Long Term Company Growth Prediction using NLP Techniques on 10-K Financial Filings

Hou Yongzhuo[1], Yang Lixuan[1], Li Hongzhe[1], Wang Zhiliang[1], and Lu Yuliang[1]

[*]The Asian Institute of Digital Finance, National University of Singapore
[**]Final Project Report, FT5005

April 24th, 2022

Abstract

10-K financial filing sentiment analysis is always challenging due to the language processing and lack of labeled data during modeling. In this project, we innovatively combine 10-K filings text materials with stock-related statistics to develop a realistic forecast of long-term stock returns. The long-term stock return is measured by the long-term Cumulative Abnormal Return (CAR), which can be used as a reference for long-term company growth. Our final target variable is a binary variable, in which one class indicates the firm realizes a positive long-term CAR, and the other class indicates a negative one. We apply text analysis techniques for text processing and text sentiment analysis. Besides, we create more features based on relevant literature and statistical indicators. We try several basic machine learning models, such as XGBoost, Gradient Boosting, and LightGBM. We also implement several stacking methods to improve our model accuracy considerably. Our final model achieves 78% accuracy in the binary classification task.

1 Introduction

Accurately forecasting yields is critical for financial services, including individual investors, investment banks, corporate investment managers and hedge funds, etc., for their customers. It is equally important for investors to foresee accurately forecasting yields and design an investment or trading strategy that considers all relevant aspects of the forecast. For many years, equity market forecasting research has emphasized volatility models. Few studies have instilled the role of technological forecasting, i.e., Natural Language Processing (NLP). Nowadays, NLP has evolved into a powerful technology in many fields because of its ability to capture sentiments and feelings in the text. Many predicting applications have started adopting the NLP techniques to give their customers better experience [1].

In this project, we will be exploring what can be learned about forecasting the yield of firms based on their annual Form 10-K SEC filing, stock prices, and other financial indicators. The project is motivated by several considerations and potential applications. First of all, investors and firms expect high long-term growth in both academic and industrial aspects. Applications including robo-advisor by implemented valuation models and estimated cost of equity for firms will be demanding. Besides, in the industry, since the long-term growth rate is consistently cited as a critical justification for stock recommendations, the potential of building a platform for investors about stock recommendations is confirmed [2]. NLP, as mentioned above, can be one of the solutions that satisfy the above demands. The project is mainly built on the traditional NLP approach, including the bag of word approach, dictionary approach, and sentiment analysis. Advanced methods such as FinBert and state-of-the-art will also be reviewed and discussed as our future improvement.

The project aims to direct the investors on the likelihood of an upcoming status and long-term growth of a specific firm. Investors can use insights derived from this project to inform their trading decisions. Historically, a firm's stock prices will be related to its future status and long-term growth [3]. Therefore, a viable strategy will be buying shares of firms in the expansion stage and high long-term growth and selling shares of firms in the recession stage and low long-term growth. The project will estimate a given firm's status and long-term growth based on stock

prices and other financial indicators. Additionally, for investors with a larger landscape who may not be interested in a specific firm, the project will also provide insights into predicting industry-level long-term growth.

Long-term growth is an essential indicator for investors in the long-term profitability of their stock recommendations. Simon [4] publishment states that predicting through the process of long-term growth forecasts is of greater importance to understanding firms and their prospects. Besides, Simon [4] mentioned that for investors, long-term growth forecasts are helpful and informative but must be "used with caution," especially when experiencing abnormal returns, which will be discussed and analyzed further during Feature Engineering in the later section.

Our 10-K SEC filing dataset consists of roughly 90000 reports for filing years between 2005 and 2019, approximately 1.5 GB. The National University of Singapore provided these reports with the extraction of risk factors, the business sector, and the MD&A sector. This dataset is updated annually as 10-K filings are due 90 days after the fiscal year-end. Our second data set comes from yahoo finance, which consists of around 2300 firms between 2005 to 2019, approximately 2.9GB. For sentiment analysis, Loughran and McDonald (LM) dictionary and Harvard IV-4 dictionary are instrumented with an initial ratio of 9:1.

2 Existing Literature

Blomme et al. used FinBERT NLP methods to predict the effect of 10-K, 10-Q, and 8-K company reports on abnormal stock returns. Their research shows whether using the FinBERT language model in union with a dataset constructed out of 8-K, 10-K, and 10-Q filings published by S&P 500-listed companies extracted from the SEC's EDGAR database can predict abnormal stock returns in the near future after the publishing of a report. For instance, in the 8-K example, they created 32 features around the sentiment and financial data for analysis. They combined them with different machine learning methods to investigate the relationship between long-term returns and abnormal stocks. One conclusion states that after stacking a linear classifier on top of the pre-trained language model, recall scores for abnormal stock returns (with threshold ±1%) of 75%, 54%, and 77% were obtained for 8-K, 10-K, and 10-Q filings respectively. So it is indeed possible to predict abnormal stock returns using a domain-specific FinBERT language model combined with different machine learning models on 8-K, 10-K, and 10-Q filings of S&P 500 companies [5].

Jønsson et al. analyzed the text of over 29,000 10-K filings from 2010 to 2017 in a bull stock market and used a three-layered convolutional neural network to predict stock performance. The result shows that the model, on average and portfolio-wise, significantly predicts the company-specific stock performance, and the linguistic features of the 10-K filing have a predictive power of company-specific performance [6].

Masoud proposed an end-to-end attention-based classification model architecture that predicts the stock price movements based on the linguistic features extracted from 8-K documents. He used a dataset containing variable-length 8-K reports describing significant business events for all S&P 500 companies between 2002 and 2012 and constructed four different deep learning models for comparative research. The result shows the use of a rich mixture of quantitative financial data and stock-related text data to extract signals that would better capture the stock dynamics [7].

Bohn prepared over ten years' worth of stock data and proposed a solution that combines features from textual annual and quarterly filings with fundamental factors for long-term stock performance forecasting. He considered the data from approximately 1500 stocks that appeared in the S&P 500 between 2002 and 2016 and developed a new method of extracting features from the text for performance forecasting and applied feature selection aided by a novel evaluation function. The feature selection method he chose was to use a beam-search forward sequential selection algorithm and consists of iteratively expanding the feature set to be used with the next best option. By training predictive regression models using features from fundamental data and features extracted from filing text and comparing the results against baseline methods. Results show that adding text or data features through feature engineering can significantly improve some models [8].

3 Data Preparation

3.1 Data Collection

The purpose of our project is to utilize 10-K filings of various publicly traded companies to predict

Table 1: Literature performance metrics(All abbreviations are explained in appendix)

Ref.	Dataset	Technique	Prediction	Metrics	Results
[5]	8-K, 10-K, 10-Q, S&P500	LR+RF+DNN	Weekly	Recall	75%(8K, LR)
[6]	10-K, CRSP	CNN	Yearly	Accuracy	28.9%
[7]	8-K, S&P 500	LSTM, ATT-ERNN	Yearly	Accuracy	56.07%
[8]	S&P 500, Text data	LR+GB+RF+NN	Long term	Stdev	0.04829 (LR+ FS_all)
[9]	Word vectors	Bag of Words+DBN + RNN	Long term	Test error	39%
[10]	Historical events	NN (event embeddings) CNN	Weekly, Monthly	Accuracy& MCC	64.21%
[11]	10K,stock index	LR+RF+GB	Long term	Accuracy	72.5%(GB)

long-term company growth. There are different ways to measure the company growth, such as stock price growth, abnormal stock returns, earnings per share growth, income growth, etc. The choice of the proxy variable will lead to different prediction results. However, given the lack of enough income and financial statements data for the companies, we focus on predicting the long-term stock performance of the companies using the stock price data. We focus on two target variables: long-term moving-average stock price growth rates and long-term stock abnormal returns and checked which one is a better target variable. Besides, we need some other data to help us make better predictions, such as the SIC codes. Here are all the data and documents we used for the project.

- 10-K filings
- Stock and S&P 500 index prices
- Standard Industrial Classification (SIC) codes
- Harvard IV-4 and LM dictionary (used in the feature engineering)

To simplify the process of finding 10-K reports for each company, we choose to use the relevant documents provided by Prof.Huang, which were risk factors, item1-business, and item7-md&a. The first step is to merge all the three parts according to company tickers and the file report dates. In detail, we collect the three parts for 2275 companies. We only keep the data for 467 companies and drop others that lacked either essential text files or numerical data.

We collect the SIC codes for each company in our dataset. SIC codes are four-digit numerical codes that categorize the industries that companies belong to based on their business activities [12]. The SIC system classifies the companies into 11 major divisions using the first two digits of each SIC code, including manufacturing, services, mining, etc. We present the number of companies in each major division in the following Table 2. We can see that most companies in our dataset are in the manufacturing, finance, and services industry. At the same time, there are no companies in the agriculture, public administration, and non-classifiable establishments industry.

We collect stock prices for each listed company after its IPO using Yahoo Finance API. Zhang et al. presented the sliding window with the lookback, the gap, and the horizon window to construct stock portfolios [13]. We decide to use the same idea in building the long-term moving-average stock price growth rates and the long-term cumulative abnormal returns. The lookback window refers to the 10-K report year of each company. The horizon window refers to a period after the lookback window. Additionally, the gap is the differential periods between the lookback and horizon windows. We present the three windows used to construct our target variable in the Figure 1.

3.2 Target Variable Construction

Here is our procedure for constructing the long-term stock price growth rates. We set the gap window period to be one month and the horizon window to contain two years' stock prices data. The gap period allowed us to focus on the stock return change between sometime in the future and the report date instead of immediate stock price changes. We intentionally set the gap and horizon window periods short since we don't believe the contents of 10-K reports can have profound effects in the far future and for a long duration. The formula for growth rate g_s is

Table 2: Major Division Classification

Major Division	Count
Agriculture, Forestry, And Fishing	0
Mining	7
Construction	8
Manufacturing	216
Transportation, Communications, Electric, Gas, And Sanitary Services	29
Wholesale Trade	14
Retail Trade	26
Finance, Insurance, And Real Estate	90
Services	77
Public Administration	0
Non-classifiable Establishments	0

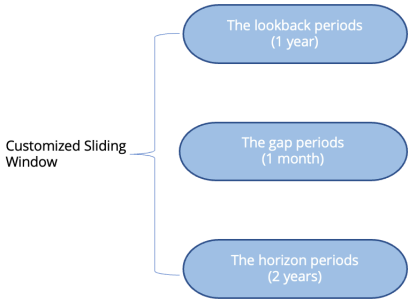


Figure 1: The lookback, gap, and horizon window.

shown as follows, where $E(p_{s,h})$ represents the average stock prices in the horizon window and $E(p_{s,l})$ represents the average stock prices in the lookback window. We split the calculated growth rates into two categories, 0 and 1, such that 0 indicates low growth rates and 1 indicates high ones.

$$g_s = \frac{E(p_{s,h})}{E(p_{s,l})} \tag{1}$$

For the abnormal returns, we mimic the short-term abnormal return in event studies and construct a customized long-term cumulative abnormal returns variable. The event was publishing 10-K reports, and we are interested in the cumulative abnormal returns following the event. We still apply the lookback, gap, and horizon window in this case, where we still set the gap window period to be one month and the horizon window period to be two years. Another reason why we set the gap window period to be one month is that we believe the public may show extreme overestimation or underestimation of stock for roughly one month after publishing the 10-K report. The overreaction towards a stock can cause bias. We then calculate the cumulative abnormal returns over the horizon window period.

In calculating long-run abnormal returns, we require a benchmark to measure the market return, the S&P 500 index in our case. We apply the capital asset pricing model (CAPM) to estimate each firm's alpha and beta using stock price and S&P 500 index data during the lookback window period. The estimated alpha and beta are used to calculate estimated individual stock prices during the horizon window period. We calculate the cumulative abnormal returns by summing up the difference between the actual and estimated stock prices. We split the calculated cumulative abnormal returns into two groups, 0 and 1, such that 0 represented negative cumulative abnormal returns while 1 represented the positive return. In summary, we have 63 % and 37% class 1 and class 0 data, respectively. Here is the mathematical formula:

$$r_{i,t} = \alpha + \beta r_{m,t} + \epsilon_{i,t} \tag{2}$$

$$\hat{r_{i,t}} = \hat{\alpha} + \hat{\beta} r_{m,t} \tag{3}$$

$$\hat{ar_{i,t}} = r_{i,t} - \hat{r_{i,t}} \tag{4}$$

$$\hat{car_i} = \sum_{i=1}^{h} \hat{ar_{i,t}} \tag{5}$$

In the formula above, $r_{i,t}$ is the actual stock prices for stock i at time t. $r_{m,t}$ is the market prices at time t, which is the S&P 500 index returns in this case. α is the firm's recent performance track record. β is the sensitivity of individual stock to the general market

Table 3: Difference between Raw Data and Processed Data

Variable Name	Raw Data Summary	Processed Data Summary
Sample SIC Code	3317	[0, 0, 1, 0, 0, 0, 0, 0]
Sample Risk Factor Texts	A downturn in government spending related to public water transmission projects would adversely affect our business.	['downturn', 'govern', 'spend', 'relat', 'public', 'transmiss', 'project', 'advers', 'affect', 'busi']
Sample Item1 Texts	Item 1. Business We are a leading North American manufacturer of large-diameter, high-pressure steel pipeline systems for use in water infrastructure applications.	['item', 'busi', 'lead', 'manufactur', 'large-diamet', 'highpressur', 'pipelin', 'system', 'use', 'infrastructur', 'applic']
Sample Item7 Texts	Table of Contents Item 7. Management s Discussion and Analysis of Financial Condition and Results of Operations Forward-Looking Statements	['tabl', 'content', 'item', 'manag', 's', 'discuss', 'analysi', 'financi', 'condit', 'result', 'oper', 'forwardlook', 'statement']

movements. $\hat{r_{i,t}}$ is the predicted stock prices for stock i at time t. $\hat{\alpha}$ and $\hat{\beta}$ are the estimated parameters. $\hat{ar_{i,t}}$ is the calculated abnormal returns for stock i at time t. $\hat{car_i}$ is the calculated cumulative abnormal returns for stock i during the horizon window period. h is the number of days in the horizon window period.

Finally, We proceed to build machine learning models using binary cumulative abnormal returns as our target variable. We use it because of support from academic researches, such as Chahine(2004) conducted the long-run abnormal return research following IPOs, and comparisons we made between the two candidate target variables [14]. For the comparisons, we tried some baseline models such that these models also favored the use of abnormal returns as our "Y".

3.3 Data Processing

In our project, our raw data are complicated text files and eleven SIC code categories. Therefore, we focus on processing the raw texts to perform text analysis on them. Text processing consists of two major steps: text cleaning and text tokenization. We applied the one-hot encoding to convert the categorical SIC code to one-hot columns for the SIC codes. We delete some SIC code categories since these categories don't contain any data in our dataset.

In the text cleaning step, we need to remove extra or unnecessary patterns in the text files. These patterns include different punctuation, leading or trailing spaces, duplicate letters, numbers, special characters, etc. Besides, we need to normalize these text files to be lowercase. We apply the regular expression to perform these actions.

In the text tokenization step, we need to remove stopwords, which are common words in English, such as and, or, on, etc. Removing these unnecessary words can give us more focus on other meaningful words. Text tokenization is a way of separating a piece of text into smaller units. Instead of tokenizing texts by hand, we utilize the advanced Harvard IV-4 dictionary and LM dictionaries. The dictionaries provide functions to tokenize and stem the texts. These tokens help us in building advanced features based on the texts. We will use these new features in our machine learning models to predict long-term stock performance. We present a table showing the difference between our raw and processed datasets in Table 3.

4 Feature Engineering

4.1 Feature Construction

4.1.1 Volatility Benchmark

It is well known that the prediction of stock market volatility has always been an essential topic of research in finance, and stock market volatility is essentially a concentrated expression of stock price fluctuations [15]. Therefore, based on the practical task of predicting long-term stock returns in this project, we create a benchmark that can reflect the stock market's volatility at different times. In other words, if different stocks are in the same stock market at the same point in time, then their corresponding stock market volatility benchmarks should be the same.

Table 4: Volatility Benchmark Construction

Time	CL=F	VIX	SP500	Volatility Benchmark
2005	0.286447	0.07971	0.114262	0.031378
2006	0.444805	0.079597	0.159729	0.062822
2007	0.546119	0.298934	0.233175	0.147784
2008	1	1	0.120329	0.180493
2009	0.37467	0.947119	1.00E-06	9.91E-07
2010	0.666244	0.531263	0.084579	0.075963
2011	0.923109	0.606933	0.140848	0.161627
2012	0.908416	0.311129	0.190078	0.173857
2013	0.971571	0.146314	0.306193	0.256716
2014	0.887109	0.142945	0.433111	0.334596
2015	0.15503	0.259429	0.490384	0.152433
2016	0.066005	0.220052	0.505094	0.108364
2017	0.189423	1.00E-06	0.661314	0.093951
2018	0.42313	0.257376	0.792209	0.404328
2019	0.291278	0.201151	0.865131	0.319512
2020	1.00E-06	0.839438	1	0.629579

Christiansen et al. [16] find that information about economic variables can help predict future volatility. Paye [17] theoretically derives from formulas that economic indicators (such as national debt spreads and default gains) will affect stock return volatility. Based on the two-component GARCH-MIDAS model results, Conrad et al. [18] believe that macroeconomic variables are one of the crucial determinants of long-term stock market volatility.

According to the above studies on stock market volatility prediction, we are inspired by the combination method proposed by Dai et al. [19]. They utilize a "kitchen sink" combination approach, combining two essential indicators (WTI crude oil futures "CL=F" and CBOE volatility index "VIX") to construct an equation with the benchmark stock index "S&P 500" over the same period. The formula is shown below:

$$SP\ 500_{t+1} = \sum_{i=0}^{p-1} \alpha_i * SP\ 500_{t-i} + \beta * CLF_t + \lambda * VIX_t + \epsilon_{i+1} \tag{6}$$

VIX reflects the market's expectation of future volatility over the remaining term of the option, which means that it is a measure of market uncertainty. CL=F represents the futures price of WTI crude oil, one of the core resources of modern industry, whose changes can indirectly lead to changes in stock prices by affecting actual economic activities. In addition, the S&P 500, a benchmark index of U.S. large-cap stocks, is used to measure the performance of U.S. stocks.

In the approach proposed by Dai [19], the combination of VIX and CL=F is used to construct a specific equation relationship between S&P 500 and the rest of the indicators. Therefore, similarly, we try to create the stock market volatility benchmark by finding the statistical distance between the two sides of the equation, i.e., using the S&P 500 as the denominator and the combination of VIX and CL=F as the numerator. The specific formula is shown below:

$$Volatility\ Benchmark = \frac{\alpha * VIX_t + \beta * CLF_t}{\gamma * SP\ 500_t} \tag{7}$$

$$\gamma = 1 - \alpha - \beta \tag{8}$$

As shown in Table 4, we get the 16-year data from 2005 to 2020 through the yahoo finance API and calculate the volatility benchmark year by year. It is worth mentioning that alpha, beta, and gamma are all adjustable parameters in the volatility benchmark formula we have constructed. It is set to calculate the loss based on the gradient descent direction when applying the neural network techniques for the backward propagation in the future.

To a large extent, major historical events tend to affect stock performance (e.g., the COVID-19,

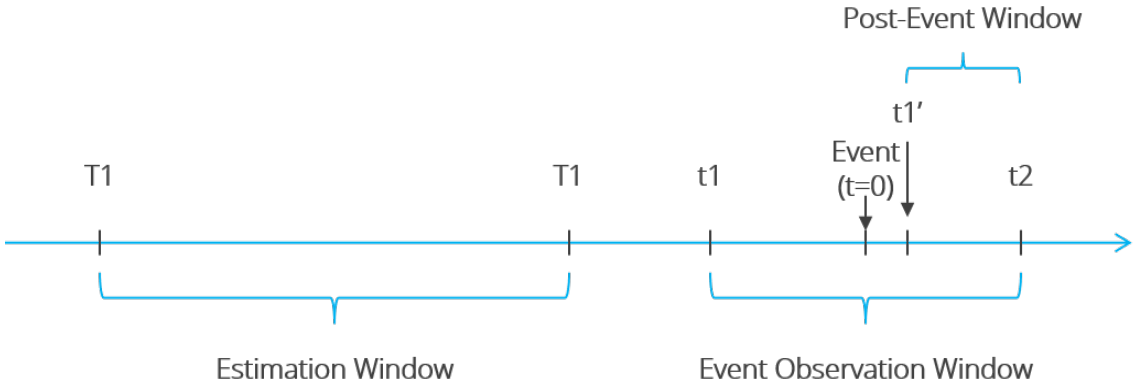


Figure 2: Event Study for Cumulative Abnormal Return Calculation

the Russia-Ukraine war, etc.), so we need to combine these force majeure events with the market predictability. It is also why we construct this volatility benchmark.

4.1.2 Market Reaction

According to the efficient market hypothesis [20], investors make quick and efficient use of possible information when buying and selling stocks, which means that all factors known to influence the price of a stock are already reflected in the price of the stock [21]. Nevertheless, investors' investment behavior is not always rational, and some investors may overreact or not react to the news announcement. Based on this theory and the wisdom of crowds [22], we take the same approach as we did in constructing the target variable for this project (i.e., event study) to construct a measure of the market's reaction to a company's 10-K reports before and after the announcement.

Since the calculation of cumulative abnormal returns is described in detail in the introduction to the construction of the target variable in this report, it is not repeated here. Briefly, as shown in Figure 2, we set the estimation period to 252 (i.e., [T1,t2] represents a full year of trading days), define the announcement date of the 10-K report as t = 0, and set the event observation window and post-event window to [-20,20] and [2,20], respectively. Then, we calculate the cumulative abnormal return corresponding to 11 different time windows ([-1,+1], [0,+1], [-1], [0], [1], [2,5], [2,10], [2,20], [-5,-2], [-10,-2], [-20,-2]). Since the event study is based on a single 10-K report announcement event for a single stock, it is impossible to conduct the independent sample T-test to determine whether the stock market response is in the confidence interval. Instead, the number of positive and negative cumulative abnormal returns under different time windows will be counted. In other words, a positive cumulative abnormal return for stock in the specific time window of a 10-K report announcement means that the market has a good expectation for the stock at that time, and a negative one means the opposite. To some extent, the characteristics we want to construct are intended to measure how the stock market reacts to different companies' 10-K reports.

4.1.3 Dictionary-based Feature

Since 2007, there has been a proliferation of financial and accounting literature studying different textual data types [23]. A developed approach to measuring sentiment at this stage is the bag-of-words approach: counting those positive/negative words [24] that are specified in the dictionary of finance and accounting terms, corresponding to the LM dictionary [25] developed by Loughran and McDonald. Therefore, based on this dictionary, we introduce the Harvard-IV dictionary [26] commonly used for general measures of sentiment tasks to construct a set of features capable of measuring the sentiment tone of companies' 10-K reports [27].

It is worth mentioning that we use a combination of the finance-specific LM dictionary and the general-purpose Harvard-IV dictionary, taking into account the public-facing property of 10-K reports. Initially, we assign a larger weight to the LM dictionary (compared to Harvard-IV) to properly analyze the sentiment tone related to financial nouns in 10-K reports without losing regular sentiment expressions. The following is the formula we use to calculate the

Table 5: Statistical Indicators Based on the Financial Technical Analysis

Category	Indicator	Description
Volume	Accumulation/Distribution Index (ADI)	Cumulative indicator that uses volume and price to assess whether a stock is being accumulated or distributed.
	Money Flow Index (MFI)	Technical oscillator that uses price and volume data for identifying overbought or oversold signals in an asset.
Volatility	Average True Range (ATR)	Technical analysis indicator that measures market volatility by decomposing the entire range of an asset price for that period.
Trend	Simple Moving Average (SMA)	It calculates the average of a selected range of prices, usually closing prices, by the number of periods in that range.
	Moving Average Convergence Divergence (MACD)	Trend-following momentum indicator that shows the relationship between two moving averages of a security's price.
Momentum	Relative Strength Index (RSI)	Momentum oscillator that measures the speed and change of price movements.
	Rate of Change (ROC)	Momentum-based technical indicator that measures the percentage change in price between the current price and the price a certain number of periods ago.
	Percentage Price Oscillator (PPO)	Technical momentum indicator that shows the relationship between two moving averages in percentage terms.
Stock Return	Daily Log Return (DLR)	Return in logarithmic calculation that assumes a continuously compounding rate of return.
	Cumulative Return (CR)	The total change in the investment price over a set time.

weights of the two dictionaries in this feature construction. The initial value of alpha is 0.9, while the value of beta is 0.1. Similar to the previous volatility benchmark calculation, the alpha and beta used here can be optimized in the backward propagation of the neural networks as the gradient decreases while calculating the loss.

$$Sentiment = \alpha * LM + \beta * HarvardIV \quad (9)$$

$$\beta = 1 - \alpha \quad (10)$$

Specifically, we use the pysentiment library (a dictionary framework for sentiment analysis) [28] written by Han to conduct data cleaning, tokenization, stemming, and lemmatization for different items in 10-K reports. It is expected to obtain the number of corresponding texts in the positive/negative dictionary sets [29]. And then, the polarity and subjectivity of the texts will be calculated. As a comprehensive report on the financial performance and operations of a listed company, the 10-K financial report contains many objective and subjective statements that should be submitted annually. The two tasks of polarity analysis and subjectivity evaluation are crucial in sentiment analysis [30]. Therefore, we use the same way as the Lydia system to calculate the polarity and subjectivity of the items in 10-k reports [31]. The formulas are shown as follows:

$$Polarity = (Pos - Neg)/(Pos + Neg) \quad (11)$$

$$Subjectivity = (Pos + Neg)/count(\star) \quad (12)$$

4.1.4 Industrial Classification

According to information on the SEC's website, SIC codes are assigned based on common characteristics shared across the company's products, services,

production, and delivery systems [32]. A company's SIC code is determined by the industry in which the company's largest product line is located. Therefore, for company 10-K reports, we can obtain the accurate classification of 10-K reports by analyzing their corresponding SIC codes. To ensure that the data set is not too sparse while considering the number of other features, we select only the first two digits of SIC codes used to obtain a broad-coverage industrial classification. The specific classifications are shown in Table 2 above.

4.1.5 Statistical Indicator

As one of the common tools of stock market investment, technical analysis is the study of historical market data (containing stock price and quantity) [33]. Its purpose is to use insights from market psychology, behavioral economics, and quantitative analysis to predict potential future market behavior using visual or statistical techniques in the form of past stock market performance. From the main mission of this project, we believe that in addition to the set of features obtained based on 10-K financial reports, statistical indicators that can reflect stock market trends are indispensable for model training. Therefore, we use the open-source financial technical analysis library to conduct the technical analysis for five categories of indicators [34], namely volume, volatility, trend, momentum, and stock return. After that, we select a series of representative indicators as an essential part of our feature construction. The details are shown in Table 5.

4.2 Feature Selection

Feature selection is the process of removing redundant features and finding the best set of features to build applicable models of studied phenomena [35]. The feature selection techniques can be roughly classified into two types, filter and wrapper methods. The filter method uses statistical techniques to evaluate the relationship between independent and dependent variables, which can output scores as metrics for choosing valuable features. The wrapper method requires us to create many models with different subsets of features and see which combination of features can result in better-performing models. Our project mainly uses mutual information and correlation coefficient techniques in the filter method for conducting the feature selection.

Correlation is a measure of the linear relationship between variables. In building machine learning models, we expect the independent variables to correlate with the target variables but not among themselves. Using the correlation coefficient technique allows us to identify highly correlated pairs of independent variables and drop unnecessary ones. We present variable pairs with correlation higher than 0.9 in Table 6.

Mutual information measures the dependence between independent and target variables. In detail, it estimates how much information the presence/absence of a feature contributes to making the correct prediction on "Y". We share the mutual information score in the Figure 3. The Figure 3 shows the great importance of those newly created features, including volatility, some SIC one-hot categorical variables, and some statistical indicators, for predicting the target variable. It also shows the usefulness of our feature engineering.

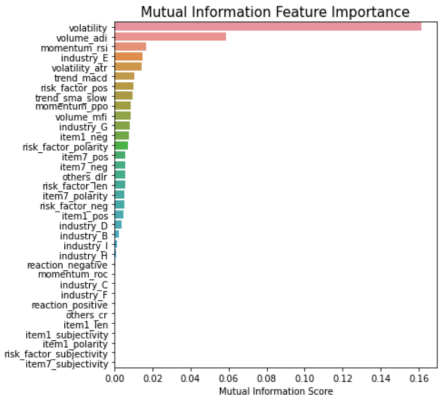


Figure 3: Mutual information scores for different features.

5 Methods

5.1 Base-Models

For our dataset, we try to build a stacking model and tune parameters by grid search. But, by observing the dataset, we find that the dataset is imbalanced. There are 2007 in class 0 and 3388 in class 1, so we use the SMOTE method to balance it after standardizing the dataset. The idea of the SMOTE algorithm is to synthesize new minority class samples. The synthesis strategy is to randomly select a sample B from its nearest neighbors for each minority class sample A and then randomly select a point on the line between A and B as a newly synthesized minority class sample.

Table 6: Correlated Independent Variable Pairs.

feature1	feature2	correlation
item7_pos	item1_len	0.988
item7_pos	item7_neg	0.933
risk_factor_pos	risk_factor_len	0.985
risk_factor_pos	risk_factor_neg	0.932
reaction_positive	reaction_negative	0.985
trend_sma_slow	volatility_atr	0.967
momentum_ppo	others_dlr	0.932

After that, we try to build our single model and tune parameters for five classifiers in level zero, they are XGBClassifier, LGBMClassifier, RandomForestClassifier, DecisionTreeClassifier, GradientBoostingClassifier.

For example, In XGBClassifier, we mainly tune 8 parameters: 'n_estimators', 'max_depth', 'min_child_weight', 'gamma', 'subsample', 'colsample_bytree', 'reg_alpha' and 'eta'. Specifically, 'n_estimators' is the number of runs the model will try to learn, and 'max_depth' specifies the maximum depth of each tree. These two parameters control how close the model is to the data. Other parameters are limited by the degree of model conservatism. For example, 'min_child_weight' is the minimum sum of instance weight (hessian) needed in a child. 'gamma' is the minimum loss reduction required to make a further partition on a leaf node of the tree. 'subsample' is the subsample ratio of the training instances. 'colsample_bytree' is the subsample ratio of columns when constructing each tree. 'reg_alpha' is L1 regularization term on weights. 'eta' is the step size shrinkage used in the update to prevents overfitting. The parameters of other models are almost the same, and we list the specific parameters in Table 7.

5.2 Stacking

For the stacking model, we try various methods. The first is to use three different stacking packages for comparison. The second is to implement the stack without external packages and manually construct the meta-models input, including the training and test sets. Third, we also build a three-layer stack to observe further whether the performance can improve.

In addition, we make the selection of base models. We first choose to keep all base models. In theory, a classifier with poor performance may lower the performance of the entire stacking model. Therefore, we filter out the classifier with poor performance later and observe the accuracy change. Furthermore, we also select and compare the meta-level classifiers to obtain a better model.

- Selection of Base-Models

  We consider that the poor performance of the classifier will drag down the entire stacking model, so we eliminated Decision Tree in the second attempt. Here are two selections:

  - XGBoost; LightGBM; RandomForest; Decision Tree; Gradient Boosting Tree
  - XGBoost; LightGBM; RandomForest; Gradient Boosting Tree

- Selection of Meta-Models

  The meta model is usually simple and provides a smooth explanation for the predictions of the base model. Here are two selections:

  - Logistic Regression
  - SVC

- Meta-Models tuning

  When meta-model is SVC, we shoud carefully tune the model to get a better performance. Here are the parameters:

kernel	'rbf'
C	[100, 10, 1, 0.1, 0.01]
gamma	'scale'

Then, we can combine these choices to get four different stacking models.

- model1
  - XGBoost; LightGBM; RandomForest; Decision Tree; Gradient Boosting Tree
  - Logistic Regression
- model2
  - XGBoost; LightGBM; RandomForest; Decision Tree; Gradient Boosting Tree

Table 7: Parameter tunning

classifiers	parameters	range	best value
XGBClassifier	n_estimators	Range(100,280,20)	260
	max_depth	Range(3,13)	11
	min_child_weight	Range(1,10,2)	1
	gamma	[i/10.0 for i in range(0,5)]	0
	subsample	[i/10 for i in range(6,11)]	1
	colsample_bytree	[i/10 for i in range(6,11)]	1
	reg_alpha	[0,0.01,0.015,0.02]	0
	eta	[0.1,0.15,0.2,0.25]	0.1
LGBMClassifier	n_estimators	[i*20 for i in range(5,11)]	120
	max_depth	[i for i in range(6,10)]	7
	min_child_weight	Range(1,10,2)	1
	reg_alpha	[0,0.05,0.1]	0
	learning_rate	[0.05,0.1,0.15]	0.1
RandomForestClassifier	n_estimators	Range(100,280,20)	200
	max_depth	Range(3,13)	12
	max_features	[3,4,5,6]	5
	min_samples_leaf	[3, 4, 5]	4
	min_samples_split	[2,3,4,5,6]	2
DecisionTreeClassifier	criterion	['gini', 'entropy']	gini
	max_depth	Range(3,13)	12
	splitter	['best', 'random']	random
	min_samples_leaf	[5, 10, 20, 50, 100]	10
GradientBoostingClassifier	n_estimators	Range(100,280,20)	260
	max_depth	Range(3,13)	12
	learning_rate	[0.15,0.2,0.25]	Default
	min_samples_split	np.linspace(0.1, 1.0, 10, endpoint=True)	Default
	min_samples_leaf	np.linspace(0.1, 0.5, 5, endpoint=True)	Default
	max_features	list(range(1,35,3))	Default

  - SVC

- model3
  - XGBoost; LightGBM; RandomForest; Gradient Boosting Tree
  - Logistic Regression
- model4
  - XGBoost; LightGBM; RandomForest; Gradient Boosting Tree
  - SVC

5.2.1 Traditional Stacking Model

We try 3 different packages for training and tune the parameters when we use SVC for the meta-model.

- sklearn.ensemble
- mlxtend
- vecstack

5.2.2 Customized Stacking Model

Without resorting to other ready-to-use stacking model packages, we manually construct the input that needs to be passed into the meta-model.

With the 'KFold' function, the training set is divided into 'n_folds' parts, and 'n_folds' training is performed. A part of the original training set will participate in each training round, and a prediction result of each part of the test set will be obtained, which will be merged later to form the new training set for the next layer. The original test set will be predicted probability simultaneously to get another prediction result. In the classification problem, we get the probability of all the categories. Due to predicting 'n_folds' times, we should finally take the average.

Combine the prediction results obtained from each among the five algorithms to train as a new training set and a new test set. The five algorithms are just

the five base models. The y_train or y_test is still the label of the original training set and test set. Finally, we can obtain the final prediction result after stacking. We can also extend this method to a multi-layer stacking model.

5.2.3 Multi-layer Stacking Model

In the third method, we try to build 3 layers to see if we can get better results. We remove the Decision Tree from base models and only use the remaining four classifiers as base models. The models in the second layer are set to the Decision Tree and SVC. After tuning the second layer, we input the new features into the meta-model, which is the Logistic Regression Model.

5.3 Comparison

We compare base models and then compare the best base model with the result of several stacking models to observe if stacking can reach higher accuracy. We do the same thing without feature engineering to explore if feature engineering is advantageous. In addition, we try to remove one of the base models, which has worse accuracy, to see if there is a better performance. Finally, we construct a 3-layer stacking model and compare it with the best model before.

5.4 Ablation Study

We test the effectiveness of our features through an ablation study using our stacking model. After deleting feature based on literature view("volatility"), dictionary approach("reaction_positive", "reaction_negative") and text classification approach("industry_B", "industry_C", "industry_D", "industry_E", "industry_F", "industry_G", "industry_H", "industry_I"), our best stacking model dropped by 0.03,0.03 and 0.02 on accuracy, which means these three new features all have relatively contribution to our results. By adjusting the length of time from one month to one year and generating a new dataset for ablation experiments, it is found that three features contribute more significantly, proving that our model has good generalization ability.

6 Result

The purpose of our project is to utilize 10-K reports of various publicly traded companies to predict long term company growth and we will summarize our report from three perspectives:

6.1 Performance Metrics

We use 'accuracy score' to compare different models. By comparative study of our models, we find the following conclusions:

- In the base models, the best one is Gradient Boosting Tree, which can reach 76.55% accuracy. The worst one is the Decision Tree, which has only 60.40% accuracy. Therefore, we remove the Decision Tree from our base models for later comparisons.
- We try many stacking models using different methods. The classification accuracy of the stacking model is significantly greater than that of the single classifier. We present the critical statistics in Table 9.
- We also compare the performance with that without feature engineering. The result shows that it will drop 4%-5% without feature engineering. We present the key statistics in Table 10.
- After removing weaker classifiers(Decision Tree) from base models, we can get a better model after tuning. We present the important statistics in Table 11.
- We try to use Decision Tree and SVC as the second layer and Logistic Regression as the meta-model. The accuracy will drop significantly if we use a three-layer stacking model. We also provide the important statistics in Table 11.

The base model accuracy are shown in the following Table 8.

6.2 Dataset Construction

Our datasets are from 10-K reports(1.5 GB), stock, S&P 500 index prices(2.9GB), and Standard Industrial Classification (SIC) codes ranging from 2005 to 2019. After concatenating all reports by ticker and date, we filter out data for 467 companies and drop about 1808 companies that lack essential text files or numerical data.

We build our "Y" (long-term stock growth) using long-term cumulative abnormal returns in event studies and construct features based on dictionary approach, text classification approach, and literature review.

Finally, we get our original dataset: 5395 rows with 39 columns. Because our dataset is imbalanced,

Table 8: Accuracy of base-models

Model	Accuracy
XGBoost	75.22%
LightGBM	73.67%
RandomForest	70.21%
Decision Tree	60.40%
Gradient Boosting Tree	**76.55**%

Table 9: Accuracy of Stacking Model

	Number of Models	Logistic Regression(%)	SVC(%)
Traditional Stacking Model			
sklearn.ensemble	5 + 1	77.29	77.06
mlxtend	5 + 1	77.21	77.14
vecstack	5 + 1	76.99	77.14
Customized Stacking Model			
KFold Cross-validation	5 + 1	**78.02**	**77.21**

Table 10: Accuracy Comparison (with/without feature engineering)

	LR(%) (Table 9)	SVC(%) (Table 9)	LR(%) (without feature engineering)	SVC(%) (without feature engineering)
Traditional Stacking Model				
sklearn.ensemble	**77.29**	77.06	73.45	73.89
mlxtend	**77.21**	77.14	72.71	72.20
vecstack	76.99	**77.14**	72.94	72.64
Customized Stacking Model				
KFold Cross-validation	**78.02**	77.21	73.53	74.56

Table 11: Accuracy of Stacking Model(w/o Decision Tree)

	Number of Models	Logistic Regression(%)	SVC(%)
Traditional Stacking Model			
sklearn.ensemble	4 + 1	77.29	76.92
mlxtend	4 + 1	76.84	76.99
vecstack	4 + 1	77.43	**77.36**
Customized Stacking Model			
KFold Cross-validation	4 + 1	**78.17**	77.21
Multi-layer Stacking Model	4 + 2 + 1	68.58	

we use the SMOTE technique and split the training dataset and test dataset with ratios of 0.8 and 0.2, respectively. To ensure accuracy, we also use stratified cross-validation during training.

6.3 Feature Importance

After measuring our model performance, it is also essential to understand how the features in our model contribute to the prediction. One useful metric to quantify the contributions is the feature importance. Generally speaking, there are two measures, model-specific and model-agnostic (permutation) feature importance measures.

For the model-specific measure, it shows the level of importance or relevance for using the specific feature in the prediction. For the model-agnostic feature importance, it is defined to be the decrease in a model score when a single feature value is randomly shuffled [36]. In other words, we can randomly drop a feature and see how much the model score drop in each round, which shows the individual feature's importance. It is model agnostic because we isolate the model effect and examine only the feature effects.

As shown in Section 6.1, the best basic model is the Gradient Boosting Tree. We are going to show the feature importance of this model in Figure 4 and Figure 5 using the two measures we mentioned above. We attach other feature importance plots in the appendix.

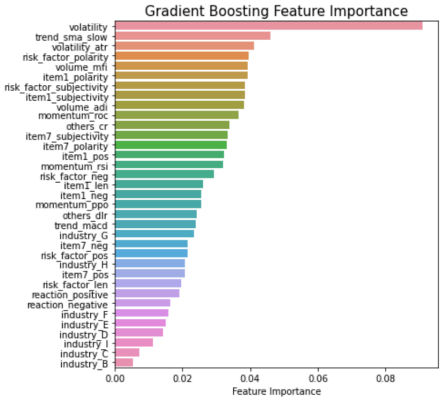


Figure 4: Gradient Boosting Tree Feature Importance

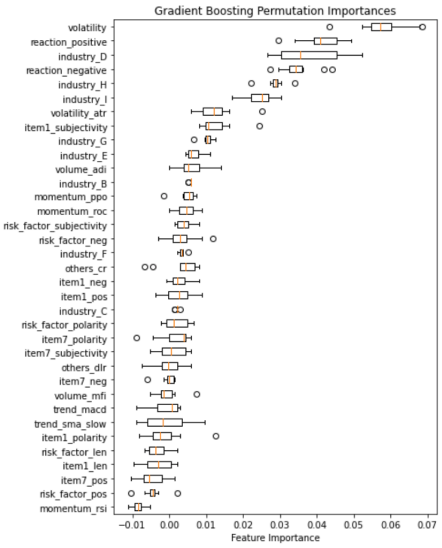


Figure 5: Gradient Boosting Tree (model agnostic) Feature Importance

From the plots, we can see the volatility feature contributes majorly to the model prediction. Other features created by feature engineering also show significant portions of contributions, which also match the results by using mutual information in section 4. Besides, there are some discrepancies for some feature importance scores, which show a high level of importance in model-specific measures but under-perform in the model-agnostic measure, such as trend_sma_slow and volume_mfi, which are both statistical indicators. However, the features constructed using the dictionary approach, such as reaction_postive and reaction_negative, show higher importance in the latter measure. The result reflects the usefulness of our customized features, especially the ones based on text analysis.

6.4 Technique for Improvement

To improve our final results, we finally apply eight techniques:

- We change the stock return method from long-term average simple return to long-term cumulative abnormal return in event studies.
- We construct new sets of features based on the dictionary approach, text classification approach, and the literature review, in addition to the statistical indicators calculated using external packages.
- We use the SMOTE method to solve the imbalanced data issue and standardize the dataset afterwards.
- We train the model using stratified CV.
- We use grid search to tune the parameters of the zeroth layer of the stacking model.
- We try different classifiers in the first layer of the stacking model.
- We try to build a deeper stacking model.
- We explore stacking models with different structures.

6.5 Stacking Model Performance

We can see that from Table 9, almost every stacking model performs better than base models, which means that as long as base models and meta-models are correctly selected, they can get better performance after stacking. In addition, the choice of meta-models is also related to performance. From Table 9, it can be seen that Logistic Regression is better than SVC in performance.

From the result of base models in Table 8, we find that the performance of the Decision Tree is worse than the others. We consider that the poor performance of one of the base models will degrade the performance of the entire stacking model. Therefore, we remove the Decision Tree from the base models and train the stacking model again. The results in Table

11 show that the performance can get better after trying different methods, which is 78.17%, more significant than the original one of 78.02%. However, the performance can sometimes be a little worse by using the seemingly advanced methods, which indicates that we should try out different base models and see which ones are the best.

Additionally, we try the multi-layer stacking model, in which the second layer is Decision Tree and SVC, and the meta-model is Logistic Regression. From Table 11, We can easily see that the performance of the multi-layer stacking model is much worse. It is not that the more stack layers, the better. Generally, the number of stack layers is two layers to achieve the best performance. The more layers are, the easier it is to overfit. Similarly, we should select simple classifiers for the meta-model to prevent overfitting.

6.6 Comparison

By deleting the features based on literature review, text classification, and the dictionary approach, we find that the model's accuracy in the test data drops by 4%-5%, which proves that the new features have obvious positive effects on the model.

7 Discussion and Conclusion

It is always worth discussing the potential and advanced solutions to conduct the long-term growth rate, as implementing 10-K filings analysis on its own is never a goal. More evidence and evaluation that could support the investors' and firms' financial decisions would be crucial.

During the project, we only consider the cumulative abnormal returns to predict the long-term growth rate. However, we should think about more indicators and factoring drivers used in the industry. Macroscopically, the firms' long-term growth rate in the industry may depend heavily on the economic long-term growth rate if the industry is cyclical. The current growth of productivity, demographic changes, and labor force participation in the economics are essential determinants [37]. Microscopically, focusing on a particular firm, the determinant of stock returns relies on the firm's valuation, where discounted cash flow (DCF) and Gordon Growth Model (GMM) are widely accepted methods. Therefore, influencers such as stock dividends and stock splits should also be acknowledged [38].

To extend the project's performance, the current NLP approach can be replaced by advanced procedures such as FinBERT. FinBERT is a language model based on BERT for financial NLP tasks. We can use FinBERT to evaluate 10-K filings and conduct due diligence, achieving the state-of-the-art on FiQA sentiment scoring and Financial PhraseBank [39].

Overall, the project is successful as the final accuracy of the stacking model has approached an accuracy of 78%. The result from Feature Engineering based on an Ablation study is also impressive. The features based on literature review, text classification, and the dictionary approach have improved the accuracy by an average of 5%. We also discuss many aspects of how we can improve in the future.

References

[1] G. P. Faisal Khalil, "Is deep learning and natural language processing transcending the financial forecasting? investigation through lens of news analytic process," *Computational Economics*, 2021.

[2] W.-T. W. Zhan Gao, "Predicting long-term earnings growth from multiple information sources," *International Review of Financial Analysis*, vol. 32, pp. 71–84, 2014.

[3] L. e. a. Moriaty, "Deal or no deal: Predicting mergers and acquisitions at scale," *International Conference on Big Data*, 2019.

[4] J. N. Andreas Simon, "Long-term growth forecasts and stock recommendation profitability," 2013.

[5] J. D. Sander Blomme, "Predicting the effect of 10-k, 10-q and 8-k company reports on abnormal stock returns using finbert nlp methods," *faculteit economie en bedrijfskunde*, 2020.

[6] J. B. J. Kasper Regenburg Jønsson, "Predicting stock performance using 10-k filings," *COPENHAGEN BUSINESS SCHOOL*, p. 88, 2018.

[7] M. Masoud, "Attention-based stock price movement prediction using 8-k filings," *Stanford Project*.

[8] T. A. Bohn, "Improving long term stock market prediction with text analysis," *Western Graduate Postdoctoral Studies*, 2017.

[9] A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara, "Predicting stock market trends by recurrent deep neural networks," *Pacific RIM International Conference on Artificial Intelligence*, 2014.

[10] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," *24th International Conference on Artificial Intelligence*, 2015.

[11] S. Chan, "Long term stock price-growth-prediction using nlp on 10 k financial reports," *GitHub*, 2020.

[12] E. Picardo, "Standard industrial cllasification (sic code) definition," *Investopedia*, 2021.

[13] Z. Zhang, S. Zohren, and S. Roberts, "Deep learning for portfolio optimization," *The Journal of Financial Data Science*, vol. 2, no. 4, p. 12, 2020.

[14] S. Chahine, "Long-run abnormal return after ipos and optimistic analyst' forecasts," *International Review of Financial Analysis*, vol. 13, no. 1, p. 21, 2004.

[15] R. F. Engle, E. Ghysels, and B. Sohn, "Stock market volatility and macroeconomic fundamentals," *Review of Economics and Statistics*, vol. 95, no. 3, pp. 776–797, 2013.

[16] C. Christiansen, M. Schmeling, and A. Schrimpf, "A comprehensive look at financial volatility prediction by economic variables," *Journal of Applied Econometrics*, vol. 27, no. 6, pp. 956–977, 2012.

[17] B. S. Schrimpf, "'déjà vol': Predictive regressions for aggregate stock market volatility using macroeconomic variables," *Journal of Financial Economics*, vol. 106, no. 3, pp. 527–546, 2012.

[18] C. Conrad and K. Loch, "Anticipating long-term stock market volatility," *Journal of Applied Econometrics*, vol. 30, no. 7, pp. 1090–1114, 2015.

[19] Z. Dai, H. Zhou, X. Dong, and J. Kang, "Forecasting stock market volatility: A combination approach," *Discrete Dynamics in Nature and Society*, p. 9, 2020.

[20] B. G. Malkiel, "Efficient market hypothesis," *Finance*, pp. 127–134, 1989.

[21] P. Liu, S. D. Smith, and A. A. Syed, "Stock price reactions to the wall street journal's securities recommendations," *The Journal of Financial and Quantitative Analysis*, vol. 25, no. 3, p. 12, 1990.

[22] J. Surowiecki, "The wisdom of crowds," p. 7, 2005.

[23] T. Loughran and B. McDonald, "Textual analysis in accounting and finance: A survey," *Journal of Accounting Research*, vol. 54, no. 4, pp. 1187–1230, 2016.

[24] D. Garcia, X. Hu, and M. Rohrer, "The colour of finance words," *Economics*, p. 67, 2020.

[25] T. Loughran and B. McDonald, "The use of word lists in textual analysis," *Journal of Behavioral Finance*, vol. 16, no. 1, pp. 1–11, 2015.

[26] T. Loughran and B. McDonald, "When is a liability not a liability? textual analysis, dictionaries, and 10-ks," *The Journal of finance*, vol. 66, no. 1, pp. 35–65, 2011.

[27] T. Loughran and B. McDonald, "Measuring readability in financial disclosures," *the Journal of Finance*, vol. 69, no. 4, pp. 1643–1671, 2014.

[28] H. Zhichao, "pysentiment," *Github*, 2019.

[29] S. Yan, C. Bin, and H. Zhuo, "Textual big data analytics in economics and finance: A literature review," *Economics*, vol. 18, no. 4, pp. 1153–1186, 2019.

[30] J. M. Chenlo and D. E. Losada, "An empirical study of sentence features for subjectivity and polarity classification," *Information Sciences*, vol. 280, pp. 275–288, 2014.

[31] Spinn3r, "Lydia/textmap system," 2009.

[32] SEC, "Standard industrial classification (sic) code list," *Division of Corporation Finance*, p. 1, 2021.

[33] P. Ravisankar, V. Ravi, G. Rao, , and I. Bose, "Detection of financial statement fraud and feature selection using data mining techniques," *Decision support systems*, vol. 50, no. 2, pp. 491–500, 2011.

[34] D. L. Padial, "Technical analysis library in python," *Github*, 2022.

[35] A. Gupta, "Feature selection techniques in machine learning," *Analytics Vidhya*, 2020.

[36] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[37] "Long-run growth," *Boundless Economics*, 2022.

[38] A. Rotkowski, "How to estimate the long-term growth rate in the discounted cash flow method," *Forensic Analysis Insights-Business Valuation*, 2013.

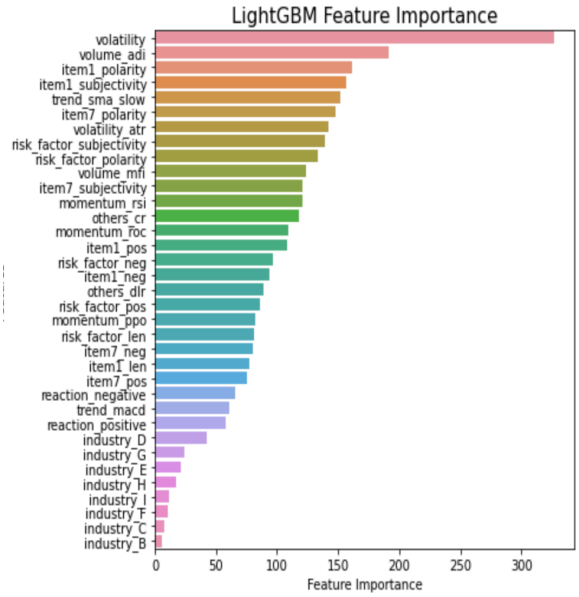
[39] D. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," 2019.

Appendix1: Dictionary

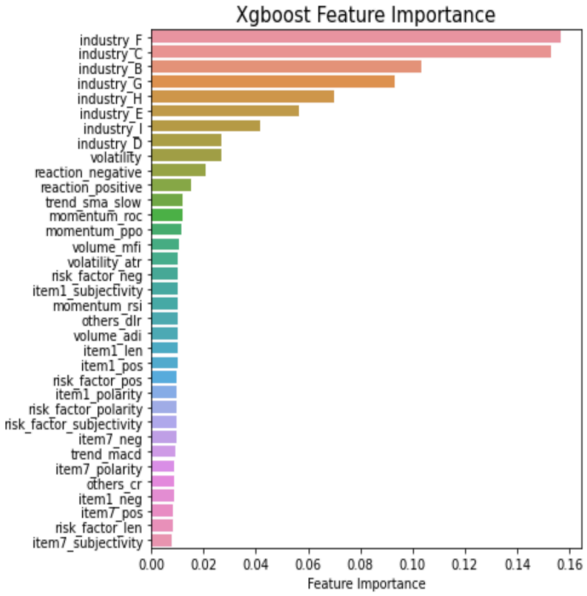
Abbreviation	Meaning
LSLR	Least Squares Linear regression
LR	logistic regression
SGD	Stochastic Gradient Descent
GB	Gradient Boosting
RF	Random Forest
NN	Neural Network
FS_all	Apply feature selection technique with all features.
industry_B	SIC codes
industry_C	
industry_D	
industry_E	
industry_F	
industry_G	
industry_H	
industry_I	
volume_adi	Statistical indicators
volume_mfi	
volume_atr	
trend_sma_slow	
trend_macd	
momentum_rsi	
momentum_roc	
momentum_ppo	
others_dlr	
others_cr	
reaction_positive	Dictionary approach variable
reaction_negative	
risk_factor_pos	
risk_factor_neg	
risk_factor_polarity	
risk_factor_subjectivity	
risk_factor_len	
item1_pos	
item1_neg	
item1_polarity	
item1_subjectivity	
item1_len	
item7_pos	
item7_neg	
item7_polarity	
item7_subjectivity	

Appendix2: Decision Tree Feature Importance

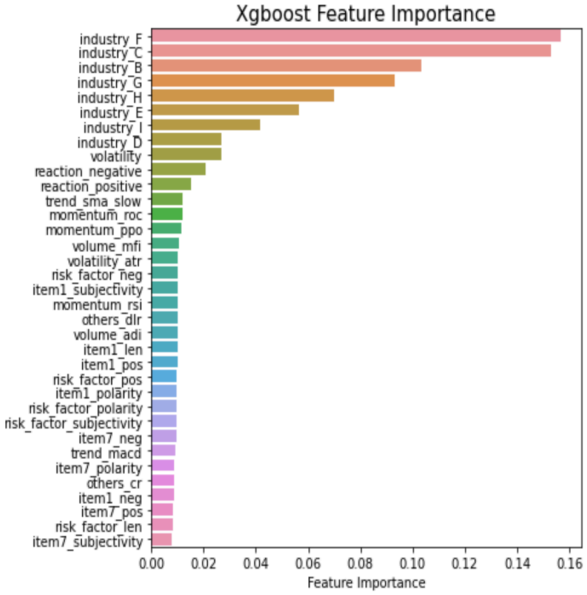
Appendix3: LightGBM Feature Importance



Appendix4: XGBoost Feature Importance



Appendix5: Random Forest Feature Importance



We accept peer report option1.