Stock Market Prediction

Michael Pixton, Saul Ramirez, Daniel Reich
CS 472, Fall 2020
Department of Computer Science
Brigham Young University

Abstract

With the inherent difficulty to predict the stock market, a machine learning approach was taken to predict closing prices of major technology focused companies namely: Google, Amazon, Facebook, Microsoft, Apple. Using features from the previous day, we designed a machine learning model that gave excellent accuracy for predicting the day's closing price. This shows that markets do follow trends and if other parts of the market can be modeled, then closing price can as well.

1 Introduction

The stock market has been and will continue to be a leader in the machine learning world. Modeling market trends and predicting market bids attracts individuals and companies who hope to grow their prestige and outdo the competition. Today, most all trading is done electronically making it easier than ever to design programs and models to trade for you. Much research has already been done in this field and many large corporations exist solely to better understand how to model the free market. Thus the learning and approaches done in this paper are not new or unique but they were performed on new data and were eye opening to understanding the difficulties when predicting market trends as well as see other possibilities for market modeling. Additionally, free trade has always had some inherent risk. Despite many complex and interactive models, it is still considered difficult to model correctly, particularly very far into the future. Our research attempts to use only a few features correctly predict trends on a few key technology stock options.

Various companies of a similar stock group were chosen as the representatives of this model. Google, Amazon, Facebook, Microsoft, and Apple all hold large market shares as technology focused companies and thus were good candidates for designing a general model to predict technology market movement in a single day.

1.1 Motivation

Predicting and modeling the stock market posed an interesting challenge; it was less about trying something someone else hadn't done before, and more about letting us try it out and maybe stumbling upon a better solution. It was also personally motivating for some of us; with the ability to invest well, we can better secure a future for ourselves thus making modeling the stock market a self-interest focused project.

2 Methods and Data

2.1 Data Source

The data for this project was obtained using Yahoo Fianaces's API. This source provides stock quotes, movers, financial reports, charts and more. The API contains daily information on stocks and data since 2016 was used for simplicity. This API does not contain a large number of features; and this could hinder the development of an accurate model. However, the features available here are also available in many other places making this information pertinent and making our modeling techniques relatively easy to replicate. This data is relevant to the question at hand as we are attempting to predict the market a day in advance and need daily information about market changes within each of the selected companies.

The data source did change during the course of this project as some data sources proved to be more difficult to work with despite some having more interesting information. Yahoo Finance was the best way to obtain the needed daily stock information and thus was selected for use for the entirety of the project.

2.2 Data sets

Data concerning the technology companies of interest: Google, Amazon, Facebook, Microsoft, and Apple was obtained by pulling five datasets from Yahoo Fianace's API. Each day has numeric values for the day's High, Low, Open, Close, Volume, Adj Close. Each dataset contains approximately 1210 rows representing each day from

February 17, 2016 till the time of making this study and model, December 04, 2020.

Initially, all features were used within these datasets to predict Adj Close, but Close was in many cases exactly the same and would dominate the other features within the model. Thus, Close was dropped giving the model five initial features. These five initial features were then used to model the day's Adj Close price.

2.3 Selected Models

Various models were attempted to find a Regressor with the best accuracy. A Multilayer Perceptron with Backpropagation was fitted with some hyperparameters that showed promise. An ensemble with Ridge Regression, Random Forest Regressor, and KNN Regressor was found to show good results as well. Between these two models, a few other modeling methods were attempted, but did not give as good initial results so they were not included in the analysis.

3 Initial Results

Initially, the data was inspected, with the focus on seeing trends with the Daily Adjusted Closing Price. The Volume Traded seems to be of interest as this could show more change in a single day. The next two plots below show the change in market price and change in Volume Traded both over time. The last plot shows the correlation of Adjusted Close Price between the selected technology companies.

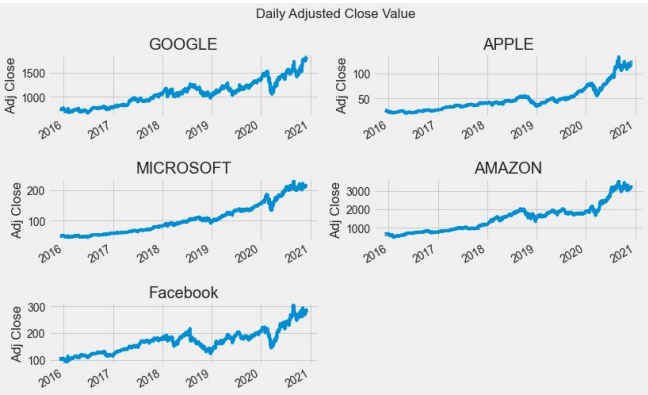

Figure 3.1: The Adjusted Close Price for the 5 Tech companies in the S&P 500

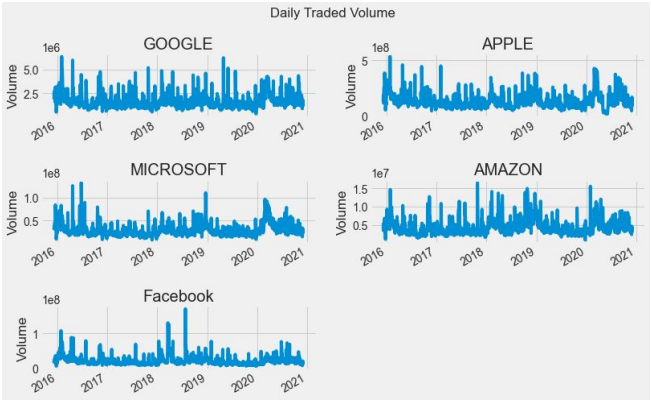

Figure 3.2: Volume of daily trades for the 5 Tech companies in the S&P 500

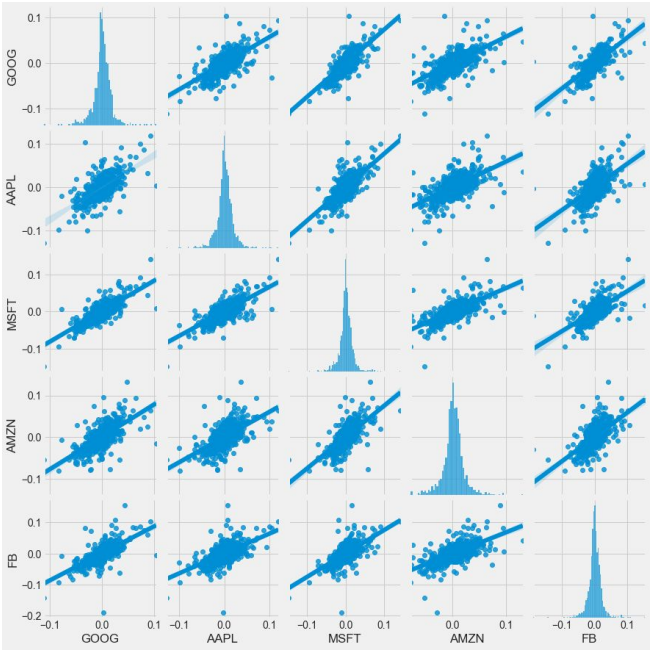

Figure 3.3: Correlation in daily Adjusted Close Price change percentage between the 5 major tech companies.

The Multilayer Perceptron Model was originally fit to the data and given an R-squared value of around .995 This implies that our model can already predict readily 99.5% of the variation in the Daily Adjusted Close Price by using the initial features. However, this method assumes all data, knowing the opening, closing, volume, highs and lows for the day it calculates the adjusted close price after the market has already closed making this model's results useless.

Another model that showed promise was an ensemble of a Ridge Regression model, a Random Forest Regression model and a K-neighbors Regression model. For this model, we change the predictions to be for 1, 5, 10, 20, 50, 100, 365 days in advance. This model is attempting to predict the

future Adjusted Close Price by some number of days in advance. The chart below shows the results from this ensemble.

Company	Days In Advance	R-squared
GOOG	1	0.993
AAPL	1	0.996
MSFT	1	0.997
AMZN	1	0.996
FB	1	0.987
GOOG	5	0.979
AAPL	5	0.998
MSFT	5	0.995
AMZN	5	0.996
FB	5	0.980
GOOG	10	0.986
AAPL	10	0.994
MSFT	10	0.996
AMZN	10	0.996
FB	10	0.979
GOOG	50	0.962
AAPL	50	0.989
MSFT	50	0.996
AMZN	50	0.994
FB	50	0.933
GOOG	100	0.958
AAPL	100	0.957
MSFT	100	0.995
AMZN	100	0.980
FB	100	0.862
GOOG	365	0.830
AAPL	365	0.932
MSFT	365	0.983
AMZN	365	0.856
FB	365	0.848

Figure 3.4: R-Squared terms for ensemble of various days in advance.

This ensemble shows promise and helps identify that as predictions on the market increase in time scale, our accuracy decreases. However, the Multilayer Perceptron model showed more promise and had more flexibility when changing parameters so the Multilayer Perceptron was used as our final model.

As stated previously, one of the issues with the initial dataset of features, is that it contained Close Price and Adjusted Close Price. The plot below shows how accurate our model was if Close price was included within the features.

Figure 3.4: Google's Adjusted Close Price results using Day High, Day Low, Close, and Volume as input features.

4 Model and Feature Improvement

In order to improve the model, some features were engineered to help our model better predict. A moving day average was created for 10, 20, and 50 days. This feature represents the average of Adjusted Close Price for the last set number of days. This allows the model some direct time correlation which is factually very significant when predicting stock prices; a stock price is much more heavily influenced by the last 10 days of movement rather than a random 10 days two years ago.

Another feature engineered was a Percent Change from the previous day. This feature correlates just the current day's Adjusted Closing Price with the previous day. With new features, that help with time correlation, we were now ready to refit our model. This time, more hyper parameters were tested on the Multilayer perceptron model. The best parameter selection included 40 hidden layers, a constant learning rate, and a validation fraction of 1/5.

In order to create a more accurate and useful model, the approach to the original model was changed. To create a more useful Adjusted Close Price Model, the 'Afternoon Stock Model' was created. For this methodology, the Day High, Day Low, Close, Volume, 10 Day Moving Average, 20 Day Moving Average, and 50 Day Moving Average

features were shifted one day toward the future. The intuition is that at the beginning of everyday all historical data is known, and the only data known for the current day is the stock's opening price. Using these features as input, the model predicts the stocks Adjusted Close Price for that afternoon. By using the Afternoon Stock Model, day traders can determine if the stock they are holding is worth getting rid of before the day's end, or if they should hold on to it.

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Figure 4.1: Normalization Formula for Adjusted Close Price

The two figures below show the application of the normalization formula on Google's closing price. The first figure is the predictions unnormalized, and the second, is the predictions that were first normalized, then fitted, and finally converted back to the original scale.

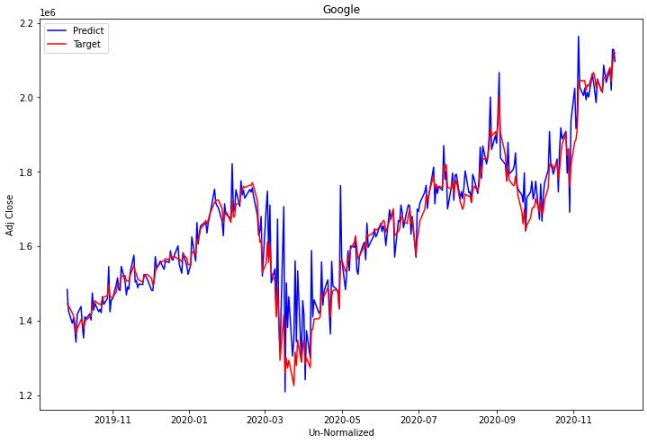

Figure 4.2: Predictions vs Actual of Google's Un-Normalized Adjusted Close Price

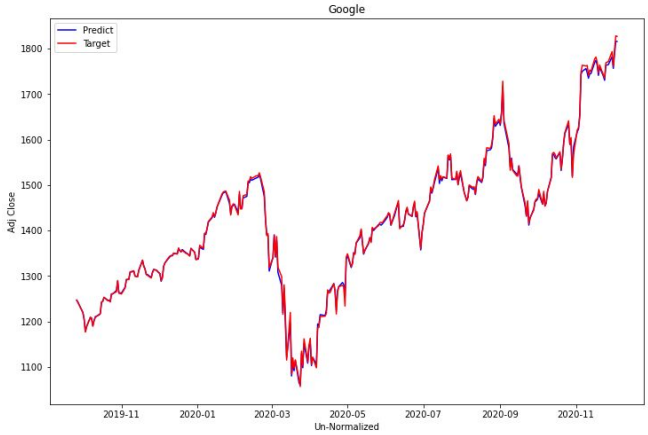

Figure 4.3: Predictions vs Actual of Google's Adjusted Close Price Normalized First, Fitted, then Converted Back

5 Final Results

The new model fits the data much better and it is more intuitive than the previous model. Instead of modeling the current day's Adjusted Closing Price using the data from that day, which in reality would normally be inaccessible, we now use the previous day's High, Low, Closing, Volume, Averages and current day's Opening Price. This information changes the previous methodology and allows us to create more valid predictions for a future stock's Adjusted Closing Price. The next figure shows the final model's training, targets, and predictions.

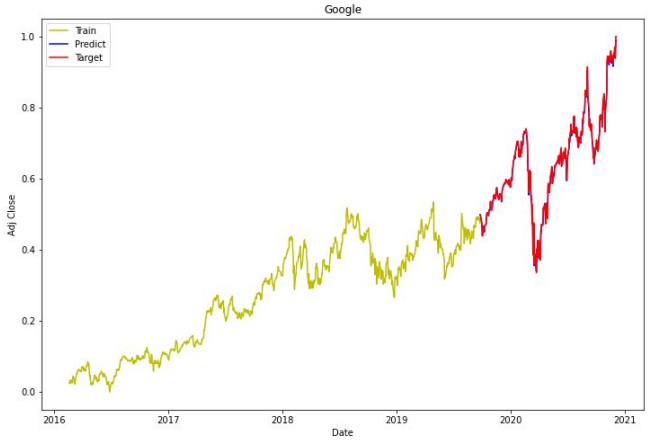

Figure 3.5: Afternoon Stock Model using Day High, Day Low, Close, Volume, 10 Day Moving Average, 20 Day Moving Average, and 50 Day Moving Average datasets adjusted 1 day forward with the opening price staying the same.

The overall R-squared of the final mode is .981 over five iterations, showing that now our model can predict 98.1% of variation in a stock's Adjusted Closing Price using features that would readily be available at the beginning of

that day. This is an additional 8% increase from the original model plus this model is more applicable to traders.

6 Conclusions

The final Multilayer Perceptron Model shows an excellent learning of features originally given. We were able to devise a model that can predict the closing price of a technology stock with 98% of the variation accounted for. The final testing was done primarily on Google's stock prices but all companies could be applied with similar results. The changes performed on the model throughout the process helped refine the model and reach our goal more accurately. We consider this model to be useful and insightful when performing market analysis. One key takeaway was the significance of normalizing that targets which helped increase accuracy. Another key takeaway from this project is finding a model that predicts something useful. The original model was not useful as it required data that would not normally be available on a given day.

In conclusion, the Multilayer Perceptron Model following the "Afternoon Stock Model" shows promise and can help traders evaluate if they should sell or keep their stock on a daily basis.

7 Future Work

The project was completed using a multi-layer perceptron with backpropagation algorithm. This model is simple to set up, but is not as efficient when attempting to analyze time-series data. Future work on this type of problem could be to reproduce the exercise presented above using a recurrent neural network (RNN) and Long Short Term Memory (LSTM) models. RNNs are models that remember inputs and learn to forget data when it is no longer important. Using a model with memory would be optimal in being able to forecast the adjusted close price for a single day and using the output to predict the day after that.

Another option for future work would be to revisit the ensemble method that showed promise initially. This model has the inherent advantage of combining multiple models together thus reducing bias.