Group: MCMC

Team Members:

Xiang Cheng

Ya Bo Li

Hao Nan Wang

Rong Zhou Li

Project Report

Abstract:

Accurately predicting stock price is a hot research topic among trading companies. Being able to predict stock price accurately not only helps company make money, but also helps to supervise the market and avoid potential economic crisis. Most of the prediction today are based on professional data such as portfolios. Our research tried to do the prediction based on public voice instead of these financial indicators. We pick AAPL (stock of apple) as our prediction target, and subreddit post as the feature data we used to extract people's opinion. different feature extraction methods like TF-IDF and Word2vec are performed on the feature data. The result feature vectors together with the stock status are feed into machine learning algorithm to train a logistic model . We tried out different tricks in feature extraction and gradually enhance our model. At the end, all improved model achieves an accuracy better than 50%, with highest accuracy 56.84%. The noise is very big, and is hard for us to say our model works well in predicting the stock price. However, the experiments did show some useful method that could be used to improve the accuracy of predicting stock price.

Introduction (Uniqueness of our research from the previous studies):

Earlier studies on stock market prediction based on public voice use Twitter Data or New Articles as the source. News always reflect what have happened on the stock market, therefore, not an ideal data source for predicting the future. In addition , because of the unpredictability in News and Twitter moments stock market prices follow a random walk pattern and cannot be predicted with more than 50% accuracy[1]. Unlike the previous study. we pick subreddit for apple and apple's product as our source. They are more targeted community, and the comment are more sentimental rather than just stating the fact. So it serves well as the source to extract public opinion about the company. what's more, most of the previous studies label each post as positive and negative by hand. This restrict the size of their source data, and its high cost of labor makes the prediction process not applicable to the industrial situation. In our research, we tried on merging all the post in one day, and run an unsupervised model on the merged text to automatically extract the sentiment feature. Though this may result in lower accuracy. **It is now possible to train model on a much larger dataset and makes the prediction process more automatic.(make more**

sense in real situation in industry) We also performed some original tricks on feature extraction, like set post from the previous day and label from the current day together as the learning data, and include the label from the previous day as the current day's feature.

Dataset of Interest

> Google Finance AAPL stock data from 1/1/2014 to 11/30/2017,
> All subreddit post from 1/1/2014 to 11/30/2017

With the advent of social media, the information about public feelings has become abundant. Social media is transforming like a perfect platform to share public emotions about any topic and has a significant impact on overall public opinion. Reddit is a social medial which allow people to share their opinions. Reddit text is short in length and can be quite opinionated, dense in information, dependent on the modelling of economic context and challenging to parse, due to the different vocabularies used (Sinha, 2014). In our project, we use Reddit comments to predict the predict trend of Apple Inc. Because Apple is a hardware company, its stock price and its core product have a tight connection. Reddit has subreddit which is useful feature we can take advantage of to filter the user comments of iPhone and iPad.

Data Collection

Framework: Spark

A total of over 250 gigabytes of compressed reddit comment history has been downloaded. The data covers the comment history of reddit from January 1st, 2014 to December 31st, 2017. To extract the data of interest, we used Spark to extract reddit comments from the subreddit r/apple only. We hypothesized that the public opinions of a company and its services would have a significant correlation with its stock price.

Daily stock price of Apple has been acquired from Google Finance. The acquired data covers the daily OHLC(open-high-low-close) statistics over a period from January 1st, 2014 to December 31st, 2017. We had originally planned to use the Python Yahoo Finance Api, but it had been deprecated in late 2017. Instead, Pandas DataReader Module was utilized to obtain Apple's stock price over our

desired time span. The stock price gave us a good approximation of Apple's business well-doing over the course of three years.

Data cleaning and preprocessing (Problems encountered and solution)

We had planned to implement our own crawler with PRAW which is an python package accessing reddit API. However, we ran into a limitation set by reddit which forbids collecting comment of a subreddit further than 25 pages. Therefore, we downloaded historical data of all subreddits in JSON format.

The size of all reddit comment history combined from 2014 to 2017 has exceeded the storage limit of the cluster, so we distributed the data download and extraction process between the team members. Because of sheer data volume and computationally heavy preprocessing, we have to perform most of the data processing tasks at midnight. However, it is still a tedious work due to the size. To facilitate this, a bash script has been written that automatically uploads and extracts data.

As mentioned, we need specific data, r/apple. During the uploading and extraction procedure, we use spark filtering data set by subreddit, which dramatically decreases the size of data set.

For further work, it is necessary to convert the time from timestamp to humanized date. The output of spark (part-000*) no longer has JSON syntax. Specifically, output file uses single-quoted structure, which cannot be recognized as JSON. Not to mention loading them with json.loads(). To solve this, instead of rewrite the whole files, changing single quotes to double quotes, we found a more useful package ast which can loads single-quoted content.

Sentiment Analysis:

type of knowledge extracted: people's opinion of Apple

Sentiment analysis was an important part of our project. Since text information cannot be fitted into machine learning model, we need to convert Reddit comments into vector. As stated before, previous research focus on extract sentiment index from text information, but those work usually try to count the positive and negative words in a day. This cause inaccurate problem, since there is not a weight for each comment. In this project, we use two methods, TF-IDF and Word2vec. Those two methods can take the weight into account when words

be converted to vector. TF-IDF (term frequency–inverse document frequency) is an effective method to convert a words to an vector which reflect how important a word is in a documentation[4]. Word2vec representation is better and a recent technique which map words to low dimension.[3] Word2vec not only map each words to unique vector, but also take the relative relationship of words into account.

Model Training

This research is not focused on model selecting, so we used logistic regression which are commonly used in training features from text. we did some experiment at our base setting (see next part) and select reg = 0.01 as our parameters

Experiments & Improving the model

We conduct three experiment each contains two or more models. Both model from the same experiment are under control and only different on a specific step. The model which performs better in the previous test will be the base setting of the next model.

Base setting

After cleaning, each row of the data will be a tuple, (merged text, label). All models start with common word filtering on the merged text, and go through a feature extraction model (either Word2vec or TF-IDF). After that, the type of the data become (Dense vector, label). We take the dense vector as feature and trained on a logistic model.

First Experiment

Different feature extraction model(Word2vec VS. TF-IDF)

Second Experiment: Introducing delay factor

Base on the consideration that It may take time for people to see the twitter post and take action on the stock market, we come up with the ideal to introduce the delay factor. In model with delay factor n, we will pair up the feature and the label after n days. To do this we first sort the data by time, and index each row. Then, we transform features and label to two rdd, minus n from the index of the feature, filter out negative index and do a natural join. We tried on n = 0,1,2,3 in our experiment

Third Experiment: Include the stock status of the previous day in feature

The previous performance of a stock also influence people's choice on stock market. By considering this, we take in the difference between closing and opening price as a feature and compare the result with the model without considering this.

Result

Below are the result of each experiment and our analysis. To evaluate the data, we split the original dataset into the training set(80%) and the test set(20%). The training was performed on the training set, and the model trained will be used to do the prediction on the test set. and the accuracy of this prediction will be used to evaluate the performance.

Experiment 1

setting:

Fit in same input dataset without time delay to Word2vec and TF-IDF model.

hypothesis:

Word2vec has better performance than TF-IDF.

accuracy:

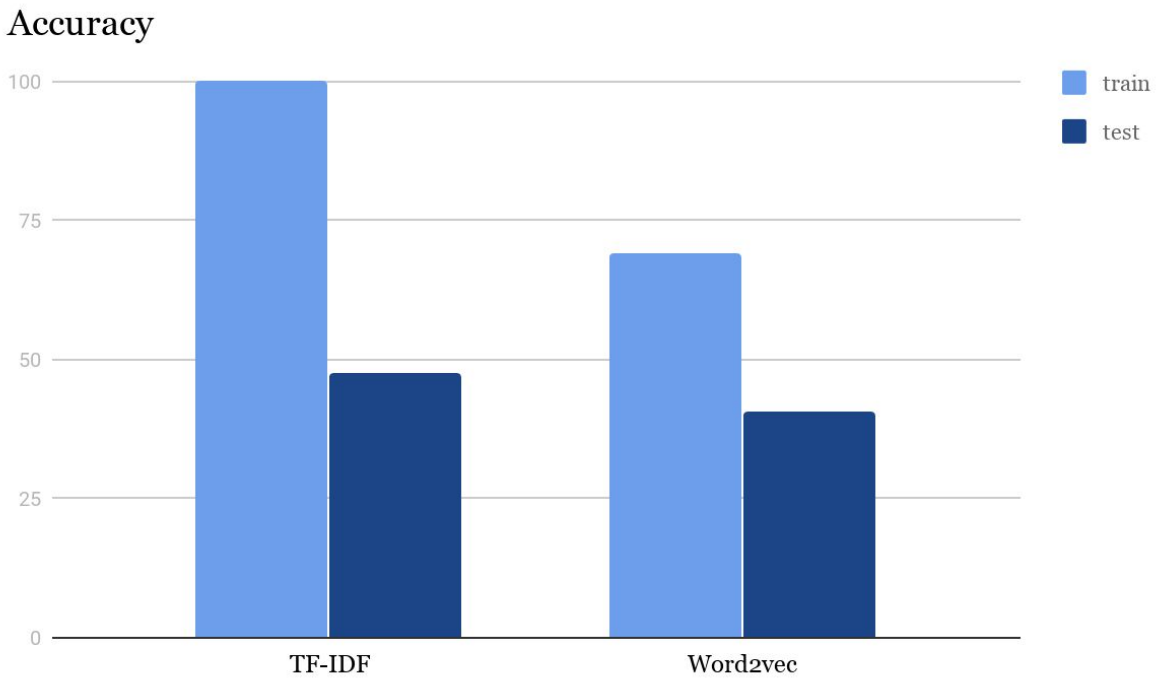
TF-IDF

training data: 99.99% testing data: 47.75%

Word2Vec

training data: 69.28% testing data: 40.83%

visualization:



analysis:

The result of experiment 1 is different from our expectation. We take a closer look at the two model. We think the reason is as following:

1. TF-IDF can filters out nonsense words during the Inverse Document Frequency stage. This feature is useful to denoise the Reddit dataset.
2. The relation and co-occurrence between words does not as important as we expected for stock price prediction.

experiment 2

setting:

Use Word2vec model to extract 2000 dimensions feature from dataset with delay factor and cluster those vectors with logistic model. Set the delay to 0 day, 1 day, 2 days and 3 days.

hypothesis:

The posts on Reddit require some time to be reviewed and reflected on the stock market.

accuracy:

Delay by 0 day:

training data:69.28% testing data:40.83%

Delay by 1 day:

training data: 67.32% testing data: 52.06%

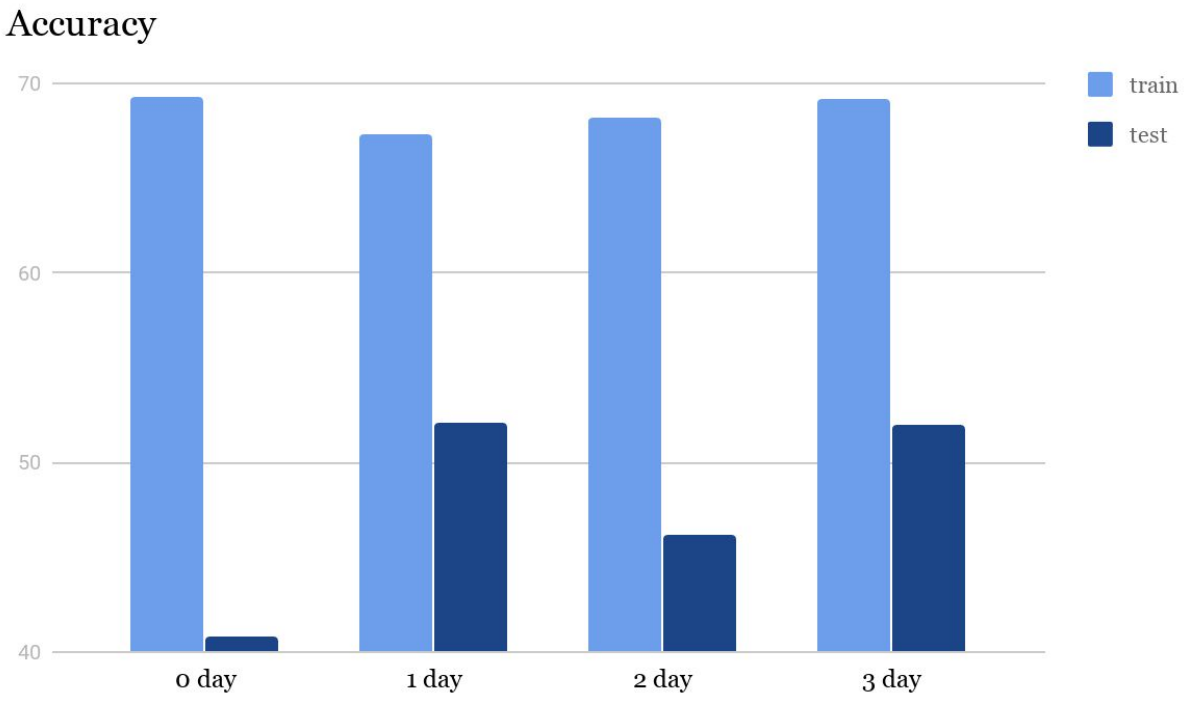
Delay by 2 day:

training data: 68.14% testing data: 46.17%

Delay by 3 day:

training data: 69.17% testing data: 51.94%

visualization:



Analysis:

The result of experiment 2 is as our expectation. The prediction accuracy is increasing when delay factor included in feature space. But there is not a obvious relation between the delay time and the prediction accuracy.

Experiment 3:

setting:

Include the stock price difference of the last trading day in feature space. Use Word2vec model extract 2000 dimension features and logistic model to cluster.

hypothesis:

The increase or decrease of stock price from previous trading day could influence the trader mood and the price trend of next trading day.

Accuracy:

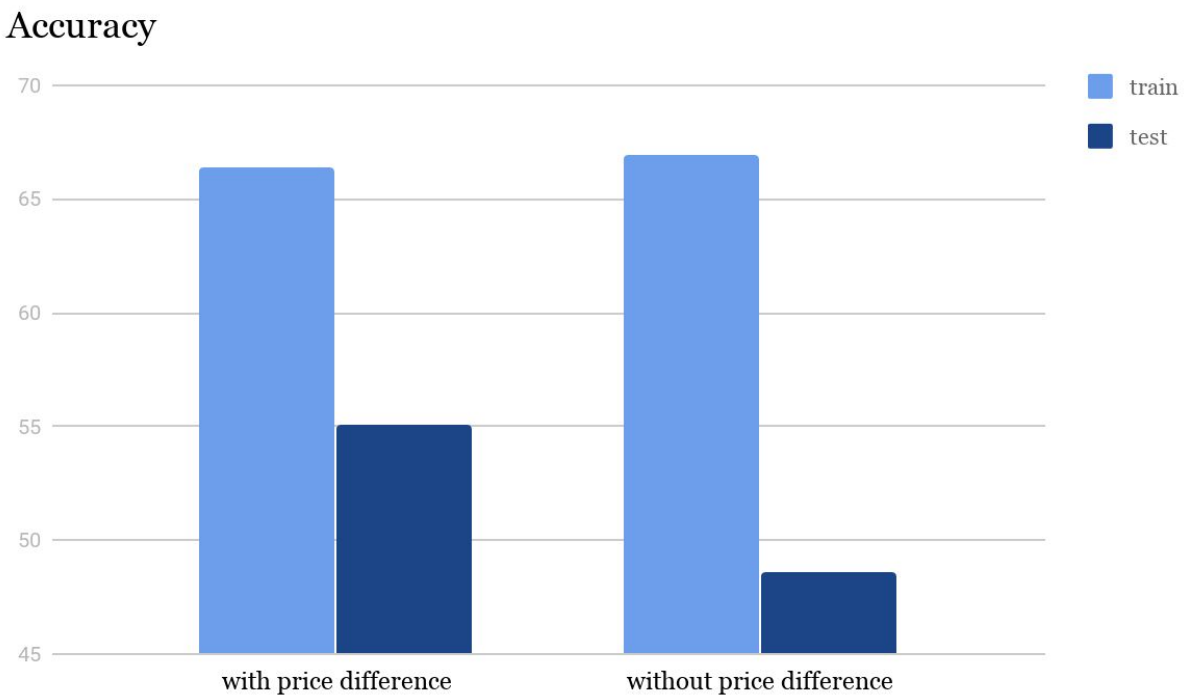
with price difference:

training data: 66.41% testing data: 55.06%

without price difference:

training data: 66.95% testing data: 48.58%

visualization:



Analysis:

The result of experiment 3 is consistent with our hypothesis. The prediction accuracy is increased after included price difference into feature space.

Conclusion

Based on the result of the experiments, the following method could be consider when improving the accuracy of stock prediction model:

- Filter out non-sense
- Introduce a delay factor of 1
- Include previous stock status as part of the feature

We also come up with some reasons why our models are not performing well:

- The high randomness of stock price: predicting the stock price using machine learning may not be a good choice
- naive machine learning model: the model supported in pyspark is limited. A better result is expected to get by using RNN/LSTM (can take the relation between day and day into consideration).
- Public voice still small factor of the change in stock market: according to Fama–French three-factor model, the three main factor of stock returns are "(1) market risk, (2) the outperformance of small versus big companies, and (3) the outperformance of high book/market versus small book/market companies"[5].

These are some thought that future research could consider.

REFERENCE:

[1]Qian, Bo, Rasheed, Khaled, Stock market prediction with multi- ple classifiers, Applied Intelligence 26 (February (1)) (2007) 2533, http://dx.doi.org/10.1007/s10489-006-0001-7.

[2]International Journal of Recent Engineering Research and Development (IJRERD) Volume No. 01 – Issue No. 03, ISSN: 2455-8761
www.ijrerd.com, PP. 01-04 Application of Sentiment Analysis in Stock Markets Abhishek Chander NV[1], CH.Vanipriya[2]

[3]C. Dougal, J. Engelberg, Garc ´ıa, and C. A. Parsons, "Journalists and the stock market," *The Review of Financial Studies*, vol. 25, no. 3, 2012.

[4] https://en.wikipedia.org/wiki/Tf–idf

[5] https://en.wikipedia.org/wiki/Fama–French_three-factor_model