An Analysis of Stock Market Prices using Regression and Pattern Matching

Patrick Gomes and John Gonsalves

Abstract

Our goal for this project was to use company stock data available from Yahoo! Finance to analyze a given company's stock and make predictions for its near future value. In order to do this, we looked for patterns in the data at specific intervals (quarters) during the year. We conjectured that it might be the case that a certain company always has a decrease in stock towards the end of the summer. Second, we wanted to explore the effects on a company's stock value for the volume of stocks traded during the day.

Introduction

For many professions, it is important to be financially aware of the top performing businesses in a given field. Do specific companies perform better or worse in a given season? What type of correlation does volume, a very important indicator when measuring the worth of a market move have on stock price. Is it possible to predict prices of a company's stock in the near term future?

Our main dataset for this project will involve stock prices from a range of technology companies provided by Yahoo! Finance. We plan on writing a web crawler in Python to grab the data from their site. Attributes of the dataset we'll be working on include: date, open price, highest price of the day, lowest price of the day, close price, volume, and adjusted close price. From this list we will primarily be utilizing the date, open price, close price, and volume.

The results we hoped to get from this experiment included a better idea of the influence of season on a company's stock, the impact of volume on a company's stock, and a predictor for the near future of a company's stock, something that has always been regarded as a challenging task of financial time series analysis. From this information, we can hopefully provide investors with patterns that help influence their investment decisions.

Technique used in this project (e.g., what clustering, classification, or regression method and why)

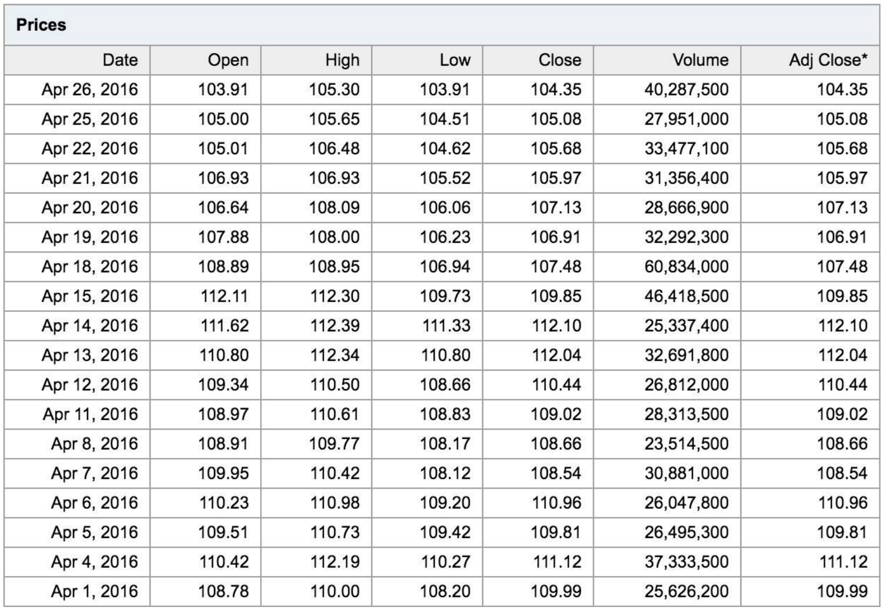
There were two main techniques that we used for our project: Regression and Pattern Mining. We used the Scikit library to implement these features. We also used the Ordinary Least Squares model to retrieve statistics on our regression. For the correlations, we used a mixture of python libraries including, numpy, matplotlib, sklearn, and statsmodels in order to plot our graphs.

For our volume correlation we performed a linear regression on a scatter plot of volume to change in price. Linear regressions combined with the information from the Ordinary Least Squares model provided us with a best fit line. Using this best fit line we easily visualized of the actual correlation between volume and a stock's performance.

For the Pattern Mining we are training Scikit with the previous data of a single stock's prices. For each year, we will normalize the price attributes to account for the change in the value of currency over time. We will be analyzing two different patterns: the company's recent trends and its performance in previous years during the same quarter. From scikit we will be using LinearSVC for these predictions.

Datasets and experiments.

We are extracting our data from Yahoo! Finance. The website (www.finance.yahoo.com) has a quote search bar which we use to lookup the stock information from. From there we navigate to the Historical Prices where we can set our ranges. We parse the data into a Pandas DataFrame for easier management. The columns of the Data Frame are: date, open price, close price, highest price, lowest price, and volume. Below is an example of a small portion of the data we retrieved from Apple stocks:



Prices

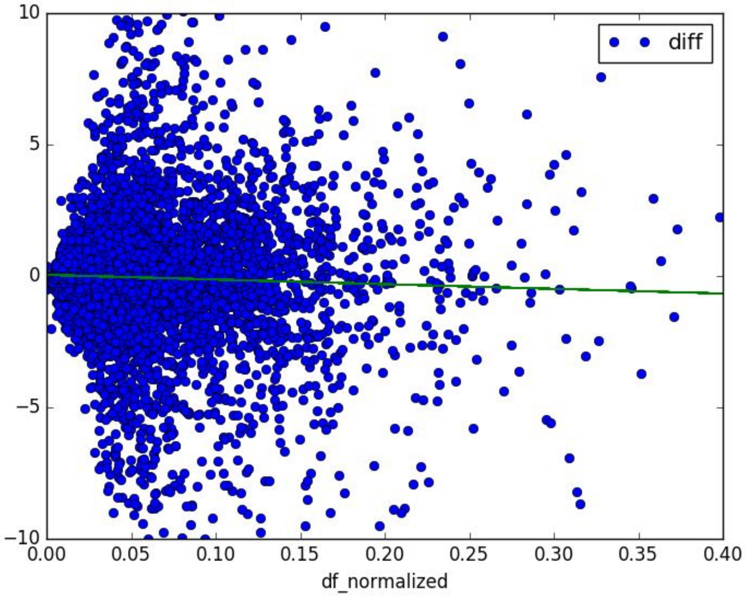
Date	Open	High	Low	Close	Volume	Adj Close*
Apr 26, 2016	103.91	105.30	103.91	104.35	40,287,500	104.35
Apr 25, 2016	105.00	105.65	104.51	105.08	27,951,000	105.08
Apr 22, 2016	105.01	106.48	104.62	105.68	33,477,100	105.68
Apr 21, 2016	106.93	106.93	105.52	105.97	31,356,400	105.97
Apr 20, 2016	106.64	108.09	106.06	107.13	28,666,900	107.13
Apr 19, 2016	107.88	108.00	106.23	106.91	32,292,300	106.91
Apr 18, 2016	108.89	108.95	106.94	107.48	60,834,000	107.48
Apr 15, 2016	112.11	112.30	109.73	109.85	46,418,500	109.85
Apr 14, 2016	111.62	112.39	111.33	112.10	25,337,400	112.10
Apr 13, 2016	110.80	112.34	110.80	112.04	32,691,800	112.04
Apr 12, 2016	109.34	110.50	108.66	110.44	26,812,000	110.44
Apr 11, 2016	108.97	110.61	108.83	109.02	28,313,500	109.02
Apr 8, 2016	108.91	109.77	108.17	108.66	23,514,500	108.66
Apr 7, 2016	109.95	110.42	108.12	108.54	30,881,000	108.54
Apr 6, 2016	110.23	110.98	109.20	110.96	26,047,800	110.96
Apr 5, 2016	109.51	110.73	109.42	109.81	26,495,300	109.81
Apr 4, 2016	110.42	112.19	110.27	111.12	37,333,500	111.12
Apr 1, 2016	108.78	110.00	108.20	109.99	25,626,200	109.99

The data we extract is dependent on the user-input, which can be any valid company code included in any of the supporting files. The first thing we had to do to all the data was normalize it to get rid of outside factors such as inflation. The values were normalized month by month. After normalizing, for each day we defined a difference price which is equal to close price - open price). These values were then used for the volume regression.

The pattern mining data is separated into its own lists. The data was separated by months and years to create a list of normalized values per month per year. Creating linear regressions on these data points and averaging all the months together allowed us to create an average expected pattern for each month which was used as our predictor. Other information created from the data was fluctuation and deviation, the average change per day and average highest/lowest change per month respectively.

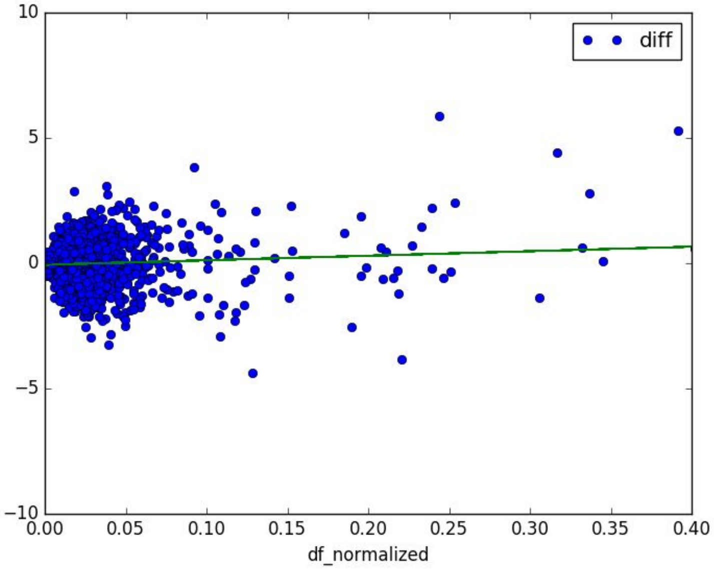
Results and Discussion

The following regression on the volume and the stock's change in value for Apple (AAPL) shows a negative correlation between volume and the stock's performance that day. This is the opposite of what we were expecting, however a possible explanation might be that a larger company like Apple gets larger volumes when they are doing poorly.



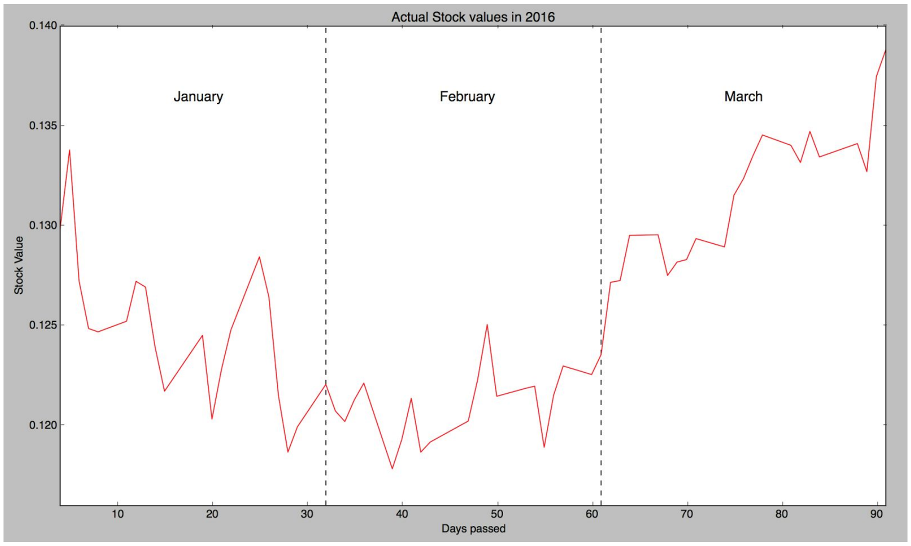
```
OLS Regression Results
------------------------------------------------------------------------------
==============================================
                      coef         std. dev
------------------------------------------------------------------------------
df_normalized     -1.3301      0.398
==============================================
```

In comparison, the graph below which was produced after running our program on a smaller company Accorda Therapeutics gives us a positive coefficient, which is what we were originally looking for. This is probably due to the smaller company getting more recognition and benefitting from it.

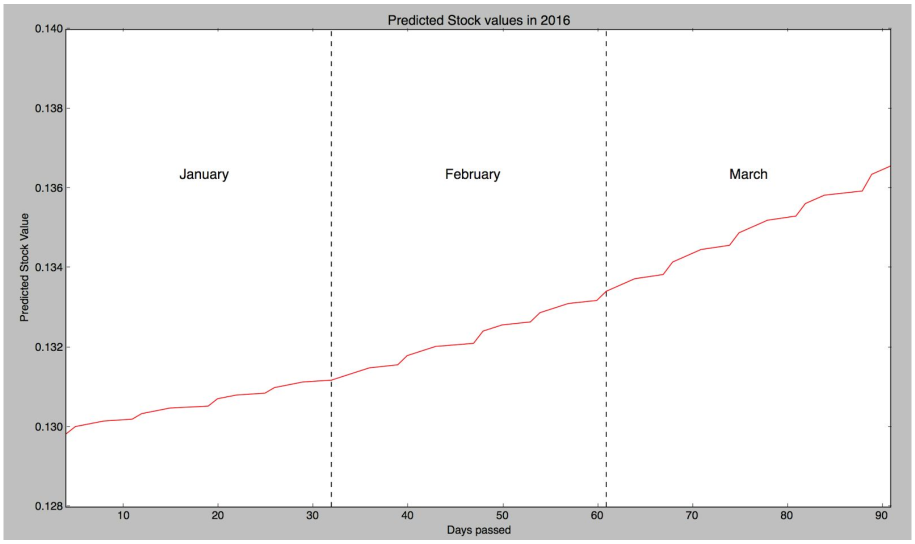


```
OLS Regression Results
------------------------------------------------------------------------------
df_normalized     1.2899      0.263      4.907      0.000        0.774     1.805
==============================================================================
```

The graph below represents the actual stock prices for Apple between the three month period of January to March. Given the time constraints, our predictor only acted on the first quarter of the year, the time period however is easily flexible.



And the following graph is our predictors output:



The prediction graph fails to predict the massive drop at the beginning of the 2016 year, but when the market bounced back up our prediction model was within a few percent, implying long term correctness. There are corrections that could be made to the algorithm that would hopefully lead to a better result but we are unsure of how it would perform unless we were to actually test it.

January Fluctuation = 0.0095749908656
January Highest Deviation = 1.15137100573
January Lowest Deviation = 0.916769922996

February Fluctuation = 0.00817186885369
February Highest Deviation = 1.09771570949
February Lowest Deviation = 0.921968981055

March Fluctuation = 0.00804821230373
March Highest Deviation = 1.1017793796
March Lowest Deviation = 0.920379759324

These values aren’t included in any of the graphs but can be used to see if our data prediction model falls within the expected bounds for a month.

Conclusion

In general, we weren’t able to find any concrete correlation between volume and a stock’s performance. If we were to continue working on the project, could reduce the number of points, or implement a clustering algorithm. The majority of points existing in the single cluster in our graph definitely influenced our OLS and best fit line. Clustering the points and reducing the amount of data taken would help to alleviate this. The combination of regressions were able to predict decently long term but would miss trends in between. A better approach would be to create an algorithm that learns as more data becomes available by modifying the slope to better fit the current trend, while still keeping the previous patterns included.