Stock Predict

Tengjiao Wang

Project URL: https://github.com/wangtengjiao1997/cse482.git

**ABSTRACT**

The goal of the whole project is to develop a accurate predictive modeling with GDP, Dow Jones index, Nasdaq index, exchange rate to predict will the stock increase or decrease at one month later. This modeling used classification to predict and I got about 69% accuracy as the results.

# INTRODUCTION

The problem I’m investigating is the stock price. I’m creating a model to predict the will the stocks’ price increase or decrease at one month later. As more and more people trying to use stock to earn some part-time money and stock is also a very important part in financial world, so this predicting would be very helpful in people’s life. I’m using classification to get the result and I’m using the Apple Inc’s stock as the test sample in my project. The biggest challenges I met in the project is processing the data, because there are a lot of bad data. They have some empty data or some strange characters and I can’t do calculus in this kind of data. So, I spent a lot of time to clear the not useful data and some bad characters.

# DATA

I download data online and did some adjust and merge some data together by my own. I collected the following data:

U.S/Euro exchange rate :

https://fred.stlouisfed.org/series/DEXUSEU

GDP in California:

<https://fred.stlouisfed.org/series/CARGSP>

Dow Jones Industrial Average:

<https://fred.stlouisfed.org/series/DJIA>

NASDAQ 100 Index:

<https://fred.stlouisfed.org/series/NASDAQ100>

The Apple Inc stock:

https://finance.yahoo.com/quote/AAPL/history?p=AAPL

These data are all CSV data. I collected them from the website above. I merged them by date, but the GDP is annual data and others are daily data. So, I did some adjustment to change the GDP’s data to daily with same value in the whole year.

The data I collected is time series data, they have different time range. But the final data I merged is from 2009-04-23 to 2017-11-15. The data of stock has some unnecessary data like high price, low price which is not helpful, so I discard them. Then I found that some data in the exchange rate is only ‘.’ Character, so I have to clear this character and finally merge them together.

In my preprocessing steps, I replace the missing values with the value of its adjacent data. I also make the GDP’s data to daily by adding days in the date column and make the value is the same at the whole year, because the data of GDP is annual. I must make it to be daily, then I can merge it with other data. After I merged all the data I collected. I created a new dataframe with the same data that I collected before and adding a new column ‘Predict’ which is about will the stock increase or decrease as the class. I use the stock’s price at 30 days later minus the stock’s price at now to get will the stock increase or decrease at 30 days later and add it to the ‘Predict’ column. Therefore, I can use classification to predict the price.

My table has these attribute: Date, NASDAQ100 index, Dow Jones Industrial Average, GDP in California, U.S/Euro exchange rate, the stock’s close price and volume, and the prediction. This table has 2159 rows and 8 columns.

# METHODOLOGY

This is the prediction of stocks. So I collected the data that are most influent the stock price. The GDP most directly reflect how’s economy of that state. The exchange rate reflect the international economy of the country. The Dow Jones Index shows the mean of the price of stock. NASDAQ index reflect the change of the price of stocks. These are the most important factors that will reflect the stocks price change. And I’m using decision tree to make the classification. I create the model with test\_size = 0.8.

I write all my code in one file : processing.ipynb. It include I collect data from the file I download. Then I make some adjustment to discard missing value and strange character. After that, I merge all data together and create class ‘predict’. In the end, I use decision tree to predict will the stock price increase or decrease and show the accuracy.

# EXPERIMENTAL EVALUATION

## Experimental Setup

I’m using jupyter notebook to write all my code. I’m using the accuracy as the result to evaluate my project.

## Experimental Results

I’m using classification to do the prediction and get the accuracy to evaluate if the project is successful. I got 69% accuracy in the project and I think this accuracy is pretty good. I also tested which factor will influent the stock price most by using decision tree to test each factor one by one and I got NASDAQ100 index is the factor that influent price most which has 68% accuracy.

# CONCLUSIONS

In this project, I found that data is the most important thing in prediction. If I can get more data that will influent the prediction or the more detailed data, the accuracy will be higher.

# REFERENCES (at least 3 references)

1. “Apple Inc. (AAPL) Stock Historical Prices & Data.” Yahoo! Finance, Yahoo!, 28 Apr. 2019, finance.yahoo.com/quote/AAPL/history?p=AAPL.
2. “Real Total Gross Domestic Product for California.” FRED, 19 Nov. 2018, fred.stlouisfed.org/series/CARGSP.
3. “Dow Jones Industrial Average.” FRED, 27 Apr. 2019, fred.stlouisfed.org/series/DJIA.
4. “NASDAQ 100 Index.” FRED, 26 Apr. 2019, fred.stlouisfed.org/series/NASDAQ100.

Grading criteria

Note that the project accounts for 10% of your final grade. The project will be graded based on the following criteria:

1. Presentation - structure/organization and clarity of writing (including tables and figures).
2. Technical - Correctness and thoroughness of the analysis performed. What are the challenges faced and how well did you address them? How do you evaluate the performance of the method you'd applied to the data? How much detailed discussion you provide to explain the results you'd obtained (e.g., discussion about why the method works or didn't work on the data)?
3. Difficulty level - How large is the dataset used? How much effort you had to spend to collect, integrate, preprocess, and analyze the data? Are you implementing the project on a cluster or a single machine? What tools did you use (do you have to implement them or are you simply using existing libraries)?