Machine Learning Engineer Nanodegree

Capstone Project

Projjol Banerji
August 22nd, 2017

I. Definition

Project Overview

Stock price prediction has been an interesting and hard problem for quite a while now. The information it provides can give an individual or even a firm a strong overview of what is to come helping them plan for the same. My motivation for the project stems out of my interest in the 2008 economic crash. While I was quite young at that point, I remember it as a defining moment in our history. Later on, as I grew up, I tried to collect as much information about the event as possible. Throughout this process, all that was there in my mind was, how could Wall Street not see this coming? The answer to that I would think is cognitive bias. I think this is a blatant example of the normalcy bias, wherein even in moments of grave danger, one pretends as if everything is fine. With the knowledge that I acquired through the course of the nanodegree, I felt I could help create a prediction tool, which would provide people with data that can help them get over this bias, and help me understand the stock market better.

In this project, I will be using technical analysis as a method to make sense of the data and make accurate predictions from it. In technical analysis past trading activity and price changes of a security are better indicators of the security's likely future price movements, which is the eventual goal in this project. Technical analysis derives from two underlying principles: * Market price discounts every factor that may influence a security's price * Market price movements are not purely random but move in identifiable patterns and trends that repeat over time For technical analysis to be used, technical indicators will be used for forecasting future prices. More on this in the later sections.

The other forms of financial analysis that are available to us but not used in this project are : * Fundamental Analysis: Evalutaion of a security via going through the financial statements of a company to determine the intrinsic value of the givne security. Once this has been determined, comparing the intrinsic value versus the market value, gives the investor insight as to whether the security is undervalued or overvalued, thereby letting him buy more or sell respectively. This sort of analysis will not help us, as the primary objective of fundamental analysis and this project are not the same * Quantitative Analysis: It is a branch of financial analysis wherein events and behaviours are predicted using mathematical measurements and statistical modelling. Quantitative analysis is an important tool and is used in a variety of situations from the evaluation of an instrument to predicting changes in a country's GDP (Gross Domestic Product)

In this project I will be creating a stock market predictor, which has ingested data from 16 years and

would predict the estimated value of the market on a given query date.

Problem Statement

The problem at hand is to accurately predict future market prices for a given index or asset. For this, I will use the S&P 500 index. Reasons for using this index over others are: * Historic data dating back to the 1990s is easily available for this index, allowing the learning algorithm to see more examples and thereby more variations in the data * Another popular index is the DOWJ 30 index, but the problem with that index is that it monitors the prices for only 30 selected companies. The S&P 500 index on the other hand is considered a litmus test for the American economy as a whole and trends followed in it affect not only the American business economy but global economies * Given the influence of the index and the far reaching data, it made sense to me to use the S&P 500 index

Concerning the random walk theory

In 1973, Burton Malkiel released a monumental book known as "A Random Walk Down Wall Street" in which he mentions how stock prices work as random walks, i.e. their current and past prices do not affect the future, i.e. they are non-deterministic in nature.

Given this information, using index values as-is will not help us. To mitigate the problem, what the model will actually receive as input would be a set of technical indicator values over a given period of time. As the model learns the relationship that these indicators have for given index values, I do belive it can understand the underlying trend the data is following.

Methodology behind the solution

In this section, I would like to breifly list out the steps taken out for the solution:

- My first idea was to use the zipline package to source data from the Yahoo Finance API, but given how that particular endpoint of the API is deprecated I downloaded a CSV from the Yahoo finance website and used it. This file contains the date, opening, closing, low, high, and adjusted closing prices for the index
- Create values for the selected technical indicators using the datapoints received in the previous step
- Split the data into test and train datasets
- Build model and check for performance on the basis of the metrics that will be determined in the next section

Metrics

I belive the fundamnetal problem at hand is one of regression and for regression models, predicting the Mean Absolute Error (MAE) is a good metric to determine the performance of the agent. To define MAE, it is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The formula is as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |predicted_i - correct_i|$$

In this case, the MAE calculates the mean difference in values over the predicted and correct values. A high MAE shows that the model is not working the way it should and has not learnt the trend of the underlying data. A low MAE shows the very opposite of that, i.e the model has learnt the trend of the underlying data and fit itself well to it.

II. Analysis

Data Exploration

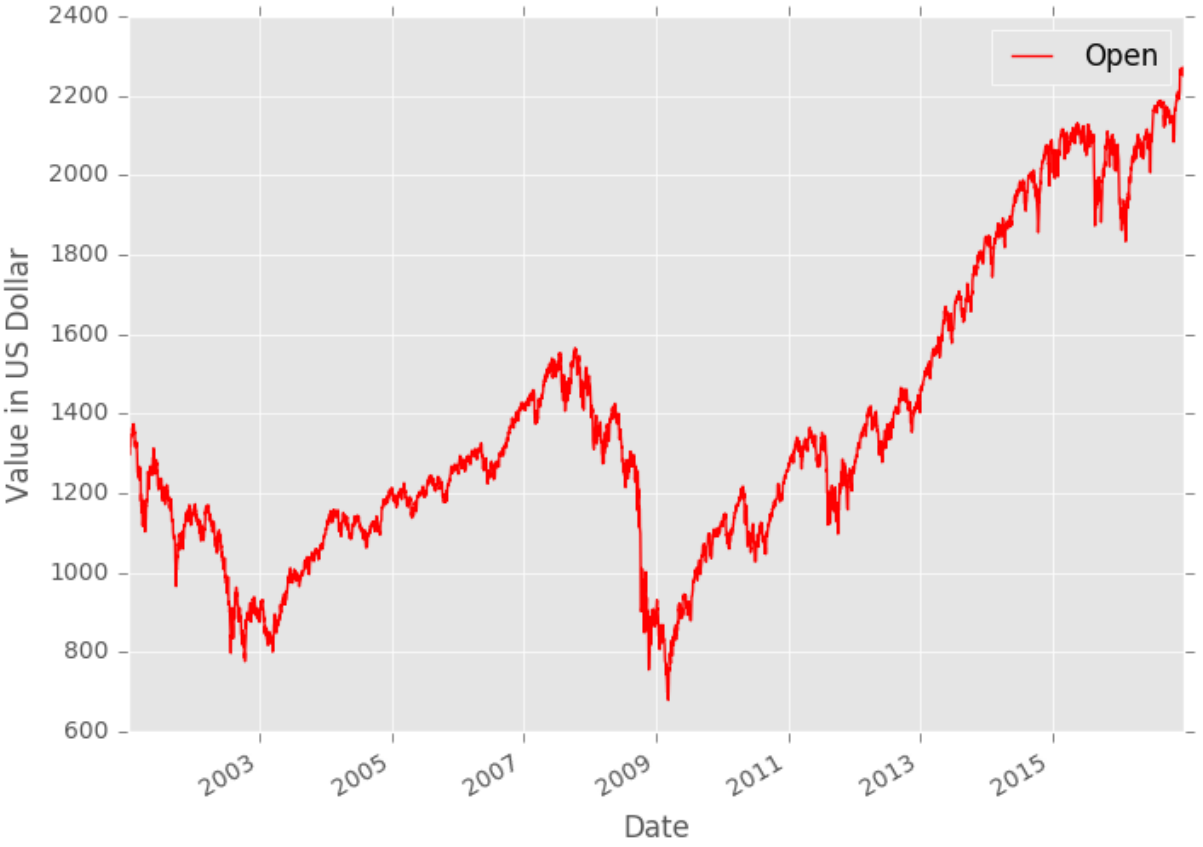
The data being used in this project is sourced via the Yahoo Finance API. Since the API in itself is not well documented and several parts of it have been depracted, after a certain amount of googling I found a page that still works and provides a CSV for a given date range in this case [1 Jan 2001 - 1 Jan 2017].

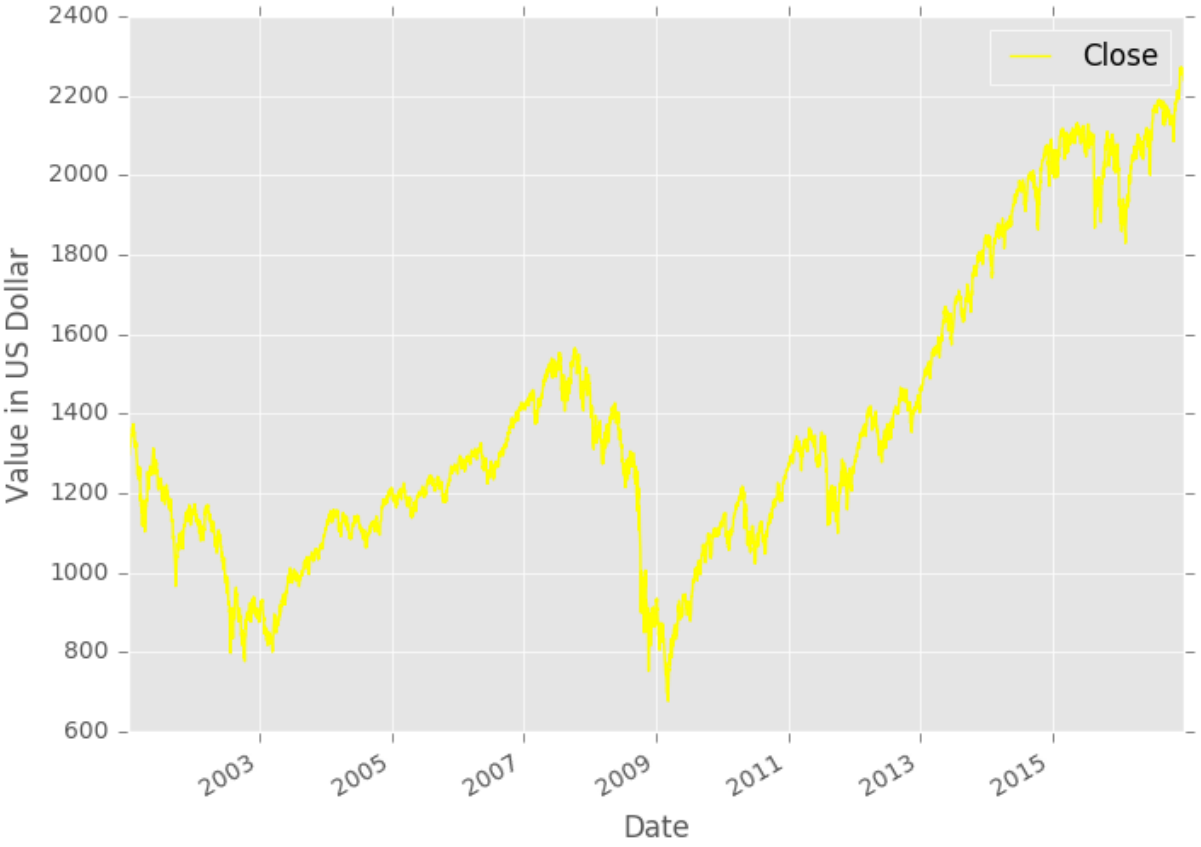
	Open	High	Low	Close	Adj Close	Volume
count	4025.000000	4025.000000	4025.000000	4025.000000	4025.000000	4.025000e+03
mean	1372.709349	1380.982987	1363.801700	1372.926283	1372.926283	3.139891e+09
std	372.095835	372.038163	372.151102	372.250755	372.250755	1.531072e+09
min	679.280029	695.270020	666.789978	676.530029	676.530029	3.560700e+08
25%	1119.359985	1125.500000	1112.670044	1118.859985	1118.859985	1.674500e+09
50%	1275.650024	1282.739990	1266.739990	1275.530029	1275.530029	3.177710e+09
75%	1525.750000	1532.459961	1519.750000	1525.750000	1525.750000	4.051570e+09
max	2270.540039	2277.530029	2266.149902	2271.719971	2271.719971	1.145623e+10

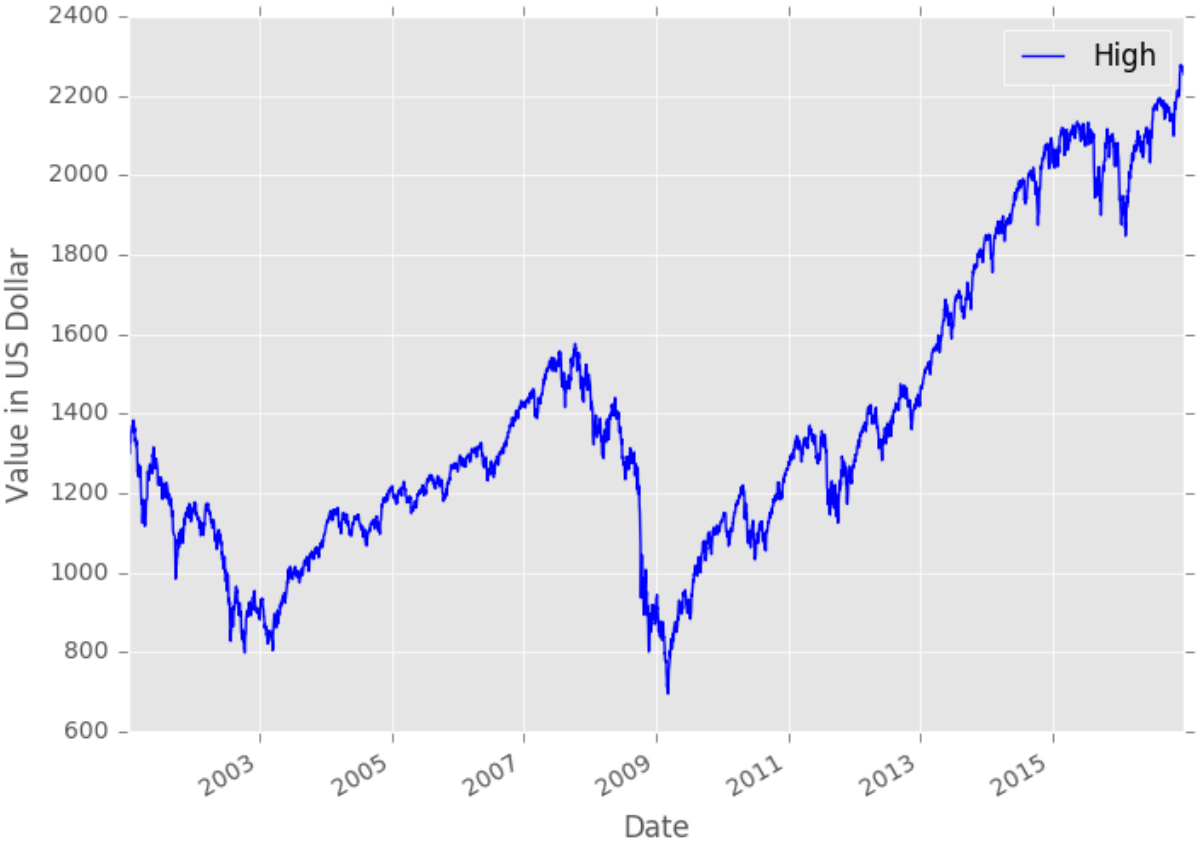
As mentioned before the CSV has the following columns: * Date: The date the below parameters were observed * Open: Price of the stock [in our case value of the index] in the beginning of the day
* High: Highest recorded price/value recorded in that day of trading
* Low: Lowest recorded price/value recorded in that day of trading
* Close: Price/value recorded at the end of the day * Adjusted Close: Adjusted close price for stock splits/joins in the day [note: this feature is more important when considering specific companies and not entire indexes such as the S&P 500] * Volume: Total number of shares traded in that day

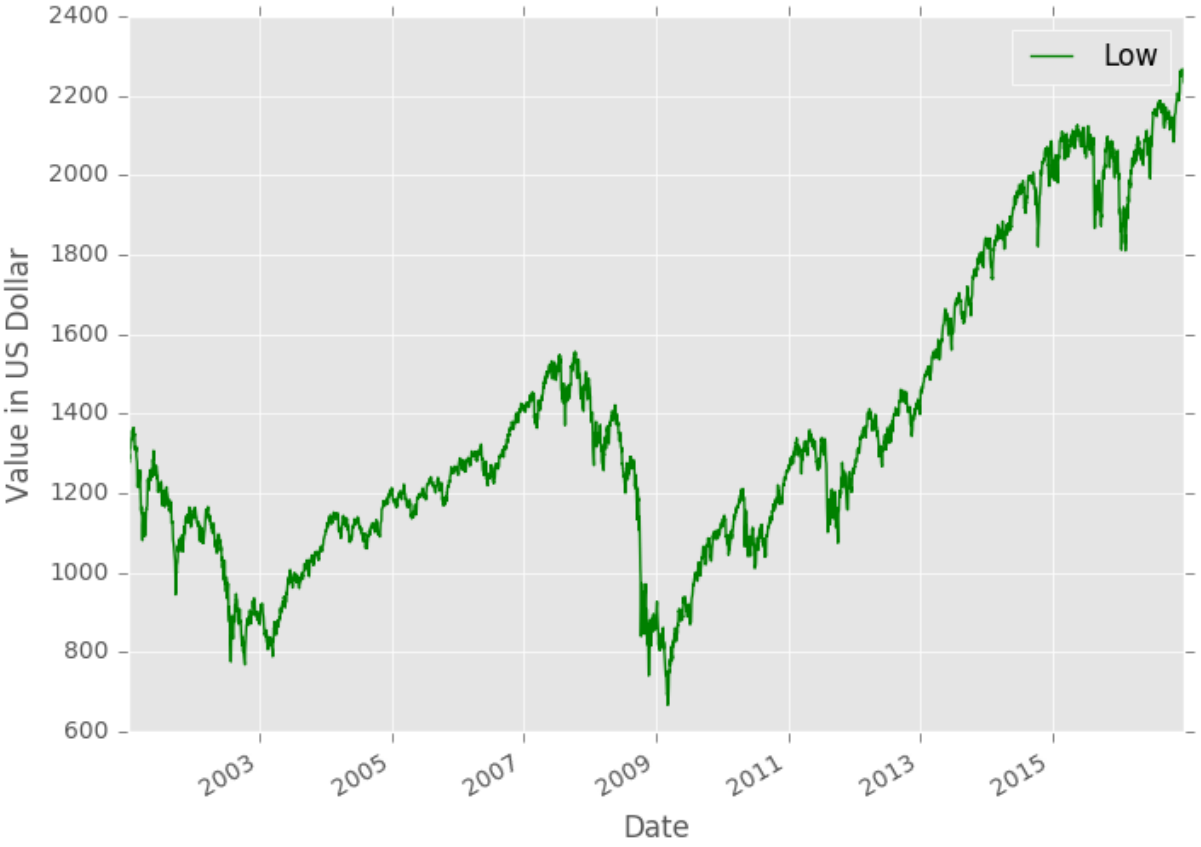
Exploratory Visualization

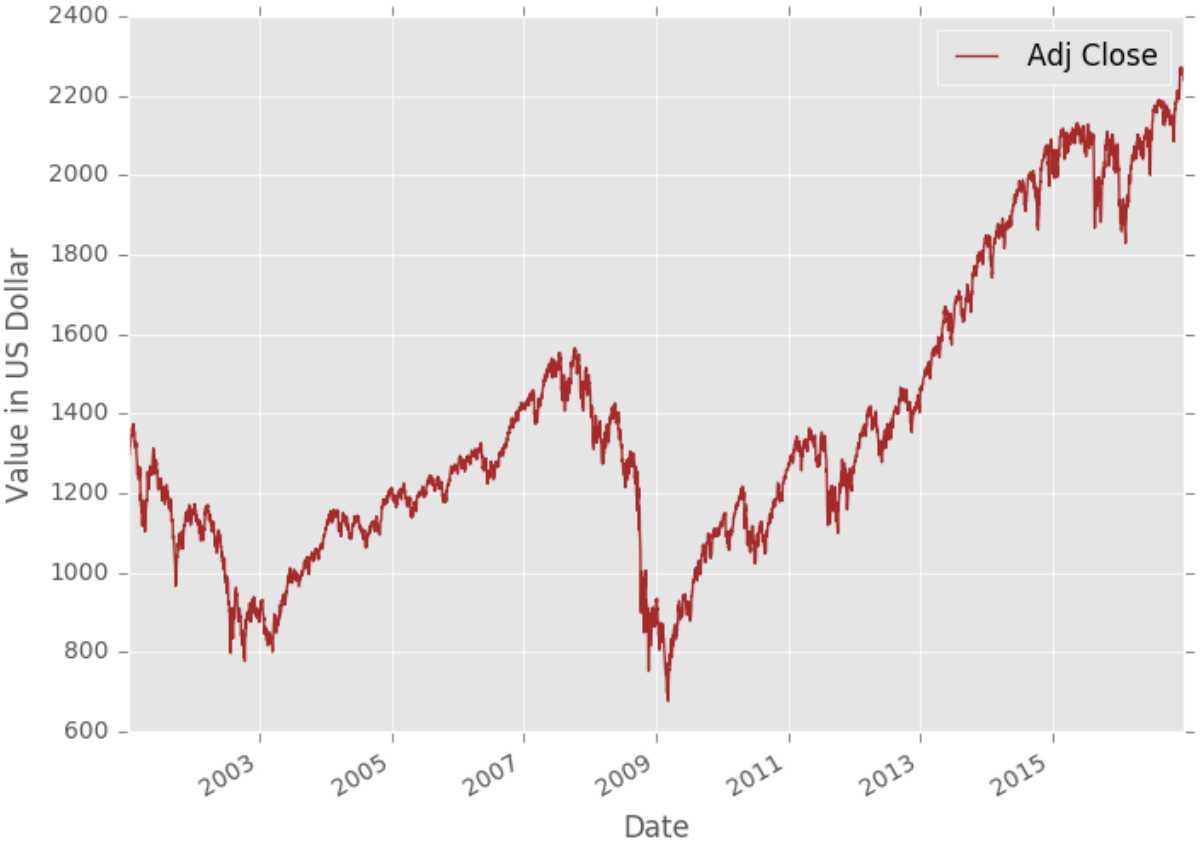
Below are plots of all the price points that we get from historical data csv file:

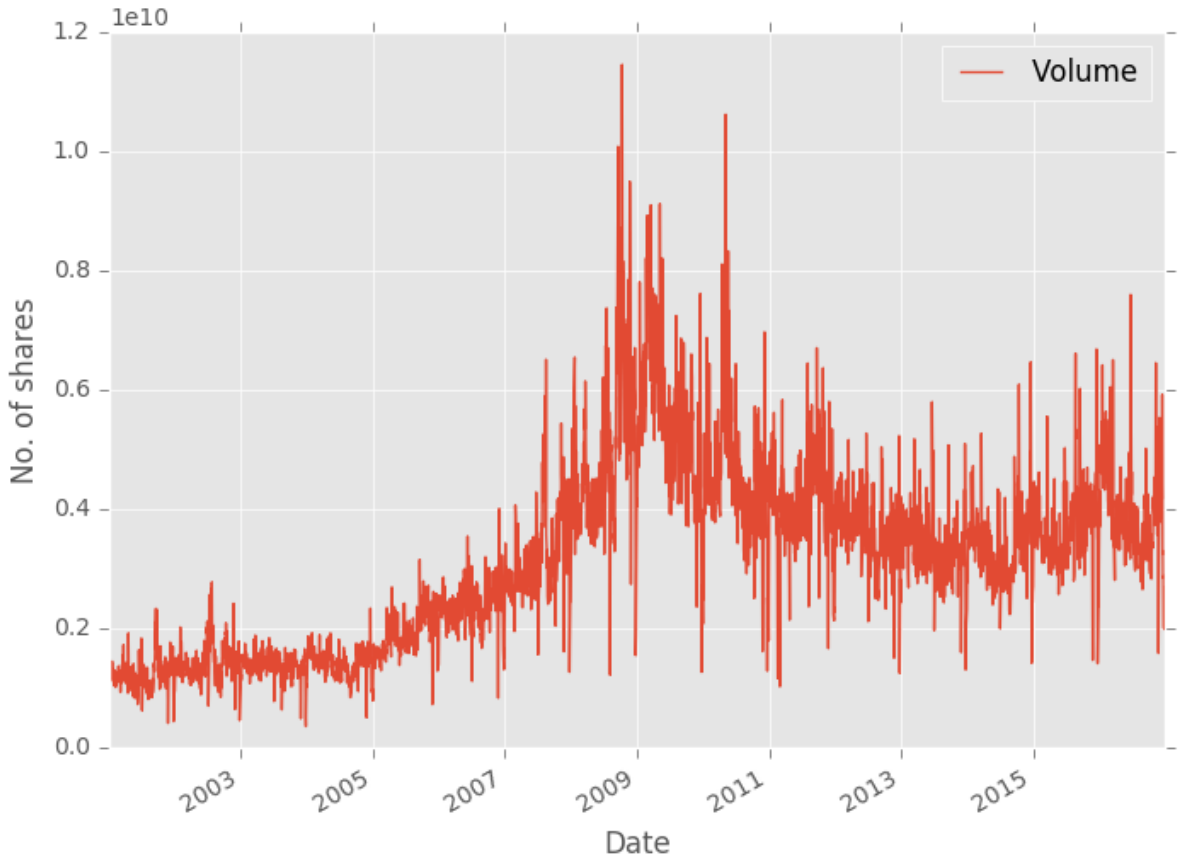

Open
Value in US Dollar
2400
2200
2000
1800
1600
1400
1200
1000
800
600
2003
2005
2007
2009
2011
2013
2015
Date


Close
Value in US Dollar
2400
2200
2000
1800
1600
1400
1200
1000
800
600
2003
2005
2007
2009
2011
2013
2015
Date

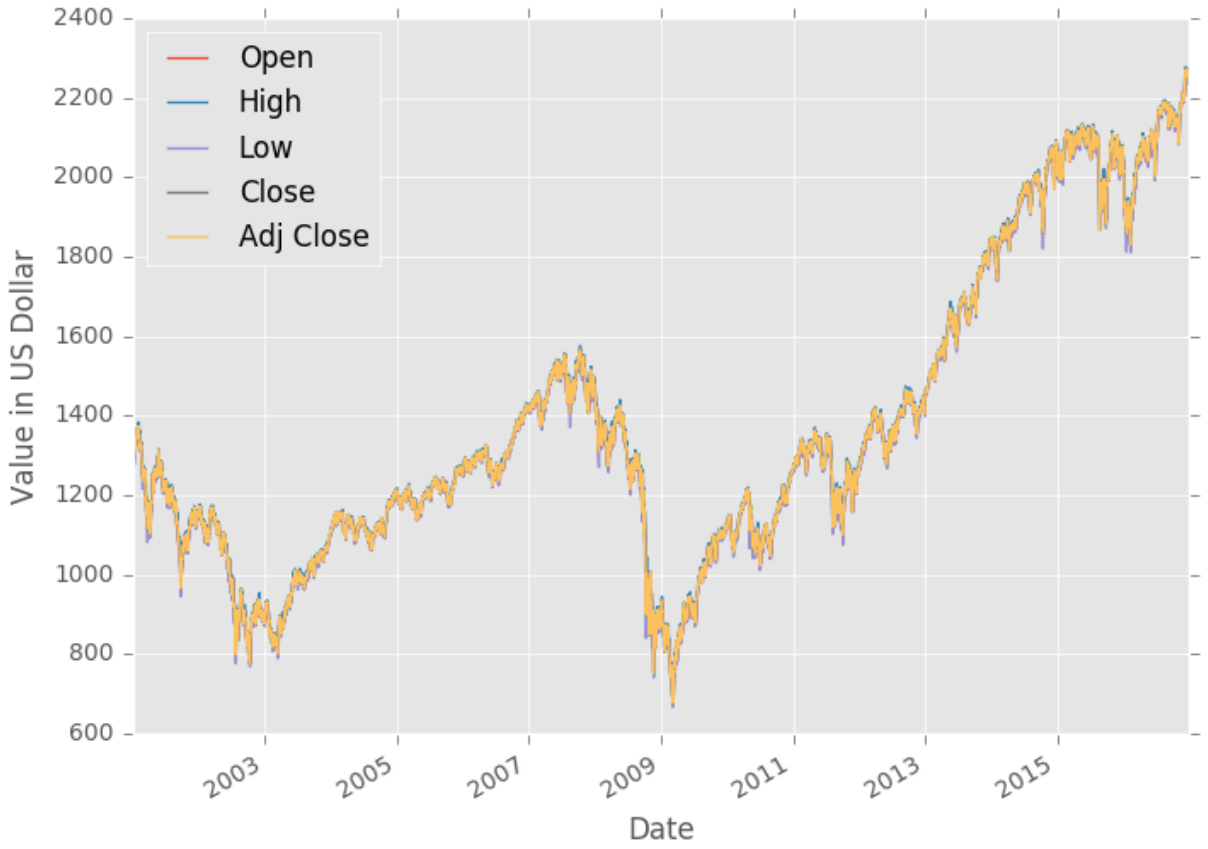

High
Value in US Dollar
Date
2400
2200
2000
1800
1600
1400
1200
1000
800
600
2003
2005
2007
2009
2011
2013
2015


Low
Value in US Dollar
2400
2200
2000
1800
1600
1400
1200
1000
800
600
2003
2005
2007
2009
2011
2013
2015
Date


Adj Close
Value in US Dollar
2400
2200
2000
1800
1600
1400
1200
1000
800
600
2003
2005
2007
2009
2011
2013
2015
Date



All the features together:



An interesting observation is that all the values of the individual features are quite similar. The reason for this is that in a stable index such as the S&P 500, individual parameters don't stray afar from each other, as a whole the index does well or poorly. It's worthwhile to note the spike in the graphs around 2008-2009 which corresponds to the global economic meltdown, very low volumes for very high prices.

Algorithms and Techniques

Given that this is a regression problem, I would like to use linear regression as my learner for a given input. The input itself will be a set of technical indicators. The dependant value, or the value that needs to be predicted would be the 'price' variable. Linear regression helps us model the relationship between a scalar dependant variable [y] and a set of explanatory variables [X]. If the learner is successfull, it would understand the trend of the underlying data and the corresponding MAE would be small. For further refinement and improvement I would like to try a SVM implementation.

Benchmark

Luckily, there exists an industry standard that monitors the fluctuaions in the value of the 'price' component of our data. It is the CBOE VIX [Chicago Board Options Exchange Volatility Index] value. The VIX value changes everyday and is a benchmark to analysts and traders as to how much

volatility one might see in the data for that trading day. From the CBOE's website, I downloaded VIX values from 1998-2017. Computing an average VIX value on these inputs, I got the value as : 14.82. Thus, the goal for the linear regression learner is to predict prices that are better/lower than the average VIX value. To carry out this comparision, the metric defined above, the MAE will be useful.

III. Methodology

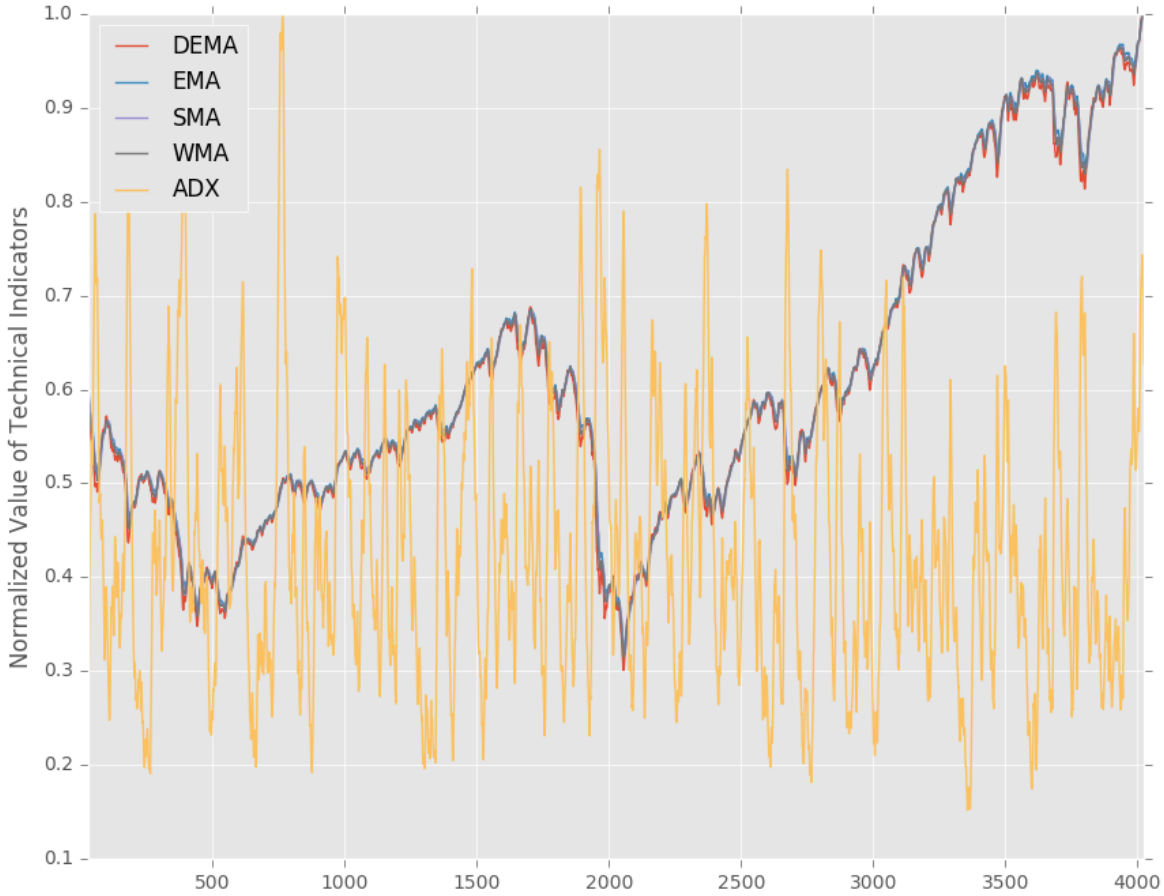
Data Preprocessing

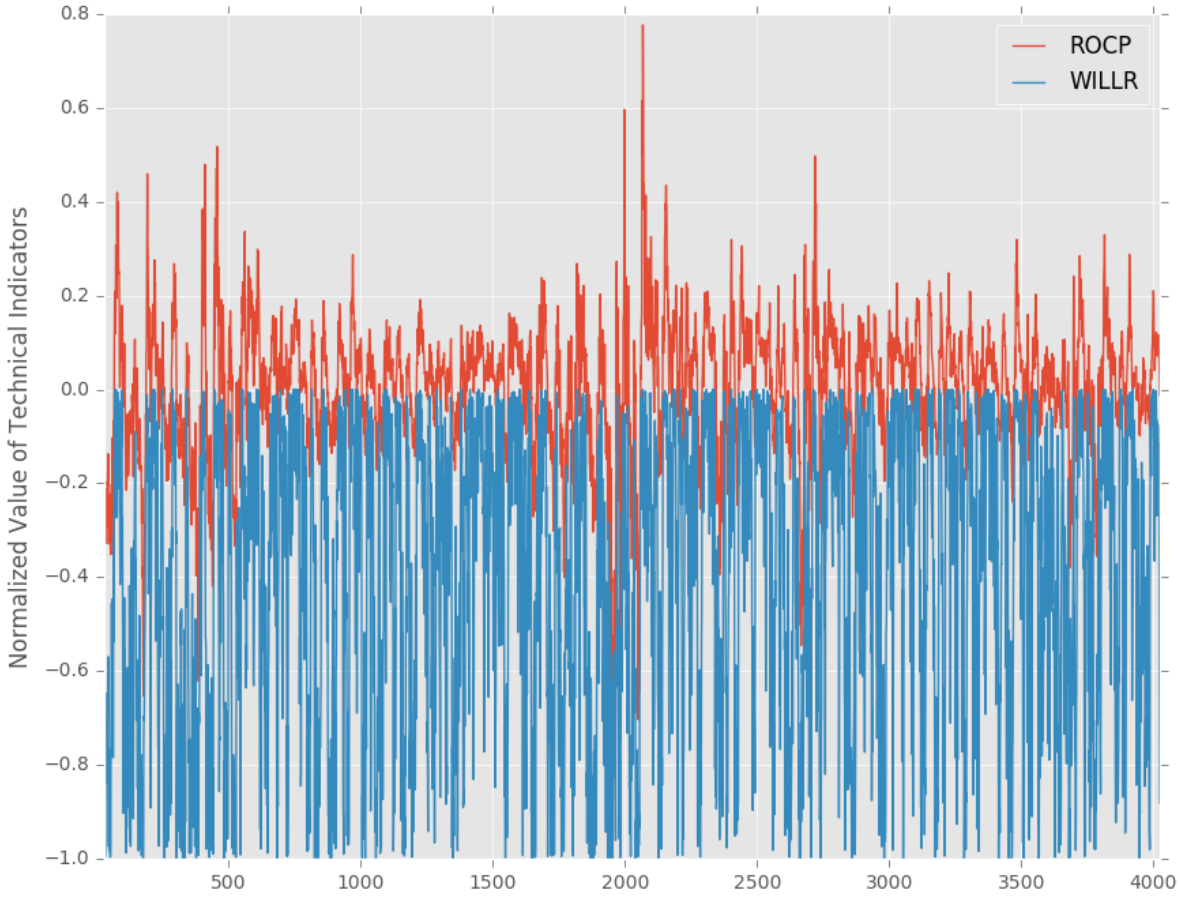
The first step in data preprocessing would be to parse the historic data into the selected technical indicators which can then be used for prediction. Following is a summary of the technical indicators being used:

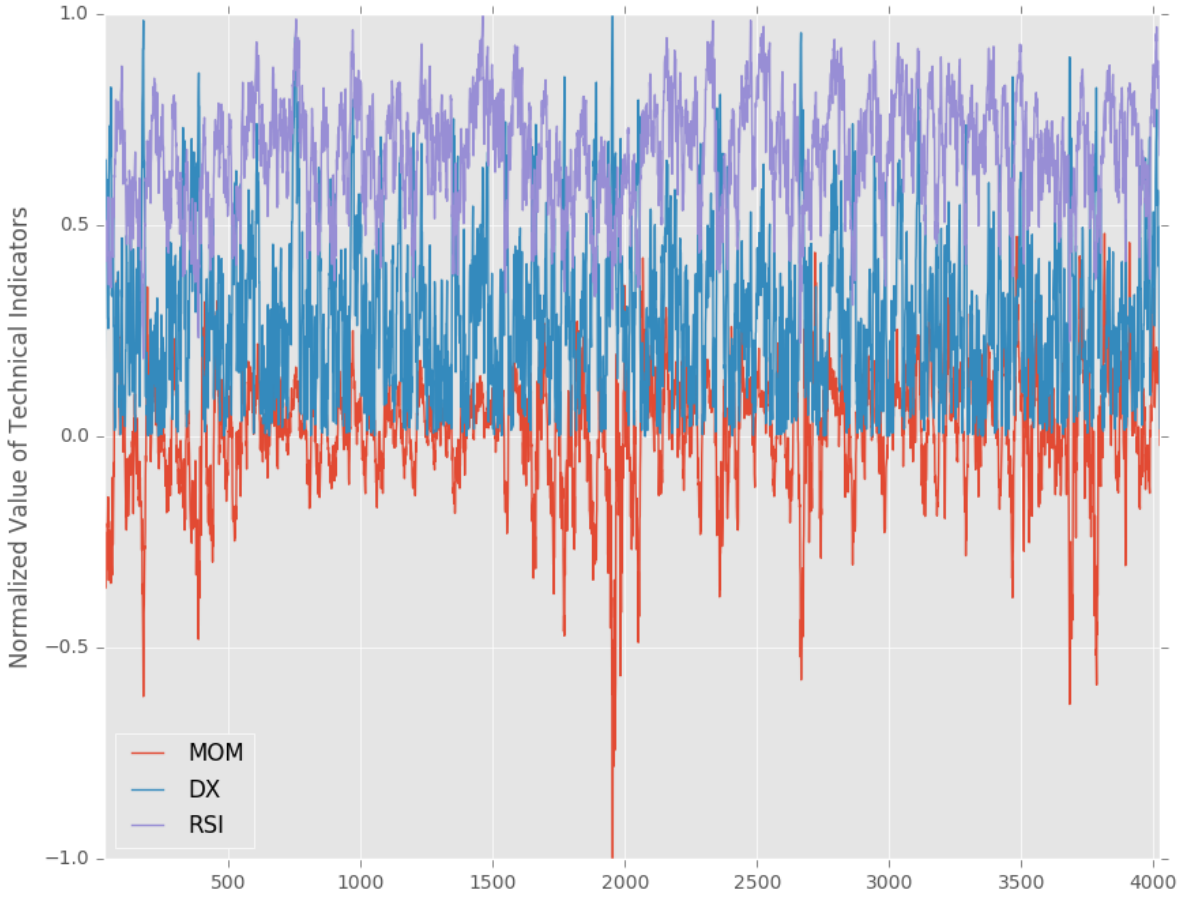
- Momentum: It measures the securities rate-of-change. Like in physics, it is a measurement that provides details of the acceleration/deceleration of the stock in question. On the basis of whether the speed is upwards/downwards bullish/bearish interpretations might be made
- Relative Strength Index: The RSI creates a measure of value between 0-100, key points being above 70 and below 30. On the basis of these two keypoints one can determine whether the stock is overbought or undersold. If a particular stock is above 70, it can be considered as overbought, similarly for a stock below 30, it is considered to be undersold
- Directional Movement Index: DX is used for identifying when a definable trend is present in an instrument. That is, the DMI tells whether an instrument is trending or not.
- Simple Moving Average: It is an arithmetic moving average calculated by adding the closing price of the security for a number of time periods and then dividing this total by the number of time periods. This is important to traders, as when short-term SMAs cross long-term SMAs it gives them a signal to buy/sell
- Exponential Moving Average: It is similar to a simple moving average, except that more weight is given to the latest data, this gives traders a better idea about day-to-day nuances but can also give false buy/sell signals, which is why it is used alongside SMA
- Weighted Moving Average: Similar to EMA, however WMA assigns a unique weight to every element in the period, not just the latest observations. Additionally, the sum of all weights is 1
- Double Exponential Moving Average: It is an attempt to reduce the amount of lag time found in traditional moving averages. It is a composite implementation of single and double EMAs producing another EMA with less lag than either of the original two
- Average Directional Index: is an indicator used in technical analysis as an objective value for the strength of a trend. ADX is non-directional, so it quantifies a trend's strength regardless of whether it is up or down. Higher the number, stronger the trend, lower the number, weaker it is
- Rate of Change Percentage: It is a momentum indicator that measures the percentage change in price between the current price and the price n periods in the past
- Williams %R: It has values on an inverted scale, i.e. 0 at the top and 100 at the bottom. Similar to RSI, it has two keypoints, 20 and 80. Stock values below 20 are considered overbought and those over 80 as oversold

After selecting all the technical indicators, they're normalized between -1 and 1 to bring all the

values to the same scale. Following are plots of multiple indicators post normalization:




ROCP
WILLR
Normalized Value of Technical Indicators
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
500
1000
1500
2000
2500
3000
3500
4000



As it can be seen, normalizing these values or using technical indicators instead of the input historic data has not removed the underlying trend that can be seen [one can still see major peaks and troughs in the 2008-09 period]

Implementation

It is time to implement the linear regression learning model. The linear regression formula used will be simple:

$$y = \sum_{i=1}^{n} k_i x_i + b$$

In this equation, y is predicted price, xi is the technical indicator and ki and b are co-effcients that need to be tuned to get the right value.

Once the regression model is fixed, I would split the data into two parts, the training set and the test set. The need to do this as has been made clear during the nanodegree is to overcome the problems of overfitting and to check how accurate our predictions are.

Refinement

The previous section saw the implementation of a simple learning model using linear regression. In this section, I will buid a new learning agent using SVM. A major difference in my approach to SVM vis-a-vis the linear regression model would be the selection of hyperpaarameters. The best way for to choose hyperparameters for the model are via cross-validation I feel. In cross validation one creates ultiple train-test sets or folds and runs the model on many combinations of these folds. One point that needs to be kept in mind here is that during the course, cross-validation was used by running random training data on the model, however, given the time sensitive information held by this dataset random selection is not possible. To mitigate this issue for every nth fold, the train set should run on 1..n and the test set should run on the n+1th fold. For example :

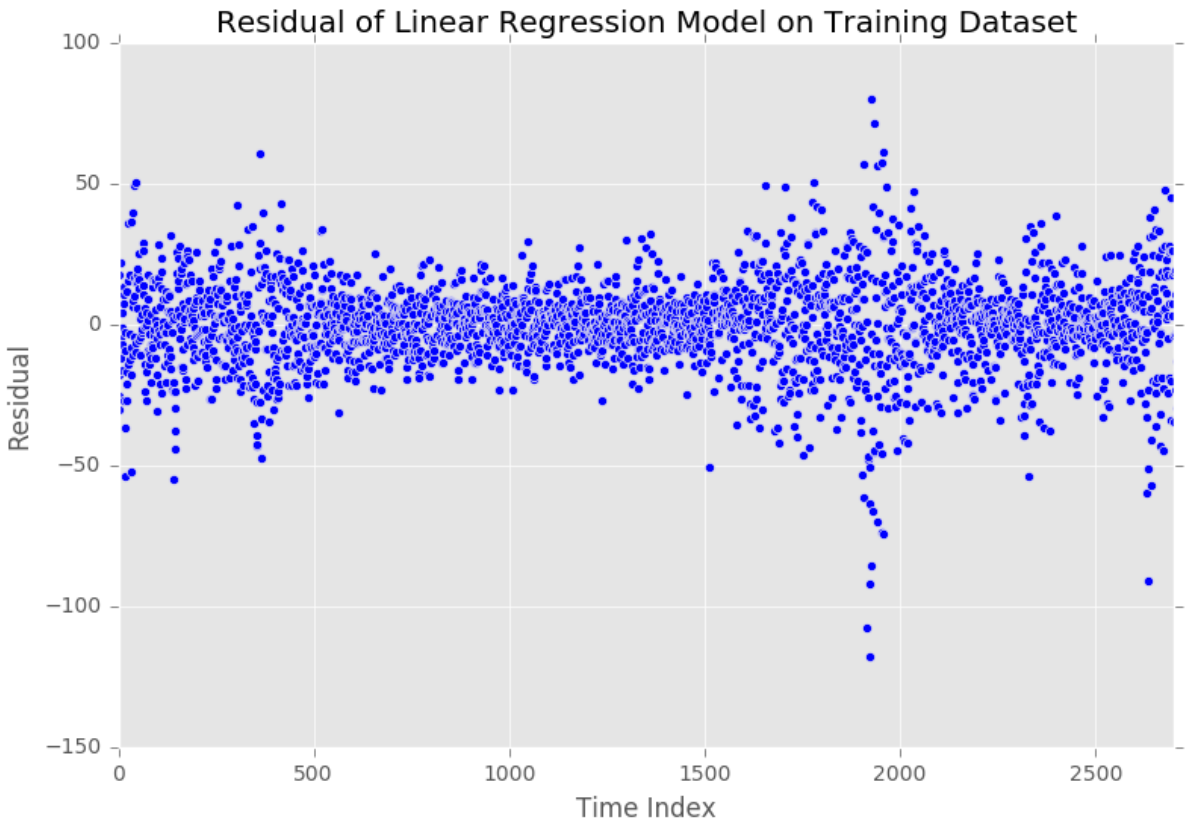
If there are 3 folds,

- Train should run on fold 1, fold 2
- Test should run on fold 3

IV. Results

Model Evaluation and Validation

As has been taught during the course, before running a linear regression, it is a good idea to check the residual plot. If the residual plot is tightly clustered, regression becomes more accurate versus a loosely clustered plot where prediction would be tougher due to the spread of the data.



As the residual sends a positive signal to us, I ran the regression model on the testing dataset and I received a MAE of 13.10, which is lesser than our benchmark MAE which is 14.82. For SVM regression, a similar process as linear regression is followed the additional steps being the quirk mentioned above that consecutive folds must be used. With this is in place, the SVM regression had an MAE of 45.54. Compared to the benchmark MAE, this performs poorly and my assumption that a more complex model would understand the underlying trend of the data better is wrong.

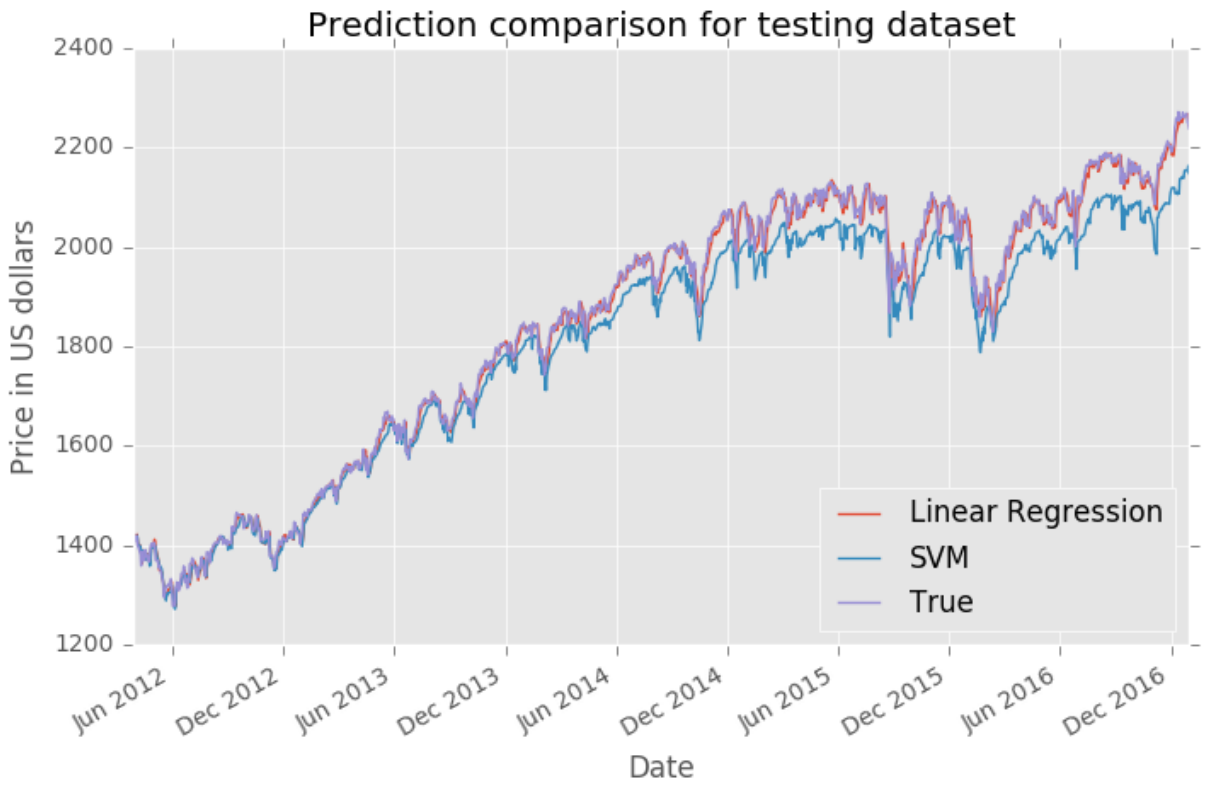
Justification

As the previous section showed, a complex model such as SVM could not follow the underlying trend of the data and could not acheive a low enough MAE. Another justification can be that my tuning of the C parameter in the SVM was not upto the mark which is why the poor result was seen.

V. Conclusion

Free-Form Visualization

In this section I have added the visualization for the true historic data, linear regression curve and SVM model. As it can be seen, the linear regression model closesly follows the true historic data, whereas the SVM model is off by a little.



Reflection

In this project, a supervised learning model is built for stock prediction. I used Yahoo's finance API to get the required historical data [16 years in total]. For my metrics, I used the average closing VIX value released by the CBOE. The input stock data was transformed into values using technical

indicators which were inputs to the prediction model. Models tried in this project were: * Linear Regression * SVM As can be seen from results in the previous section, the linear regression model is closer to the original data versus the SVM model. Stock prediction by its nature is a hard problem I believe. A previous notion that I had before the capstone project, which I now think is rather naive, was that given enough data, building an adequetly performing model is easy. Post this project, I disagree with this point of view. The stock market and the S&P index in particular has a lot of data however given the time specific nature of the data, it cannot be used as :

- Input directly
- Without the use of technical indicators

Another learning that I had during this project was the importance of the data pre-processing step. In other projects during the course of the nanodegree, the data pre-processing step was very clearly layed out and easy to understand. In this project, learning about the technical analysis method that many analysts use in their day to day life was an interesting point, as it made me realize that for models to accurately understand the data, it needs to be formatted in a way that reduces the noise and adds useful information.

Improvement

An idea I'm interested in exploring is the possibility of using boosting in this problem. The reason this idea seems interesting is because a number of technical indicators add a different value to the underlying data and a weak learner associated with each technical indicator and together they might increase the efficiency of the base learner.

VI. References

- Techinal Analysis
- Fundamental Analysis
- Quantitative Analysis
- Mean Absolute Error
- Techical Indicators
- Exponential Moving Average
- Simple Moving Average
- Double Exponential Moving Average
- Directional Moving Index
- Cross Validation of Time Series Data