STAT 542 Final Project Report

Machine Learning for Trading

Sharan Subramaniyan (ssbrmny2)
Tarun Chhabra (tchhabr2)
Meiruo Xiang (mxiang3)

Department of Statistics (MS)
University of Illinois at Urbana-Champaign

I. Introduction

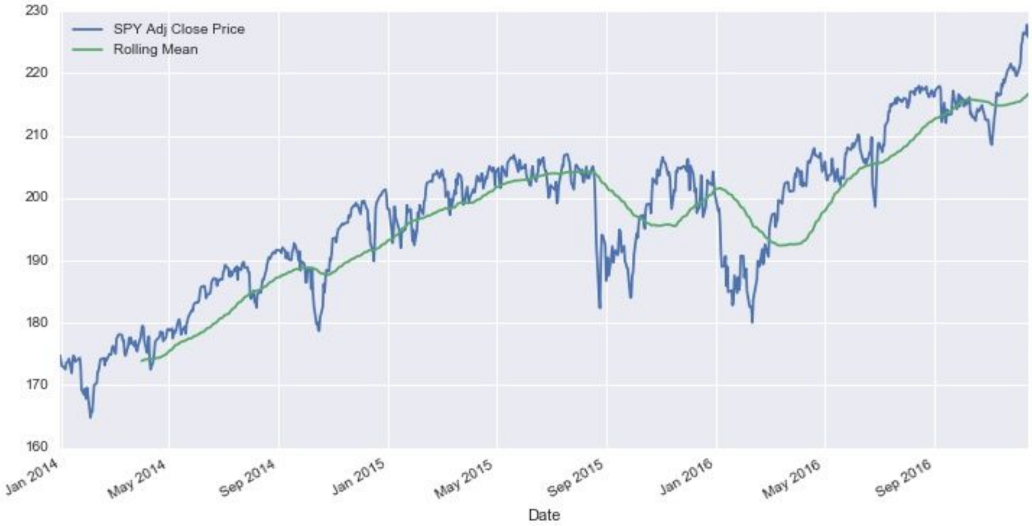
The objective of this project is using machine learning algorithm to find a model or set of rules that generate buy/ sell signals in the market. This model will give an edge in the market if followed with good risk management and execution. In our research, we used regression methods to predict the daily returns and used daily trading strategy to gain edge over the market and formulated another strategy using the classification method to predict the level of daily return as a buy/sell signal.

II.Feature Engineering

Used OHLCV (open, high, low, close and volume data) data for 1080 days from google finance to create various financial indicators as the predicting variable. Features used for different strategies are a subset of the overall features created.

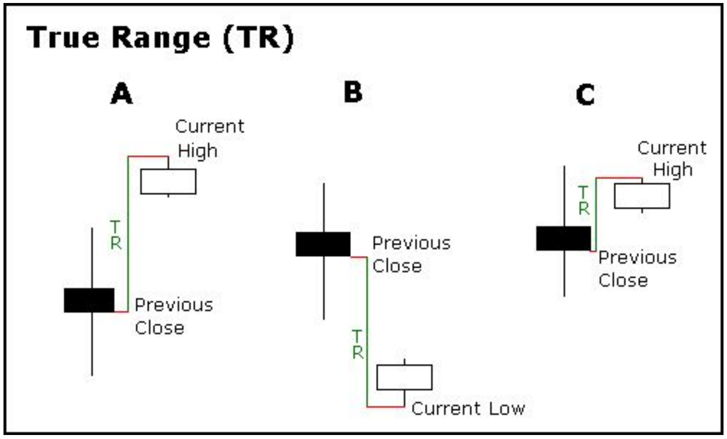
1. Feature Creation

(1). **Moving Averages**: is a calculation to analyze data points by creating series of averages of different subsets of the full data set. It is the rolling mean defined by a “window” and as new data becomes available, old data is dropped and new data is included. It is also called a moving mean or rolling mean.

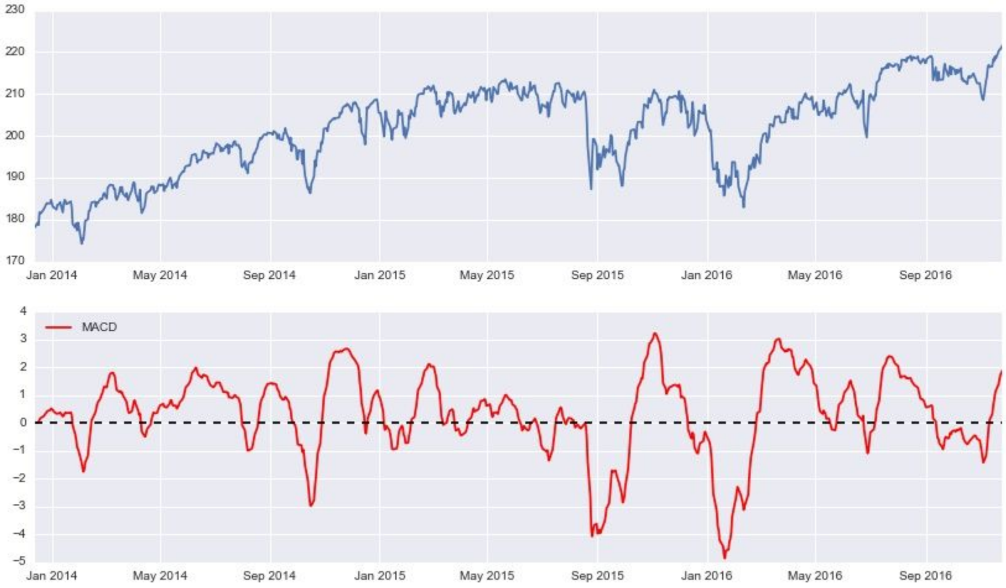


Graph representing Adjusted Close Price for SPY stock and it's Rolling mean with 63 day window. It can observed that there is no value for first few days for Rolling mean which was expected as first 63 days would not have any values for Rolling mean.

(2). **Average true range**: is an N-day smoothed moving average of the true range values. It is a measure of price volatility.



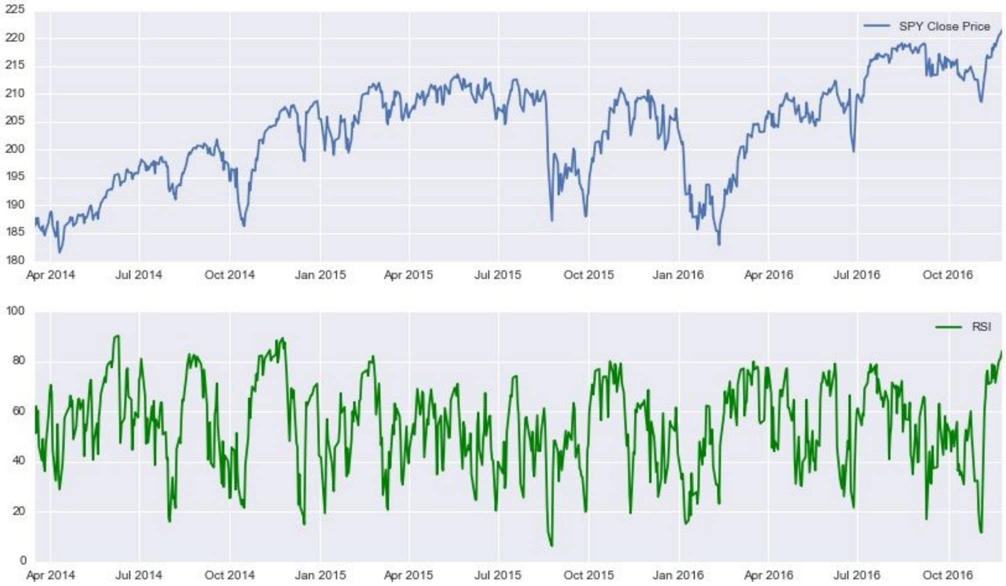
(3). **MACD** (Moving average convergence divergence): is a trend following indicator. It measures the difference between a fast and a slow moving average. Average of the MACD line and a histogram of the difference between the above two are also considered.



(4). **RSI** (relative strength index): is a technical indicator used in the analysis of financial markets. It is intended to chart the current and historical strength or weakness of a stock or market based on the closing prices of a recent trading period.

$$RSI = 100 - \frac{100}{(1+RS)}$$

Where RS = Average gain / Average loss for the selected period (eg. 6 days).

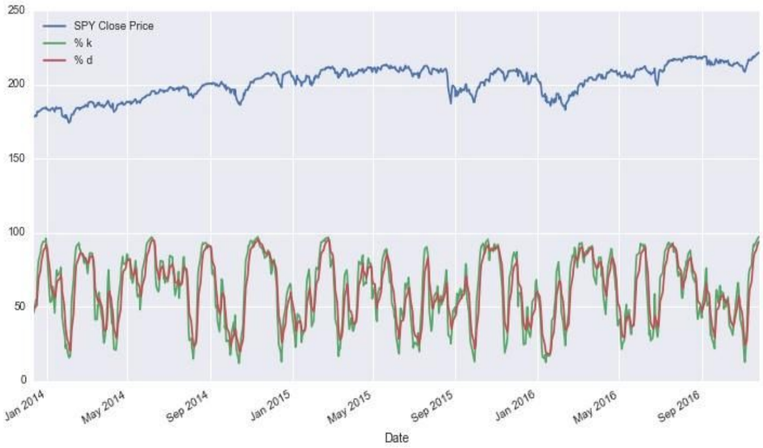


(5). **Stochastic Oscillator**: is a momentum indicator that uses support and resistance levels. This method attempts to predict price turning points by comparing the closing price of a security to its price range.

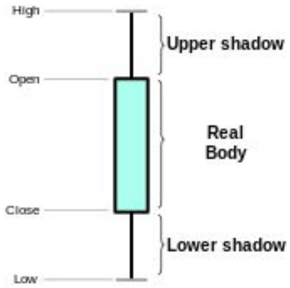
The 5-period stochastic oscillator in a daily timeframe is defined as follows:

$$\%K = \frac{(Price{-}L5)}{(H5{-}L5)}, \quad \%D = 100 * (\frac{(K1{+}K2{+}K3)}{3})$$

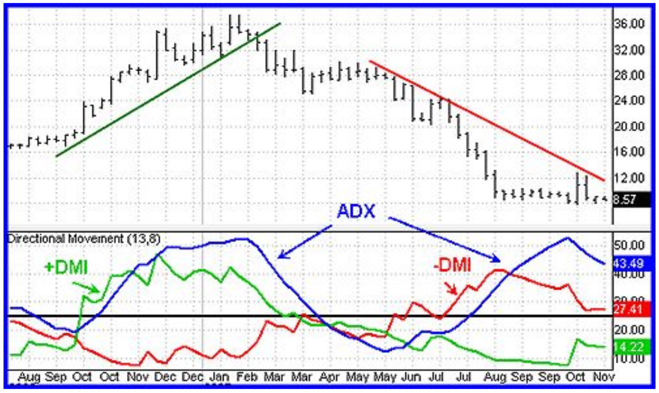
where H5 and L5 are the highest and lowest prices in the last 5 days respectively, while %D is the 3-day moving average of %K (the last 3 values of %K).



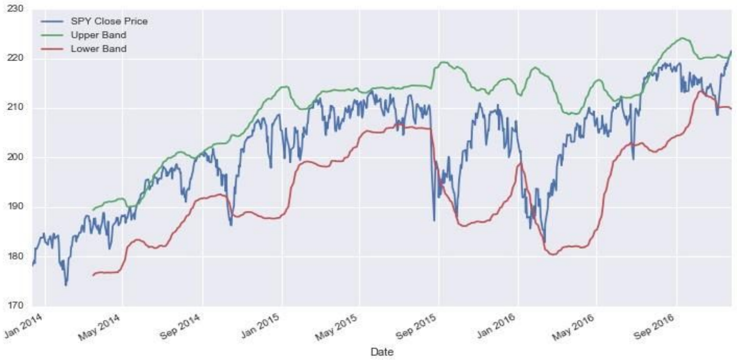
(6). **Candle Patterns**: Price patterns are useful as features, since they reflect herd psychology about the market, they can be single or multiple day patterns. Here we use small range day and large range day.



(7). **ADX** (Average directional index): the positive directional indicator (abbreviated +DI) and negative directional indicator (-DI). The A.D.X. combines them and smooth the result with a smoothed moving average.

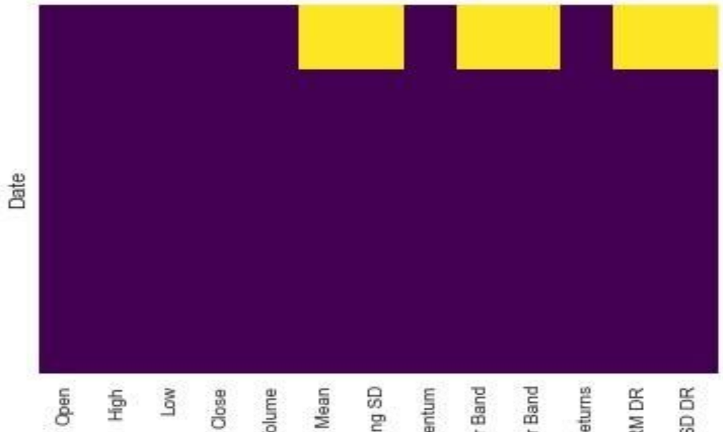


(8). **Bollinger Bands**: are volatility indicators, which form a channel around the prices. It consists of a moving average and an upper and a lower bands which are at a distance of k standard deviations away from the moving average.

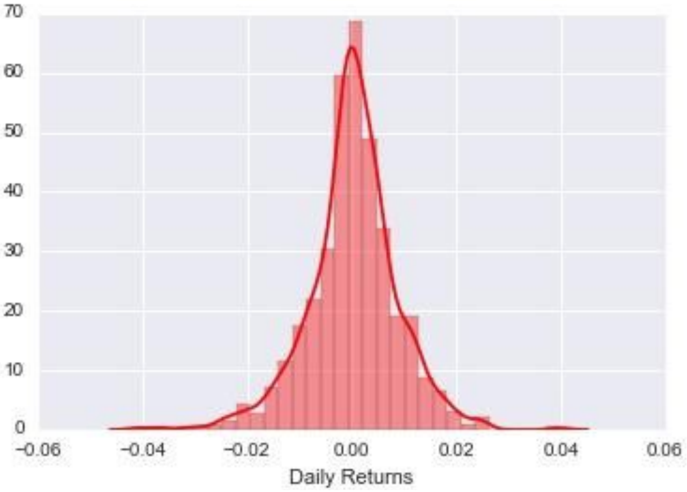


2. Data Exploration

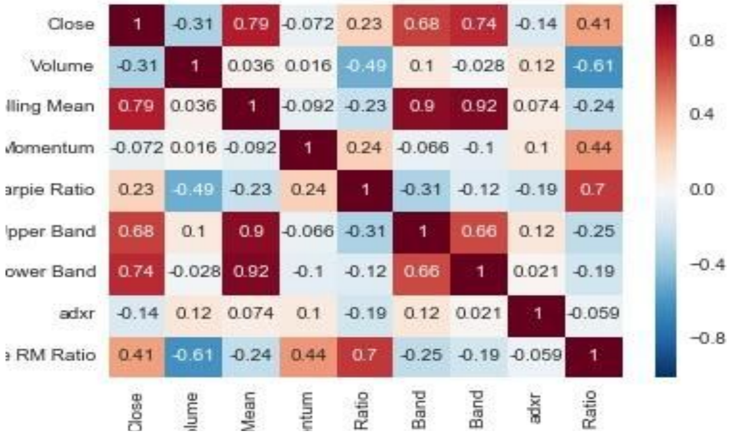
The basic exploration of our data shows: Average up day = 0.421 % and Average down day = 0.413%, which means the extent of move is on average about the same for up and down days. And it also implies there is a bias towards the long side (ie. System must be long more than it is short).



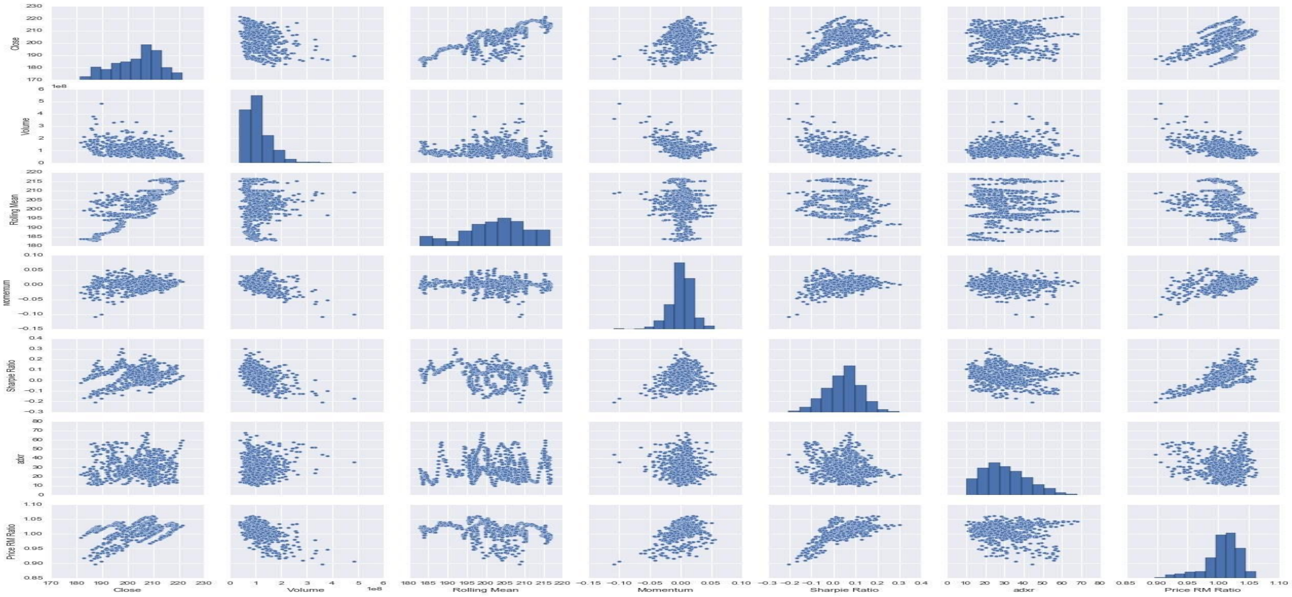
The heat map above shows the predictors that have missing values. Clusters of missing values at the top indicates that those predictors are not yet defined for that part of the data. Which prompted us to subset the data after the indicators were created.



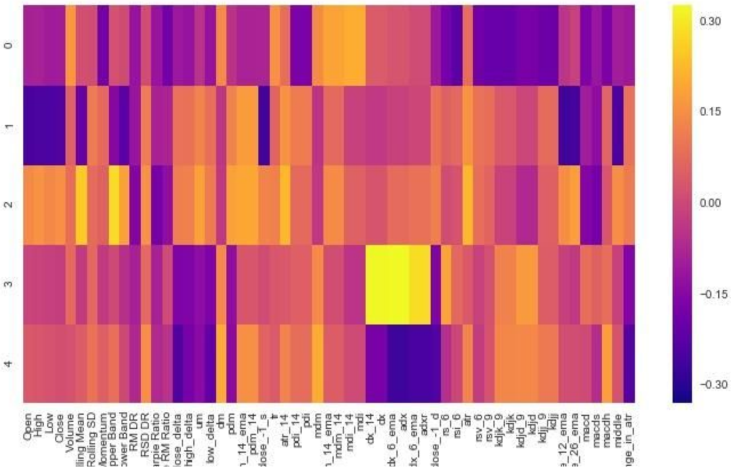
Above graph is the distribution of Daily Returns, according to the plot, the returns are more or less normally distributed. However, the market is notorious for outliers, which makes distribution actually have very fat tails, and this is not captured in this sample.



The heatmap of correlations between variables gives an idea of how the various predictors are related – ideally we want as many predictors as possible that are not correlated.



The scatterplot matrix gives the change of a predictor with another predictor.



We used principal Component analysis to reduce the number of predictors from 50+ to 5 principal components, which will be used in the following research. The plot above shows how each of the original predictors can be represented in terms of the 5 principal components. We are reducing the dimensions for LDA and QDA Classification problem For regression problems, the predictors would be a subset of available indicators.

III Regression Modeling

1. Predictors and Response Variable

Some of the basic indicators have been used as predictors for regression models. The description of indicators and their significance has been described in the feature engineering section. Rolling Mean, Adjusted Closing Price, Rolling Standard Deviation, Momentum, Bollinger Coefficient, Sharpe Ratio and Price to Rolling mean ratio are used as the predictors. The response variable is next day's daily returns.

2. TRADING STRATEGY

The objective of regression analysis is to predict the next day's daily return. The trading strategy is to predict if the daily return on SPY on next day is going to be positive or negative and based on the prediction, position is taken at the start of the day. So if prediction for next day is a positive return, long position is taken and if the prediction is negative, short position is taken. In order to select the best model, trading methodology known as Back testing is used.

3. Data Preparation

Normalization: Since values of predictors are at different scales, the first part of the data preparation involves normalization of the predictors. This is a critical step for regression analysis of trading data as otherwise some of the indicators may have huge influence on the prediction results which may not reflect the situation in the market.

Back Testing and Prediction Data: Data is divided into two parts, one for backtesting for selection of model and the other one for strategy implementation. Training data is 63 day window (data for last 63 days) to fit the model and testing is done for next 5 days.

4. Back testing

In order to test efficacy of regression model and to select the best model, back testing is used. Back testing basically means that historical data is used to fit the model and predictions are made using the fitted model, the algorithm which performs the best is chosen.

Stock markets are immensely volatile and hence in order to use machine learning for prediction of daily returns, we train the model on frequent basis as similar trends may have dramatically different impact on the returns in different time periods. Hence for our model for the training data, a 63-day window was used and the next 5 days were used to test the data. Two metrics were used to judge the efficacy of the model:

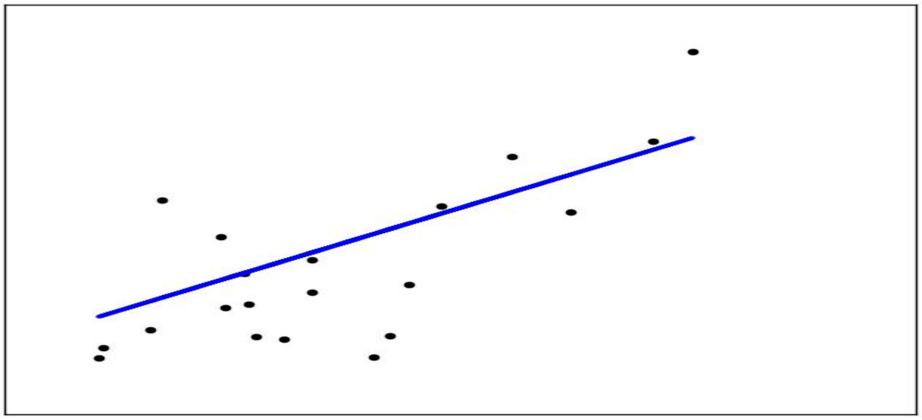
(1). RMSE – Root mean square error is one of the standard metrics used for selecting the regression models
(2). Score- Score metric refers to number of correct prediction of sign of the next day's return.

After training the model, and fitting the parameters, RMSE and score values are computed for next 5 days. The process is repeated several times for the whole training data. Finally, the using RMSE values for all the back tests, a mean value is computed and is used as a metric and sum of score values is taken for all the back tests. Model which has best combination of score and RMSE metrics is selected.

5. Regression Models and Results

(1). **Linear Regression**

LinearRegression, in its simplest form, fits a linear model to the data set by adjusting a set of parameters in order to make the sum of the squared residuals of the model as small as possible.



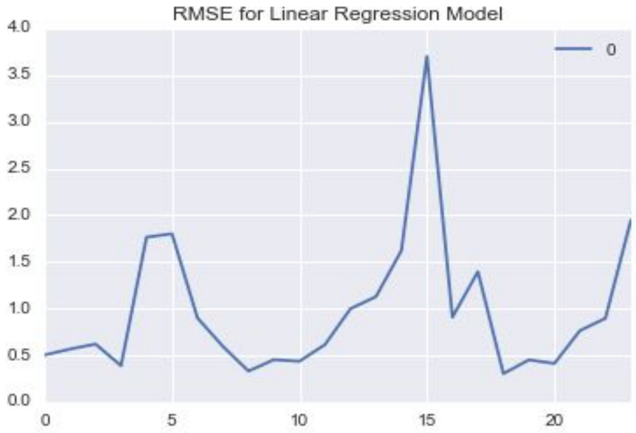
Linear models: $y = X\beta + \epsilon$

- X: predictors
- y: response variable
- β: Coefficients
- ϵ: Observation noise

Coefficients are computed by minimizing the RSS.

$$RSS= \Sigma\ (y_i -x_{i1}\beta_1 - x_{i2}\beta_2 - x_{i3}\beta_3 - x_{ip}\beta_p)^2$$

For this analysis, the model is fitted after every 5 days using the data of last 63 days, hence all the coefficients are computed on weekly basis to capture the changing trends in the market. Analysis is focused on prediction and not getting insight into impact of predictors on the response variables. Hence we don’t need to do diagnostics tests for normality assumption of the residuals and don’t need to untangle the highly correlated predictors.



Graph shows the values of RMSE values for different back tests. As we can observe the values vary from approximately 0.4 to 3.5

We use mean value of RMSE for model evaluation. Score metric finds the sum of all the individual back test scores.

MEAN RMSE	0.980339
SCORE	0

Mean value of RMSE is 0.980339 and total score is 0 and hence it doesn’t seem to be good model for our Trading Strategy.

(2). **Elastic Net**

The elastic net is a regularized regression method that linearly combines the L1 and L2 penalties of the lasso and ridge methods

.

LASSO- **lasso (least absolute shrinkage and selection operator)** is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. It was introduced by Robert Tibshirani in 1996 based on Leo Breiman’s Nonnegative

Garrote. Lasso was originally formulated for least squares models and this simple case reveals a substantial amount about the behavior of the estimator, including its relationship to ridge regression and best subset selection and the connections between lasso coefficient estimates and so-called soft thresholding.

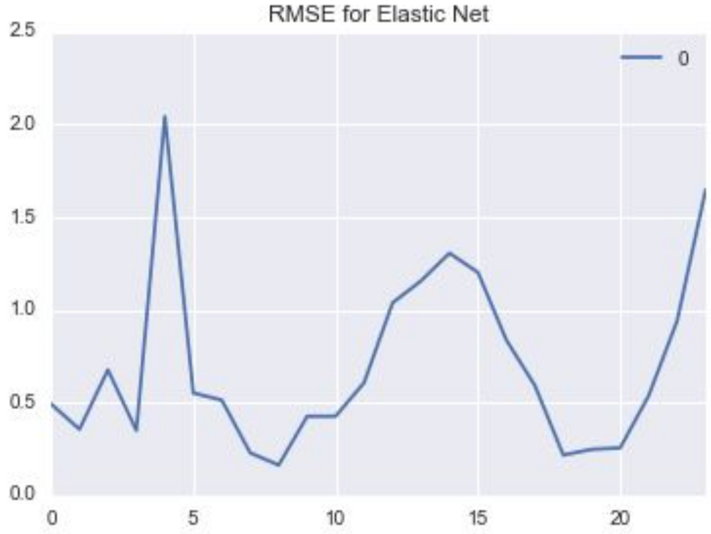
minimize $\quad$ RSS= $\Sigma\ (y_i - x_{i1}\beta_1 - x_{i2}\beta_2 - x_{i3}\beta_3 - x_{ip}\beta_p)^2$

subject to $\quad$ $\Sigma\ ||\beta_i|| \ <= s$

RIDGE- regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Also known Tikhonov regularization. Basically there is an added constraint to minimizing the RSS.

minimize $\quad$ RSS= $\Sigma\ (y_i - x_{i1}\beta_1 - x_{i2}\beta_2 - x_{i3}\beta_3 - x_{ip}\beta_p)^2$

subject to $\quad$ $\Sigma\ \beta_i{}^2 <= s$



Graph shows the values of RMSE values for different back tests. As we can observe the values vary from approximately 0.3 to 2

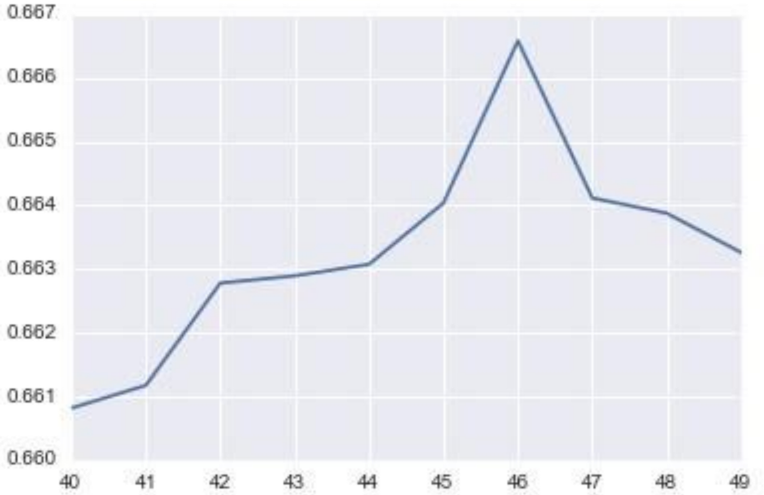
MEAN RMSE	0.70221
SCORE	10

Mean value of RMSE is 0.70221 and total score is 10 and hence it is one of the potential candidates for final model.

(3). **KNN Regression**

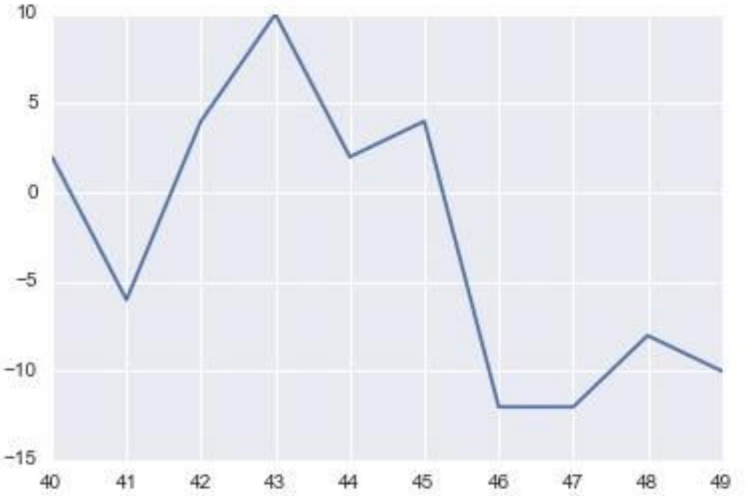
The ***k*-Nearest Neighbors algorithm** (or ***k*-NN** for short) is a non-parametric method used for regression.The input consists of the *k* closest training examples in the feature space. In *k-NN regression*, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbors.

Mean RMSE values for different k.



In order to select best k values, we observe the mean RMSE values for different values of K.

SCORE values for different K



We observe the Score values for different K values. We observe that the best score value is 10 for K values as 43. Since score values are our priority, so we check the mean Rmse value for the k =43.

MEAN RMSE	0.663
Max SCORE	10

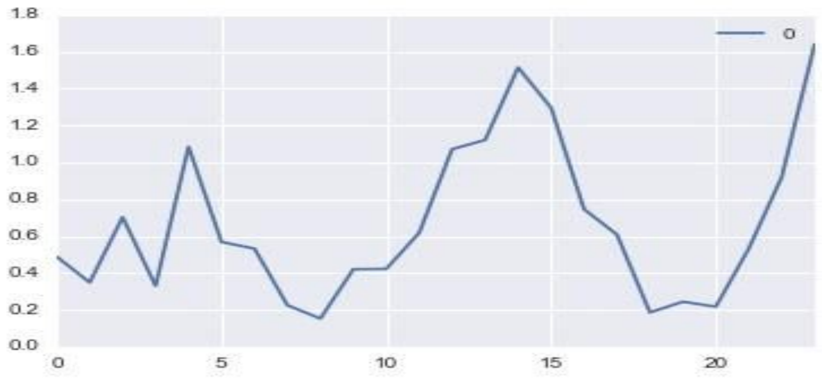
It can be observed that mean RMSE value for k=43 is 0.663 and score for K=43 is 10. So KNN with k=43 is one of the potential candidates as it has score value of 10 similar to Elastic Net but it has better RMSE value.

(4). **Kernel Ridge Regression**

Kernel ridge regression (KRR) combines Ridge Regression (linear least squares with l2-norm regularization) with the kernel trick. It thus learns a linear function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a nonlinear function in the original space.

The form of the model learned by Kernel Ridge is identical to support vector regression (SVR). However, different loss functions are used: KRR uses squared error loss while support vector regression uses ϵ-insensitive loss, both combined with l2 regularization. In contrast to SVR, fitting Kernel Ridge can be done in closed-form and is typically faster for medium-sized datasets. On the other hand, the learned model is non-sparse and thus slower than SVR, which learns a sparse model for $\epsilon > 0$, at prediction-time.

Mean RMSE value for different back tests



MEAN RMSE	0.667
SCORE	16

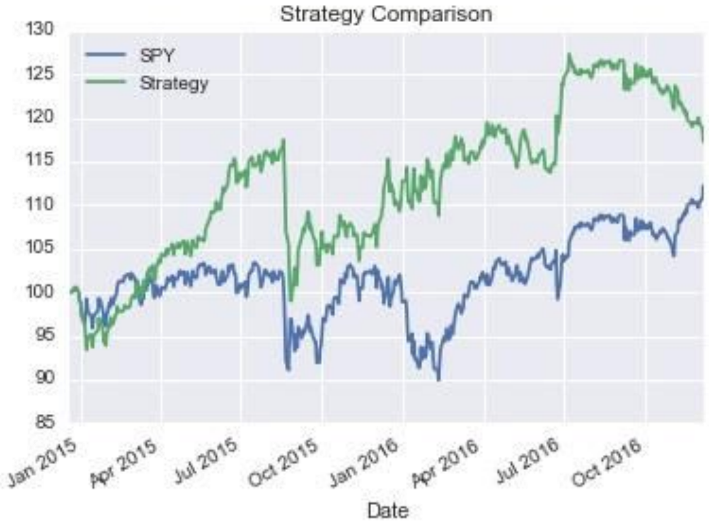
Since the Score value for Kernel Ridge regression model is maximum and it has reasonable values for Mean Rmse, we select the Kernel Ridge regression as our model to be used to simulate the strategy.

6. Final Model

Based on our analysis from back testing, we selected Kernel Ridge Regression model to implement our strategy. 500 days of trading data is used for strategy implementation. We will fit Kernel Ridge model using 63 days of data (prior to prediction date) and will be used to predict daily returns for next 5 days.

Using the predicted values of daily returns, we take our position at the start of the day. If the predicted value is positive, we take a long position and if the value is negative, we take a short position. We would compare our strategy of daily trading using and Market values

Kernel Ridge Regression:



Our model is doing reasonably well in comparison to the market. Different Strategies will yield different results. Using daily trade strategy, we actually do better than market.

For more complex strategies, which give variety of options like not trading (no trading on a given day) and doubling the trade bets (2 times the normal position) we use classification models. Model performance is based on testing data of last 98 days.

IV. Classification Modeling

1. Create Response Variable

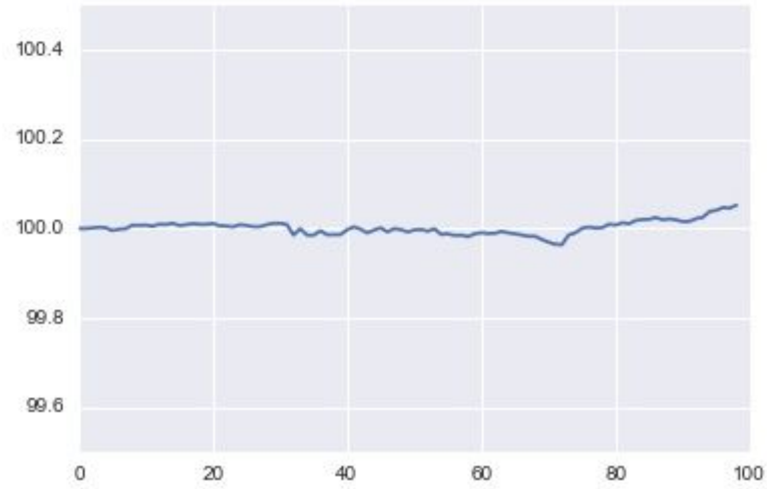
The response variable has 5 values: 2, 1, 0, -1, -2, which are generated from different levels daily return. For daily return less than -0.007, label it as -2; for daily return from -0.007 to -0.003, label it as -1; for daily return from -0.003 to 0.001, label it as 0; for daily return from 0.001 to 0.005, label it as 1; for daily return above 0.005, label it as 2. The response variables stands for the buy/sell signal in the stock market. 2 means strong buy signal ; 1 means general buy signal ; 0 means hold; -1 means general sell signal and -2 means strong sell signal.

2. Evaluation metrics

The system cannot be evaluated solely based on accuracy, since “It does not matter how often you win or lose, but how much you win when you win and how much you lose when you lose” – George Soros (paraphrased). Hence we will be evaluating results based on expectancy. Expectancy is conceptually, the amount you win on average per trade you take. Here it is measured in expected (average) % gain /loss per trade.Every system will be presented with:
(1). Accuracy
(2). Expectancy
(3). Total % made/lost (as modelled by account equity)
(4). Equity curve - The equity curve is a representation of how the account fluctuates as each trade is taken. X-axis means the number of trades, if the response variable equals to 0, there will be no trade. So the range of X-axis is 0 to 98. Y-axis stands for how much money we made. For example if the account starts at 100, and there is a winning trade that makes 2% , the account goes to 102. Then if the next trade makes 10 %, the account goes to 102 + 102*0.1 , which basically models the compounding effect of the trades.

3. Classification Models and Results

(0). **Index Returns**



Equity curve for Index
X axis represents number of trades, and Y axis equity value

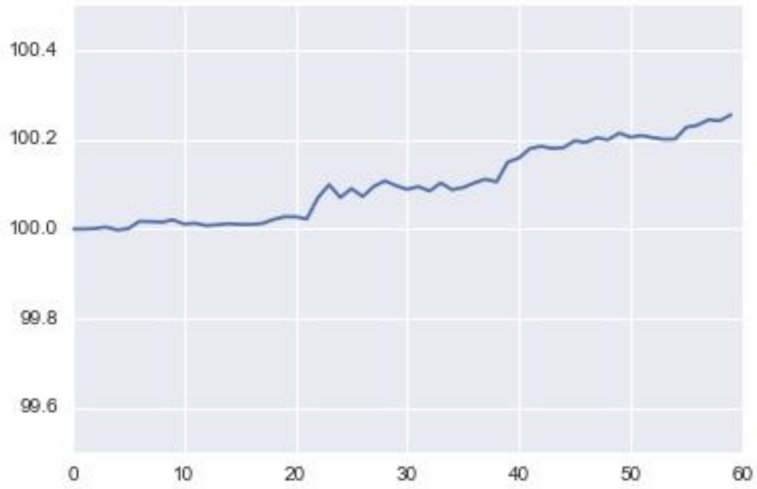
The above graph shows the returns the account will have made if we just traded the index every day during the 98 days test period with compounding effects. As is evident from above, we would have made less than 0.1 % gain.

Now below, we take the trades the models generated, and plot the corresponding equity curves for each model, which graphically shows how the account moved by taking trades suggested by the model.

(1). **Linear Discriminant Analysis**: LDA is based upon the idea of searching for a linear combination of variables that best separates multiple classes. The idea is to model the distribution of X in each of the classes separately, and then use Bayes theorem to obtain P(Y |X = x). Assuming in population π_k the probability density function of x is multivariate normal with mean μ_k and a constant covariance matrix Σ. The Linear Score Function is:

$$\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)+log(\pi_k)$$

where π_k is the prior probabilities of training data: $\pi_k = n_k/n$



Equity curve for Strategy 2 using LDA
X axis represents number of trades, and Y axis equity value

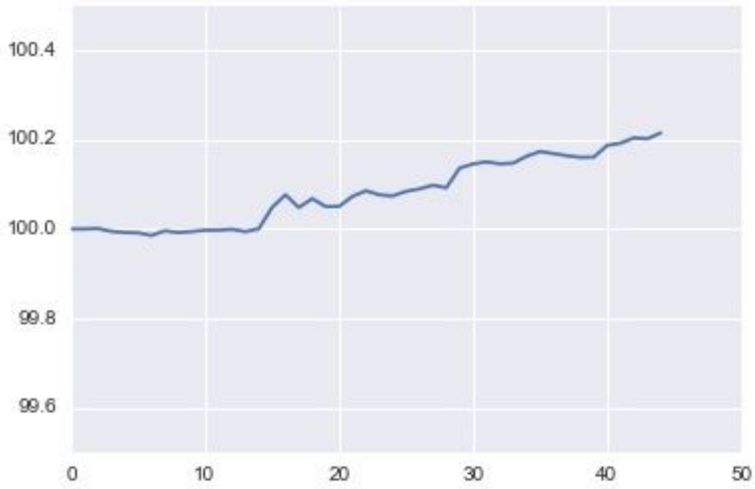
Accuracy	0.3160
Expectancy	0.0025
% made	0.2557

Strategy with LDA model as source for classification has done better than the market.As the return value is approximately 0.256% which is greater than the market return for this time frame.

(2). **Quadratic Discriminant Analysis**: is used for where the covariance matrices are different for x in different population π_k . The Quadratic Score Function is:

$$-\frac{1}{2}log\,|\Sigma_k|\; -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)+log(\pi_k)$$

For each record in the test dataset, find the K that will minimize the value of Quadratic Score Function. The K we found is the predicted value.



Equity curve for Strategy 2 using QDA
X axis represents number of trades, and Y axis equity value

Accuracy	0.3265
Expectancy	0.0029
% made	0.2152

Strategy with QDA model as source for classification has done better than the market.As the return value is approximately 0.215% which is greater than the market return for this time frame.

(3). **Support Vector Machine (SVM):** is a classification method using a separating hyperplane to separates the observations according to their labels of classes. The optimal separating hyperplane problem has less assumptions, which would results in more modeling misspecification. For non-separable case, we introduce a tuning parameter C >0 for 'cost' which accounts for errors, then the optimization problem is defined as:

$$minimize \, \tfrac{1}{2} \parallel \beta \parallel^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i(x^T\beta + \beta_0) \geq (1 - \xi_i),\ \xi_i \geq 0,\ i = 1, \ldots, n$ [2]
We use cross-validation to select C. Large C puts more focus on misclassification, and small C puts more focus on data which is away from the boundary. Using kernel trick in SVM will save a lot of computational time. Below are three common used kernel:

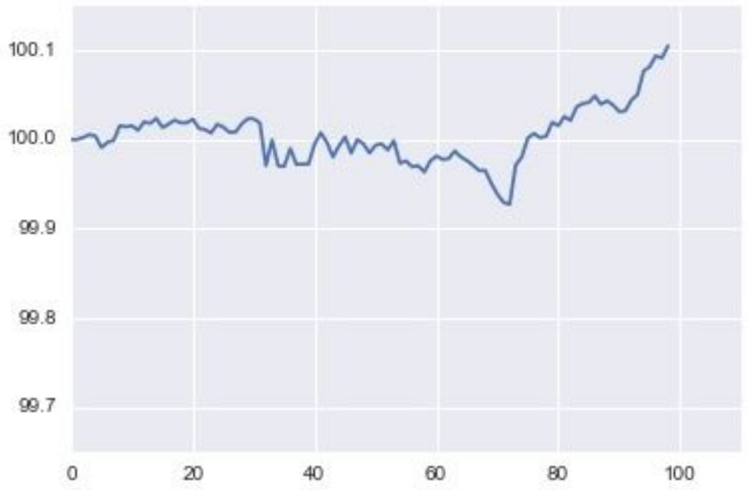
Radial Basis kernel: $K(x_1, x_2) = exp(- \parallel x_1 - x_2 \parallel^2 / c)$
dth degree Polynomial kernel: $K(x_1, x_2) = (1 + x_1 T x_2)^d$
Sigmoid kernel: $K(x_1, x_2) = tanh(\gamma x_1{}^T x_2 + r)$

Support Vector Classifier (kernel=Radial Basis kernel)

Accuracy	0.3571
Expectancy	0
% made	0

Expectancy and %made are both 0. This may not be a valid model for classification for this data.

Support Vector Classifier (kernel=poly, decision function = ovo)



Equity curve for Strategy 2 using SVM (Poly kernel)
X axis represents number of trades, and Y axis equity value

Accuracy	0.1837
Expectancy	0.0005
% made	0.1048

This model is doing better than the market as market returns are just under 0.1% but returns using this model is 0.1048%.

Support Vector Classifier (kernel=sigmoid, decision function = ovo)

Accuracy	0.3571
Expectancy	0
% made	0

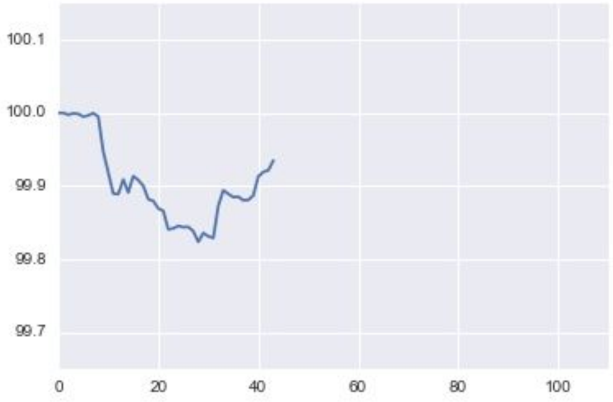
Expectancy and %made are both 0. This may not be a valid model for classification for this data.

(4). KNN (K-nearest neighbors): Given a positive integer K and a test observation x0, the KNN classifier first identifies the K points in the training data that are closest to x0, represented by N0. It then estimates the conditional probability for class j as the fraction of points in N0 whose response values equal j:

$$Pr(Y = j|X = x_0) = \frac{1}{K}\sum_{i\in N_0} I(y_i = j)$$

Finally, KNN applies Bayes rule and classifies the test observation x0 to the class with the largest probability. [1] We tuned K from 2 to 59, and choose K=25 for best accuracy.

Performance on Testing Data



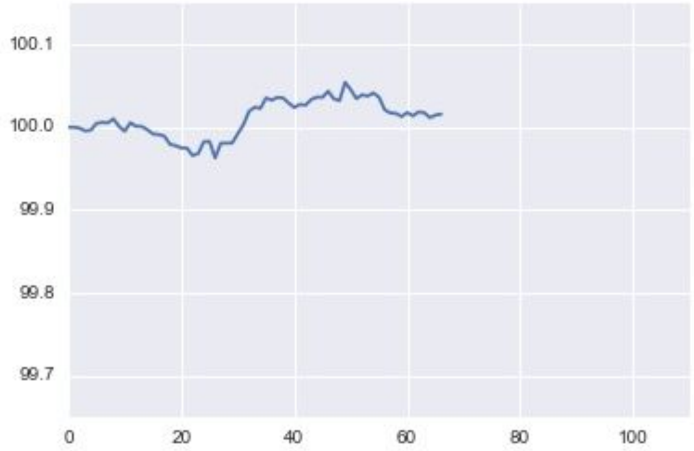
Equity curve for Strategy 2 using KNN (k=25)
X axis represents number of trades, and Y axis equity value

KNN with best k= 25 :

Accuracy	0.387
Expectancy	-0.00097
% made	-0.0651

Performance on the test data hasn't been good as we are losing money. As the percentage made is negative and so is the expectancy. KNN with k=25 may not be the ideal model for us.

(5). Random Forest: provides an improvement over bagged trees by way of a small tweak that decorrelates the trees[1]. There are three important tuning parameters in random forest:
NTREE : number of trees.
MTRY : number of variables considered at each split.
NODESIZE : terminal node size, same as nmin. [2]
We tuning max_features (Nodesize) from 2 to 8, n_estimators (Mtry) from 2 to 100, for finding the maximum expectancy. We choose max_features=6, n_estimators=2



Equity curve for Strategy 2 using Random Forest
X axis represents number of trades, and Y axis equity value

Accuracy	0.2143
Expectancy	0.0002
% made	0.0158

Though returns are positive but strategy is under performing with respect to market. So this again may not be the best model.

4. Final Model

	LDA	QDA	SVM radial	SVM poly	SVM sigmoid	KNN	Random Forest
Accuracy	0.3160	0.3265	0.3571	0.1837	0.3571	0.387	0.2143
Expectancy	0.0025	0.0029	0	0.0005	0	-0.00097	0.0002
% made	0.2557	0.2152	0	0.1048	0	-0.0651	0.0158

Since QDA gives largest expectancy and also good accuracy and percentage of made, we choose QDA model with data dimension reduced to 5 by PCA as the final model for classification.

V. Conclusions

Long term trading strategies require deep understanding of macroeconomic factors and company's strength and weakness.For short term investments, capturing market trends and predictions based on stock based indicators is more critical and that's where machine learning algorithms can contribute the most. Different Trading strategies aimed at beating the market index, require different machine learning algorithms to assist the investors in making decisions.

Trading Strategy 1: Daily trading - Use of Regression Models
For a strategy which involves daily trading and a fixed amount of investment each day, we used regression analysis to predict the daily returns and more importantly its tendency to gain or to lose value. For the regression analysis, we select the model using the technique called back testing in which we use historical data is used and selection is based on how particular model responds to the strategy implementation.We used models like Linear Regression, Elastic Net regression, KNN Regression and Kernel Ridge regression as perspective models and observed that Kernel Ridge regression performed the best. Using the Kernel Ridge regression model, we implemented our strategy for next 500 days in which training of model was done using 63-day window and fitted model was used for prediction of daily returns for next 5 days. Prediction values for next day daily returns were used to take position for next day. Our model beat the market by a good margin. It should be noted that this trading strategy required taking a position every single day and trading costs were not considered into the performance indicators.

Trading Strategy 2: Flexible Positions - Classification
1st trading strategy has some primitive characteristics as there are a lot of restrictions and is not flexible in terms to trade investments. In order to release some of the constraints, another strategy is formulated which allows holding the current position, doubling the position's bets etc. We use classification models to implement this strategy. Based on the daily returns, the trades are labelled as 2 ,1, 0 ,-1 ,-2. 2 means strong buy signal ; 1 means general buy signal ; 0 means hold; -1 means general sell signal and -2 means strong sell signal.Now the problem has become supervised classification. In order to select the model, we use last 98 days of data. Selection metrics try to infuse more practicality to the model and tries to emulate the real life trading. The metrics used are Accuracy of the classification, Expectancy and % made. Expectancy and % made reflect how implementation of this strategy using a particular algorithm would impact the equity of the investor.

To implement Trading Strategy 2, we used classification models like LDA, QDA, SVM with different kernel tricks, Random Forest with different parameters, and KNN classification model. Each one of them have been tested on the last 98 days of close market price data and corresponding indicators. Expectancy and % made metrics are our priority as they reflect how much money we are going to make if we use Trading Strategy 2 using the given classification algorithm.

While comparing the different models, we observed that QDA has performed the best on the testing data as it has the best expectancy value of 0.0029 in comparison other classification algorithms. And hence we choose QDA as our best model and we select to implement strategies similar to Trading Strategy 2 for our future investments.

We can compare performance of the strategy with respect to market, using the equity curves of the strategy and comparing it with market equity curve or we can compare their comparison on stock fluctuation curve depending on the business needs. In our case, equity curve makes more sense as given the option of holding back a deal, both graphs would differ as the compounding effect will have varied impact on different representation techniques . But in general, it can be observed that the models from the final models list above that made more than 0.1% essentially beat the index. These models are QDA, LDA, SVM with polynomial kernel.

Future Work outside the scope for course project: Future work which is outside the scope for this project can involve considering more diverse predictors such as sentiment predictors – using NLTK package, and scraping data from web and mining for phrases indicative of mood. Also we can consider some of the fundamental inputs which reflect the macroeconomic factors if available. For a fully fledged investment model, we can quantify risk involved in each trade made and corresponding charges associated with it. Portfolio management can also be one of the natural step ups for this project.

VI. References

[1] James, Gareth, et al. *An introduction to statistical learning*. Vol. 6. New York: springer, 2013.
[2] STAT 542 Course slides by Professor Zhu