Natural Language Processing Project Report

Jacqueline Zhao(xuanziz, NY), Yiran Cao(yirancao, NY), Bowen Zhao(bowenzha, NY), Nikhila Kulukuru(nkulukur, NY)

October 13, 2023

Abstract

This paper employs Natural Language Processing (NLP) techniques to address a finance-related inquiry. It focuses on the classification of upside or downside price movements of S&P 500 companies based on textual information extracted from their 10-K year-end SEC filings. The methodology involves the application of unsupervised methods, including K-Means Clustering and Latent Semantic Analysis (LSA), to discern document structures and uncover latent topics. Subsequently, supervised learning methods such as Random Forests and Logistic Regression are applied to different text representations.

The findings indicate that the Random Forest Classification model using the Bag of Words Representation yields accurate predictions of stock price movements. Conversely, the Logistic Regression model employing word embeddings from the gensim package and LSA matrix proves effective in capturing semantic meanings within the text, achieving a commendable accuracy rate. The paper suggests potential enhancements by incorporating topic modeling outputs as features in the supervised learning methods or expanding this project to the Russell 3000 for increased training data. Additionally, the exploration of alternative financial-based word embedding techniques, such as FinBERT, could contribute to further model refinement.

1 Introduction

In this project, we are addressing a financial problem related to the stock price movement of the S&P 500 companies using their 2022 year-end 10K filings. This problem is intriguing and important because understanding the relationship between financial disclosures and stock price movements can offer valuable insights for investors and stakeholders. NLP techniques are highly relevant for this task, as they can extract and quantify subtle semantic cues and patterns from the vast textual data in 10-K filings, potentially highlighting predictive indicators of stock performance.

To tackle this project, we are drawing upon 10K filings of companies listed in the S&P 500 from the U.S. Securities and Exchange Commission's (SEC) Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system. These documents are pivotal as they contain a wealth of information, including financial summaries, managerial insights, and potential risks for each fiscal year. Given our goal to forecast stock price movements for 2022 based on these filings, their content is directly pertinent. The data can be accessed through the SEC's EDGAR platform: `https://www.sec.gov/edgar.shtml`. In this problem, we specifically look at only the MDMA (Management Discussion and Analysis) section of the 10-K filings. We particularly look into the MDMA section since it provides an overview of the company's financial performance during the reporting period. This section also discusses any changes or trends in the company's financials. This can help us understand how the company will perform in the near future.

We are leveraging unsupervised learning methods to gain deeper insights into our data. Our initial approach involves generating word clouds to identify the most frequent terms within documents associated with target variables '1' and '0.' Furthermore, we employ clustering algorithms to segment our data and Latent Semantic Analysis (LSA) to uncover key topics within each cluster. These unsupervised techniques enable us to assess whether distinct patterns emerge among companies with rising or falling stock prices, which can serve as valuable input for our more advanced supervised learning models.

Additionally, we leverage supervised learning to predict stock price movements of companies listed in the S&P 500, based on the content of their 10-K filings for the fiscal year 2022. The binary response variable we're focused on indicates whether a specific company's stock price either rises or falls during the month of its filing, which spans from December 1st, 2022, to December 31st, 2022. The 10K filings of S&P 500 companies are particularly relevant as they offer comprehensive financial information, management insights, and potential risk factors. Given the influence these factors can have on investor sentiment, analyzing the textual content of these filings can provide crucial insights that might not be evident from purely numerical data, making it instrumental in predicting stock market dynamics. The supervised learning models that we plan to implement to predict the forecast model are Random Forests and Logistic Regression.

2 Methods

2.1 Data

2.1.1 Data Collecting and Pre-Processing

In our text data, a document is the text of the MDMA section of a 10-K filing of a single company. Each observational unit of text is the entire MDMA paragraph which is a part of the 10-K filing. We are working with 495 documents in total. We aimed to work with all companies in the S&P 500. However, some companies are missing 2022 10-K filings and hence we are only working with 495 filings. The size of our vocabulary without stemming is 25,737, and the size of our vocabulary with stemming is 16,994. We decided to use stemming while creating the final word frequencies document term matrix.

Before we create the document-term matrix, we want to extract the 10-K filings using the SEC API and clean the text. We clean the text using various regex statements which remove scrub words, replace ASCII characters, remove HTML markup, remove extra spaces in the text, etc. We also calculate the number of words in every document after all the cleaning techniques have been implemented. We also notice that the first 90 characters of every document just have the section heading and HTML information which may not be useful to us. Hence, we remove the first 90 characters of every document.

Our next step is to calculate our target variables which imply whether the stock price went up or down in the month of December. We get the stock price of the company based on Yahoo Finance on 31st December 2022 and 1st December 2022. We then calculate the difference of the stock prices between these 2 dates. If the stock price is above 0, we encode the target variable 'priceMove' as 1 and if the stock price is below 0, we encode the target variable 'priceMove' as 0. Finally, 81 companies have an upside price move and 414 companies have a downside price move. Since our dataset is unbalanced, it might be hard to get good classification results while running the supervised and unsupervised learning algorithms.

2.1.2 Evaluation Metrics

For the unsupervised methods, we use a more qualitative evaluation metric. We run the algorithms over choices of K = 2/5/10 for the clusters and topics. We pick the final K based on how well the clusters are separated or how many documents fall into the different clusters or topics.

For the supervised methods, we use the accuracy score to evaluate the different models. The accuracy score can be denoted as the number of correctly classified documents / total number of documents. This helps us understand what proportion of documents are classified accurately. Finally, we want to pick a supervised learning algorithm that has a high accuracy score.

2.2 Methods

2.2.1 Unsupervised Methods

The unsupervised methods used in this project are K-Means Clustering, NMF, and LSA. Before we run any of the unsupervised learning methods, we run a word cloud on all the text that we have extracted from the 10-K filings. There are 23,251,920 words across all the documents of our dataset. The word cloud shown below in Figure 1 shows us the most important words across all documents after stop words are removed. We do not see any surprises in the large words. The 10-K filing is a year-end financial filing and hence having words such as 'year end', 'cash flow', 'decemb' etc make perfect sense. Other words such as 'increas', 'net sale', 'oper income' etc which are highlighted may be important as well while classifying the documents.

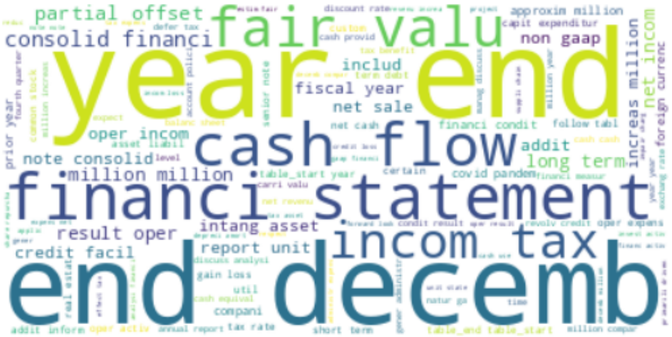


Figure 1: Word Cloud of Most Important Words

First, we run the K-Means Clustering algorithm. We have chosen to first employ this method in our analysis since we believe that our text data can be clearly separated into multiple clusters. Since we have binary labeling in the stock price movement (0 means the stock's price has gone down and 1 means the stock's price has gone up), we hypothesize that K = 2 might give us clusters in the data. We also run the algorithm for K = 5 and K = 10 to see if these clusters are better as well. Additionally, K-means is computationally efficient and works well with our high-dimensional data.

We also wish to run NMF and LSA algorithms on this text. However, NMF may not be the best method for our data since it cannot handle noisy data very well. Our data is very sparse and hence the clusters provided by NMF are not separated neatly. Additionally, the topics have similar words that repeat again and again. This does not make sense in this project since we are looking for topics that have terms that are unique to them. However, we use the LSA (Latent Semantic Analysis) algorithm to help with discovering the latent semantic structure within the document matrix. We again run the LSA algorithms for K = 5 and 10. As we can see, the LSA with 5 clusters models topics that effectively imply price move in each of the topics. We use the LSA matrix with K = 5 as an input to the supervised learning methods to capture semantic patterns. As a future improvement, we could also include LSA for feature engineering to reduce dimensionality and capture semantic patterns which could give us higher accuracy models.

2.2.2 Supervised Methods

The supervised methods used in this project are Random Forest Classifier and Logistic Regression. Before we apply these methods, we need to create features from the text data. We use 3 main methods to represent the features of the text - Bag of Words Representation, TF-IDF Representation calculated through both Python and manually, and finally Word Embeddings through the gensim package.

First, let's try to understand the different representations of the text data:

1. Bag of Words Representation - This representation gives us the word count of particular words in each document. The first step is to tokenize the text by breaking it into individual words or tokens. This is usually done by splitting the words based on spaces. The next step is to remove stop words from the tokens. Stop words are words that occur commonly in language that do not have much meaning, e.g. 'the', 'an', etc. Then the words or tokens are stemmed to reduce words to their root form. Finally, we count the number of times a word appears in every document and create a matrix of it.

2. TF-IDF Representation - TF-IDF stands for Term Frequency Inverse Document Frequency. After cleaning the text similar to how it is cleaned in the bag of words representation, we can apply the TF-IDF concept to the tokens. TF-IDF is a numeric representation of tokens that reflects the importance of a word relative to a big collection of documents. In this case, we explore the TF-IDF matrix construction through both Sklearn's inbuilt function as well as a manual computation.

3. Word Embeddings - We also try to explore word embeddings through the Gensim library of Python. We leverage the GloVe model to create embeddings for the documents. Using a pre-trained word embedding model helps us capture any semantic relationships within a document.

4. LSA (Latent Semantic Analysis) - We also use the LSA from the Unsupervised section as an input to the Supervised Learning methods. The LSA matrix starts off with a Bag of Words matrix and further reduces the dimensionality of this matrix using an SVD decomposition. These reduced dimensions captured by LSA often capture semantics in our documents. We hypothesize that using the LSA matrix will help us get models with higher accuracy.

For each of the text representations, we run 2 supervised models -

1. Random Forest - Random Forest is a supervised learning method for classification and regression tasks. In this case, we use Random Forest as a classification technique. It is a very powerful and flexible technique that combines the principles of bagging and a decision tree algorithm. Essentially, by combining multiple decision trees, random forest tends to reduce over-fitting and increases the model's robustness. We can also create feature importance plots through Random Forest. In this case, we can pick out which words are the most important. Additionally, due to averaging, the model is resistant to outliers and noisy features or data.

2. Logistic Regression - Logistic Regression is a statistical method often used for binary classification tasks. This method makes perfect sense for this problem since we are trying to predict whether the stock price will move up or down. Logistic regression works well on small to medium-sized data and is very computationally efficient. We incorporate the L1 regularization into this technique to prevent overfitting.

3 Results

3.1 Unsupervised Methods

For the K-Means Clustering algorithm, we run the algorithm for K = 2, 5, and 10. We create bar plots of the number of documents in each of the clusters for all options of K. From the bar chart of K = 2, we can see that the documents are based on their price movement. We see that the bar plot with K = 5 and K = 10 gives us well-partitioned clusters - some of bigger sizes and some of smaller sizes (Bar charts can be found in Appendix 1). We also run the word clouds of all the clusters for different choices of K. We can see that the word clouds for the clustering with K = 5 (Appendix 2) gives us the best division of topics. The words highlighted are meaningful to predict the response variable. Hence, we can safely say that for K-Means clustering, the division of the documents into 5 clusters makes the most sense.

For the LSA algorithm, the configuration with K = 5, reveals the most effective division of clusters and topics (Appendix 3). A closer examination of the word clouds generated by these clustering algorithms showcases interesting insights. While words like 'million' are common across all clusters, some word clouds exhibit unique terms such as 'revenue' or 'loss,' which significantly enhance our comprehension of these clusters. Furthermore, an exploration of the words within the 5 topics generated by the LSA algorithm highlights terms like 'emea' and 'notional,' further aiding our understanding of the underlying semantic nuances within the documents. In general, the clusters tend to share financial terminology in their word clouds, except one cluster that prominently features terms like 'risk' and 'loss,' conveying a negative sentiment. This outcome aligns with expectations since basic K-means clustering does not inherently consider semantic meanings when clustering text data. Instead, it relies solely on numerical data representations and distance metrics based on raw word frequencies, overlooking the semantic relationships between words.

Conversely, LSA offers more distinct topics, which more directly correlate with price movements. Hence, the integration of LSA with K-means clustering emerges as a potent approach for our data clustering task. LSA extracts semantic information from the text, reduces dimensionality, and enhances the quality of features employed by K-means. This synergy can potentially result in more meaningful and interpretable clusters, ultimately improving our data analysis.

3.2 Supervised Methods

Accuracy		
Representation/SL methods	Random Forest	Logistic Regression
Word Frequency	**84.564%**	79.195%
TF-IDF (sklearn)	83.893%	75.839%
TF-IDF (manual)	83.221%	76.510%
Gensim	83.636%	**83.434%**
LSA	83.030%	**83.434%**

For Logistic Regression, we can see that the highest accuracy score is achieved by using Gensim and LSA, which give us significantly better results than the other representations. This aligns with our expectations because they capture the semantic meanings of our text well. Furthermore, it is surprising that TF-IDF implemented manually performs worse than the sklearn TF-IDF in both Logistic Regression and Random Forest given that sklearn TF-IDF is computed incorrectly.

For Random Forests, the accuracy scores are very close across the different representations. This may be due to the fact that Random Forest is more robust since it is non-linear and more flexible and can thus capture complex relationships between features and target variables. As a result, the representation of data is not as important as it is for Logistic Regression. This can also help explain why the highest accuracy score in this case is achieved not by using representations that consider semantic meanings but by word frequency. In addition, its robustness may also contribute to it consistently performing better than Logistic Regression. To further evaluate the performance of Random Forest using word frequency, we will perform a feature importance as shown below.

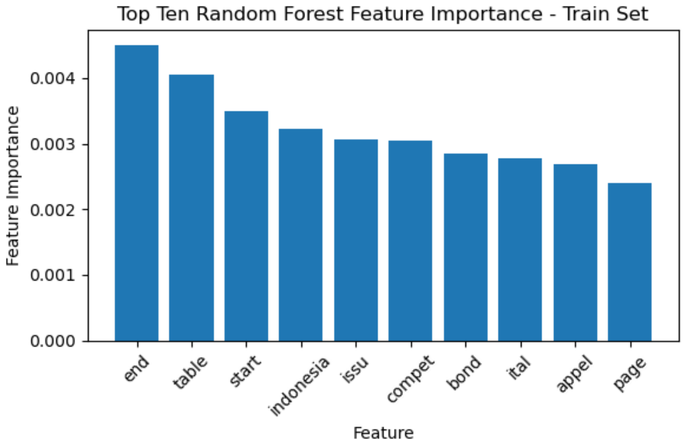


Figure 2: Random Forest Feature Importance with Word Frequency

The features below are the words that contribute the most to distinguishing between the two classes (0 and 1). In our binary classification problem, these words should be strongly associated with one class or the other. These 10 words are considered more important by the model when making decisions. We can see that these features are mostly actual words instead of random combinations of letters. This means that the model is somewhat effective in differentiating between meaningful words from nonsense text and in extracting relevant information from the text to make classifications. However, if we look at the semantic meanings of these words, only a few words such as *end* and *start* more directly imply price move. On the other hand, words such as *table* and *page* do not seem relevant to price move. There may be many reasons for this, one of which is overfitting especially when there are too many features relative to the amount of data (495 x 16994). Another reason is that word frequency does not capture semantic meanings well. As a result, some of these important features are not directly related to price moves in context. This further supports that gensim may be a better representation of our text.

4 Discussion

The project explores ways to perform binary classification on S&P 500 companies' stock price movements based on their 10-k filings. We first employ unsupervised learning methods including K-Means Clustering and LSA to examine the data to gain insights on the ideal number of partitioning and the major topics of our 10-k filings texts. Initially, we aimed to have two clusters (K = 2) in our K-Means analysis since our classification task is binary. However, our exploration revealed that the optimal number of clusters appears to be K = 5, as indicated by the word clouds. This suggests that our dataset holds more complexity and substructure than the simple binary classification problem we seek to address. The data could represent various subgroups, such as different industries or company sizes, which do not neatly fit into just two categories. Nevertheless, that doesn't change our binary classification task.

The word clouds generated from these clusters mostly share financial terminologies, highlighting the limitation of K-Means Clustering in capturing the semantic nuances of textual data. In contrast, the LSA topics provide a more direct correlation with price movements. One topic comprises positive words like "increase," directly implying an upward price movement, while three other topics feature negative terms like "risk," "loss," and "catastrophe," indicating a downward price trend. Interestingly, the LSA configuration that best aligns with our context also suggests K = 5. Although these topics are not binary and reflect a more complex structure, that still doesn't change our binary classification task. In fact, we intend to leverage these unsupervised learning assignments for feature engineering to reduce the dimensionality of our data and specifically for LSA to also capture the semantic meanings of our text. This reduction could potentially enhance the performance of our binary classification model when we transition to supervised learning.

The subsequent phase of our project focuses on the binary classification task, where we employ two supervised learning methods, namely Random Forest and Logistic Regression, across five different text representations: word frequency, sklearn TF-IDF, manual TF-IDF, gensim, and Latent Semantic Analysis (LSA). A notable distinction among these representations is that gensim and LSA have the capacity to capture semantic relationships within our text documents, which the others do not. As we observed in both the unsupervised learning and the feature importance part of the supervised learning segments of the project, the partitioning of our text data without considering these semantic relationships may introduce inaccuracies in predicting our binary labels. Therefore, we anticipate that our supervised learning models will achieve higher accuracy when using gensim and LSA.

Our Logistic Regression results confirm this expectation, demonstrating that gensim and LSA significantly outperform the other representations. In contrast, the Random Forest results appear less clear. For one, the Random Forest results exhibit striking similarities across these representations.

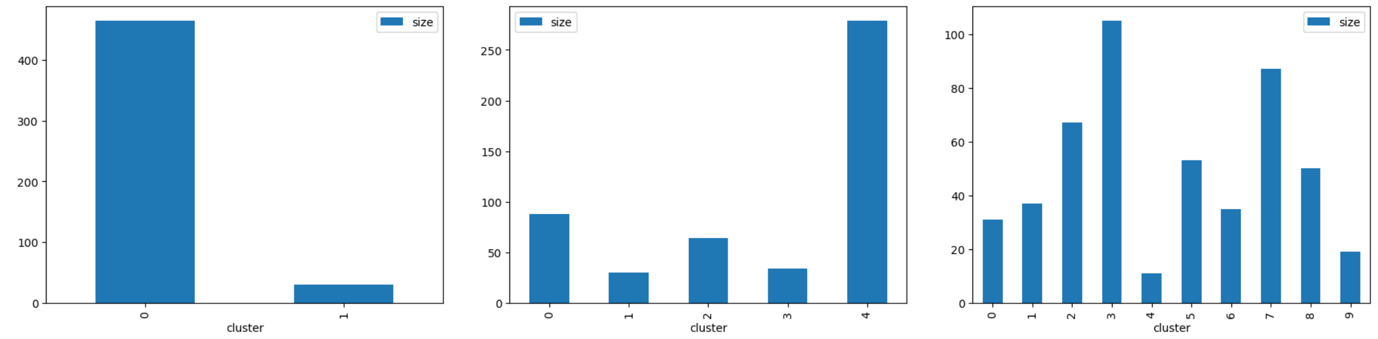
For another, amidst these similar scores, it assigns the highest importance to word frequency rather than gensim or LSA. This behavior can be attributed to the robustness and flexibility of the Random Forest algorithm, which can accommodate variations in data representation.

In conclusion, our study underscores the important need for our text representations to effectively capture semantic meanings to enable the successful performance of binary classification models, particularly when employing Random Forest and Logistic Regression. However, it's important to acknowledge certain limitations in our findings.

Firstly, gensim and LSA, while promising in their ability to capture semantic relationships, do not generate features in the same manner as traditional methods. This absence of traditional feature importance assessment makes it challenging to verify the direct relevance of these features to price movements within the given context. Additionally, it's important to acknowledge that our binary classification models, while effective, may not capture more nuanced or subtle relationships between text representations and price movements. Furthermore, the choice of classifiers may impact the model's performance, and alternative algorithms could yield different results. Moreover, the quality and quantity of the data used in training these models can significantly influence their performance and generalizability. A more extensive dataset or data preprocessing techniques could potentially address some of the limitations.

In future research, it might be beneficial to explore alternative feature engineering methods or experiment with different machine learning algorithms to improve the model's ability to capture and interpret semantic information. Understanding these limitations is crucial for refining our approach and obtaining more reliable results in the context of predicting price movements.

A Appendix 1



(a) K=2

(b) K=5

(c) K=10

Figure 3: Bar Plots for K-Means Clustering

B Appendix 2

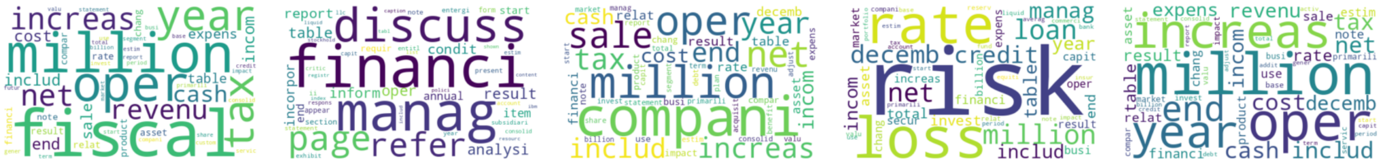
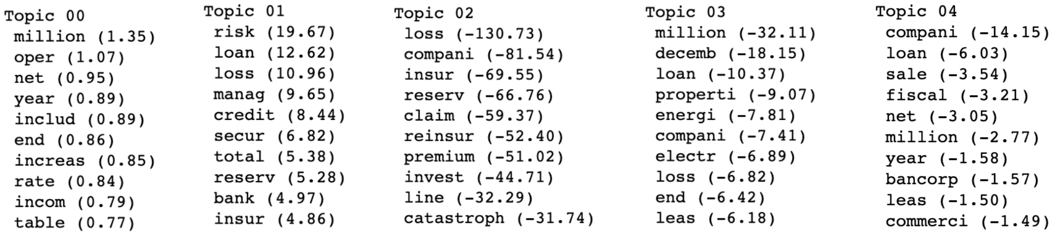


Figure 4: Word Cloud for K=5

C Appendix 3



```
Topic 00             Topic 01            Topic 02                 Topic 03               Topic 04
 million (1.35)       risk (19.67)        loss (-130.73)           million (-32.11)       compani (-14.15)
 oper (1.07)          loan (12.62)        compani (-81.54)         decemb (-18.15)        loan (-6.03)
 net (0.95)           loss (10.96)        insur (-69.55)           loan (-10.37)          sale (-3.54)
 year (0.89)          manag (9.65)        reserv (-66.76)          properti (-9.07)       fiscal (-3.21)
 includ (0.89)        credit (8.44)       claim (-59.37)           energi (-7.81)         net (-3.05)
 end (0.86)           secur (6.82)        reinsur (-52.40)         compani (-7.41)        million (-2.77)
 increas (0.85)       total (5.38)        premium (-51.02)         electr (-6.89)         year (-1.58)
 rate (0.84)          reserv (5.28)       invest (-44.71)          loss (-6.82)           bancorp (-1.57)
 incom (0.79)         bank (4.97)         line (-32.29)            end (-6.42)            leas (-1.50)
 table (0.77)         insur (4.86)        catastroph (-31.74)      leas (-6.18)           commerci (-1.49)
```

Figure 5: LSA topics for K=5