Abstract

This report uses publicly available data to predict the change of stock price by using

Natural Language Processing (NLP) methodology. The text data is five years (2018~as of now 2022) of transcripts from he quarterly earnings calls of six publicly traded, NASDAQ-listed companies as the basis to compare six different text-based predictive analysis approaches over two different forecast horizons, one business day after the earnings call and five business days after the earnings call.

Stock performance is assessed by comparing the change in the price of the company’s stock relative to the change in NASDAQ index over the same time period. The methodology leverages the following NLP processes to vectorize the transcript text: TF-IDF, Word2Vec.

This report explore the following consideration:

1. What is the call’s sentiment? What are common words/topics used across company earnings calls?
2. Whether longer time forecast horizon have more predictive power compared to a relatively shorter time forecast horizon as the market needs more time to digest “soft” text-based document after earning calls
3. In addition, we also explore KNN, Naive\_Bayes, and Adaboost on our Word2Vec dataset to see if we could get better results and build predictive model.

Introduction

Machine learning literature reviews show that standard machine learning models can

predict short-term directional movements in stock prices with enough accuracy to be profitable in some mature financial markets.

Two academic surveys of machine learning and stock predictions outline the success of traditional machine learning algorithms to create profitable stock trading predictions (Hsu et al. 2016, Lv et al., 2019). Correspondingly, a research sing earning calls transcripts of S&P 500 ,17025 earning calls transcripts over 485 companies shows that the proposed model is superior to the traditional machine learning baselines and earnings call information can boost the stock price prediction performance.

Our empirical experiments show that the proposed model is superior to the traditional machine learning baselines and earnings call information can boost the stock price prediction performance. In addition to considering markets and models that generate effective stock price predictions, the Hsu et al. team also analyzed the length of the prediction horizon and a model’s profitability and accuracy. A short forecast horizon, in hours for example, may have more profitable results, especially in markets with high degrees of volatility. However, longer horizons (starting at one day), show increases in the accuracy of predictions (Hsu et al., 2016).

Because text-based approaches intend to accelerate the rate at which investors can glean

insights that may disseminate more slowly than financial data, this analysis uses two different forecast horizons to test this theory. The Day +1 forecast horizon predicts the performance of a company’s stock when there are only a few hours for the public to assimilate this information. The Day +5 forecast horizon provides a prediction of a stock’s performance after industry experts have had an opportunity to absorb the complex public disclosures.

Another intention here is to find the best machine learning prediction model possible based on vectorization of Word2Vec. However, more samples need to be incorporated to build more worthy NLP-based predictive models.

Business Understanding

Earnings calls are periodic (often quarterly) statements hosted by management of public companies to discuss the company’s financial performance with analysts and investors.

Traditionally, Financial market analysis focus on accounting and other hard data to extract signals. However, With advances in machine learning and NLP technology, decision-makers gradually use “soft” text-based documents like earnings call to interpret the hard data.  So, the information disclosed during the earning call is very important for the investors to make decisions. Furthermore, this project can be used potentially by financial institutions to analyze and predict the change of stock price and risk after the earnings calls.

Data Understanding

1. Earnings calls transcripts, total 113, collected from following companies in Nasdaq and time period from 2018~as of Aug. 2022

<https://seekingalpha.com/earnings/earnings-call-transcripts>

|  |  |  |  |
| --- | --- | --- | --- |
| company | number of transcripts | company | number of transcripts |
| Apple | 19 | Costco | 19 |
| Amgen | 18 | Microsoft | 20 |
| Amazon | 18 | Micron Technology | 19 |

2. Stock Market Data

https://www.kaggle.com/paultimothymooney/stock-market-data

Data Preparation

1. We join the earning calls dataset with stock price for each company for the time horizon we set.
2. In terms of data labeling, we use formulas calculated the date for the Day +1 and Day +5 by excluding Saturdays and Sundays, for which stock data would not be available. Lastly, the Label was calculated by comparing the Stock\_ratio and the Sector\_ratio for the two forecast horizons, as outlined below:

|  |  |
| --- | --- |
| Day +1 Stock Ratio | Open stock price the day after earnings call/  Open stock price on the day of the earnings call |
| Day +1 Sector Ratio | Open NASDAQ index price on the day after the earnings call/ Open NASDAQ index price on the day of the earnings call |
| Day +5 Stock Ratio | Open stock price the fifth business day after earnings call/ Open stock price on the day of the earnings call |
| Day +5 Sector Ratio | Open NASDAQ index price on the fifth business day after the earnings call/ Open NASDAQ index price on the day of the earnings call |
| Label | The label displays “1” if the Sector Ratio > Stock Ratio |

1. Text preprocessing:

Here, we use Python NLTK to implement following process

* remove punctuation
* remove numbers
* remove entities - companies
* remove stopwords
* implement word\_tokenize
* stemming and lemmatization

Modeling-Text Vectorization

**TF-IDF**

As we know, TF-IDF is simple and easy to use. It is simple to calculate, it is computationally cheap. However, TF-IDF cannot help carry semantic meaning, and it can suffer from memory-inefficiency since TF-IDF can suffer from the curse of dimensionality. We need to use SVD to address it.

**Word2Vec**

TF-IDF does not take into consideration the context of the words in the corpus whereas word2vec does. We use the Word2Vec representation of words to convert the above document term matrix to a smaller matrix, where the columns are the sum of the vectors for each word present in the document. Word2Vec approach is that the size of the embedding vector is very small. Each dimension in the embedding vector contains information about one aspect of the word. We do not need huge sparse vectors, unlike the bag of words and TF-IDF approaches.

Modeling: Machine Learning

**Logistic Regression**

Tf-Idf + Logistic regression is a very nice baseline for many tasks.

**SVM**

It is a fast and dependable algorithm and works well with fewer data. With their ability to generalize well in high dimensional feature spaces, SVMs eliminate the need for feature selection, making the application of text categorization easier. Another advantage of SVMs over the conventional methods is their robustness.

**Random Forest**

The Random Forest classifiers are suitable for dealing with the high dimensional noisy data in text classification. An RF model comprises a set of decision trees each of which is trained using random subsets of features.

**XGBoost:**

XGBoost is a scalable and highly accurate implementation of gradient boosting that pushes the limits of computing power for boosted tree algorithms. However, when you have very small training sets ( less than 100 training examples) or when the number of training examples is significantly smaller than the number of features being used for training.

Conclusion

In terms of sentiment analysis, although in Day+1 and Day +5, negative sentiment account for more percentage than positive sentiment, we can observe that after 5 day of the release of earnings calls transcripts, positive sentiment increases slightly. The business insight would be after the earnings calls, some concern and questions from the performance of the companies for the future quarters would decrease, leading the selling pressure to slow down.

The findings also indicate that text-based machine learning models have the potential to

improve the predictive capabilities of machine learning models to predict stock performance at the five-day forecast horizon. The average scores for each measure improved for the text-based models in the Day+5 predictions compared to the Day +1 predictions. This indicates that the text-based models may work by accelerating insights that are more slowly disseminated by traditional means.

KNN has the best performance compared to other models. Thus, we use the Word2Vec-KNN model to do the prediction with new cases which is included in our code.

Future Work

* More text data sources could also be included into future models.  e.g. SEC reports/news/ social media opinion from KOL(key opinion leader)
* Use Doc2vec for text vectorization and train the model
* (Doc2Vec is another widely used technique that creates an embedding of a document irrespective to its length. While Word2Vec computes a feature vector for every word in the corpus, Doc2Vec computes a feature vector for every document in the corpus.)
* Embed sentiment in model, we’ve considered to calculate the top similar word associated to the word ”product” or “customer” as the new features, and count the instances of each of these new target words for every record (topple) in the corpus to train the model.

Contribution:

Xiongxiang Fan(xf634)

**Nearly all of the coding. 70% of data collecting.**

1, stock\_price csv data collecting, NASDAQ\_HistoricalData csv data collecting.

2, stock price data cleaning, generating 'Day1' and 'Day5' labels' results.

3, all of the earning calls' texts cleaning.

4, union stock price labels with cleaning transcripts storing as 'complete\_union.csv'.

5, using word cloud, pie chart and bar chart for visualizing

6, TF-IDF, word2vec vectorizing

7, TF-IDF-SVM, TF-IDF-LogisticRegression, TF-IDF-Random\_Forest and TF-IDF-XGBoost modeling.

8, word2vec-LogisticRegression, word2vec-Naive\_Bayes, word2vec-KNN, word2vec-Logistic\_Regression and word2vec-Adaboost modeling.

9, new text data predicting function

10, README.txt

Chieh-Hsin Chen (cc7204)

1. Conducted paper and literature research

2. Constructed ideas for the whole project

3. Collected and prepared the whole dataset of earnings call transcripts

5. Programmed the code of SVM, XGBoost, Random Forest and Logistic Regression ML model for Word2Vec dataset.

6. Produced final presentation and final report.

Reference