

**Introduction to FinTech**

**Data Analysis Project 2**

**Textual Analysis using Natural Language Processing**

|  |  |
| --- | --- |
|  | |
| Kai Ren | 2401212437 |
| WeiJie Zhang | 2401212474 |
| Yang Lan | 2401212401 |
| YiMing Jiang | 2401212398 |
| Rui Hu | 2401212392 |

**Peking University**

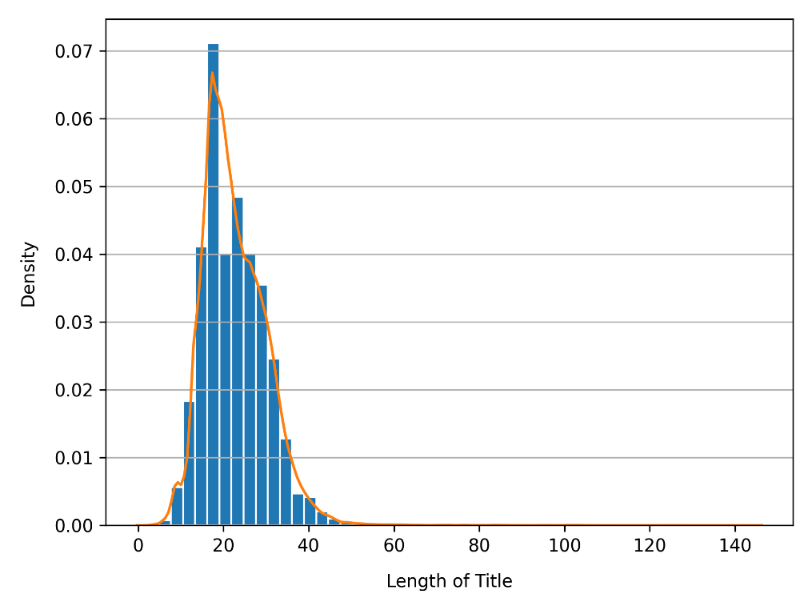
2024.11.4

# Abstract

For Data Analysis Project 2, we use text data to forecast stock returns based on GPT word embedding model and various dimension reduction methods. Focusing on stocks listed on the main board, we collected research report text data from Dongfang Wealth using web scraping techniques, covering a period of five and a half years from January 1, 2019, to June 1, 2024, totaling 87,429 entries. Based on these raw data, we use the GPT text-embedding-3-small model to map the title of each text to a 768 dimensional vector and standardize it. The dimension of 768 is relatively high and may contain more redundant information. We used three dimension reduction methods, namely PCA, Lasso and XGBoost, to reduce the dimension of each sample to less than 250 dimensions. We use minute frequency data to calculate ‌ volume-weighted average price (vwap) in the first ten minutes of the morning trading. Based on vwap, long-term return and short-term return are constructed as labels to analyze the influence period of the text data. According to the time interval, the training sample was divided into three parts: training set, verification set and test set (from January 1, 2024 to June 1, 2024). Finally, we compare the effect of three dimension reduction methods on the results and the prediction effect of text data on short and long term returns. In the short term, there is no obvious difference between different dimension reduction methods. In the long term, F1 index has a certain decline after dimension reduction, among which Lasso method has the most serious decline, which may be related to the importance of judging by when dimension reduction is adopted. At the same time, we note that has the best prediction results, which may be related to the length of the influence period of text information.

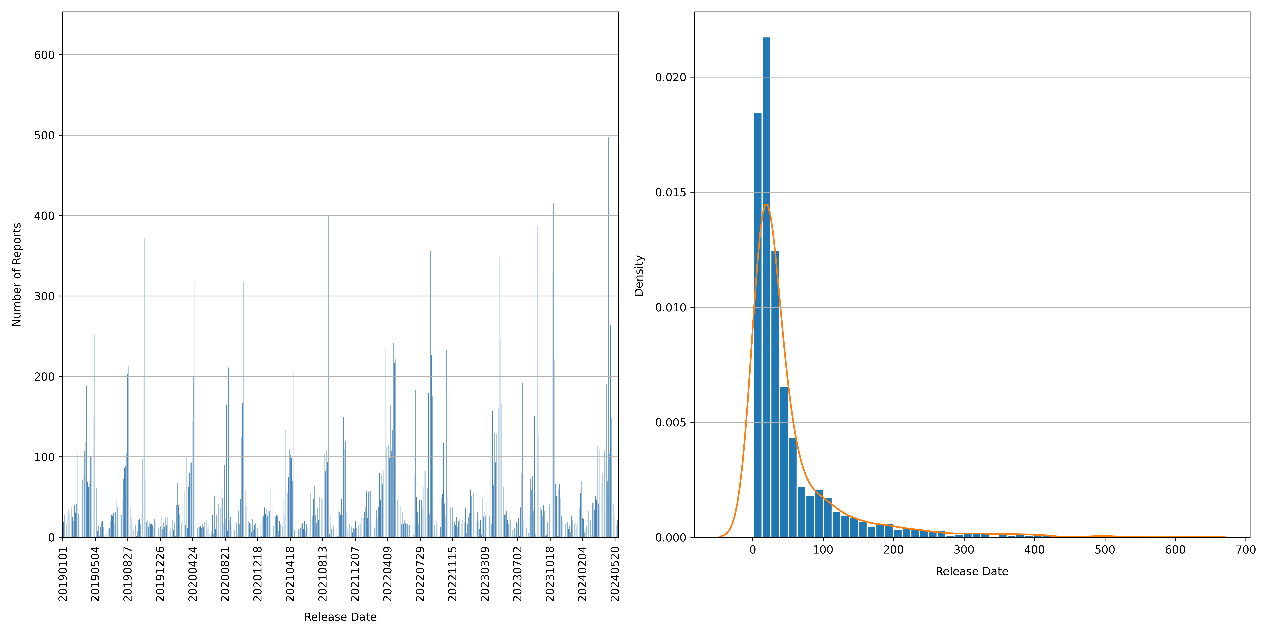
# Ⅰ. Data Collection

By analyzing the data request formats between the web pages and databases, we mimicked a series of POST requests to obtain all individual stock research report titles from the data center on Eastmoney.com for the period from January 1, 2019, to June 30, 2024, totaling 129,080 entries. Subsequently, according to our research objectives, we narrowed down the stock selection to all listed stocks on the Shanghai and Shenzhen Main Boards as of early October, totaling 3,153 stocks, yielding 87,429 entries. The average title length was 22.67 characters, with a minimum length of 2 characters (e.g., "Reversal" by Zhongtai Securities for Zhongfu Industrial, 600595.SH, on January 9, 2024) and a maximum length of 144 characters (e.g., "Earnings Review: Mixed FY19/1Q20: Expect domestic revenue to recover, expense ratio to stabilize; exposure to overseas retail market a risk; Buy" by Goldman Sachs Gao Hua Securities for Haier Smart Home, 600690.SH, on May 8, 2020).



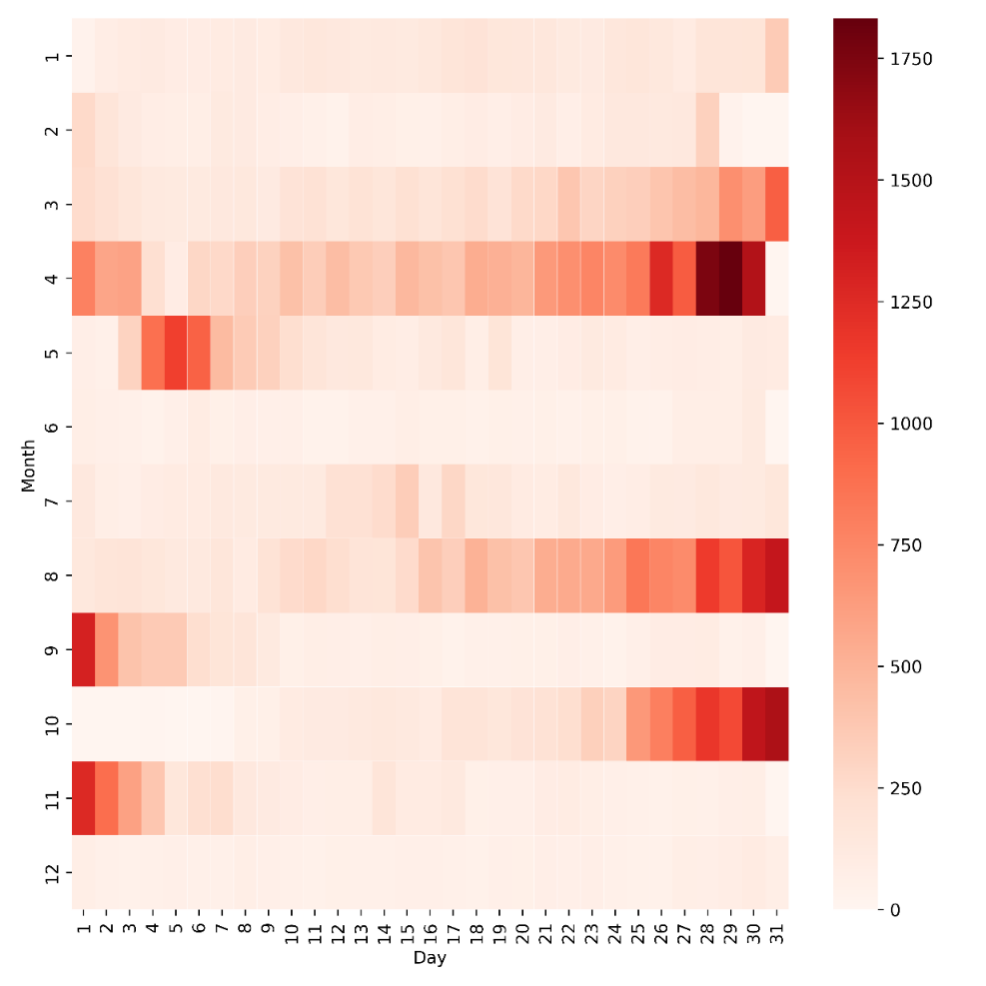
**Figure 1.** Distribution of Title Length in Individual Stock Research Reports.

From a temporal perspective, within the time frame from January 1, 2019, to June 30, 2024 (a total of 2,007 days), there were 1,711 days on which at least one report was published. The average daily publication volume was 43.56 reports, with 34 days having a publication volume of only one report, and the highest volume reaching 622 reports (on May 4, 2023).



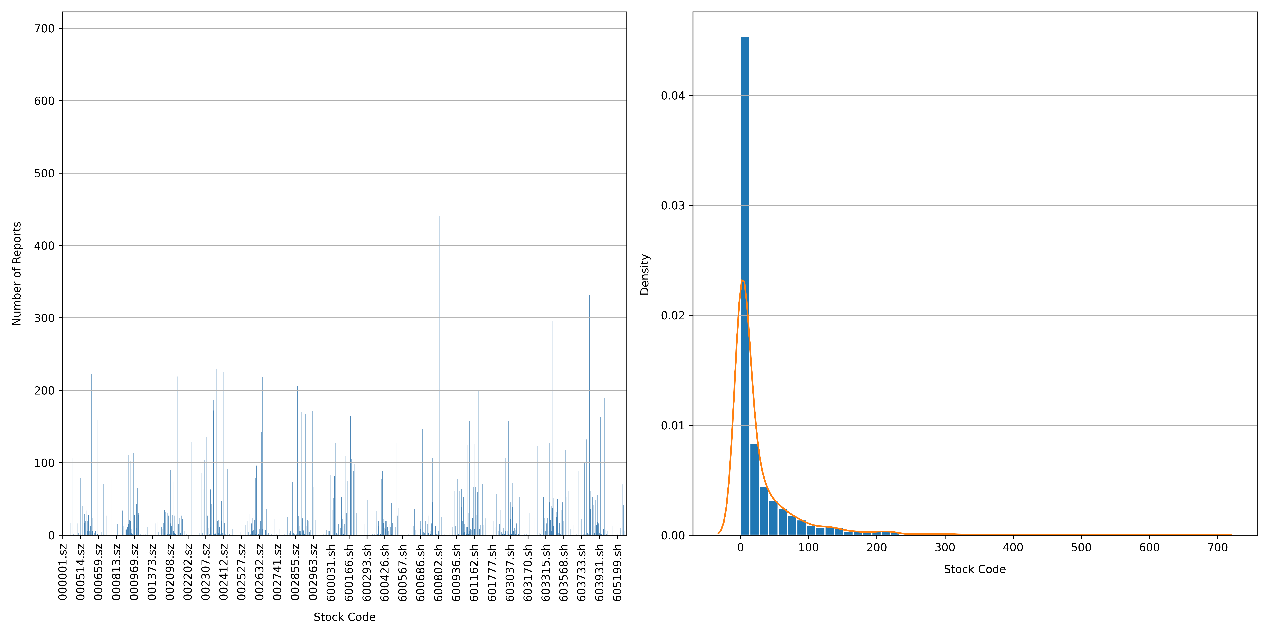
**Figure 2.** Bar chart and distribution of the number of daily stock reports published.

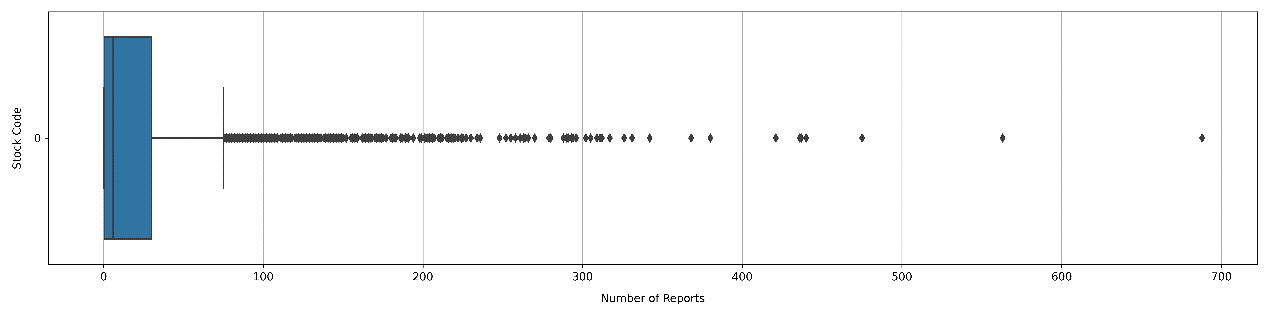
As shown in the above chart, the number of published reports demonstrates a certain periodic pattern. By further aggregating daily data for each year, we can obtain the heatmap shown below, revealing three peaks in report publications around the end of April, from late August to early September, and from late October to early November.



**Figure 3.** Heat map of the number of daily stock reports published.

From the stock perspective, among all main board stocks listed on the Shanghai and Shenzhen exchanges as of early October (a total of 3,153 stocks), only 2,392 stocks had at least one research report within the study period. The average number of reports per stock was 27.73, with a median of 6 reports. The stock with the most reports was Kweichow Moutai (600519.sh), which had 688 reports.



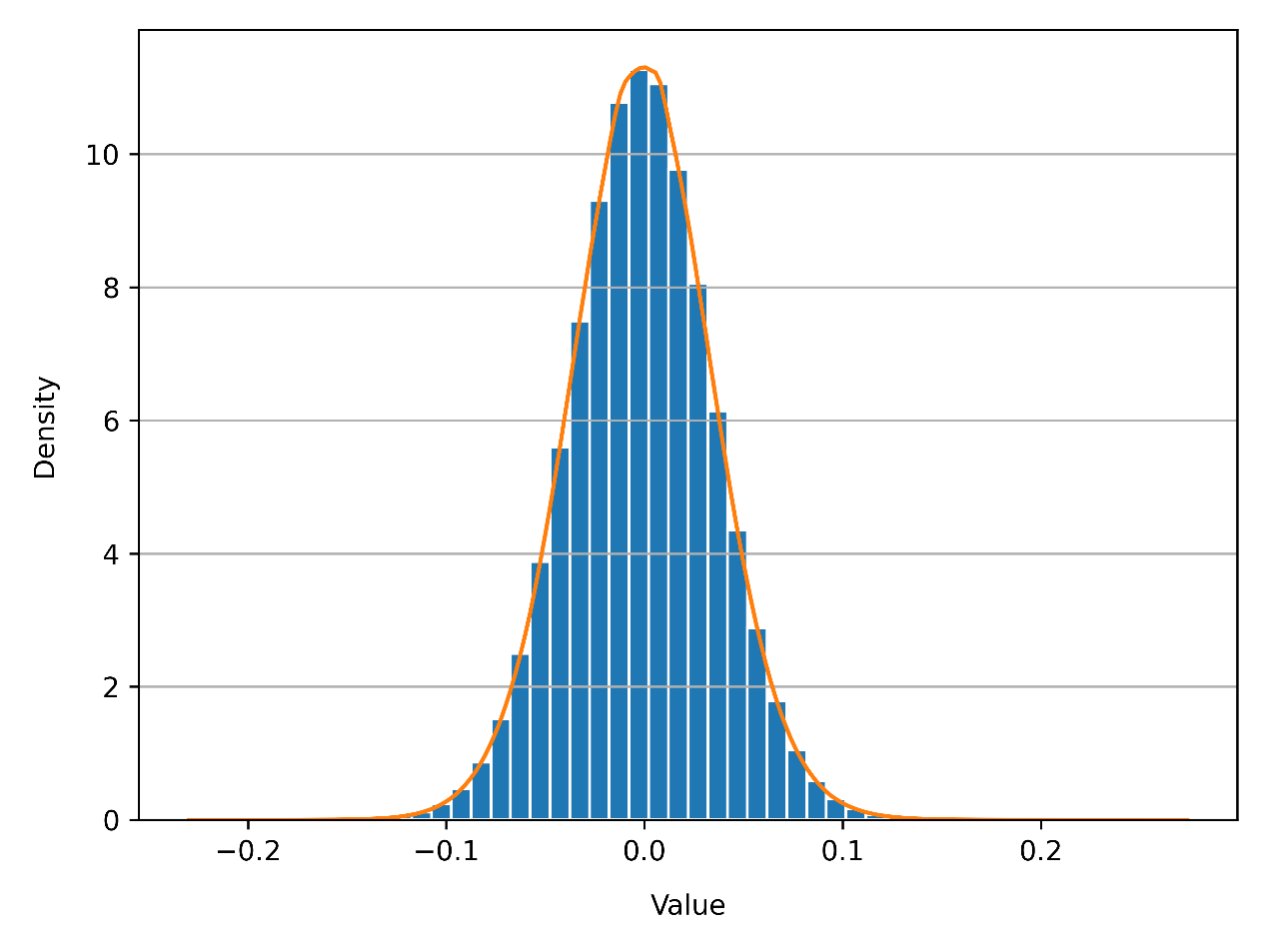


**Figure 4.** Bar chart, distribution, and boxplot of the number of reports published per stock.

# Ⅱ. Factor Engineering

If a report is published on a non-trading day, the corresponding text data will automatically shift forward to the next trading day. This adjustment implies that text information from non-trading days takes effect on the next trading day, preventing the model from using future information.

For label construction, we used 5-minute frequency data to calculate the VWAP for the first 10 minutes after the market opens (09:30 - 09:40), and used as the return rate from day t to t+1. Similarly, represents the return rate from day *t* to *t+n*. For day *t−1*, represents the return rate from day *t* to *t+1*, represents the return from day t to t+3, and represents the return from day t to t+10. These labels across different periods are used to evaluate the model's predictive performance on both short-term and long-term returns.



**Figure 5**. Distribution of data obtained after text embedding, with maximum value 0.2710, minimum value -0.2268, mean value -0.0012, standard deviation 0.0361.

For text embedding, we utilized OpenAI's text-embedding-3-small model, inputting research report titles gathered during data collection. Traditional word embedding methods, such as Word2Vec and GloVe, cannot fully capture the semantic changes of words in different contexts. The text-embedding-3-small model is based on Transformer architecture, trained on large-scale text data, and has rich language understanding ability. This model can generate context-aware dynamic word vectors, identify and process finance-specific terms and expressions. This not only improves the accuracy of financial text analysis, but also improves the ability to understand market dynamics and investment sentiment. This model outputs a 768-dimensional vector. The data distribution is roughly uniform, with a maximum value of 0.2710, a minimum of -0.2268, a mean of 0.0012, and a standard deviation of 0.0361.

# Ⅲ. Dimension Reduction

**3.1 Purpose of Dimension Reduction**

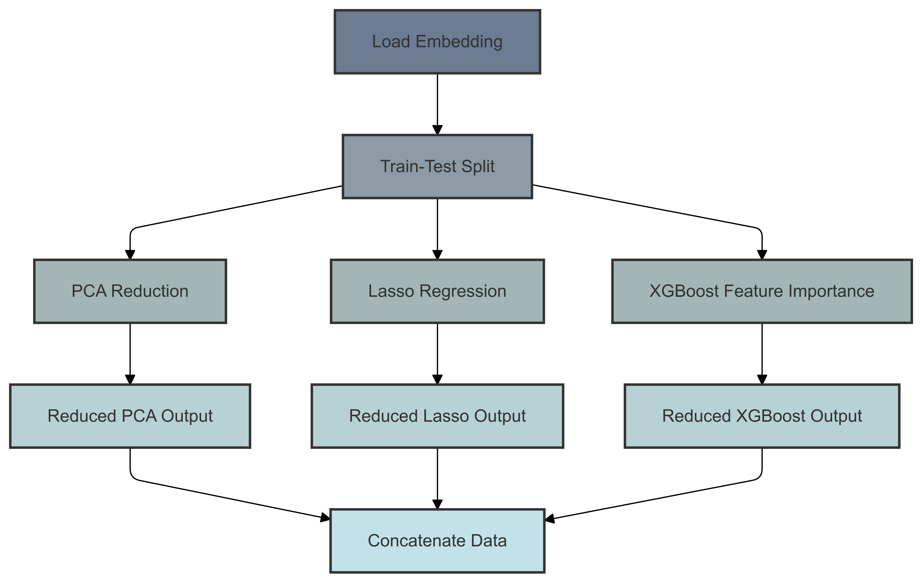
In predictive modeling, especially when working with text embeddings that result in a high-dimensional feature space—such as our embedding layer with a dimensionality of 768—abundant information may cause overfitting, this becomes a significant concern. While these high-dimensional embeddings capture intricate details and nuances of the textual data, they can also include a considerable amount of noise. This excess information can mislead the model, causing it to learn patterns that do not generalize well to unseen data.

Reducing dimensionality is crucial in this context as it helps to increase the signal-to-noise ratio, allowing the model to focus on the most meaningful features. By distilling the information into a more manageable number of dimensions, we can extract economically meaningful insights while preventing overfitting. This process not only enhances the model's robustness but also improves its predictive performance on stock returns.

**3.2 Process of Dimension Reduction**

To manage the high dimensionality of the text embeddings (768 dimensions), we employed three distinct dimension reduction techniques: PCA (Principal Component Analysis), Lasso regression, and XGBoost feature importance analysis. Each method utilizes the same input data but yields different outputs, allowing for a comparative analysis of their effects on model performance.

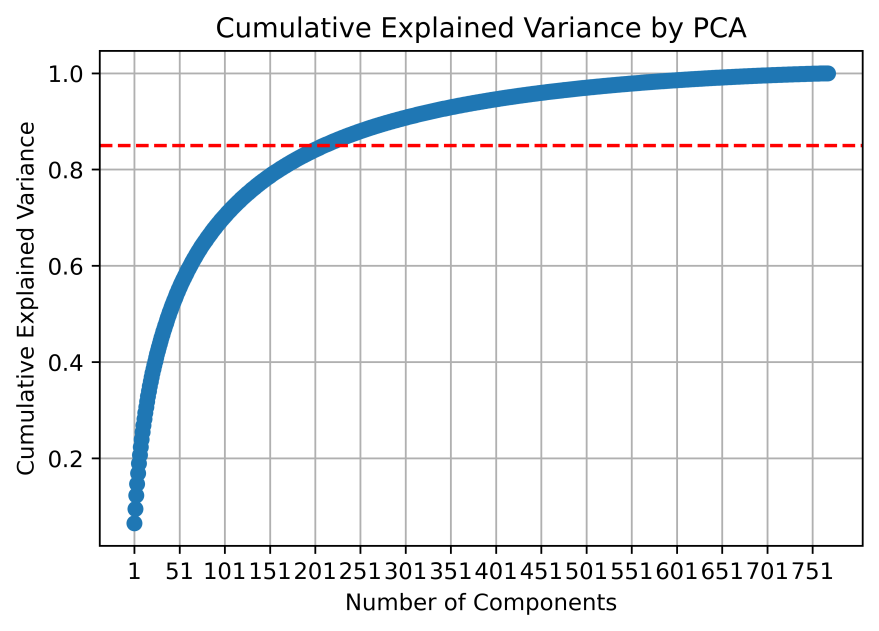
* Train-Test Split: Split the data into training and testing sets based on a predefined date to facilitate model validation.
* PCA Reduction: Apply PCA to the training data to identify principal components that explain a cumulative variance threshold (e.g., 85%). The output includes the reduced training and testing sets, focusing on the selected principal components.
* Lasso Regression: Implement Lasso regression on the same input data with , using L1 regularization to select significant features. The output consists of the reduced training and testing sets, retaining only features with non-zero coefficients.
* XGBoost Feature Importance: Use XGBoost to assess feature importance based on the same training data with . The output includes the reduced training and testing sets, highlighting the most impactful features based on their importance scores.
* Data Concatenation: Concatenate the reduced data from each method back to the original DataFrame to maintain data integrity for further analysis.



**Figure 6.**  Pipeline for Dimension Reduction

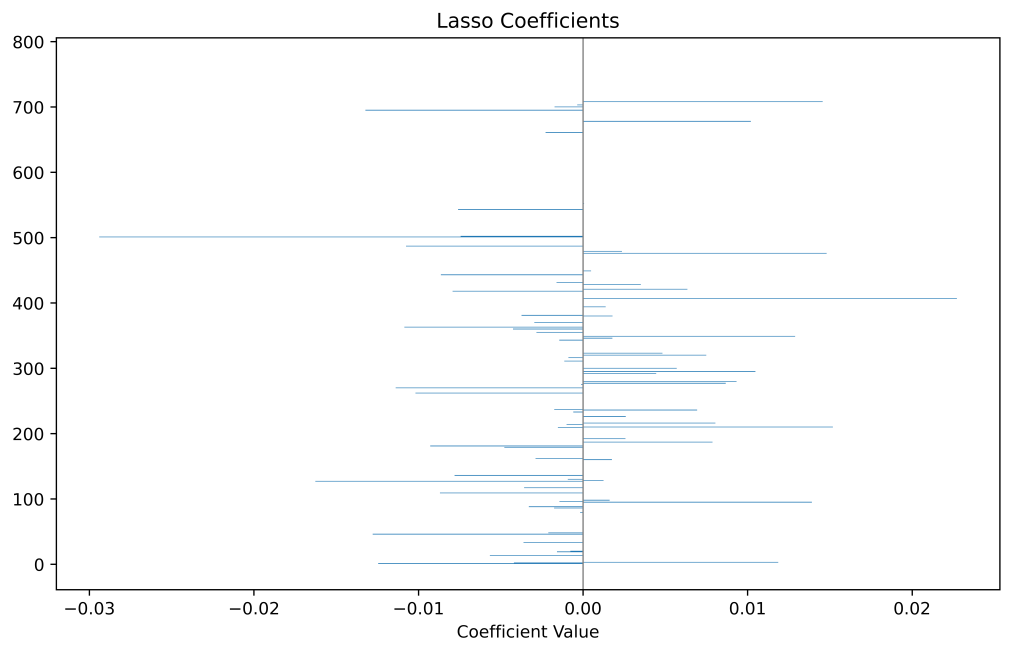
**3.3 Result of Dimension Reduction**

By adopting PCA, we condensed the 768 dimensions of one text into 211, which means the 211 principle components can explain 0.85 variance, and Fig 3.2 shows umulative explained variance by PCA.



**Figure 7.** Cumulative Explained Variance by PCA

Through Lasso Regression, 77 dimensions are selected to explain the original data, Fig 3.3 shows their coefficients. Meanwhile, after applying XGBoost, 230 dimensions are sifted to dominate the original data meaning.



**Figure 8**. Coefficients for Lasso selection

# **Ⅳ. Results**

We employed three methods: Principal Component Analysis (PCA), Lasso regression, and XGBoost, using short-term return as the importance assessment criterion to perform dimensionality reduction on the original embedding data. Meanwhile, we utilized three models—Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU)—with , , and as labels to explore the predictive power of text data on both long-term and short-term returns. After training the models on the training set, we tested them on the test set.

**Table 1.** The summary of out-sample modeling result.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | |
| Embedding | | PCA | | Lasso | | XGB | |
| ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 |
| MLP | 0.51 | 0.51 | 0.50 | 0.50 | 0.52 | 0.51 | 0.52 | 0.52 |
| LSTM | 0.53 | 0.50 | 0.52 | 0.52 | 0.53 | 0.50 | 0.53 | 0.51 |
| GRU | 0.52 | 0.51 | 0.53 | 0.51 | 0.52 | 0.51 | 0.52 | 0.51 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | |
| Embedding | | PCA | | Lasso | | XGB | |
| ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 |
| MLP | 0.53 | 0.52 | 0.51 | 0.50 | 0.54 | 0.48 | 0.53 | 0.52 |
| LSTM | 0.54 | 0.50 | 0.53 | 0.51 | 0.54 | 0.47 | 0.54 | 0.51 |
| GRU | 0.55 | 0.50 | 0.54 | 0.50 | 0.53 | 0.50 | 0.55 | 0.50 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | |
| Embedding | | PCA | | Lasso | | XGB | |
| ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 |
| MLP | 0.52 | 0.51 | 0.51 | 0.51 | 0.55 | 0.45 | 0.51 | 0.50 |
| LSTM | 0.51 | 0.51 | 0.53 | 0.50 | 0.54 | 0.49 | 0.51 | 0.51 |
| GRU | 0.51 | 0.51 | 0.53 | 0.50 | 0.53 | 0.49 | 0.51 | 0.51 |

Based on the results from the test set, we reached the following conclusions: in short-term predictions, there were no significant differences between the original data and the results from various dimension reduction methods. This indicates that there is indeed redundant information in the original 768-dimensional embedding, and the dimension reduction methods performed well in extracting text information. However, in the long term, the F1 scores after dimension reduction all showed a decline, with the Lasso method experiencing the most significant drop to 0.48. This may be related to our use of as the importance criterion during the dimension reduction process. Additionally, we observed that had the best predictive results, which may be related to the duration of the impact of text information on returns.

Overall, we explored stock price prediction based on text data, focusing on two aspects: the duration of return forecasts and the dimension reduction of text word embeddings. Building on this research, future experiments can delve deeper into these areas.