Analyzing S&P 500 Stock Data:
A Data Mining Approach

Abstract

Investigating stock has become an approach that helps people increase their income. With the development of technology, more and more tools, software and techniques had grown to assist both professional experts and investors in analyzing the stock market trend. If a technique can provide a strong predictive model for the current stock market, it will significantly help users to predict the upcoming trend and maximize the profits. However, the amount of financial data and the stock trend are seemingly endless, and they are changing all the time. It could be extremely hard to come up with a large and well-structured dataset on a wide array of companies. The user Cam Nugent in Kaggle data website has posted a dataset with past 5 years historical stock prices for all companies currently found on the S&P 500 index[1]. Generally, the goal of this report is to utilize the dataset that provided by Cam Nugent to come up a reasonable prediction.

Contents

1 Introduction

The S&P 500 is the stock market index that tracks the stock of 500 large-cap US companies. It represents the stock market's performance by reporting the risks and returns of the biggest companies. Investors use it as the benchmark of the overall market. The S&P rates are information for how likely a debt will be repaid. They aren't investment recommendations nor do they predict the probability of default. S&P also rates the creditworthiness of individuals bonds. There are five different types of bonds, all of which vary in their ratio of risk versus return.

The S&P 500 tracks the market capitalization of the companies in its index. Market cap is the total value of all shares of stock a company has issued. It's calculated by multiplying the number of shares issued by the stock price. A company that has a market cap of $100 billion receives 10 times the representation as a company whose market cap is $10 billion. The total market cap of the S&P 500 is $23.5 trillion. It captures 80 percent of the market cap of the stock market. In addition, the index is weighted by a float-adjusted market cap. It only measures the shares available to the public. It does not count those held by control groups, other companies, or government agencies.

But why the investor choose the S&P 500 index? To answer this question we need to check the difference between the S&P index and other Stock Market Indices. First, The S&P 500 has more large-cap stocks than the Dow Jones Industrial Average. The Dow tracks the share price of 30 companies that best represent their industries. Its market capitalization accounts for almost one-quarter of the U.S. stock market. Then, The Dow tracks the share price of 30 companies that best represent their industries. Its market capitalization accounts for almost one-quarter of the U.S. stock market. Depend on all these differences, it is not necessary for the investors to focus on all these three stock indexes.

After we know what is S&P 500, we also need to understand how to use this index to do the investment. Although people cannot make profits on the S&P, S&P index fund is still a helpful tool to advise investor how to weight the shares of stocks in their portfolio according to the market cap. In other words, the S&P 500 should be used as a leading economic indicator of the U.S. economy. Therefore, S&P index will be a good tool to predict the stock price for each day.

2 Related work

To analyze the stock market data is challenging and rewarding, each kind of stock has its unique and difficulty. In the past years, many relevant types of research engaged to analyze various properties on stock market data and made predictions on trend and direction of stock in the future. Those are valuable experiences and solutions for us to know and learn. Therefore, we present some related work to our study about stock market data and the approaches they used.

Kaggle is an open source website and it allows users to share data resources and data science projects. There is an analysis posted and modified by Janio Alexander Bachmann 4 months ago about S&P 500 Stock Data in Kaggle, which uses the same data as us and has the most top feedback in this part. It used Prophet to forecast the time series of S&P 500, and Prophet is a kind of time series forecasting model developed by a Facebook's research team. In his approach, Python is used to read the data and fit Prophet, and it is easy for him to import various packages and plot diagrams. We also use Python as our programming language, and some technologies and strategies based on Python we use to forecast low price are similar to him.

Another approach relevant to our study is also from Kaggle, Pankul Jain implements LTSM model to predict stock price movement. He collects the past 60 days closing price of AAL company as training data to train the model and tests the performance of the model by plotting the predicted stock prices and actual stock prices of other companies. In our consideration, a clear forecasting should be a training data to fit the classifiers model such as linear regression, and a testing data for the trained model to generate the predicted price. Meanwhile, the output should contain professional demonstration in terms of plotting, and the Hakki_kaan's PLOTLY TUTORIAL serves well as an example for this.

We integrate approaches from these related work and make some considerations for our study. With our recent programming experience, Python is the first choice for us to use owing to its convenient and functional. Next, a model can help us to fit data into it and forecast the stock more accurately. Finally, audiences prefer to see a clear and meaningful diagram, we need to use some technologies on plotting to make our project more productive.

3 Data processing

This part, we define our data from the source and analysis the attributes to choose the part we need.

Data information

The dataset we used is from Cam Nugent's post on Kaggle, it is about all S&P 500 companies historical data for the past 5 years. This dataset is downloaded as .csv file.

Type of attributes:

- Date
- Open stock price
- Highest point
- Lowest point
- Close stock price
- Volume
- Stock name

Data preparation

Since S&P 500 stock data is recorded daily, so it provides a huge amount of datasets that correspond to so many different companies. After calculation, this data contains over 600,000 rows of data, which is too large to use WEKA to analyze all of them. Moreover, stock price predictions between different companies are highly independent, which is that the stock price of a company does not significantly influence another company usually. Therefore, we decide to select one company to analyze its stock data and offer relevant prediction and regression.

Amazon is one of the world-leading corporations, and it conducts the most impact on this planet. Thus, we select Amazon's stock price data as our target data. To be exact, the number of rows that Amazon stock data is 1259, which means there are 1259 days of data that contain the stock open price, close price, highest price, lowest price, and volume.

4 Data mining

4.1 comparing classifiers

According to the study from the lectures, we learned linear regression. In the process of data mining, we used this classifier to analyze the dataset and then compare the result with persistent model to see which one fits the data better. For this project, Jupyter Notebook and python3 are used to process the data.

The steps are as follow:

1. **Data Search, Extract and Load:** The data we used are sourced from Kaggle (see reference [1]). The raw dataset is a large dataset which contains over 600,000 rows. Therefore, we extract AMAZON stock data from the original dataset and load the data into

a new CSV file in order to create a new dataset for us to analyze and predict, since the project only focused on analyzing the stock data of AMAZON.

2. **Class Assigner:** This step selects the Class we want to predict and the Classes that have a significant influence on the Class we want to predict. In this case, we are analyzing and predicting the stock data that contains the stock open price, close price, highest price, lowest price, and volume for the next 30 days for AMAZON.

3. **Cross-Validation Fold Maker:** This step is setting up the training set and test set for the use of each classifier. In this case, we need to prepare two classifiers.

4. **Classifier Performance evaluator:** This step evaluates and compares the result of the classifiers we have used on the dataset in order to pick the classifier that fits the most to the data.

4.2 Expert Data Mining

After selecting and executing the classifiers we proceed to analyze the results and findings. This projects includes two classifiers. The first one is persistent model, and the second one is autoregression. The project compares the result of these two classifiers then find out which classifier fits the data better, and which one provides better result on the prediction.

4.2.1 Import Useful Package

At the very beginning of the Python code, just like usual, we import the necessary libraries and packages, shown as the following figure:

```
In [27]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import matplotlib.dates as dates
         import matplotlib as mpl
         import mpl_finance

         from mpl_finance import candlestick_ohlc
         import seaborn as sns
         import datetime as dt
         #from fbprophet import Prophet
         import statsmodels.api as sm
         from scipy import stats
         from pandas.core import datetools
         import warnings
         import os
         warnings.filterwarnings("ignore")
         plt.style.use("seaborn-whitegrid")
```

Then we print the data frame from the dataset, This shows the preview of the dataset. Notice that the whole dataset is huge, we are just showing the first five for convenient. Starting from the

left column to right column are the date, stock open price, stock highest price, stock lowest price, stock close price, stock volume, and the stock name.

Out[28]:

	date	open	high	low	close	volume	Name
0	2013-02-08	15.07	15.12	14.63	14.75	8407500	AAL
1	2013-02-11	14.89	15.01	14.26	14.46	8882000	AAL
2	2013-02-12	14.45	14.51	14.10	14.27	8126000	AAL
3	2013-02-13	14.30	14.94	14.25	14.66	10259500	AAL
4	2013-02-14	14.94	14.96	13.16	13.99	31879900	AAL

4.2.2 Line Chart of Amazon stock price

Just like we mentioned in the Data Processing section, the whole datasets that contained in S&P 500 stock is incredibly huge and we are only interested in the stock data of Amazon, so we extracted the dataset of Amazon and then print out it in the form of the line chart. Clearly, even though the line goes up and down sometimes, the general trend of Amazon stock is going up in the past five years.

In [30]:
```
fig,ax1 = plt.subplots(figsize=(20, 10))
plt.plot(data[['open','close','high','low']])
plt.show()
```

Amazon stock trend for the past 5 years

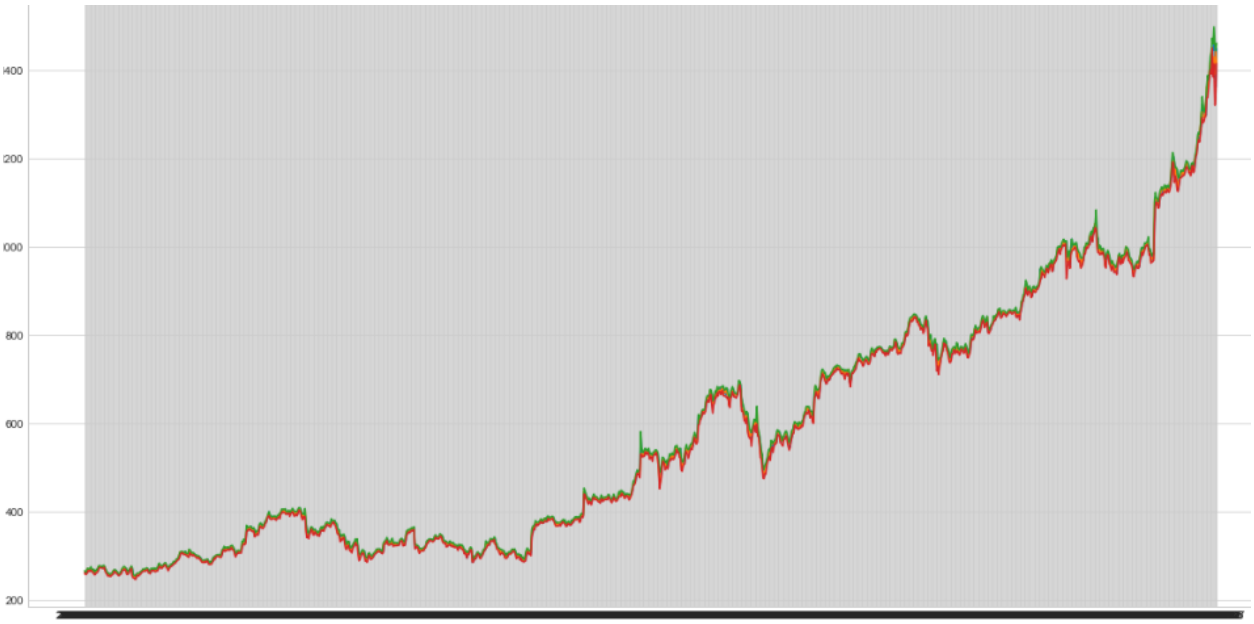


Figure 1

4.2.3 Candlestick Chart

The following chart indicates the market's open, high, low and close price of a specific day. It provides a clear and general look for the details of the stock price of Amazon. The "candle body" represents the range between the open and close price of the market in a specific day.

Plot2:

Figure 2

4.2.4 Bar Chart

The bar chart offers the stock volume trade on the market in a specific day.

```
In [32]: data[['volume']].iloc[1220:].plot.bar()
         plt.show()
```

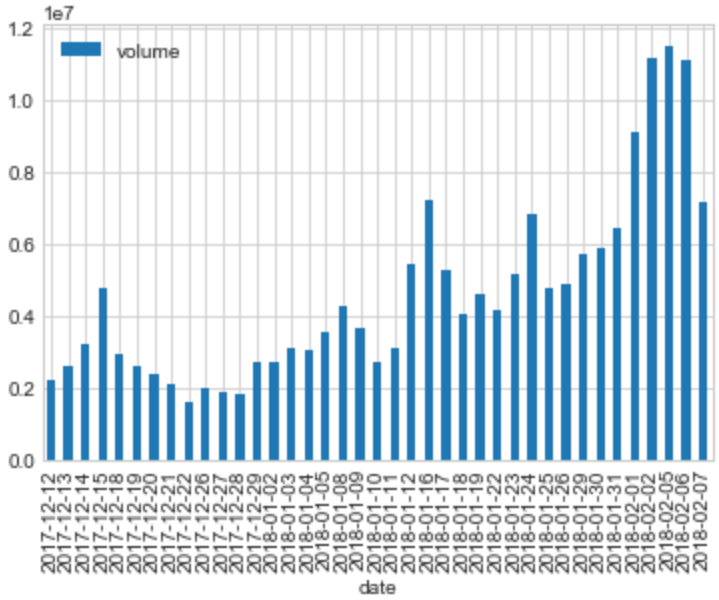


Figure 3

4.2.5 Lag plot

Now we use the lag plot to find the autocorrelation of the lowest point of Amazon

```
In [33]: from pandas.tools.plotting import lag_plot
         lag_plot(data['low'])
         plt.show()
```

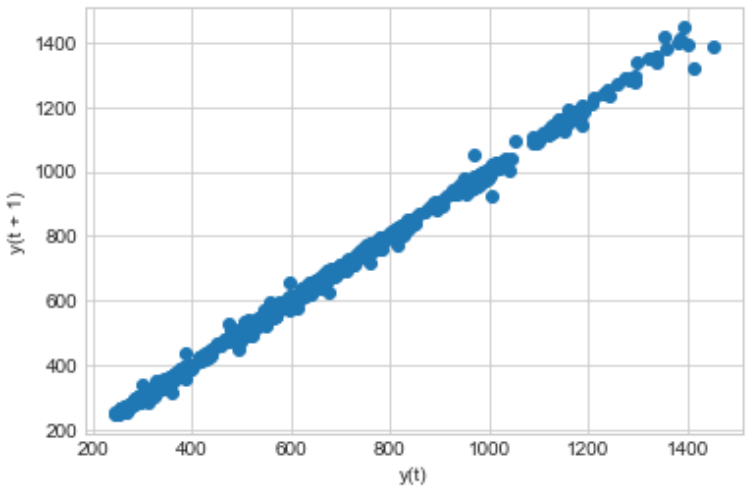


Figure 4

4.2.6 AutoCorrelation Plot

This is a good way to give a predictive model and show the relationship between observation and historic values changes over time.

```
In [34]: from statsmodels.graphics.tsaplots import plot_acf
         plot_acf(data['low'])
         plt.show()
```

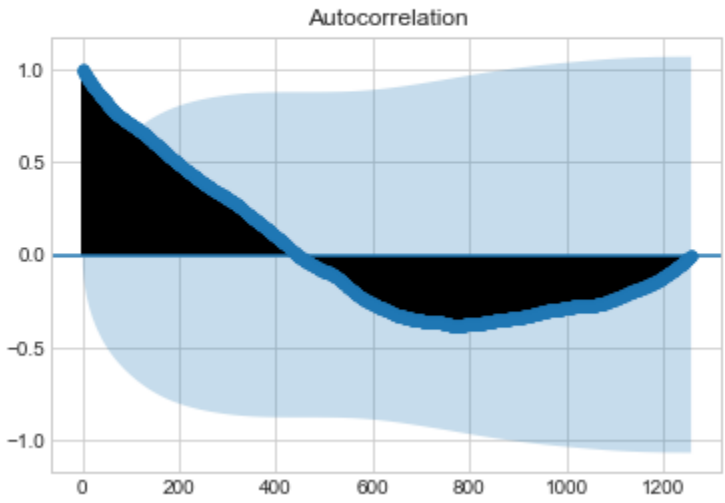


Figure 5

4.2.7 Persistent Model

The Figure-6 is aimed to show the difference between the true value and the value from the regression analysis. After we get the result from the predicted value, we can check the accuracy of our data mining.

```
print('Test MSE: %.3f' % test_score)
# now we plot the predict model vs the true value
reg_val, = plt.plot(predictions,color='r',label=u'Predicted Linear')
true_val, = plt.plot(test_y,color='g', label='True Values')
plt.legend(handles=[true_val,reg_val])
plt.ylabel('Dollars')
plt.xlabel('Days')
plt.show()
```

```
Test MSE: 590.314
```

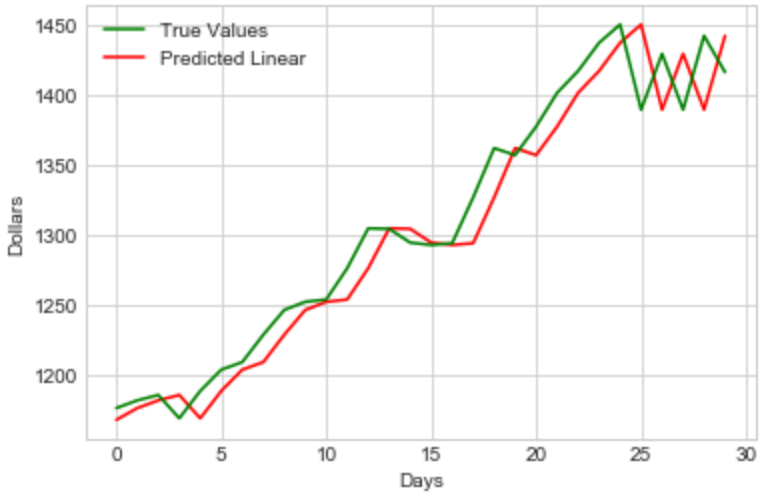
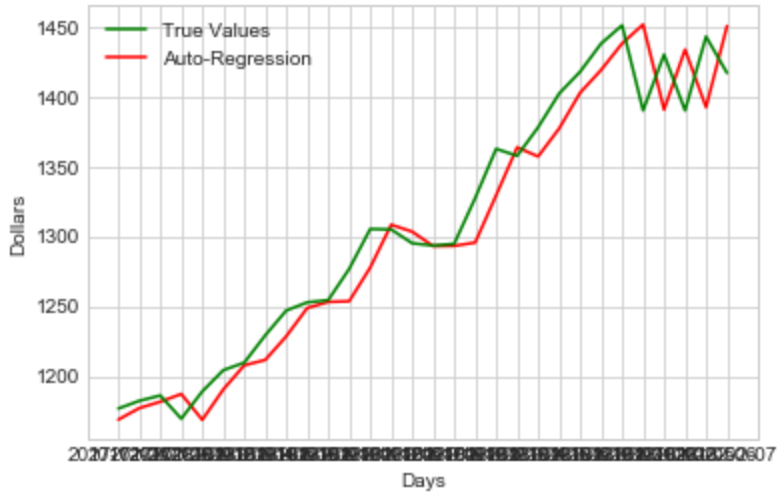


Figure-6

4.2.8 Autoregression Model

```
predicted=1168.839097,  expected=1176.760000
predicted=1177.082225,  expected=1182.260000
predicted=1181.626281,  expected=1186.100000
predicted=1187.166556,  expected=1169.470000
predicted=1168.686242,  expected=1189.010000
predicted=1190.400661,  expected=1204.200000
predicted=1207.683533,  expected=1209.590000
predicted=1211.523970,  expected=1229.140000
predicted=1228.431133,  expected=1246.870000
predicted=1248.697426,  expected=1252.700000
predicted=1253.030633,  expected=1254.330000
predicted=1253.527487,  expected=1276.680000
predicted=1277.599870,  expected=1305.200000
predicted=1308.234148,  expected=1304.860000
predicted=1303.277317,  expected=1295.000000
predicted=1292.935823,  expected=1293.320000
predicted=1293.131118,  expected=1294.580000
predicted=1295.462206,  expected=1327.310000
predicted=1329.316128,  expected=1362.540000
predicted=1363.589745,  expected=1357.510000
predicted=1357.051695,  expected=1377.950000
predicted=1377.037418,  expected=1402.050000
predicted=1402.626734,  expected=1417.680000
predicted=1418.943404,  expected=1437.820000
predicted=1437.904212,  expected=1450.890000
predicted=1451.480881,  expected=1390.000000
predicted=1390.571375,  expected=1429.950000
predicted=1433.546191,  expected=1390.000000
predicted=1392.293915,  expected=1442.840000
predicted=1450.404000,  expected=1416.780000
Test MSE: 598.638
```



Now we are going to compare with two MSE results:
From our Persistent model, we get MSE is 590.314
From our Autoregression model, we get MSE is 598.638

Our basic model provides a basic linear-line of performance for the close stock price and the date. The regression model is selected the lag value using statistical tests and trains a linear regression model.

Conclusion

As mentioned at the very beginning of this report, the Stock price has grabbed the notice of a large number of populations. It is expected to come up with a reasonable model that can predict the upcoming trend of the stock market. Luckily, there are a lot of resources online that collects the datasets of various stock information.

However, S&P stock data that has provided in Kaggle is too large to analyze and give predictions to all of them. Therefore, we extract the part of it and analyzed it. Amazon's stock price is a good example and has reasonable datasets.

Since S&P stock data contains the stock price up to recent five years, so the previous data might be not very relevant. In order to have a higher accuracy, we extract the first 97% of data to be training data and the rest of them to be the testing data. To be more specific, there are totally 1259 datasets for Amazon stock data in the past five years, the first 1229 datasets to be the training data, and the rest 30 datasets to be the testing data.

The Given MSEs from the Persistent model and the Autoregression model are relatively close(590.314 and 598.638 respectively). Therefore, we can say that the predictions we provided with two line charts are accurate and trustable.

Acknowledgments

This report is the project of the Fall 2018 Data mining course (CSC421) in the University of Victoria. We thank everyone who has provided us with support and assistance, especially thank Dr. Alex Thomo for his teaching and guidance during the course.

Reference

[1] “Kaggle” [Online Resource]. Available: https://www.kaggle.com/camnugent/sandp500/home

Janio Alexander Bachmann, “S&P 500 || Simple Forecasting with Prophet” [Online]. Avaliable: https://www.kaggle.com/janiobachmann/s-p-500-simple-forecasting-with-prophet

Hakk_kaan, “PLOTLY TUTORIAL-3” [Online].
Avaliable: https://www.kaggle.com/hakkisimsek/plotly-tutorial-3

Pankul Jain, “LTSM - stock price movement prediction” [Online].
Avaliable: https://www.kaggle.com/pankul/lstm-stock-price-movement-prediction