Git Hub: <https://github.com/gechang1996/CSE482>

**Stock Market Prediction**

**Chang Ge**

**ABSTARCT**

1. For this project, I used 2000G one-year given twitter data to get the information I want though Jupyter. For each twitter data file, I searched key words about apple, amazon, google, and Facebook, so I can collect the sentences for checking the polarity of each day. I passed though those sentences into opinion finder to get polarity of each day. Finally, I use sklearn to predict the stock price. Unfortunately, since the data is too small (only one year), it seems that the accuracy is very low. And it is apparent that using historical data alone has better accuracy.

**1.INTRODUCTION**

**Goal:**

As we all know, stock prediction become more and more popular in the world, since if we can predict a stock, then we can make money. So, I decide to make stock prediction be my project.

I am going to predict the future stock price by sentiment analysis on twitter data. I will collect sentiment analysis on each tweet by some special words to decide if the stock market is Positive Or Negative(I can use a percentage to decide the level of stock market situation, eg:100% is very Positive which means all of the people believe the stock market will be better and there stock price will get higher, otherwise, 0% means nobody trust the stock market). It’s kind of relate the people’s idea with stock price. Then the processed data will be to apply sklearn method on a combination of historical price and twitter sentiment data and evaluate whether the prediction results are better than using historical data alone

**Challenge:**

1:There are about 2000G data I need to process, so I need to separate them and process them.

2:When I was using Opinion Finder, I found that if a single file is too large, then the program will be stuck, and I must rerun it alone.

**Data Processing difficulty:**

After I processed the 2000 G date, first I got all tweets contained key words about google, face book, amazon and apple. Then I passed them through opinion finder and collected the outputs. After all of these, I got four files each contains date and its polarity (e.g. 1/1/2000,neg). For all of that, I spent over 80 hours(Downloading data, processing them in Jupyter, analyzing in Jupyter). I only use my laptop, a single machine. I used Opinion Finder .

1. **DATA**

The row data is a 2000G tweets data. Each file contains tweet data in a time period (e.g. 10 sec, 2 hours etc.). The data was passed through Jupyter and an output which contains data and tweets with keywords (e.g. “google”,”face book”, etc.) showed up. And there are about more than 1000 outputs: each contains one day tweets with keywords about each company.

For figure 1 to figure 2, I am showing the process by opinion finder. The figure 3 shows the result of polarity of one day.



Figure 1

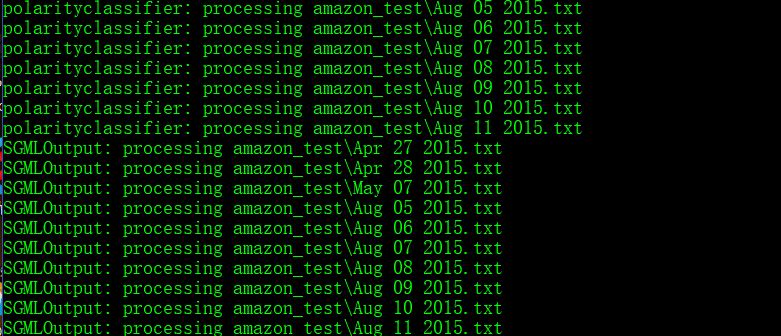
****

Figure 2

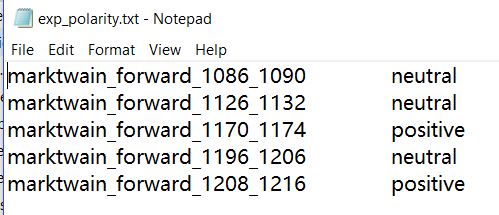
****

Figure 3

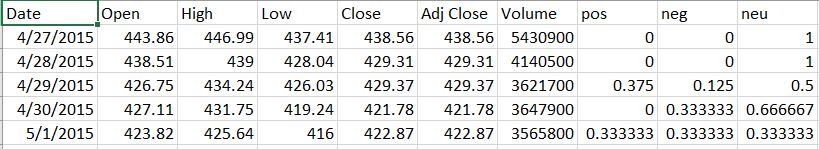


Figure 4

Figure 4 shows the excel file sample I was using for predicting the stock market. For each line, there are pos, neg, neu representing the polarity of each day. The summation of those three are 1.(e.g. If there are 3 positive results,4 negative results, 3 neutral results, then pos is 3/10, neg is 4/10, neu is 3/10). By the polarity result, I start predicting the stock price.

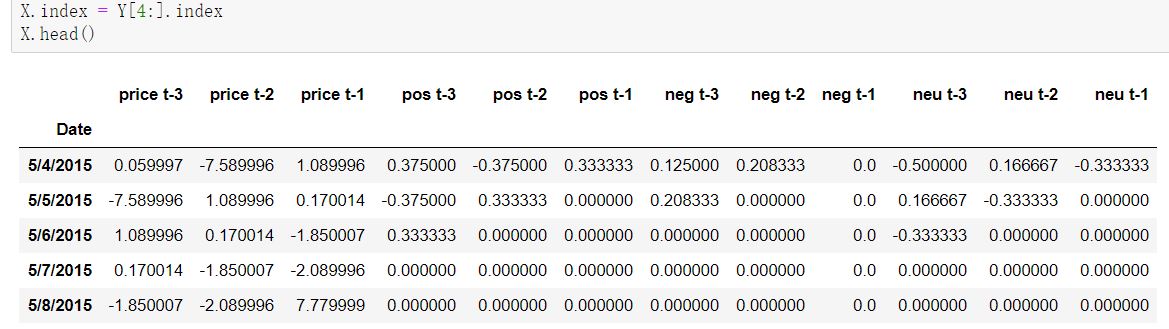


Figure 5

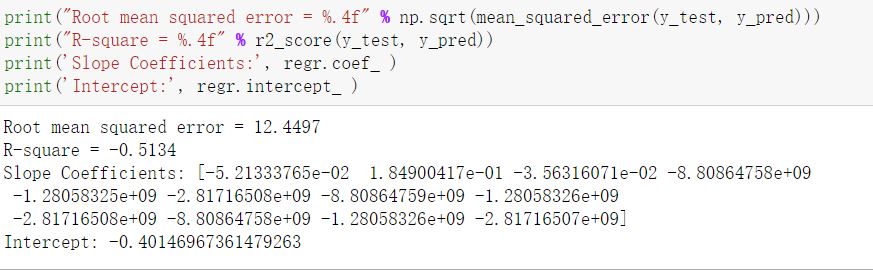
Just like t-1,t-2,t-3 for stock price, I also get the t value for polarity and apply them into linear model from sklearn. The target attribute is “Close Price for today minus Close Price for previous day”, and the predictor attributes used are price, neg, pos, neu t-1,t-2,t-3.

Figure 6

1. **METHODOLOGY**

I used linear predictive modeling. For the training data and testing data, I both used half of the processed data.

Here are the codes for processing the data

* project1-collecting.ipynb: Filter the 2000G by key words about amazon, google, Facebook, apple. Get the output files(e.g. (date, sentiment sentences))
* project1-processing1.ipynb: get the paths for each file from project1-collecting.ipynb which will be used in opinion finder.
* project1-processing2.ipynb: filter all the result files from opinion finder. And get the output files(e.g. (date, pos, neg, neu)).
* project1-processing3.ipynb: combine what we get in project1-processing2.ipynb with stock price csv files. Get the outputs which will be used for data analyze.
* project1-analyze.ipynb: use sklearn model to analyze the results. Separate the data half in train data and half in test data.
* project1-analyze\_baseline.ipynb: uses heuristics, simple summary statistics to predict the stock price.

1. **EXPERIMENTAL EVALUATION**
   1. **Experimental Setup**

**1: I use windows OS and Jupyter for this project. No AWS cluster.**

**2: For the following results, the baseline one only uses the stock price to predict the stock price ups and down. My results use both polarity of each day and stock price to predict stock price ups and down.**

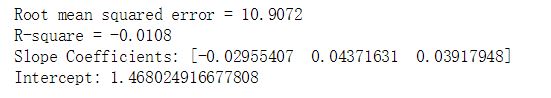
**3: I used root mean squared error to predict the stock price.**

* 1. **Experimental Results**

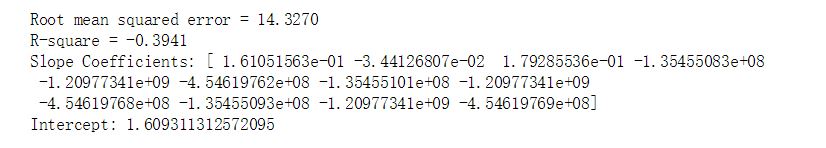
**Since Root mean squared error is better for smaller result and R-square is better for larger result, unfortunately, the project is not successful, and I’ll put the critical thinking about it later in the section “Critical Thinking”.**

**Amazon Result:**

**Baseline**

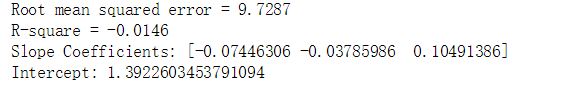
****

**My Result**

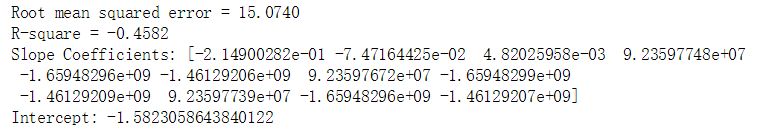
****

**Google Result:**

**Baseline**

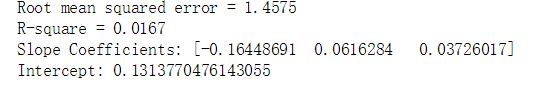
****

**My Result**

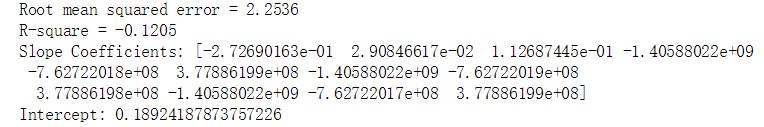
****

**Facebook Result**

**Baseline**

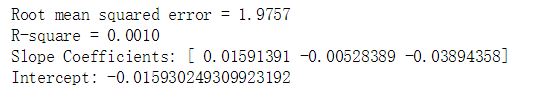
****

**My Result**

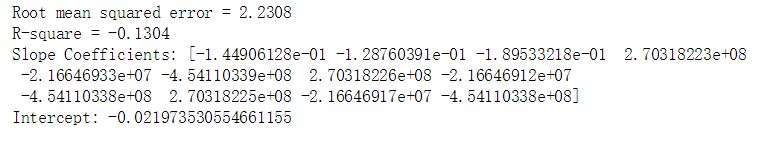
****

**Apple Result**

**Baseline**

****

**My Result**



1. **CONCLUSIONS**The project is not successful : Using price alone to predict stock price is more accurate than using both polarity and stock price. Here is the reason I think why the project fails.

**Critical Thinking:**

(1): There are only one-year data, so the result is likely to be inaccurate. The model cannot be applied to the whole stock market in all ages.

(2): When I processed data, what I did was decide the polarity of one day. But I think I also need to divide workdays and weekends, since the stock market only open in workdays. And of course, the stock market will close due to festival, So I need to discern those days as well. Last but not least, there are three parts in a workday and for each part we can check its polarity:

* Time before the stock market open.(first part)
* Time when the stock market open.(second part)
* Time after the stock market open.(third part)

The reason why we need to divide one day into the three parts is:

* The Open price of a stock is affected by the polarity of time duration from previous stock market closing to this stock market opening.(e.g. If the stock market closed at 1/1/2000 4:30 AEST and it reopen at 1/2/2000 9:30 AEST, then we need to check the polarity between this time duration.) It could be calculated by the first part and third part.
* The most important part is the second part. If we can use real-time monitoring into it and check the polarity of each second, we may predict the future price of a stock.

1. **REFERENCES**

[1] Alice Zheng ,Stanford University, *Using AI to Make Predictions on Stock Market,* 2017

[2] Bonde, Ganesh, and Rasheed Khaled. “*Extracting the best features for predicting stock prices using machine learning*.” Proceedings on the International Conference on Artificial Intelligence (ICAI). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012.

[3] Shen, Shunrong, Haomiao Jiang, and Tongda Zhang. “*Stock market forecasting using machine learning algorithms*.” Department of Electrical Engineering, Stanford University, Stanford, CA (2012): 1-5.

[4] Chen, Kai, Yi Zhou, and Fangyan Dai. “A LSTM-based method for stock returns prediction: A case study of China stock market.” *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015.