Big Data Analytics Final Report Template

Authors Name/s per 1st Affiliation *(Author)Jiahao Li jl4930, Suyang Xu sx2212, Yue Zhang yz3190*
Electrical Engineering
Columbia University
jl4930@columbia.edu

Abstract*—*Stock market prodiction has always been a hot topic in many researchs and experiments. But until now, no one could give out a model which could give satisfying prediction with high accuracy. The one reason is the stock market is somehow irrational and sometimes there is no correlation between the data in history with the price in future. But, what we could do is to make our model become more accurate and credible than the model before. So our project in course Big Data Analysis is to use the data of the financial statement of stocks in the first 5 years and give the prediction of the 6th year. We build a deep learning model(2 layer MLP) with tensorflow in python and we find this model could give people relevantly valuable advice about which stock is worth to choose.

Keywords: Stock Market Prediction, Python, Tensorflow, deep learning, MLP, finance

INTRODUCTION

Although the stock market prediction is a frequent topic in many researches and experiments, there is still a large space for the prediction model to improve. So we want to challenge this task and manage to make a more accurate and credible model for the people who invest in stock market. Our data set source is the financial statement of the stock market of Shanghai and Shenzhen in China.

RELATED WORKS

In paper Machine Learning for Financial Market Prediction by Tristan Fletcher. The machine learning model he use is SVM and he prove reasons that researcher could use machine learning as method to derive stock prediction.

In paper Applying Machine Learning to Stock Market Trading by Bryce Taylor, the learning model he use is Naïve Bayes model.

But the question is: we find those research choose only the price of a stock as their input data. So we decide to make an innovation, based on the Book Security Analysis writen by Benjamin Graham, we know that there is relevently strong correlation between the data on financial statement. So we decide to use the core feature on financial statement as our input data.

SYSTEM OVERVIEW

We don't need to gather all the data from the financial statement because we don't need the detailed information about an company. Based on our economic knowledge, I decided to use those features on financial statement: Earnings per share, net profit, operating profit, net asset, debt of a company. we write Python script to collect data of financial statement from Netease (163.com). After normalize and processing all those data, we put them into our deep learning model. (2 layer MLP). In the section below, we will give you detail about how we normalize and process our data.

ALGORITHM

We use python to code and jupyter notebook as our platform, we use tensorflow to build our deep learning model.

1. data collecting

In data collection part, we use some Python package to help us collect data from website. Two main packages that we used are 'requests' and 'BuautifulSoup4'.

We use 'requests' to fetch HTML files of websites that we need. Then we can sparse the HTML file and extract useful information from the source file. Because nearly all stocks' information are included in the source file, we can solve the task in this way. While in some other cases, needed information may be contained in HTML element like javascript or Flash.

In this case, we have good luck thus do not need to do more effort.

With function defined above, we can implement data collection procedure. We get data from profit report, debt report and history stock price, then write these into csv file. Then we finished the data collection part.

2 data processing

The structure of the input dataset would be introduced in this part.

As we have mentioned, the 5 years of a certain feature is a group, for example, we gather the net profit of a given

stock 5 times, from Dec/31/2011 to Dec/31/2015, the data in the last day of a year.

2015	2014	2013	2012	2011
129,457	115,685	95,054	79,111	73,035

Figure 1 Net profit of consecutive 5 years.

But then, we would meet a problem, the dataset we collect from the financial statement is usually not in the same magnitude. Obviously, we would not want to put those 2 features directly into the model because the model will "think" the second set of data is not important and they will almost not influence the model. so we have to normalize them firstly.

2015	2014	2013	2012	2011
4,174,000	4,145,400	3,971,700	3,138,500	3,084,400

Figure 2 Net profit of consecutive 5 years.

2015	2014	2013	2012	2011
0.15	0.14	0.13	0.12	0.12

Figure 3 Earnings per share of 5 consecutive years.

We have 2 ways to normalize them. The first way is to use the net profit of every year to devide the net profit of the first year(2011) and mutiply normalization factor (0.25 in here) , and then, the data after processing would be like this.

0.338315	0.335997	0.321918	0.254385	0.25

Figure 4 Net Profit after Normalization method 1

$$NPn_x = NP_x / NP_1 * 0.25$$

The other way is to use the net profit of a given year to devide the total asset of a company at that year. The data after normalization would be like this.

0.081487	0.100159	0.109077	0.106033	0.111516

Figure 5 Net Profit after Normalization method 2

31,968,600	26,734,600	23,072,500	20,308,600	17,878,100

Figure 6 Net asset

480,260,600	387,146,900	341,046,800	275,685,300	258,710,000

Figure 7 Debt

$$NPn_x = NP_x / (NA_x + D_x)$$

Note: the subscript of a variable represent year. NP denotes net profit, NA denotes net asset and D denotes debt.

The data in financial statement will be saved in csv file and the data will be posed in the following structure.

In the table below, in column 0, we could visit the price of all the stock in csv file in the year 2015(Dec/31) , in column 1, we could visit the price of the all the stock in the year 2014(Dec/31)…

In the column 5, we could visit the Earnings Per Share of all the stocks in the year 2015(Dec/31).

column	content	year
0-4	Earnings Per Share	2015-2011
5-9	Net Profits	2015-2011
10-14	Operating Profits	2015-2011
15-19	Net Asset	2015-2011
20-24	Debt	2015-2011
25-29	Stock Price	2015-2011
30	Stock Price	2016

Table 1 csv file structure

And we could also give out a section of our coding implementation.

```
NetProfitFirstYearRatio = []
      for i in range(num):
            temp = []
            firstYearNetProfit = data[i][9]
            for j in range(5,10):

      temp.append( data[i][j]/firstYearNetProfit*0.25)
            NetProfitFirstYearRatio.append(temp)
```

We could not get label of a stock directly from the stock, so we have to calculate labels by ourselves.

SOFTWARE DESIGN FLOW AND PSEUDO CODE

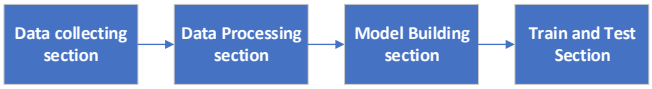

Figure 8 top level software design

Here is pseudo code of the process:

1. Data Collecting

In data collection part, we use Jupyter notebook to run our Python script. We first import 'requests' and 'BuautifulSoup4' package. Second, we run script to get information from profit and debt report.

```
# Collect data from financial report
#
def  getFinancialStatement(stock number range):
    finSt = []
    for i in stock number range:
        re = requests html file from website of stock i
        sp = bs4(re)
        co = sp.findClass(classname)
        if cannot find class classname:
            continue
        row = []
        for k in co:
            row.append(k)
        finSt.append(row)
    return finSt
```

After that, we collect stock price based on the valid stock number got from previous processing.

```
# Collect history stock price
#
# Because history data is on different webpages of different
# year, we need to visit websites multiple times for single
# stock

def  getStockPrice(stock number range):
    stkPr = []
    for i in stock number range:
        row = []
        flag = False
        for year in year range:
            re =
            requests html file from website of stock I of year
           sp = bs4(re)
            co = sp.findClass(classname)
            k = sp.findClass(priceclassname)
            row.append(k)
            if cannot find class classname:
                flag = True
               break
        if flag:
            continue
        else:
            stkPr.append(row)
    return stkPr
```

2. data processing

import data from data.csv file and filter unrelevant string and do type transformation.

```
data = []
temp = []

for line in data.csv:
temp = []
    for i in split line:
        i.transform into float and filter unrelevant str
        temp.append(i)
    data.append(temp)
```

After transform all those data from string to np.array, we have to process and normalize them before putting them into the model.

```
def normalization(data):
for i in data:  #we select normalization factor by ourselves
according to specific data
      Earnings per share = data[i][0:4]

      Net Profit After Normalization 1 = net profit / (net
asset+debt)*normalization factor

      Operating Profit After Normalization 1 = operating
profit /(net asset + debt)*normalization factor

      Net Asset After Normalization 1= Net Asset /(Net Asset
+ Debt)

      Net Profit After Normalization 2 = [net profit 2015,
net profit 2014, net profit 2013, net profit 2012, net
profit 2011] /net profit 2011 * normalization factor

      Operating Profit After Normalization 2 = [Operating
profit 2015, Operating profit 2014, Operating profit 2013,
Operating profit 2012, Operating profit 2011]/Operating
profit 2011 * normalization factor

      Net Asset After Normalization 2 = [Net Asset 2015, Net
Asset 2014, Net Asset 2013, Net Asset 2012, Net Asset 2011]
/ Net Asset 2011 * normalization factor

      Debt After Normalization 1 = [Debt 2015, Debt 2014,
Debt 2013, Debt 2012, Debt 2011] / Debt 2011 * normalization
factor

    Price After Normalization = [Price 2015, Price 2014,
Price 2013, Price 2012, Price 2011] / Price 2011 *
normalization factor
      Judge Net Profit Increase = judgeIncrease() #if the
net profit was increasing consecutively in the 5 year return
1, else return 0

      Judge Operating Profit Increase = judgeIncrease() #if
the operating profit was increasing consecutively in the 5
year return 1, else return 0

      company size = calculate net asset + debt and return
[0,0.2,0.4,0.6,0.8] for 0.8 the biggest company and 0 the
smallest

    return [Earnings per share ,Net Profit After
Normalization 1 ,Operating Profit After Normalization 1 ,Net
Asset After Normalization 1 ,Net Profit After Normalization
2 ,Operating Profit After Normalization 2 ,Net Asset After
Normalization 2 ,Debt After Normalization 1 ,Price After
Normalization ,Judge Net Profit Increase ,Judge Operating
Profit Increase ,company size]
```

We define label by ourselves and generate Label. The stock was divided into 4 type, excellent stock, with label 3, good stock, with label 2, ordinary stock, with label 1, bad stock, with label 0.

```
def generate label:
    max_price = max{p5, p4, p3, p2, p1} #p5 denote the price
in 2015
    min_price = min{p5, p4, p3, p2, p1}
    mean_price = mean{p5, p4, p3, p2, p1}
    if p6 > max_price + (max_price - min_price)/4:
        y = 3 #excellent stock
    if  max_price < p6 <max_price + (max_price - min_price)
        y = 2 #good stock
    if  mean_price < p6 < max_price
        y = 1 #ordinary stock
    if  min_price < p6 < mean_price
        y = 0 #bad stock
```

3. Model Construction and Training

We use tensorflow to build our model, here is the pseudo code. We use L2 regularization to prevent overfitting and Adam gradient descent as our optimizer.

```
h = relu(w1 * x + b1) #20 hidden units, we set 20 based on
experiment result
prediction = w2 * h + b2 #4 output units

loss = softmax cross entropy (y_tf, one_hot(y))
loss = loss + l2 regularization

correct = equal number (argmax(prediction), y)
accuracy = mean (correct)
```

Training

```
5000 iteration
Use adam optimizer
Learning rate 1e -2
```

EXPERIMENT RESULTS

Describe the experiment results of your algorithm. Show how did you evaluate the performance of your algorithm.

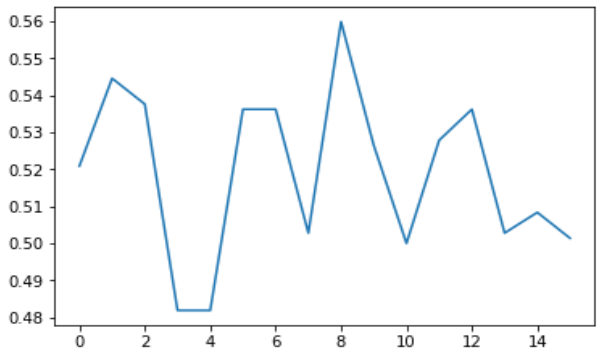

Figure 9 multi-times Test Accuracy

```
accuracy_total = np.array(accuracy_total)
print(np.mean(accuracy_total))
```

```
0.519063
```

Figure 10 average Test Accuracy

Using the model we generate, we could give the prediction of a model of a certain stock. For example, We choose "贵州茅台" (gui zhou mao tao, a wine company in China), the model will judge that it is an excellent stock

.

```
#input a stock number you want to test
stock = '600519'
for i in range(np.shape(data_string)[0]):
    if (data_string[i][0] == stock):
        print(data_string[i][0],data_string[i][1])
        break

data_select = abstractFeature( changeStringToFloat([data_string[i]]))
label_select,_ = generateLabel(data_select)
#print(label_select)

print(sess.run(tf.argmax(h_tf,1), feed_dict={x_tf: data_select, y_tf: label_select}))
pred = sess.run(tf.argmax(h_tf,1), feed_dict={x_tf: data_select, y_tf: label_select})
index = ['bad','ordinary','good','excellent']
print(data_string[i][1]+" is "+ index[int(pred)]+" stock.")
```

```
600519 贵州茅台
[3]
贵州茅台 is excellent stock.
```

Figure 11 Given Stock prediction

CONCLUSION

We define 4 labels so the random prediction accuracy would be 25%, but our model could reach almost 52%. So our stock market prediction model could give valuable reference to people who invest in stock market in China

.

REFERENCES

1. Tristan Fletcher, University College Machine Learning for Financial Market Prediction ,Computer Science, University College London

2. Vatsal H. Shah, Machine Learning Techniques for Stock Prediction

3. Yuqing Dai, Yuning Zhang Machine Learning in Stock Price Trend Forecasting, standford university

4. Gregory Provan, Stock Market Prediction

5. Saahil Madge Predicting Stock Price Direction using Support Vector Machines, Independent Work Report Spring 2015

6. Bryce Taylor, Applying Machine Learning to Stock Market Trading

APPENDIX

Individual student contributions – table

	CUID1 jl4930	CUID2 Sx2212	CUID Yz3190
Last Name	Jiahao Li	Suyang Xu	Yue Zhang
Fraction of (useful) total contribution	37.5%	37.5%	25%
What I did 1	Project Design	Data collection and processing	Presentation slide
What I did 2	Data Processing and Model construction	Report writing	Topic selection and discussion
What I did 3	Train model and adjust parameter	Video recording	Process video
What I did 4	Report, ppt		