can be interpreted as using values from 1, 5 or 21 of the prior trading days.

Overall, the models perform better when fed data from larger time windows i.e., the results from T=21 are better than T=5 and T=5 are better than T=1. It is important to note that some of the results have very similar results and due to randomly generated activations, some randomness is expected. This makes it more important to look at trends rather than absolute values.

Looking at Figure 10 and Figure 11, they present test predictions for the best models. The red line represents the actual volatility. The blue, orange, and green lines represent the FFNN, RNN, and LSTM respectively. With only the previous volatility as predicator, the models seem to have a clear lag and relatively smooth estimate with few extreme predictions. When all predicators are used, the results seem to fit the actual values in red a bit better. Evidently, sentiment seems to be a leading indicator as the prediction of future volatility have a better fit without lag. However, in some instances the model predicts a sharp increase in volatility when the opposite is true.

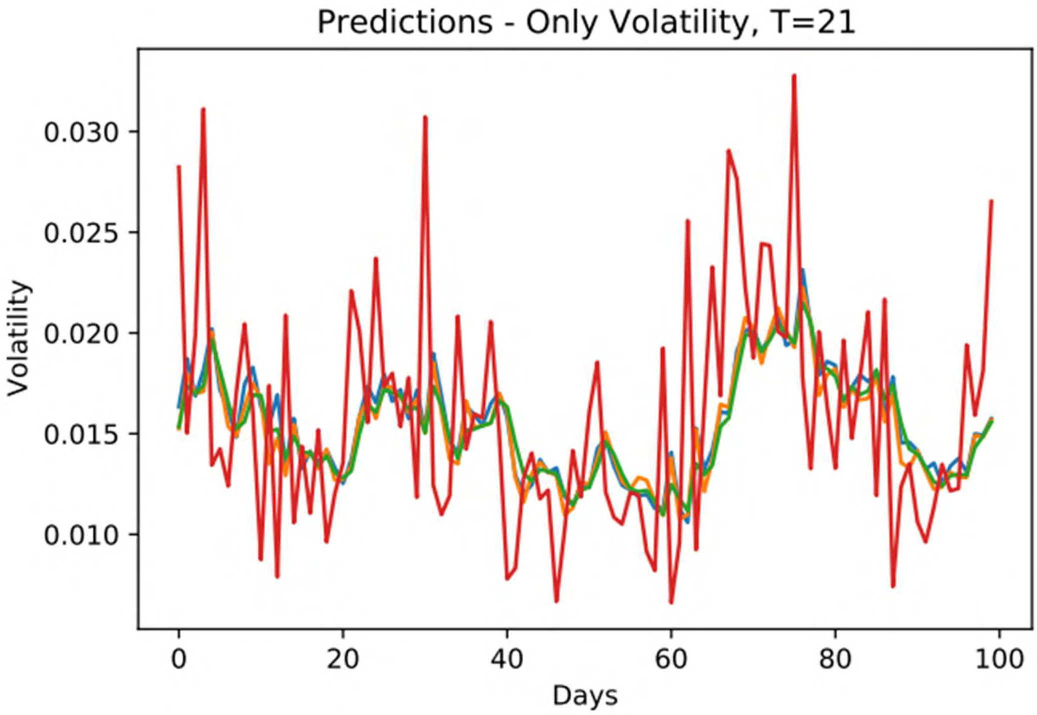


Figure 10: Plot of Predicted volatility vs Actual volatility without sentiment variables

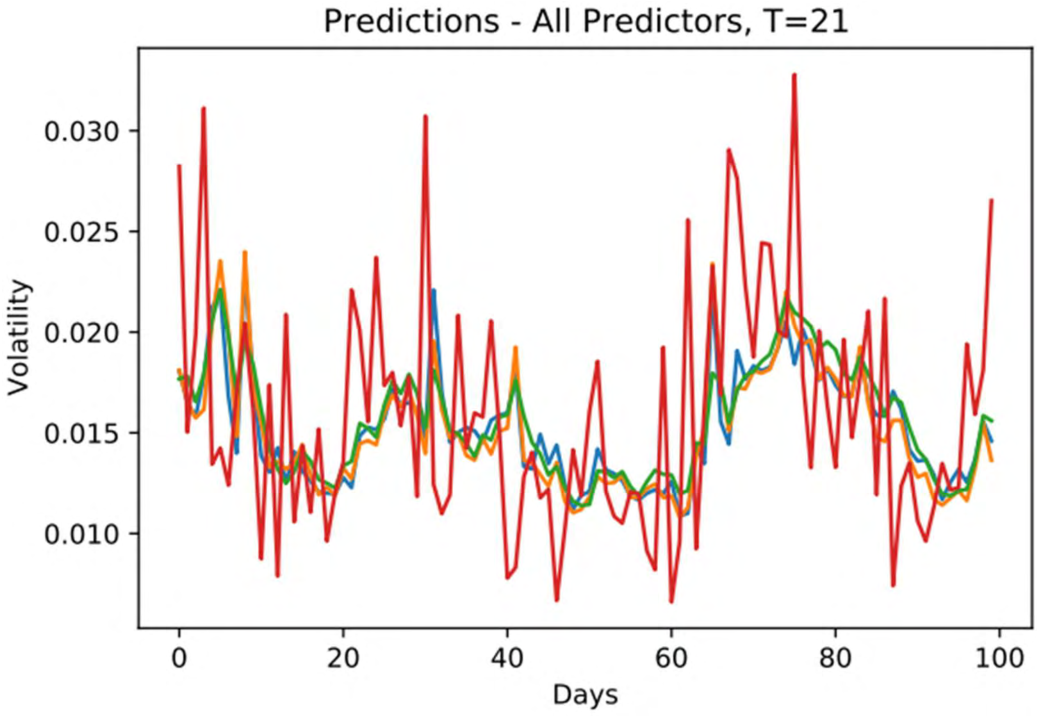
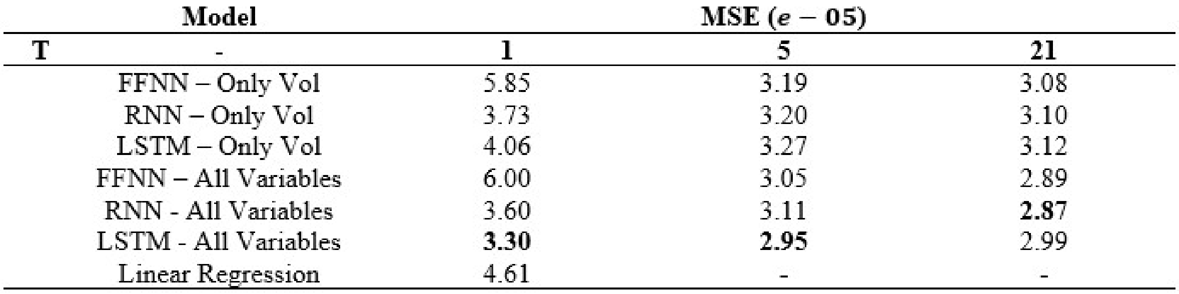


Figure 11: Plot of Predicted volatility vs Actual volatility with sentiment variables

Mean Squared Error

Table 18 presents the MSE for the models' prediction. For T=1, the LSTM with all the predictors achieves the best performance. Roughly 10% better than the second best, the RNN with all predictors. Remember that the LSTM have significantly more parameters. Input for T=5 is a sequence of data for the five prior trading days. Here, the LSTM with all predictors has the lowest MSE. T=21, has the longest input sequence. The models are also achieving the lowest MSE. Indicating that there is a relation between past and future values. Here, the RNN with all predictors has the best MSE. It is the best overall MSE score too.

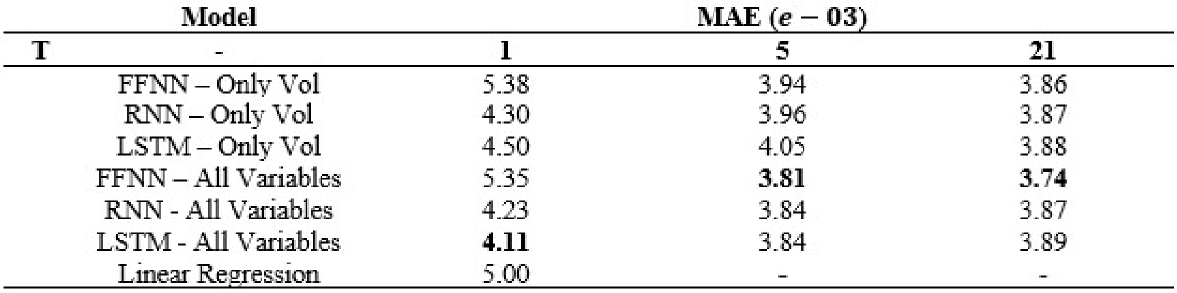


	Model	**MSE ($e-05$)**		
T	-	**1**	**5**	**21**
	FFNN – Only Vol	5.85	3.19	3.08
	RNN – Only Vol	3.73	3.20	3.10
	LSTM – Only Vol	4.06	3.27	3.12
	FFNN – All Variables	6.00	3.05	2.89
	RNN - All Variables	3.60	3.11	**2.87**
	LSTM - All Variables	**3.30**	**2.95**	2.99
	Linear Regression	4.61	-	-

Table 18: Comparison of MSE

Mean Absolute Error

Table 19 presents the MAE for the models' prediction. Compared to the MSE results, the MAE results have several similar tendencies. The results for T=1 have the LSTM with all predicators as the best model too. However, for T=5 it is the FFNN with all variables and not the LSTM that has the lowest MAE at 3.81, marginally better than the RNN and LSTM. The major difference in the results is that for T=21, the MAE for the FFNN is improved compared to T=5, but for both the RNN and LSTM the results are worse. This was not the case for the MSE.



	Model	**MAE ($e-03$)**		
T	-	**1**	**5**	**21**
	FFNN – Only Vol	5.38	3.94	3.86
	RNN – Only Vol	4.30	3.96	3.87
	LSTM – Only Vol	4.50	4.05	3.88
	FFNN – All Variables	5.35	**3.81**	**3.74**
	RNN - All Variables	4.23	3.84	3.87
	LSTM - All Variables	**4.11**	3.84	3.89
	Linear Regression	5.00	-	-

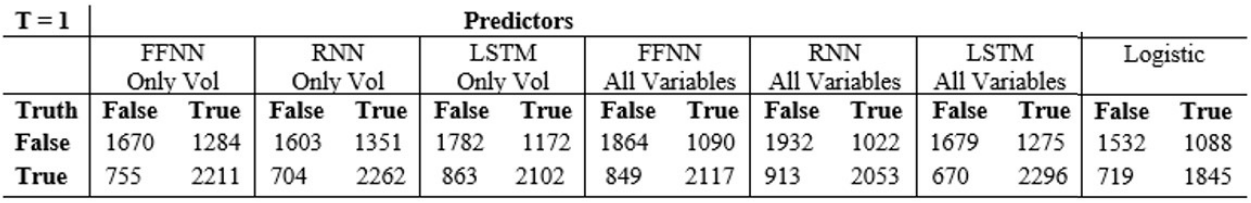
Table 19: Comparison of MAE

Classification

This section covers the results of the classification models. The models predict whether the next day will either have increased or decreased volatility compared to the day before. Both a naïve model and logistic regression model serve as benchmarks to the DL models. The naïve model predicts the same value, one, for each prediction. The metrics used to evaluate the performance of the models are accuracy and AUC-ROC. The predictions are visualized through confusion matrices.

Confusion Matrix

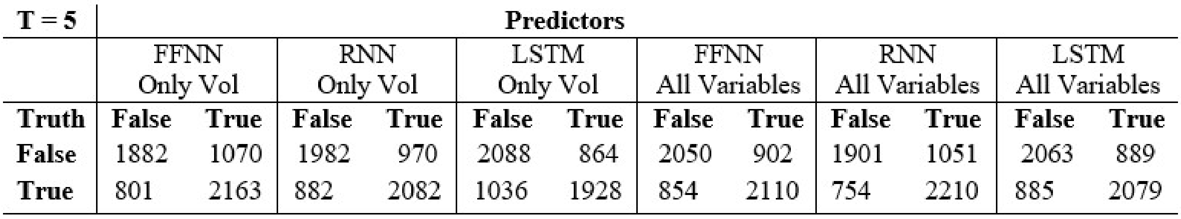
The confusion matrix is introduced to increase the interpretability of the classification results. The matrices display the results for the different time window. The result of allowing 1 trading day of data is presented in Table 20. However, the confusion matrix for T=5 is in focus due to achieving the best results.



T = 1	Predictors													
	FFNN Only Vol		RNN Only Vol		LSTM Only Vol		FFNN All Variables		RNN All Variables		LSTM All Variables		Logistic	
Truth	**False**	**True**	**False**	**True**	**False**	**True**	**False**	**True**	**False**	**True**	**False**	**True**	**False**	**True**
False	1670	1284	1603	1351	1782	1172	1864	1090	1932	1022	1679	1275	1532	1088
True	755	2211	704	2262	863	2102	849	2117	913	2053	670	2296	719	1845

Table 20: Confusion matrix for 1 trading day of data

Table 21 showcases that the addition of sentiment variables to the RNN model seems to raise the ability to classify higher volatility days but also reduce the ability to classify lower days. This is seen through the increase in true positives and a decrease in true negatives. It also raises the misclassification of low volatility days as the model has increased the number of false positives. The FFNN and LSTM in general see a better improvement across the board by the addition of sentiment. Both models see improvement in correctly classifying true positives and true negatives while FFNN see a miniscule increase in misclassification of lower volatility and LSTM in higher volatility days.



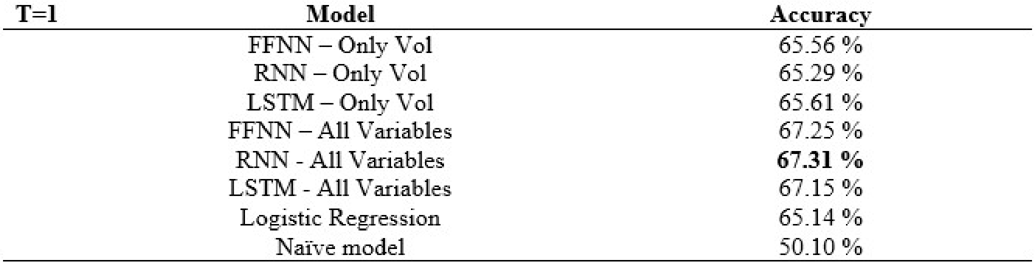
T = 5	Predictors											
	FFNN Only Vol		RNN Only Vol		LSTM Only Vol		FFNN All Variables		RNN All Variables		LSTM All Variables	
Truth	**False**	**True**	**False**	**True**	**False**	**True**	**False**	**True**	**False**	**True**	**False**	**True**
False	1882	1070	1982	970	2088	864	2050	902	1901	1051	2063	889
True	801	2163	882	2082	1036	1928	854	2110	754	2210	885	2079

Table 21: Confusion matrix for 5 trading days of data

Overall, the results from the confusion matrix do suggest that the addition of sentiment variables enables the models to correctly classify true positives and true negatives. While the matrix offers a readable format of the output and suggests improvement, the other metrics such as accuracy and AU-ROC offer ease of comparison between the models.

Accuracy

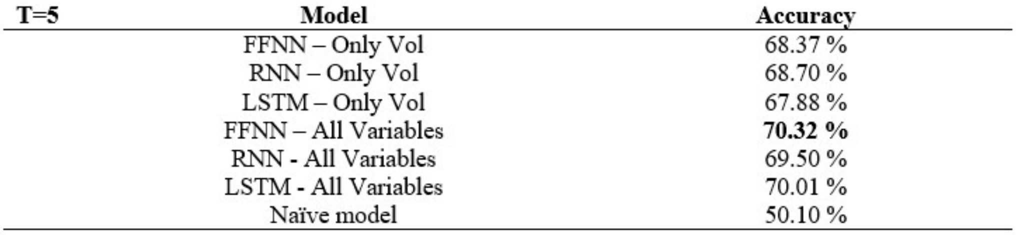
Table 22 presents the accuracy of the classification with the models set to use time window equal to one. All of the models beat the naïve benchmark model. Out of the seven different models the Recurrent Neural Network trained with the sentiment variables performs the best and achieves an accuracy of 67.31%. The logistic model is included for T=1. It includes all the variables, but it still performs worse than the deep learning models. Therefore, no effort is made to simulate time windows with the creating of lagged variables for further comparison.



T=1	Model	Accuracy
	FFNN – Only Vol	65.56 %
	RNN – Only Vol	65.29 %
	LSTM – Only Vol	65.61 %
	FFNN – All Variables	67.25 %
	RNN - All Variables	**67.31 %**
	LSTM - All Variables	67.15 %
	Logistic Regression	65.14 %
	Naïve model	50.10 %

Table 22: Accuracy for 1 trading day of data

The result of allowing more data to be used can be seen in Table 23. All of the models see improvements in accuracy when T=5, which allows for more data points to be used in the classification process. The benchmark remains the same as it does not rely on any prior knowledge of data. However, now it is observed that the FFNN now achieves the highest accuracy of 70.32%. Note that the models without the sentiment variables still perform worse compared to when they are included. Meaning, that sentiment appears to convey some information that the networks can utilize in order to improve their accuracy.



T=5	Model	Accuracy
	FFNN – Only Vol	68.37 %
	RNN – Only Vol	68.70 %
	LSTM – Only Vol	67.88 %
	FFNN – All Variables	**70.32 %**
	RNN - All Variables	69.50 %
	LSTM - All Variables	70.01 %
	Naïve model	50.10 %

Table 23: Accuracy for 5 trading days of data

Overall, the observations from the confusion matrix do seem to line up well with the comparison of accuracy between the models. The addition of longer sequences of data provided to the models have increased the classification capabilities. However, the effect of adding sentiment variables to the models do remain constant. The classification results are not improved by increasing the time window to 21 like it did for the regression results. Therefore, these results have not been included.

AUC- ROC

The last sections evidently displayed that increasing the time window from 1 to 5 improves the general performance of the models. This section present the ROC curves for the best time window, T equal to 5. Figure 12 includes the curves for all the models, with and without sentiment variables.

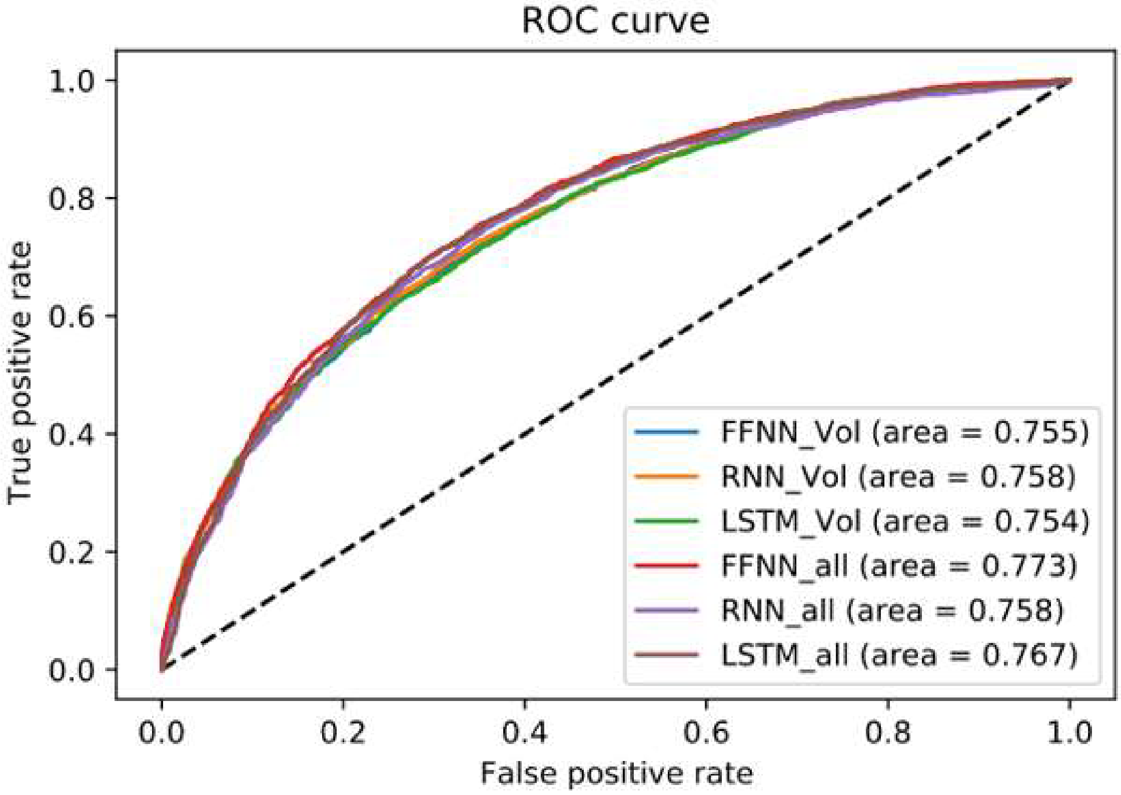
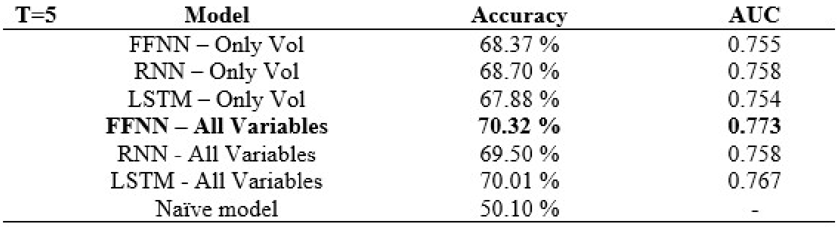


Figure 12: ROC Curves for DL models

When interpreting Figure 12, it can be seen that the models are quite close in classification abilities at the various thresholds. The performance of the models is clearer when looking at the AUC scores presented in the same figure. The RNN models have the same AUC when including and excluding the sentiment variables. The other models do see an improvement from the addition of sentiment. All of the models achieve an AUC > 0.5 indicating that the models performs better than random guessing. Further on, the FFNN model sees the highest increase and yields the best AUC score of 0.773 which translates to being the best model at distinguishing between the positive and negative classes. In this case, being the best at distinguishing between high and low volatility days.

Metrics summarization

Table 24 summarizes the metrics used for classification. When comparing the metrics used to evaluate the models, the best performer is the Feed Forward Neural Network trained with volatility and sentiment variables. While some of the metrics are quite close, it should be noted that there is an improvement when sentiment is included in the classification.

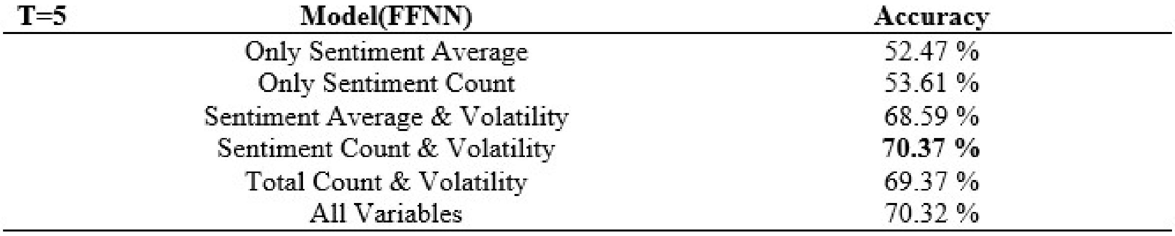


T=5	Model	Accuracy	AUC
	FFNN – Only Vol	68.37 %	0.755
	RNN – Only Vol	68.70 %	0.758
	LSTM – Only Vol	67.88 %	0.754
	FFNN – All Variables	**70.32 %**	**0.773**
	RNN - All Variables	69.50 %	0.758
	LSTM - All Variables	70.01 %	0.767
	Naïve model	50.10 %	-

Table 24: Metric summarization for classification

Controlling the Variables

Summarizing across the different metrics, it can be seen that the FFNN model performs the best. However, before proceeding to the discussion, it can be insightful to look at what drives this performance. The question of whether it is the average sentiment variables or the count variables that influences the performance needs to be investigated. Table 25 displays the FFNN model with different combinations of the variables that makes it possible to compare the results.



T=5	Model(FFNN)	Accuracy
	Only Sentiment Average	52.47 %
	Only Sentiment Count	53.61 %
	Sentiment Average & Volatility	68.59 %
	Sentiment Count & Volatility	**70.37 %**
	Total Count & Volatility	69.37 %
	All Variables	70.32 %

Table 25: Accuracy for variable isolation

While the models are only compared for the accuracy measure, it can be seen that previous volatility impacts the next day's volatility the most. A noteworthy takeaway from this comparison is that the model with total count as a predictor performs worse than the model where count and sentiment are combined. The distribution of text as negative, positive, or neural by a strict classification rule based on the sentiment scores, seem to have an effect when classifying whether the next day's volatility is higher or lower than the previous day. The average sentiment probability scores themselves performs the worst. The strict classification approach thereby provides more accurate predictions. The model does see improvement from reducing the number of variables albeit this difference is marginal at best of 0.05 % improvement over the model with all the variables. This can be attributed to randomness of the activation process the model is trained with. Therefore, only comparing accuracy for marginal differences, as can be seen here, is not good enough to determine the best variable selection.

6 Discussion

The goal of this thesis has been to see if sentiment serves as a useful predictor in prediction of market volatility, as per the research question. In the previous chapter, the results from the prediction models were presented. This chapter discusses the validity of these results. In addition, it includes suggestions for further research.

6.1 Sentiment as a Predicator

Overall, the deep learning prediction models all get improved results when adding the two types of sentiment predicators to the models based on prior volatility. This indicates that sentiment from news and social media is significant in the prediction of future volatility. Additionally, longer sequences of data increased performance by allowing the models to capture trends in the data.

The linear regression model and the results from the DL-models clearly coveys the importance of sentiment as a predictor. However, the impact from two different representations of sentiment used in this thesis seem to be significantly different. This could imply that the way sentiment is represented is essential to make use of sentiment as a predicator of volatility.

In this thesis, sentiment have been defined as the collective opinion of the market, ranging from positive to negative. The probabilities provided by FinBERT have been the basis for the two types of sentiment predictors, sentiment average and sentiment count. Since the thesis aims to investigate if sentiment is a useful predictor of volatility, isolating the effect from sentiment is key.

The linear regression used for inference analysis provides useful insights. We regressed each predictor to volatility and found all predicators significant. However, with varying explanatory power. All of the sentiment count variables had higher R-squared than the sentiment average variables. The impact of the total count variable, a variable not affected by sentiment scores, is similar to the count based sentiment variables. The real impact from sentiment count variables might be artificially high due to the correlation with total count.

This narrative is supported by literature. The results we achieved were similar to the findings presented in the literature review. The discovery by Antweiler & Frank (2002), that an above number of messages could forecast higher levels of volatility is in line with the results from this thesis' count based predictors. In addition, the studies by Tetlock (2005) and Kothari et al. (2009) links negative sentiment with increased volatility. This is confirmed by the sentiment average variables in this thesis.

It could be possible that an entirely different representation of sentiment could have enhanced the results. Looking at the comparable studies in the literature review, sentiment have been represented differently and thus achieved superior results.

6.2 Comparable Studies

The up/down classification predictions in this thesis achieved an accuracy of 70.3%. While this is a major improvement from the baseline of 50.1%, it is a significant difference compared to Bollen et al. (2011) that achieved 86.7%. The data gathered in the study is approximated to be roughly 9 850 000 tweets. The first big difference between the thesis and the study is that this thesis collects company specific text while they collect non-specific text to the DJIA. They investigate whether the public mood can predict DJIA values rather than if specific text mentioning DJIA can be predictive. They also apply a different technique to extract what they believe to be relevant text for sentiment. Instead of using all of the data after the standard pre-processing of stop-words and punctuation removal, they explicitly extract tweets that contain certain expressions. These expressions consist of keyword combinations like “I feel”, “I am feeling”, “I’m feeling”, “I’m”, “Im”, “I am”, and “Makes me”. The process of doing so is to measure the mood of the public through categorical labeling of tweets, rather than the specific sentiment tied to text. After this process they do not mention how many data points are left for analysis. Sparce data could increase variance of the predicted outcome, thereby effecting the validity and consistency of the results.

This is a major difference in approach than what has been done in this thesis. In this thesis tweets after filtering are deemed as relevant to the sentiment of the ticker without applying specific keyword matching of the content. The advantage of applying keyword matching in the study is that it ensures strong intent behind the text used in the analysis but at the cost of potentially reducing the size of the dataset drastically. Due to this, if specific keyword matching for the company specific text had been applied, the outcome may be insufficient observations for less popular companies. The thesis also investigates whether company specific text can be predictive for the movement of the stocks rather than if the general hivemind of social media is highly correlated or even predictive with the movement of an index. Recall that the idea of the thesis is a general model to predict stock specific volatility, while the study constructs a specific model to measure public mood for index level movements.

On the basis of this, the accuracy of 70.3% cannot be said to be inherently bad compared to 86.7% of Bollen et al. (2011). First, the study and the thesis aim to measure movement of assets differently in the form of company specific text vs nonspecific categorical mood labeling. Further on, the thesis looks at multiple assets rather than an index. Finally, there is a significant difference in what is deemed as relevant text to use for sentiment analysis. Although, when comparing the model of this thesis to the results of a more specified model, with different representation of sentiment, it underperforms.

6.3 Model Architecture

It is intriguing to see that the models have relatively similar results. Especially given the different model architectures. The three different DL models were utilized because RNNs and LSTMs possibly could have better performance on time series than the FFNN as they are better suited for sequential data. This was not the case, and they only had a slightly better result for the regression problem and a worse result for the classification problem. The FFNN seems to capture the same information from the data. The FFNN model is smaller, compared to the LSTM and the RNN, and it has far fewer trainable parameters in total. Due to this, the model trained faster and the need for computing power is lower. It could be argued that since the FFNN achieves similar results as the more complex models, it should be preferred. This does however depend on the impact of marginally improved results.

6.4 Noisy Data

In the data preparation section of this thesis various techniques used to reduce the noisiness that often comes with textual data was introduced. Through these techniques the dataset was reduced quite drastically. However, from the explorative data analysis of the sentiment scoring it could still be inferred that the data contained noisy observations.

The implication of noisy textual data for the thesis is particularly impactful as the thesis combines the use of a pre-trained sentiment model to provide scores for training volatility models. Which implies that an overabundance of noisy observations would directly interfere with the training process of the volatility models. The aim of a general model for multiple assets may also further amplify this effect. Mainly, since the models are not provided with the stock symbol of the observation for prediction. This leaves no clear way for the models to differentiate potential noise filled stocks from influential stocks.

Section 6.2 looked at the comparison of the best classification model in this thesis and the model from Bollen et al. (2011). The differences between the implementation process and overall goal of the models were discussed but one valid explanation for the large differences in accuracy could be noisy data affecting the model. The potential for noise when working with large datasets of text is always high since human interaction through language is inherently complex, and even state-of-the-art deep learning models cannot guarantee correct labeling of intent derived from the observations.

6.5 Weaknesses

Recall that the model for sentiment classification, FinBERT, is not properly trained and optimized for social media texts. From the previous discussion about noisy data this becomes a significant weakness of the volatility models' predictive capabilities. The implication of problematic sentiment classification of social media text comes from the fact that a large part of the data is collected from Twitter. This is most likely affecting the true values for the sentiment variables used as input to the deep learning models for regression and classification. This is due to how the sentiment average variables are calculated. All of the observations are equally weighed and are indifferent of the collection source. The result of this is that Twitter ends up contributing the most to the daily observation of sentiment for the respective stock symbol. Which in turn skews the representation of sentiment in favor of Twitter.

While the problems regarding the high number of neutral text probabilities might be related to the training process of FinBERT, it also may indicate that the data need further filtering. When dealing with large datasets of text one of the most important steps is to apply proper filtering methods to ensure high quality of the text. In section 4.1, source specific filtering was introduced to the data, but this may not have been enough. Due to the scale of the data, it is not feasible to go through the text manually for verification of the quality. The filtering applied is the process of reviewing a sample of the dataset to obtain insight of what may cause noise in the data. While the filtering did reduce the dataset drastically, it hardly captures all unique issues tied to the observations. The effect of noisy data that may impact the results of this thesis could be reduced by keyword specific collecting methods. For future research a suggestion would be to add a list of words alongside the query of interest when gathering data. By doing so when collecting data, rather than filtering the data based on this list of words, the problems that may come with a significant reduction of the dataset can thereby be avoided. Overall, what may negatively impact the prediction results could be attributed to the processing of the textual data, resulting in noisy numeric data for the training process of the volatility models.

6.6 Further Research

The weaknesses affecting the FinBERT model in properly classifying financial social media texts do warrant further research into the field of social media texts in financial settings. After reviewing the thesis, an idea would be to incorporate the deep learning model RoBERTa, described in Liu et al. (2019), to FinBERT. The model is an extension of the BERT model which FinBERT also builds on but is heavily trained on social media texts compared to both BERT and FinBERT. An example would be to create an ensemble model of the two or incorporate the training process of RoBERTa to FinBERT. Essentially creating the FinBERTa.

Another way to implement sentiment in volatility models would be to combine the industry standard GARCH model with sentiment factors. The calculation of volatility used in financial value-at-risk modeling is heavily dominated by the autoregressive method, GARCH (Bollerslev, 1986). From the findings of this thesis and previous studies, one can see that sentiment does in fact provide predictive capabilities for volatility in financial markets. It would be interesting to see if a sentiment volatility model trained with GARCH volatility could outperform the industry standard of a GARCH (1,1) model.

7 Conclusion

This thesis has investigated if the public available information in the form of news and social media can aid in the prediction of volatility. Text from the 100 largest companies in the S&P500 has been gathered from Twitter, Reddit and Eikon and analyzed by the pre-trained sentiment model called FinBERT. The sentiment output from FinBERT have been used to represent sentiment in the form of average sentiment probabilities and count based sentiment values. A sole, count based variable is also included. These variables, in addition to volatility, has served as training data for the deep learning models. FFNN, RNN, and LSTM models have been used to predict day-ahead volatility. The prediction of day-ahead volatility has been both formulated as a regression problem and a classification problem. The regression output is a single point forecast, and the classification output predicts if the day- ahead volatility is higher or lower than the previous day's volatility.

The best model for the regression problem differs when evaluating the MAE and MSE. Overall, the results are very similar. However, the FFNN with a sequence length of 21 trading days, has the lowest MAE in addition to lowest model complexity. Due to this, the FFNN model should be preferred for implementation. The classification of higher or lower volatility had greater differences between the models. There, the FFNN(T=5) model achieved the best results. A question is raised of whether this performance is driven by the count variables or the sentiment probabilities. Through inference analysis it can be seen that the count of strictly classified negative sentiment provides the highest R-squared of 13.1 %. The effect of the count variables is reaffirmed by isolating the variables for the classification models.

The findings from this thesis are sufficient to answer the research question: *"Can the sentiment of public available information in news and social media aid in prediction of stock market volatility?"*. The inference analysis clearly displays a statistically significant relationship between volatility and sentiment. Further on, both the regression and classification models display improved performance by adding the sentiment variables, even when controlling for impact from the other variables.

Although, when comparing the results to studies in the literature review, there is a significant difference. The best classification model in this thesis achieves an accuracy of 70.32 % while a comparable study achieves 86.7%. This could be attributed to an overall difference in model goals and representation of sentiment but may come from weaknesses such as the text filtration or FinBERT's lack of training on social media text.

References

Agarap, A. (2018, March 22). *Deep Learning using Rectified Linear Units (ReLU).* Retrieved from Arxiv: https://arxiv.org/pdf/1803.08375.pdf

Alammar, J. (2018, June 27). *The Illustrated Transformer*. Retrieved from Github: https://jalammar.github.io/illustrated-transformer/

Alammar, J. (2018). *Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)*. Retrieved from Github: https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Antweiler, W., & Frank, M. (2002, January 28). *Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards*. Retrieved from Wernerantweiler : https://wernerantweiler.ca/public/noise-a.pdf

Araci, D. (2019, August 27). *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models.* Retrieved from Arxiv: https://arxiv.org/pdf/1908.10063.pdf

Araci, D., & Genc, Z. (2020, July 31). *FinBERT: financial sentiment analysis with BERT*. Retrieved from Prosus: https://www.prosus.com/news/finbert-financial-sentiment-analysis-with-bert/

Awartani, B., & Corradi, V. (2005). Predicting the volatility of the S&P-500 stock index via GARCH models: the role of asymmetries. *International Journal of Forecasting* , 167 – 183.

Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., & Blackburn, J. (2020). *Aaai.* Retrieved from The Pushshift Reddit Dataset: https://ojs.aaai.org/index.php/ICWSM/article/view/7347/7201

Boghe, K. (2020, July 22). *We Need to Talk About Sentiment Analysis*. Retrieved from Medium: https://medium.com/swlh/we-need-to-talk-about-sentiment-analysis-9d1f20f2ebfb

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 1-8.

Bollerslev, T. (1986). GENERALIZED AUTOREGRESSIVE CONDITIONAL HETROSKEDASTICITY. *Journal of Econometrics*, 307-327.

Bricken, A. (2021, November 14). *Does BERT Need Clean Data? Part 2: Classification.* Retrieved from Towards Data Science: https://towardsdatascience.com/does-bert-need-clean-data-part-2-classification-d29adf9f745a

Briggs, J. (2021, July 27). *Why Are There So Many Tokenization Methods For Transformers?* Retrieved from Towards Data Science: https://towardsdatascience.com/why-are-there-so-many-tokenization-methods-for-transformers-a340e493b3a8

Byström, H. (2016, May). Language, news and volatility. *Journal of International Financial Markets, Institutions and Money*, pp. 139-154.

Chan, C. (2018, July 5). *What is a ROC Curve and How to Interpret It*. Retrieved from DISPLAYR: https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/

Choubey, V. (2020, July 8). *Text classification using CNN* . Retrieved from Medium: https://medium.com/voice-tech-podcast/text-classification-using-cnn-9ade8155dfb9

Coval, J., & Shumway, T. (2001). Is Sound Just Noise? *The Journal of Finance*, 1887-1910.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018, October 11). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Retrieved from Arxiv: https://arxiv.org/pdf/1810.04805.pdf

Doran, J., Peterson, D., & Price, S. (2012). Earnings Conference Call Content and Stock Price: The Case of REITs. *Journal of Real Estate Finance and Economics*, 402-434.

Eden Ai. (2022, November 23). *Top 10 Language Detection APIs*. Retrieved from Eden Ai: https://www.edenai.co/post/top-10-language-detection-apis

Eisenstein, J. (2019). *Introduction to Natural Language Processing.* MIT Press.

Gallagher, S., Rafferty, A., & Wu, A. (2004). *Natural Language Processing*. Retrieved from Stanford: https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/nlp/overview_history.html

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Chapter 10: Sequence Modeling: Recurrentand Recursive Nets.* Retrieved from Deep Learning: https://www.deeplearningbook.org/contents/rnn.html

Google Developers. (2022, July 18). *Machine Learning*. Retrieved from Google: https://developers.google.com/machine-learning/crash-course/classification/accuracy

Graves, A. (2008). *Supervised Sequence Labelling with Recurrent Neural Networks.* Retrieved from University of Toronto: https://www.cs.toronto.edu/~graves/preprint.pdf

Hayes, A. (2022, August 20). *Implied Volatility vs. Historical Volatility: What's the Difference?* Retrieved from Investopedia: https://www.investopedia.com/articles/investing-strategy/071616/implied-vs-historical-volatility-main-differences.asp

Hinton, G. (2014). *Overview of mini-batch gradient descent.* Retrieved from University of Toronto: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Hull, J. (2017). *Options, Futures, and Other Derivatives, Global Edition.* Pearson Education Limited.

Hyndman, R., & Athanasopoulos, G. (2021, May 31). *Forecasting: Principles and Practice*. Retrieved from Otexts: https://otexts.com/fpp3/accuracy.html

Ingram, R., & Frazier, K. B. (1980). Environmental Performance and Corporate Disclosure. *Journal of Accounting Research*, 614-622.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical - with Applications in R - Second Edition.* Springer.

Jurafsky, D., & Martin, J. (2021, December 29). *Speech and Language Processing.* Retrieved from Stanford: https://web.stanford.edu/~jurafsky/slp3/

Kearny, C., & Liu, S. (2013, February 9). *Textual Sentiment in Finance: A Survey of Methods and Models.* Retrieved from SSRN: https://deliverypdf.ssrn.com/delivery.php?ID=769100067111074029002089021000091101005063061035027036094120126100090092095070005078024006023047020040034099066102102090000096008086008054041107122066086124005079030068062031023106006093001101012004022064112086

Keras. (2022). *Dropout layer*. Retrieved from Keras: https://keras.io/api/layers/regularization_layers/dropout/

Kingma, D., & Ba, J. (2014, December 22). *Adam: A Method for Stochastic Optimization.* Retrieved from Arxiv: https://arxiv.org/pdf/1412.6980.pdf

Kothari, S., Li, X., & Short, J. (2009). The Effect of Disclosures by Management, Analysts, and Business Press on Cost of Capital, Return Volatility, and Analyst Forecasts: A Study Using Content Analysis. *The Accounting Review*, 1639-1670.

Kraus, M., & Feuerriegel, S. (2017, October 11). *Decision support from financial disclosures with deep neural networks and transfer learning.* Retrieved from Arxiv: https://arxiv.org/pdf/1710.03954.pdf

Li, G., Feng, S., & Jun, T. (2016). Textual analysis and machine leaning: Crack unstructured data in finance and accounting. *The Journal of Finance and Data Science*, 153-170.

Lipton, Z., & Berkowitz, J. (2015, June 5). *A Critical Review of Recurrent Neural Networks.* Retrieved from Arxiv: https://arxiv.org/pdf/1506.00019.pdf

Liu et al. (2019, July 26). *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* Retrieved from https://arxiv.org/abs/1907.11692

Loughran, T., & McDonald, B. (2010, March 5). *When is a Liability not a Liability? Textual Analysis, Dictionaries, and 10-Ks*. Retrieved from SSRN: https://deliverypdf.ssrn.com/delivery.php?ID=520006116096072014069103103064125005120037062046029025071064119071069122074069091005018107016026040058048089071090102102098113040072009047028111100125094125031036028079016123098113116112002086124091109096016122

Loughran, T., & McDonald, B. (2016, May 20). *Textual Analysis in Accounting and Finance: A Survey*. Retrieved from SSRN: https://deliverypdf.ssrn.com/delivery.php?ID=533082078095104088117085096006005122024088054014066064078069083101081066118113090078049030032063122035021064073001124006127107024018060008053088090014127015100067023010002107071003068094068010114006019004087011

Manning, C. (2015). Computational Linguistics and Deep Learning. *Computational Linguistics*, 701-707.

Manning, C., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrival.* Cambridge: Cambridge University Press.

Manzan, S. (2017, May 11). *INTRODUCTION TO FINANCIAL ECONOMETRICS.* Retrieved from Volatility Models: http://faculty.baruch.cuny.edu/smanzan/FINMETRICS/_book/index.html

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January 16). *Efficient Estimation of Word Representations in Vector Space.* Retrieved from Arxiv: https://arxiv.org/pdf/1301.3781.pdf

Narkhede, S. (2018, June 26). *Understanding AUC - ROC Curve*. Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Niebles, J. C., & Krishna, R. (2017). *Lecture: Visual Bag of Words.* Retrieved from Stanford: http://vision.stanford.edu/teaching/cs131_fall1718/files/14_BoW_bayes.pdf

Nielsen, M. (2019, December). *CHAPTER 5: Why are deep neural networks hard to train?* Retrieved from Neural Networks and Deep Learning: http://neuralnetworksanddeeplearning.com/chap5.html

Nybo, C. (2020, October 1). *Sector Volatility Prediction Performance Using GARCH Models and Artificial Neural Networks.* Retrieved from Arxiv: https://arxiv.org/ftp/arxiv/papers/2110/2110.09489.pdf

Olah, C. (2015, August 27). *Understanding LSTM Networks.* Retrieved from Colah's blog: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Pennington, J., Socher, R., & Manning , C. (2014, August). *GloVe: Global Vectors for Word Representation.* Retrieved from Stanford: https://nlp.stanford.edu/pubs/glove.pdf

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, February 15). *Deep contextualized word representations.* Retrieved from Arxiv: https://arxiv.org/pdf/1802.05365.pdf

Poon, S.-H., & Granger, C. (2003). Forecasting Volatility in Financial Markets: A Review. *Journal of Economic Literature* , 478-539.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving Language Understanding by Generative Pre-Training.* Retrieved from OpenAI: https://s3-uswest-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Schmidhuber, J., & Hochreiter, S. (1997). Long Short-Term Memory. *Neural Computation*, 1735–1780.

Schoonjans, F. (2017, January 4). *MedCalc manual: Easy-to-use statistical software.* Ostend: Independently published. Retrieved from MedCalc: https://www.medcalc.org/manual/roc-curves.php

SEC. (2013, April 2). *SEC Says Social Media OK for Company Announcements if Investors Are Alerted*. Retrieved from U.S. Securities and Exchange Commission: https://www.sec.gov/news/press-release/2013-2013-51htm

Sharma, A. (2022, July 19). *A Beginner's Guide to Exploratory Data Analysis (EDA) on Text Data (Amazon Case Study)*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/04/beginners-guide-exploratory-data-analysis-text-data/

Sharma, N. (2022, November 14). *Hugging Face Pre-trained Models: Find the Best One for Your Task*. Retrieved from neptune.ai: https://neptune.ai/blog/hugging-face-pre-trained-models-find-the-best

Tetlock, P. (2005, March 21). Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *Journal of Finance*. Retrieved from Giving Content to Investor Sentiment: The Role of Media in the Stock Market.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., . . . Polosukhin, I. (2017, June 12). *Attention Is All You Need.* Retrieved from Arxiv: https://arxiv.org/pdf/1706.03762.pdf

Wex Definitions Team. (2021, April). *Legal Information Institute*. Retrieved from Sarbanes-Oxley Act: https://www.law.cornell.edu/wex/sarbanes-oxley_act

Wikipedia. (2022, July 16). *Multinomial logistic regression*. Retrieved from Wikipedia : https://en.wikipedia.org/wiki/Multinomial_logistic_regression

Williams, S. (2022, March 17). *The 10 Most Popular Recurring Investments of Retail Investors*. Retrieved from Nasdaq: https://www.nasdaq.com/articles/the-10-most-popular-recurring-investments-of-retail-investors

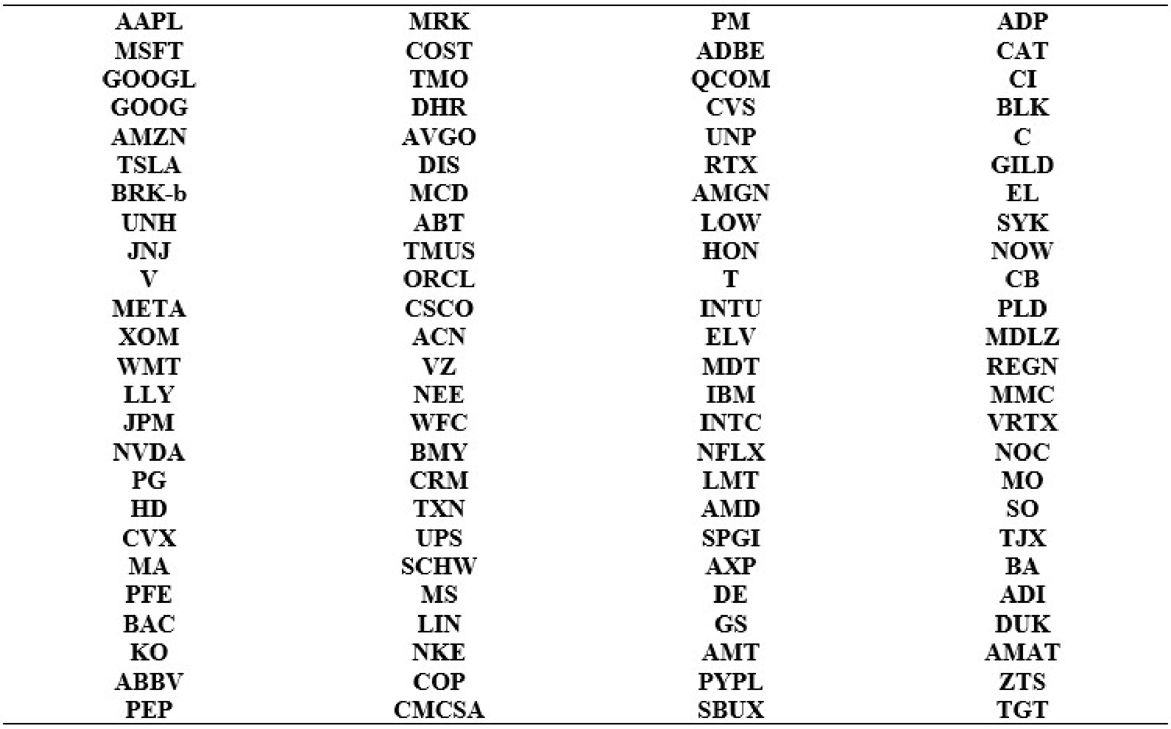
Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., & Macherey, W. (2016, September 26). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.* Retrieved from Arxiv: https://arxiv.org/pdf/1609.08144.pdf

Yalçın, O. (2020, December 12). *Sentiment Analysis in 10 Minutes with Rule-Based VADER and NLTK*. Retrieved from Towardsdatascience: https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-rule-based-vader-and-nltk-72067970fb71

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2017, August 9). *Recent Trends in Deep Learning Based.* Retrieved from Arxiv: https://arxiv.org/pdf/1708.02709.pdf%C2%A0

Appendix

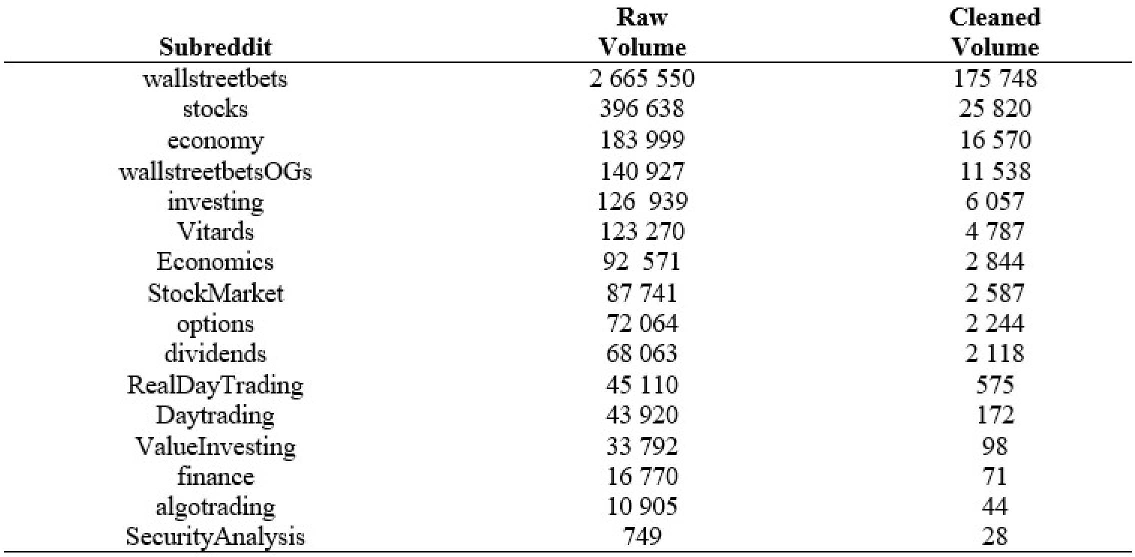
A.1: Ticker list



AAPL
MSFT
GOOGL
GOOG
AMZN
TSLA
BRK-b
UNH
JNJ
V
META
XOM
WMT
LLY
JPM
NVDA
PG
HD
CVX
MA
PFE
BAC
KO
ABBV
PEP
MRK
COST
TMO
DHR
AVGO
DIS
MCD
ABT
TMUS
ORCL
CSCO
ACN
VZ
NEE
WFC
BMY
CRM
TXN
UPS
SCHW
MS
LIN
NKE
COP
CMCSA
PM
ADBE
QCOM
CVS
UNP
RTX
AMGN
LOW
HON
T
INTU
ELV
MDT
IBM
INTC
NFLX
LMT
AMD
SPGI
AXP
DE
GS
AMT
PYPL
SBUX
ADP
CAT
CI
BLK
C
GILD
EL
SYK
NOW
CB
PLD
MDLZ
REGN
MMC
VRTX
NOC
MO
SO
TJX
BA
ADI
DUK
AMAT
ZTS
TGT

Appendix A 1: Overview of tickers

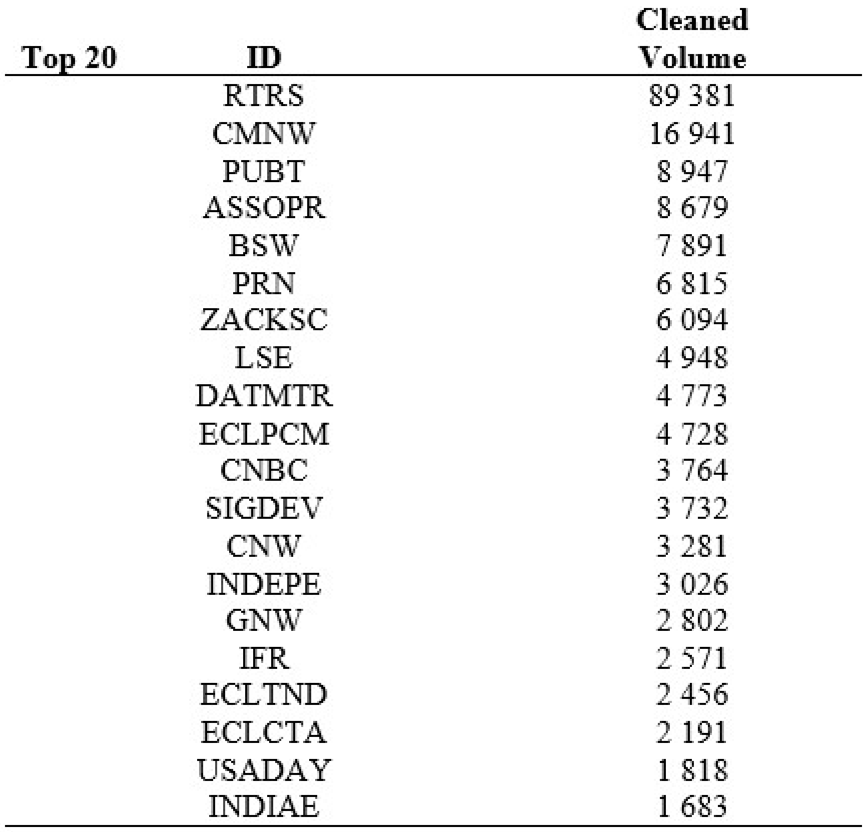
A.2: List of subreddits



Subreddit	Raw Volume	Cleaned Volume
wallstreetbets	2 665 550	175 748
stocks	396 638	25 820
economy	183 999	16 570
wallstreetbetsOGs	140 927	11 538
investing	126 939	6 057
Vitards	123 270	4 787
Economics	92 571	2 844
StockMarket	87 741	2 587
options	72 064	2 244
dividends	68 063	2 118
RealDayTrading	45 110	575
Daytrading	43 920	172
ValueInvesting	33 792	98
finance	16 770	71
algotrading	10 905	44
SecurityAnalysis	749	28

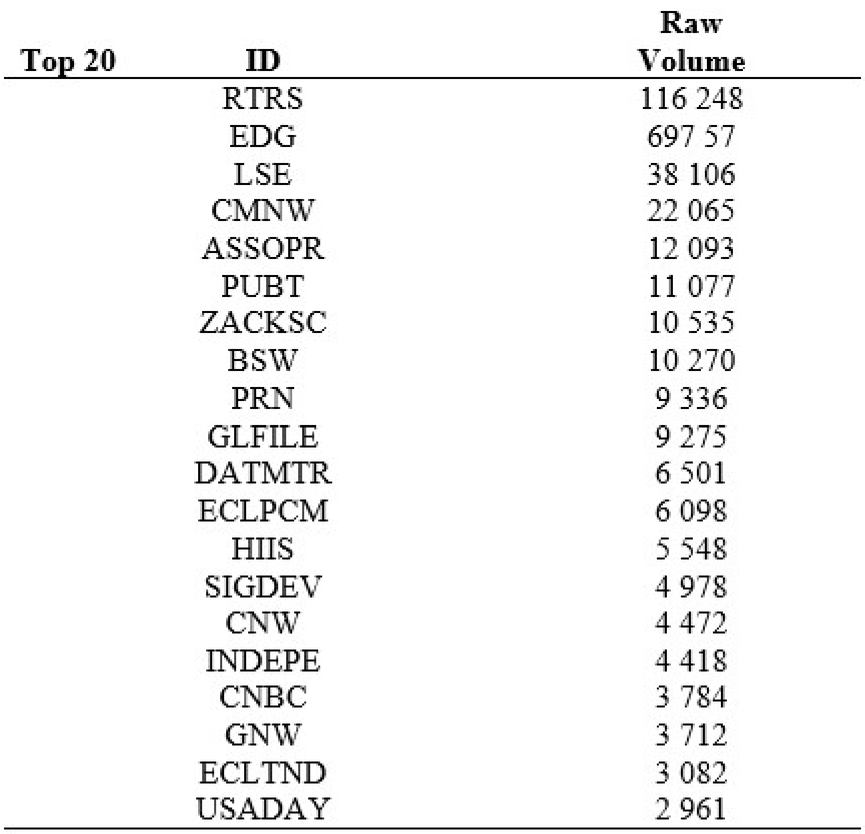
Appendix A 2: Overview of subreddits

A.3 & A.4: List of Newspapers



Top 20	ID	Cleaned Volume
	RTRS	89 381
	CMNW	16 941
	PUBT	8 947
	ASSOPR	8 679
	BSW	7 891
	PRN	6 815
	ZACKSC	6 094
	LSE	4 948
	DATMTR	4 773
	ECLPCM	4 728
	CNBC	3 764
	SIGDEV	3 732
	CNW	3 281
	INDEPE	3 026
	GNW	2 802
	IFR	2 571
	ECLTND	2 456
	ECLCTA	2 191
	USADAY	1 818
	INDIAE	1 683

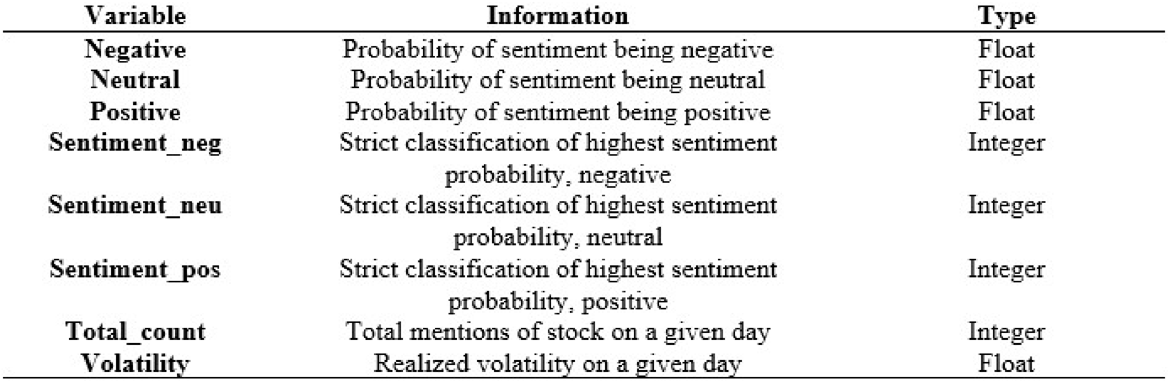
Appendix A 3: Overview of newspapers for cleaned data



Top 20	ID	Raw Volume
	RTRS	116 248
	EDG	697 57
	LSE	38 106
	CMNW	22 065
	ASSOPR	12 093
	PUBT	11 077
	ZACKSC	10 535
	BSW	10 270
	PRN	9 336
	GLFILE	9 275
	DATMTR	6 501
	ECLPCM	6 098
	HIIS	5 548
	SIGDEV	4 978
	CNW	4 472
	INDEPE	4 418
	CNBC	3 784
	GNW	3 712
	ECLTND	3 082
	USADAY	2 961

Appendix A 4: Overview of newspapers for raw data

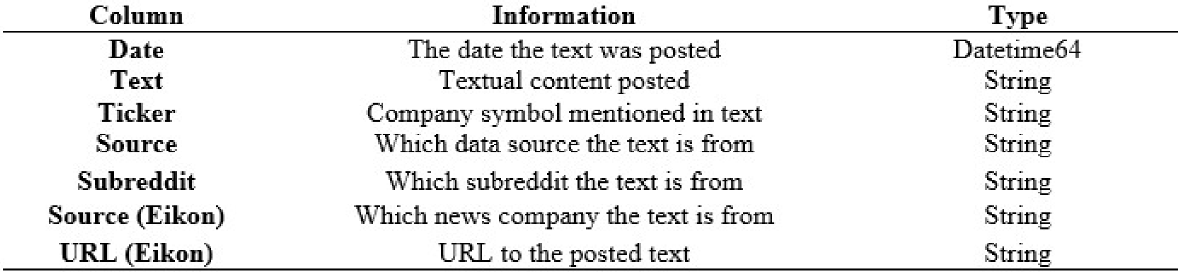
A.5: Variable description



Variable	Information	Type
Negative	Probability of sentiment being negative	Float
Neutral	Probability of sentiment being neutral	Float
Positive	Probability of sentiment being positive	Float
Sentiment_neg	Strict classification of highest sentiment probability, negative	Integer
Sentiment_neu	Strict classification of highest sentiment probability, neutral	Integer
Sentiment_pos	Strict classification of highest sentiment probability, positive	Integer
Total_count	Total mentions of stock on a given day	Integer
Volatility	Realized volatility on a given day	Float

Appendix A 5: Variable description of explanatory variables

A.6: Retrieved object description



Column	Information	Type
Date	The date the text was posted	Datetime64
Text	Textual content posted	String
Ticker	Company symbol mentioned in text	String
Source	Which data source the text is from	String
Subreddit	Which subreddit the text is from	String
Source (Eikon)	Which news company the text is from	String
URL (Eikon)	URL to the posted text	String

Appendix A 6: Column description for retrieved object