Follow the Words: Forecasting Stock Price Movement and the Predictive Power of Earnings Calls

Taylor C. Beckett

Advisor: Dr. Xiaoyan Li

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Arts

Department of Computer Science

Princeton University

April 2021

I hereby declare that I am the sole author of this thesis. I pledge my honor that this paper represents my own work in accordance with University regulations.

_____

Taylor C. Beckett

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

_____

Taylor C. Beckett

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

Taylor C. Beckett

Abstract

This thesis explores the expanding intersection of finance, Natural Language Processing, and linguistic analysis. This paper examines if a linguistic analysis of a company's earnings call holds predictive power in forecasting the movement of the company's stock price on the day after the earnings conference call. Our question is answered through extracting linguistic features from the transcripts of earnings conference calls and building machine learning models to predict stock price movement. First, an analysis of 13 different machine learning classifier's performances on predicting stock price movement on days both with and without earnings calls is provided. Then, an analysis is conducted on samples the day after an earnings call. A discussion of the effectiveness of several different linguistic analysis techniques in a financial context concludes the research. Ultimately, it is found that while predicting stock price movement is a difficult task due to the amount of noise in financial data, with observed accuracy improvements over baseline models as evidence, there is predictive power in linguistic analyses of earnings conference calls, and linguistic features can be used to supplement financial data in predicting stock price movement.

Acknowledgements

First and foremost, thank you to my advisor, Dr. Xiaoyan Li. Thank you for all the energy and time you put into working with me, always keeping me on track each and every week, and guiding me through this entire process. I would not have accomplished as much as I have in this thesis without you. Additionally, thank you for helping me get through COS 217 sophomore year.

Thank you to Professor Pukthuanthong for kindly sharing with me your corpus of earnings calls. I would not have discovered nearly as much without them. Your willingness to share embodies the spirit of openness and collaboration necessary in research.

To Grace - thank you for all the support throughout this entire year and especially these past two-and-a-half weeks. I could not have crossed the finish line without you.

Thank you to all my COS partners throughout my time here - Claire, Natalie, Andrew, Arman, Henry, Ayushi, Dale, and Matteo. Thank you for helping me get through every COS class and for keeping my passion for Computer Science alive.

Thank you to Princeton Baseball for allowing me to live my dream of playing college baseball. Thank you to my 42 teammates and friends for life. My favorite times at Princeton have been with all of you. To my class, thank you for spending the past four years with me. Always remember to "push if you want to go to Omaha!".

To Sy, Jake, and Keith - thank you for being my best friends from the moment we stepped on campus. Princeton would not have been the same without you, and I am thankful for your friendships.

Thank you to Charlotte and Nicole for growing up with me and for being the best sisters a brother could ask for.

Most importantly, thank you to my parents. Mom and Dad, thank you for always being in the stands. Thank you for always pushing me to be my best and for always being there when I came up short. Thank you for giving me every chance to succeed. Thank you for the opportunity to attend Andover and Princeton. I am eternally grateful.

Finally, thank you to Princeton. Thank you for giving me the opportunity to be here. It has been an honor and an experience I will never forget.

To Mom and Dad

Contents

List of Figures

Chapter 1

Introduction

The goal of this thesis is, using a combination of financial data and linguistic features extracted from company's earnings call, predict the movement of a company's stock price. Our second and main objective is, through the first goal, to observe if a linguistic analysis of the transcripts of a company's earnings call has predictive power over the movement of the company's stock price the day after an earnings conference call. This thesis will report the creation of two original data sets, the approach and implementation of machine learning models, a discussion on the results, and finally conclusions to be drawn and a discussion of future work in this area.

My motivation for this thesis comes from a combination of both academic and non-academic interests. In the Spring of 2020, I conducted my Junior Independent Work in a seminar called "Deep Learning for Audio". In the seminar, I developed a Deep Learning neural network that classified songs into genres based on their auditory features. Through my work, I was exposed to Deep Learning and making predictions on data for the first time. The next semester in the Fall of 2020 in my Computer Vision class, I further learned how Machine Learning algorithms can be used in classification problems. I immediately was fascinated about the power of utilizing algorithms to make predictions, and I knew it was an area I wanted to explore in my thesis.

Outside of academic work, I have always had an interest in the stock market and have been fortunate to do some academic research in connection to my studies. This thesis offered a unique opportunity to combine two of my interests. In researching Machine Learning applications for the stock market, I found the intersection of earnings calls and stock price prediction particularly interesting as it is an area that is decidedly less studied than other

areas of finance, and it adds elements of non-financial data into the financial world.

With roughly 6,000 currently publicly traded companies on the stock exchanges of the United States, each with four earnings calls per year, there are about 24,000 earnings conference calls that occur every year [1]. As Price, Doran, Peterson, and Bliss state, earnings conference calls are one of the most important and informative ways that company executives rely company information to the public [2]. As such, earnings conference calls and how they are perceived by the public have a great influence on how the stock price of the company moves in the days following. This is a well-known fact as a significant portion of the financial services industry is dedicated, in part, to earnings calls. According to the U.S. Bureau of Labor Statistics, there are over 300,000 financial analysts in the United States, and it is a sector that is projected to grow at a rate of 5% over the next ten years, faster than the average projected job growth [3]. One of the responsibilities of financial analysts is to listen to the earnings conference calls of the companies they cover. Following the call, analysts write summary reports of the call and often make buy or sell recommendations to their clients and to the public. While this thesis does not do the job of the financial analysts as it relates to earnings calls, it will explore if there is predictive information that analysts can use to supplement their analyses. Analyzing these calls in a non-financial lens and connecting them to stock performance introduces an opportunity to understand the relationship between companies, investors, and the public in a new way.

Chapter 2

Problem Background & Related Work

2.1 Problem Background

In the United States, publicly traded companies are required by the Security and Exchange Commission (SEC) to file their earnings at the end of every quarter of the fiscal year [1]. Each publicly traded company, therefore, holds an earnings call four times a year at roughly three-month intervals. This mandate was outlined in the Regulation Fair Disclosure enacted by the SEC in August of 2000 [1]. Along with the earnings of the company, the company also reports other financial metrics such as total income, profit, and revenue. One of the most highly scrutinized metrics reported by the company is earnings per share, which is the amount of money the company profited per the number of shares outstanding in the company. Financial analysts will publish expectations of these metrics leading up to an earnings call and whether the company does or does not meet expectations is a common way to judge the success of a company. Accompanying the reporting for financial metrics, a company will hold a "earnings call". An earnings call "is a conference between the management of a public company, analysts, investors, and the media to discuss the company's financial results during a given reporting period" [4].

An earnings call is typically structured as follows. There is a section of prepared remarks from the executives of the firm, usually the CEO or COO in addition to the head of investor relations at the company. Following the prepared statements, during which the executives reveal their earnings, summarize challenges the company faced, and provide insight on what is coming next, there is (most often) a section where financial analysts, who

most often are members of the world's largest banks, have the opportunity to ask questions of the executives [4]. During this time, analysts attempt to glean as much information about the company as possible. Executives must be truthful in their answers to these questions although they try to paint their company in the best possible light. Earnings reports are subject to much scrutiny in the financial world with stock prices having the potential to crash or quickly rise based on the financial metrics reported and whether they meet Wall Street expectations. While this is a well-known fact, this thesis strives to look beyond the financial metrics and into the earnings conference call itself; not just what is reported, but *how* it is reported. The word choices of executives on the call, the overall tone of the call, and the linguist qualities all have the possibility to reveal more information than just the reported numbers.

Natural Language Processing (NLP) is one of the most rapidly growing fields of computer science. NLP is "an aspect of Artificial Intelligence that helps computers understand, interpret, and utilize" human languages and text [5]. Sentiment analysis and complexity, the two NLP processes explored in this thesis, are sub-branches of the larger category of NLP. While both can be traced back to the 1960s, computer-based sentiment analysis did not become a deeply studied topic until the turn of the century. In a 1993 paper Stephen Matsuba wrote that "computed-assisted literary research...[was] in a state of crisis" [6]. Then, in the late 1990s and early 2000s, the Internet Revolution occurred and suddenly there was mass availability of subjective texts already in digital form on the Web and sentiment analysis gained new popularity as a area of research [7]. Mäntyla, Graziotin, and Kuutila from the University of Stuttgart, in a comprehensive review of sentiment analysis research from 2016, found 99% of the papers they reviewed were published after 2004 [7]. As such, sentiment analysis and complexity are two relative new topics to the field of computer science.

Large financial institutions can be extremely reluctant to utilize new technologies like Natural Language Processing or machine learning. While tech companies like Google and Apple have been researching and deploying Natural Language tools since the internet

outbreak, the financial world has lagged behind [8]. However, this is changing as financial companies are realizing that Natural Language Processing is able to parse text more accurately and more importantly, much quicker than a human. As Professor Shulman of MIT's Sloan Business School asserts, financial data is not limited to numbers and "data that can help make timely decisions comes in text" [8]. This thesis attempts to join in the efforts of using NLP in a financial context and extract information that could be used to help make timely investment decisions after earnings calls.

2.2 Related Work

Relative to other areas of financial research, there have not been a great number of studies conducted on the effect of earnings call linguistic features on stock price; most research on earnings reports and stock price has been focused on the reported financial data. However, linguistics and finance seems to be an area of increased attention in the academic and professional worlds as most of the related studies reviewed were produced in the last five years. This may be due to the rise of better computational power and the increased interest in Natural Language Processing. Furthermore, it has now been over 20 years since Regulation Fair Disclosure was created to require quarterly reporting. Before 2000, earnings calls were far less common [4]. It has been a sufficient amount of time for enough data points and observations to become available, which may explain the increased interest in this area of research. In the studies we found on linguistic features and stock price, nearly all found evidence of predictive power in the earnings call transcripts.

In *Earnings Conference Calls and Stock Returns: The Incremental Informativeness of Textual Tone* (2011) Price, Doran, Peterson, and Bliss from Lehigh University and Florida State University found that earnings calls have significant prediction power in forecasting abnormal returns and trading volume [2]. Their work focused on the analysis of Real Estate Investment Trusts and American Depository Receipts. Through regression analysis of the

tone of earnings calls, they were able to show that company's with high TONE scores in earnings calls tended to observe much higher than average returns on their security over the next 60 days. They defined TONE as

$$TONE_j = First\ Principal\ Component \left\{ \frac{Positive_j}{Negative_j}, \frac{Active_j}{Passive_j}, \frac{Strong_j}{Weak_j}, \frac{Overstated_j}{Understated_j} \right\}$$

The variables in each ratio in the First Principal Component refers to linguistic dictionary word counts that will be explained in a discussion of our own data as we take a somewhat related approach.

Using a combination of linguistic analysis features and financial data, Laput, Paruthi, and Singhal from the University of Michigan were able to achieve a 1% increase in accuracy from their baseline results of random guessing in their paper *G-Money: Earnings Calls Predict Stock Performance* (2020) [9]. Similar to Price, Doran, Peterson, and Bliss, they extracted linguistic features from word counts such as in their definition of sentiment:

$$Sentiment_i = \frac{|positive\ words|}{|negative\ words|}$$

In addition to these word count features, Laput, Paruthi, and Singhal also utilized transfer learning by using a neural network for overall sentiment analysis that was trained on a separate data set of movie reviews. This was a considered approach for us, but as the Michigan research found, this feature was insignificant. For their classification models, Laput, Paruthi, and Singhal used individual classifiers such as Logistic Regressions and Support Vector Machines in addition to ensemble techniques such as boosting and aggregation. In their discussion of their results, they find ensemble models to be more effective that single classifier models. While this thesis looks at all price movement in the days after an earnings call, Laput, Paruthi, and Singhal only analyzed predicting cumulative abnormal returns (CAR), a slightly more selective objective. In addition, in their discussion of their results, they commented that while a 1% increase of accuracy over random guessing does not seem

very successful, when considering the difficulty of the problem, it is actually a meaningful increase [9]. In relation to this thesis, it serves as evidence of the existence of predictive power in earnings calls.

A Master's Thesis from Solberg and Karlsen from the Norwegian School of Economics titled *The predictive power of earnings conference calls : predicting stock price movement with earnings call transcripts* (2018) finds similar small improvement in using earnings calls from companies listed on the NYSE and NASDAQ in the years 2014-2017 to predict stock price movement the next day [10]. In extracting linguistic features, they utilized the Loughran-McDonald Financial Dictionary, one of the dictionaries used in this thesis, to extract a positive or negative sentiment from the earnings call. The manner they conducted their linguistic analysis in was similar to the two works above. To predict the stock movement on the next day, the team employed four different algorithms: Naïve Bayes, Logistic Regression with Lasso Regularization, Stochastic Gradient Boosting, and Support Vector Machines. Stochastic Gradient Boosting, an ensemble model, and Logistic Regression both significantly outperformed their benchmarks. Similar to the team from the University of Michigan, Solberg and Karlsen reached the conclusion that earnings calls contain predictive power for next day's stock price direction, but some classifiers are more suited to the task than others.

In *Towards Earnings Call and Stock Price Movement* (2020), Ma, Bang, Wang, and Liu from Standard & Poor's found that while stock movement is a very difficult task, their experimentation confirms that "earnings call transcripts have a certain predictive power for future stock price movements" [11]. To extract linguistic features from their set of 17,025 earnings calls from 2009 to 2019, the team created sentence embedding vectors and sentence attention vectors from only the Question and Answer section of the calls. This was a different, more complex type of linguistic analysis than the word count extraction we observed in the first few studies. These features were then used as inputs to a 6-layer neural network that learned the features and finally a 4-layer neutral network to produce the final price movement

prediction. A neural network approach was considered for this thesis, but due to time and resource constraints, this route was not followed. However, given the amount of noise in financial data, a neural network could be a promising approach. In their conclusions, Ma, Bang, Wang, and Liu found evidence of predictive power, but reiterated the difficulty of stock price movement prediction.

Ulrich, Pratt, and Thun-Hohenstein from Stanford University used a Bernoulli Naïve Bayes Classifier to achieve 70% - 75% accuracy in predicting stock price movement after earnings calls [12]. This accuracy, found in *Machine Learning Analysis of Company Earnings Releases* (2018), is an outstanding result that has not been replicated in any other examined paper. Most of the related papers read have found predictive power in earnings calls, but this paper achieved a significantly more impressive accuracy. Ulrich, Pratt, and Thun-Hohenstein's results are certainly an outlier in this area of research and this suggests that their methodology may have been too simplistic. The team only used word frequencies for their linguistic features with a single Bernoulli Naïve Bayes model. Regardless of doubts about their results, the impressive overall accuracy warrants exploration in this thesis.

Finally, Qin and Yang from the Renmin University of China's School of Information and the Hong Kong University of Science and Technology, respectively, used linguistic analyses of textual earnings call transcripts and audio recordings to predict financial risk in their paper *What You Say and How You Say It Matters: Predicting Stock Volatility Using Verbal and Vocal Cues* (2019) [13]. They used a Multimodal Deep Regression Model and a BiLSTM along with simpler regressions to predict short and long term volatility. They found that the more complicated models were more effective at prediction, but still only produced a small gain over baseline benchmarks. Furthermore, they also found that short-term volatility was extremely hard to predict and long-term predictions were more reliable. While we do not examine audio recordings in this thesis, our research is an exploration of short-term predictions, which they showed to be difficult.

Overall, all related research reviewed found there is significant, albeit limited, pre-

dictive power in company's earnings calls in both predicting long and short term movement. While these studies served as inspiration for the direction of this thesis, our paper explores many different areas not examined in the works above. This thesis creates two original data sets as well as uses different machine learning techniques than the reviewed papers in both the single classifier and ensemble classifier category. Additionally, in regards to linguistic analysis, this paper examines several linguistic features not explored in the reviewed related works. Examining this problem with two original datasets and a variety of linguistic analysis techniques will deliver new insights not seen in the reviewed papers.

Chapter 3

Approach

3.1 Earnings Call Collection & Financial Data

The biggest challenge in this project was collecting earnings call transcripts. Earnings calls must be, in accordance with Regulation Fair Disclosure, public information. Transcripts as well as audio recordings of an earnings call are readily available on company's investor relations web pages or on many financial reporting websites. However, it proved very challenging to find a large database of earnings call transcripts as there is not an official database of calls or even a publicly available database for bulk downloads. Any promising leads in gaining access were unsuccessful as they usually sat behind significant paywalls or in institutional databases that require special permission to access. It is curious as to why there is no database available for single download considering company's are mandated by law to offer reports to the public. Related work studies all seemed to be able to acquire significant corpora of earnings calls although they did not offer where they were able to find these datasets in their papers. Our intuition is that earnings calls, as previously stated, are effective and reliable catalysts for stock price movement. When it comes to market insights and edges, the financial world in notoriously secretive. Researchers and companies may be trying to protect their own insights and trading strategies by not providing access to their compiled earnings calls.

Through two different approaches that will be described below, we created two different data sets which are denoted as Dataset Created and Dataset Pro.

3.1.1 Dataset Created

Dataset Created will refer to the dataset we manually created using calls taken from company's investor relations websites and SeekingAlpha, a financial reporting website. We were unable to web scrape calls as many websites have now implemented protection against web scrapping. In total, we complied 804 earnings calls from 21 different companies across a variety of industries. These earnings calls were collected from a time period of 10 years, from the third quarter of 2010 through the third quarter of 2020. Due to a time consuming compilation process, this corpus of earnings calls is extremely small.

To obtain the financial information necessary to predict stock price movement, 10 years of stock price data for each company from the same period was downloaded from the NASDAQ Historical Stock Data database [14]. This stock data was significantly easier to find than the earnings calls. The data downloaded was initially very bare, consisting of the date, the opening and closing price of the stock, and the total trading volume. To create the first iteration of the dataset, the earnings calls and stock prices were joined on their dates in order to create a single dataset of stock prices and earnings calls. Dataset Created was completed before gaining access to Professor Pukthuanthong's earnings call corpus.

3.1.2 Dataset Pro

As stated above, the Dataset Created is very small and leaves a lot to be desired in terms of scale, especially when the goal is to utilize machine learning algorithms to make predictions. In an effort to increase the amount of data available to be analyzed and to gain more accurate insights, we sought out larger sources of data even after compiling Dataset Created. In our efforts, we connected with Professor Kuntara Pukthuanthong, the Robert J. Trulaske Professor of Finance at the University of Missouri College of Business who possesses a large corpus of earnings call transcripts. Professor Pukthuanthong was kind enough to share with us her corpus of earnings calls. Her kindness and willingness to share in the interest of

academic research allowed us to create Dataset Pro, a second, much larger dataset.

The provided corpus consists of 96,779 earnings calls from 6,651 unique companies. All of the earnings calls were scrapped off of SeekingAlpha when it did not have web scrapping protection. The scrapped calls belonged to the 13 year time period of 2004-2016. We were incredibly fortunate to have access to such a high number of transcripts. Indeed, 96,779 calls were more than we had observed in any related paper.

Since North American firms are the only ones bound to Regulation Fair Disclosure, this thesis only looks at firms based in North America. Non-North American firms were able to be weeded out through our download of stock price data.

As was the case with Dataset Created, we need to acquire stock price data for each firm with calls in the provided corpus. As there are over 6,000 firms represented, manually downloading the stock prices one by one as we did in making Dataset Created was not a feasible option. Instead, we utilized Wharton Research Data Services, a financial database compiler run by the Wharton School of Business at the University of Pennsylvania, through Princeton Library's subscription. Wharton Research Data Services will also be referred to as WRDS. After extracting the stock tickers from the names of the earnings call transcript files, we were able to input the list of tickers into a WRDS database and bulk download stock price data for every company. The stock price database belongs to the Center for Research of Security Prices (CRSP) which is an affiliate of the University Chicago Booth School of Business [15]. In downloading stock price data, by requiring that a company's security must have a Committee on Uniform Security Identification Procedures Number (CUSIP), firms that are based outside of the United States and Canada were excluded from the stock price download [16]. We also excluded any earnings calls whose company was not in the stock price download which meant the call was from a non-North American company. This shrunk the number of calls from 96,779 to 67,702, reduced the number of firms from 6,651 to 4,529, and reduced the period of observation to 2006-2016. While this is a significant reduction, 67,702 eligible earnings calls were still greater than any other reviewed paper. Once the

stock price was downloaded, it was merged with the earnings call corpus. Section 3.2 will describe how usable features were extracted from the raw text of earnings calls.

3.1.3 Financial Features

The objective of this thesis is to examine any predictive power of earnings calls using a *combination* of linguistic features and financial data reported in earnings reports. The reason for including reported financial features as well is that these features are well-known to having an affect on the stock price after an earnings call. For example, if a company, in their quarterly report, announces the company has taken on a lot of debt or increased their revenue, the market will usually react accordingly. In addition, as stated in the Problem Background, one of the key features in company's earnings calls and quarterly reports is analyst expectations. Whether the company beats or falls short of analysts estimates has a great affect on the reaction to the earnings call and subsequently the stock price. For this reason, the stock price data downloaded from WRDS and the NASDAQ stock price database were augmented with further information about stock price history, reported financial metrics, and analyst estimates.

The simplest augmentation of financial data was the stock price history. Operating off the closing price of the stock, we created a new feature called 'YesterdayMovement' which is the signed magnitude of the movement of the price the day before. We also formed a binary variable 'YesterdayMove' which was set to 1 if the price increased the day before and 0 if the price fell. We also generated variables that held the 5- and 10-day moving averages of the price (named in that way). Our dependent variable our work tries to predict is called 'Move' which is a binary variable signifying if the stock priced increased (1) or decreased (0) on that day.

Next, we were able to access financial metrics reported at the end of each quarter for each company through the CRSP/Compustat merged database. Compustat is a database of reported financial metrics that CRSP has linked to their database using a linking key which

allowed us to seamlessly join price data from CRSP to reported metric data from Compustat [15]. From CRSP/Compustat, we downloaded several different reported financial metrics ranging from revenue for the quarter to debt for the quarter. For guidance on which metrics are seen as most important in the quarterly earnings reports, we were advised by Bobray Bordelon, the Princeton Librarian for Financial and Economic Data. Mr. Bordelon was also helped us gain access to WRDS.

Finally, we downloaded analyst expectations of the reported financial metrics from another database accessed through WRDS: the Institutional Brokers' Estimate System (IBES) [17]. Through this database we were able to download, for every quarterly report for each company, the mean analyst expectation for a range of financial metrics. This mean estimation serves as a good indicator of what financial professionals were expecting from a company's earnings report. In addition to the estimates, we were able to download the *actual* reported value of each financial metric. In total we downloaded expectations and actuals for Earnings Per Share (EPS), Operating Profit (OPR), Net Income (NET), and EBITA (EBT). From the expectations and actuals, we created several more variables such as difference between the expectations and actuals and a binary variable on whether the actual reported metric beat analysts expectations.

A summary of all financial features is available in section 3.3

3.2 Linguistic Feature Extraction

While obtaining financial data is necessary, the most critical part of this thesis was turning the raw textual data of the earnings call transcripts in usable features that can be introduced to machine learning algorithms. This section describes how this process worked.

Along with the corpus of earnings calls, Professor Pukthuanthong also shared a CSV of linguistic features that were extracted from the earnings calls as part of her prior research. These features came in the form of Positive and Negative Scores based on two

different financial dictionaries which will be explored in the following sections. While we were fortunate to have some feature extraction completed already for Dataset Pro, we extracted the same features for Dataset Created and conducted additional linguistic analysis for both Datasets. In total, we extracted 10 different linguistic features from both sets of earnings calls. These 10 features can be grouped into three separate groups and will each by analyzed in the sections below.

3.2.1 Sentiment Analysis and Text Cleaning

Sentiment Analysis is a well-defined method in which, based on the frequency of certain word choices, each call can be given a "score". Before calculating the scores of each call, Natural Language Processing (NLP) practices require earnings calls to be 'cleaned' [18]. In written text, there is a great deal of information that delivers no insight into the qualities of the text. This useless information includes punctuation and "stop words". Stop words are defined as a set of very commonly used words in a language. In English, these include words such as 'and', 'is', 'the', or 'a'. These words are simply used in construction of sentences; they are not important to the meaning of the sentence. Removing these punctuation and stop words are the first necessary step in NLP [18].

Next, the call must be tokenized and lemmatized. Tokenize refers to breaking the call, which is read by the computer as a single long string, into smaller, more manipulable chunks such as words. After running a full earnings call through a tokenizer, the string of the earnings call is converted to an array of strings, with each index of the array containing a single word. Lemmatizing refers to the process of reducing inflection in words in a set of text. In English, words have a base 'root' in addition to (sometimes) suffixes and prefixes. In terms of NLP, many suffixes, such as -ing or -ed only function to changing the tense of the word, which is unnecessary information, and does not change what the base word is communicating. For example, after getting processed by a lemmatizer, the word 'buying' would be changed to 'buy'. Lemmatizing and tokenizing are both mandatory steps in preparing text for Natural

Language Processing [18].

After removing all stop words and punctuation and subsequently running each call though a tokenizer and lemmatizer, the sentiment scores were ready to be extracted. In terms of sentiment score, we extracted four different sentiment scores: positive sentiment, negative sentiment, polarity, and subjectivity. Positive and Negative sentiment scores are calculated in percentages of the total word count of the call. The formula for each are as follows with WC denoting 'word count':

$$Positive\ Sentiment = \frac{Positive\ WC}{Total\ WC} * 100$$

$$Negative\ Sentiment = \frac{Negative\ WC}{Total\ WC} * 100$$

$$Subjectivity = \frac{Pos\ WC + Neg\ WC}{Total\ WC}$$

$$Polarity = \frac{Pos\ WC - Neg\ WC}{Pos\ WC + Neg\ WC}$$

To classify words as 'Positive' or 'Negative' we used two different textual analysis "dictionaries". Tokenizing the earnings call into words allowed us to match each word of the call to the dictionaries. Utilizing dictionaries is widely used in sentiment analysis. In the 1960s, Harvard researchers developed a dictionary to analyze word tonality in a sociology setting. Through their analysis, the Harvard researchers formatted the dictionary such that words that are associated with a "positive sentiment" were classified under the dictionary as Positive words. This dictionary is commonly referred to the General Inquirer (GI) Dictionary [19]. In the development of Natural Language Processing as a an area of interest in Computer Science, this dictionary was commonly used to classify words and calculate the tone of a given piece of text. As it is one of the most widely used dictionaries in NLP, the General Inquirer dictionary is one of the two we leveraged.

The other dictionary we utilized comes from Tim Loughran and Bill McDonald, two researchers from the University of Notre Dame. In their 2011 paper *When is a Liability*

not a Liability? Textual Analysis, Dictionaries, and 10-Ks, Loughran and McDonald found that using traditional sentiment dictionaries such as General Inquirer frequently misclassified words as negative when they are not negative in a financial context [20] . As a result of their research, Loughran and McDonald created a "Master" dictionary in the same format as the General Inquirer dictionary, but with taking financial context into account. Formally, this dictionary is referred to as the Loughran-McDonald Financial Sentiment Dictionary (LM). The LM dictionary is one of the most widely cited and utilized dictionaries in financial NLP, and many of the papers we reviewed as related works used the Loughran and McDonald's work. As such, this is the second dictionary we used.

Despite the findings of Loughran and McDonald, we decided to extract features from both the Loughran-McDonald Dictionary and the General Inquirer Dictionary. This allows use to introduce another research question which is whether Loughran-McDonald's findings, that the GI dictionary is not sufficient for textual analysis in financial contexts, translate to earnings calls as well. Loughran and McDonald conducted their analysis on 10-K documents, which are the documents companies submit to officially report their earnings [20]. One of the key differences between 10-Ks and earnings calls is 10-Ks are written documents while earnings calls are spoken words translated into text.

With the four sentiment scores produced using both dictionaries, we extracted a total of eight sentiment scores for each earnings call in Dataset Created and Dataset Pro.

3.2.2 Flesch Reading Ease

To complement the sentiment scores, we extracted two more scores that deal with complexity, another area of NLP. Complexity scores are also know as 'readability tests' designed to evaluate the difficulty of reading and understanding a piece of text. One of the complexity metrics this thesis employs is known as the Flesch Reading Ease Score (FRE) [21]. In the larger NLP subject, this is one of the most popular measures of complexity. The score is calculated as follows:

$$FRE = 206.835 - 1.015\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right)$$

FRE scores are inversely related to ease of reading with higher scores equaling a lower degree of difficulty. Along with the equation, the following table can be used to translate scores into meaning:

Score	Notes
100.0 - 90.0	Very easy to read. Easily understood by an average 11-year-old student
90.0 - 80.0	Easy to read. Conventional English for consumers.
80.0 - 70.0	Fairly easy to read.
70.0 - 60.0	Plain English. Easily understood by 13- to 15-year-old students.
60.0 - 50.0	Fairly difficult to read.
50.0 - 40.0	Difficult to read.
30.0 - 10.0	Very difficult to read. Best understood by university graduates.
10.0 - 0.0	Extremely difficult to read. Best understood by university graduates.

Figure 3.1: Interpreting Flesch Reading Ease

Flesch Reading Ease scores were computed for all calls in both Dataset Created and Dataset Pro.

3.2.3 SMOG Index

The SMOG index grade is another measure of complexity that was extracted by this thesis. The SMOG index was created by G. Harry McLaughlin in 1969 as a more accurate iteration of the Gunning Fog Index, a complexity measure that was seen in some other related papers [22]. The formula for counting the SMOG index grade is as follows. For reference, a polysyllable

is a word of 3 or more syllables.

$$SMOG = 1.043\sqrt{\#\ of\ polysyllables * \frac{30}{\#\ of\ sentences}} + 3.1291$$

The output of the SMOG index is in the form of a "grade level" or the necessary grade level needed to understand the text in one reading.

SMOG	**Grade level**	**SMOG**	**Grade level**
17+	College Grad	12.0 - 12.9	High School Senior
16.0 - 16.9	College Senior	11.0 - 11.9	High School Junior
15.0 - 15.9	College Junior	10.0 - 10.9	High School Soph.
14.0 - 14.9	College Sophomore	9.0 - 9.9	High School Freshman
13.0 - 13.9	College Freshman	8.0 - 8.9	8th Grader

Figure 3.2: Interpreting SMOG Index Grade

SMOG index grade scores were computed for all calls in both Dataset Created and Dataset Pro.

3.3 Features Summary

Figure 3.3 represent all of the main data features extracted and used to predict the movement of the stock price. All observations have features 1 through 5. Only days after an earnings call have features 6 through 29. Features 6 through 12 are only reported on the days of the company's earnings call. Linguistic features are taken from the earnings calls. On the days that are not immediately following an earnings call, features 6 through 29 are replaced with zeros.

Dependent Variable
1. Move (Movement of the price compared to the day before; binary variable)
Financial Data - Stock History (NASDAQ/CRSP)
2. YesterdayClose (Closing price of the stock on the day before)
3. YesterdayMovement (Movement of the price yesterday; continuous variable)
4. 5-Day Moving Average (Average closing price over the last 5 days)
5. 10-Day Moving Average (Average closing price over the last 10 days)
Reported Financial Data - CRSP/Compustat
6. actq (Total Reported Assets)
7. ciq (Reported Comprehensive Income)
8. dd1q (Reported Long Term Debt Due in One Quarter)
9. ltq (Total Reported Liabilities)
10. revtq (Total Reported Revenue)
11. capxy (Reported Capital Expenditures)
Analyst Expectations Financial Data - IBES
12. EPS_DIFF (Difference between expected and actual - EPS; continuous variable)
13. BEAT_EPS (actual beat expectations - EPS; binary variable)
14. EBT_DIFF (Difference between expected and actual - EBITA; continuous variable)
15. BEAT_EBT (actual beat expectations - EBITA; binary variable)
16. NET_DIFF (Difference between expected and actual - Net Income; continuous variable)
17. BEAT_NET (actual beat expectations - Net Income; binary variable)
18. OPR_DIFF (Difference between expected and actual - Operating Profit; contin. variable)
19. BEAT_OPR (actual beat expectations - Operating Profit; binary variable)
Linguistic Features
20. LM_Negative (% of words in the call from the Loughran-McDonald Negative dictionary)
21. LM_Positive (% of words in the call from the Loughran-McDonald Positive dictionary)
22. LM_Subjectivity (Subjectivity using the LM Dictionary)
23. LM_Polarity (Polarity using the LM Dictionary)
24. GI_Negative (% of words in the call from the Harvard IV4 Negative dictionary)
25. GI_Positive (% of words in the call from the Harvard IV4 Positive dictionary)
26. GI_Subjectivity (Subjectivity using the Harvard IV4 Dictionary)
27. GI_Polarity (Polarity using the GI Dictionary)
28. Flesch (Flesch Reading Ease Score)
29. SMOG (SMOG Index Grade Score)
1 Dependent Variable; 28 Independent Variables

Figure 3.3: Data Features Summary

3.4 Models

In this section, we will discuss the machine learning models or classifiers that were evaluated and leveraged during this thesis. The words classifier and model will be used interchangeably. In total, 13 different models were used at some point in the process. We analyzed eight

different 'single' models and five different ensemble models. We use the term single models to refer to the classifiers that are not ensemble models. The eight single models were K-Nearest Neighbors, Decision Tree, Multinomial Naïve Bayes, Gaussian Naïve Bayes, Bernoulli Naïve Bayes, Logistic Regression, Nu-Support Vector Classifier, and Support Vector Classifier. The five ensemble models include Hard Voting Classifier, Soft Voting Classifier, Bagging Classifier, AdaBoost Classifier, and Stacking Classifier. Ensemble modeling "combines several base models" or trains a model on different subsets of the data "in order to produce one optimal predictive model" [23]. How each ensemble model used accomplishes this task is detailed in the sections below.

Four of the single models, Decision Tree, Multinomial Naïve Bayes, Gaussian Naïve Bayes, and Nu-Support Vector Classifier, performed poorly in the first iteration of testing and were eliminated from further iterations in the interest of focusing energy on the most promising models. As such, they will not be elaborated on in this section. Instead, we will discuss the details of the four remaining single models and the five ensemble models.

3.4.1 Logistic Regression (LogReg)

Logistic regression is a linear model that is not natural to machine learning, but first evolved from the world of pure statistics. In a logistic regression, for each possible class, the probability that that class is the correct one is output. Each of these probabilities is modeled using a logarithmic function through a linear combination of each feature in the dataset.

Mathematically, a logistic regression finds the probability that a sample, in our case a trading day, belongs to a certain class through the following logistic function [24]:

$$P(X) = \frac{e^{\beta_0+\beta_1*x_1+...+\beta_n*x_n}}{1+e^{\beta_0+\beta_1*x_1+...+\beta_n*x_n}}$$

Where β_i equals the learned weight on feature x_i. The logistic function will produce a number between 0 and 1. For our logistic regression model, we used l2 regularization. Regularization

helps improve numerical stability as the model is learning the data. Finally, the model uses a threshold to convert a probability to a prediction. If the probability for a class is above the threshold, that is the class that is predicted by the model. The default threshold is 0.5, and this is the threshold we used for our classification problem.

We chose to use logistic regression as one of our models for two reasons. First, it is a staple of classification and a body of machine learning research would be incomplete without logistic regression included. Secondly, several of the related papers, in particular Solberg and Karlsen, logistic regression was shown to be effective [10].

3.4.2 Bernoulli Naïve Bayes (BNB)

Bernoulli Naïve Bayes uses Bayes rule for probability to conduct training and classification. For the classification, Bernoulli Naïve Bayes employees the following rule [24]:

$$P(x_i|y) = P(i|y)x_i + (1 - P(i|y)(1 - x_i)$$

where y is a feature and x_i is a class.

The key idea behind Bernoulli Naïve Bayes is that it assumes that every variable is a binary variable. In order to ensure that we are able to use Bernoulli Naïve Bayes correctly, we took advantage of the 'binarize' aspect of the Bernoulli Naïve Bayes sci-kit learn model that transforms any continuous variables into a binary distribution.

We used a Bernoulli Naïve Bayes model in this thesis because of the impressive results Ulrich, Pratt, and Thun-Hohenstein obtained from using the same model [12]. While the results have not been replicated to nearly the same degree in another paper we found, the fact this model produced such high results warrants its inclusion in this research.

3.4.3 K-Nearest Neighbors Classifier (KNN)

K-Nearest Neighbors is an example of instance-based learning as the classifier does not attempt to actually 'build' a model like the other single classifiers [24]. Instead, it simply stores each sample, or instance, of the training data. The output of a given sample is voted on by the training instances that are closest to the given sample, or its 'nearest neighbors'. Closest is defined as the Euclidean distance from the given sample to a training instance that has been stored in the sample space. The number of neighbors and therefore votes depends on at what value k is set. For this thesis, we set k to the default value of 5. The optimal value of k is highly dependent on the data and there lacks a formal process to find the optimal value.

A key element of K-nearest neighbors is the requirement of the data to be scaled so larger magnitude features and features that have a higher variance do not have out sized effects on the classification issue. To scale the data in this thesis, we used the scikit-learn make_pipeline() function in combination with the scikit-learn StandardScaler() function which automatically scales the training data before passing it to the KNN algorithm [24].

We used K-Nearest Neighbors for this thesis as it is one of the classifiers we were most familiar with prior to beginning research in addition to it being a different 'type' of single model than the others as it is a instance-based learning technique.

3.4.4 Support Vector Classifier (SVC)

Support Vector Classifiers are a subset of Support Vector Machines which are a set of supervised machine learning methods [24]. Unlike logistic regressions, a support vector classifier attempts to separate training samples into groups. Support vector classifiers work by constructing a hyper-plane or a set of hyper-planes in a high dimensional space that will separate the data into groups. What group a sample falls into is the predicted classification of that sample.

Support Vector Classifiers are most effective when the data is linearly separable and the hyper-planes can be created by a linear kernel. However, as we do not have linearly separable data, a Radial Basis Function (rbf) kernel was used instead of a linear kernel. An rbf kernel allows the Support Vector Classifier to create hyper-planes that are not linear.

SVCs create high-dimensional or even infinite spaces in which to create hyper-planes. As such, similar to with KNN, the data must be scaled. The make_pipeline() and StandardScaler() functions were again used to scale the data before training the SVC. The main reason we used Support Vector Classifiers in our research is because of the models demonstrated effectiveness in high-dimensional spaces.

3.4.5 Hard & Soft Voting

Voting is the first of the ensemble models. The idea behind a Voting Classifier is to combine different classifiers and have each predict on their own what the correct label should be. Then, each of the models will place their "vote" for what the model outputs overall. The winner of the vote is determined based on what voting rule is used. In this thesis, we used two voting models each with a different voting rule [24].

The first voting rule is known as 'hard voting' or simple majority voting where the most frequent class predicted is the output of the entire model.

The other voting rule is called 'soft voting' or Weighted Average Probabilities. Under this voting rule, the classifier returns the argument maximum of the probabilities averaged from each classifier. Instead of having each classifier predict the label, in this voting method, each classifier will output the probability that each class is the correct one. Then, the class with the highest weighted probability is output by the soft voting classifier as the final predicted class. For this thesis, each individual classifier was given equal weight although this is a tunable parameter.

In both the soft voting model and the hard voting model, all four of the most prominent single models (K-Nearest Neighbors, Logistic Regression, Support Vector Machine, and

Bernoulli Naïve Bayes) were used as the different estimators.

We used hard and soft voting classifiers in our collection of ensemble models as it is one of the simplest ensemble models and a strong place to begin our exploration of ensemble classifiers.

3.4.6 AdaBoost (Boosting)

The principle of AdaBoost, or Boosting, is to continuously fit a sequence of models that are only slightly better than random guessing (known as weak learners) on repeatedly adjusted forms of the data [24]. The final prediction is based off a weighted vote of the sequence of weak learners. AdaBoost takes advantage of so-called 'boosting iterations' to modify the data on each successive round of classification. Boosting iterations begin by weighting each of the training samples with a weight of $\frac{1}{N}$ where N is the number of samples. As such, the first round of boosting consists of just training the single weak learner on the training data. At each step in the boosting algorithm, the weights are reclassified and the training data is fed back into the classifier. The weights are reclassified such that incorrectly classified observations have their weights increased and correctly classified observations have their weights decreased. The purpose of re-weighting in this way is to give more attention to the observations that are misclassified as the next weak learner encounters the training sample, hopefully reducing the amount of incorrectly classified observations.

For this thesis, 30 Decision Tree Classifiers were used to build our AdaBoost Classifier. Although a single Decision Tree Classifier performed poorly in our experimentation as stated above, Decision Tree Classifiers were used in previous studies, and it was worth seeing if it would perform better underneath an AdaBoost structure. In general, boosting was shown Laput, Paruthi, and Singhal to be an effective ensemble model for stock prediction [9].

3.4.7 Bagging

Bagging, which is also known as bootstrap aggregating, is an ensemble method that"aims to reduce the variance of estimates by averaging multiple estimates together" [24]. In practice, this entails utilizing multiples of the same weak classification model. In this case, we used 30 Logistic Regression Classifiers. We chose to use Logistic Regressions as the base estimator for this ensemble model because, as will be shown the Results chapter, out of all the single models, Logistic Regressions performed the best.

A bagging classifier works by creating random subsets of the original data and using one of the random subsets to train one of the single classifiers, in our case a Logistic Regression. Then each of the randomly trained classifiers comes together as a 'panel of experts' to vote on each of the testing samples in order to arrive at a final classification [24]. An illustration of our Bagging Classifier is shown below.

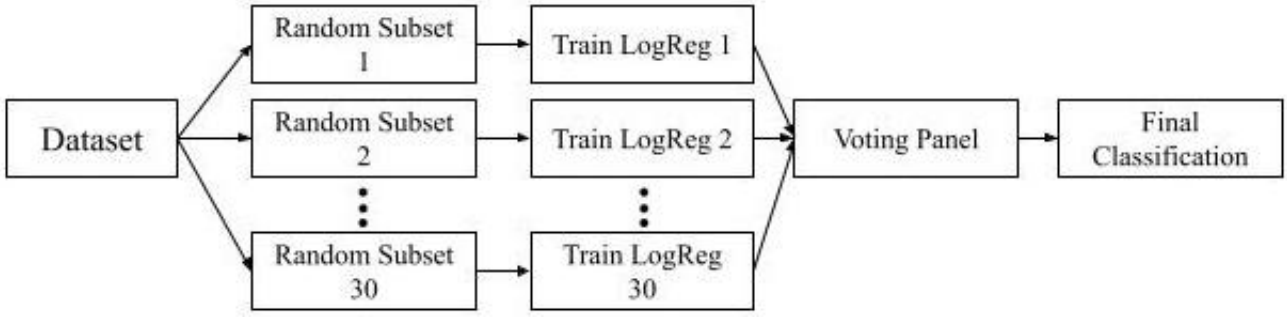


Figure 3.4: Diagram of Bagging Classifier

3.4.8 Stacking

Stacking, also known as 'stacked generalization', works similarly to Bagging with a few differences [24]. While bagging uses multiples of the same base estimator, Stacking uses several different estimators. Unlike Bagging, each estimator in Stacking is trained on the entirety of the training data. Finally, instead of all the classifiers converging to vote on the final classification of the model, stacking utilizes a final layer of another classification model.

This 'final estimator' takes as input the predictions from the base estimators. The final estimator is trained, using cross validation, on the predictions of the base estimators and the true classification of the observation. When it is time to make a prediction, the final estimator's output of the final estimator is the prediction of the model. A diagram of our Stacking Classifier is shown below.

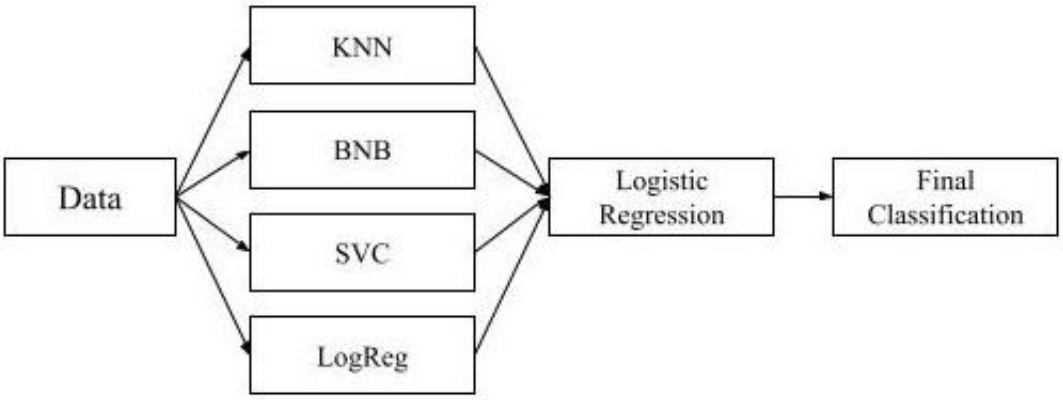


Figure 3.5: Diagram of Stacking Classifier

In our stacking classifier, as our base estimators, we used all four of the prominent stand-alone models (K-Nearest Neighbors (KNN), Logistic Regression (LogReg), Support Vector Classifier (SVC), and Bernoulli Naïve Bayes (BNB)). As the final layer, we leveraged a second Logistic Regression. As the output of the base estimators is the input to the final estimator, stacking uses the strength of each individual base estimator. Overall, the method, in combining the estimators, reduces the biases existing each individual model.

3.4.9 Cross Validation

One of the significant and most common problems in classification problems is 'overfitting' which happens when a classifier simply remembers the *actual* samples and their true classification instead of learning information about the data [24]. To combat this problem, data is commonly separated into training, validation, and testing sets. In this way, the model learns information about the data on the training data, confirms that the information learned is effective on the validation data, and is tested on samples it has never seen before from the

testing data. This is an effective way to avoid overfitting, but it has the unwanted effect of severely limiting the amount of data the models are trained on. As a fix to this problem, we used the common technique of cross validation.

Cross validation involves segmenting the data into a training and testing set, removing a strictly defined validation set. The training set is then split into k different equal sets or 'folds'. The model is then trained k times on the split training data. At each iteration, one of the k folds serves as the validation set. Then the final 'weights' of the classifier are the averages across each of the k training iterations and tested on the testing data. While this process is computationally expensive, it maximizes the amount of data on which the models are trained. An example of five-fold cross validation, the method used for all models in this thesis, is shown in the figure below [24].

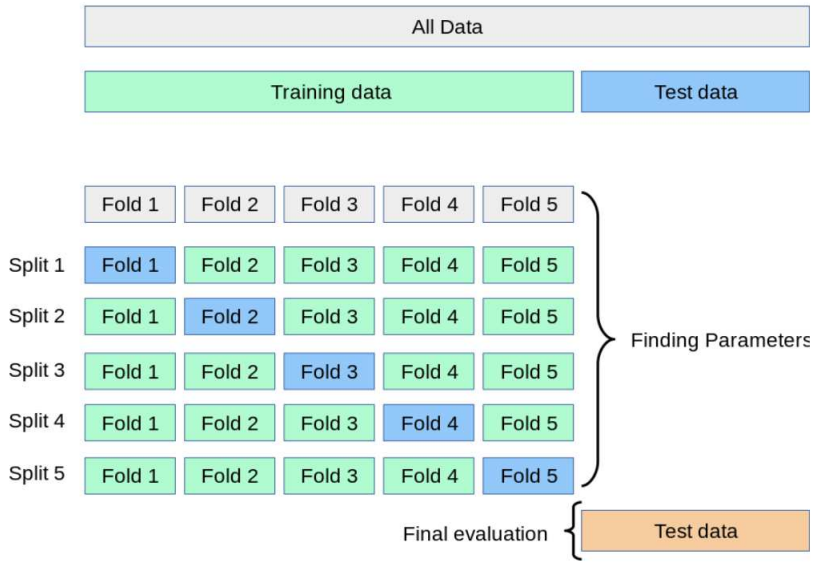


Figure 3.6: Five-Fold Cross Validation

Chapter 4

Implementation

In the implementation of the code required for this project, we used several different open source libraries and tools. We choose Python for the programming language due to not only our familiarity with the language, but also because of the vast ecosystem of libraries compatible with Python that make it an ideal language for machine learning. We used Anaconda to create a Python virtual environment and to also serve as our Python package manager. Finally, we utilized Visual Studio Code from Microsoft as the code editor. While this might seem like an irrelevant piece of information, this choice was very intentional. VSCode has the ability to edit and run Jupyter Notebooks from within the editor which proved key to visualizing data as it was being manipulated and ensuring that it was being executed in the correct way. In a Jupyter-style notebook, printed data is automatically formatted in a cleaner and more structured way that simply printing data through the terminal.

For data manipulation, we relied heavily on pandas, an open source data analysis and manipulation tool. All of the data and results were stored, manipulated, and analyzed in pandas DataFrames. Pandas was particularly helpful as DataFrames can be used as inputs to the machine learning models. All data visualization from confusion matrix heat maps to different charts and graphs, was completed using seaborn, an open source data visualization library that is built on top of matplotlib.

For implementing the selected machine learning models, we utilized scikit-learn, an open-source machine learning API that was started as a Google Summer of Code project in 2007 [24]. We choose to use scikit-learn for its powerful algorithms paired with its easy to use interface. Furthermore, there is a robust online community that frequently provides

examples of how to implement scikit-learn most effectively. In addition, scikit-learn not only provides descriptive documentation for each model, but also discussions and guides how each model works mathematically. This documentation was extremely helpful in selecting which models to train both for single models and especially for ensemble models. All machine learning models, single and ensemble alike, were implemented using scikit-learn.

The last section of the coding portion of this thesis was cleaning all earnings calls and transforming raw text into usable financial features. This process is described at length in section 3.2.1. In total, three different libraries were used to complete this NLP portion. For processing the earnings calls, in terms of cleaning the text of the earnings calls (lemmatizing, tokenizing, and removing punctuation and stopwords), a platform called Natural Language Toolkit (NLTK) was used. NLTK is a Python tool that facilitates creating and executing NLP processes [25]. In terms of NLP, it is one of the most widely used tools when coding in Python. Through NLTK, we are able to download lists of English stop words and create a tokenizer and lemmatizer which could be applied to our transcripts.

Two separate open source libraries, pysentiment2 and textstat, were employed to extract linguistic features and scores from the calls after they were cleaned. pysentiment2 is a "library for sentiment analysis in dictionary framework" [26]. Fortunately, two of the libraries available in pysentiment2 are the General Inquirer Dictionary and the Loughran-McDonald Financial Dictionary, the two dictionaries we examine in this paper.

The second library, textstat, was used to extract the two complexity scores from each earnings call. textstat is described as an "easy to use library to calculate statistics from text" [27]. Two functions from this library, flesch_reading_ease and smog_index, were used to calculate the Flesch Reading Score and SMOG Index Grade features in all earnings calls. Input to these two functions used raw text and did not require the cleaning from NLTK. These two open source resources were chosen as they were identified as two of the most widely used Python NLP libraries.

Chapter 5

Evaluation

5.1 Metrics

Before discussing the findings of this thesis, it is important to first identify how the results will be evaluated. There are several main ways we will evaluate the work of this paper. This section will also go into the iterations of experiments we ran.

First of all, in any classification problem, the main evaluation method is accuracy, and this is the main evaluation metric we will use. Accuracy is simply defined as the number of correctly classified samples over the total number of samples. For this paper however, we will add another level to how we evaluate success in terms of accuracy. Due to the difficulty of this problem as financial data has a great deal of noise, we anticipated before undertaking this task that the pure accuracies would not be be overly impressive. This hypothesis was confirmed by the related studies reviewed which all discussed the difficulty of predicting stock price movement. As such, we most importantly wanted to look at the accuracy of our models versus a baseline accuracy. For our baseline, we used a simple model that simply predicted 1, or stock price increases, every time. We used this baseline as the movement of stock prices was remarkably evenly split with percentages very close to 50% increase, 50% decrease for both data sets. All results will have a row highlighting overall accuracy and another that examines accuracy compared to the baseline. Another reason we wanted to examine accuracy versus baseline is to allow us to examine if adding earnings call linguistic features holds any predictive power. This cannot be accomplished by simply looking at pure accuracy, but must be compared to something.

To visualize accuracy and to understand what contributes to the accuracy of a classifier, we relied on confusion matrices. A confusion matrix is a table that visualizes the performance of a classifier. An example of a confusion matrix in a binary classification problem is shown in Figure 5.1 below [28].

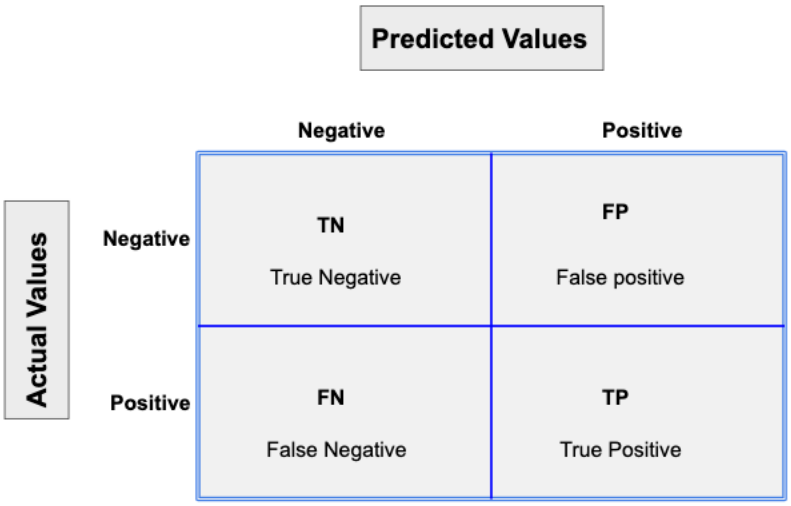


Figure 5.1: Binary Classification Confusion Matrix

In the context of this thesis, we define negative, or 0, as the stock price decreased on the trading day of the sample. We define positive, or 1, as the stock price increased on the trading day of the sample.

The bottom right quadrant of Figure 5.1 shows the amount, in proportion or percentage, of true positive examples that are classified correctly as positive. The top left quadrant shows how many of the truly negative samples are correctly classified as negative. The false negative quadrant holds the number of positive samples misclassified as negative and vice versa for the false positive quadrant. From a confusion matrix, we are able to calculate several more evaluation metrics common to machine learning and classification problems. In the following descriptions of these metrics, 'positive', denoted as P refers to the sum of true positive classifications and false negative classifications. 'Negative', denoted as N refers to the sum of true negative predictions and false positive predictions.

The other evaluation metrics complementing accuracy are precision, recall, speci-

ficity, false positive rate, false negative rate, and f1 score. Precision is calculated as $\frac{TP}{(TP+FP)}$, and a perfectly precise classifier would have a precision of 1. Recall, also known as sensitivity or true positive rate, follows the expression $\frac{TP}{P}$ and denotes the proportion of positive predictions that were truly positive. The opposite is specificity, also referred to as the true negative rate, which is shown using $\frac{TN}{N}$. A perfect classifier would have a recall and specificity of 1. False positive rate is $\frac{FP}{N}$ and false negative rate is $\frac{FN}{P}$. Finally, f1 score is a score that combines recall and precision in the following way:

$$\frac{2 * precision * recall}{precision + recall}$$

and can be interpreted as a weighted average of precision and recall. A perfect classifier would have a f1 score of 1.0 while a completely wrong classifier would have a f1 score of 0.0.

While the function of these metrics is to provide a deeper insight into the performance of a classifier outside just accuracy, they also provide useful contextual information into the ability to apply the classifier in real life situations. The ultimate goal of any predictive model is to deploy the classifier into use in real life whether it is in finance, health, or policy. One of the key aspects to understand about a classifier is where does the classifier miss. Depending on the task, there might be a preference for a classifier that is more accurate in positive cases over one that is one accurate in negative cases. In an extreme example, if a classifier was used to predict whether a patient has cancer or not, it would be important for the classifier to have a higher recall, or, in other words, catch more cases of cancer than a classifier with a lower recall, but an overall higher accuracy. While the problem in this thesis does not have nearly as high of stakes, there is an opportunity to use our results to effect real-life investment decision making and analyzing these metrics remains important.

The final metric we analyzed was the coefficients of each feature in the classification problem. The magnitude of coefficients, or weights, given to each feature by a classifier reveals which variables are most important to the classification of stock price movement. An

analysis of the coefficients will reveal if our extracted linguistic features have an important effect, and therefore if earnings calls have predictive power, in predicting stock price movement. In addition, a comparison of the coefficients of the linguistic scores against one another will reveal if different linguistic features are more important than others in the context of our task.

5.2 Sequence of Experiments/Tests

To restate, the goal of the thesis is to use a combination of financial data and linguistic analysis of earnings call to predict the movement of a company's stock price. Through this goal, we identified a second goal of observing if the linguistic analysis of an earnings call has predictive power over the movement of stock prices specifically the day after the earnings call. In pursuit of these goals, we identified a sequence of experiments that allow us to reach conclusions to both objectives. Overall, we began by conducting all experiments on all trading days regardless of whether a earnings call happened on the day or not. In order to evaluate our second goal of exploring the predictive power of earnings calls, we isolated the samples that contain earnings calls and analyzed just those samples using the same metrics as stated above.

First, we conducted two initial experiments using just financial data and just linguistic score data with the single classifiers to observe accuracies when only one of the two were used. Next, after evaluating the results of these two tests, we trained models using both the financial data and linguistic analysis scores together. Finally, we finished testing using the ensemble models with the financial data and linguistic data again combined. Each of these experiments were conducted on both Dataset Created and Dataset Pro.

Chapter 6

Results

6.1 Using Just Financial Data

The first set of tests were run on just financial data with the no linguistic scores. However, we included a single binary variable 'Earnings Call' that indicated if there was an earnings call on the day before the sample. So, in total, the data the models were trained on only features 2-5 found in the data summary table (Figure 3.3) in addition to an earnings call flag. Additionally, during this experimentation, the variable YesterdayMovement was converted to a binary variable instead of a continuous variable as some classifiers were unable to handle negative values. Other financial data was excluded as they are part of earnings reporting. Only single models were run on this experiment and the results from this experiment determined what single model classifiers were going to be explored more in-depth. As such, models that do not have a section in Models (Section 3.4) are included.

Figure 6.1 on the next page depicts the results of training models on just financial data with the earnings call flag on Dataset Pro. As a note, the '5' in front of all classifier names is simply indicating the model used five-fold cross validation. Overall, there was a very small increase in accuracy over our baseline (which is shown as 0.000 on the x-axis). Overall, there is an average increase over the baseline of 2.44% across all classifiers, a small increase considering that other studies achieved similar improvements over baselines while looking at more difficult classification tasks. Looking closer, several classifiers including Gaussian Naïve Bayes and Multinomial Naïve Bayes barely improve over the baseline. In addition, a few classifiers such as Logistic Regression and K-Nearest Neighbors, separate themselves

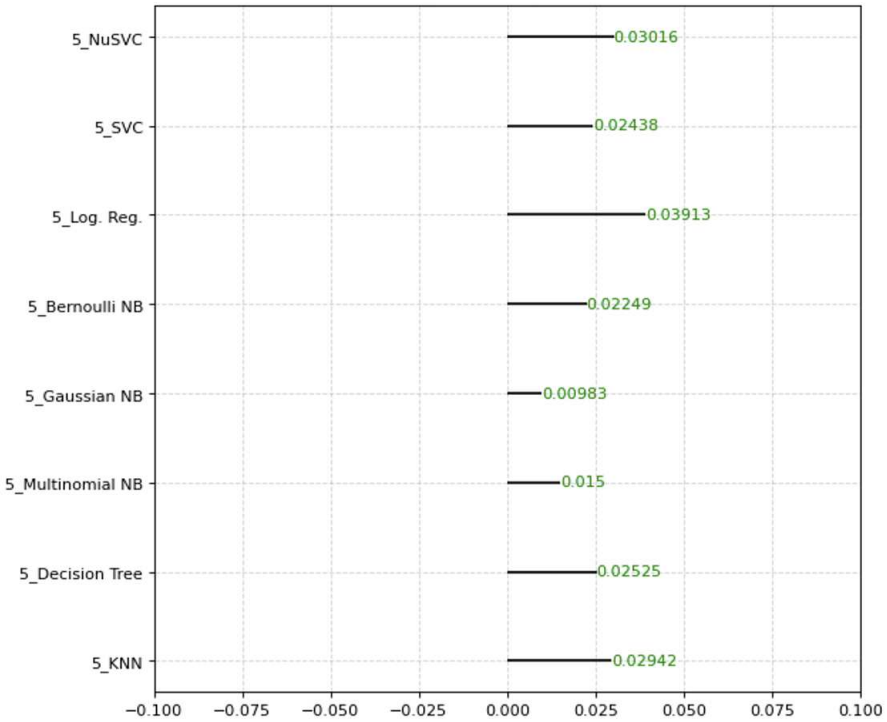


Figure 6.1: Just Financial Data – Dataset Pro

from the rest of the classifiers, indicating they might be more suited to this classification task.

We ran the same experiment with just financial data using Dataset Created. Figure 6.2 below shows the experiments with the same features selected for training and prediction.

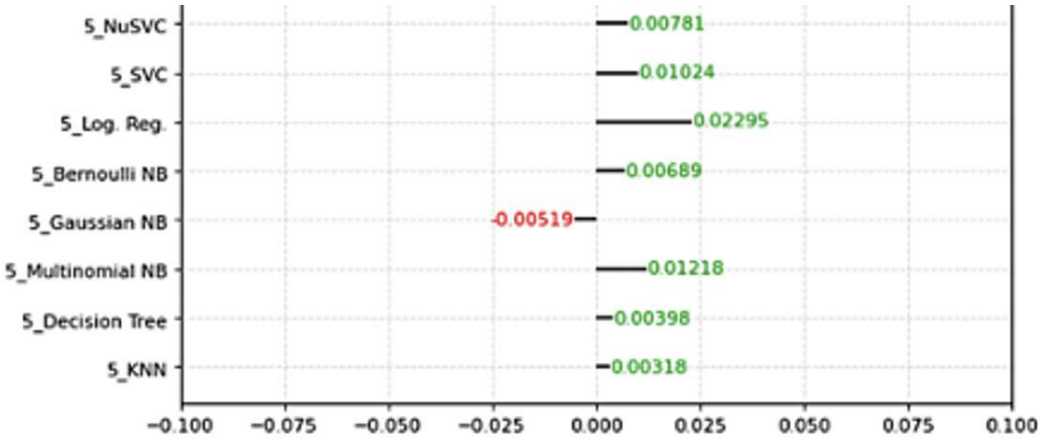


Figure 6.2: Just Financial Data – Dataset Created

Using Dataset Created shows much poorer results than the same test on Dataset

Pro. One classifier, Gaussian Naïve Bayes actually performed worse than the baseline while the across all classifiers, there was an average improvement of less than 1% (0.78%) over the baseline. However, Logistic Regression and SVC again performed the best over the baseline, an early indication they might be best suited to this prediction problem.

6.2 Using Just Linguistic Data

The second set of experiments were conducted using mainly linguistic score features with a small amount of financial data. The 'just linguistic data' tests included all LM and GI Dictionary positive and negative sentiment scores (features 20-21 and 24-25 in the features table) in addition to YesterdayClose and YesterdayMovement again as a binary variable (features 2 and 3). We included the two financial features because, otherwise, most of the observation days would have no information from which to make a prediction.

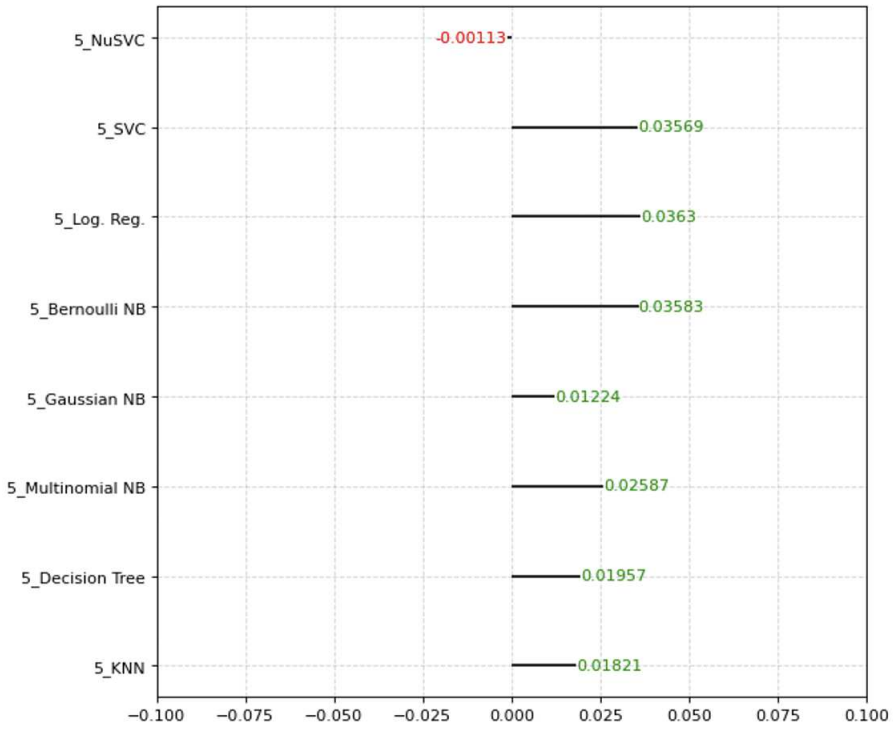


Figure 6.3: Just Linguistic Features – Dataset Pro

Figure 6.3 shows the results of the just linguistic feature classifier on Dataset Pro.

Here, the average increase over baseline accuracy was 2.6%, extremely comparable to the results using just financial data. With these results, we continue to see the same trend with the same few classifiers outperforming the others. Compared to using just financial data, several classifiers performed better using the variables included in this experiment which is the first indication that linguistic analysis of earnings calls may hold predictive power. In addition, the average increase over the baseline was slightly more in this test that the previous one.

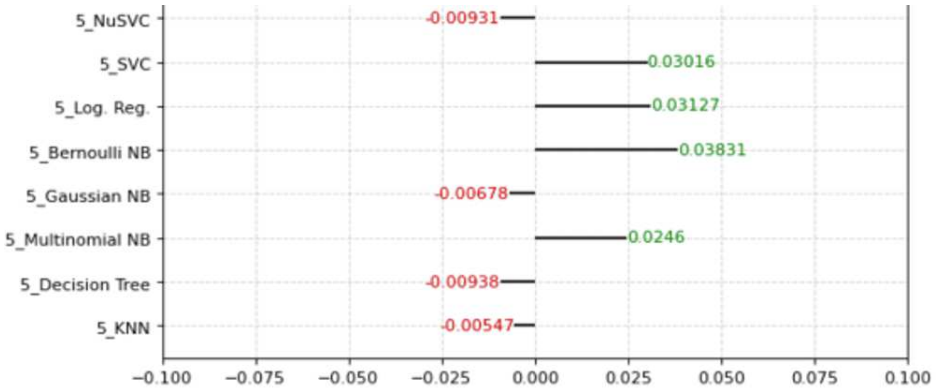


Figure 6.4: Just Linguistic Features – Dataset Created

Figure 6.4 above shows the results of the same test using Dataset Created. In these results, we see a much more mixed bag of results with four classifiers outperforming the baseline and four classifiers under performing the baseline. However, the average improvement over baseline accuracy was 1.1% which is slightly better than the improvement just using financial data with Dataset Created (0.78%). While this is not enough whatsoever to conclude that linguistic features of earnings calls have predictive power, it served as a positive sign we were moving in the right direction at this point in our work. Even more clear than with Dataset Pro, three classifiers in particular, Bernoulli Naïve Bayes, Logistic Regression, and Support Vector Classifier showed the most promise for this task.

In all of the past four results, there was a small, but consistent improvement over baseline accuracy. This suggests that the data features included have predictive power (especially YesterdayClose and YesterdayMovement, the two variables included in both tests).

However, no tests show strong improvement over accuracy suggesting the need to explore combining financial data and linguistic scores together in the next iteration of testing.

6.3 Model Reduction and Feature Tuning

After these first two iterations of testing, we evaluated all models and input variables before moving forward. In the interest of time and only focusing further iterations of testing on the most promising models, we removed four of the original singles models. To decide which models to remove, we looked at average performance compared to the baseline across all four results shown above for each model. We removed Gaussian Naïve Bayes ($< 1\%$ average increase over baseline), Nu-SVC ($< 1\%$ average), Multinomial Naïve Bayes (1.9% average), and Decision Tree ($< 1\%$ average). This left us with four single models to explore in-depth with financial data and linguistic scores combined. K-Nearest Neighbors actually had a smaller average increase than Decision Tree (1.1% vs 1.9%); however, KNN was kept instead of Decision Tree as Decision Tree was identified as a classifier to be used in the AdaBoost ensemble model. In addition, we were encouraged by KNN's strong performance in the just financial data experiment on Dataset Pro (Figure 6.1). With the reduction in models, we were left with four single models: Logistic Regression (LogReg), Support Vector Classifier (SVC), Bernoulli Naïve Bayes (BNB), and K-Nearest Neighbors (KNN).

After the first two iterations of testing, some of the features were changed in an effort to be more descriptive. As more information about the earnings call, which included more linguistic analysis and reported financial results, was added after these tests, we removed the earnings call flag as the inclusion of the earnings report information was sufficient to show an earnings report event. Secondly, as described above, the 'YesterdayMovement' variable was reported as a binary variable in the just financial data and just linguistic feature experiments. In order to give the classifiers more information, as we know recent stock price history is a strong indicator of next day performance, this variable was changed to a continuous variable

that detailed the signed magnitude of the stock price change the day before. The reduction from eight to four classifiers also enabled this change because some of the original classifiers were unable to handle negative values.

6.4 Linguistic and Financial Data Combined

The second iteration of testing used all 28 independent variables detailed in the data summary table in Figure 3.3. This section will report results on using all features as training and testing data for the four single models that were identified in the section above. Each table will report all metrics discussed in the evaluation chapter and one of the most important rows in the tables will be the "Versus Baseline" row that is highlighted in green or red depending if the classifier over or under performed the baseline.

Eval Metric	Baseline	KNN	SVC	LogReg	BNB	Averages
Accuracy	48.47%	53.08%	52.59%	**59.40%**	52.38%	54.36%
Versus Baseline	0.00%	4.61%	4.12%	**10.93%**	3.91%	5.89%
Precision		51.47%	51.21%	**59.91%**	50.81%	53.35%
Recall		**50.61%**	40.15%	48.28%	45.76%	46.20%
Specificity		55.39%	64.23%	**69.79%**	58.57%	61.99%
False Pos Rate		44.61%	35.77%	**30.21%**	41.43%	38.01%
False Neg Rate		**49.39%**	59.85%	51.72%	54.24%	53.80%
F1 Score		51.03%	45.01%	**53.47%**	48.15%	49.42%
Total Obser.		9544104	9544104	9544104	9544104	9544104

Figure 6.5: All Features Single Classifiers – Dataset Pro

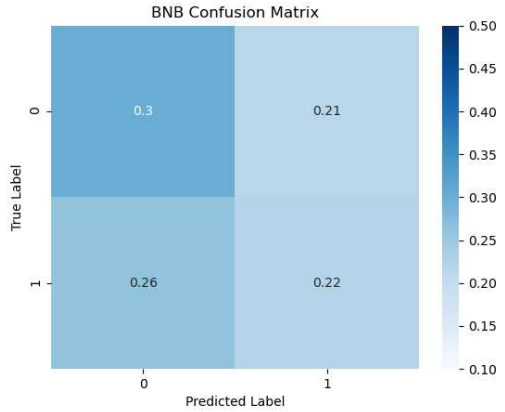
Figure 6.5 shows the results of using both financial data and linguistic scores to train each of the single classifiers. Here, 48.47% is the baseline accuracy which means on 48.47% of the over 9 million company trading days, the stock price increased. Overall, combining the linguistic scores and financial data, changing 'YesterdayMovement', and including additional financial data reported during earnings calls have a significant positive effect on the performance of the classifiers. Compared to 2.44% and 2.6% from the first iteration of testing, there is an average improvement over the baseline accuracy of 5.89%. Even when

you remove the strong improvement generated by the Logistic Regression, there is a still an average improvement of 4.21% over the baseline, much greater than the improvement from the first round of tests.

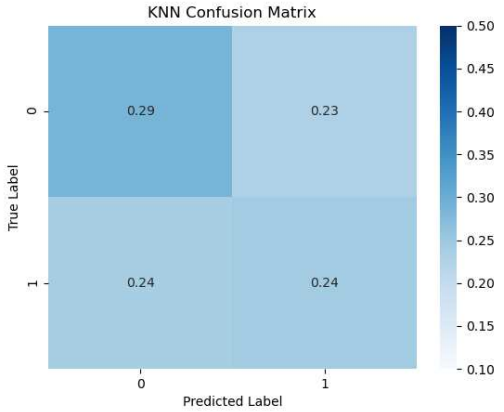
While Logistic Regression is the best classifier in this iteration of testing by a comfortable margin, it does not perform the best in all evaluation metrics. Specifically, KNN has a higher recall meaning that KNN correctly identifies positive, or price increase, samples more effectively than Logistic Regression. Furthermore, as false negative rate is simply 1 - Recall, KNN has a lower false negative rate. However, Logistic Regression has a higher F1 Score, showing recall and precision together operates at a higher average than KNN. Next, we will examine the confusion matrices generated by this test.

Figure 6.6 on the next page shows the confusion matrices for each classifier that corresponds to the table of results on Dataset Pro, Figure 6.5. The number in each quadrant corresponds to the percentage, in decimal form, of total observations that fall into this category where all four percentages should add to 1. It should be noted that some confusion matrices in this paper may not add to exactly 1 due to rounding. A higher proportion corresponds to a darker shade of blue. A perfect classifier would be dark blue in the top left and bottom right quadrants (the true negative and true positive quadrants) and completely white in the top right and bottom left. From these confusion matrices, we see all classifiers struggled with correctly classifying both positive and negative classes. The BNB and KNN matrices show that they struggled with both classes relatively equally by the roughly equal shades of blue in all quadrants. In contrast, the SVC matrix shows the classifier struggled greatly with correctly classifying positive, or price increasing, samples, shown in the very light shade of blue in the bottom right quadrant. The LogReg matrix is starting to resemble a more perfect matrix with the darkest shade in the top left and a relatively dark shade of blue in the bottom right. This translates to the high overall accuracy of the logistic regression classifier.

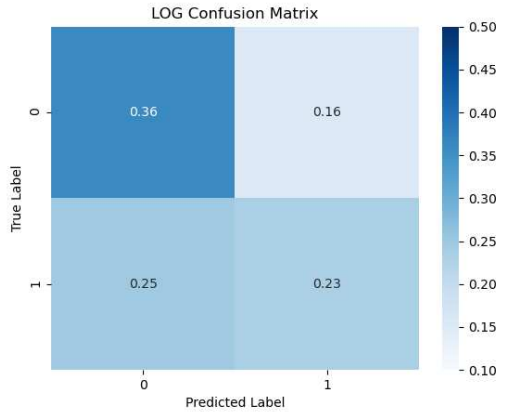
The most important idea to take from this figure of confusion matrices is that it



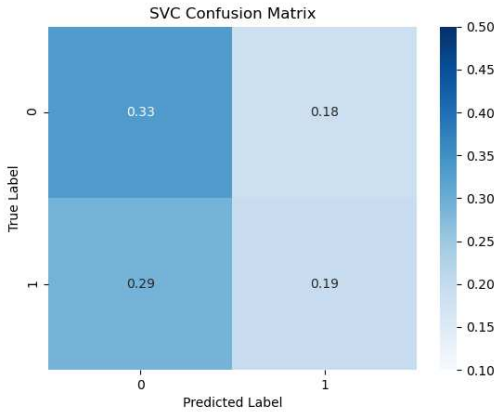
(a) BNB Confusion Matrix



(b) KNN Confusion Matrix



(c) LogReg Confusion Matrix



(d) SVC Confusion Matrix

Figure 6.6: Single Classifiers Confusion Matrices – Dataset Pro

reveals the potential for ensemble models to work effectively. As stated above in the classifier subsection, section 3.4, ensemble models perform well when the classifiers are diverse and the model is able to draw on different strengths. The confusion matrices reveal that each of the four single model classifiers are strong in different areas as no two matrices look exactly the same. This was the primary motivation as to why ensemble models were explored.

Next, we will discuss the results of the same experiment now using Dataset Created as the dataset in Figure 6.7 on the next page.

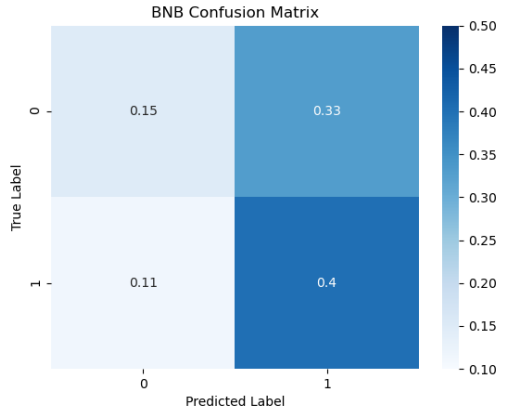
The first thing to note is the marked difference in number of observations between Dataset Created and Dataset Pro. While Dataset Pro had over 9.5 million observations,

Eval Metric	Baseline	KNN	SVC	LogReg	BNB	Averages
Accuracy	51.78%	53.78%	55.29%	**55.90%**	55.42%	55.10%
Versus Baseline	0.00%	2.00%	3.51%	**4.12%**	3.64%	3.32%
Precision		**55.24%**	54.22%	55.03%	54.91%	54.85%
Recall		56.65%	**87.82%**	81.16%	77.82%	75.86%
Specificity		**50.70%**	20.35%	28.78%	31.36%	32.80%
False Pos Rate		**49.30%**	79.65%	71.22%	68.64%	67.20%
False Neg Rate		43.35%	**12.18%**	18.84%	22.18%	24.14%
F1 Score		55.93%	**67.04%**	65.59%	64.39%	63.24%
Total Obser.		51278	51278	51278	51278	51278

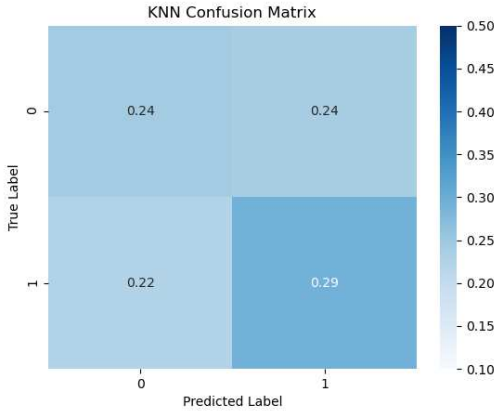
Figure 6.7: All Features Single Classifiers – Dataset Created

Dataset Created as just over 51,000. Secondly, the baseline accuracy is 51.78% compared to 48.47% in Dataset Pro. There are several possible reasons for this significant different. First of all, Dataset Pro covers a period that includes the 2007-2008 global financial crisis where the stock market lost an large amount of value. Furthermore, Dataset Created covers four years, 2017-2020, that are not seen in Dataset Pro and that saw sustained growth in the markets. While it is true that Dataset Created covers the first few months of the COVID-19 Pandemic, after an initial sharp sell-off after the world shut down, the stock market saw strong growth in a bull market that still continues today. As such, it is unsurprising the baseline accuracy is higher in Dataset Created.

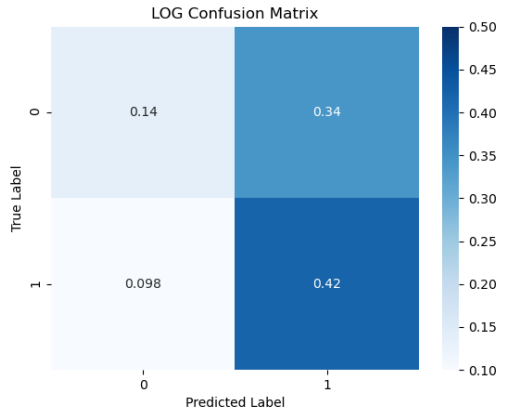
Similarly to the results on Dataset Pro, using all the features produces a stronger average improvement over the baseline compared to the first iteration of testing (3.32% verus 1.1% and 0.78%). Logistic Regression, again, performed batter than all other single classifiers in terms of overall accuracy. However, other than accuracy, it was not the best classifier in any other metric with SVC outperforming it in recall, false negative rate, and f1 score and KNN doing better in precision, specificity, and false positive rate. Curiously, both Logsistic Regression, SVC, and BNB have a very high recall and therefore a low false negative rate. However, this is accompanied by a very high false positive rate. As will be more clearly visualized in a discussion of the confusion matrices below, those three classifiers struggled with correctly identifying negative, or price decreasing, samples.



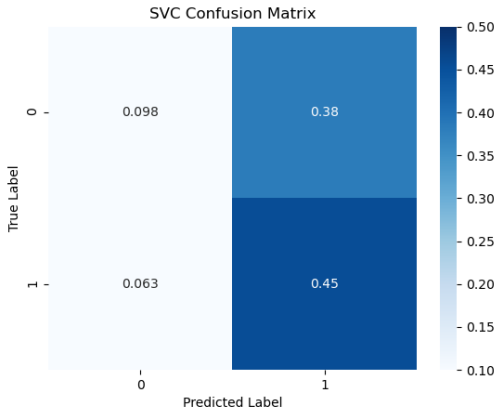
(a) BNB Confusion Matrix



(b) KNN Confusion Matrix



(c) LogReg Confusion Matrix



(d) SVC Confusion Matrix

Figure 6.8: Single Classifiers Confusion Matrices – Dataset Created

The confusion matrices in Figure 6.8 below are a strong visualization of why BNB, SVC, and LogReg have such high recall scores with low precision scores. All three classifiers have very dark shades of blue in the bottom right true positive quadrant, showing they all predict positive examples well. However, each classifier has nearly an equally dark shade of blue in the top right false positive quadrant and a nearly white color in the top left true negative quadrant. As such, these classifiers really struggled with discerning negative samples from positive examples and skewed towards predicting positive as the correct class. The KNN matrix, which is more uniformly blue in all quadrants, indicates that KNN would be the most effective 'real world' classifier even though it has the lowest overall accuracy. A

classifier that predicts 1, or that the stock price will rise, most of the time, as does SVC, BNB, and LogReg, might end up hurting investors as it would fail to catch the cases where the stock price is going to fall. As three of the matrices look similar, ensemble models may be less effective for Dataset Created, although they still will be explored.

6.5 Ensemble Models

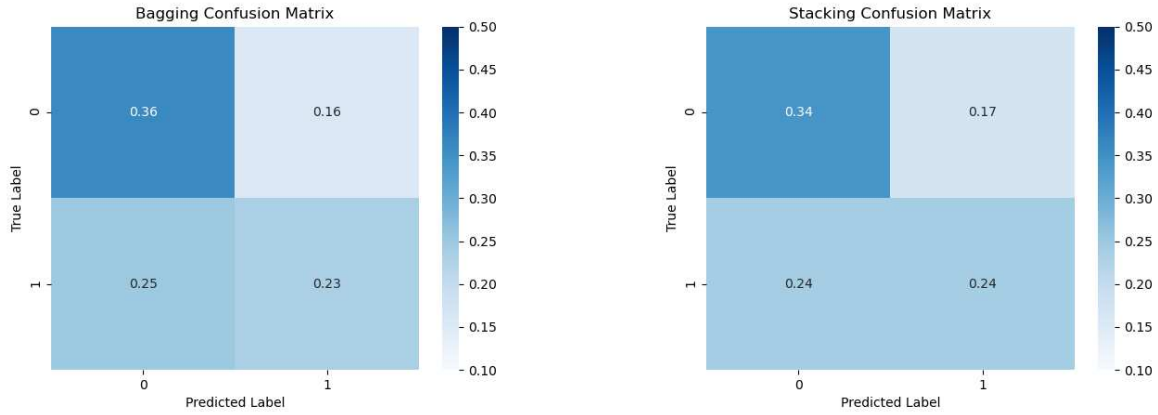
As we determined from the confusion matrices of the single models on Dataset Pro that ensemble models may be appropriate, our final iteration of testing used the five ensemble models discussed in section 3.4. Again, the models were trained with all 28 independent variables from Figure 3.3 on both Dataset Pro and Dataset Created.

Eval Metric	Baseline	H. Voting	S. Voting	Stacking	Bagging	Boosting	Averages
Accuracy	48.47%	55.01%	52.70%	58.52%	**59.36%**	52.38%	55.59%
Versus Baseline	0.00%	6.54%	4.23%	10.05%	**10.89%**	3.91%	7.12%
Precision		55.51%	51.14%	58.24%	**59.86%**	50.81%	55.11%
Recall		34.72%	47.14%	**50.04%**	48.22%	45.51%	45.13%
Specificity		**73.98%**	57.90%	66.45%	69.77%	58.81%	65.38%
False Pos Rate		26.02%	**42.10%**	33.55%	30.23%	41.19%	34.62%
False Neg Rate		**65.28%**	52.86%	49.96%	51.78%	54.49%	54.87%
F1 Score		42.72%	49.06%	**53.83%**	53.41%	48.02%	49.41%
Total Obser.		9544104	9544104	9544104	9544104	9544104	9544104

Figure 6.9: All Features Ensemble Models – Dataset Pro

In Figure 6.9 above, we detail the results from the ensemble models on Dataset Pro. As we suspected, ensemble models, on average, performed better over the baseline than the single models. The average improvement over baseline accuracy for these ensemble models was 7.12% versus 5.89% with the single models. Stacking and Bagging produced improvements over 10% and by this metric, were far away the most effective. Furthermore, a third ensemble model, Hard Voting, outperformed all single classifiers except for Logistic Regression. However, Hard Voting has a very high specificity and a low recall, meaning it struggled with positive samples similar to some of the single models seen above. Stacking and Bagging on the other hand, had the highest two precision and recall percentages meaning

they did not significantly favor predicting one class over the other. Finally, this assertion is reinforced by the fact the two highest f1 scores are from Stacking and Bagging. While the ensemble models performed better than the single classifiers on average, Logistic Regression turned in the highest overall accuracy over baseline of all single and ensemble models with 10.93%.



(a) Bagging Confusion Matrix

(b) Stacking Confusion Matrix

Figure 6.10: Ensemble Classifiers Confusion Matrices – Dataset Pro

The confusion matrices above, for Stacking and Bagging, look nearly identical to the Logistic Regression from Figure 6.6c. It seems that the Logistic Regression classifier in the stacking model held the most influence. Furthermore, it is unsurprising that Bagging shows a similar confusion matrix to the single Logistic Regression as the Bagging model was made up of 30 logistic regression models. However, although both models seem to have been driven by the single Logistic Regression classifier, there are signs that stacking improved over the single Logistic Regression model. The Stacking model produced a very similar precision score to the single logistic regression, but increased its recall by nearly 2% over Logistic Regression's recall. As such, the Stacking confusion matrix looks closer to a perfect confusion matrix as the true negative quadrant is still the darkest blue while the true positive quadrant is equal to the false negative quadrant. Conversely, logistic regression had its false negative quadrant greater than its true positive quadrant.

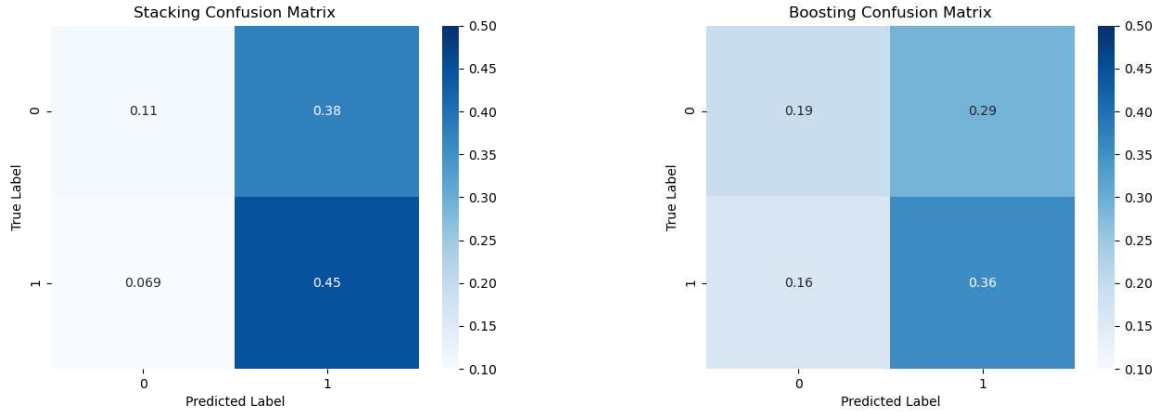
Next, we will discuss the results of the ensemble results using Dataset Created as the data.

Eval Metric	Baseline	H. Voting	S. Voting	Stacking	Bagging	Boosting	Averages
Accuracy	51.78%	55.64%	55.56%	55.54%	**55.99%**	54.87%	55.52%
Versus Baseline	0.00%	3.86%	3.78%	3.76%	**4.21%**	3.09%	3.74%
Precision		54.57%	54.60%	54.45%	55.10%	**55.14%**	54.77%
Recall		85.71%	84.21%	**86.63%**	81.08%	68.85%	81.30%
Specificity		23.36%	24.79%	22.16%	29.04%	**39.85%**	27.84%
False Pos Rate		76.64%	75.21%	77.84%	70.96%	**60.15%**	72.16%
False Neg Rate		14.29%	15.79%	**13.37%**	18.92%	31.15%	18.70%
F1 Score		66.68%	66.25%	66.87%	65.61%	**61.24%**	65.33%
Total Obser.		51278	51278	51278	51278	51278	51278

Figure 6.11: All Features Ensemble Models – Dataset Created

Here, unsurprisingly, the ensemble models did not improve accuracy by much over the single models for Dataset Created. Ensemble models did produce an average performance over baseline that was greater than the single models, but only by 0.42% (3.79% versus 3.32%). In fact, the best ensemble model, Bagging, produced nearly the exact raw accuracy than Logistic Regression, the best single classifier (55.99% versus 55.90%). Similar to Dataset Pro, we can identify the classifiers that dominate the ensemble model decision making. All classifiers except for Boosting arrived at similar accuracies to SVC and Logistic Regression single classifiers and produced precision, recall, and specificity scores near those two single models. Boosting, an ensemble model built on Decision Trees rather than the four single models, is the only classifier who displays recall under 80% and a specificity above 30%. Similar to the discussion of KNN for Dataset Created, Boosting may be a better option than the other classifiers that have higher accuracies because it does not skew as severely to predicting one class over the other.

The confusion matrices in Figure 6.12 are strong visuals to confirm the discussion about the ensemble model results on Dataset Created. The Stacking confusion matrix on the left looks incredibly similar to the confusion matrices in Figure 6.8 with the exception of the KNN confusion matrix. The three single classifiers that had similar problems (BNB, SVC, LogReg) constitutes a majority in any model and therefore has great influence over the



(a) Stacking Confusion Matrix (b) Boost Confusion Matrix

Figure 6.12: Ensemble Classifiers Confusion Matrices – Dataset Created

final prediction. The confusion matrix for Boosting on the other hand, begins to get closer to the ideal confusion matrix, even though Boosting shows the lowest overall accuracy out of the ensemble models on Dataset Created. However, it still produces a relatively strong improvement over the baseline of 3.09%. The results of the ensemble models on Dataset Created serve as evidence that ensemble models, while they can be extremely useful, require certain conditions to work most effectively.

6.6 Company Breakdown

As we review the prior two sections, a question arises as to why experiments on Dataset Pro produce significantly better results than the same experiments on Dataset Created. One reason might be due to the very small sample size of Dataset Created opposed to Dataset Pro. Dataset Created has nearly 200 times less observations than Dataset Created. With an extremely smaller sample size, there will be greater variation as models have less chance to learn the data. A second reason reveals itself when you look at the company by company breakdown of each Dataset.

The two figures in the following pages will show, first for Dataset Created, the average accuracy of the samples belonging to each of the 21 companies in the data set. The

second figure will show the 10 highest and the 10 lowest companies by average accuracy from Dataset Pro. For both Datasets, the average accuracy was calculated across the five ensemble models shown in the section above.

Company	Averages
AMZN	**64.44%**
AAL	64.24%
AAPL	59.57%
LMT	59.06%
FB	57.92%
INTC	57.57%
WMT	57.44%
TGT	56.06%
QCOM	55.94%
T	55.94%

Company	Averages
GOOGL	55.83%
BA	55.31%
PEP	54.87%
GS	54.63%
KO	54.26%
TSLA	52.07%
BAC	51.26%
MS	50.73%
JPM	50.59%
NFLX	49.90%
KOS	46.64%

Figure 6.13: Dataset Created Company Breakdown By Accruracy – Sorted High to Lowest

Figure 6.13 shows the company-by-company breakdown of the 21 companies in Dataset Created. The accuracies range from a low of 46.64% (KOS) to a high of 64.44% (AMZN). 17 out of the 21 companies, or 81% fall into a range from 50% to 60%, showing that, for this Dataset, most companies show fairly similar performance with few companies falling outside of this range at both extremes.

Company	Average
OLY	94.35%
LINK	91.81%
MLSS	87.86%
MATH	81.19%
MITK	80.48%
TTP	80.48%
MMA	78.31%
NLTX	77.45%
FMP	74.39%
TARO	73.85%

(a) Highest 10 Dataset Pro Companies

Company	Average
MIK	48.89%
WLKP	48.84%
DCN	48.56%
BKFS	48.39%
FCB	48.12%
OCUL	47.76%
PFNX	47.74%
BOOT	46.91%
TRCO	45.42%
Z	42.83%

(b) Lowest 10 Dataset Pro Companies

Figure 6.14: Dataset Pro Company Breakdown by Accuracy

In contrast, observations of the same company breakdown for Dataset Pro show that there is a far greater range of average accuracies. Figure 6.14 shows the 10 highest performing companies against the 10 lowest performing companies. The range in this Dataset increases from Dataset Created to a low of 42.83% to a high of 94.35%. While it is true that the

vast majority of companies fell into the range of 50% to 60% as seen with Dataset Created (see Figure A.3 in the Appendix), Dataset Pro had 76 companies over 64.44%, the highest average seen in Dataset Created, and 231 companies over 60%. On the other hand, Dataset Pro had 48 companies under 50% and only 2 under 46.64%, the lowest average from Dataset Created. Dataset Pro had a smaller percentage of companies over 60% than Dataset Created as there are 4,529 firms in Dataset Pro, but the hundreds of companies that averaged well over 60% and even into the 90% range are enough to increase the overall accuracy of Dataset Pro over Dataset Created.

Figures 6.13 and 6.14 also give insight into the fact that some companies stocks are more predictive than others. A further analysis that is not in the scope of this thesis is needed to parse out exactly why this is the case. A brief look at some of stock prices of the highest performing stocks in Dataset Created reveal they are traded far less frequently than others. Most of the stock movement, in both directions, was very small. It makes sense that stocks that do not undergo large price changes are easier to predict. However, the point that some stocks are easier to predict than others still holds true. In Dataset Created, all of the firms are some of the most traded and well-known companies in the world and there is still a fairly large average accuracy range.

6.7 Day After Earnings Calls

Possibly the most important, and certainly most interesting, section of results comes when we only look at samples from the day after earnings calls occur. This section will help answer part two of this thesis' goal: is there predictive power in a linguistic analysis of a companies earnings call on the stock movement the next day? To answer this question, we took the same predictions from the models from the the sections above, but extracted only the trading days following an earnings call. From there, we calculated the same evaluation metrics as above. The results in this section are from the ensemble models for both Datasets. The

results from the single models just earnings call samples can be found in the appendix.

Eval Metric	Baseline	H. Voting	S. Voting	Stacking	Bagging	Boosting	Averages
Accuracy	49.89%	53.88%	53.41%	53.39%	**54.54%**	53.58%	53.76%
Versus Baseline	0.00%	3.99%	3.52%	3.50%	**4.65%**	3.69%	3.87%
Precision		53.68%	53.14%	53.27%	**54.46%**	53.53%	53.62%
Recall		55.16%	**55.98%**	53.55%	54.31%	52.88%	54.38%
Specificity		52.60%	50.84%	53.22%	**54.76%**	54.28%	53.14%
False Pos Rate		47.40%	49.16%	46.78%	**45.24%**	45.72%	46.86%
False Neg Rate		44.84%	**44.02%**	46.45%	45.69%	47.12%	45.62%
F1 Score		54.41%	**54.52%**	53.41%	54.38%	53.20%	53.99%
Total Obser.		67702	67702	67702	67702	67702	67702

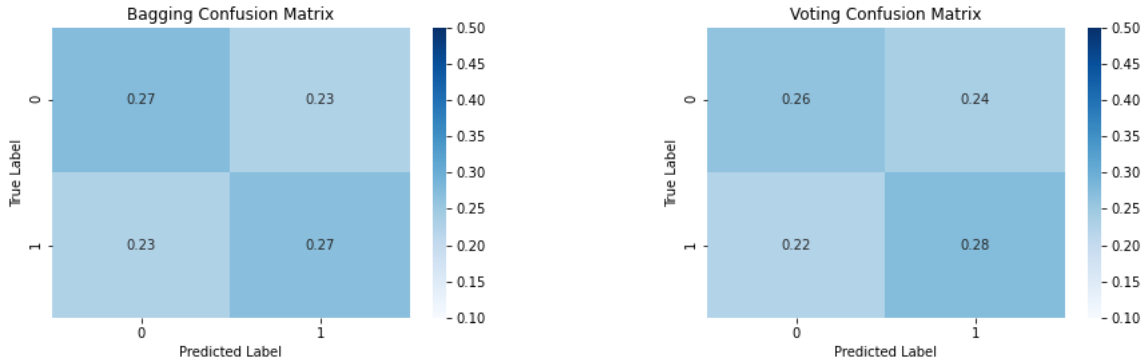
Figure 6.15: Evaluation of Earnings Call Samples – Dataset Pro

Figure 6.15 details the results from isolating earnings call samples from Dataset Pro. The first thing to notice is the baseline accuracy has increased from 48.47% with the entire observation set to 49.89% in the day after earnings calls. This means that out of the 67,702 observations, 49.89% of the time, the stock priced increased. This suggests that as an event, earnings calls have a slight positive effect on stock prices.

Secondly, all five ensemble classifiers showed significant improvement over the baseline on days after earnings calls with no classifier improving by less than 3.5% and an overall average performance increase of 3.87%. While this might not seem like significant improvement, it serves to remember that all related work papers reviewed commented on the difficulty of this problem and stated very high accuracies are not to be expected. For example, Laput, Paruthi, and Singhalfrom from the University of Michigan, although they were forecasting a slightly more selective event, were satisfied with achieving a 1% increase over their baseline [9]. Additionally, in investing, being correct on price movement even 3.87% more often can translate to significant financial gains.

Bagging continued to be the most effective classifier with a 4.65% increase over the baseline. Even more interesting than the accuracy improvements, all classifier's recall, precision, and specificity scores have leveled out, meaning no classifier was significantly skewed to predicting a single class. While in the Section 6.5 we observed false positive and false negative rates significantly higher than 60%, all false rates for all classifiers are

under 50% here, showing balanced classifiers across the board and effective learning of the differences between positive and negative samples.



(a) Earnings Call Bagging Confusion Matrix (b) Earnings Call H. Voting Confusion Matrix

Figure 6.16: Selected Earnings Call Confusion Matrices – Dataset Pro

This can be most easily visualized again through confusion matrices shown in Figure 6.16 where the confusion matrices for Bagging and Hard Voting are shown. Both matrices are closer to perfect confusion matrices than any other matrix we have seen so far in this results chapter. The darkest shades of blue are, in both matrices, in the true positive and true negative quadrants. For the first time for *both* positive and negative samples, the rate of correct classification is greater than incorrect classification.

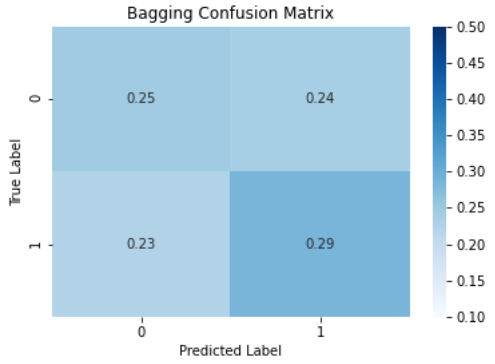
One thing to address is the significant drop in accuracy when going from all observations in Dataset Pro to just the day after earnings calls. This is most likely because the accuracies over all observations is greatly benefited from dozens of highly predictable stocks. In the day after earnings call samples, these companies have a far lesser effect, leading to lower overall accuracies. This, however, does not discount the strong improvements shown across all classifiers the day after earnings call. In fact, these results suggest that the classifiers are learning the day after earnings calls more effectively as they show better confusion matrices and false classification rates under 50%.

Moving onto Dataset Created, similar trends continue to emerge. In this case, the baseline accuracy increases to 51.25%, but all classifiers still show improvement over the

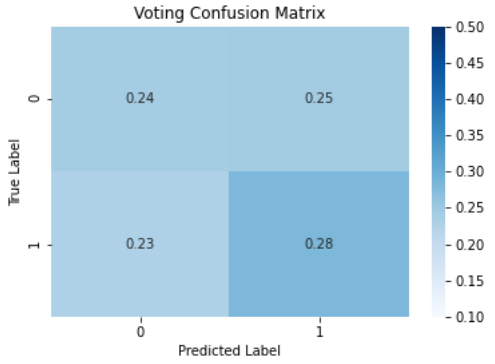
Eval Metric	Baseline	H. Voting	S. Voting	Stacking	Bagging	Boosting	Averages
Accuracy	51.25%	52.49%	53.61%	**54.60%**	53.36%	53.86%	53.58%
Versus Baseline	0.00%	1.24%	2.36%	**3.35%**	2.11%	2.61%	2.33%
Precision		53.54%	54.59%	**55.95%**	54.37%	54.71%	54.63%
Recall		55.10%	56.31%	53.64%	55.83%	**57.77%**	55.73%
Specificity		49.74%	50.77%	**55.61%**	50.77%	49.74%	51.33%
False Pos Rate		50.26%	49.23%	**44.39%**	49.23%	50.26%	48.67%
False Neg Rate		44.90%	43.69%	46.36%	44.17%	**42.23%**	44.27%
F1 Score		54.31%	55.44%	54.77%	55.09%	**56.20%**	55.16%
Total Obser.		804	804	804	804	804	804

Figure 6.17: Evaluation of Earnings Call Samples – Dataset Created

baseline. The average increase was 2.33% with Stacking being the most effective classifier with an improvement over baseline of 3.35%. Again, the results must be placed in the context of the difficulty of the problem. Similar to Dataset Pro, none of the classifiers are skewed heavily to predicting one class over the other as recall and specificity scores are similar in every model. All classifiers continue to show a greater ability than before to discern between negative and positive examples with all false classification rates falling under 50% except for the false positive rates for Hard Voting and Boosting. Even in these cases, the rate is only greater than 50% by 0.26%.



(a) Earnings Call Bagging Confusion Matrix



(b) Earnings Call H. Voting Confusion Matrix

Figure 6.18: Selected Earnings Call Confusion Matrices – Dataset Created

These strong results again translate to better confusion matrices in Figure 6.17. In both matrices, while they are not as strong as the ones from Dataset Pro, nonetheless show phenomenal improvement over the matrices from Figure 6.12. True positive samples have the

highest proportions, showing the classifiers are strongest in correctly classifying positive, or price increase, samples. Both confusion matrices show more trouble in correctly classifying negative samples, but again, these matrices are much stronger than the matrices from the ensemble models on all observation days from Dataset Created. The Bagging and Hard Voting matrices confirm the fact that the models are more effective at learning the data when just looking at the day after earnings calls.

The inclusion of earnings call features, both financial and linguistic, provide a great deal of assistance to each classifier with discerning between positive and negative samples, a trend seen in both Datasets from the confusion matrices. In addition, we see strong improvement over the baseline in all classifiers with all classifiers showing more balanced performance. Therefore, we are able to conclude that the information reported in earnings calls, reported financial metrics, performance against analysts expectations, and linguistic analysis, have predictive power. To understand if linguistic analysis has predictive power, we must look at the coefficients of each feature to see what is driving prediction.

6.8 Coefficients

Feature	Coefficient
5 Day Moving Average	3.1296
YesterdayClose	-2.4492
YesterdayMovement	1.1234
10 Day Moving Average	-0.6860
BEAT_EPS	0.0491
BEAT_OPR	-0.0420
NET_DIFF	0.0376
EBT_DIFF	0.0225
LM_Positive	0.0196
OPR_DIFF	0.0152
ciq	0.0140
GI_Positive	0.0131
LM_Polarity	0.0120
LM_Negative	-0.0120

Feature	Coefficient
EPS_DIFF	0.0113
SMOG	-0.0110
GI_Negative	-0.0085
BEAT_NET	0.0079
revtq	0.0044
capxy	-0.0035
dd1q	0.0030
GI_Polarity	0.0028
ltq	-0.0022
actq	0.0015
Flesch	0.0014
LM_Subjectivity	-0.0001
GI_Subjectivity	0.0001
BEAT_EBT	0.0000

Figure 6.19: Coefficients of Features – Logistic Regression Dataset Pro

Figure 6.19 above details the coefficients, ranked from highest to lowest magnitude, of all features in the Datasets. The coefficients in the table are obtained from the Logistic

Regression model on Dataset Pro on scaled data, meaning the magnitude of coefficients can be directly compared. Logistic Regression is the only classifier that provides feature coefficients, and as logistic regression also proved to be the most effective single model, we are able to have confidence in its coefficients.

The 10 features that are concerned with linguistic analysis are highlighted in yellow. Looking at the tables, price history features are listed as the top five most important classifiers by a large margin which is unsurprising as it is well known that the trends of a stock have a great deal of influence on the price direction in the future. We are most interested in comparing linguistic features to other metrics that are reported during an earnings call. The top feature from this category is BEAT_EPS which, again, is unsurprising as it is a common measure of the success of an earnings report.

Turning to linguistic analysis features, positive and negative sentiment scores are more predictive than other sentiment features as all positive and negative scores are among the top six linguistic features. LM_Positive, the positive sentiment calculated using the Loughran-McDonald Dictionary, is the most influential linguistic analysis feature. Polarity and subjectivity, with both subjectivity scores at the bottom of the list and GI_Polarity only slightly above them, do not have as large an effect on the prediction. The complexity of the call, measured by Flesch and SMOG, show a mixed amount of influence. The Flesch Reading Ease score has minuscule importance with a very small coefficient while the SMOG Index Grade exhibits a coefficient comparable to some of the sentiment scores.

It is encouraging to see that the signs of the coefficients match the meaning of the feature. Both positive sentiment scores have positive coefficients which translate to an increased probability of a positive, price increase prediction. Likewise, both negative sentiment scores have negative coefficients. The negative coefficient of SMOG indicates the more complex a call, the less likely for there to be a price increase, showing the importance of executives to communicate in an easy to understand manner on an earnings call.

Moreover, we find that the Loughran-McDonald (LM) Financial Dictionary is more

useful for this linguistic analysis task than the General Inquirer (GI) Dictionary as each LM score has a higher coefficient than its GI counterpart. This shows that the LM Dictionary is more effective at extracting predictive information when predicting stock prices from earnings calls. It serves as proof that the work of Loughran and McDonald in trying to build a better dictionary for financial linguistic analysis was indeed successful. Furthermore, this underscores the point that *context* matters a great deal and must be considered in linguistic analysis.

Finally, and most importantly, from the combination of the performance results on the day after earnings calls seen in Section 6.8 and the coefficients, we can conclude that linguistic analysis of earnings calls have predictive power over the movement of the stock price the next day. The results from the prior section show including information from earnings reports increase the ability for classifiers to learn the data. The coefficients show us that increased performance is driven by linguistic features. Looking at the coefficients, all positive and negative sentiment scores are more important than other reported raw financial metrics except for ciq (Reported Comprehensive Income). The reported financial numbers are all aspects of the earnings reports that are analyzed and used by financial analysts. While reported metrics compared to analysts expectations are indeed important, we find that several linguistic analysis scores show comparable influence and even, in several cases, greater influence in forecasting stock price movement the day after an earnings call.

6.9 Yearly Breakdown

The final aspect of results to report is the yearly breakdown of average accuracies across earnings call samples for both Datasets seen in Figure 6.20. For both Datasets, the baseline shows some variation year to year. For Dataset Pro, consistent improvement over accuracy displayed serves as evidence that the findings of this thesis, and the assertion that linguistic analysis of earnings calls hold predictive power, are consistent over time, and do not represent

Year	Baseline	Average	Avg. Vs Baseline
2016	50.16%	53.33%	3.17%
2015	49.23%	53.30%	4.07%
2014	50.82%	53.69%	2.87%
2013	51.46%	53.84%	2.38%
2012	48.89%	53.59%	4.70%
2011	47.91%	54.99%	**7.08%**
2010	47.74%	52.99%	5.25%
2009	50.37%	54.79%	4.42%
2008	49.01%	53.09%	4.08%
2007	49.73%	56.68%	6.95%
2006	51.64%	54.47%	2.83%

(a) Dataset Pro

Year	Baseline	Average	Avg. Vs Baseline
2020	51.61%	48.71%	-2.90%
2019	57.14%	61.90%	4.76%
2018	48.81%	55.95%	7.14%
2017	55.95%	55.24%	-0.71%
2016	51.19%	55.71%	4.52%
2015	55.95%	53.10%	-2.85%
2014	54.32%	53.09%	-1.23%
2013	51.25%	51.00%	-0.25%
2012	41.55%	50.39%	8.84%
2011	44.44%	48.61%	4.17%
2010	41.67%	51.67%	**10.00%**

(b) Dataset Created

Figure 6.20: Earnings Call Year By Year Breakdown By Average Ensemble Accuracy

an anomaly.

Dataset Created shows a far greater range of outcomes for improvement over baseline accuracy with some years actually performing worse than the baseline. However, the years where there is under performance are at much smaller margins than those of the over performance years. This does not discount the findings of this thesis. The higher variation in the performance on Dataset Created could be due to its small sample size. In Dataset Created, there are roughly 80 earnings calls per year while in Dataset Pro, there are roughly 6,000. The consistent performance over the years with Dataset Pro carries more weight in concluding the predictive power of linguistic analysis exists year after year.

Chapter 7

Future Work

While the work of this thesis provides strong insights into the predictive power of earnings calls and the predictions of stock price movement, there are several avenues of future work that we have identified.

7.1 Classifiers and Scope

First is the use of neural networks instead of machine learning classifiers. Financial data, as stated multiple times in this paper, is incredibly noisy, and neural networks can be extremely effective at separating noise from signal. In addition, the research from Standard and Poor's detailed in the Related Work chapter used neural networks instead of machine learning classifiers and found promising results [13]. A neural network would be a logical next step in building off the research from this thesis. In a similar area, there is opportunity to tune the many parameters of the classifiers, both single and ensemble, that were researched in this paper. This thesis was more concerned with exploring predictive power than with producing the highest overall accuracy and as such, did not go through the formal process of parameter tuning. Parameter tuning of a classifier has been shown to make the classifier much more effective at the given task and may lead to overall higher accuracies than were achieved in this paper.

Furthermore, while this paper focused on the price movement the day immediately following an earnings call, there is an opportunity to expand the scope of prediction to dates such as one week, 30 days, or 60 days after a call. Increasing the prediction scope to be more long-term would indicate how long the predictive power of earnings calls last.

7.2 Linguistic Analysis

Turning to linguistic analyses, we have identified two possible directions for future work. The first is breaking down the earnings call into its individual pieces - prepared statements and question and answer - and extracting the same features from both. In this way, it could be studied if one section of the earnings call is more predictive than the other. Our early hypothesis is that the question and answer section would be more predictive as the Q&A section requires executives to answer questions organically and without a fully prepared script. As such, word choices in this section may reveal more information than carefully worded prepared statements. Following this avenue would provide insight to the robustness of this hypothesis.

Secondly, there is an interesting opportunity to extracted linguistic features not only from earnings call transcripts, but from the actual audio recordings of the call. While Natural Language Processing on audio is more complicated than textual linguistic analysis, spoken words hold information that is unobtainable through text such as verbal inflection, cadence, and pauses. Considering the predictive power of textual analyses of an earnings call, it is anticipated that an analysis of audio recordings will provide even further predictive power.

7.3 Real World Applications

Finally, there are a few different ways this thesis could be applied in a real world context. The first would be to change the way the results of this research are evaluated. Instead of comparing predictions to a baseline accuracy, it would be interesting to compare the predictions of our classifiers to predictions made by actual financial analysts. In this way, we would be able to see if machine learning models are better predictors of stock movement after earnings calls than humans.

Secondly, there is an opportunity to explore building a trading strategy around

earnings call events. For this strategy to be effective, the scope of prediction would most likely need to be increased as stated above as investing with the intention of making a return in one day is difficult. This was exemplified by all baseline accuracies for each experiment being around 50%. Nonetheless, when it is shown that there are indicators that have predictive power in forecasting stock price movement, there is phenomenal opportunity to trade off these factors, and linguistic analyses of earnings calls are no different.

Chapter 8

Conclusions

8.1 Contributions

This thesis makes several contributions to the continuation of research of this topic. Dataset Created, with all of its financial features and extracted linguistic analysis scores in addition to the raw and cleaned earnings calls, will be publicly available for scholarly research. The Dataset, in addition to the code used to clean the calls and extract linguistic scores is available at the GitHub link found below.

https://github.com/taylorbeckett/seniorthesis

8.2 Conclusions

Through our results chapter, we are able to reach several conclusions. First and foremost, through an analysis of results on samples the day after earnings call, in conjunction with an analysis of feature coefficients, we conclude that there is predictive power in a linguistic analysis of an earnings calls in forecasting the movement of the stock price the next day. This conclusion holds over time and through different cycles of the economy. We found that positive and negative sentiment scores of calls are more predictive than polarity or subjectivity, and sentiment scores overall are more predictive than complexity scores. However, SMOG Grade Index is a much better measure of complexity in relation to earnings calls and stock price predictions than the Flesch Reading Ease metric. We observed that, in terms of sentiment analysis, the Loughran-McDonald Financial Dictionary is indeed more effec-

tive the the General Inquirer Dictionary for financial linguistic analysis which highlights the importance of context when conducting Natural Language Processing.

In terms of models, Logistic Regressions are the best single classifier for this task and Bagging and Stacking are the best ensemble models. Generally, ensemble models are more suited to this task and deliver better results at both every day stock price movement prediction and prediction solely the day after earnings calls.

We confirmed several strongly held convictions around predicting stock prices. First is price history and the trends of stocks are strong indicators of future performance. Secondly, traditional metrics of measuring the success of earnings calls have significant power in predicting stock price movement. In particular, comparison of actual EPS to expected EPS holds a great deal of weight.

We reach a similar conclusion as other related works. Predicting stock price movement the day after earnings calls continues to be a difficult task as financial data remains incredibly noisy. However, there are predictive indicators in both financial and non-financial data.

Finally, this thesis helps frame a new way at looking at earnings conference calls and reveals new ways to extract predictive information from them that moves beyond the numbers. Indeed, when examining earnings calls and earnings reports, it is a good idea not to only "follow the money", but to also "follow the words".

Bibliography

[1] Will Kenton. *Regulation Fair Disclosure (Reg FD)*. Investopedia. Sept. 10, 2019. URL: https://www.investopedia.com/terms/r/regulationfd.asp (visited on 04/15/2021).

[2] McKay Price et al. "Earnings Conference Calls and Stock Returns: The Incremental Informativeness of Textual Tone (October 10, 2011)". In: *Journal of Banking and Finance* 36 (2012), pp. 992–1011.

[3] *Financial Analysts : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics*. U.S. Department of Labor Statistics. Mar. 9, 2021. URL: https://www.bls.gov/ooh/business-and-financial/financial-analysts.htm (visited on 04/15/2021).

[4] Brian Beers. *What Is an Earnings Conference Call?* Investopedia. May 24, 2018. URL: https://www.investopedia.com/small-business/what-is-an-earnings-conference-call/ (visited on 04/15/2021).

[5] Keith D. Foote. *A Brief History of Natural Language Processing (NLP)*. DATAVERSITY. May 22, 2019. URL: https://www.dataversity.net/a-brief-history-of-natural-language-processing-nlp/ (visited on 04/15/2021).

[6] Stephen Naoyuki Matsuba. "Finding the Range: Linguistic Analysis and Its Role in Computer-Assisted Literary Study". In: *Computers and the Humanities* 27.5/6 (1993), pp. 331–340.

[7] Mika V. Mäntylä, Daniel Graziotin, and Miikka Kuutila. "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers". In: *Computer Science Review* 27 (2018), pp. 16–32. URL: https://www.sciencedirect.com/science/article/pii/S1574013717300606.

[8] Tracy Mayor. *Why finance is deploying natural language processing*. MIT Sloan. Nov. 3, 2020. URL: https://mitsloan.mit.edu/ideas-made-to-matter/why-finance-deploying-natural-language-processing (visited on 04/15/2021).

[9] Gaurav Paruthi Gierad Laput and Gaurav Singhal. "G-Money: Earnings Calls Predict Stock Performance". University of Michigan. 2016.

[10] L. E. Solberg and Jørgen Karlsen. "The predictive power of earnings conference calls : predicting stock price movement with earnings call transcripts". In: 2018.

[11] Zhiqiang Ma et al. "Towards Earnings Call and Stock Price Movement". In: *arXiv:2009.01317 [cs, q-fin]* (Aug. 23, 2020). arXiv: 2009.01317. URL: http://arxiv.org/abs/2009.01317 (visited on 04/15/2021).

[12] Chaz Pratt Thomas Ulrich and Phillip Thun-Hohenstein. "Machine Learning Analysis of Company Earnings Releases". CS229, Stanford University. 2016.

[13] Yu Qin and Yi Yang. "What You Say and How You Say It Matters: Predicting Stock Volatility Using Verbal and Vocal Cues". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 390–401. DOI: 10.18653/v1/P19-1038. URL: https://www.aclweb.org/anthology/P19-1038.

[14] *Nasdaq, Inc. Common Stock (NDAQ) Historical Data*. URL: https://www.nasdaq.com/market-activity/stocks/ndaq/historical (visited on 04/15/2021).

[15] Wharton Research Data Services. *CRSP*. WRDS. URL: https://wrds-www.wharton.upenn.edu/pages/get-data/center-research-security-prices-crsp/ (visited on 04/15/2021).

[16] Mary Hall. *What Is a CUSIP Number?* Investopedia. June 29, 2019. URL: https://www.investopedia.com/ask/answers/what-is-a-cusip-number/ (visited on 04/15/2021).

[17] Wharton Research Data Services. *IBES*. WRDS. URL: https://wrds-www.wharton.upenn.edu/pages/get-data/ibes-thomson-reuters/ (visited on 04/15/2021).

[18] Diego Lopez Yse. *Your Guide to Natural Language Processing (NLP)*. Medium. Apr. 30, 2019. URL: https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1 (visited on 04/15/2021).

[19] *General Inquirer Home Page*. General Inquirer Home Page. URL: http://www.wjh.harvard.edu/~inquirer/ (visited on 04/15/2021).

[20] TIM LOUGHRAN and BILL MCDONALD. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks". In: *The Journal of Finance* 66.1 (2011), pp. 35–65. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/29789771.

[21] Rudolf Flesch. *Guide to Academic Writing Article - Management - University of Canterbury - New Zealand*. How to Write Plain English. July 12, 2016. URL: https://web.archive.org/web/20160712094308/http://www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml (visited on 04/15/2021).

[22] G. Harry Mc Laughlin. "SMOG Grading-a New Readability Formula". In: *Journal of Reading* 12.8 (1969), pp. 639–646. ISSN: 00224103. URL: http://www.jstor.org/stable/40011226.

[23] Dan Nelson. *Ensemble/Voting Classification in Python with Scikit-Learn*. Stack Abuse. URL: https://stackabuse.com/ensemble-voting-classification-in-python-with-scikit-learn/ (visited on 04/15/2021).

[24] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[25] *Natural Language Toolkit — NLTK 3.6 documentation*. Natural Language Toolkit. URL: https://www.nltk.org/ (visited on 04/15/2021).

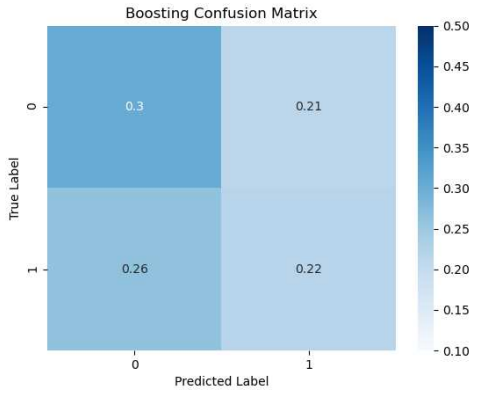
[26] Nick DeRobertis. *pysentiment2: Sentiment Analysis in Python using a Dictionary Approach*. Version 0.1.1. URL: https://github.com/nickderobertis/pysentiment (visited on 04/15/2021).

[27] Shivam Bansal Aggarwal Chaitanya. *textstat: Calculate statistical features from text*. Version 0.7.0. URL: https://github.com/shivam5992/textstat (visited on 04/15/2021).

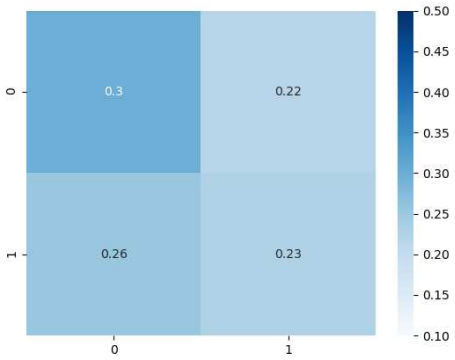
[28] Nagesh Singh Chauhan. *Model Evaluation Metrics in Machine Learning*. KDnuggets. URL: https://www.kdnuggets.com/model-evaluation-metrics-in-machine-learning.html/ (visited on 04/15/2021).

Appendix A

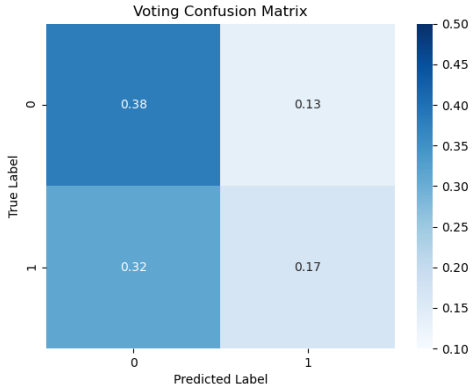
Additional Results



(a) Boosting Confusion Matrix

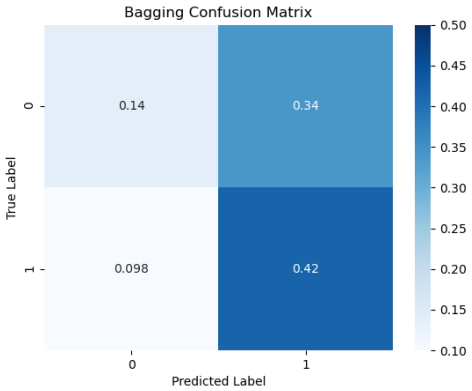


(b) S. Voting Confusion Matrix

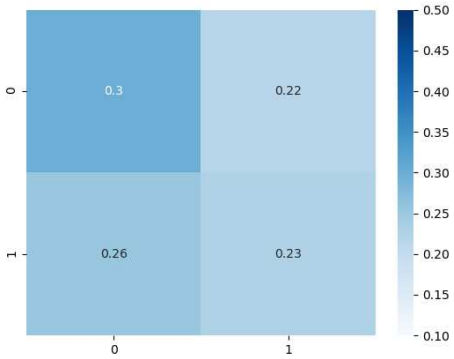


(c) H. Voting Confusion Matrix

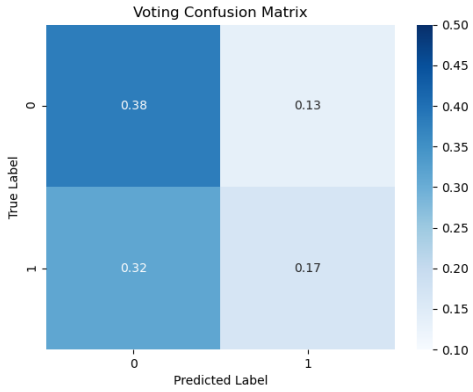
Figure A.1: Other Ensemble Classifiers Confusion Matrices – Dataset Pro



(a) Bagging Confusion Matrix



(b) S. Voting Confusion Matrix



(c) H. Voting Confusion Matrix

Figure A.2: Other Ensemble Classifiers Confusion Matrices – Dataset Created

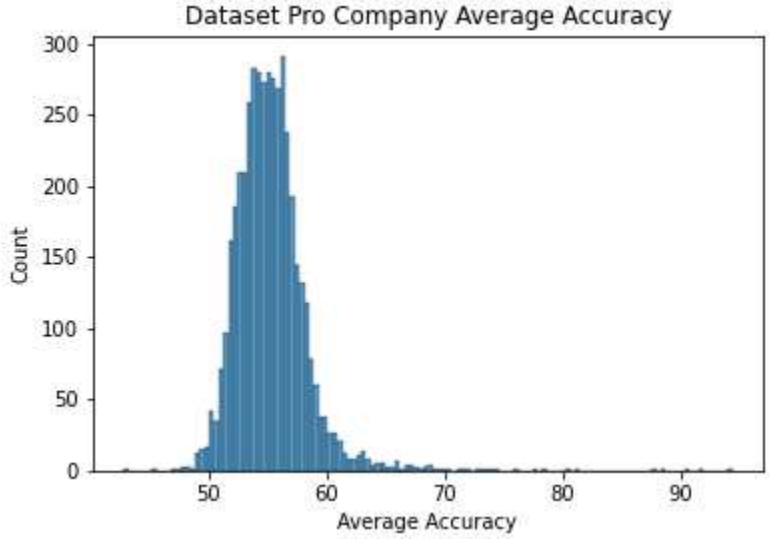
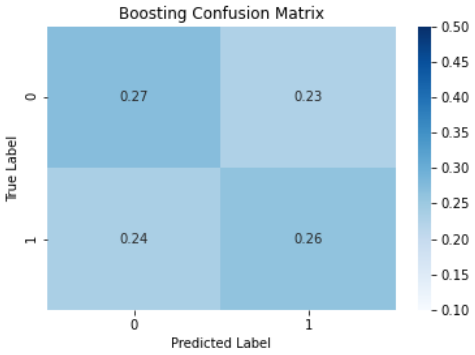
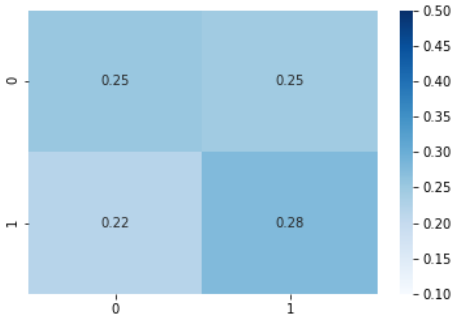


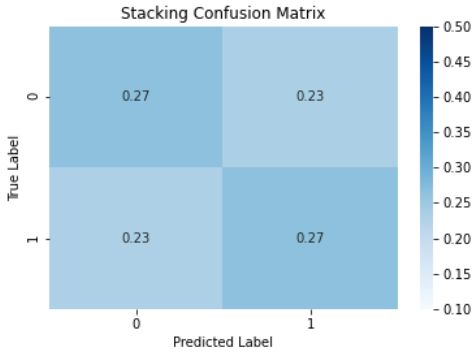
Figure A.3: Histogram of Company Accuracy Breakdown – Dataset Pro



(a) Boosting Confusion Matrix



(b) S. Voting Confusion Matrix

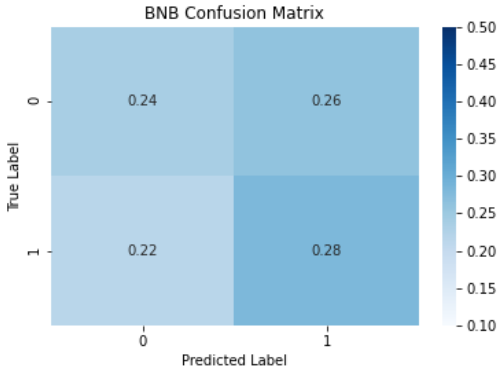


(c) Stacking Confusion Matrix

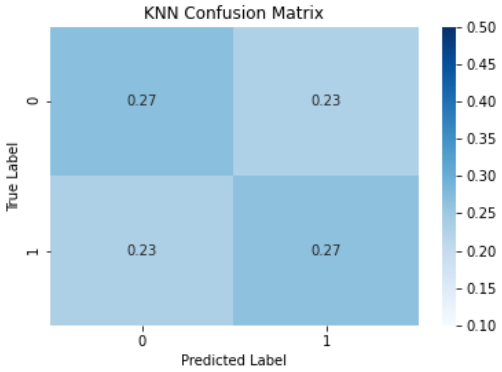
Figure A.4: Other Ensemble Classifiers Confusion Matrices – Dataset Pro Day After Earnings Call

Eval Metric	Baseline	KNN	SVC	LogReg	BNB	Averages
Accuracy	49.89%	53.73%	53.68%	**53.99%**	52.09%	53.37%
Versus Baseline	0.00%	3.84%	3.79%	**4.10%**	2.20%	3.48%
Precision		53.64%	53.74%	**53.89%**	51.82%	53.27%
Recall		53.52%	51.51%	54.02%	**56.65%**	53.92%
Specificity		53.94%	**55.85%**	53.97%	47.55%	52.83%
False Pos Rate		46.06%	**44.15%**	46.03%	52.45%	47.17%
False Neg Rate		46.48%	48.49%	45.98%	**43.35%**	46.08%
F1 Score		53.58%	52.60%	53.95%	**54.13%**	53.57%
Total Obser.		67702	67702	67702	67702	67702

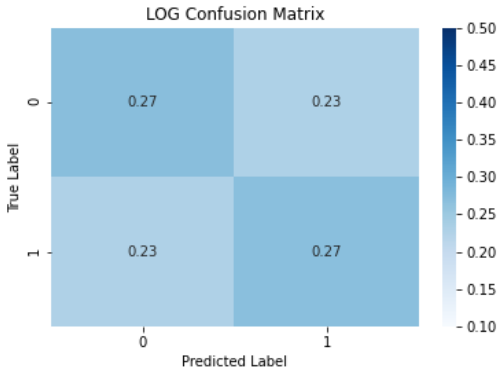
Figure A.5: Earnings Call Samples Single Classifiers – Dataset Pro



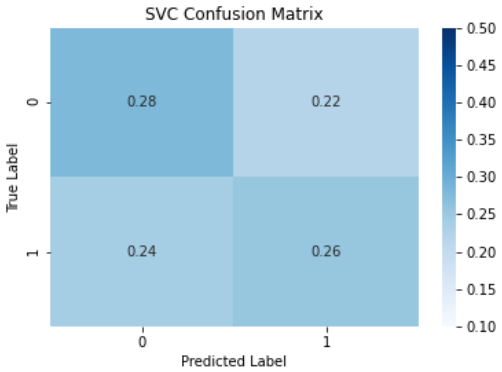
(a) BNB Confusion Matrix



(b) KNN Confusion Matrix

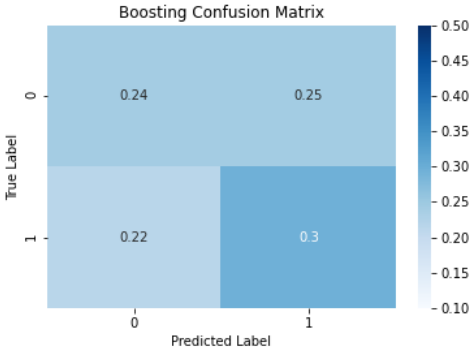


(c) LogReg Confusion Matrix

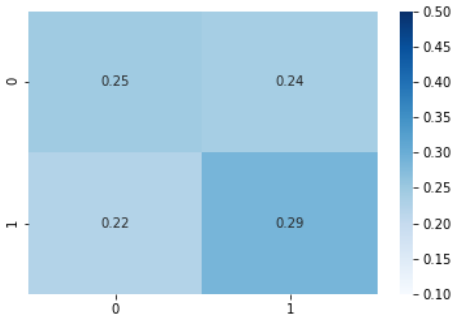


(d) SVC Confusion Matrix

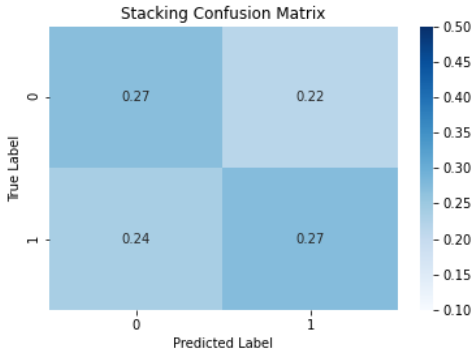
Figure A.6: Single Classifiers Confusion Matrices Earnings Call Samples – Dataset Pro



(a) Boosting Confusion Matrix



(b) S. Voting Confusion Matrix

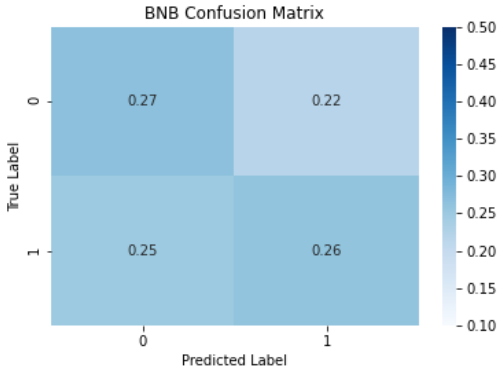


(c) Stacking Confusion Matrix

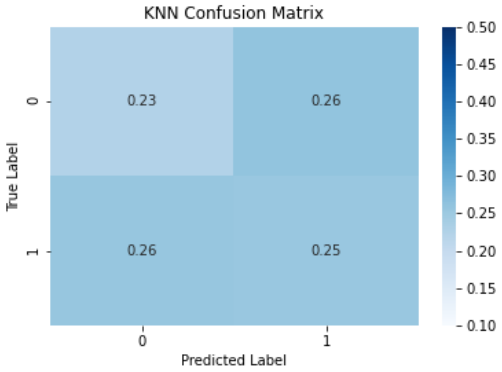
Figure A.7: Other Ensemble Classifiers Confusion Matrices – Dataset Created Day After Earnings Call

Eval Metric	Baseline	KNN	SVC	LogReg	BNB	Averages
Accuracy	51.25%	48.13%	51.99%	**55.35%**	52.99%	52.11%
Versus Baseline	0.00%	-3.12%	0.74%	**4.10%**	1.74%	0.86%
Precision		49.40%	52.73%	**56.42%**	54.40%	53.24%
Recall		49.76%	**60.92%**	56.55%	50.97%	54.55%
Specificity		46.43%	42.60%	54.08%	**55.10%**	49.55%
False Pos Rate		53.57%	57.40%	45.92%	**44.90%**	50.45%
False Neg Rate		50.24%	**39.08%**	43.45%	49.03%	45.45%
F1 Score		49.58%	**56.53%**	56.48%	52.63%	53.81%
Total Obser.		804	804	804	804	804

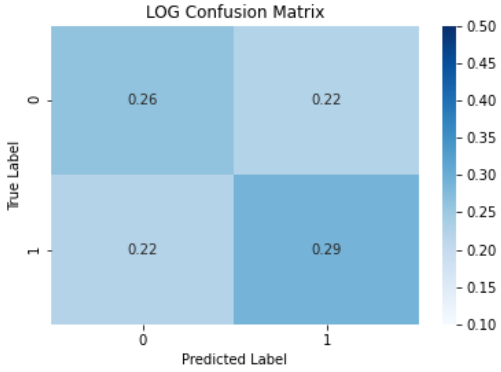
Figure A.8: Earnings Call Samples Single Classifiers – Dataset Created



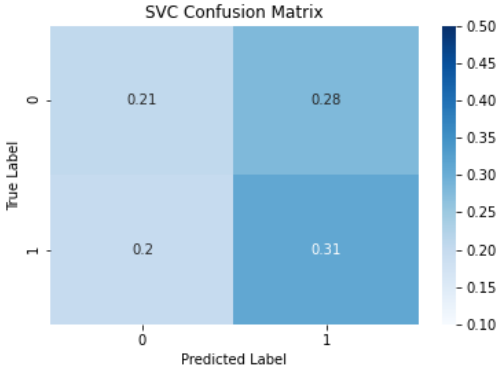
(a) BNB Confusion Matrix



(b) KNN Confusion Matrix



(c) LogReg Confusion Matrix



(d) SVC Confusion Matrix

Figure A.9: Single Classifiers Confusion Matrices Earnings Call Samples – Dataset Created

Year	Baseline	Average	Av. Vs Base
2016	46.79%	56.09%	9.30%
2015	47.63%	56.14%	8.51%
2014	48.72%	55.83%	7.11%
2013	50.87%	55.48%	4.61%
2012	48.61%	55.78%	7.17%
2011	48.01%	56.79%	8.78%
2010	49.47%	55.71%	6.24%
2009	49.62%	56.21%	6.59%
2008	45.56%	57.77%	**12.21%**
2007	48.03%	56.58%	8.55%
2006	48.68%	56.08%	7.40%
2005	47.76%	56.46%	8.70%

Figure A.10: Yearly Breakdown All Observations – Dataset Pro

Year	Basline	Average	Av. Vs Base
2020	50.95%	57.23%	**6.28%**
2019	53.91%	57.79%	3.88%
2018	50.25%	55.03%	4.78%
2017	53.91%	58.03%	4.12%
2016	52.91%	56.37%	3.46%
2015	49.13%	54.68%	5.55%
2014	52.90%	56.57%	3.67%
2013	53.44%	55.17%	1.73%
2012	50.40%	52.64%	2.24%
2011	49.52%	53.33%	3.81%
2010	51.76%	54.04%	2.28%

Figure A.11: Yearly Breakdown All Observations – Dataset Created