



SCHOOL of
GRADUATE STUDIES
EAST TENNESSEE STATE UNIVERSITY

East Tennessee State University
Digital Commons @ East Tennessee
State University

Electronic Theses and Dissertations


Student Works

5-2023

Predicting High-Cap Tech Stock Polarity: A Combined Approach using Support Vector Machines and Bidirectional Encoders from Transformers

Ian L. Grisham
East Tennessee State University

Follow this and additional works at: <https://dc.etsu.edu/etd>

 Part of the [Artificial Intelligence and Robotics Commons](#), [Computational Linguistics Commons](#), [Data Science Commons](#), [Finance Commons](#), and the [Probability Commons](#)

Recommended Citation

Grisham, Ian L., "Predicting High-Cap Tech Stock Polarity: A Combined Approach using Support Vector Machines and Bidirectional Encoders from Transformers" (2023). *Electronic Theses and Dissertations*. Paper 4207. <https://dc.etsu.edu/etd/4207>

This Thesis - unrestricted is brought to you for free and open access by the Student Works at Digital Commons @ East Tennessee State University. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ East Tennessee State University. For more information, please contact digilib@etsu.edu.

Predicting High-Cap Tech Stock Polarity: A Combined Approach using Support Vector
Machines and Bidirectional Encoders from Transformers

A thesis
presented to
the faculty of the Department of Computing
East Tennessee State University

In partial fulfillment
of the requirements for the degree
Master of Science in Computer Science, Artificial Intelligence and Machine Learning

by
Ian Grisham
May 2023

Dr. Brian Bennett, Chair
Dr. Ghaith Husari, Committee Member
Dr. Ahmad Al Doulat, Committee Member

Keywords: Quantitative Finance, Sentiment Analysis, Support Vector Machine, BERT

ABSTRACT

Predicting High-Cap Tech Stock Polarity: A Combined Approach using Support Vector Machines and Bidirectional Encoders from Transformers

by

Ian Grisham

The abundance, accessibility, and scale of data have engendered an era where machine learning can quickly and accurately solve complex problems, identify complicated patterns, and uncover intricate trends. One research area where many have applied these techniques is the stock market. Yet, financial domains are influenced by many factors and are notoriously difficult to predict due to their volatile and multivariate behavior. However, the literature indicates that public sentiment data may exhibit significant predictive qualities and improve a model's ability to predict intricate trends. In this study, momentum SVM classification accuracy was compared between datasets that did and did not contain sentiment analysis-related features. The results indicated that sentiment containing datasets were typically better predictors, with improved model accuracy. However, the results did not reflect the improvements shown by similar research and will require further research to determine the nature of the relationship between sentiment and higher model performance.

Copyright 2023 by Ian Grisham
All Rights Reserved

ACKNOWLEDGEMENTS

I would like to acknowledge each of the professors and mentors I have had at ETSU. In particular, I would like to thank Edward Hall and Matthew Harrison for laying the foundation of my career in computer science, always encouraging exploration, and rewarding inquisitive behavior. I would also like to recognize Brian Bennett for helping me navigate the graduate program, mentoring me throughout the capstone process, and providing graduate research opportunities.

Lastly, I would like to thank my friends and family for their support and unwavering confidence. To my father, mother, stepfather, and wife, thank you for keeping me grounded and pushing me to be the best version of myself. To Edgar Guerra, thank you for enriching my life and becoming a lifelong friend. With the help of those around me, I have come to know the goodness of life, and for that, I am truly grateful.

“Good company in a journey makes the way seem shorter.” – Izaak Walton

TABLE OF CONTENTS

ABSTRACT.....	2
ACKNOWLEDGEMENTS.....	4
LIST OF TABLES.....	8
LIST OF FIGURES.....	9
CHAPTER 1. INTRODUCTION.....	12
1.1 Background.....	12
1.2 Problem Statement and Research Questions.....	12
1.3 Proposed Approach.....	13
CHAPTER 2. LITERATURE REVIEW.....	14
1.1 Efficient Market Hypothesis.....	14
1.1.1 Random Walks.....	14
1.1.2 Martingales.....	15
1.2 EMH Resistance.....	16
1.2.1 Buying Strategies.....	16
1.2.2 Longer Holding Periods.....	16
1.2.3 Machine Learning.....	17
1.2.4 Sentiment Analysis.....	19
CHAPTER 3. APPROACH.....	22
3.1 Data.....	22
3.1.1 Selection Methodology.....	22

3.1.2 Price	23
3.1.2.1 Outliers.....	23
3.1.3 Text.....	26
3.1.3.1 News Headlines	26
3.1.3.2 Twitter Posts	26
3.1.3.3 Text Preparation.....	26
3.1.4 Target.....	27
3.2 Model Design	27
3.2.1 Bidirectional Encoder Representations from Transformers	27
3.2.2 Support Vector Machine.....	30
3.3 Implementation.....	31
3.3.1 BERT Sentiment Analysis.....	32
3.3.2 Feature Engineering.....	37
3.3.2.1 Optimal Moving Averages.....	37
3.3.2.2 Momentum.....	38
3.3.2.3 Volume.....	39
3.3.2.4 Volatility	40
3.3.2.5 Trend	40
3.3.3 Modified Recursive Feature Elimination	41
3.4 Validation	42
3.4.1 Metrics	42
3.4.1 BERT: Cross-Validation.....	43
3.4.1 Support Vector Machine: Walk-Forward Validation	44

CHAPTER 4. EXPERIMENTAL RESULTS	45
4.1 BERT.....	45
4.1.2 Support Vector Machine.....	52
4.1.2.1 No Sentiment	52
4.1.2.2 With Sentiment	61
4.2 Discussion	70
CHAPTER 5. CONCLUSION.....	73
REFERENCES	75
VITA.....	78

LIST OF TABLES

Table 1: Lexicon Mappings	27
Table 2: Momentum Feature Descriptions	38
Table 3: Volume Feature Descriptions	39
Table 4: Volatility Feature Descriptions.....	40
Table 5: Trend Feature Descriptions.....	41
Table 6: BERT Results	46
Table 7: SVM No Sentiment Results.....	53
Table 8: SVM Sentiment Results.....	62
Table 9: SVM No Sentiment vs. Sentiment Results	70
Table 10: Query Issues.....	71

LIST OF FIGURES

Figure 1: Chosen Securities' Prices.....	23
Figure 2: Outliers	25
Figure 3: MLM [20].....	28
Figure 4: NSP [20].....	29
Figure 5: SVM Kernel Trick.....	30
Figure 6: Implementation Flowchart	31
Figure 7: Imbalanced Dataset EDA	33
Figure 8: Balanced Dataset EDA.....	33
Figure 9: WordCloud	34
Figure 10: Chord Diagram	36
Figure 11: Trend Following.....	37
Figure 12:Cross-Validation.....	43
Figure 13: Walk-Forward Validation.....	44
Figure 14: Fold 1 Results	47
Figure 15: Fold 2 Results	48
Figure 16: Fold 3 Results.....	49
Figure 17: Fold 4 Results	50
Figure 18: Fold 5 Results	51
Figure 19: Apple No Sentiment CFM.....	54
Figure 20: AMD No Sentiment CFM	54
Figure 21: Google No Sentiment CFM.....	55
Figure 22: Amazon No Sentiment CFM.....	55

Figure 23: Intel No Sentiment CFM.....	56
Figure 24: Alphabet No Sentiment CFM.....	56
Figure 25: Micron No Sentiment CFM.....	57
Figure 26: Microsoft No Sentiment CFM.....	57
Figure 27: Nokia No Sentiment CFM.....	58
Figure 28: Netflix No Sentiment CFM.....	58
Figure 29: Nvidia No Sentiment CFM.....	59
Figure 30: Oracle No Sentiment CFM.....	59
Figure 31: Twitter No Sentiment CFM.....	60
Figure 32: Tesla No Sentiment CFM.....	60
Figure 33: Apple Sentiment CFM.....	63
Figure 34: AMD Sentiment CFM.....	63
Figure 35: Alphabet Sentiment CFM.....	64
Figure 36: Amazon Sentiment CFM.....	64
Figure 37: Intel Sentiment CFM.....	65
Figure 38: Google Sentiment CFM.....	65
Figure 39: Micron Sentiment CFM.....	66
Figure 40: Microsoft Sentiment CFM.....	66
Figure 41: Nokia Sentiment CFM.....	67
Figure 42: Netflix Sentiment CFM.....	67
Figure 43: Oracle Sentiment CFM.....	68
Figure 44: Nvidia Sentiment CFM.....	68
Figure 45: Twitter Sentiment CFM.....	69

Figure 46: Tesla Sentiment CFM..... 69

CHAPTER 1. INTRODUCTION

1.1 Background

Historically, investors have disagreed about the predictability of stock returns. According to the Efficient Market Hypothesis (EMH), share prices reflect all available information, adjust in real-time, and always trade at their fair value [1]. Therefore, predicting the market is, theoretically, impossible, and price oscillation is best described as a random walk, a path created by a series of discrete but random steps that causes an object to wander randomly from its origin.

Although scholars cite a broad range of evidence in support of the EMH, fundamental and technical investors often dispute its validity. For example, Warren Buffet, Chairman and CEO of Berkshire Hathaway, has consistently beaten the market and produced excess returns for more than 50 years—theoretically impossible, according to the EMH [2]. The recent prevalence of quantitative analysis and algorithmic trading in the financial sector also suggests otherwise and should be considered when assessing the legitimacy of the EMH [3].

A primary example, Renaissance Technology’s Medallion Fund, a quantitative analysis-based hedge fund established in 1988 by mathematician Jim Simon, has averaged approximately a 66% return per annum and produced negative returns in only a single year (1989). Even during economic unrest (e.g., the 2008 market crash and the 2020 pandemic), the Medallion Fund had annual returns approaching 76% and 90%, respectively [4].

1.2 Problem Statement and Research Questions

According to previous literature, sentiment analysis is typically considered an influential factor for security price prediction and a powerful machine learning (ML) feature. However, financial text is often difficult to model correctly because it relies on numbers and symbols that are generally removed during text pre-processing steps. Yet, a finely tuned bidirectional encoder

transformer representation (BERT) model may have the functionality to predict sentiment accurately for this type of text.

This problem leads to several research questions that guide this study.

RQ1. Are BERT models able to digest and understand financial domain text?

RQ2. Does BERT sentiment improve monthly return classification accuracy?

1.3 Proposed Approach

In this experiment, a finely-tuned BERT model was proposed to produce sentiment scores for a variety of publicly available text that could subsequently be used as feature inputs. A comparison between datasets that do or do not contain sentiment scores was used to evaluate the effectiveness public sentiment can have on a model's ability to predict the polarity of 30-day price changes.

CHAPTER 2. LITERATURE REVIEW

1.1 Efficient Market Hypothesis

1.1.1 Random Walks

In 1965, renowned economist Eugene Fama theorized that share prices reflect all information and always trade at their fair value in an efficient market [5]. Fama defined an “efficient market” as a market where the same information is freely available to all market participants, and there is “profit-maximizing” competition between them [5].

Competition between all equally informed, profit-maximizing participants is instantaneously factored into the market, and at any point in time, the share price reflects the true, intrinsic value of the company. However, not all market participants are equally informed and agree on predicted prices. Moreover, in theory, varying levels of disagreement and informativity between market participants at any given time would “cause the actual price to wander randomly” [5]. If the contrary were true, “intelligent market participants” could systematically use this knowledge to beat the market [5].

Fama acknowledged that the random-walk hypothesis was likely not an exact representation of market behavior and “no amount of empirical testing (would be) sufficient to establish (its) validity without a shadow of a doubt” [5] but was a reasonable representation of market behavior. Technical (i.e., chartists) and fundamental analysts who wished to disprove this theory would have had to consistently beat a simple buy-and-hold strategy for a randomly selected portfolio.

Consistently beating a buy-and-hold strategy would imply that the investor had been systemically applying techniques to purchase and sell shares at optimal times and, thus, would have had some advantage over a random-selection strategy. To disprove the EMH, mutual-fund

proponents of the time demonstrated higher portfolio returns than those randomly chosen (i.e., Fisher-Lorie portfolios). However, Fama noted that “if initial loading charges of the funds (were) considered” [5] that a random strategy would have still outperformed the mutual fund portfolios.

1.1.2 Martingales

Although in agreement about the random nature of stock prices, Paul Samuelson criticized Fama’s Random Walk model. Samuelson stated that it “(was) not particularly related to perfect competition or market anticipations” [6]. He later states, “Taken literally, a random walk dictates with certainty that in time the price of a luxury Rolls Royce relative to the price of one green pea can reach equality or any ratio you can name” [7].

Using a more characterized investor behavior, Samuelson proposed that prices follow a martingale when a fair game is assumed. A fair game implies that the chances of winning and losing are equal. Therefore, predicting whether a participant wins or loses is a prediction based on random probability. The nuance between Samuelson’s and Fama’s theories lies in future prices' predictability.

Consider a game where a fair coin (i.e., a coin with equal probability to land on either heads or tails) is flipped. Heads increments a participant’s score by 1, and tails decrements the same participant’s score by 1. On the 10th flip of the coin, Samuelson argued that the expected value of the participant’s score is closest to the score documented by the previous flip and, in opposition to Fama, not some random value. It logically follows that future prices are best estimated using the price of the previous day; however, the likelihood of the price being lower or higher than the estimation cannot be argued. Like Fama, Samuelson states the random nature of deviation from price estimation resulted from competing market participants who are not in complete agreement (i.e., an efficient market). [7]

1.2 EMH Resistance

1.2.1 Buying Strategies

Rosenberg et al. [8] established a strong opinion opposing efficient market theory. To test for market inefficiency, they adopted two buying strategies. For their first strategy (i.e., book/price), Rosenberg et al. bought stocks with “a high ratio of book value of common equity per share to market price share” [8]. Their second strategy (i.e., specific-return-reversal) bought stocks where the “difference between the investment return of the previous month and a fitted value for that return based upon common factors in the stock market (for) the previous month” [8] was negative. Both strategies reported statistically significant excess returns—t-statistics of 3.7 and 11.54, respectively. These statistics implied that their hypothesis for efficient markets could be confidently rejected. Additionally, the two strategies were statistically independent of one another, and as a result, the agreement between the two strategies was a strong indication that markets were not efficient [8].

1.2.2 Longer Holding Periods

Fama and French [9] later state that portfolios with longer holding strategies show evidence of predictable returns. Using assumptions about the “nature of the price process” [9], they showed that the negative autocorrelation found in portfolio returns could explain up to 40% of the variance found in longer holding periods but typically explained less than 5% of the variance for shorter holding periods. Although able to demonstrate some predictability for longer holding periods, they noted that they could not reliably infer anything about their short-term time-series properties [9].

Similarly, Poterba and Summers [10] reported results that suggested variance ratios for detecting mean-reversion in longer holding periods were higher than that of shorter holding

periods. Using data from 1871 to 1986 for various states within the United States, seventeen other countries, and individual firm returns, they concluded that stock returns are positively and serially correlated in the short term and negatively autocorrelated for longer terms. They estimated that: “Transitory components in stock prices (had) a standard deviation of between 15 and 25 percent and account for more than half of the variance in monthly returns.” [10].

One explanation offered by Poterba and Summers for improved mean-reversion in longer holding periods is that “price fads” decay over time and cause the price to converge to its true value. In short, the divergence between a company’s share price and actual price caused by erroneous market trends tends to be corrected over time [10].

DeBondt and Thaler [11] credited the divergence described by Poterba and Summers to overreactions by market participants: That the occasional irrationality of market participants (e.g., investment decisions rooted in optimism or pessimism) caused short-term share prices to deviate from their intrinsic value. Their results appeared to be in stark agreement with Kahneman and Tversky’s [12] theory that market participants were regularly over-optimistic in their ability to predict some future price for a company and were likely the cause for price divergences seen in the market. An additional illustrious finding for DeBondt and Thaler was that return variance grew less proportionally with respect to time, and longer holding period portfolios exhibited higher levels of mean-reversion [11]. The lack of proportionality between time and return variance suggested that longer holding periods may indicate higher levels of predictability.

1.2.3 Machine Learning

More recently, Milosevic [13] evaluated the predictability of long-term stock growth using support vector machines. Milosevic gathered 28 financial indicators (e.g., Book value, market capitalization, 1-month net price change, dividend yield, and earnings per share) for 1729

S&P 1000, FTSE 100, and S&P 350 Europe stock indices for a 3-year period (i.e., 2012-2015). Stocks that increased in value by 10% or more during a one-year period were considered good investments, otherwise bad [13]. Intuitively, stocks that met the criteria to be considered good investments would be ‘market-beaters’ as the 30-year average market inflation-adjusted return is approximately 8.29% [14].

Milosevic balanced the dataset via under-sampling and proceeded with multi-model analysis for an equal representation of good and bad investments. Of the eight models (i.e., decision trees, SVM, JRip, Random Tree, Random Forest, Logistic Regression, Naïve Bayes, and Bayesian Networks) tested, Random Forest was the best performer—achieving approximately 75% accuracy using 10-fold cross-validation [13].

Milosevic manually selected 11 features from the 28 financial indicators to improve the model's performance using a random strategy. Although a more exhaustive approach would have likely performed better, Milosevic found that these 11 features slightly improved the model:

1. Book Value
2. Market Cap
3. Dividend Yield
4. Best EPS
5. PE Ratio
6. P.X. to Book Ratio
7. Best DPS
8. CUR ratio
9. Quick Ratio
10. Total Debt to Equity
11. History Price

Although a slight improvement of 1.4%, Milosevic noted that the algorithm executed more efficiently and in less time when using the limited feature set. Furthermore, they deduced that company growth information did not relate to long-term growth predictability [13].

Similarly, Chen and Hao [15] used feature-weighted support vector machines and KNN (k-nearest-neighbor) models to predict trend direction and closing price for both the Shanghai and Shenzhen Stock Exchange Composite Index using data from 1994-2015. In addition to the novel OHLCV financial format (i.e., open, high, low, close, volume), Chen and Hao used nine other technical indicators as feature inputs.

After normalizing the data, they evaluated the models' performance for 1, 5, 10, 15, 20, and 30-day forecast windows. When predicting closing price rather than direction, they found that larger prediction windows negatively impacted prediction accuracy; on the contrary, trend direction predictions generally improved as the window was extended.

Although there was variation in their results, Chen and Hao's models showed an overall positive trend in prediction accuracy the farther away they attempted to predict. Their models did show some anomalous behavior for the 10-day window (i.e., slightly higher accuracy than both the 15 and 20 windows); however, the largest (i.e., 30-day) window achieved the highest accuracy for all tests [15].

1.2.4 Sentiment Analysis

Machine learning algorithms that attempt to predict the stock market most often generate predictions using historical trends, statistical inference, and, more recently, public sentiment. Public sentiment reflects investors' attitudes toward the overall market, market subsectors, or particular assets and has been a promising feature addition when taking a machine learning approach.

Various different machine learning algorithms and techniques have been used to try and predict securities' price and trend. Yet, to date, nothing definitive can be said about which techniques are better or worse for stock market prediction (SMP): no single algorithm guarantees

an optimal model. Rather, a model's complexity—i.e., the type, quality, and quantity of data with which a model is trained—has a higher impact on its forecasts' accuracy. For example, models that included only market, textual, or market indicator data performed achieved a nearly identical 70% accuracy rating. In comparison, models that used a combination of the three scored 10-15% higher on average [16].

For example, Bharathi and Geetha [17] suggested a possible relationship between people's attitudes about securities to those securities' prices. Bharathi and Geetha mined and analyzed textual data from sources such as RSS stock news feed to increase their Sensex moving average-based SMP model's accuracy by approximately 15%.

Bharathi and Geetha's [17] approach used a pre-processing step to remove inconsistent, improperly formatted, false, and duplicated data from their dataset: a step that prevented this "dirty data" from degrading the model's quality. The cleansed data was then split into sentences, tokenized, and filtered to remove tokens corresponding to punctuation, symbols, and misspelled or incorrectly used words.

Using a dictionary approach, each token was matched with its corresponding dictionary key to retrieve a numerical connotation score: a value ranging from -1 for strongly negative tokens to +1 for strongly positive tokens. Each sentence's token sentiment scores were then summed to produce a sentence-level sentiment score (SSS). Bharathi and Geetha [17] compared these SSS scores with their SMP 5-, 10-, and 15-day scores, treating scores with the same signs as positive or negative indicators according to the sign and scores, with different signs as neutral indicators.

Bharathi and Geetha [17] tested their SMP+SSS model against their SMP model, using two years (2005-2007) of historical price and textual data for the company ARBK from the

Amman Stock Exchange (ASE) and RSS stock news feeds. Over 499 instances, the SMP produced an accuracy rating of 64.32%, while the SMP+SSS model produced an accuracy rating of 78.75%.

These analyses, however, have typically used news and social media content, rendering them susceptible to opinion and bias. Training with biased data can produce dysfunctional models. In a model that includes public sentiment analysis, skewed data from social media and news platforms may cause a model to incorrectly forecast a stock's behavior. For example, the Syrian Electronic Army's seizure of Associated Press's (a popular, neutral news reporting agency) Twitter account on April 23, 2013, followed by its posting of fake articles detailing an attack on the White House, caused an immediate 136 billion US\$ market crash [18]—likely the result of algorithmic trading applications all predicting a crash and simultaneously dumping security shares.

CHAPTER 3. APPROACH

3.1 Data

3.1.1 Selection Methodology

Securities with lower market caps are less resistant to price volatility since share ownership is typically not as diversely spread among market participants and are more likely to succumb to economic instability [19]. Consequently, low-cap securities are likely to experience large, unexpected changes in price (i.e., volatility). Low-volume securities may also exhibit high volatility due to large rifts in their ask-bid spreads. Thus, large-cap and mega-cap securities (i.e., total market cap exceeding \$10B) were chosen where the average daily shares exchanged between buyers and sellers exceeded 15M (i.e., high volume).

The pool of chosen securities represented approximately 0.007% of all stocks in NASDAQ, the New York Stock Exchange (NYSE), and the American Stock Exchange (AMEX). Although a variety of securities may represent a more complete view of the total market, statistical edges, or distinct advantages that result in a predicted positive return, in one sector of the market would likely compete with statistical edges in other sectors of the market. They may complicate the model's ability to find any edge at all. As a result, the remaining securities were filtered to include only those within the technology domain.

After the aforementioned filtration, 14 securities remained: Apple (APPL), AMD (AMD), Amazon (AMZN), Google (GOOG), Alphabet (GOOGL), Intel (INTC), Microsoft (MSFT), Micron (MU), Netflix (NFLX), Nokia (NOK), Nvidia (NVDA), Oracle (ORCL), Tesla (TSLA), and Twitter (TWTR). The remaining securities represented approximately 0.0017% of all stocks in the chosen U.S. exchanges, yet they accounted for roughly 20% of the \$42.9T total valuation.

The collection period length was arbitrarily chosen to be five years from January 1, 2017 to January 1, 2022.

3.1.2 Price

All price data from this declared date range was collected via the Yahoo! Finance library ‘yfinance’ and contained the following standard, daily price features: opening price, low, high, closing price, total volume. The daily closing price for each security is shown in Figure 1.

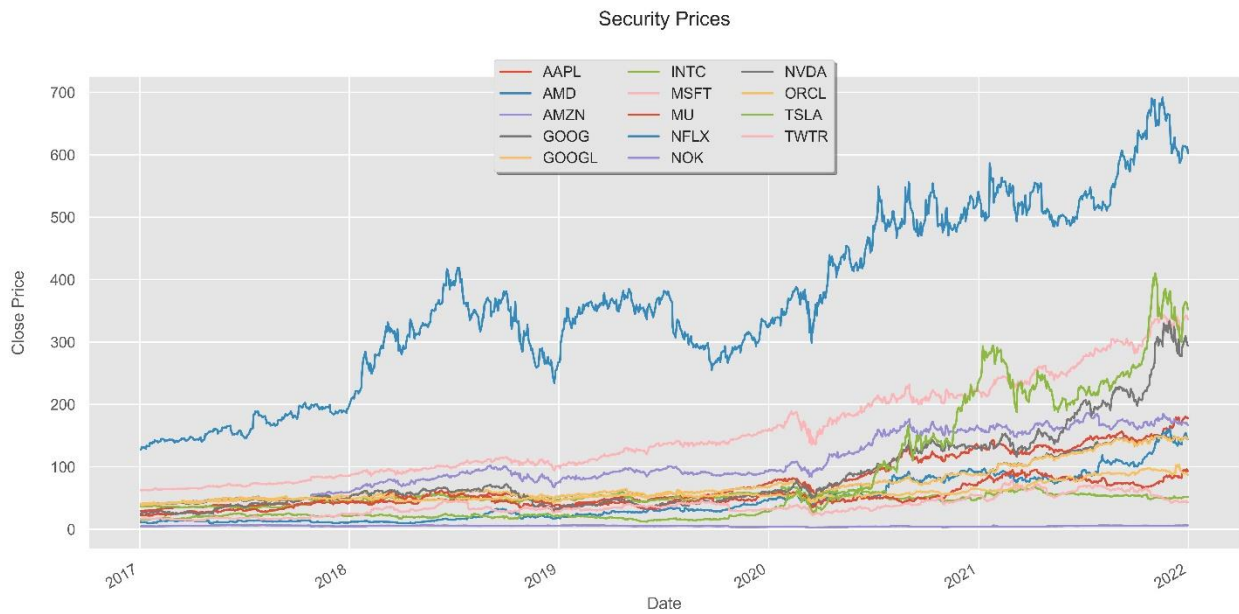


Figure 1: Chosen Securities' Prices

3.1.2.1 Outliers

Removing data points ± 3 standard deviations is characteristic of outlier removal; however, daily returns are typically not normally distributed. Daily returns were expressed as logarithmic returns to achieve an approximately normal distribution. Outlier removal was handled carefully, as these were usually the days that produced the highest profits and losses. Outliers may have been idiosyncratic (i.e., high-volatility) or market-wide (e.g., market crash) and potentially valuable to the dataset.

To identify idiosyncratic outliers, each security was isolated, and observations ± 3 standard deviations of the average logarithmic daily return were flagged (Figure 2). Subsequently, flagged values matching across all datasets were considered market-wide and removed from the set. Any remaining values were considered idiosyncratic and removed from the dataset.

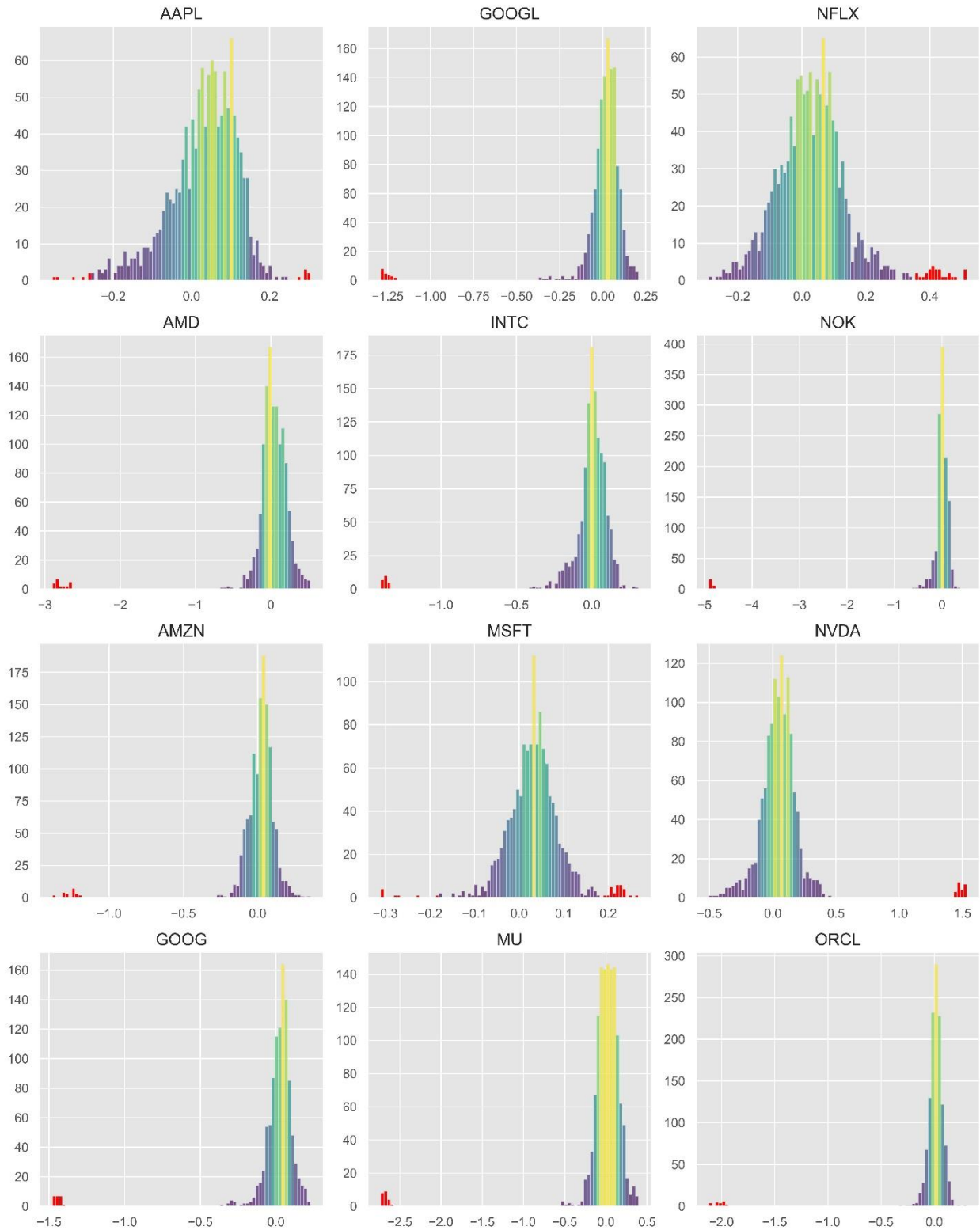


Figure 2: Outliers

3.1.3 Text

3.1.3.1 News Headlines

News headlines were collected via the paid service NewsFilterIO (NFIO). NFIO accesses and indexes headlines from 25 news sources within 500ms of their creation. These articles were tagged using named entity recognition (NER), a technique used to identify and classify named entities into pre-defined categories such as name, company, or location. The results were documented by both date and tagged entity. All headlines for tagged entities corresponding to the 14 chosen securities were joined to their respective datasets.

3.1.3.2 Twitter Posts

Twitter posts, commonly called tweets, were collected via an archive search available through the Twitter API. A method available to academic API users, ‘full archive search’ allows the user to retrieve any tweet using customizable queries. Queries were customized to retrieve all tweets and relevant metadata (i.e., creation date, tags, mentions, like count, retweet count) for each of the 14 named securities.

3.1.3.3 Text Preparation

Textual data can be notoriously difficult to prepare for analysis due to its unstructured nature. Internet shorthand (such as common abbreviations), digital icons (i.e., emojis), symbols, capitalization, and punctuation typically reflect emotional cues but can be challenging to capture or translate correctly. Certain sentiment analyzers, such as Natural Language Toolkit’s (NLTK) valence aware dictionary and sentiment reasoner (VADER) are equipped to analyze symbols and attributes associated with social media text but are not complex enough to also capture the relationships found within financial text. As a result, the text could not be processed in its raw form.

All instances of internet shorthand (e.g., Lol, Brb, Gtg) were mapped to their expanded formats, and symbols were removed. Emojis were mapped to word embeddings found in NLTK Vader’s public lexicon. Punctuation and capitalization were removed from the text but reserved to intensify the magnitude of the predicted sentiment. NLTK’s Vader punctuation mapping heuristic was used to identify the magnitude values for punctuation and capitalization (Table 1).

Table 1: Lexicon Mappings

Key	Magnitude
‘!’	0.292 * Sentence_Sentiment
‘?’	+ - 0.180 Sentence_Sentiment
‘XXX’	+ - 0.733 Word_Sentiment

3.1.4 Target

The prediction target, indicating positive or negative monthly returns, was calculated by taking the sign of the monthly difference in close price. A prediction of 1 would mean that the stock’s closing price on day n+21 was higher than the closing price of day n, and conversely, a 0 would indicate that the stock’s closing price on day n+21 would be less than the closing price of day n.

3.2 Model Design

3.2.1 Bidirectional Encoder Representations from Transformers

Typically, one of the biggest challenges in supervised natural language processing (NLP) is providing enough data to a model for it to perform at a markedly good level. Large, labeled datasets exist but are often overgeneralized and unsuited for specific applications. When relevant

data is extracted from these large datasets for application-specific tasks, there are fewer observations, and consequently, the model's ability to learn suffers.

NLP models are typically pre-trained on large amounts of unlabeled text to develop a general understanding of language and then finely tuned on application-specific datasets for specialized tasks (e.g., language inference, paraphrasing, sentiment analysis). In pre-training, pre-BERT models developed a general understanding of language by reading left-to-right, right-to-left, or a combination of both in different operations to predict a masked word based on sentence context. However, BERT models are bi-directional and predict the masked word by simultaneously evaluating the strings of tokens on both sides of the masked word, giving the model a deeper understanding of sentence context and flow (Figure 3).

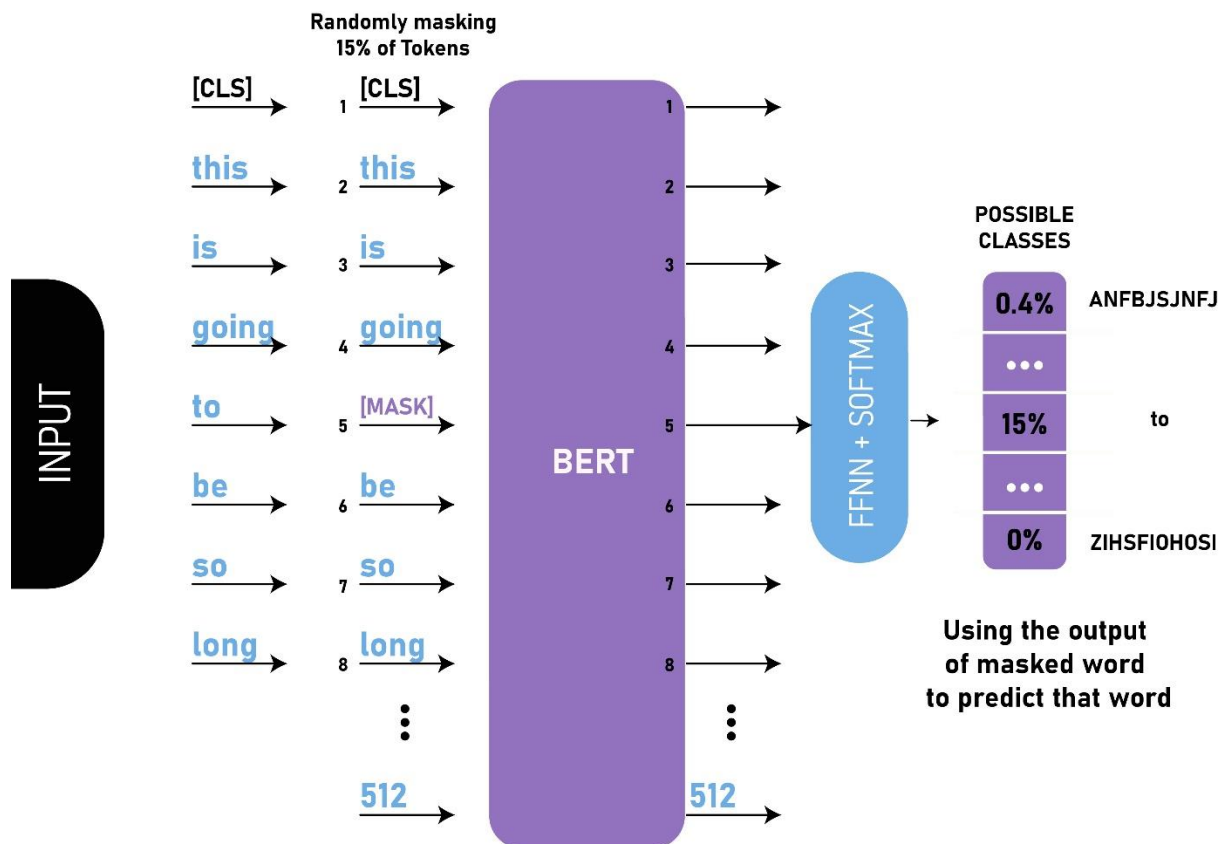


Figure 3: MLM [20]

BERT is also pre-trained to understand sentence relationships using a technique called next sentence prediction (Figure 4). Sentences are fed to the model in pairs: half of the time, the second sentence should follow the first, and half of the time not. Typically, these sentences are selected from a large corpus and chosen sequentially or randomly depending on the aforementioned cases [21].

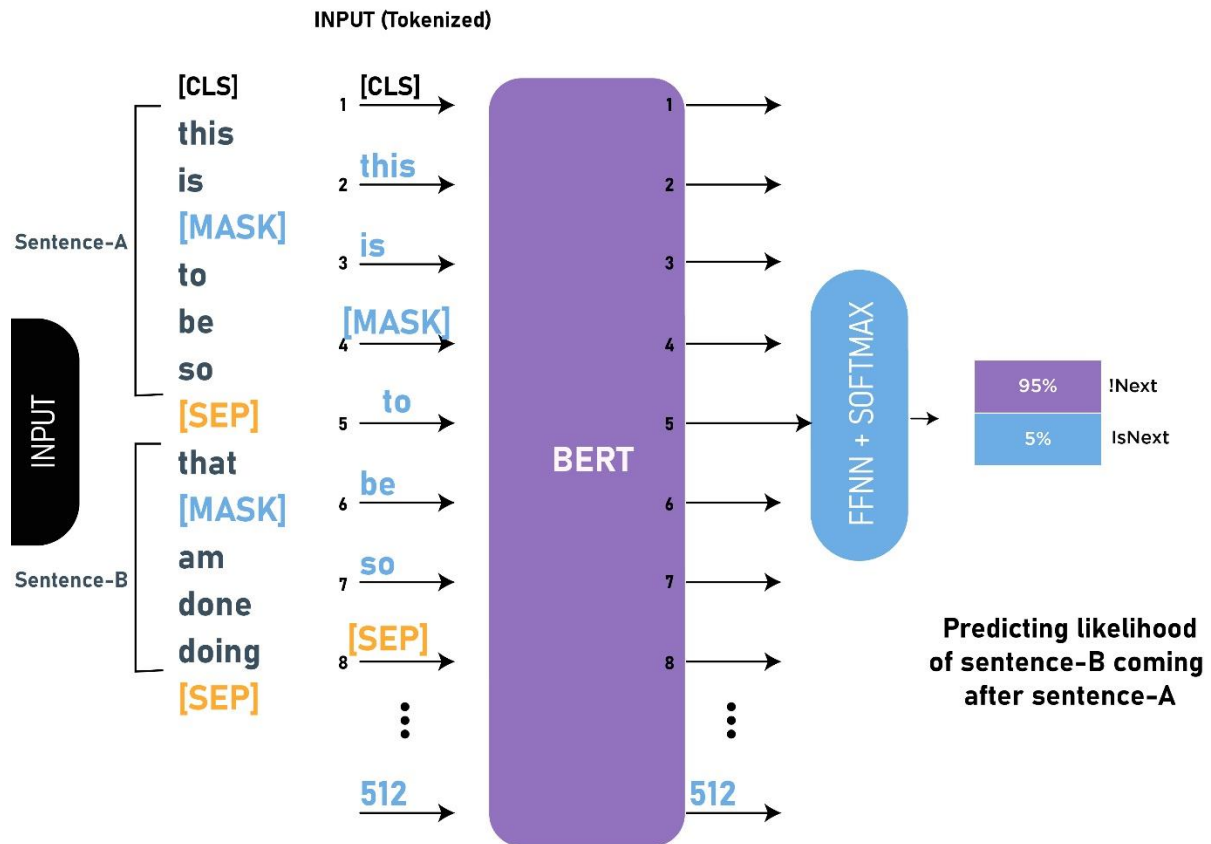


Figure 4: NSP [20]

3.2.2 Support Vector Machine

Support vector machines (SVMs) are powerful predictors commonly used in classification problems but may also be used for regression. In classification problems, SVMs attempt to separate the data via decision boundaries/surfaces that best isolate the labels. The separating vector(s) contain margins of error and become a hyperplane when referred to in totality. For linearly separable, 2-dimensional data, the hyperplane will be a line or set of lines. For non-linearly separable, 2-dimensional data, a 'kernel trick' is used to raise the data to a higher dimension; this generally assists in the separability of the data (Figure 5).

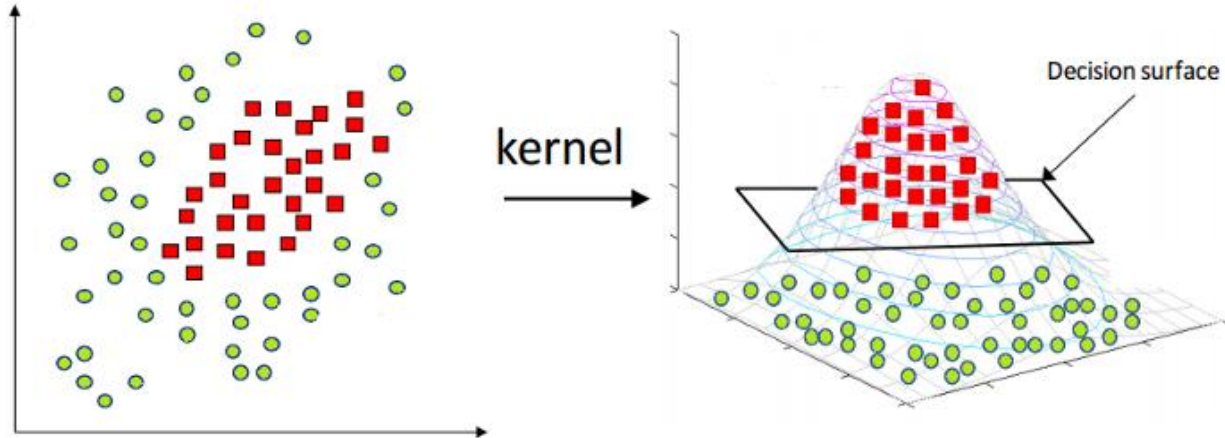


Figure 5: SVM Kernel Trick

A hyperplane is of dimension D , where D is a dimension one less than that of the ambient space. For example, 3-dimensional data will be separated by some 2-dimensional plane, and some 3-dimensional object will separate 4-dimensional data. For dimensions higher than four, visualization of the hyperplane is difficult or even impossible but mathematically describable [22].

3.3 Implementation

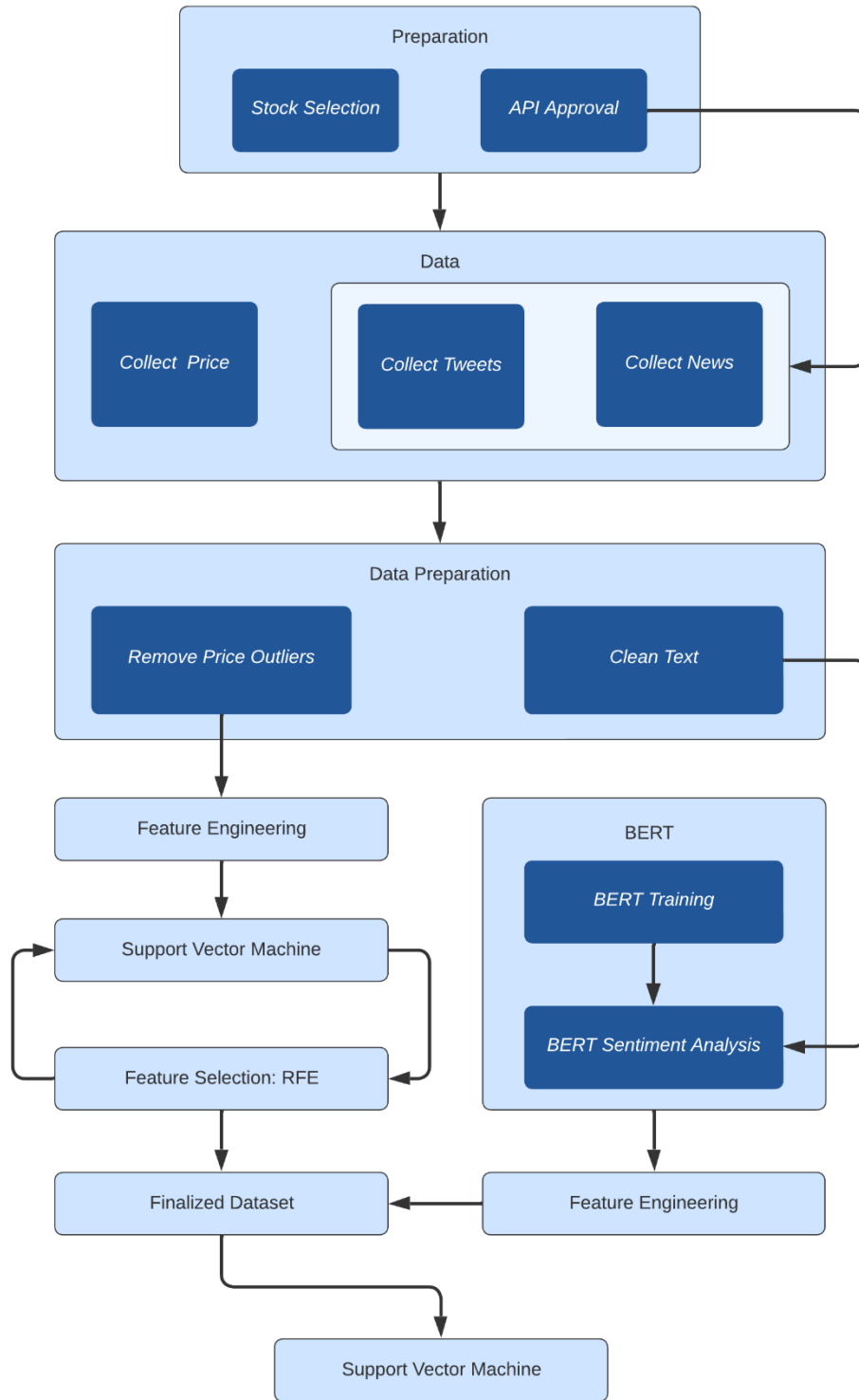


Figure 6: Implementation Flowchart

3.3.1 BERT Sentiment Analysis

Sentiment analysis for financial text is often a difficult task to perfect. Typically, numbers and symbols are removed as part of the text pre-processing step; however, for financial text, both numbers and symbols are a part of the vocabulary and contain important aspects of the overall sentiment (e.g., price change by +9%). As a result, removing either symbols or numbers could negatively impact model performance. BERT base model was finely tuned to solve this problem using the Financial PhraseBank [23].

The Financial PhraseBank is a dataset comprised of financial news headlines labeled by six licensed economists. Disagreement amongst the economists resulted in four variations of the training dataset (i.e., 25% agreement, 50% agreement, 75% agreement, and 100% agreement). Ideally, the model would have been trained using higher agreement variations of the Financial PhraseBank; however, observations were far fewer for these datasets and likely would not have contained adequate training data. The 75% agreement was chosen for fine-tuning to balance agreement and training size.

According to Malo et al. [23], disagreement among economists was the highest when determining whether a headline was negatively/positively charged or neutral. This resulted in a largely imbalanced set (Figure 7).

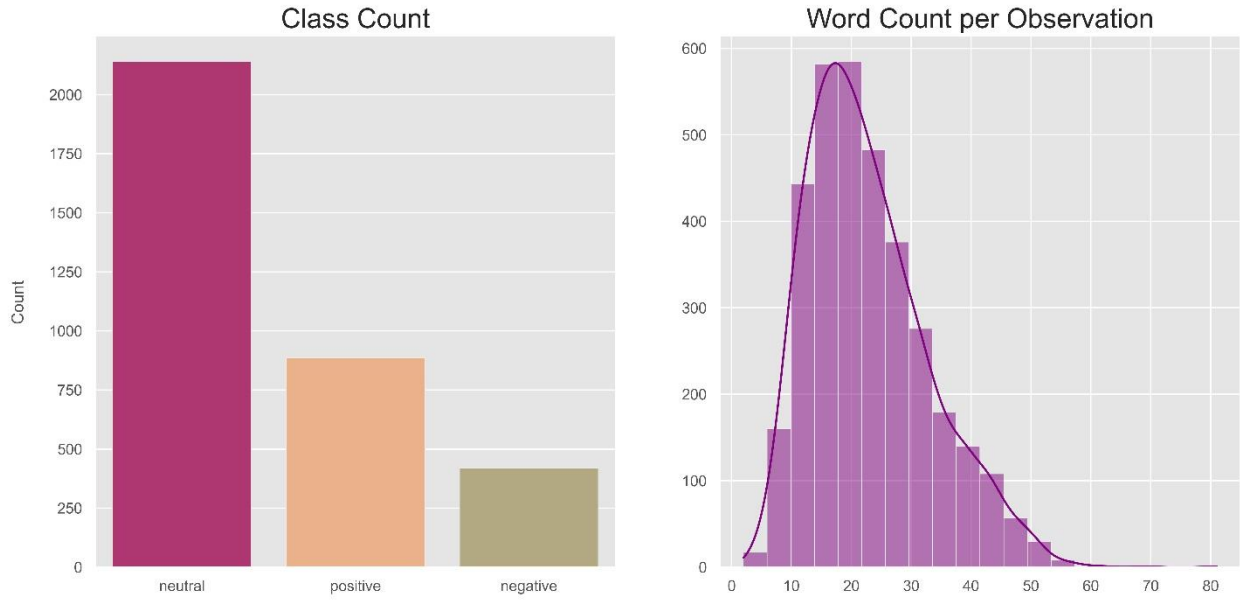


Figure 7: Imbalanced Dataset EDA

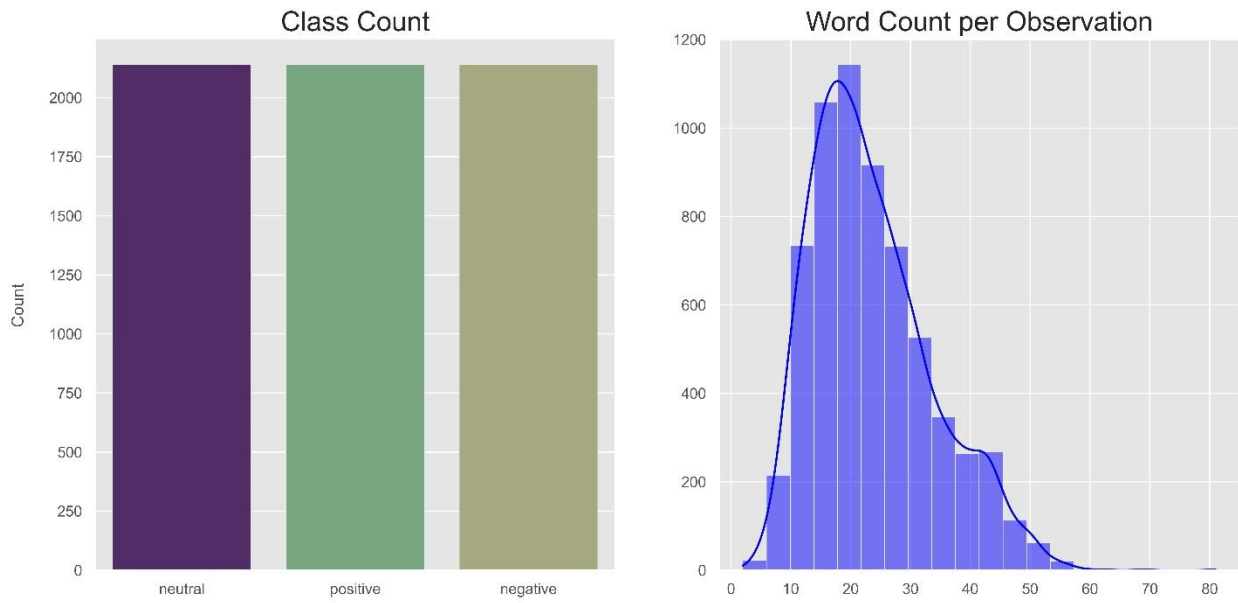


Figure 8: Balanced Dataset EDA

frequency bigrams and the relationships between them. Thicker edges represent bigram relationships found in higher-than-average frequencies.

The word cloud shown in Figure 9 indicated that the bigram ‘corresponding period’ was found in high frequencies for both labels, but the chord diagram shows that a deep relationship exists between ‘corresponding period’ and ‘profits rose’ for the positive label. Figures 9 and 10 show that complicated relationships do exist within each of the labels and may exhibit significant predictive power if the model could achieve a deep understanding of language and financial vocabulary.

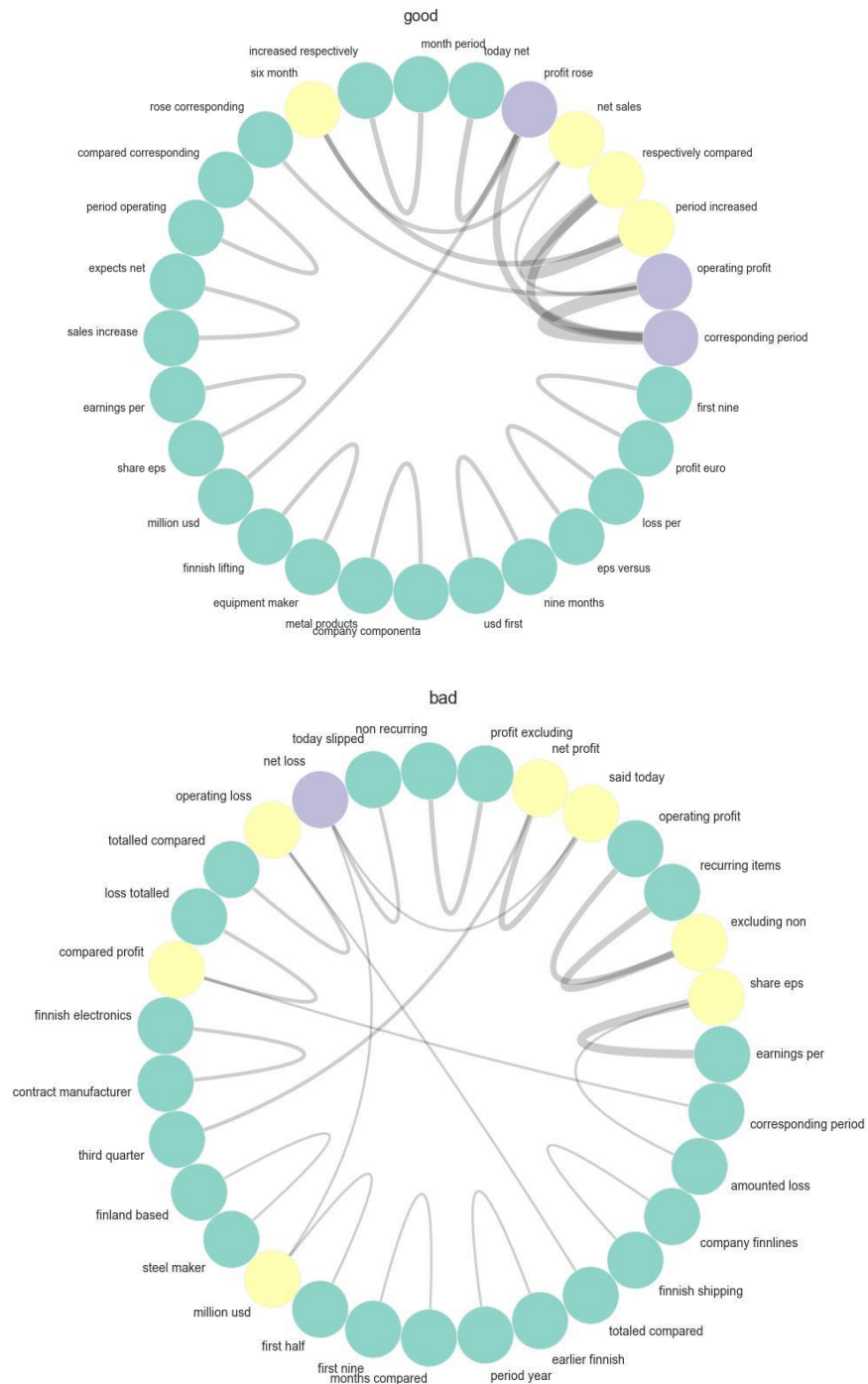


Figure 10: Chord Diagram

3.3.2 Feature Engineering

On its own, security price is typically not a useful feature. The high cardinality of price results in excessive noise and negatively impacts the model's ability to discover patterns.

However, feature engineering can obtain useful information (e.g., statistical transformations, feature crossing, bucketing, etc.).

3.3.2.1 Optimal Moving Averages

Optimal moving averages were calculated using a trend-following technique: a grid-searching method to identify slow- and fast-moving averages that best describe the data. For each search, the algorithm took a buy position when the slow-moving average crossed the fast-moving average from below and a sell position for the converse (Figure 11). Optimal slow- and fast-moving averages were identified upon maximization of the cumulative return (C.R.).



Figure 11: Trend Following

3.3.2.2 Momentum

Momentum quantifies both the speed and direction, or velocity, of security price change and is typically expressed in various rates of change. Momentum can indicate shifts in bearish or bullish behavior for a given period and may have significant predictive power. A list of momentum-derived features used to describe the data appear in Table 2 [24].

Table 2: Momentum Feature Descriptions

Feature	Description
Relative Strength Index	Magnitude of recent gains and losses over 21-days
Stochastic RSI	Strength and weakness of the relative strength indicator (RSI) over a 21-day
True Strength Index	Weighted average of security price momentum across 7, 14, and 28-day periods.
UL Oscillator	Weighted average of security price momentum across 7, 14, and 28-day periods.
Stochastic Oscillator	Difference of a security's close price relative to the highest high over a 21-day period.
Williams %R	Difference of a security's close price relative to the highest high over a 21-day period.
AO Oscillator	Rate of change between recent market momentum and market momentum using a 21-day look-back period
21D Rate of Change	21-day Rate of change (closing price)
Percentage Price Oscillator	Difference between fast, slow-moving averages as a percentage of the larger moving average
Percentage Volume Oscillator	Difference between fast, slow volume-based moving averages as a percentage of the larger moving average.
Kaufman Adaptive Moving Average	21-day moving average that becomes more sensitive during consistent trends and less sensitive during times of volatility

3.3.2.3 Volume

Volume measures the quantity of a security's financial asset (e.g., shares or contracts) exchanged between buyers and sellers over a given period. Volume can measure liquidity, supply/demand, and market strength. Securities trading at low volumes exhibits volatile behavior due to the number of buyers and sellers not being able to agree on a fair price. When combined with price features, volume can indicate a trend by evaluating how buyers and sellers react to current market conditions. A list of volume-derived features used to describe each security is shown in Table 3 [24].

Table 3: Volume Feature Descriptions

Feature	Description
On Balance Volume	Volume flow relative to price
Chaikan Money Flow	Amount of Money Flow Volume over a 21-day period
Force Index	Amount of power needed to move the price of an asset
East of Movement	Relationship between asset price 21-day rate of change and volume
14-day EOM SMA	14-day ease of movement simple moving average
Volume Price Trend	Balance between a security's supply and demand
Volume Weighted Adjusted Price	Average daily security price based on both price and volume
Money Flow Index	Buying/Selling Pressure
Negative Volume Index	Relationship between price and institutional investor capital volume

3.3.2.4 Volatility

Volatility measures the mean dispersion, or uncertainty, in a security's returns. Highly volatile securities typically have large high-low dispersion and are considered riskier investments as considerable changes in security value can happen in short periods. Risk is associated with opportunities that quickly increase and/or decrease portfolio value. All volatility-derived features used to describe the data are shown in Table 4 [24].

Table 4: Volatility Feature Descriptions

Feature	Description
Standard Deviation (rolling)	Variation of current price from 21-day moving average
Keltner Channel	Variation of current price from recent high and low price
Median Variation	Variation of current price from median
Average True Index	Decomposed range of security price over 21-day period
Ulcer Index	Depth and duration of percentage drawdown in price from previous high (investment risk)

3.3.2.5 Trend

Trend measures the direction of a security price movement over a given period. It is commonly used to identify patterns for historical and current price behavior and is a popular trading strategy chartists use. Various methods are used to calculate trends, but in its simplest form, it can be represented by the sign of the securities return over a given period. Like most statistical features, trend direction is better represented over longer periods as noise is prevalent in the short term. A list of trend-derived features used to describe the data is shown in Table 5 [24].

Table 5: Trend Feature Descriptions

Feature	Description
MA Convergence Divergence	Strength of security price movement
TRIX	Triple exponentially smoothed 21-day rate of change moving average
Mass Index	21-day rate of change high-low spread
Detrended Price Oscillator	High-low spread cycle length
KST	Price momentum for 10-, 15-, 20-, and 30-day price cycles
Schaff Trend Cycle	Smoothened difference between fast- and slow-moving averages
Average Directional Movement	21-day rate of change in trend momentum
Commodity Channel Index	Current price relative to 21-day average price
AROON	Number of periods since price recorded 21-day high or low

3.3.3 Modified Recursive Feature Elimination

Recursive feature elimination (RFE) is a feature selection algorithm that recursive evaluates model performance and eliminates the weakest feature until a desired number of features remains [25]. The primary purpose of RFE, like principle component analysis (PCA), is to reduce the data dimensionality. However, a modified RFE approach can be used to reduce dimensionality and remove noise from the dataset.

Using modified RFE, the model was fit on all but one feature. If model performance increased, the temporarily removed feature was added back to the required feature pool. If model performance decreased or stayed the same, the temporarily-removed feature was added to a table of detrimental features. The feature that lowered model performance the greatest was removed.

This process continued until no features could be removed without negatively impacting performance. In total, five features were removed, and 82 features were retained.

3.4 Validation

3.4.1 Metrics

Classification accuracy was used to evaluate general model performance. Informally, accuracy describes how well the model is classifying across all labels. Formally, it is the ratio of correctly predicted labels to total labels (E.Q. 1). However, classification accuracy can distort true performance if the test set labels are imbalanced, and the cost of misclassification differs between classes.

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

Equation 1: Accuracy

Using precision and recall allow for a more refined evaluation of performance. Informally, precision describes how correct the model is across all positive class predictions. Formally, precision is the ratio of true positives (i.e., correctly predicted positive labels) to the sum of true and false positives (E.Q. 2). Recall describes how correctly the model can identify true positive labels. Formally, recall is the ratio of true positives labels to the sum of true positive and false negative labels (E.Q. 3).

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

Equation 2: Precision

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

Equation 3: Recall

The appropriate metric was identified by evaluating the cost of both false positives and negatives. Misclassification of a good investment (i.e., false negative) would have translated to a missed opportunity but no decline in portfolio value. Conversely, misclassifying a bad investment (i.e., false positive) would have resulted in portfolio value loss. As a result, precision was chosen as the more relevant metric.

3.4.1 BERT: Cross-Validation

Cross-validation is a sampling technique used to assess how well a model will perform on unseen data. By taking multiple random or stratified samples, different training and validation sets can be built to evaluate average model performance (Figure 12). If model performance is similar across all validation sets, the model is likely well generalized and can be fit on the entire dataset. If not, there are likely problems (e.g., overfitting, underfitting, data leakage) affecting the model's overall fit.

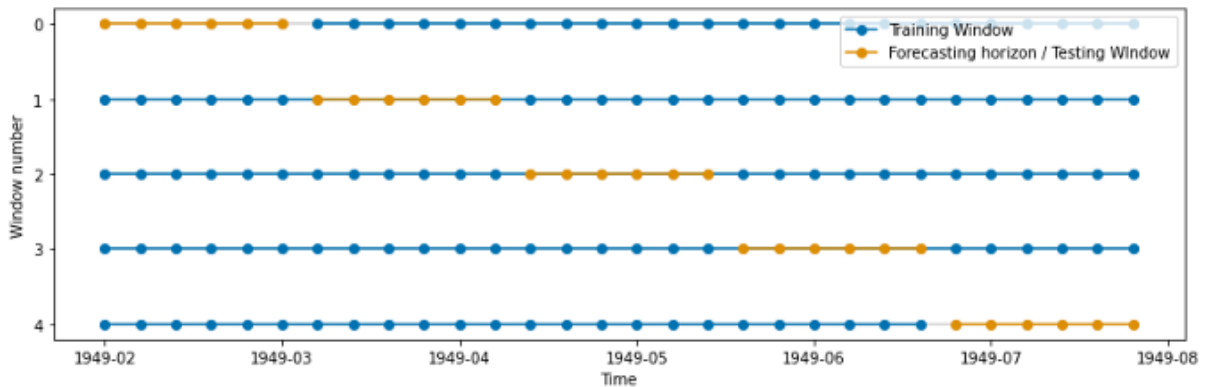


Figure 12: Cross-Validation

3.4.1 Support Vector Machine: Walk-Forward Validation

A different form of validation had to be used to evaluate average SVM performance. Had cross-validation been used, it would have introduced errors to the train and test sets. By taking random or stratified data samples, future values would be incorporated into the train set, allowing the model to look ahead (i.e., data leakage)—ultimately inflating model performance. A modified technique, walk-forward validation, was instead used in its place.

To build the test and train sets using a sliding window variation of walk-forward validation, a length-defined window was created and walked forward one day at a time (Figure 13). This method created the necessary data folds so model performance could be averaged across multiple fits while eliminating the possibility of data leakage.

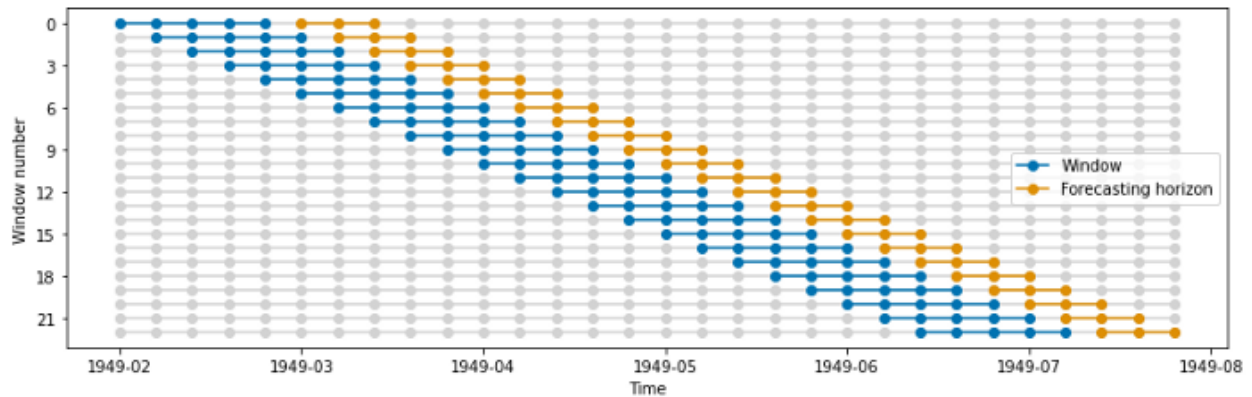


Figure 13: Walk-Forward Validation

CHAPTER 4. EXPERIMENTAL RESULTS

4.1 BERT

Classification accuracy for each fold was measured by dividing the number of correct predictions by the number of total predictions to give a face-value representation model performance. False positives (i.e., low sentiment predicted as high sentiment) held more weight as misclassified sentiment could have influenced similar behavior when used as support vector machine inputs, so precision and recall were also used to evaluate performance. Average model performance and precision/recall for each class using 5-fold cross-validation (Figures 14-18) is shown in Table 6.

With an average accuracy of 92%, the model performed exceptionally well across all folds. Fold 5 exhibited the highest accuracy at 95%, yet a range of only 4% shows that the model was consistent across all folds. The model was, on average, more precise when labeling the ‘neutral’ class. However, the result was inconsequential as ‘neutral’ sentiment was typically not a strong factor in target prediction.

Although precision was highest for the ‘neutral’ class, the model still does particularly well labeling both the ‘positive’ and ‘negative’ precisely. Slightly lower precision for both the ‘positive’ and ‘neutral’ classes was likely the result of oversampling. By oversampling, the dataset was balanced but at the cost of duplicate observation. As a result, there was less variation in the ‘positive’ and ‘negative’ training data classes for the model to learn from. Recall was highest for the ‘negative’ class-- an idyllic result. Misclassification of the ‘negative’ class could lead to portfolio loss if the model takes a trade based on positively labeled sentiment.

Table 6: BERT Results

Fold	Label	Precision	Recall
1	Negative	0.90	0.97
	Neutral	0.94	0.85
	Positive	0.89	0.93
Accuracy	91%		
2	Negative	0.87	0.98
	Neutral	0.97	0.87
	Positive	0.90	0.92
Accuracy	91%		
3	Negative	0.90	0.98
	Neutral	0.96	0.92
	Positive	0.92	0.88
Accuracy	93%		
4	Negative	0.90	0.95
	Neutral	0.92	0.93
	Positive	0.94	0.90
Accuracy	92%		
5	Negative	0.97	0.95
	Neutral	0.96	0.95
	Positive	0.92	0.97
Accuracy	95%		
5-Fold Average	Negative	0.91	0.97
	Neutral	0.95	0.90
	Positive	0.91	0.92
Average	92%	0.92	0.93

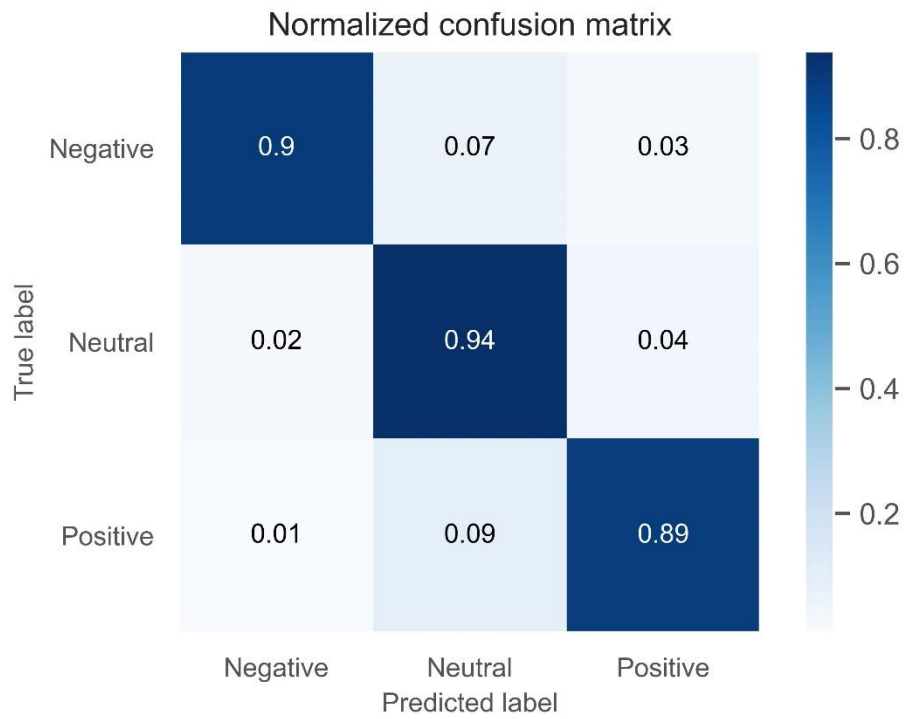
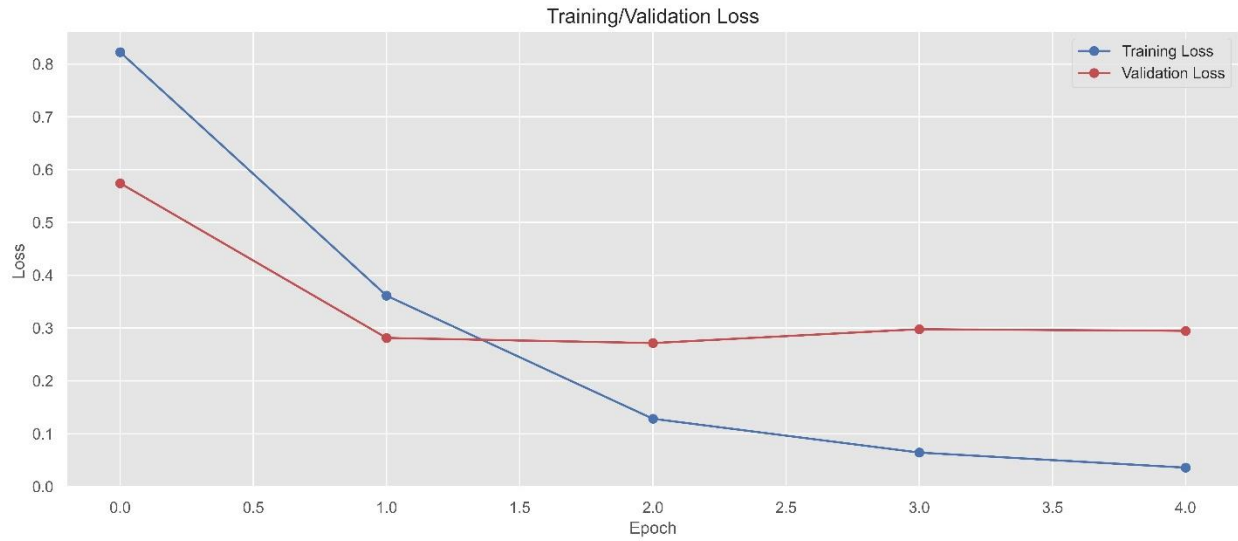


Figure 14: Fold 1 Results

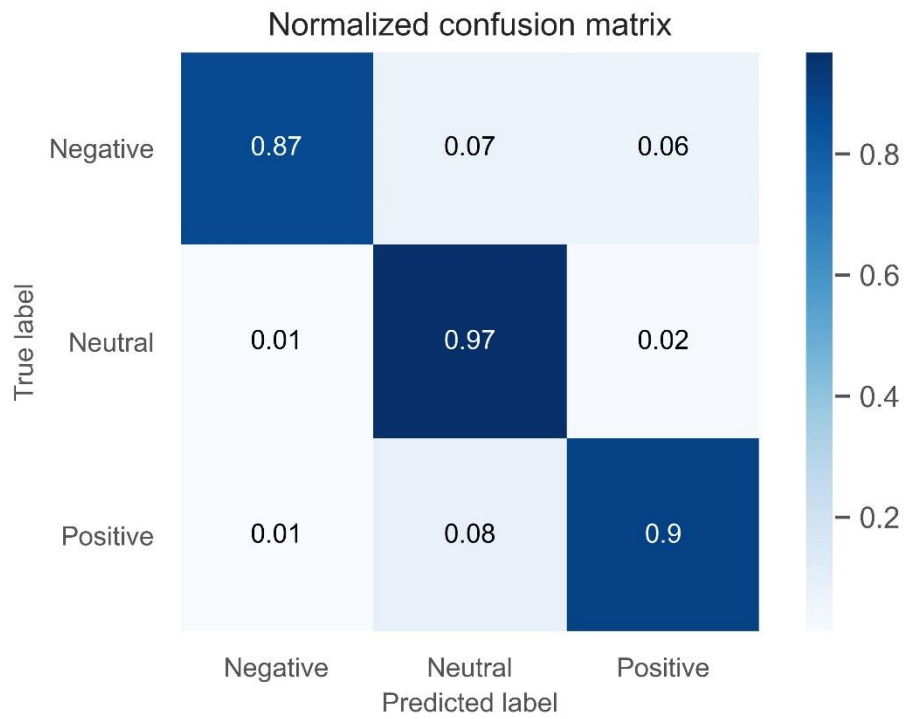
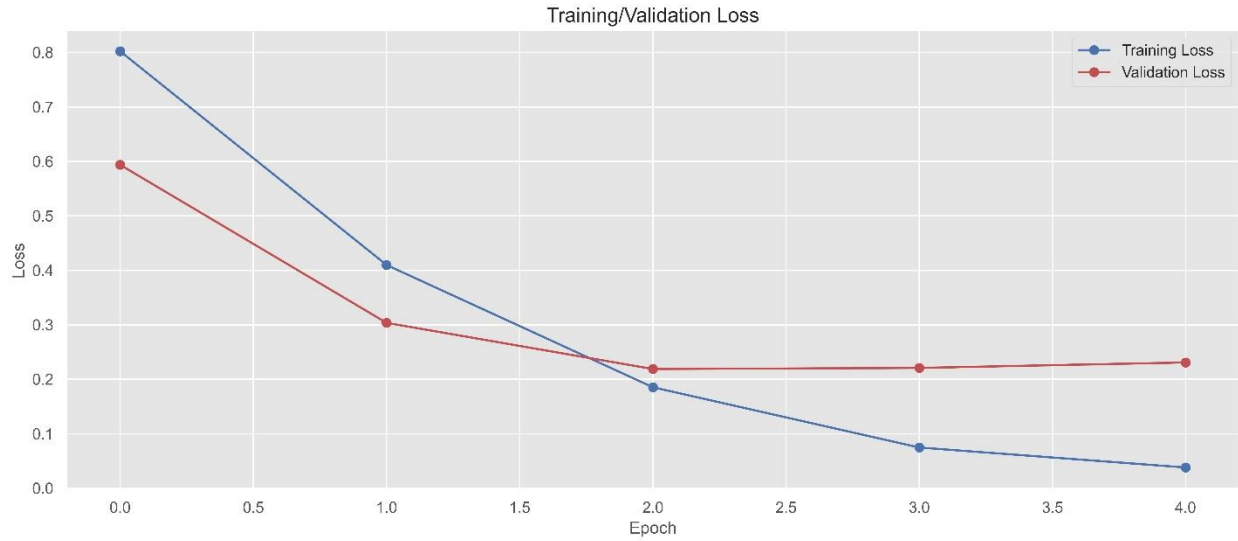


Figure 15: Fold 2 Results

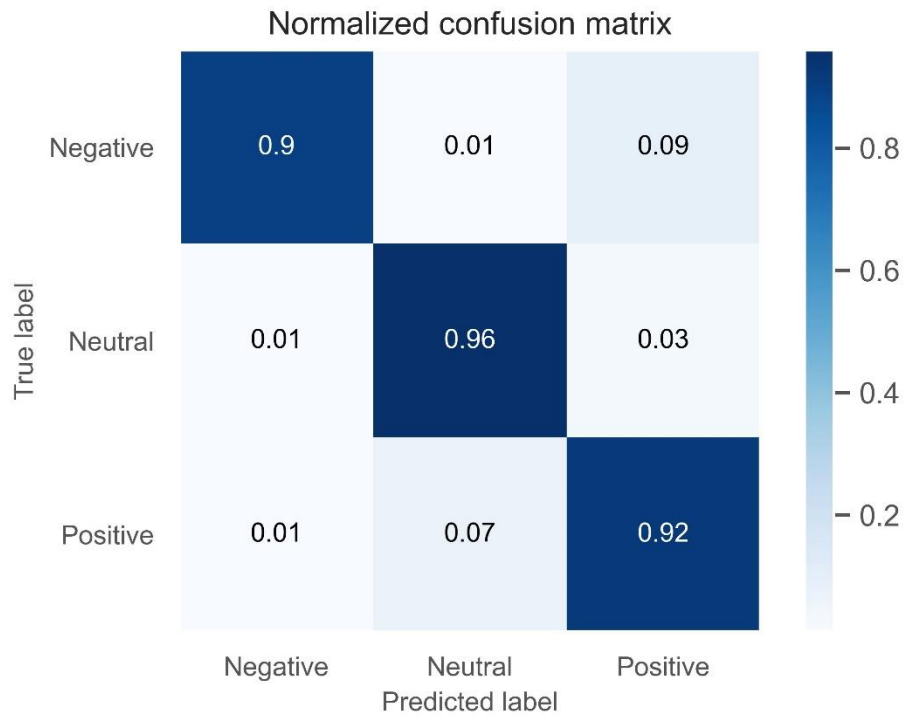
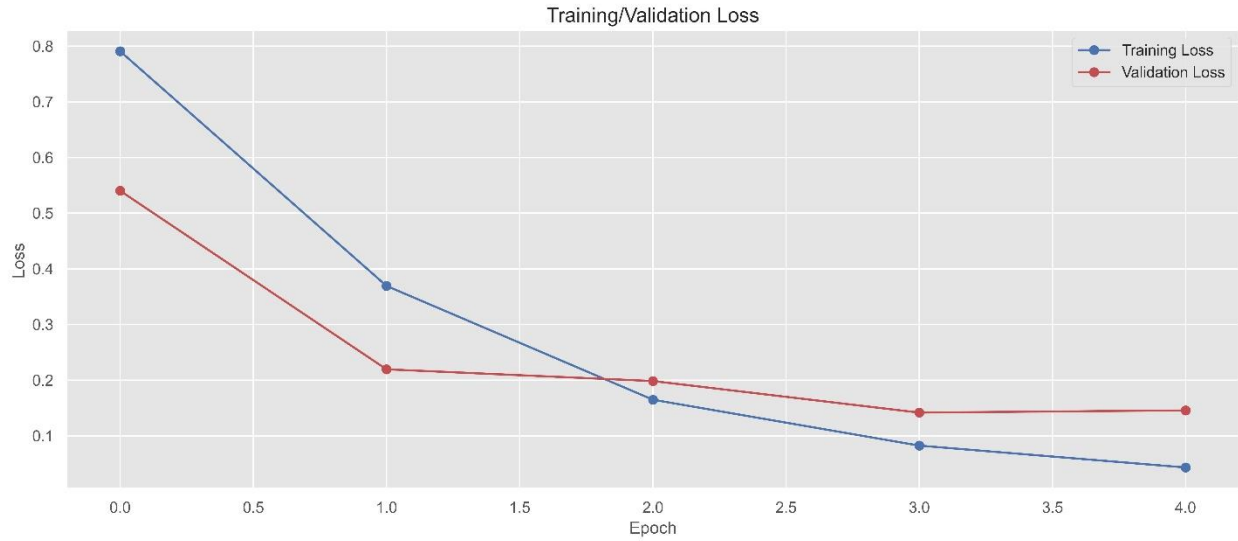


Figure 16: Fold 3 Results

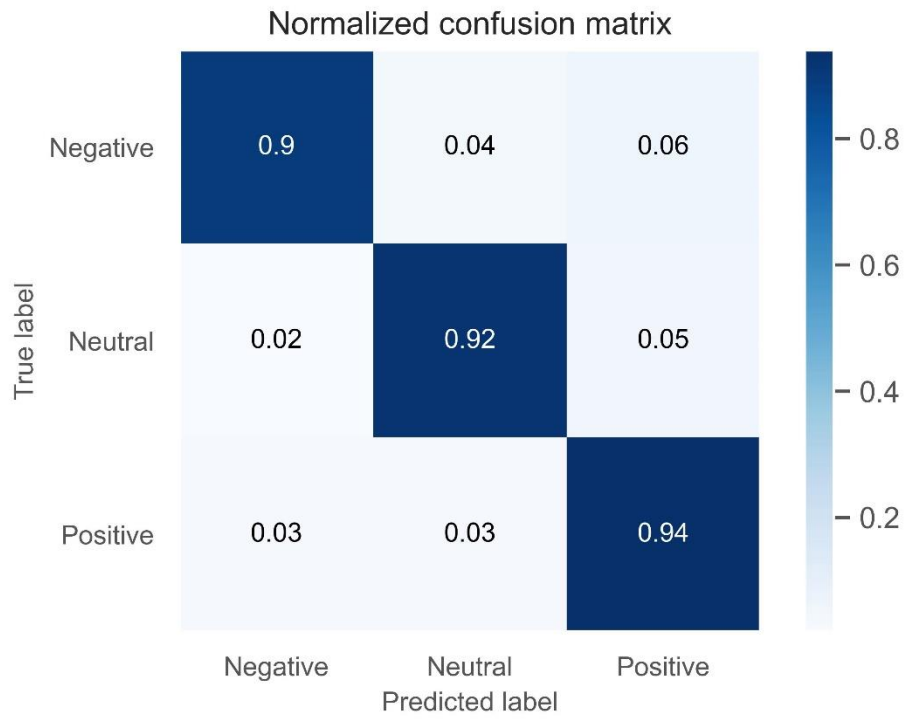
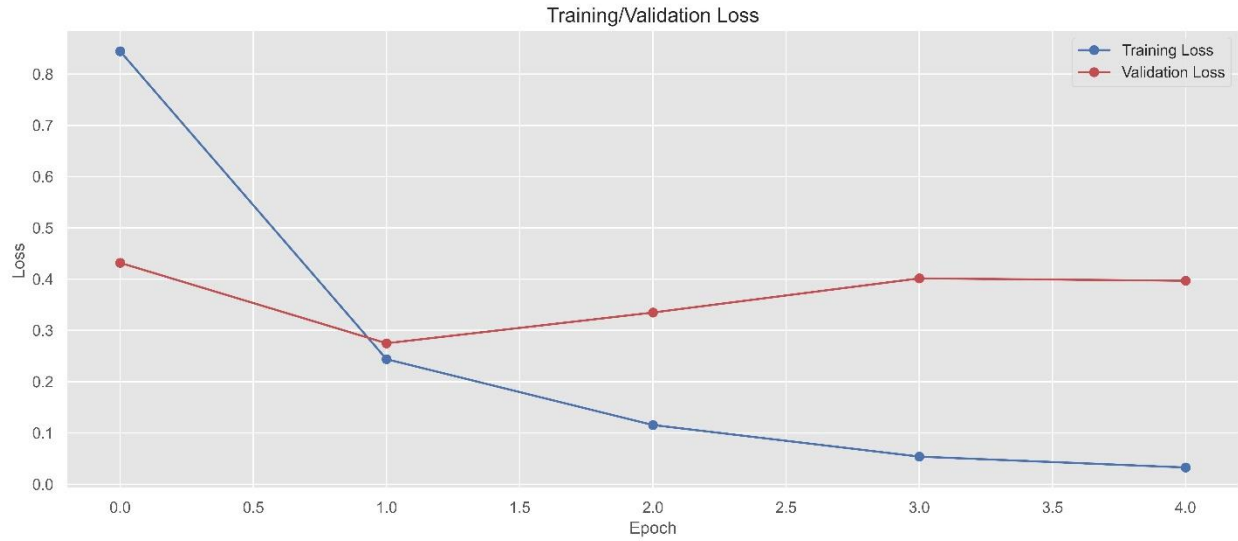


Figure 17: Fold 4 Results

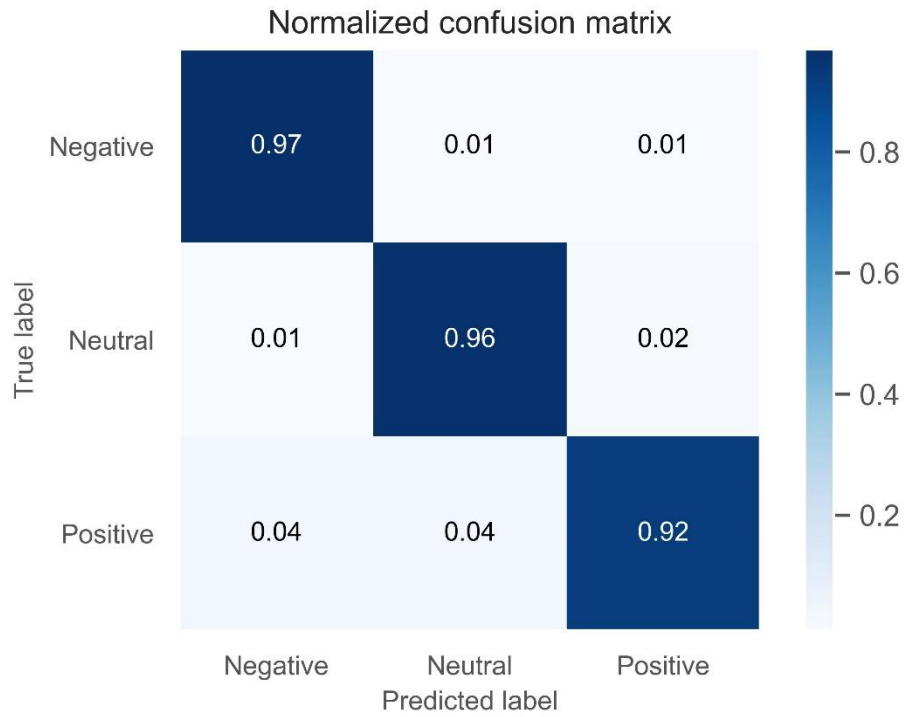
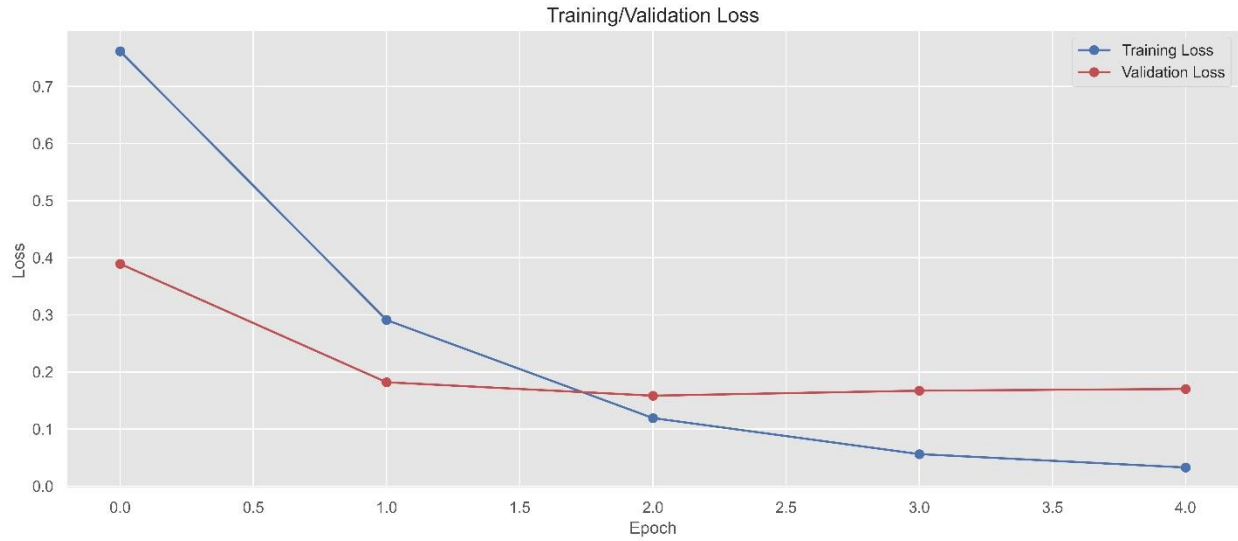


Figure 18: Fold 5 Results

4.1.2 Support Vector Machine

4.1.2.1 No Sentiment

The model performed reasonably well without sentiment, with an average classification accuracy of 64%; a deeper look showed that average precision and recall were 0.71 and 0.63 respectively (Table 7). This indicates that the model was typically better at not misclassifying good investments as bad investments but did not identify a fair amount of good investment opportunities. Although not a perfect result, high precision was paramount to avoid portfolio loss.

Table 7: SVM No Sentiment Results

Company (ticker)	Accuracy	Precision	Recall
Apple (AAPL)	0.60	0.61	0.59
Advanced Micro Devices (AMD)	0.75	0.60	0.86
Amazon (AMZN)	0.67	0.86	0.62
Alphabet (GOOGL)	0.61	0.89	0.57
Google (GOOG)	0.67	0.94	0.61
Intel (INTC)	0.58	0.62	0.57
Microsoft (MSFT)	0.57	0.48	0.58
Micron Technology (M.U.)	0.75	0.61	0.84
Netflix (NFLX)	0.66	0.95	0.60
Nokia (NOK)	0.56	0.52	0.56
Nvidia (NVDA)	0.63	0.62	0.63
Oracle (ORCL)	0.54	0.81	0.53
Tesla (TSLA)	0.61	0.52	0.63
Twitter (TWTR)	0.70	0.86	0.65
AVERAGE	0.64	0.71	0.63

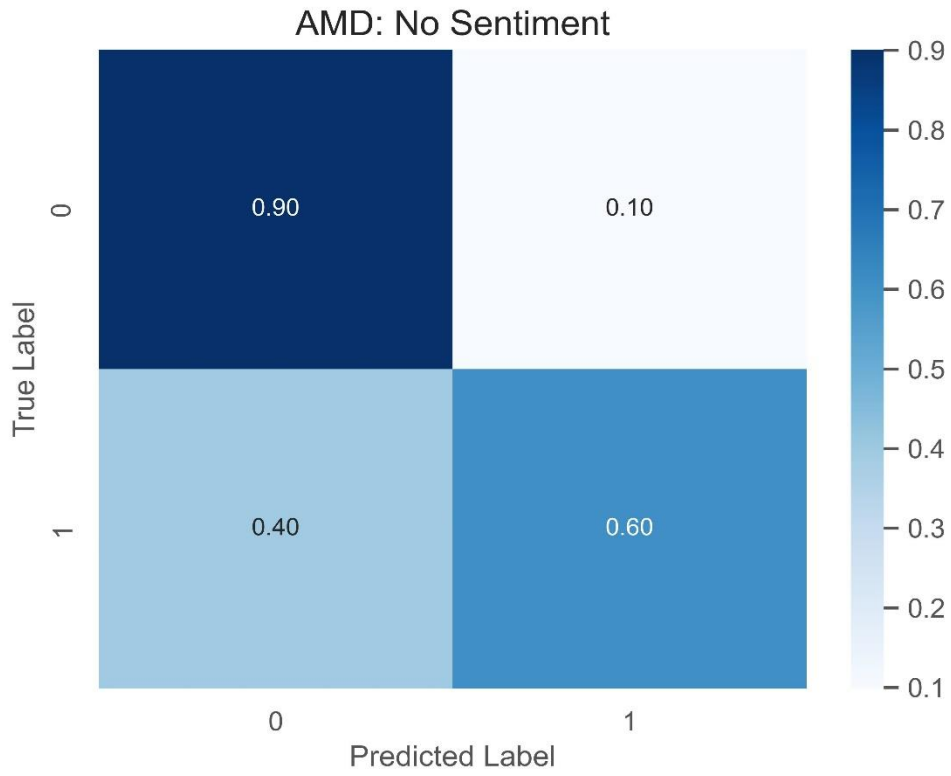


Figure 20: AMD No Sentiment CFM

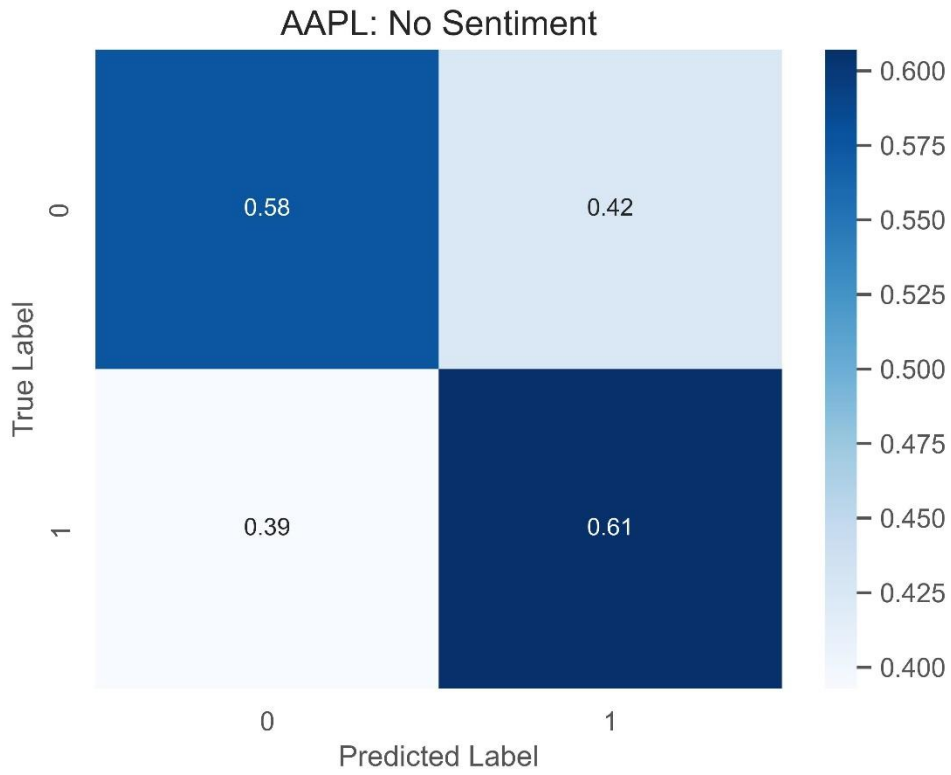


Figure 19: Apple No Sentiment CFM

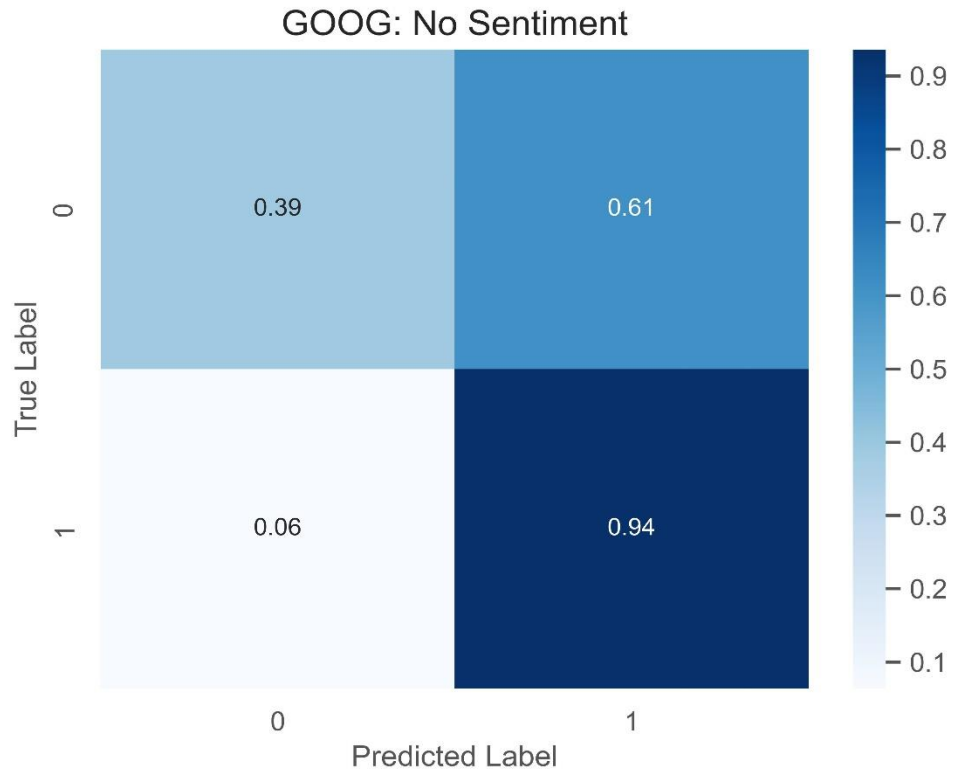


Figure 21: Google No Sentiment CFM

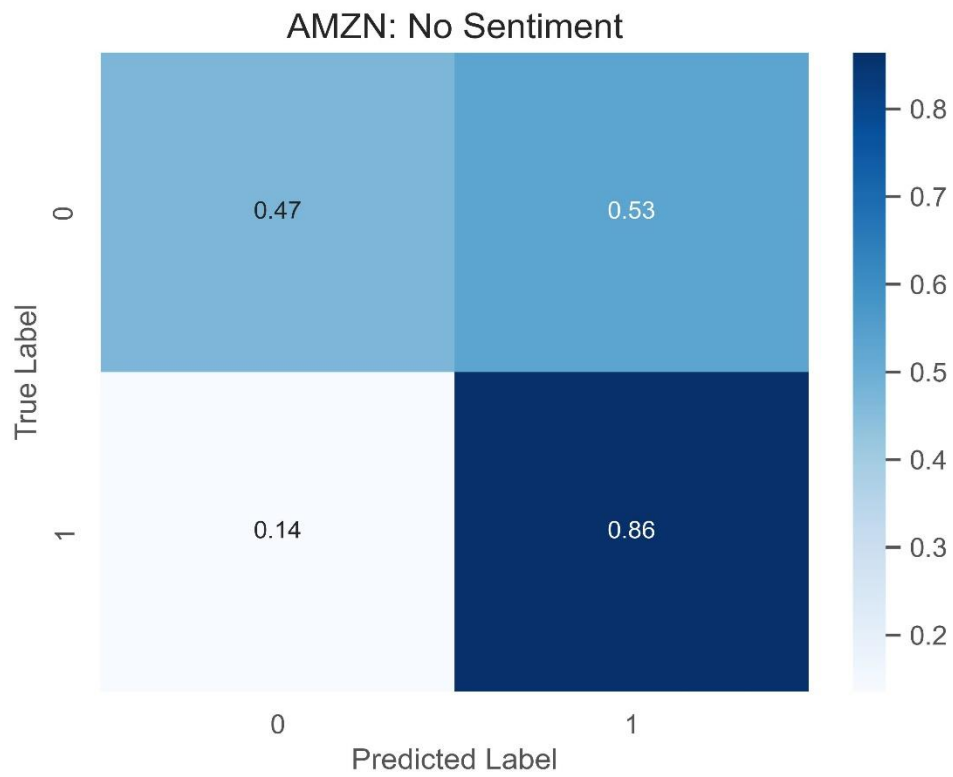


Figure 22: Amazon No Sentiment CFM

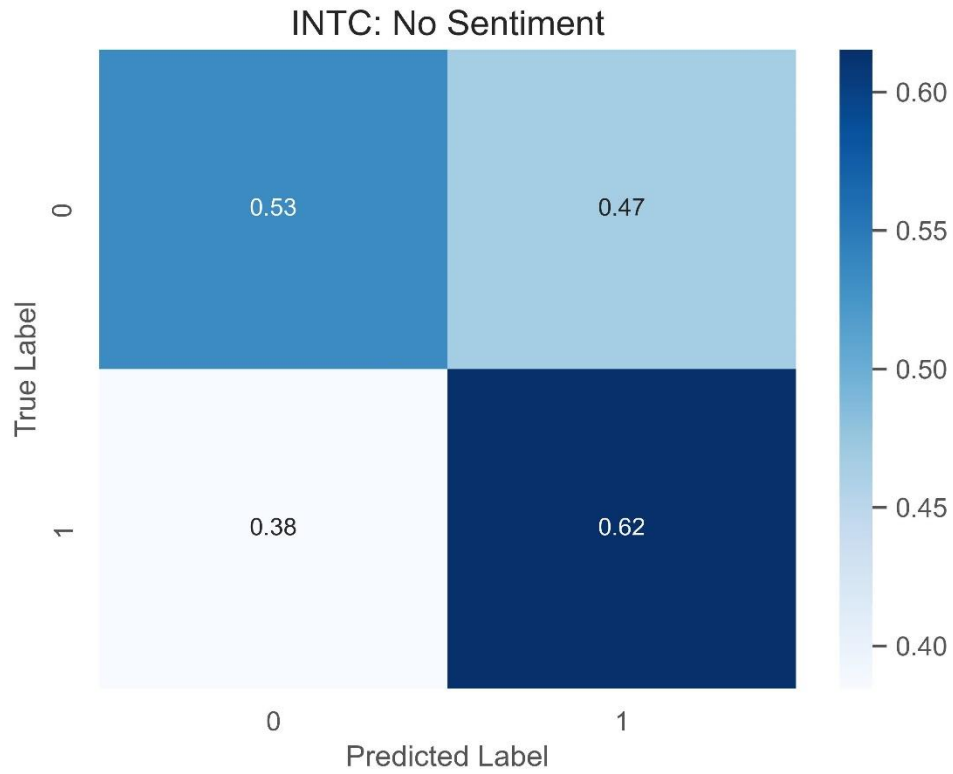


Figure 23: Intel No Sentiment CFM

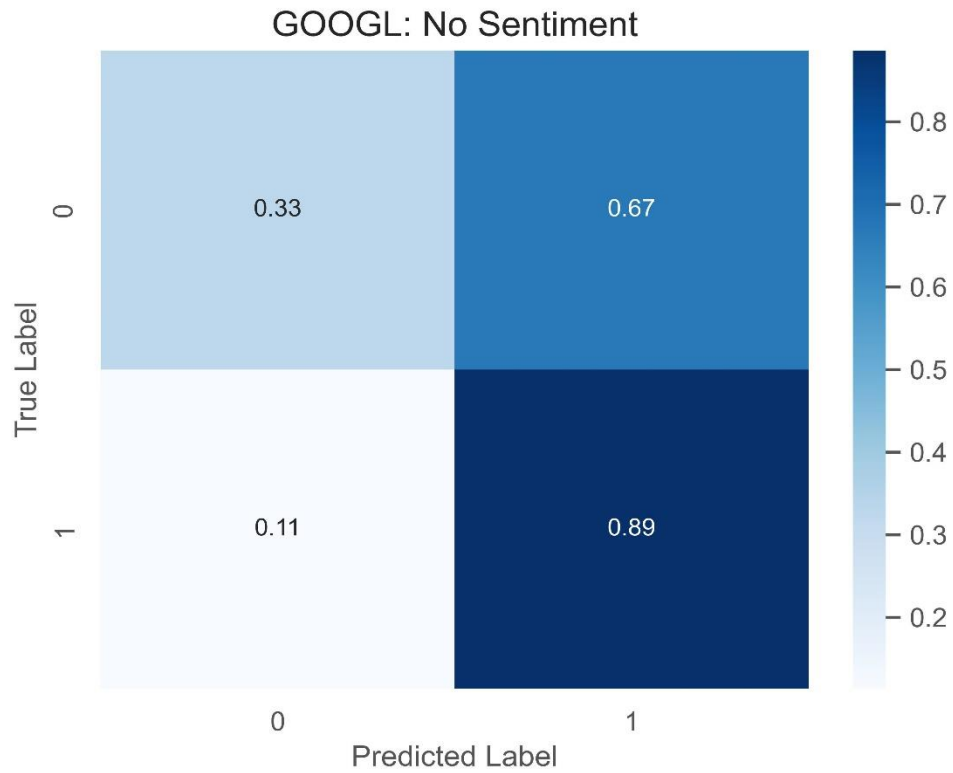


Figure 24: Alphabet No Sentiment CFM

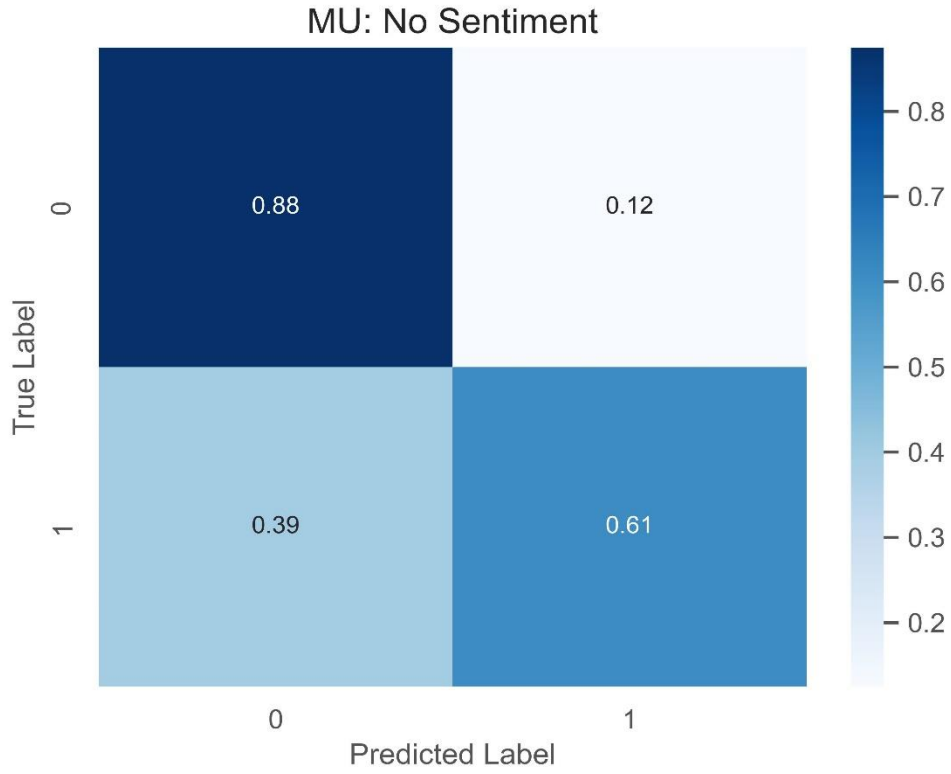


Figure 25: Micron No Sentiment CFM

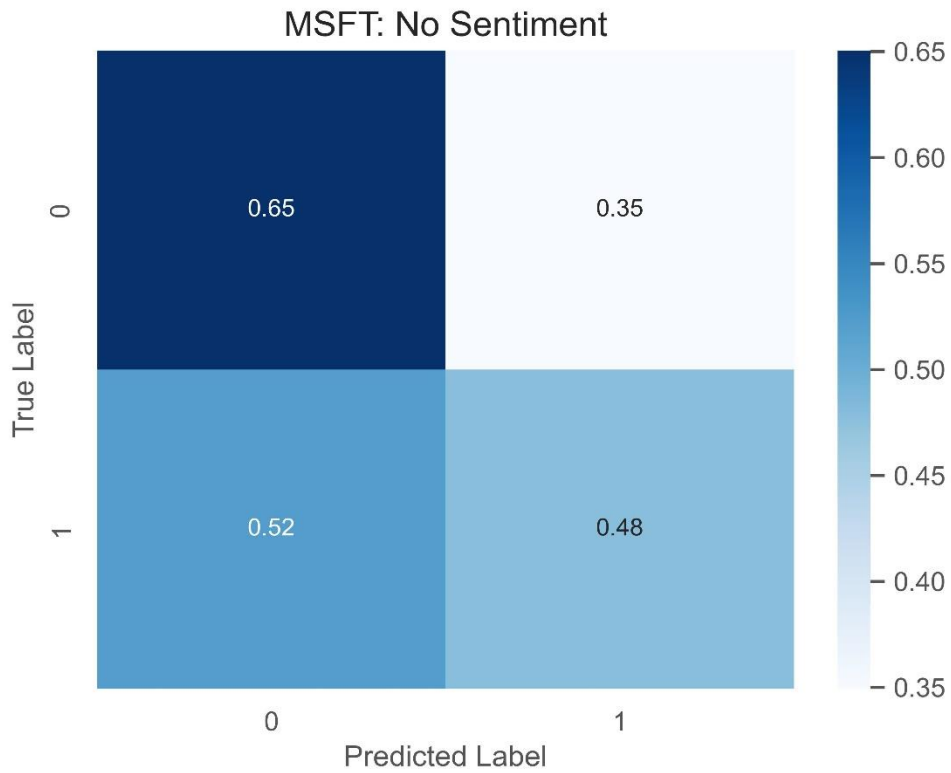


Figure 26: Microsoft No Sentiment CFM

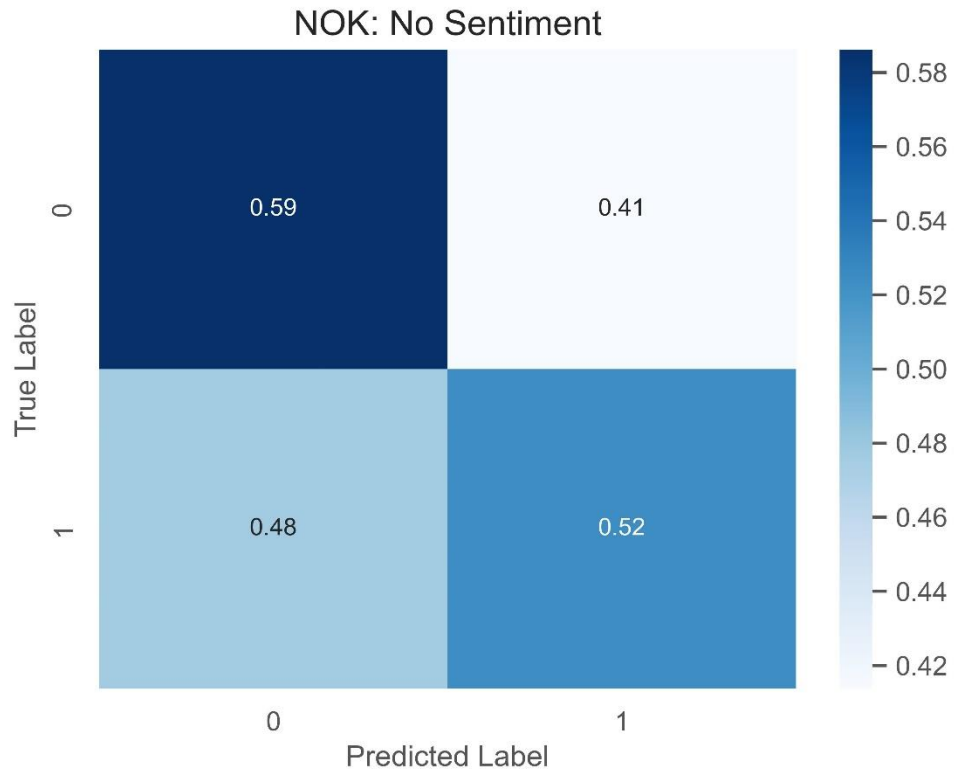


Figure 27: Nokia No Sentiment CFM

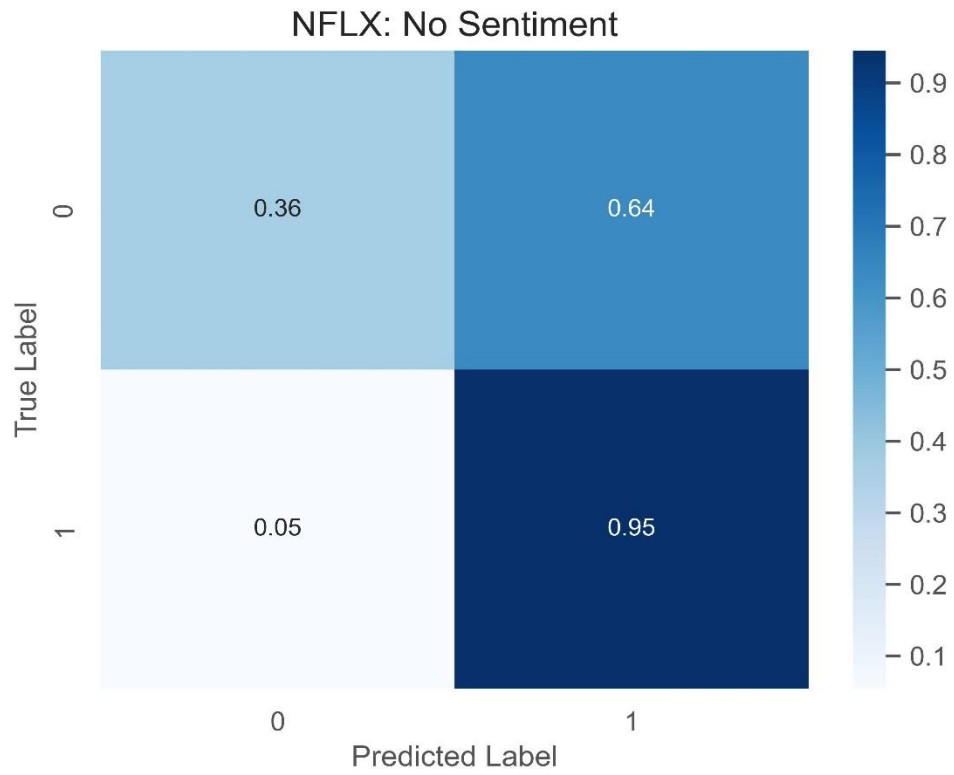


Figure 28: Netflix No Sentiment CFM

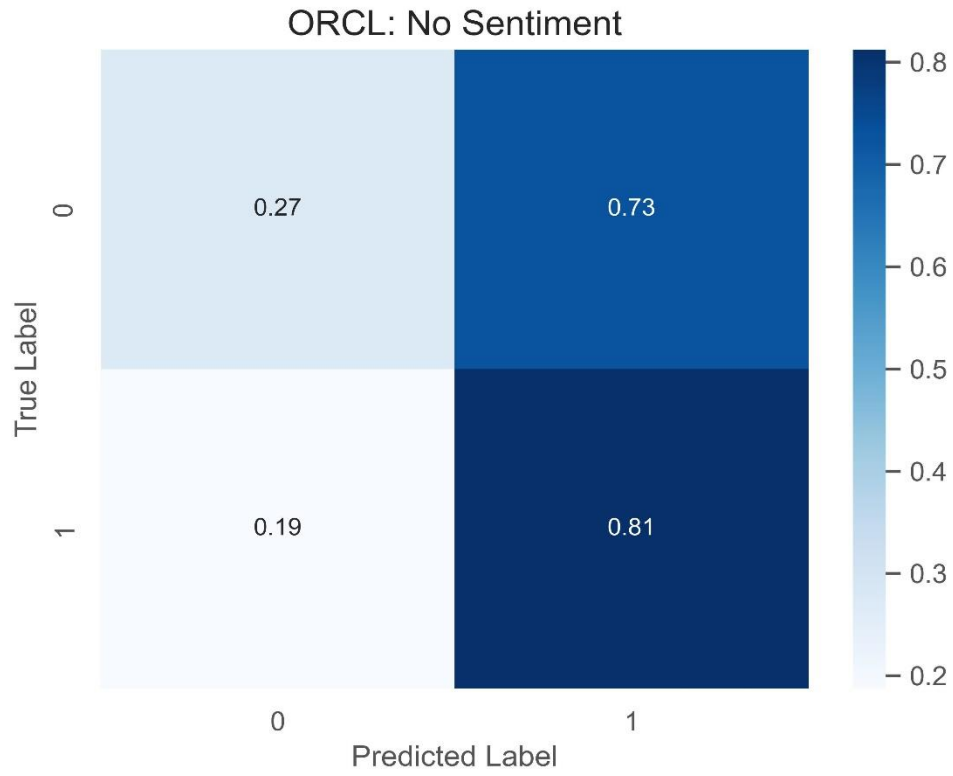


Figure 30: Oracle No Sentiment CFM

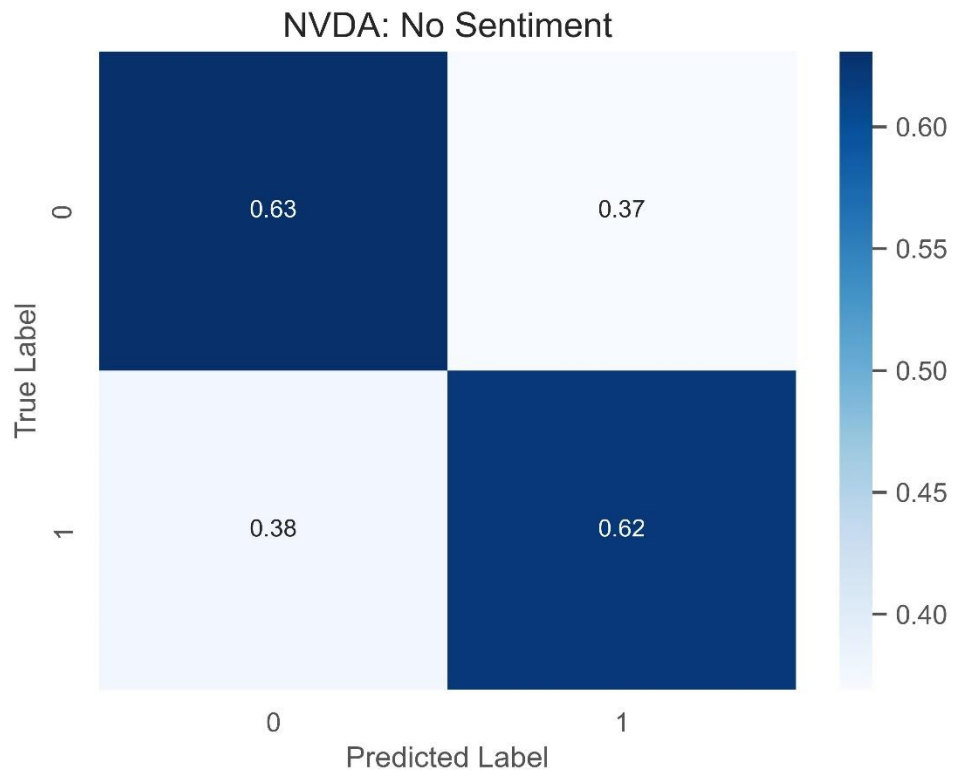


Figure 29: Nvidia No Sentiment CFM

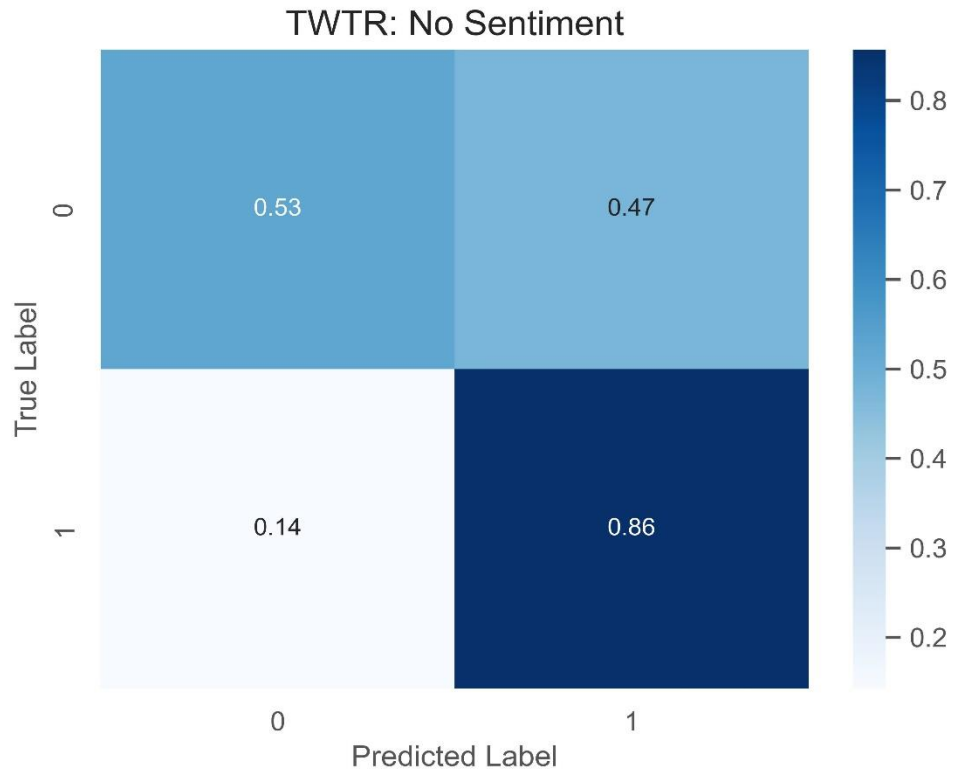


Figure 31: Twitter No Sentiment CFM



Figure 32: Tesla No Sentiment CFM

4.1.2.2 With Sentiment

Classification accuracy, precision, and recall for each security, in addition to average model performance for the dataset including sentiment are available in Table 8.

With sentiment included, the model's average classification accuracy was 66%, and precision and recall were 0.72 and 0.66, respectively. Precision was, yet again, higher than recall—indicating that the addition of sentiment did not negatively affect the model's ability to not misclassify good investments as bad.

Table 8: SVM Sentiment Results

Company (ticker)	Accuracy	Precision	Recall
Apple (AAPL)	0.60	0.61	0.59
Advanced Micro Devices (AMD)	0.78	0.62	0.91
Amazon (AMZN)	0.67	0.84	0.63
Alphabet (GOOGL)	0.66	0.91	0.60
Google (GOOG)	0.67	0.93	0.61
Intel (INTC)	0.62	0.64	0.61
Microsoft (MSFT)	0.61	0.50	0.63
Micron Technology (M.U.)	0.79	0.62	0.93
Netflix (NFLX)	0.67	0.97	0.60
Nokia (NOK)	0.59	0.55	0.59
Nvidia (NVDA)	0.63	0.61	0.63
Oracle (ORCL)	0.54	0.80	0.53
Tesla (TSLA)	0.65	0.53	0.69
Twitter (TWTR)	0.74	0.89	0.69
AVERAGE	0.66	0.72	0.66

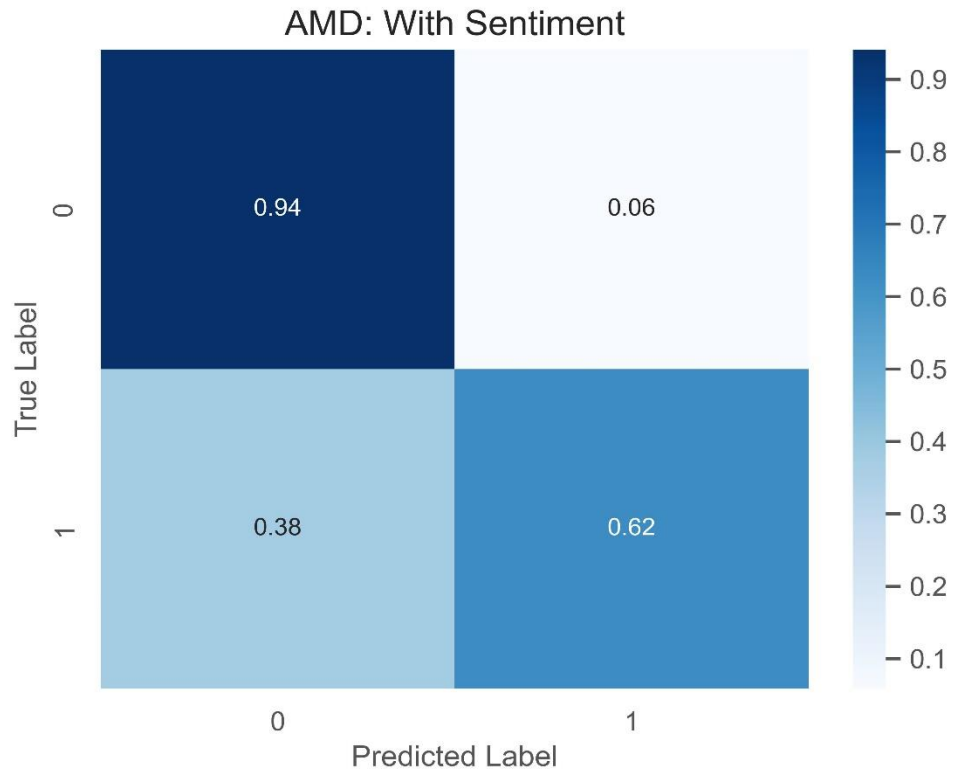


Figure 34: AMD Sentiment CFM

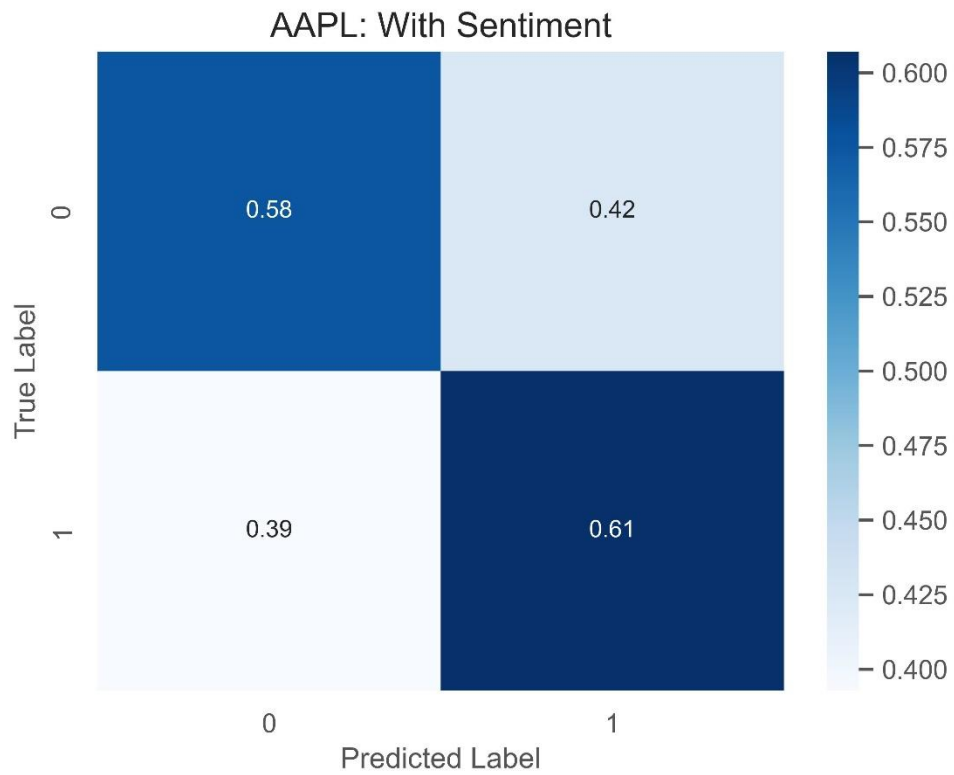


Figure 33: Apple Sentiment CFM

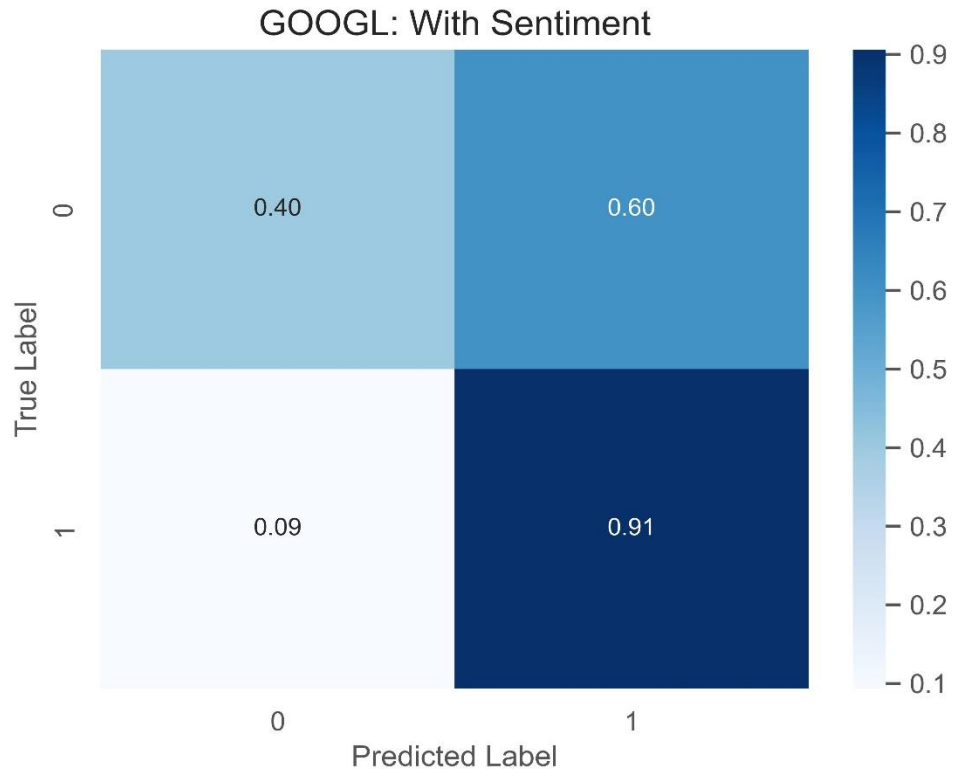


Figure 35: Alphabet Sentiment CFM

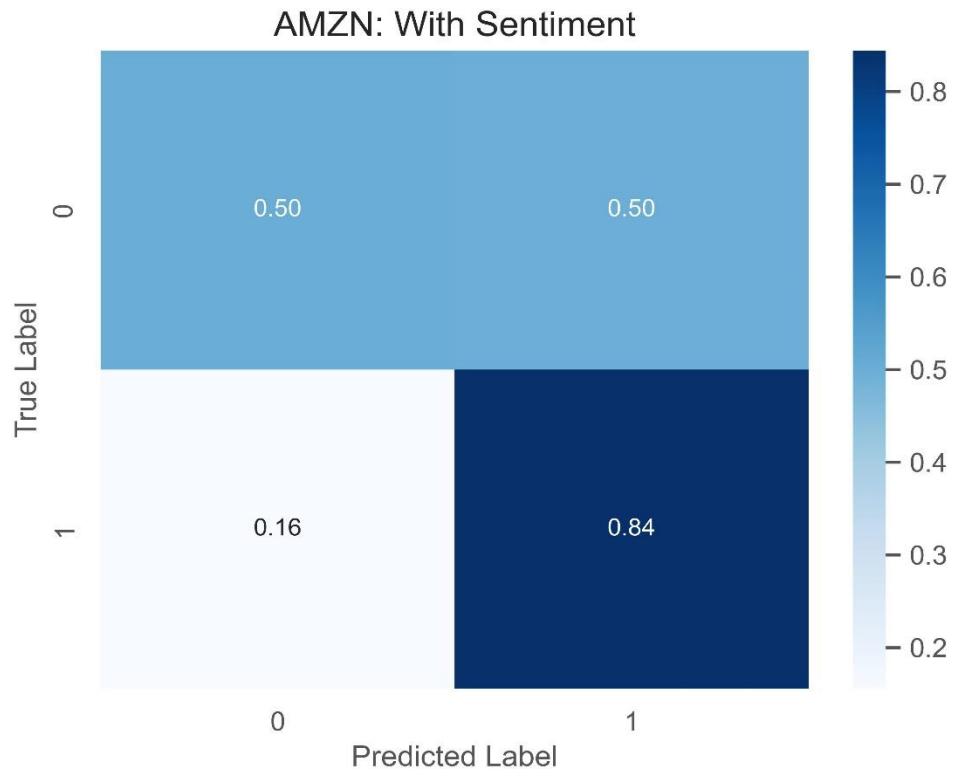


Figure 36: Amazon Sentiment CFM

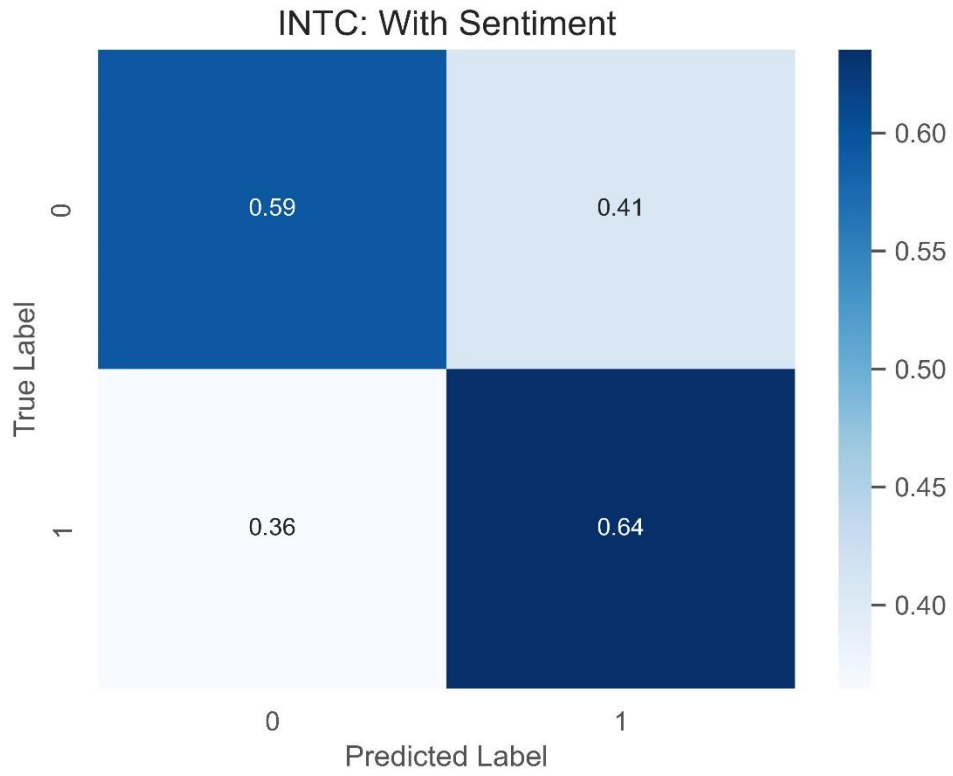


Figure 37: Intel Sentiment CFM

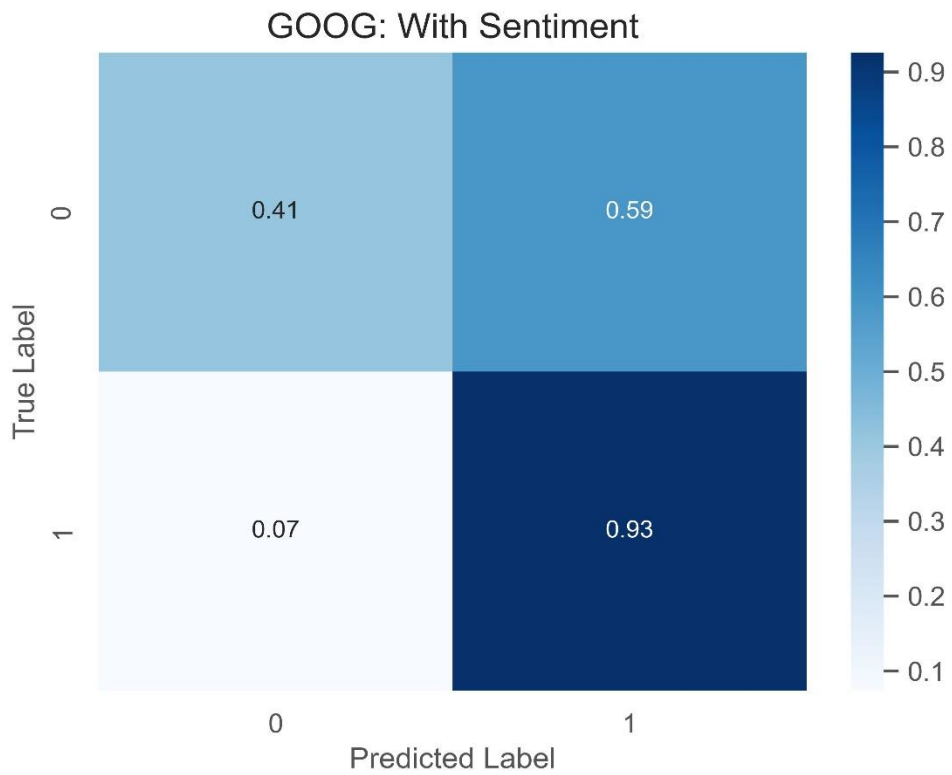


Figure 38: Google Sentiment CFM

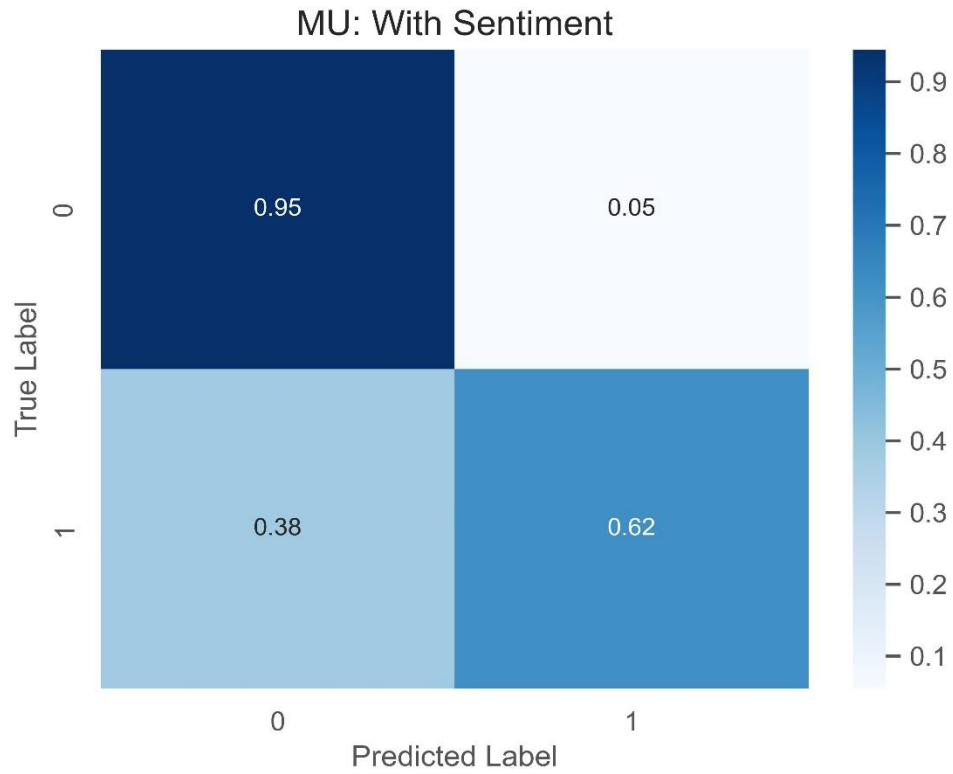


Figure 39: Micron Sentiment CFM

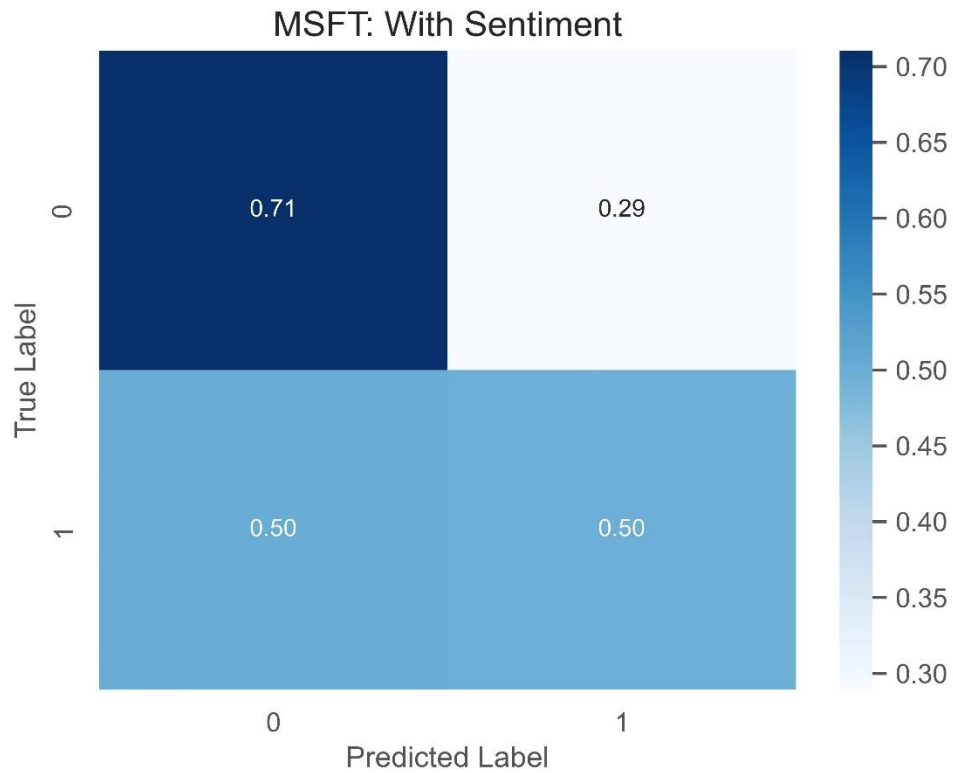


Figure 40: Microsoft Sentiment CFM

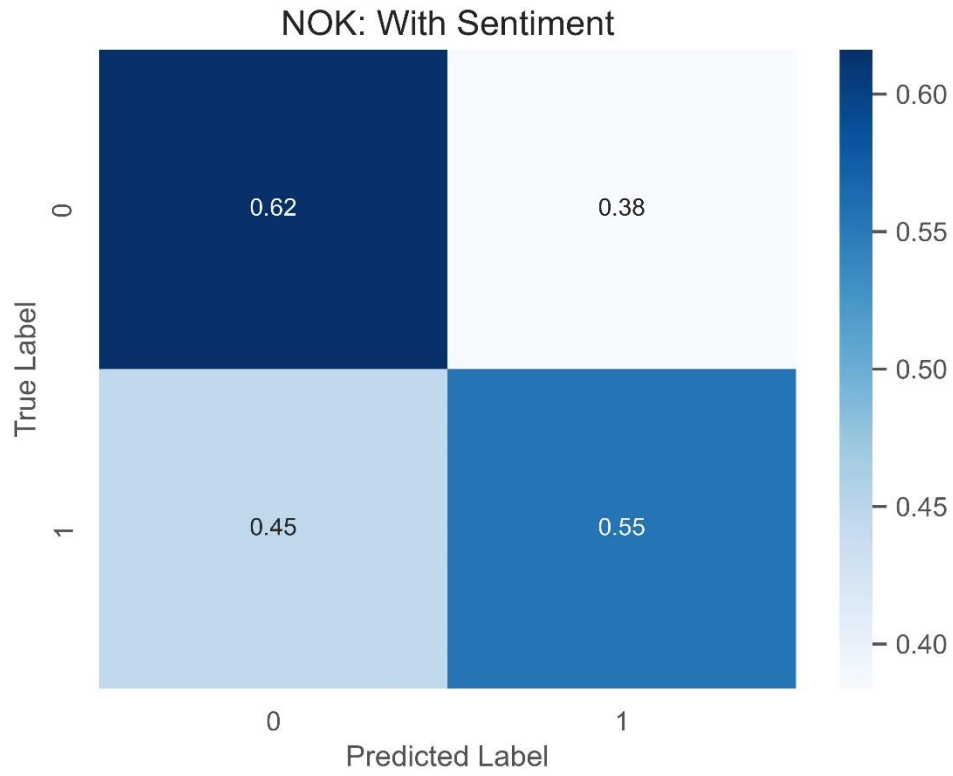


Figure 41: Nokia Sentiment CFM

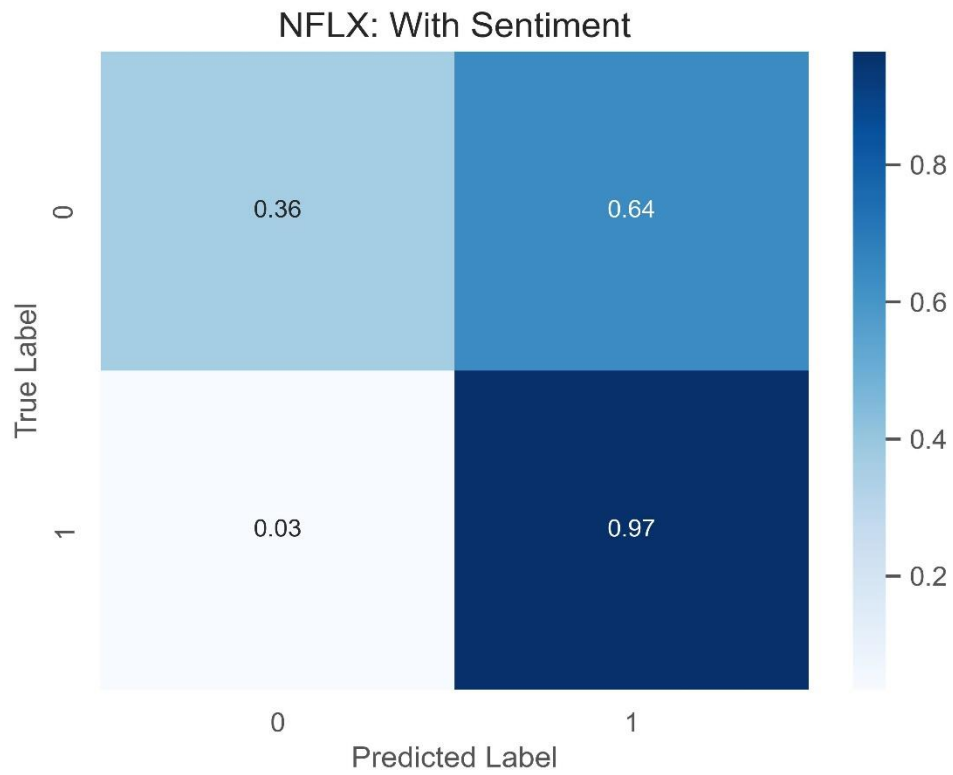


Figure 42: Netflix Sentiment CFM

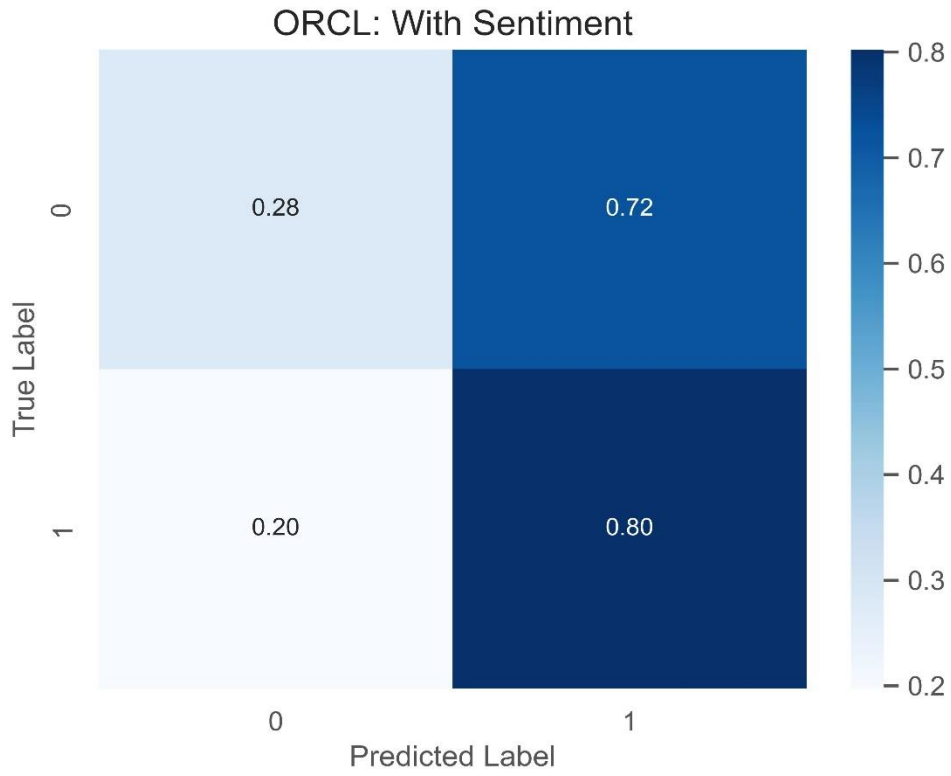


Figure 43: Oracle Sentiment CFM

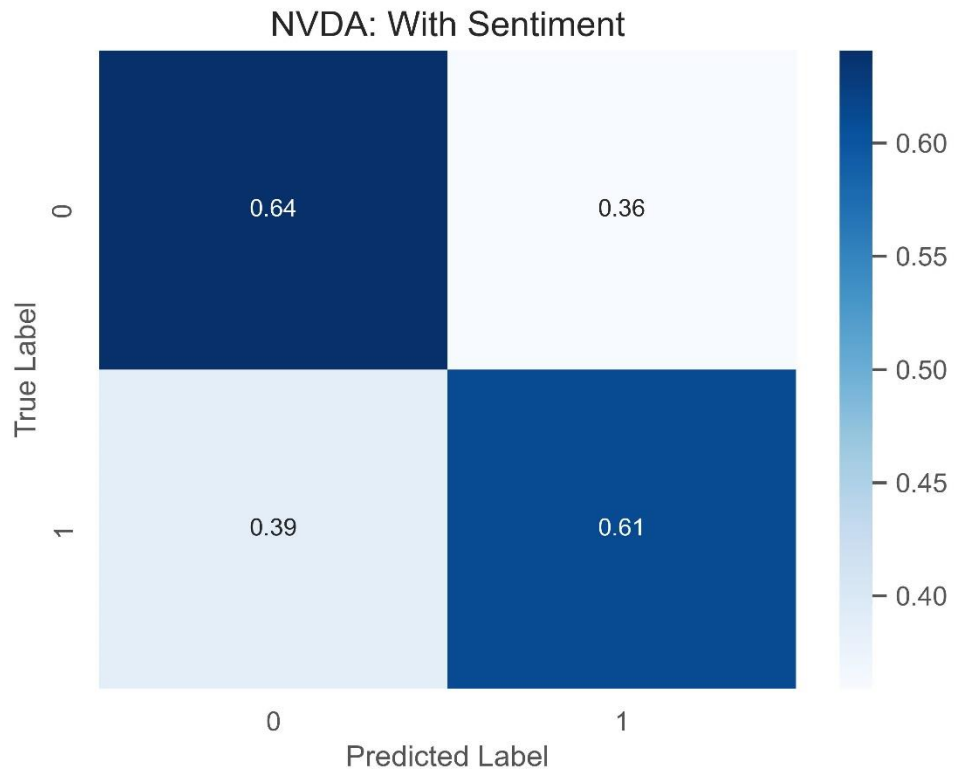


Figure 44: Nvidia Sentiment CFM

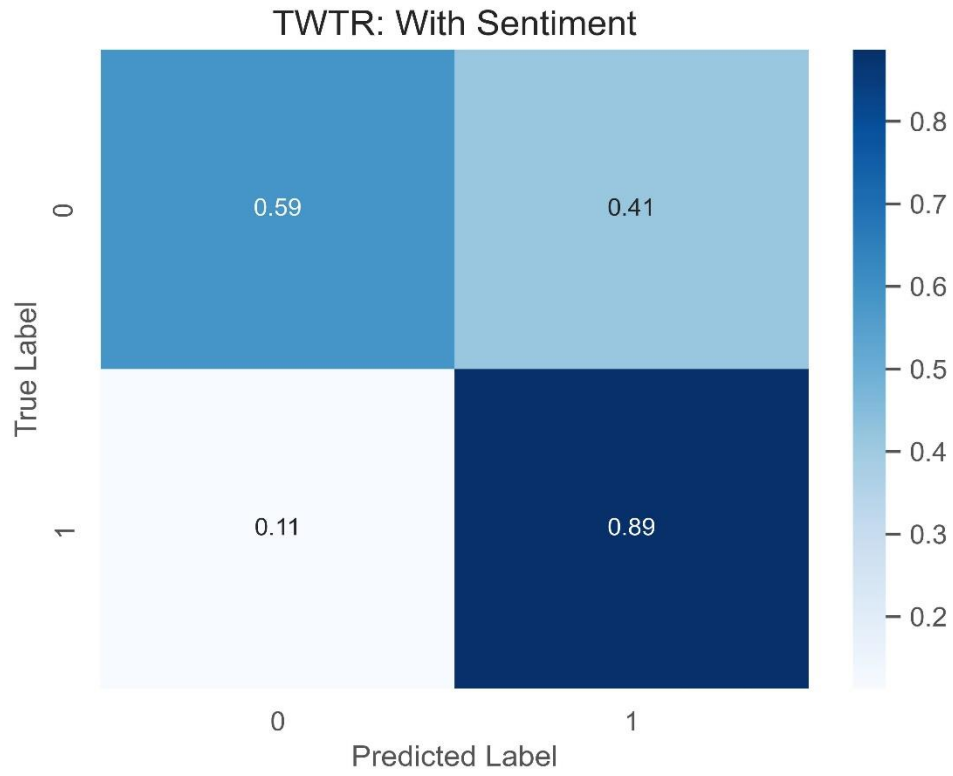


Figure 45: Twitter Sentiment CFM

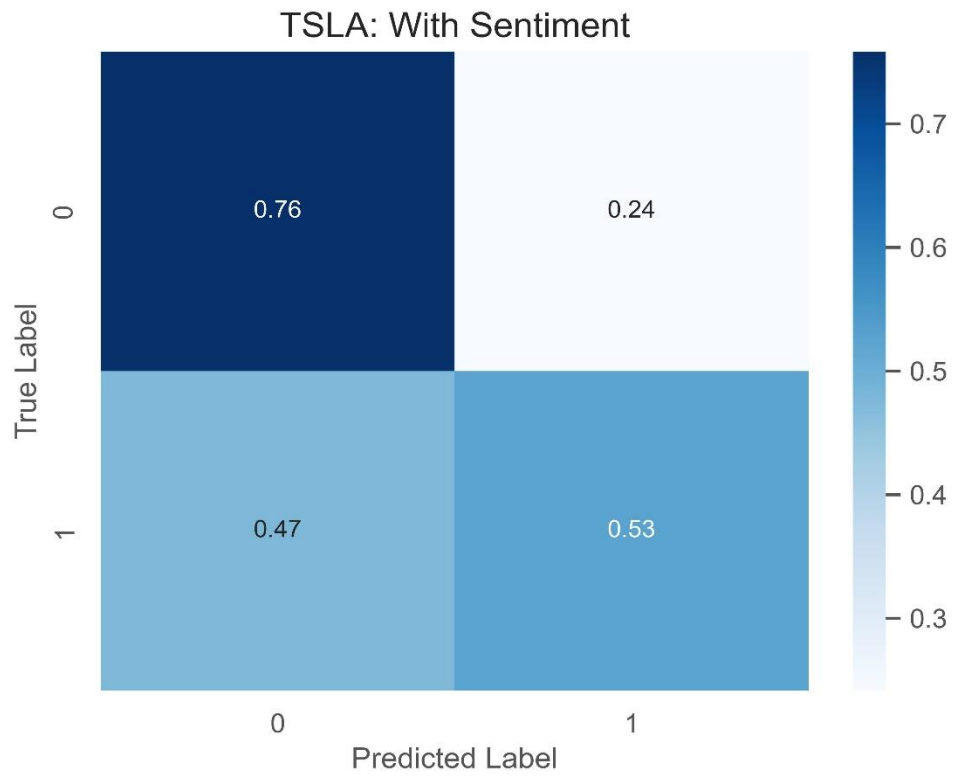


Figure 46: Tesla Sentiment CFM

4.2 Discussion

Table 9: SVM No Sentiment vs. Sentiment Results

Percent Difference (Sentiment vs. No Sentiment)			
Company (ticker)	Accuracy	Precision	Recall
Apple (AAPL)	+0%	+0%	+0%
Advanced Micro Devices (AMD)	+3%	+2%	+5%
Amazon (AMZN)	+0%	+2%	+1%
Alphabet (GOOGL)	+5%	+5%	+3%
Google (GOOG)	+0%	-1%	+0%
Intel (INTC)	+4%	+2%	+4%
Microsoft (MSFT)	+4%	+2%	+5%
Micron Technology (M.U.)	+4%	+1%	+9%
Netflix (NFLX)	+3%	+2%	+0%
Nokia (NOK)	+3%	+3%	+3%
Nvidia (NVDA)	+0%	-1%	+7%
Oracle (ORCL)	+0%	-1%	+0%
Tesla (TSLA)	+4%	+1%	+6%
Twitter (TWTR)	+4%	+3%	+4%
AVERAGE	+2.43%	+1.43%	+3.36%

All metrics improved with the addition of sentiment; average accuracy, precision, and recall increased by 2.43%, 1.43%, and 3.36% respectively (Table 9). Comparatively, recall increased 50% more than the next highest metric—accuracy. As a result, the model was able to

identify good investment opportunities more often when sentiment was included. High BERT recall (0.97) for the ‘negative’ sentiment label implies that the model lets few ‘negative’ labels go unnoticed at the cost of misclassifying good investment opportunities based on sentiment. It should logically follow that good investment recall would decrease for a dataset that includes wrongly classified ‘negative’ sentiment, yet the opposite was observed.

An unequal increase in precision may mean that positively labeled sentiment simply had a greater effect on the model’s decision space than negatively labeled sentiment. This may be further supported by the fact that negative sentiment should have influenced the model’s ability to reduce bad investment misclassification. Had negative sentiment held equal weight, precision may have similarly increased.

However, the most likely explanation is that the techniques used to query the textual information need more refinement. Queries returned all responses containing the identified keywords, yet not all responses may be relevant to the task. The small collection of Google-related queries in Table 10 shows that, although focused articles and tweets are returned, they also include information that can dilute the overall sentiment.

Table 10: Query Issues

Entry No.	Example Query Responses (Google)	Predicted Sentiment
1	"Google saved \$3.7b in taxes through Irish, Dutch tax structure."	Positive
2	"Hedge funds are very aggressively buying Netflix and dumping Google."	Positive
3	" Priceline and Expedia shelled out \$4 billion on google advertising in 2016."	Neutral

Entry 1, an example of a relevant result, was correctly labeled for positive sentiment. Entries 2 and 3, however, show queries that introduced error into the model. For entry 2, the sentiment of the sentence was relative to the noun; for Netflix, the overall sentiment was indeed positive, but for Google (i.e., the topic of the query), the sentiment was negative. Yet, BERT did not identify that as such. Entry 3 is an example where sentiment was labeled as neutral, yet it could be inferred that it was, in fact, positive-- as \$4 billion in revenue is typically a strong indicator of success.

CHAPTER 5. CONCLUSION

In conclusion, the NLP model performed exceptionally well at predicting sentiment for text heavily influenced by a financial vocabulary. Not only was classification accuracy, precision, and recall high, but they were also balanced—indicating the model performed well across all labels. However, as shown by this research, a considerable amount of feature engineering would have improved the results as raw sentiment did not exhibit any significant predictive qualities. Nevertheless, including sentiment slightly improved the SVM's ability to predict a security's trend polarity.

With and without the addition of sentiment, the SVM could function at a markedly high level using methodologies appropriate for industrial use. However, it is likely that the model is susceptible to concept drift (e.g., degrading statistical edges, improved lookback periods) over time and would highly benefit from continual maintenance and evaluation. Moreover, the model could be improved using various techniques, yet it indicates an abstract form of success in its current state.

In the future, the querying methods should be improved. Although the quantity of returned results will be reduced, the quality of the results would benefit from techniques that focus on retrieving via more specific queries. NER could detect and ensure the main subject of a particular payload matches the security queried.

The model could be improved using other methods as well. For example, the model currently produces a dictionary of label probabilities; the label with the highest probability is the output. However, a threshold may be identified to collapse the probabilities to one other than the highest probability if that probability is not within a certain level of certainty. This would allow users to control the risk and may artificially raise the classification accuracy.

Although a significant amount of feature engineering was used to produce the feature-rich dataset used by the SVM, other techniques could have been used to produce more helpful features. In the future, feature crossing and bucketing should be used to explore the usefulness that the compound features could provide.

REFERENCES

- [1] Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), 383–417. <https://doi.org/10.2307/2325486>.
- [2] Koch, C. G. (2020, September). The Warren Buffett Project: A Qualitative Study on Warren Buffett. Retrieved May 5, 2022, from <https://digitalcommons.usf.edu/cgi/viewcontent.cgi?article=9654&context=etd>
- [3] Chen, MY., Sangaiah, A.K., Chen, TH. et al. Deep Learning for Financial Engineering. *Comput Econ* 59, 1277–1281 (2022). <https://doi.org/10.1007/s10614-022-10260-8>
- [4] Cornell, Bradford. (2020). Medallion Fund: The Ultimate Counterexample? *The Journal of Portfolio Management*. 46. [jpm.2020.1.128](https://doi.org/10.3905/jpm.2020.1.128). 10.3905/jpm.2020.1.128.
- [5] Fama, E. F. (1965). Random Walks in Stock Market Prices. *Financial Analysts Journal*, 21(5), 55–59. Retrieved May 9, 2022, from <http://www.jstor.org/stable/4469865>
- [6] Samuelson, Paul A. 1965. "Rational Theory of Warrant Pricing". *Industrial Management Review* 6 (2): 13-39. Retrieved May 9, 2022, from https://link.springer.com/chapter/10.1007/978-3-319-22237-0_11#:~:text=It%20assumes%2C%20explicitly%20or%20implicitly,riskiness%20from%20the%20common%20stock.
- [7] Samuelson, Pual A. 2009. "An Enjoyable Life Puzzling Over Modern Finance Theory". *Annu. Rev. Fin. Econ.* 2009.1:19-35. Retrieved May 9, 2022, from arjournals.annualreviews.org.
- [8] Rosenberg, B., Reid, K. & Lanstein, R. (1985). *Persuasive Evidence of Market Inefficiency*. Princeton: Princeton University Press. Retrieved May 9, 2022, from <https://doi.org/10.1515/9781400829408-007>

- [9] Fama, E. F., & French, K. R. (1988). Dividend yields and expected stock returns. *Journal of Financial Economics*. Retrieved May 9, 2022, from <https://www.sciencedirect.com/science/article/abs/pii/0304405X88900207>
- [10] Poterba, James, and Lawrence Summers (1988), “Mean Reversion in Stock Returns: Evidence and Implications,” *Journal of Financial Economics*, 22, 27-60
- [11] Werner F. M. De Bondt, & Thaler, R. (1985). Does the Stock Market Overreact? *The Journal of Finance*, 40(3), 793–805. <https://doi.org/10.2307/2327804>
- [12] Kahneman, D., & Tversky, A. (1982). The Psychology of Preferences. *Scientific American*, 246(1), 160–173. <http://www.jstor.org/stable/24966506>
- [13] Milosevic, N. (2018, November 22). Equity forecast: Predicting long term stock price movement using machine learning. *arXiv.org*. Retrieved May 9, 2022, from <https://arxiv.org/abs/1603.00751>
- [14] Lake, R. (2021, October 11). What is the average stock market return? *SoFi*. Retrieved May 24, 2022, from <https://www.sofi.com/learn/content/average-stock-market-return/>
- [15] Chen, Y., & Hao, Y. (2017, March 1). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*. <https://www.sciencedirect.com/science/article/pii/S0957417417301367>
- [16] Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., & Kim, H.-C. (2021, November 8). Stock market prediction using Machine Learning Techniques: A decade survey on methodologies, recent developments, and Future Directions. *MDPI*. Retrieved February 7, 2022, from <https://www.mdpi.com/2079-9292/10/21/2717>
- [17] Bharathi.Sv, Shri & Geetha, Angelina. (2017). Sentiment Analysis for Effective Stock Market Prediction. *International Journal of Intelligent Engineering and Systems*. 10. 146-154. 10.22266/ijies2017.0630.16. Retrieved February 25, 2022, from <http://www.inass.sakura.ne.jp/inass/2017/2017063016.pdf>

- [18] Karppi, T., & Crawford, K. (2016). Social Media, Financial Algorithms, and the Hack Crash. *Theory, Culture & Society*, 33(1), 73–92. Retrieved February 13, 2022, from <https://doi.org/10.1177/0263276415583139>
- [19] Jena SK, Tiwari AK, Dash A, Aikins Abakah EJ. Volatility Spillover Dynamics between Large-, Mid-, and Small-Cap Stocks in the Time-Frequency Domain: Implications for Portfolio Management. *Journal of Risk and Financial Management*. 2021; 14(11):531. Retrieved October 30, 2022, <https://doi.org/10.3390/jrfm14110531>
- [20] Mass Language Model. (2020). GeeksForGeeks. Retrieved from <https://www.geeksforgeeks.org/understanding-bert-nlp/>.
- [21] Anna Rogers, Olga Kovaleva, Anna Rumshisky; A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics* 2020; 8 842–866. Retrieved October 27, 2022, doi: https://doi.org/10.1162/tacl_a_00349
- [22] Huang, S., Cai, N., Pacheco, P. P., Narrantes, S., Wang, Y., & Xu, W. (2018). Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. *Cancer genomics & proteomics*, 15(1), 41–51. <https://doi.org/10.21873/cgp.20063>
- [23] Malo, Pekka & Sinha, Ankur & Takala, Pyry & Korhonen, Pekka & Wallenius, Jyrki. (2014). Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts. *Journal of the American Society for Information Science and Technology*. 10.1002/asi.23062.
- [24] Pandas-TA. Documentation - Technical Analysis Library in Python 0.1.4 documentation. (n.d.). Retrieved December 24, 2022, from <https://technical-analysis-library-in-python.readthedocs.io/en/latest/ta.html>
- [25] Sklearn.feature_selection.RFE. scikit. (n.d.). Retrieved December 24, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

VITA

IAN L. GRISHAM

Education: M.S. Computer Science, East Tennessee State University,
Johnson City, Tennessee, 2023

B.S Computer Science, East Tennessee State University,
Johnson City, Tennessee, 2021

Professional Experience: Graduate Research Assistant, East Tennessee State University,
Johnson City, January 2022 – May 2023