Machine Learning for Financial Market Forecasting

Machine Learning for Financial Market Forecasting

Jaya Johnson

A Thesis in the Field of Software Engineering

for the Degree of Master of Liberal Arts in Extension Studies

Harvard University

May 2023

Abstract

Stock market forecasting continues to be an active area of research. In recent years machine learning algorithms have been applied to achieve better predictions. Using natural language processing (NLP), contextual information from unstructured data including news feeds, analysts calls and other online content have been used as indicators to improve prediction rates. In this work we compare traditional machine learning methods with more recent ones, including LSTM and FinBERT to assess improvements, challenges and future directions.

Acknowledgements

I want to thank my thesis director and advisor Professor Hongming Wang, for her expertise, advice, encouragement, and understanding. Her guidance was invaluable, and her knowledge and patience helped provide me with an outstanding research experience. My thesis year was an absolute pleasure and a great learning adventure.

Thank you to all the Harvard professors and teaching assistants whose knowledge, professionalism, patience, and capabilities provided a tremendous learning opportunity for me.

On a personal note, I want to thank my family, Jayin and Josh, for their invaluable support and encouragement over the past few years. Jayin, my wonderful boy, there is no way I could have done this without you.

Contents

List of Figures

List of Tables

Chapter I.

Introduction

Stock market prediction continues to be one of the most significant challenges in research due to the volatile, non-parametric, and nonlinear data sets (Huynh et al., 2017). Earlier research has attempted to use various computational methods to model financial time series, starting with Neural Networks (Zhang & Wu, 2009), Fuzzy Systems (Moghaddam et al., 2016), Hidden Markov models (jae Kim & Han, 2000), and other hybrid combinations (Huynh et al., 2017). Various degree of success rates have been observed.

News articles and stock research have traditionally been important influences in market trends. Many of these studies have only recently started to include non-technical sources of information, such as news media, that impact the behavior of investors (Zhai et al., 2007). Recent developments in AI and Natural Language Processing (NLP), have made it possible to quantify their impact and apply them in models as feature sets (Zhai et al., 2007).

This thesis applies AI techniques that include historic price indicators and news articles to predict the direction of individual stocks and market indices. It

also provides exploratory research into how these techniques predict the stock/index market and what limitations these algorithms have.

1.1. Motivation

Textual data provides additional market insight and has improved stock market prediction. In particular, sentiment analysis is used significantly in many applications in finance, for instance, stock market prediction. It attempts to categorize the emotion in textual data as positive, negative, or neutral by identifying positive and negative words in their context (Ashtiani, 2018). Some examples of data sources are social media (Twitter, Reddit), news sites (Reuters, Bloomberg, Wall Street Journal), blogs, forums, and financial reports (SEC documents) (Mao et al., 2012).

Sophisticated language models have been introduced in the past few years. These transformer-based models enable the algorithm to accurately and effectively understand the context of a word (Vaswani et al., 2017). The bidirectional encoder representations from transformers (BERT) is a language model that considers the word in its entire context while learning to improve the word embedding models (Devlin et al., 2018).

One of the fundamental issues with NLP models for sentimental analysis is that these models often need the correct financial language and vocabulary in the training corpora. In most cases, the training corpora are more general (Araci, 2019). FinBERT resolves this issue; it is a pre-trained NLP model that analyzes the sentiment of the

financial text.

Using machine learning algorithms is a trend that can be attributed to the exponential growth in computing power, availability of such resources on the cloud, and advancement of algorithms that not only process vast amounts of data but continually address the problems with previously applied solutions. Different Support Vector Machine (SVM) models are proving to be effective and have been used in past research to predict the opening price of indices (Yang et al., 2022a).

Much recent research has focused on using long short-term memory (LSTM) neural networks to predict market trends (Zhai et al., 2007) (Lin & Chen, 2018) The characteristics of selective memory and maintaining an internal state are suitable for market predictions. Most of the experiments have been effective for short-term forecasting. In many instances, LSTM has performed better than other recurrent neural networks (RNNs). These experiments used technical indicators to predict index prices and direction, leading us to add fundamental data indicators to the models to evaluate their accuracy.

Technical and fundamental indicators are both used in market trend prediction, including stocks and indices. In stock market prediction, technical indicators have always been the primary source of feature selection (Zhai et al., 2007). Technical indicators computed using mathematical formulae and historical prices to analyze the patterns of past trends and predict future movements (Murphy, 1999). Fundamental indicators are external macroeconomic indicators that are political or economical to

make the prediction. Unstructured data from news articles, financial reports, and micro-blogs are most commonly used, to factor in the fundamental indicators (Murphy, 1999).

Some of the technical indicators used commonly are explained below.

Stochastic %K - %K represents the percentage of a security's price and the difference between the highest and lowest price over time.

Stochastic %D - is used to demonstrate the long term trend for current prices.

Momentum - Momentum is the rate at which the security's price changes.

Rate of change - Rate of change (ROC) refers to how fast price changes over time. It does not measure the magnitude of individual changes.

Williams %R - also known as the Williams Percent Range, ranges between 0 and -100 and is a momentum indicator.

The accumulation/distribution indicator (A/D) is a cumulative indicator using the volume and price to determine whether a security is distributed or accumulated.

Disparity - The disparity index is a technical indicator that compares the security's most recent closing price to the moving average. It is calculated as a percentage.

Figure 1: Summary of price features (Zhai et al., 2007)

Feature	Formula	Feature	Formula
Stochastic %K	$\frac{C_t - LL_n}{HH_n - LL_n}$	William's %R	$\frac{H_n - C_t}{H_n - L_n} \times 100$
Stochastic %D	$\frac{\sum_{i=0}^{n-1} \%K_{t-i}}{n}$	A/D Oscillator	$\frac{H_t - C_{t-1}}{H_t - L_t}$
Momentum	$C_t - C_{t-4}$	Disparity 5	$\frac{C_t}{MA_5} \times 100$
Rate of Change	$\frac{C_t}{C_{t-n}} \times 100$		

C_t is the closing price at day t, L_t is the lowest price at day t, H_t is the highest price at day t, MA_n is the moving average of the past n days, LL_n and HH_n is the lowest low and highest high in the past n days, respectively.

Zhai et al. evaluated the technical indicators described above to predict daily stock price trends (Zhai et al., 2007). Price momentum was one of the indicators that had the most impact on the price movement (Zhai et al., 2007). Prior research also finds news articles to have a significant impact on the behavior of stock prices (Vargas et al., 2017).

The various NLP techniques to extract a sentiment value from contextual data that impact market trends, as well as the promising AI techniques and algorithms applied to financial engineering research have motivated us to explore the impact of technical and fundamental indicators using machine learning to predict stock and index prices.

1.2. Related Work

This section reviews all related literature and summarizes the relevant content to our thesis. Considerable progress has been made in stock market research since applying various machine learning techniques improves the accuracy of predictions and performance.

The internet has contributed to vast online data, including various content in financial news data, research publications, and stock reviews. Manning et al. discuss how NLP has developed considerably, given the massive volume of contextual data, the highly advanced machine learning algorithms, and the availability of increased computing that gives a much richer understanding of language models in their context (Socher et al., 2012) .

News sentiment analysis using deep learning methodologies is an area with some exciting findings. Techniques such as word vectors and polarity based on both positive and negative sentiment analysis of articles have an impact on stock price changes and help predict future trends (Souma et al., 2019). However, NLP in the financial domain might need to be tailor-made to address a specific problem, according to Chen et al. Li et al. (2021). Many studies have been conducted to review the impact on market predictions. Li, Wang, et al.(Li et al., 2020) used the BERT model to obtain investor sentiment and study its impact on stock yield. They confirmed the relationship between the two (with 97% accuracy). RNNs including LSTM are deep learning models used more recently in stock market prediction. In their re-

search, Alexander, Härdle, et al. (Dautel et al., 2020b) compared different RNNs for predicting FX market movements. They evaluated traditional RNNs, feed-forward networks, and LSTMs for the performance of the time series data seemed to be best with LSTMs compared to the more traditional methods (Dautel et al., 2020b).

Below is a summary of the papers evaluated and their outcomes.

Paper - Reference	**Results Summary**		
	Algorithm	**Prediction Output**	**Results**
Chun Yang et al.	SVM	Shanghai Stock Exchange	RMSE 14.730
Huy D. Huynh et al.	BGRU	S&P 500	Accuracy 65%
Yuexin Mao et al.	Linear regression	S&P 500 Stock Indicators	Accuracy 68%
Tay et al.	SVM	Futures Contract Price	NMSE 1-1.3
Vargas et al.	RNN CNN	S&P 500 Price	64%
Lin et al.	LSTM	Stock Price	50%-60%
Huang et al.	LSTM	Chinese Stock Market	5 - 6 MSE
Torralba at al.	LSTM	Philippine Stock Exchange	20 - 30 %

1.2.1 NLP and Financial Market Prediction

Natural Language Processing (NLP) has been an increasingly important area of research in finance, as it harnesses unstructured financial data to add value to various domains of predictive analytics, especially stock market analysis. Stock news headlines, reviews, 10K filings, and earnings statements are different financial artifacts that can be valuable input features in stock market prediction. Sentiment analysis of news articles, and headlines, in particular, help make more informed trading decisions (Osterrieder, 2023).

Andrawos (Andrawos, 2022) has compared and analyzed textual data that

might help predict stock market pricing. They analyzed 10-K and 10-Q reports of 48 companies in the SP 500 from 2013 to 2017 and tested their report from 2018-2019. They concluded that contextual data improved the model's performance.

Villamil et al. describes a well-established behavioral finance relationship where new releases often impact stock prices (Villamil et al., 2023). They use bidirectional RNNs to reach an accuracy of 80.7%. Supporting this methodology, we review the paper by Wójcik et al. that examines the impact of financial statements on stock and foreign exchange markets. They use the tone/sentiment of these statements and measure the direction of the markets. They found non-linear methods to be more effective than linear methods. One of the methods used in this study was support vector regression using polynomial and radial kernels (Wójcik & Osowska, 2023).

In this thesis, we extract the sentiment of news articles from Reuters and analyze their impact on various models to predict the direction of the index. For the baseline methodology, DistilBERT is chosen as the library for sentiment analysis. Per SANH et al., who introduce DistilBERT, claim it to be a smaller, faster, cheaper, and lighter version of the full version of BERT (Sanh et al., 2019).

1.2.2 BERT and FinBERT

BERT is a pre-trained language model. BERT is first trained on a significant source of text, such as Wikipedia. This language model can be applied to other NLP tasks, a good example being sentiment analysis.

One of the significant areas for improvement of these pre-trained language

models is that they are trained with a general corpus, and analyzing financial text is complicated and does not yield the same results as non-financial text. The lack of context in financial language makes analyzing sentiment in financial text challenging.

Many studies have compared the advantages of using pre-trained FinBERT models with BERT. In their paper, Yang et al. compared BERT and FinBERT to run economic sentiment classification. They have compiled large-scale corpora that are representative of financial and business communications. Their corpora include Corporate Reports 10K & 10Q, Analyst Reports, and Earnings Calls Statements (Yang et al., 2020).

FinBERT has shown significant promise compared to BERT. For instance, Huang et al. compare several machine-learning algorithms with FinBERT with labeled sentences from financial reports (Huang et al., 2022). They conclude that FinBERT achieves higher accuracy than other approaches. Peng et al. have also compared BERT with FinBERT, specifically the performances of the two models using a variety of financial text processing tasks. FinBERT outperformed all models in the financial data sets. Thus, we are motivated to compare the sentiment score output of BERT and FinBERT as an additional input feature to the model (Peng et al., 2021).

Desola et al. look into more effective methodologies to extract information from 10-K reports. They concluded that BERT performed better on Earning Calls transcripts. The performance improvement indicated that BERT had better contex-

tualization than FinBERT (DeSola et al., 2019) .

Although many studies show significantly better results with the sentiments derived from FinBERT language models when classifying financial data, Chuang and Yang note that there are nuances in their preference towards certain domains, and they highlight this to help NLP practitioners improve the robustness of their models (Chuang & Yang, 2022). Overall FinBERT performs better on earnings calls versus 10-Ks.

Based on these studies, we used FinBERT for sentiment analysis and DistilBERT, a condensed form of BERT, as a baseline. This work could show the difference between the two language models BERT (trained on more general corpora) and FinBERT (trained on more specific corpora) and their impact on news headlines, various earnings calls and 10K statements.

1.2.3 Logistic Regression and Stock Market Prediction

Statistical methods have long been used for stock market predictions. Different techniques have been used to build models for stock predictions; logistic regression is the most commonly used classification model for smaller data sets.

Logistic regression is one of the baseline methodologies used for this work. It has long been used as a preliminary algorithm to predict indexes and stock prices. Yang et al. investigate correctly exploring and predicting the up and down trends for stock prices (Yang et al., 2022b) using look group penalized logistic regression to create a model with technical indicators. They use the confusion matrix and

AUC scores for bench-marking to improve prediction accuracy. In their work, logistic regression fails with many parameters with few samples. As the number of parameters in this thesis are limited to only a few, logistic regression was a suitable choice to baseline the study.

Ballings et al., also use single classifier models (Neural Networks, Logistic Regression, Support Vector Machines, and K-Nearest Neighbor) as baseline methods to compare their impact on predicting stock prices of 5767 stocks (Ballings et al., 2015).

1.2.4 SVM and Stock Market Prediction.

SVM is a supervised machine learning method used extensively in predicting stock market trends in the last few years. Many studies show improved results. Agusta et al. proposed a system to predict the optimal time to buy and sell stocks with SVM (Agusta et al., 2022). Yang et al. used it for stock market forecasting (Chuang & Yang, 2022). Kang et al. proposed using a hybrid Support Vector Machine (SVM) to forecast daily returns of popular stock indices in the world (Kang et al., 2023) . Achyutha, et al. analyzed the impact of different tweets and the performance of an organization (Achyutha et al., 2022).

One of the essential parameters to tune the SVM model is the type of kernel to be used. The above papers all had significant success with different kernel types. Yang leveraged the kernel function selection and kernel parameter selection to optimize the results (Chuang & Yang, 2022) . He used the linear kernel function to get the most

accurate results. Agusta et al. also leveraged the non-linear kernel functions to improve results (Agusta et al., 2022). The performance also increased when labeling the parameters was implemented with about 77% accuracy. For this study comparison of the different kernel methods and their effectiveness on prediction are to be tested.

SVM has proven to be better than traditional neural network methods. For instance, in their paper, Tay et al. look at the application of SVM in financial time series forecasting (Tay & Cao, 2001). They compare this methodology with a back-propagation neural network. They have examined five futures contracts from the Chicago Mercantile Market. Their experiments with SVMs found Gaussian kernel functions perform better than the polynomial kernel. They concluded that SVMs performed better than BP networks.

LSTM is the most recent neural network methodology used for time series prediction. Many studies are conducted to compare SVM with LSTM, with LSTM having better results on stock market prediction because of the nature of the data (time series). In his paper, Zhang has evaluated the accuracy of SVM and LSTM models to assess efficient markets hypothesis (EMH) by predicting the typical stock indexes of the American stock market and the Chinese stock market (Zhang et al., 2021). He concludes that running random walks shows little difference between LSTM and SVM models. In this study, we too compare SVM and LSTM but add another fundamental indicator, the sentiment of the articles, to see if that enhances the performance of the LSTM model.

1.2.5 LSTM and Stock Market Prediction.

Long short-term memory (LSTM) cells are used in recurrent neural networks (RNNs); these cells then learn to predict the future from sequences of different lengths. RNNs work with any sequential data.

LSTM models have been the most recent trend in deep learning algorithms with promise in stock market prediction and financial engineering. Koosha et al. have used and compared different machine learning models to predict the price of Bitcoin (Koosha et al., 2022) . The challenge that most pricing models face is the volatility of the data.

In their paper, Li et al. compare different machine learning models and develop their own. Their model is O-LGT, where the model's layers use LSTM and GRU (Li et al., 2023). Some researchers explore the volatile but profitable foreign exchange (FX) markets (Yıldırım et al., 2021). Dautel et al. (Dautel et al., 2020a) have also used LSTM in foreign exchange market predictions. Livieris et al. look at prediction models for gold prices as the volatility significantly impacts many world financial activities (Livieris et al., 2020) .

Although LSTM shows promise, tuning the parameters is an area that could be explored further in this study.

1.3. Research Problem

This project utilizes a combination of technical indicators such as market movement and fundamental indicators such as sentiment analysis of news articles to predict the direction of the S&P 500 index and stock prices. Based on prior research, index prices and stock prices are influenced by both technical and fundamental indicators.

The baseline for experiments is using traditional statistical methods including linear regression. BERT is used for sentiment classification. SVM is another baseline method used in the experiments. One of our goals (see below) is to determine if LSTM is better than SVM at predictions for time-series data in the stock market.

We also introduce sentiment scale derived from FinBERT as one of the input features to the LSTM model and compare and contrast the output and results.

Since short term news articles tend to have the most impact on the price of stocks, we go back a year to gather the data. Our data includes stock prices and news articles of 3 companies that influence the S&P 500 index. These companies are Microsoft, Apple, and Amazon.

This project aims to answer the following questions:

(i) How does LSTM perform compared to traditional machine learning methods such as SVM?

(ii) How does FinBERT compare with BERT methods to predict the direction of the S&P?

(iii) How does using FinBERT and technical indicators from prior research

impact the overall prediction of the movement of the S&P?

(iv) What is the impact of different parameters for fine tuning the algorithm?

Chapter II.

Methodology

2.1. Overview

This project aims to use a combination of fundamental and technical indicators to predict the direction of the S&P 500 index. Based on previous research, we use stocks of companies with the most weight to calculate the S&P 500 Index. These companies are Microsoft Corp. (MSFT), Apple Inc. (APPL), and Amazon (AMZN). Different statistical and machine learning algorithms are compared for accuracy and best fit.

Sentiments from news articles published on Reuters.com constitute the fundamental data. The historical price data of the stock tickers of the companies mentioned above constitute the technical indicators. These prices are downloaded from yahoo.com. NLP methodologies are used to analyze sentiment from news articles about stocks.

Once the sentiment score and class are extracted from the tone of the news articles, technical indicators such as moving averages of the closing prices are used as input variables for machine learning algorithms to predict the direction.

This section gives an overview of the different methodologies as well as the indicators used in the experiments to follow.

2.2. Financial Indicators

Financial indicators are statistics used to quantify current market conditions and forecast financial trends. In investing, indicators refer to technical chart patterns derived from a given security's price and volume. Indicators can be broadly categorized into fundamental indicators and technical indicators.

Fundamental indicators

These indicators are used to calculate the actual intrinsic value of a share. In order to do this, fundamental analysis looks at economic factors, known as fundamentals. These indicators are derived from the company's financial reports, reports about various macroeconomic indicators, and textual sources like news articles (Petrusheva & Jordanoski, nown). For the experiments conducted as a part of this thesis, NLP techniques like BERT and FinBERT are used to calculate the sentiment score from news articles, and the score is used as a fundamental indicator.

Technical indicators

These indicators are used to predict the future market price of a share. The technical analysis considers past changes in a share's price and attempts to predict its future price movements and changes (Petrusheva & Jordanoski, nown). This paper uses price momentum as the technical indicator to predict the S&P index.

Momentum is the rate of price changes in a stock, security, or tradable instrument. It shows the rate of change in price movement so investors can assess the market trends. For instance, a 10-day momentum line is calculated by subtracting the closing price ten days ago from the last closing price. (Murphy, 1999)

2.3. Data Characteristics

The proposed method for data collection is detailed below. This section is divided into downloading fundamental data, downloading technical data, and preprocessing the textual content to be able to run NLP models.

Fundamental Data such as stock, reviews are collected via a web extraction tool from Reuters (www.reuters.com). These include reviews in all categories. This data is collected for six months and one year. This data is further processed to extract sentiment scale and sentiment score from the news headlines.

Technical Data includes the six-month and one-year performance of the S&P performance data, including the index's high, low, open, close, and volume. This data is used to extract the price momentum, which is $C - C_t$ where C is the close price and t is the number of days (4). The data is downloaded from Yahoo finance.

2.4. Raw Data Extraction

The fundamental data or raw headlines required to analyze the movement of the index are extracted from reuters.com. Textual data (news articles) was collected

for six months and one year. The companies for which the data was collected included Apple Inc., Microsoft, and Amazon.

The web-services protocol from the network tab extracts information from the Reuters API. The web-service call is parameterized to download articles for different companies. The library utilized to develop the module for web extraction is Scrapy. Scrapy is a python-based fast web extraction framework useful for extracting unstructured content from websites. It has many use cases in the field of data mining.

Scrapy has many powerful features that make it the right choice for extracting raw content (articles).

i) Built-in support for selecting and extracting HTML data using regular expressions. Support to make rest-api calls with dynamic parameters.

ii) An interactive shell console that makes it easy to install the library in colab notebooks and run the spiders (modules).

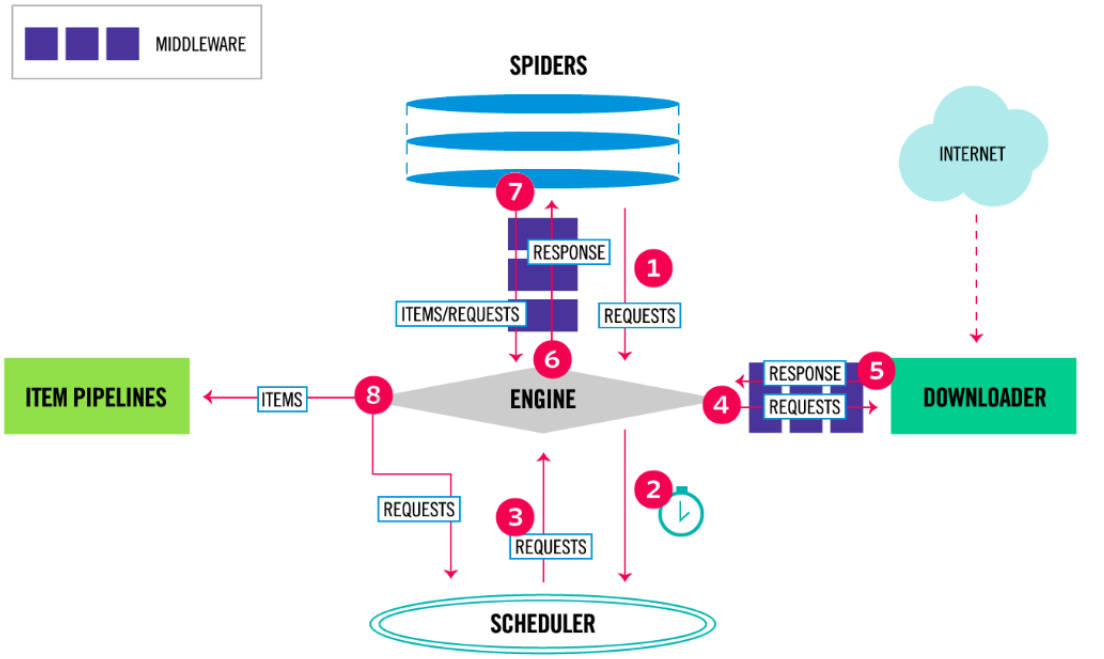
iii) Json support that enables run-time validations and basic parsing when downloading the articles and checking the return status code, and downloading select attributes.

iv) Extensible APIs that support module development, reducing the need to build pipelines from scratch. These APIs automate 90% of the orchestration when downloading the articles.

The execution engine controls the data flow in Scrapy as follows:

i) The Spider sends the initial request to the engine.

Figure 2: Scrapy Architecture.



ii) The engine schedules the requests in the order they are received.

iii) The requests are sent to the downloader via the pipeline from the engine.

iv) The downloader generates a response after downloading the page and returns it to the engine.

v) The engine sends the Spider the response via the spider middleware. The Spider processes this response.

vi) The Spider processes the response and returns content and other responses to the engine.

vii) The processed items are sent to the items pipelines, which send processed requests to the schedule where it gets the following requests.

The process repeats until there are no more requests (Zyte, 2023).

The technical data for the experiments is downloaded from Yahoo Finance. This data is exported manually into CSV files. The data points include open, close, high, and low prices.

2.5. Data Normalizing and Cleansing

Pre-processing raw textual data is a key step in NLP (natural language processing). The text identified at this stage are fundamental in NLP modeling. Normalizing is a series of steps where text documents are processed and cleansed. Data cleansing includes removing specials characters, formatting and stop words.

The fundamental data with the reviews is cleansed and persisted into a data frame with the following columns stock, date, review, and headline.

The data is pre-processed using the following steps:

i) Removal of extra spaces, punctuation, lines, and URLs.

ii) Converting data into all lowercase.

iii) Tokenization The text is split into smaller tokens using sentence and word tokenization.

iv) Stop word removal; common words are removed from the text as they do not add meaning to the analysis. Stop words usually do not add to the context of the text processed; hence they are typically eliminated from the text.

v) Perform stemming. The words in the text are stemmed or reduced to their root/base form.

vi) Perform lemmatization. It performs stemming of the word but ensures it retains meaning (Steven Bird & Loper, 2023)

The following libraries are used for text pre-processing.

Step	Library
Reg-ex cleaning	Python Reg-ex library
Tokenization	Python Tokenize library
Removal of stop words	NLTK library
Stemming	NLTK library
Lemmatization	NLTK library

2.6. BERT

BERT stands for Bidirectional Encoder Representations from Transformers. BERT is a transformer-based model. In this model each output element is connected to input elements therefore the model is able to add context by assigning weights to these elements based on their connection to each other.

This model has become a widely-used deep learning framework for natural language processing (NLP). The advantage BERT has over traditional NLP models is the ability add context to language in text from its surrounding text using the weights mentioned between the input and output elements mentioned above. This is also know as attention in NLP.

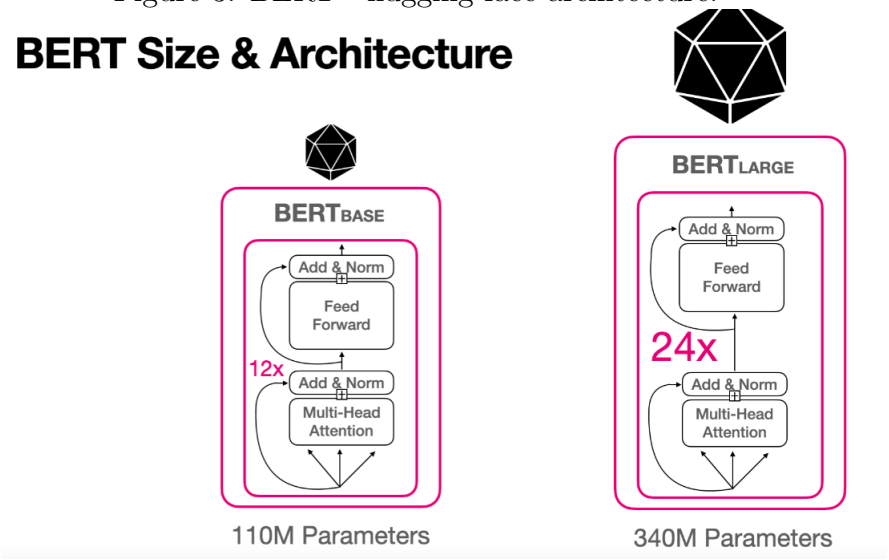
BERT was trained using text from Wikipedia - it can be tuned further using

question and answer data sets.

Language models preceding BERT could only read text sequentially however with BERT reads in both directions at once. This was enabled by Transformers which added bidirectionally to the models making them more powerful.

BERT is trained on the following NLP tasks: Masked Language Modeling and Next Sentence Prediction. Masked Language Model training obfuscates a word in a sentence and has the program predict that word. Next Sentence Prediction training creates a relationship between sentences. This process determines whether the relationship is logical or random. (AI, 2023)

Figure 3: BERT - hugging face architecture.



2.6.1 DistilBERT Parameter Tuning

For the experiments, we use a condensed version of BERT called DistilBERT from hugging face. DistilBERT is a condensed Transformer model trained by distilling

a BERT base. It has 40% fewer parameters, runs 60% faster, and preserves over 95% of BERT's performances measured on the GLUE language understanding benchmark.

The following are the parameters used to fine tune the model.

Parameter	Description/Values
vocab_size	Vocabulary size of the DistilBERT model.
max_position_embeddings	The maximum sequence length
sinusoidal_pos_embds	Whether to use sinusoidal positional embeddings.
n_layers	Number of layers that are hidden.
n_heads	Each attention layer, has attention heads that are set with this parameter.
dim	Dimensionality of the encoder layers and the pooler layer.
hidden_dim	The size of the "intermediate" (often named feedforward) layer in the Transformer encoder.
dropout	The probability for the dropout number.
activation	The activation function to be used.

Table 1: Hyperparameter options for DistilBERT

2.7. FinBERT

FinBERT is a domain-specific pre-trained language model based on Google's BERT trained on financial terms. The main driver for FinBERT is that BERT is only trained on general corpus and computes sentiment poorly for financial text.

FinBERT is a language model that adapts to the finance domain. (Huang et al., 2020). Huang et al. document that FinBERT outperforms many other NLP models and machine learning algorithms in sentiment classification.

FinBERT's main advantage is Google's original bidirectional encoder representations from transformers (BERT) model. The transformers are essential for small training sample sizes and in financial texts.

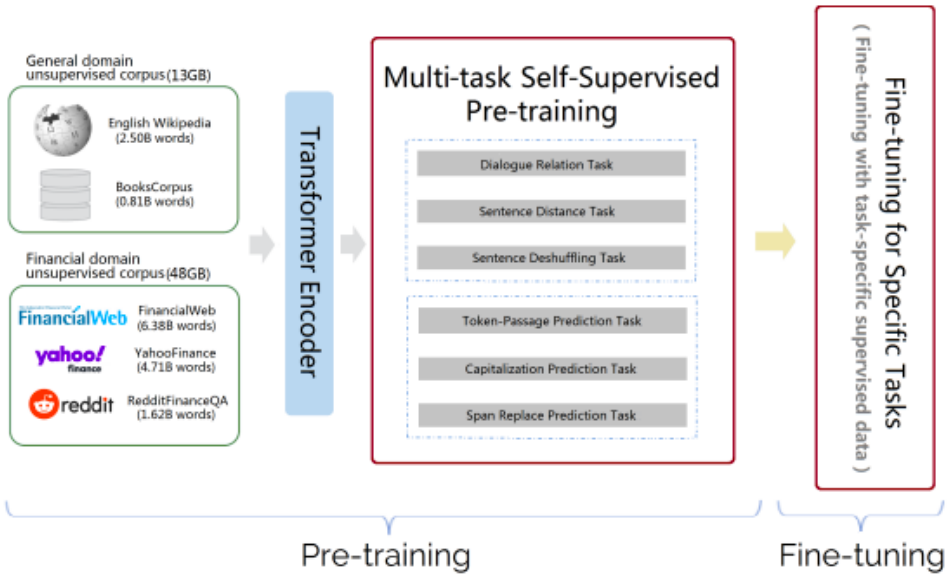
To train FinBERT, the base BERT model is pre-trained using three types of financial texts:

i) 60, 490, 10-Ks and 142, 622 10-Qs of Russell 3000 firms from the SEC's EDGAR website.

ii) S&P 500 firms' analyst reports from the Thomson Investext database; and

iii) earnings call transcripts (AI, 2023).

Figure 4: FinBERT - An illustration of the architecture for FinBERT. (Liu et al., 2021)



The above figure shows the architecture of FinBERT, the model is trained simultaneously on a general corpus and financial domain corpus.

2.8. Multiple Logistic Regression

Multiple Logistic Regression predicts a binary variable using one or more input variables. It is used to determine the numerical relationship between a set of variables. The variable to be predicted in that case is categorical.

Logistic regression allows us to model log(odds of rise in the index value) as a function of the sentiment score and market momentum (Maindonald & Braun, nown).

Multiple Logistic Regression is used in the experiments to follow two input variables (BERT score, Stock Movement) in the prediction of another categorical variable, which is the direction of the S&P for the experiments. Since the categories are up (1) and down (0), it is a binary classification.

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

2.8.1 Logistic Regression Parameter Tuning

For the experiments the sklearn library's module logistic regression is used. The following parameters provided by the library are tuned to optimize the model. (Pedregosa et al., 2011)

Parameter	Description/Values
Solver	'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga', default='lbfgs'
Penalty	regularization options are: 'l1', 'l2', 'elasticnet', 'none'
C	is a positive float to regulate over-fitting.

Table 2: Hyperparameter options for Logistic Regression

2.9. Support Vector Machines

A support vector machine (SVM) is a powerful and versatile machine learning model. It can perform linear and nonlinear classifications, regression, and even pattern detection. SVMs perform best for classification tasks with medium-sized nonlinear data sets. They need to scale better to massive data sets.

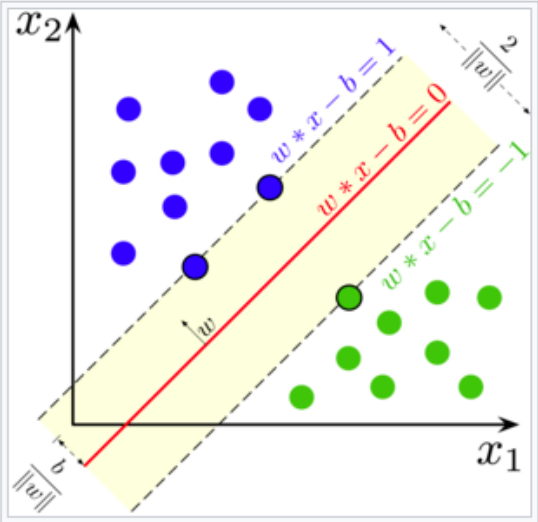
Kim (Kim, 2003) find that SVMs show promise in predicting financial time-series data. They have found this to be effective in predicting the stock price index.

Identifying the right hyperplane can get challenging in the real world. SVMs are based on finding a hyperplane that divides the data into classes. These vectors are collection of data points that are closest to the hyperplane.

Fortunately, when using SVMs, one can apply the kernel trick. The kernel allows one to add a third dimension to capture non-linear use cases. SVM constructs hyperplanes in a higher dimensional space for classifying data. The optimal hyperplane is the one with the largest distance to the training data point of any class.

The SVM has different Kernel functions that can be specified. Standard ker-

Figure 5: Maximum-margin hyperplane for an SVM. Samples on the margin are called the support vectors.



nels are included, but it is possible to customize kernels (Gron, 2017).

The kernel function can be any of the following:

i) Linear - The linear kernel is the dot product of te two vectors/kernels.

ii) Polynomial - It consists of the similarity of vectors in the training data set over polynomials of the original variables.

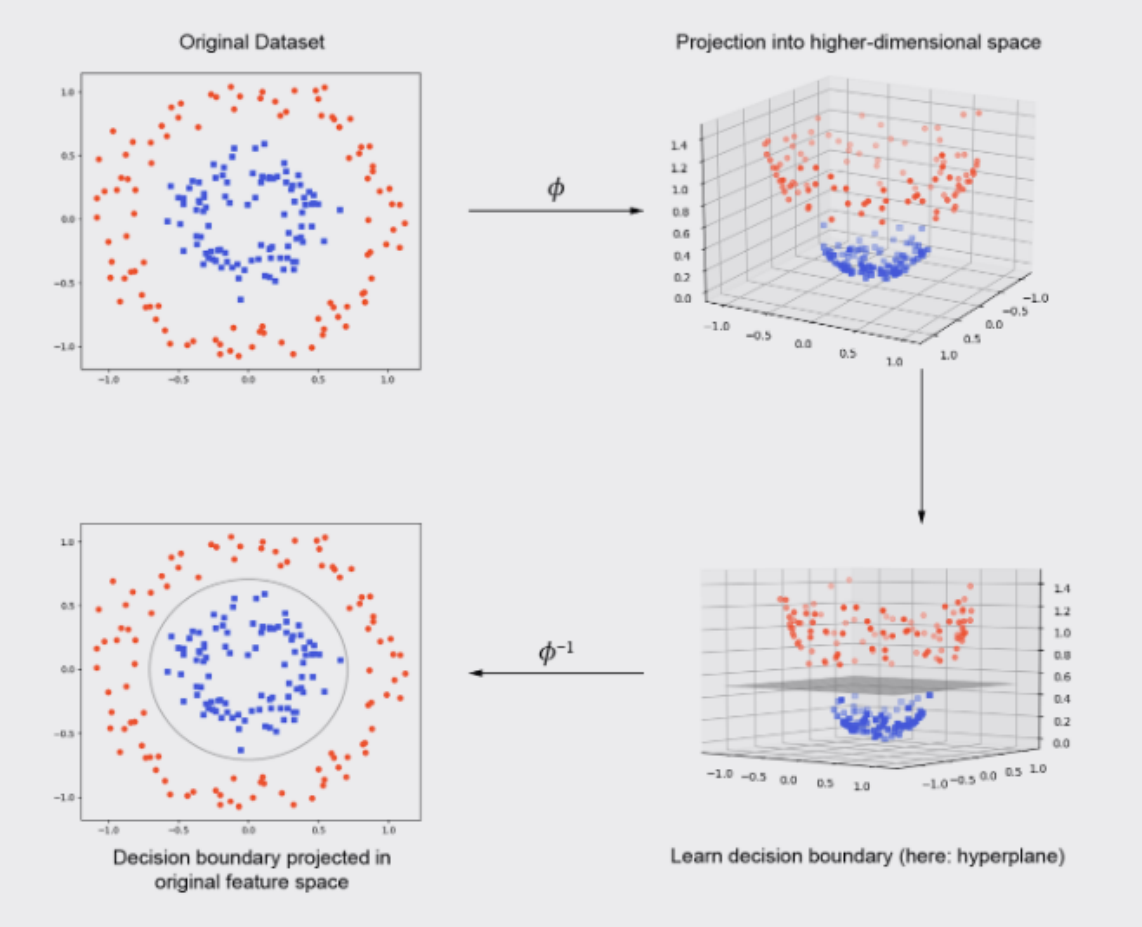
iii) Rbf - It introduces a radial basis method to improve the results.

iv) Sigmoid - This equates to a two layer, perceptron neural network. This model is used as an activation function for artificial neurons (Geron, 2019).

2.9.1 SVM Parameter Tuning

For the experiments below we use scikit learn's SVM module. The following tuning can be used to optimize the model.

Figure 6: The process of classifying nonlinear data using kernel methods (Raschka et al., 2022)



Hyper-parameter	Description
C	is the Regularization parameter.
kernel	'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'.
gamma	'scale', 'auto' or float
probability	Whether to enable probability estimates.

Table 3: SVM Hyper-parameter tuning (Pedregosa et al., 2011)

Figure 7: Kernel functions for SVMs

$$\begin{aligned}
\text{Linear:} \quad & K(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b} \\
\text{Polynomial:} \quad & K(\mathbf{a}, \mathbf{b}) = (\gamma \mathbf{a}^\top \mathbf{b} + r)^d \\
\text{Gaussian RBF:} \quad & K(\mathbf{a}, \mathbf{b}) = \exp\left(-\gamma \|\mathbf{a} - \mathbf{b}\|^2\right) \\
\text{Sigmoid:} \quad & K(\mathbf{a}, \mathbf{b}) = \tanh\left(\gamma \mathbf{a}^\top \mathbf{b} + r\right)
\end{aligned}$$

2.10. Long Short Term Memory (LSTM)

LSTM is a type of RNN. Recurrent Neural Networks (RNNs) are neural networks that specialize in processing temporal data. In an RNN, the output of the recurrent cell in the neural network in a previous step is used to determine the output in the current step. The output of these networks is used as input for the next steps. This process is repeated and resembles a feed-forward network with multiple hidden layers. The errors are calculated for each time step, and the weights are updated. This process is called back-propagation through time (BPTT) (Pajankar & Joshi, 2022). Some of the advantages of RNNs are:

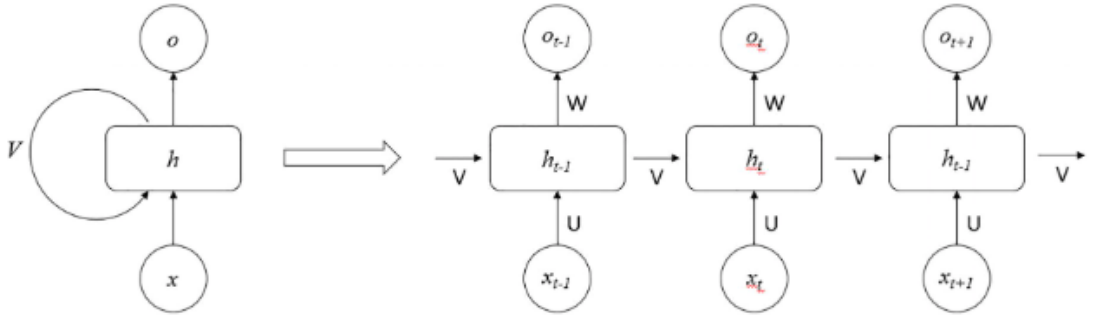
i) Handle variable-length sequences

ii) Keep track of long-term dependencies

iii) Maintain information about the order as opposed to FFNN

iv) Share parameters across the network

Figure 8: RNN architecture (Protopapas et al., 2023)



Unfolding of an RNN cell across time steps

The portion on the left shows that the input x passed to the recurrent unit cell applied with the weights W leads to the output o. When this recursion is unfolded over time as shown in the portion on the right, you can see that there are three time steps, each with inputs x_{t-1}, x_t, and x_{t+1}, respectively. At step t, the operation that occurs in the hidden layer can be expressed as

$$h^{(t)} = f\left(h^{(t-1)}, x^{(t)}; W\right)$$

However, one of the limitations of RNN is that RNNs cannot propagate the gradients that tend to become zero (or, in some cases, infinite); this is known as the vanishing gradient problem. LSTMs resolve this issue.

LSTMs maintain a cell state that can be referenced as a memory. With the cell having a memory, information from earlier time steps can be transferred to the later time steps, thus preserving long-term dependencies in sequences. LSTM unit contains a cell, an input gate, an output gate, and a forget gate. These gates address the vanishing gradient problem common in ordinary recurrent neural networks with maintaining state dynamics.

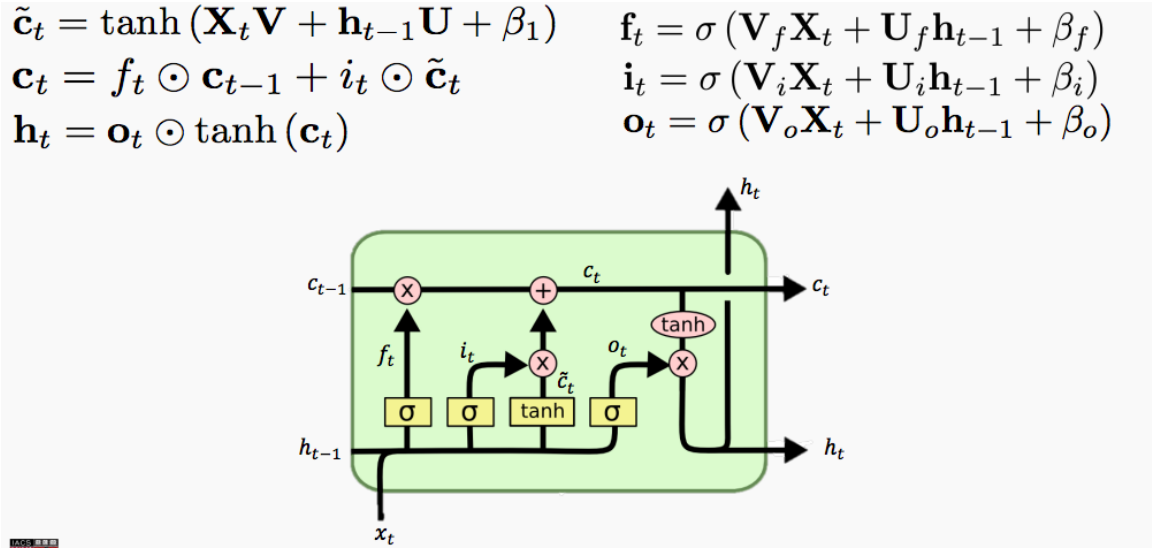
The input goes through the tanh function to determine the candidate and a

sigmoid function to determine the evaluation function. Both these outputs are then multiplied in what is called the input gate.

The new output is generated by the forget gate, which provides a new cell state. It consists of a vector multiplied by the previous cell state to make the values that are not relevant zero.

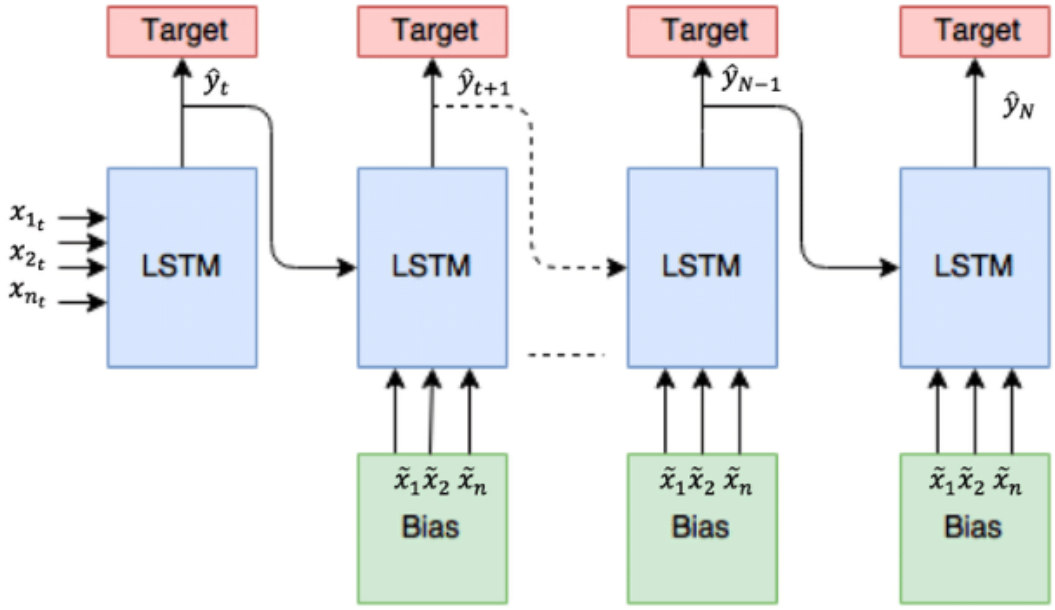
The output gate combines the cells in the last two gates and the computed cell state. The tanh function is applied to this state to provide the following cell state (Pajankar & Joshi, 2022).

Figure 9: LSTM Equation Explained (Protopapas et al., 2023)



The attention mechanism overcomes the issue with long term memory.

Figure 10: Multivariate LSTM model (Ismail et al., 2018)



2.11. Hyper-parameter Tuning

Hyper-parameter tuning requires many permutations of passing different parameters to the model to find the optimal ones that increase accuracy and give us the best results. Tuning the parameters can be cumbersome, so we used third-party libraries with various techniques for finding the optimal parameter values. A cross-validation technique where the model, and the parameters, are entered. After the best parameter values are determined, predictions are made.

GridSearchCV It is a library from sklearn that does an exhaustive review of parameter values for an estimator. GridSearchCV implements a "fit" and a "score" method. It also implements "score_samples", "predict", "predict_proba",

"decision_function", "transform" and "inverse_transform". The parameters of the estimator used to apply these methods use cross-validated grid-search over a parameter grid for optimal selection.

This is the tool to perform hyper-parameter tuning for Logistic Regression and SVM models.

Tuner The Keras Tuner is used to tune the parameters for the TensorFlow LSTM model. It selects the optimal set of hyper-parameters to tune the model.

Hyper-parameters are the parameters used by the ML model in the training process. Hyper-parameters are of two types:

i) Model hyper-parameters related to the model, such as input and hidden layers.

ii) Algorithm hyperparameters that impact the performance of the learning algorithm.

2.12. Model Evaluation

To compare the results of the experiments with different machine learning algorithms, the most common measure is calculating the model's accuracy, which involves comparing the actual prediction to the real value. Below are some of the techniques that will be used to compare the performance of the models.

Confusion Matrix

In statistical classification and machine learning, we use a confusion matrix to

evaluate the performance of the classification algorithm. The confusion matrix is a table that contains the performance of the model and is described as follows:

- The columns represent the instances that belong to a predicted class.

- The rows refer to the instances that belong to that class (ground truth).

Confusion matrices' configuration allows the user to quickly spot the areas in which the model has greater difficulty. (Saleh & Sen, 2019)

The sklearn library computes a matrix to evaluate the accuracy of classification. By definition, a confusion matrix C is such that $C_{x,y}$ is the number of observations known to be in group x and predicted to be in group y.

Therefore in binary classification, true negatives = $C_{0,0}$, false negatives = $C_{1,0}$, true positives = $C_{1,1}$ and false positives = $C_{0,1}$.

The two dimensions in the table are "actual" and "predicted."

From the matrix we get the following information: Tay & Cao (2001)

true positive (TP) This correctly denotes the presence of a characteristic.

true negative (TN) This correctly denotes the absence of a characteristic.

false positive (FP) This incorrectly denotes that a particular characteristic is present.

false negative (FN) his incorrectly denotes that a particular characteristic is absent.

The sklearn library's module metrics is used to calculate the following (Pedregosa et al., 2011):

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\textbf{Precision} = \frac{TP}{TP+FP}$$

$$\textbf{Recall} = \frac{TP}{TP+FN}$$

$$\textbf{F1} = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$$

ROC Curve

The ROC curve considers all possible thresholds for a given classifier. It displays the false positive rate (FPR) against the true positive rate (TPR). For the ROC curve, the ideal curve is towards the top left: The classifier that produces a high recall while keeping a low false positive rate is the one that performs well. (Muller & Guido, 2018)

We use the roc_curve and the auc_roc_score function from the sklearns library for the experiments. Its purpose is to compute ROC Curve and Area Under the Receiver Operating Characteristic Curve (AUC) from prediction scores. AUC Score is a measure of if the model is imbalanced or not (Pedregosa et al., 2011)

Chapter III.

Experiments and Results

3.1. System Configuration

Below is the list of system resources and software packages used in the experiments.

Computation	
Processors	GPU and TPU
Cloud Platform	Google Colab Pro
Software	
Language	Python 3.9
Data Processing	NLTK libraries
Web Extraction	Scrapy 2.8
SVM, Logistic Regression, LSTM	scikit-learn 1.2 , Keras 2.12.0
DistilBERT, FinBERT	huggingface

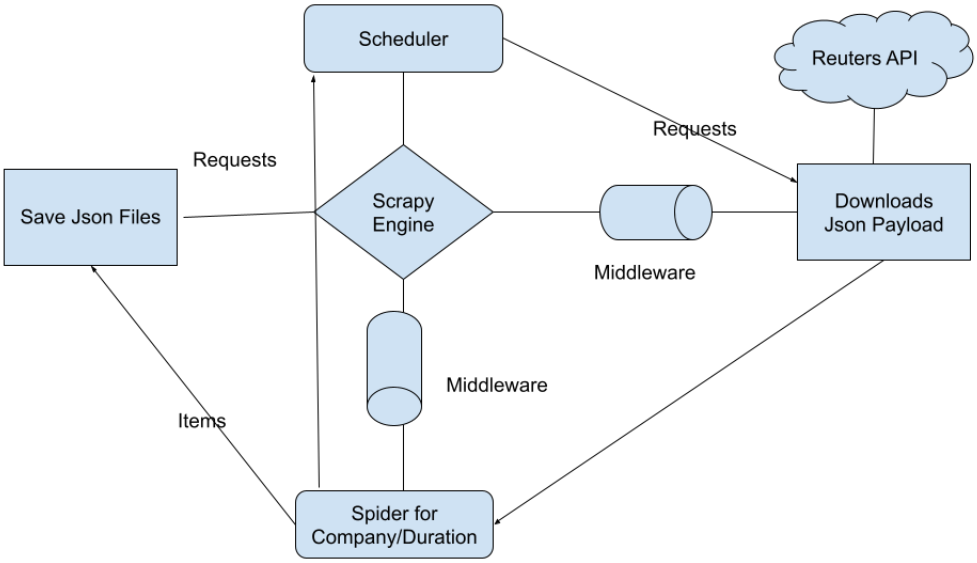
3.2. Data Collection and Analysis

One of the biggest challenges in data collection was downloading news articles from dynamic web pages. Most web article aggregators only provide six months of data that is available at no cost. The callback in the network tab that would allow us to get one year of data was leveraged.

Scrapy, a python web extraction framework was used to develop modules also known as spiders, that make the asynchronous callbacks to the Rest APIs and download the results based on the status code.

Spider Script Details	
URL	https://www.reuters.com/site-search/?
Parameters	Ticker/Company, Offset, StartDate, EndDate, of Rows.
List of Companies	Microsoft, Amazon, Apple.
Performance	6 Months Data (30 secs per ticker) 1 Year Data (2 mins per ticker)
Output Data Format	Json

Figure 11: Scrapy function call.



3.3. Data Analysis

There is two types of data that are used for the experiments.

i) Price Data. ii) Headlines from Reuters.

3.3.1 Price Data

List of companies for which Price Data is available are Microsoft, Apple, Amazon and the S&P 500 Index.

The following fields are downloaded.

Open	Close	High	Low

Price charts from left to right, Row 1. AMZN, MSFT. Row 2. APPL, S&P

Observations: As expected, the price data is random time series. The nature of the data is essential for the input features like price data. The SP prices will be processed to determine an increase or decrease from the previous day's price. The output variable is the increase/decrease in the S&P index price, which makes it binary. However, if the trends are observed, the general direction of the stocks seem to be in line with the S&P 500.

3.3.2 News Content

Articles from reuters.com are downloaded and parsed to extract the following fields:

Title	Description	Author	Publication Date	Article URL

As the data is from the JSON payload, there is no need for data cleansing. Data exploration is essential as reviewing the data content manually could be cum-

bersome. Various tools make it possible to visualize the data and interpret patterns to understand the data set better.

The following libraries are used for data exploration.

pandas	matplotlib
numpy	nltk
seaborn	wordcloud
textblob	spacy
textstat	

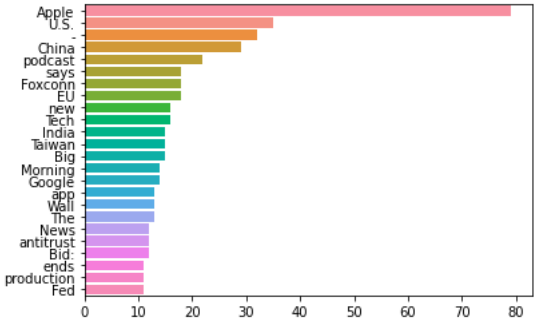
First, the raw data is analyzed to ensure there are no empty rows the format of the date time stamp, to review the article headlines.

Figure 12: Apple Inc. Raw data.

index	published_ts	description
0	2022-12-01 21:41:38.969000+00:00	The parent company of social media platform Parler and American rapper Kanye West, who now goes by Ye, have agreed to terminate the intent of the sale of Parler, according to a statement from Parlement Technologies on Thursday.
1	2022-12-01 23:15:58.870000+00:00	Apple TV+'s 2022 slavery drama "Emancipation", actor Will Smith's first film since his famous slap of comedian Chris Rock on stage at the Oscars, has received mixed early reviews from film critics.
2	2022-12-01 17:17:52.773000+00:00	Coinbase Global Inc said on Thursday customers using Apple Inc's iOS will not be able to send non-fungible tokens (NFTs) on the cryptocurrency exchange's wallet anymore.
3	2022-12-01 05:46:50.094000+00:00	Taiwan Semiconductor Manufacturing Co will offer advanced 4-nanometer chips when its new $12-billion plant in Arizona opens in 2024, spurred by U.S. customers including Apple Inc to do so, Bloomberg News reported on Thursday.
4	2022-11-24 12:34:15.634000+00:00	Manchester United's owners, the Glazer family, are considering selling the club as they explore "strategic alternatives". If the Glazers decided to follow through with the sale of the Old Trafford club, here could be some of the possible buyers:

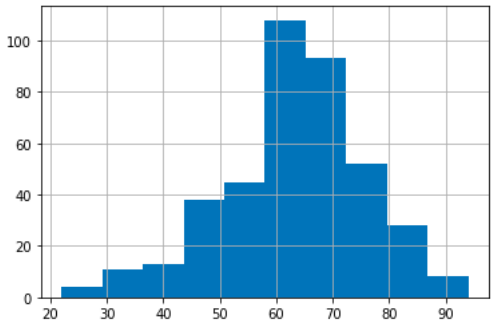
Next, the word distribution, character count, and word count were reviewed to analyze the content of the articles.

Figure 13: Apple Inc. Word Distribution.



These tools help us determine how many of the headlines contains the company searched. While observing the raw data, few headlines mentioned the company name. The company name appears in the detailed description or the article. In all cases, the content explored was accurate.

Figure 14: Apple Inc. Character Distribution.



The character distribution is between 30 and 90 characters for all articles, with a deviation of +- 10 words. The distribution is an important metric when running sentiment analysis, as the process is computationally expensive and uses heavy system resources. For articles $>$ 150 characters, the FinBERT classification would need to be batched into 150-200 headlines so we would have enough memory. For articles $<$ 100 chars, the headlines would be batched into batches of 250-300 to be classified.

DistilBERT is more efficient than FinBERT, which processed 20K rows of 80-100 character headlines in 45 minutes without significantly utilizing the system resources.

Words clouds help review the distribution of bi-grams and tri-grams to ensure enough context since BERT and FinBERT rely extensively on context. It is evident from the word cloud that a fair amount of financial terminology is used. If there was not a good distribution of the financial keywords, additional filters might need to be implemented, e.g., downloading only business and market sections.

The data is ready for sentiment analysis with satisfactory data analysis for all the companies. Failing that, one would have to review the sources, meta-tags, and search keywords and repeat the download process to ensure the correct data set for sentiment analysis is obtained.

3.4. Sentiment Analysis

BERT is used for the baseline analysis of the article sentiment. BERT and other transformer encoder architectures have been widely successful on a variety of NLP projects.

Experimental Setup

List of companies evaluated	Microsoft, Apple, Amazon.
Library model to perform sentiment analysis	Hugging Fact DistilBERT.
Labels/Classification classes	POSITIVE, NEGATIVE.
Pre-Trained Tokenizer	distilbert-base-uncased-finetuned-sst-2-english.
Processing Time	Avg: 30 mins for 300 articles using TPUs. 45 mins for 1200 articles using TPUs.

Below is the breakdown of the positive/negative sentiment analysis classified by the pre-trained DistilBERT model.

Sentiment	MSFT	AMZN	AMAZON
Negative	288	115	310
Positive	112	24	90

Table 4: Sentiment breakdown for different companies - 6 months

Sentiment	MSFT	AMZN	APPL
Negative	1111	572	1187
Positive	450	226	411

Table 5: Sentiment breakdown for different companies - 1 year

A sample of the classified sentiment from the model is shown below.

Microsoft's headlines with positive sentiment

index	Headline	sentiment
28	Microsoft powers up search for Chinese gaming hits in race against Sony	POSITIVE
34	Exclusive: Microsoft seeks to settle EU antitrust concerns over Teams -sources	POSITIVE
37	Microsoft soothes market fears with forecast for strong revenue growth	POSITIVE
59	Cisco partners with Microsoft to add Teams to its meeting devices	POSITIVE
72	Microsoft to bring Xbox games to Samsung's 2022 smart TVs	POSITIVE
96	Microsoft cloud outage hits users around the world	POSITIVE
112	Union says Microsoft will recognize unit of videogame testers	POSITIVE
136	Sony to expand Chinese game incubator in Microsoft head-to-head	POSITIVE

Microsoft's headlines with negative sentiment

9	Sony's gaming chief met EU's Vestager on Microsoft's Activision deal -source	NEGATIVE
15	Video gamers fight Microsoft bid to block lawsuit over Activision deal	NEGATIVE
16	Lockheed gets Microsoft classified cloud to speed work with Pentagon	NEGATIVE
21	Microsoft tells judges its $69 bln Activision deal would benefit gamers	NEGATIVE
33	Microsoft's president warns of talent shortage for tackling climate change	NEGATIVE
39	Microsoft seeks to dodge EU cloud computing probe with changes	NEGATIVE
47	Pentagon splits $9 billion cloud contract among Google, Amazon, Oracle and Microsoft	NEGATIVE
52	Microsoft inks Nvidia game deal to assuage regulators over Activision merger	NEGATIVE

Observations

i) The "Negative" sentiment of the news articles for all three companies indicated the sluggish market in the past year (2022-2023).

ii) As demonstrated in the tables above, some articles can fall into the "Neutral" category and make their way into the "Positive" and "Negative" categories.

iii) The sentiment distribution of the three companies was similar, where there were more "Negative" sentiments than "Positive." However, this lines up with the trend of the markets, which is mostly downwards.

iv) Evaluating pre-trained models was time-consuming, running multiple iterations, each taking 30-45 minutes. The most performant model was *"distilbert-base-uncased-finetuned-sst-2-english."*

v) There was a need for a third neutral category based on the output of the sentiments that seemed to be more neutral.

3.5. FinBERT - Financial News Sentiment Analysis

This section explains the sentiment analysis running FinBERT, a financial news sentiment analysis library. BERT in FinBERT stands for Bidirectional Encoder Representations from Transformers. FinBERT uses the Reuters TRC2 dataset and Financial PhraseBank to train.

The article headlines are passed to the FinBERT analyzer, and three sentiment categories are labeled. These categories are "Positive," "Neutral," and "Negative." There are scores associated with each sentiment. The max of these scores determines the sentiment category.

Experimental Setup

List of companies evaluated	Microsoft, Apple, Amazon.
Library model to perform sentiment analysis	Hugging Face FinBERT from ProsusAI.
Labels/Classification classes	POSITIVE, NEUTRAL, NEGATIVE.
Pre-Trained Tokenizer	ProsusAI/finbert.
Processing Time	Avg: 5 mins for 300 articles using Hi RAM, TPUs. 15 mins for 1200 articles using Hi RAM, TPUs.

Sentiment	MSFT	AMZN	AMAZON
Negative	187	67	200
Positive	110	30	92
Neutral	103	43	108

Table 6: Sentiment breakdown for different companies - 6 months

Sentiment	MSFT	AMZN	APPL
Negative	365	737	773
Positive	193	436	401
Neutral	240	388	423

Table 7: Sentiment breakdown for different companies - 1 year

A sample of the classified sentiment from the FinBERT model is shown below.

Microsoft's headlines with positive sentiment

3	Chief executives from Alphabet Inc , Amazon.com Inc and Microsoft Corp on Wednesday called on Congress to pass legislation aimed at boosting U.S. economic competitiveness against China, including in chip manufacturing.	Positive
4	Regulators are looking to update rules, which target companies abusing their market power and those setting up illegal cartels, to make them more efficient, EU antitrust chief Margrethe Vestager said on Thursday.	Positive
10	Wall Street ended sharply higher on Thursday, led by Tesla, Nvidia and other megacap growth stocks in a choppy session ahead of a key jobs report due on Friday.	Positive
12	Facebook parent Meta Platforms Inc will use Broadcom Inc's custom chips to build its metaverse hardware, becoming the chipmaker's next billion-dollar ASIC customer, analysts at J.P. Morgan said on Tuesday.	Positive
13	French defence company Thales said on Thursday it has created a new firm, dubbed S3NS, in partnership with Google to offer state-vetted cloud computing services for the storage of some of the country's most sensitive data.	Positive
15	Wall Street finished sharply higher on Tuesday, lifted by Apple, Tesla and other megacap growth stocks after strong retail sales in April eased worries about slowing economic growth.	Positive
16	Saudi Arabia's Public Investment Fund (PIF) has taken a 5.01% stake in Nintendo Co Ltd as the sovereign wealth fund increases its exposure to the Japanese video gaming industry.	Positive
22	Law firms Skadden, Arps, Slate, Meagher & Flom and Goodwin Procter emerged as top deal advisers in the first quarter of 2022, even as global dealmaking took a tumble.	Positive

Microsoft's headlines with negative sentiment

U.S. equity funds witnessed a third weekly outflow in the week to April 27 as investors worried about slowing global growth and a more aggressive Federal Reserve.	Negative
European stocks slid to a one-month low and commodity prices dropped on Monday on renewed concerns about rising interest rates and China's sputtering economy, while Wall Street shares rose, reversing losses after Twitter agreed to be bought by billionaire Elon Musk.	Negative
Apple Inc will raise the starting pay for its U.S. employees, the iPhone maker said on Wednesday, as companies face a tight labor market and a surge in unionization efforts amid rising inflation.	Negative
China's slowing economy and an inflation-driven drop in consumer spending are expected to drag down global shipments of computers and smartphones this year, according to research firm Gartner.	Negative
Stocks sank as investors worried about slowing global growth and a more aggressive Federal Reserve.	Negative
Wall Street tumbled more than 2.5% on Friday, ensuring the three main benchmarks ended in negative territory for the week, as surprise earnings news and increased certainty around aggressive near-term interest rate rises took its toll on investors.	Negative
Rising costs are likely to dent cyber security firm Darktrace's profit growth, J.P.Morgan analysts said on Tuesday, as they started coverage of the stock with an "underweight" rating.	Negative
Ahead of Russia's invasion of Ukraine, Western intelligence agencies warned of potential cyberattacks which could spread elsewhere and cause "spillover" damage on global computer networks.	Negative

Microsoft's headlines with negative sentiment

U.S. business software maker Oracle Corp is set to gain unconditional EU antitrust clearance for its $28.3 billion acquisition of U.S. healthcare IT company Cerner Corp ,three people familiar with the matter said on Tuesday.	Neutral
British Land said on Monday it had sold 75% stake in its Paddington Central assets to Singapore's sovereign wealth fund GIC for 694 million pounds ($885.9 million), with the UK-based landlord looking to invest in high-growth warehousing properties.	Neutral
The wife of Edward C. Johnson IV, whose family controls mutual fund powerhouse Fidelity Investments, has hired celebrity lawyer David Boies to represent her in a high-stakes divorce trial that opens on Friday.	Neutral
As South African artist Fhatuwani Mukheli paints a portrait of a woman at his Johannesburg studio, he is creating not only the work before him but also a digital asset destined to adorn a virtual world.	Neutral
Activision Blizzard Inc has appointed former Accenture executive Kristen Hines as its new chief diversity, equity and inclusion officer, the "Call of Duty" maker that has been under fire for the culture at the company said on Monday.	Neutral
Concerns about a possible U.S. recession are prompting some fund managers to rotate back into the big tech and growth winners of the last decade in the hope that they can better weather an economic storm.	Neutral
Amazon.com Inc wants to give customers the chance to make Alexa, the company's voice assistant, sound just like their grandmother -- or anyone else.	Neutral

Observations

i) The data is we distributed and not skewed towards "Negative" sentiments. Which indicates DistilBERT was probably classifying many "Neutral" sentiments are

"Negative". There was no change to the "Positive" sentiments.

ii) FinBERT is performant on very compute and memory heavy machines.

3.6. Results of the Experiments

List of Experiments: This section goes over the different experiments conducted and the results. All experiments will use the following metrics to evaluate the model.

i) Confusion Matrix

ii) AUC score

iii) ROC curve.

No.	Category	Algorithm	Duration	Sentiment Model	Target
1	Baseline	Logistic Regression	6 months	DistilBERT	Stock Price.
2	Baseline	SVM	6 months	DistilBERT	Stock Price.
3	Test	LSTM	6 months	FinBERT	Stock Price.
4	Baseline	Logistic Regression	6 months	DistilBERT	Index Price.
5	Baseline	SVM	6 months	DistilBERT	Index Price.
6	Test	LSTM	6 months	FinBERT	Index Price.
7	Baseline	Logistic Regression	1 year	DistilBERT	Stock Price.
8	Baseline	SVM	1 year	DistilBERT	Stock Price.
9	Test	LSTM	1 year	FinBERT	Stock Price.
10	Baseline	Logistic Regression	1 year	DistilBERT	Index Price.
11	Baseline	SVM	1 year	DistilBERT	Index Price.
12	Test	LSTM	1 year	FinBERT	Index Price.

Table 8: List of Experiments

3.6.1 Experiment 1

Description: To evaluate the performance of BERT sentiment analysis and Logistic Regression model that predicts the direction of the MSFT stock price.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT.
Data	6 months.
Model Parameter Tuning	Optimal Solver = Liblinear, Penalty = "l1"

Table 9: Model Parameters

Hyper-parameter tuning was performed using the criteria listed in the table below.

Class	LogisticRegression()
penalty	['l1', 'l2']
C	np.logspace(-4, 4, 20)
solver	['liblinear']

Table 10: Model Parameters - Logistic Regression

Listed below are performance metrics for the experiment.

Model Accuracy = 0.61

Model Precision = 0.62

Model Recall = 0.60

		Predicted		
		0	1	Total
Actual	0	40	29	69
	1	18	30	48
	Total	58	60	118

Table 11: Confusion Matrix - Logistic Regression (Exp 1.)

3.6.2 Experiment 2

Description: To evaluate the performance of the SVM model that predicts the direction of the stock price.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT.
Data	6 months.
Parameter Tuning	Optimal Parameters (CVGridSearch) = SVC(C=1000, gamma=1)

Table 12: Model Parameters

Listed below are the GridSearch Hyper Parameter Tuning parameters.

Input Array	$'C'$: [0.1, 1, 10, 100, 1000, 10000],
	'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
	'kernel': ['rbf','sigmoid','poly']

Table 13: Model Parameter Tuning - SVM

List below are performance metrics for the experiment.

Model Accuracy = 0.95

Model Precision = 0.95

Model Recall = 0.95

		Predicted		
		0	1	Total
Actual	0	50	4	54
	1	1	42	43
	Total	51	46	97

Table 14: Confusion Matrix - SVM (Exp 2.)

3.6.3 Experiment 3

Description: To evaluate the performance of the LSTM model that predicts the direction of the stock price. The table below lists the parameters for the experiment.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT.
Data	6 months.
Hyper Parameter Tuning	Optimal Parameters input_unit: 64 Dropout_rate: 0.3 dense_activation: sigmoid Score: 0.97

Table 15: Model Parameters

Listed below are the Tuner Hyper Parameter Tuning parameters.

Dropout$_r ate$	0.5 step of .1
activation	relu, sigmoid
input_rate	32 < val < 512 step of 32
Random Search	objective="accuracy", max_trials = 4, executions_per_trial =2
Tuner Search	epochs = 50, batch_size =10

Table 16: Hyperparameter Parameter Tuning - LSTM

List below are performance metrics for the experiment.

Model Accuracy = 0.63

Model Precision = 0.65

Model Recall = 0.63

		Predicted		
		0	1	Total
Actual	0	44	28	72
	1	15	28	43
	Total	59	56	115

Table 17: Confusion Matrix - LSTM (Exp 3.)

3.6.4 Experiment 4

Description: To evaluate the performance of BERT sentiment analysis and Logistic Regression model that predicts the direction of the index price for a period of 6 months.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT, AMZN, APPL and S&P 500.
Data	6 months.
Optimal Parameters	solver='liblinear', penalty='l2'

Table 18: Model Parameters

List below are performance metrics for the experiment.

Model Accuracy = 0.71

Model Precision = 0.70

Model Recall = 0.70

Class	LogisticRegression()
penalty	['l1', 'l2']
C	np.logspace(-4, 4, 20)
solver	['liblinear']

Table 19: Model Parameter Tuning - Logistic Regression

		Predicted		
		0	1	Total
Actual	0	407	160	567
	1	187	408	595
	Total	594	568	1150

Table 20: Confusion Matrix - Logistic Regression (Exp 4)

3.6.5 Experiment 5

Description: To evaluate the performance of BERT sentiment analyis and the SVM model that predicts the direction of the index price for a period of 6 months.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT, AMZN, APPL and S&P 500.
Data	6 months.
Parameter Tuning	Optimal Parameters (CVGridSearch) = SVC(C=1000, gamma=.1)

Table 21: Model Parameters

Listed below are the GridSearch Hyper Parameter Tuning parameters.

Input Array	$'C'$: [0.1, 1, 10, 100, 1000, 10000],
	'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
	'kernel': ['rbf','sigmoid','poly']

Table 22: Model Parameter Tuning - SVM

List below are performance metrics for the experiment.

Model Accuracy = 0.99.5

Model Precision = 0.99.8

Model Recall = 0.99.6

		Predicted		
		0	1	Total
Actual	0	540	5	545
	1	0	614	614
	Total	540	619	1159

Table 23: Confusion Matrix - SVM (Exp 5.)

3.6.6 Experiment 6

Description: To evaluate the performance of FinBERT sentiment analysis and the LSTM model that predicts the direction of the index price for period of 6 months.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT, AMZN, APPL and S&P 500.
Data	6 months.
Parameter Tuning	Optimal Parameters
input_unit	352
Dropout_rate	0.2
dense_activation	sigmoid
Score	1.0

Table 24: Model Parameters

Listed below are the keras_tuner Hyper Parameter Tuning parameters.

Dropout$_r ate$	0-.5 step of .1
activation	relu, sigmoid
input$_r ate$	$32 <$ val < 512 step of 32
Random Search	objective="accuracy", $\max_t rials = 4, executions\_per\_trial = 2$
Tuner Search	epochs = 50, batch_size =10

Table 25: Hyperparameter Tuning - LSTM

List below are performance metrics for the experiment.

Model Accuracy = .55

Model Precision = .84

Model Recall = .54

		Predicted		
		0	1	Total
Actual	0	540	5	545
	1	0	614	614
	Total	540	619	1159

Table 26: Confusion Matrix - LSTM (Exp 6.)

3.6.7 Experiment 7

Description: To evaluate the performance of the Logistic Regression model that predicts the direction of the stock price (MSFT) over the period of a year. The table below lists the parameters for the experiment.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT.
Data	1 year.
Parameter Tuning	Optimal Parameters 'C': 100, 'penalty': 'l1', 'solver': 'liblinear'

Table 27: Model Parameters

Listed below are the GridSearch Hyper Parameter Tuning parameters.

Class	LogisticRegression()
penalty	['l1', 'l2']
C	np.logspace(-4, 4, 20)
solver	['liblinear']

Table 28: Model Parameter Tuning - Logistic Regression

List below are performance metrics for the experiment.

Model Accuracy = .66

Model Precision = .68

Model Recall = .66

		Predicted		
		0	1	Total
Actual	0	82	52	134
	1	26	68	94
	Total	108	120	228

Table 29: Confusion Matrix - Logistic Regression (Exp 7.)

3.6.8 Experiment 8

Description: To evaluate the performance of the SVM model that predicts the direction of the stock price.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT.
Data	1 year.
Parameter Tuning	Optimal Parameters (CVGridSearch) = SVC(C=1000, gamma=1)

Table 30: Model Parameters

Listed below are the GridSearch Hyper Parameter Tuning parameters.

Input Array	$'C'$: [0.1, 1, 10, 100, 1000, 10000],
	'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
	'kernel': ['rbf','sigmoid','poly']

Table 31: Model Parameter Tuning - SVM

List below are performance metrics for the experiment.

Model Accuracy = 0.72

Model Precision = 0.72

Model Recall = 0.72

		Predicted		
		0	1	Total
Actual	0	101	30	131
	1	38	71	109
	Total	139	101	240

Table 32: Confusion Matrix - SVM (Exp 8)

3.6.9 Experiment 9

Description: To evaluate the performance of the FinBERT sentiment analysis and the LSTM that predicts the direction of the stock price (MSFT) over the period of 1 year.

The table below lists the parameters for the experiment.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT.
Data	1 year.
Parameter Tuning	Optimal Parameters
	Hyperparameters:
	input_unit: 64
	Dropout_rate: 0.0
	dense_activation: relu
	Score: 0.8656987249851227

Table 33: Model Parameters

Listed below are the GridSearch Hyper Parameter Tuning parameters.

Dropout$_r ate$	0-.5 step of .1
activation	relu, sigmoid
input$_r ate$	32 < val < 512 step of 32
Random Search	objective="accuracy", max_trials = 4, executions_per_trial =2
Tuner Search	epochs = 50, batch_size =10

Table 34: Hyperparameter Tuning - LSTM

List below are performance metrics for the experiment.

Model Accuracy = .72

Model Precision = .72

Model Recall = .72

		Predicted		
		0	1	Total
Actual	0	101	30	131
	1	38	71	109
	Total	139	101	140

Table 35: Confusion Matrix - LSTM (Exp 9.)

3.6.10 Experiment 10

Description: To evaluate the performance of the BERT sentiment analysis and Logistic Regression model that predicts the direction of the index price over the period of a year. The table below lists the parameters for the experiment.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT, AMZN, APPL and S&P
Data	1 year.
Parameter Tuning	Optimal Parameters solver='newton-cg', penalty='l2'

Table 36: Model Parameters

Listed below are the GridSearch Hyper Parameter Tuning parameters.

Class	LogisticRegression()
penalty	['l1', 'l2']
C	np.logspace(-4, 4, 20)
solver	['liblinear']

Table 37: Model Parameter Tuning - Logistic Regression

List below are performance metrics for the experiment.

Model Accuracy = .71

Model Precision = .71

Model Recall = .71

		Predicted		
		0	1	Total
Actual	0	2775	897	3672
	1	1227	2264	3491
	Total	4002	3161	7163

Table 38: Confusion Matrix - Logistic Regression (Exp 10.)

3.6.11 Experiment 11

Description: To evaluate the performance of the SVM model that predicts the direction of the index price over the period of a year. The table below lists the

parameters for the experiment.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT,APPL,AMZN,S&P 500.
Data	1 year.
Parameter Tuning	Optimal Parameters C=0.1, gamma=1, kernel="rbf"

Table 39: Model Parameters

Listed below are the GridSearch Hyper Parameter Tuning parameters.

Input Array	$'C'$: [0.1, 1, 10, 100, 1000, 10000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf','sigmoid','poly']

Table 40: Model Parameter Tuning - SVM

List below are performance metrics for the experiment.

Model Accuracy = .95

Model Precision = .95

Model Recall = .95

		Predicted		
		0	1	Total
Actual	0	3508	201	3709
	1	136	3318	3454
	Total	3644	3519	7163

Table 41: Confusion Matrix - SVM (Exp 11.)

3.6.12 Experiment 12

Description: To evaluate the performance of the FinBERT sentiment analysis and the LSTM that predicts the direction of the S&P index over the period of 1 year. The table below lists the parameters for the experiment.

Input Parameters:

Fundamental	Sentiment of news headlines.
Technical	Momentum.
Ticker	MSFT, APPL, AMZN, S&P
Data	1 year.
Parameter Tuning	Optimal Parameters
	Hyperparameters:
	input_unit: 32
	Dropout_rate: 0.0
	dense_activation: sigmoid
	Score: 0.9998965561389923

Table 42: Model Parameters

Listed below are the GridSearch Hyper Parameter Tuning parameters.

Dropout$_r ate$	0.5 step of .1
activation	relu, sigmoid
input_rate	32 < val < 512 step of 32
Random Search	objective="accuracy", max_trials = 4, executions_per_trial =2
Tuner Search	epochs = 50, batch_size =10

Table 43: Hyperparameter Tuning - LSTM

List below are performance metrics for the experiment.

Model Accuracy = .72

Model Precision = .68

Model Recall = .81

		Predicted		
		0	1	Total
Actual	0	2503	1112	3615
	1	593	2007	2800
	Total	3096	3119	6215

Table 44: Confusion Matrix - LSTM (Exp 12.)

Chapter IV.

Discussion

The table below has the performance metrics for each of the experiments to be referred to in the following discussions.

Experiments	**Performance Matrix - 6 months**		
	Accuracy	**Precision**	**Recall**
1. MSFT (LG + BERT)	61%	62%	60%
2. MSFT (SVM + BERT)	95%	95%	95%
3. MSFT (LSTM + FinBERT)	63%	65%	63%
4. S&P (LG + BERT)	71%	70%	70%
5. S&P (SVM + BERT)	99.5%	99.5%	99.5%
6. S&P (LSTM + FinBERT)	55%	84%	54%

Experiments	Performance Matrix - 1 year		
	Accuracy	**Precision**	**Recall**
7. MSFT (LG + BERT)	66%	68%	66%
8. MSFT (SVM + BERT)	72%	72%	72%
9. MSFT (LSTM + FinBERT)	72%	72%	72%
10. S&P (LG + BERT)	71%	70%	70%
11. S&P (SVM + BERT)	95%	95%	95%
12. S&P (LSTM + FinBERT)	71%	70%	71%

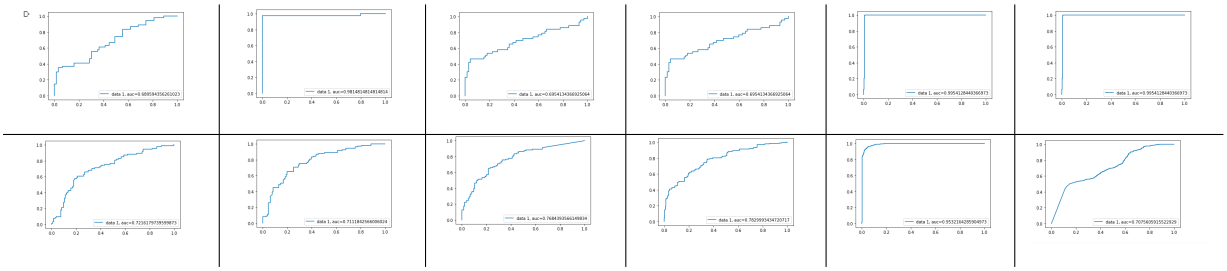


Table 45: Roc Curve for Experiments (row 1 - Experiment 1-6) (row 2 - Experiment 7-12)

4.1. Sentiment Analysis - FinBERT vs DistilBERT

One of the challenges in the sentiment analysis was to collect articles that were available at a reasonable cost. Although many sites accumulate data over some time and charge for data that is over six months, there are no filters for companies. Reuters API tags headlines in specific categories like business, world, sports, legal,

markets, technology, and breaking views. In short, Reuters API provided filters that were helpful in the data collection process.

Scrapy, the web-extraction framework, had many advantages; building Spiders was easy since the orchestration of the pipelines is all handled by the framework. With auto-throttle features, it helps with faster downloads and managing large data loads.

DistilBERT from hugging face was performant and reasonably accurate compared to other BERT-based models. Data for the entire year with all three companies were classified in less than 30 minutes. The sentiment distribution from the initial data analysis showed that the data aligned with the market trends. FinBERT performance was good for smaller data sets (six months); however, for larger data sets, high RAM usage was observed ($>$ 64GB) and a long processing time. The data was batched into 200 headlines per batch with TPU and High RAM configuration in Google Colab. FinBERT had some advantages over DistilBERT, where it added another category for classification. The sentiment was classified into three different categories "Positive," "Negative," and "Neutral." In addition to the sentiment categories, they also provide a sentiment score between 0 and 1 for each category. There were 20,000 articles classified over 30 hours in batches of 200 articles per batch.

In conclusion, Both libraries had more negative than positive articles for six months. However, FinBERT had better distribution when classifying articles for a year, with slightly more negative articles than positive ones and a more significant number of neutral and positive articles together than the negative articles.

4.2. Baseline Models with BERT Sentiment Analysis

The results of the experiments for the baseline metrics are explained below. The baseline metrics included the Logistic Regression and SVM models for binary classification. The two use cases evaluated were predicting the stock (MSFT) price and the index (S&P 500) price direction.

Stock price prediction 6 months - The baseline metrics for predicting the direction of the stock price (MSFT) with six months of data showed better results with SVM (95% accuracy) when compared with Logistic Regression (61%). In the case of stock prediction, high precision is desirable since about 95% of the time; the model predicts the direction of the stock price correctly using BERT sentiment results and the stock price momentum. The SVM model also had a high rate of recall which means 95% of the time, the model was accurately able to predict the direction (upwards or downwards) correctly.

Stock price prediction 1 year - The baseline metrics did improve marginally for the Logistic Regression model to 66% from 61% suggesting that Logistic Regression did not perform well with larger data-sets as expected. The SVM model performed better with 72% accuracy and a high level of precision and recall both at 72%.

Index price prediction 6 months - The baseline metrics for predicting the direction of the index price (S&P 500) with six months of data showed good results with both Logistic Regression (71% accuracy) and SVM (99.5% accuracy). In the

case of stock prediction, high precision is desirable since about 99.5% of the time; the model predicts the direction of the stock price correctly using BERT sentiment results and the stock price (MSFT, AMZN, APPL) momentum. The SVM model also had a high rate of recall which means 99.5% of the time, the model was accurately able to predict the direction of the index movement (upwards or downwards) correctly.

Stock price prediction 1 year - The baseline metrics remained the same for the Logistic Regression model at 71%, suggesting that adding data did not improve the Logistic Regression model. The SVM model performed better with 95% accuracy and a high level of precision and recall, both at 95%. The SVM model performed well for both six months of data and one year of data when predicting the direction of the index price.

The performance metrics show that the SVM model significantly outperformed the traditional logistic regression model. The baseline methods performed better for index price prediction than stock price prediction. DistilBERT classified the sentiments well, given the accuracy of the SVM model at 95% for six months for stock price prediction and 99.5% for index price prediction.

4.3. LSTM and FinBERT Sentiment Analysis

The results of the experiments for the end state metrics are explained below. The end state included the LSTM models for time series classification. The two use cases evaluated were predicting the stock (MSFT) price and the index (S&P 500)

price direction.

Stock price prediction - Since in stock price prediction classifying the upward direction is equally as crucial as the downward direction, precision, and recall must be high. For stock price prediction using FinBERT sentiment analysis, the LSTM resulted in a 63% accuracy with precision at 65% and recall at 63%. The model did perform better with more data points, indicating that adding data improved the accuracy (at 72%), precision (at 72%), and recall (at 72%).

Index price prediction - Index price prediction showed weak results for six months of data with accuracy at 55%; however, with a high rate of precision of 85%, even with a low accuracy rate, the model could give correct predictions 85% of the time. However, the recall was low at 54%. When the data set was increased to a year, there was higher accuracy at 71%. The model accuracy was a 16% improvement in accuracy. With both precision and recall increasing to 70% and 71%, respectively.

LSTM showed similar results for both stock price prediction and index price prediction. Lower with smaller data sets and improved with larger data sets. However, the accuracy still needs to be significantly better than SVM. Hyperparameter tuning needed careful thought and consideration and was more time-consuming than the SVM model.

4.4. Stock Price Prediction vs Index Price Prediction

All the experiments had interesting outcomes for market predictions. The results provide a fascinating insight into the algorithms that work well for index and stock price predictions. Microsoft price data was used for stock price prediction, and for index price prediction, the S&P price data was used. Three companies' fundamental and technical data were input parameters for index price prediction. These companies included Microsoft, Amazon, and Apple based on their weights in the index calculation.

The difference in the input parameters was crucial in the different models used for stock price prediction and index price prediction. The number of stocks used resulted in the stock price models having fewer input parameters than the index price models. As a result, the number of articles for a single stock was less than the total number for multiple stocks resulting in larger data sets for stock price prediction.

The stock price prediction models did not perform as well as the index prediction models with most algorithms (except the LSTM model). In both cases, the SVM models performed the best.

Additional data points did result in better model accuracy except in the case of Logistic Regression, where stock market prediction dropped accuracy with a larger data set. To summarize, data for one year yielded better accuracy in most experiments for both stock price and index price prediction.

LSTM models favored stock market prediction when compared to index market

prediction. BERT and FinBERT kept the accuracy of the model relatively the same.

4.5. Data Volume and Model Performance

Adding data for a whole year brought challenges with performance. The system thresholds were throttled with the sentiment classification tasks (FinBERT sentiment classification).

As the experiments above showed, not all models were performed with a year's data. For stock prediction, the SVM model that performed well in most experiments performed poorly with one year's data giving a 72% accuracy rate compared to a 95% accuracy with six months of data.

For index price predictions, most models performed the same with one year's data compared to six months, except for the LSTM model. The Logistic Regression model had the same accuracy at 71% for six months of data and one year's worth of data. The SVM model performed well at 95% with one year of data, just as it did for six months. The LSTM model, however, improved performance by 20% with one year of data.

Chapter V.

Conclusion

5.1. Summary

Stock market prediction continues to be an active area of research, with much progress made in prediction accuracy. This work compares traditional machine learning methods with recent ones, including LSTM and FinBERT, to assess improvements, challenges, and future directions.

Our observations show that the FinBERT model had better distribution of sentiments than the out-of-box BERT or DistilBERT libraries. For both data sets, the distributions from FinBERT were classified more specifically with the additional neutral category, indicating that DistilBERT classified neutral sentiments as positive and negative even when the tone was more neutral, which could impact the outcomes of the predictions.

However, the experiments demonstrated that the additional categories did not improve the performance of the models as expected. We were able to achieve statistically significant results with DistilBERT. The SVM models with DistilBERT classification of sentiments performed the best at $72 - 99\%$ accuracy.

The traditional Logistic Regression model did not perform well, especially with stock price prediction. Adding features may improve the accuracy of the models. However, this observation supports the existing trend to move towards machine learning algorithms, given the exponential increase in the volume of data and processing power.

The SVM models performed the best with high accuracy ranging from 72% to 99%. These models performed exceptionally well with the S&P index predictions.

The LSTM models required extensive tuning of hyper-parameters and model composition. With limited literature on optimizing different layers, input values, and error functions, this required many iterations and yielded marginal gain in accuracy. Improvement was achieved with one year's data, indicating that additional data points improved model performance, but it still did not perform as well as the SVM model. The accuracy of the models remained at 72% at their best. The accuracy is a higher percentage from research listed in chapter 1; it is lower than the accuracy of SVM models.

That the models build for predicting the direction of the S&P index with SVM lead to the best performance statistics (accuracy, precision, and recall). SVM has traditionally done well for classification problems. LSTM and SVM outperform the statistical method (Logistic Regression) in all experiments.

Some lessons learned were that performing sentiment analysis using FinBERT pre-trained model required heavy system resources and was time-consuming. When

data for an entire year was classified, it required over 30 hours for classification. DistilBERT resulted in fairly accurate models and took less time. Given the trade-off, FinBERT did not add as much value to the outcome.

LSTM models performed better with fewer feature variables which makes us rethink the approach; LSTM is also a better solution for tracking stock price classification vs. index prices classification.

SVM outperformed all other models, especially for index price movement, demonstrating that it did well in classifying the movement and did better with a bigger feature set.

Some avenues for further exploration might be developing an LSTM model with a single feature set, i.e., only price movement or sentiment score as input parameters. Adding intra-day prices along with the market sentiment might be another option giving the model data points to improve accuracy.

In conclusion, this thesis has attempted to compare the LSTM algorithm and compared it with conventional algorithms to predict market trends. The experiments provide metrics for future experiments to optimize the model's structure and parameters or compare multiple methods to explore more accurate prediction effects.

Source Code

The source code for this project can be found in the *github repository*.

References

Achyutha, P. N., Chaudhury, S., Bose, S. C., Kler, R., Surve, J., & Kaliyaperumal, K. (2022). User classification and stock market-based recommendation engine based on machine learning and twitter analysis. *Mathematical Problems in Engineering*, 2022.

Agusta, I. M. A. I., Barakbah, A., & Fariza, A. (2022). Technical analysis based automatic trading prediction system for stock exchange using support vector machine. *EMITTER International Journal of Engineering Technology*, 10(2), 279–293.

AI, H. F. (2023). Hugging face documentation.

Andrawos, R. (2022). Nlp in stock market prediction: A review.

Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *Cornell University-arXiv*.

Ashtiani, M. N. (2018). News-based intelligent prediction of financial markets using text mining and machine learning: a systematic literature review. *Expert Systems With Applications*.

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20), 7046–7056.

Chuang, C. & Yang, Y. (2022). Buy tesla, sell ford: Assessing implicit stock market preference in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 100–105).

Dautel, A. J. et al. (2020a). Forex exchange rate forecasting using deep recurrent neural networks. *Digital Finance*, 2(1-2), 69–96.

Dautel, A. J., Härdle, W. K., Lessmann, S., & Seow, H.-V. (2020b). Forex exchange rate forecasting using deep recurrent neural networks. *Digital Finance*, 2(1), 69–96.

DeSola, V., Hanna, K., & Nonis, P. (2019). Finbert: pre-trained model on sec filings for financial natural language tasks. *University of California*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Geron, A. (2019). *Hands-on machine learning with Scikit-learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol: O'Reilly Media, second edition. edition.

Gron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 1st edition.

Huang, A. H., Wang, H., & Yang, Y. (2020). Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*.

Huang, A. H., Wang, H., & Yang, Y. (2022). Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*.

Huynh, H. D., Dang, L. M., & Duong, D. (2017). A new model for stock price movements prediction using deep neural network. SoICT '17 (pp. 57–62). New York, NY, USA: Association for Computing Machinery.

Ismail, A., Wood, T., & Bravo, H. (2018). Improving long-horizon forecasts with expectation-biased lstm networks.

jae Kim, K. & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, 19(2), 125–132.

Kang, H., Zong, X., Wang, J., & Chen, H. (2023). Binary gravity search algorithm and support vector machine for forecasting and trading stock indices. *International Review of Economics Finance*, 84, 507–526.

Kim, K.-J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1), 307–319. Support Vector Machines.

Koosha, E., Seighaly, M., & Abbasi, E. (2022). Measuring the accuracy and precision of random forest, long short-term memory, and recurrent neural network models in predicting the top and bottom of bitcoin price. *Journal of Mathematics and Modeling in Finance*.

Li, C., Shen, L., & Qian, G. (2023). Online hybrid neural network for stock prices prediction: A case study of high-frequency stock trading in china market.

Li, M., Chen, L., Zhao, J., & Li, Q. (2021). Sentiment analysis of chinese stock reviews based on bert model. *Applied Intelligence*, 51, 5016–5024.

Li, M., Li, W., Wang, F., Jia, X., & Rui, G. (2020). Applying bert to analyze investor sentiment in stock market - neural computing and applications.

Lin, M. & Chen, C. (2018). Short-term prediction of stock market price based on ga optimization lstm neurons. ICDLT '18 (pp. 66–70). New York, NY, USA: Association for Computing Machinery.

Liu, Z., Huang, D., Huang, K., Li, Z., & Zhao, J. (2021). Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 4513–4519).

Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A cnn–lstm model for gold price time-series forecasting. *Neural computing and applications*, 32, 17351–17360.

Maindonald, J. & Braun, W. J. (unknown). Data analysis and graphics using r.

Mao, Y., Wei, W., Wang, B., & Liu, B. (2012). : New York, NY, USA: Association for Computing Machinery.

Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, 21(41), 89–93.

Muller, A. & Guido, S. (2018). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Incorporated.

Murphy, J. (1999). *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Prentice Hall PTR.

Osterrieder, J. (2023). A primer on natural language processing for finance. In *Fintech and Artificial Intelligence in Finance*: Europe.

Pajankar, A. & Joshi, A. (2022). *Recurrent Neural Networks*, (pp. 285–305). Apress: Berkeley, CA.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Peng, B., Chersoni, E., Hsu, Y.-Y., & Huang, C.-R. (2021). Is domain adaptation

worth your investment? comparing BERT and FinBERT on financial tasks. In *Proceedings of the Third Workshop on Economics and Natural Language Processing* (pp. 37–44). Punta Cana, Dominican Republic: Association for Computational Linguistics.

Petrusheva, N. & Jordanoski, I. (unknown). Comparative analysis between the fundamental and technical analysis of stocks.

Protopapas, P., Mark, G., & Chris, T. (2023). *Lecture Notes*. Harvard University.

Raschka, S., Liu, Y., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop Machine Learning and Deep Learning Models with Python*. Expert Insight. Packt Publishing.

Saleh, H. & Sen, S. (2019). *Machine Learning Fundamentals*. Packt Publishing.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Socher, R., Bengio, Y., & Manning, C. (2012). Deep learning for nlp. *Tutorial at Association of Computational Logistics (ACL)*.

Souma, W., Vodenska, I., & Aoyama, H. (2019). Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science*, 2(1), 33–46.

Steven Bird, E. K. & Loper, E. (2023). Natural language processing with python.

Tay, F. E. & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *omega*, 29(4), 309–317.

Vargas, M. R., de Lima, B. S. L. P., & Evsukoff, A. G. (2017). Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)* (pp. 60–65).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Villamil, L., Bausback, R., Salman, S., Liu, T. L., Horn, C., & Liu, X. (2023). Improved stock price movement classification using news articles based on embeddings and label smoothing. *arXiv preprint arXiv:2301.10458*.

Wójcik, P. & Osowska, E. (2023). The impact of federal open market committee post-meeting statements on financial markets – a text-mining approach.

Yang, C., Ou, K., & Hong, S. (2022a). Application of nonstationary time series prediction to shanghai stock index based on svm. In *Proceedings of the 3rd Asia-Pacific Conference on Image Processing, Electronics and Computers*, IPEC '22 (pp. 302–305).: Association for Computing Machinery.

Yang, Y., Hu, X., & Jiang, H. (2022b). Group penalized logistic regressions predict

up and down trends for stock prices. *The North American Journal of Economics and Finance*, 59, 101564.

Yang, Y., Uy, M. C. S., & Huang, A. (2020). Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

Yıldırım, D. C., Toroslu, I. H., & Fiore, U. (2021). Forecasting directional movement of forex data using LSTM with technical and macroeconomic indicators. *Financial Innovation*, 7, 1–36.

Zhai, Y. et al. (2007). Combining news and technical indicators in daily stock price trends prediction. In *Advances in Neural Networks-ISNN 2007* (pp. 1087–1096).: Springer.

Zhang, W., Yan, K., & Shen, D. (2021). Can the baidu index predict realized volatility in the chinese stock market? *Financial Innovation*, 7(1), 1–31.

Zhang, Y. & Wu, L. (2009). Stock market prediction of sp 500 via combination of improved bco approach and bp neural network. *Expert Systems with Applications*, 36, 8849–8854.

Zyte, I. (2023). Scrapy 2.8 documentation.