CPE 695 Applied Machine Learning Final Project Stock Price Prediction

Ayushi Chaturvedi
Department of Electrical and
Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030
Email: achatur1@stevens.edu

Shreyansh Sharma
Department of Electrical and
Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030
Email: ssharm7@stevens.edu

Abrar Alam
Department of Electrical and
Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030
Email: aalam1@stevens.edu

***Abstract*—Predicting trends in stock market prices has been an area of interest for researchers for many years due to its complex and dynamic nature. Intrinsic volatility in stock market across the global market makes the task of prediction challenging. Forecasting and diffusion modeling, although effective can't be the panacea to the diverse range of problems encountered in prediction, short-term or otherwise. Market risk, strongly correlated with forecasting errors, needs to be minimized to ensure minimal risk in investment. The stock market is known to be a complex adaptive system that is difficult to predict due to the large number of factors that determine the day to day price changes. This is done in machine learning through regression which tries to determine the relationship between a dependent variable and one or more independent variables. Here, the independent variables are the features and the dependent variable for prediction is the price. Here there are three types of algorithms to compare and prove which ones works better for predicting the modal. The parameters used will be more in the manner of the late compost of the amount of the data processed by the algorithm to the efficiency, it has on predicting the price of the stock. This study aims to use linear and polynomial regression models to predict price changes and evaluate different models success by withholding data during training and evaluating the accuracy of these predictions using known data. Firstly, the paper talks more on linear Regression models since, its the first algorithm that has been used and then the paper will talk more about the Random Forest and finally more emphasis is put on Support vector regression or Neural network as they are used more these days.**

I. Introduction

This project concerns closing prices of stocks, therefore day trading was not modeled. The model for the stock market was only concerned with the closing price for stocks at the end of a business day, high-frequency trading is an area of research, but this study preferred a simplified model of the stock market. Here, the independent variables are the features and the dependent variable that predict is the price. It is apparent that the features are not truly independent, the volume and outstanding shares are not independent as well as the closing price and the return on investment not being independent. However, this is an assumption that has been made to simplify the model in order to use the chosen regression models. This study aims to use linear regression model, Random Forest, Support vector machines to predict price changes and evaluate different models success by with holding data during training and evaluating the accuracy of these predictions using known data. Hence, for the purpose of this project, variations in stock prices over a 3-month period can be predicted by using one or more organizations quarterly financial reports in the form of data sets which will be collected from the following repositories:

- https://www.quandl.com/
- https://www.quantopian.com/
- http://www.nasdaq.com/

II. Motivation

Stock market price prediction is a problem that has the potential to be worth billions of dollars and is actively researched by the largest financial corporations in the world. It is a significant problem because it has no clear solution, although attempts can be made at approximation using many different machine learning techniques. The project allows techniques for real-world machine learning applications including acquiring and analyzing a large data set and using a variety of techniques to train the program and predict potential outcomes.

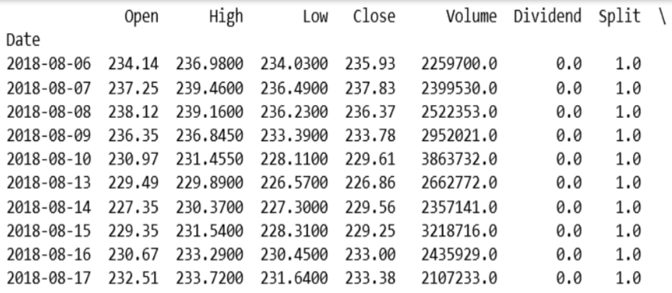
III. Date Set Representation

The data set which is collected from the Goldman Sachs (quandl) as a collection of comma separated values where each row consisted of a stock on a specific day along with data on the open stock , high and low of each stock price with close pointer and final volume. The Python scientific computing library numpy was used along with the data analysis library pandas in order to use dataset which is fetched from quandl site, further into pandas Data Frames in order that were indexed by date. The actual use of the training data set upto 75% and the remaining test data of 25% is used. The data frames are assigned corresponding to individual rows and the adjacent close column of the stock DataFrame is converted to a list. The Date can be represented in terms of the indexed rows and zero refers to the initial Start date of the test until the current date. The lists are converted to numpy arrays and those are finally converted to the single dimension vectors. After slicing the data, reshaping is done. For example, some libraries, such as scikit- learn , may require one dimensional array with one column and outcomes for each column. It is important to know how to reshape NumPy arrays so that data meets the expectation of specific python library. The data set predicted by the linear regression has been shown in the running code, after that the efficiency of the algorithm works with the learning rate. After running the code for several times the efficiency of the algorithm will increase and it increases up to approximately 56%. depending the type of data it fetches at that moment.

IV. Machine Learning Algorithms Implemented

A. Linear Regression

Linear regression is a method used to model a relationship between a dependent variable (y), and an independent variable (x). With simple linear regression, there will only be one independent variable x. There can be many independent variables which would fall under the category of multiple linear regression. In this circumstance, only independent variable is the date. The date will be represented by an integer starting at 1 for the first date going up to the length of the vector of dates which can vary depending on the time series data. The dependent variable, of course, will be the price of a stock. Here for the purpose of project, the data of the Goldman Sachs stock dataset from Quandl for the period of approximately 2 months is taken and the prices are closed stock. Here the use of the dataframes with converted adjacent close columns of the stock data frames into a list are shown.



```
              Open      High       Low   Close     Volume  Dividend  Split  \
Date
2018-08-06  234.14  236.9800  234.0300  235.93  2259700.0       0.0    1.0
2018-08-07  237.25  239.4600  236.4900  237.83  2399530.0       0.0    1.0
2018-08-08  238.12  239.1600  236.2300  236.37  2522353.0       0.0    1.0
2018-08-09  236.35  236.8450  233.3900  233.78  2952021.0       0.0    1.0
2018-08-10  230.97  231.4550  228.1100  229.61  3863732.0       0.0    1.0
2018-08-13  229.49  229.8900  226.5700  226.86  2662772.0       0.0    1.0
2018-08-14  227.35  230.3700  227.3000  229.56  2357141.0       0.0    1.0
2018-08-15  229.35  231.5400  228.3100  229.25  3218716.0       0.0    1.0
2018-08-16  230.67  233.2900  230.4500  233.00  2435929.0       0.0    1.0
2018-08-17  232.51  233.7200  231.6400  233.38  2107233.0       0.0    1.0
```

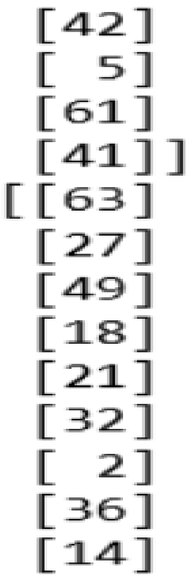
Fig. 1. Portion of the dataset imported from Quandl

These are converted to the numpy arrays and which further converts to 1D vectors, using the length and the reshaping the data in order to select the stock price in an order learned by the algorithm.

B. Random Forest

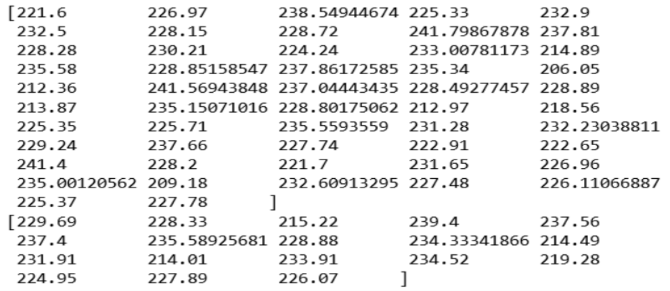
Random forest algorithm is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. The basic steps of the algorithm works this way:

1) Pick N random records from the dataset.
2) Build a decision tree based on these N records.
3) Choose the number of trees you want in your algorithm and repeat steps 1 and 2..



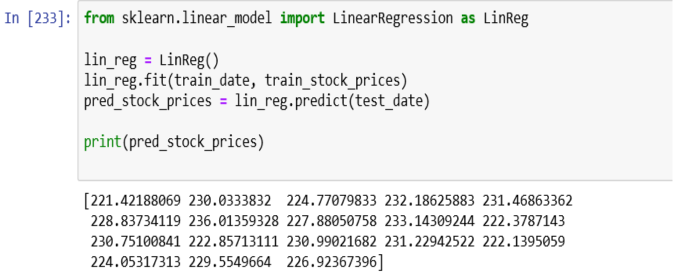
```
  [42]
  [ 5]
  [61]
  [41]]
 [[63]
  [27]
  [49]
  [18]
  [21]
  [32]
  [ 2]
  [36]
  [14]
```

Fig. 2. Indexed training and testing date converted into a list of numpy array



```
[221.6        226.97       238.54944674 225.33       232.9
 232.5        228.15       228.72       241.79867878 237.81
 228.28       230.21       224.24       233.00781173 214.89
 235.58       228.85158547 237.86172585 235.34       206.05
 212.36       241.56943848 237.04443435 228.49277457 228.89
 213.87       235.15071016 228.80175062 212.97       218.56
 225.35       225.71       235.5593559  231.28       232.23038811
 229.24       237.66       227.74       222.91       222.65
 241.4        228.2        221.7        231.65       226.96
 235.00120562 209.18       232.60913295 227.48       226.11066887
 225.37       227.78      ]
[229.69       228.33       215.22       239.4        237.56
 237.4        235.58925681 228.88       234.33341866 214.49
 231.91       214.01       233.91       234.52       219.28
 224.95       227.89       226.07      ]
```

Fig. 3. Training and testing stock prices converted into numpy arrays



```
In [233]: from sklearn.linear_model import LinearRegression as LinReg

          lin_reg = LinReg()
          lin_reg.fit(train_date, train_stock_prices)
          pred_stock_prices = lin_reg.predict(test_date)

          print(pred_stock_prices)

          [221.42188069 230.0333832  224.77079833 232.18625883 231.46863362
           228.83734119 236.01359328 227.88050758 233.14309244 222.3787143
           230.75100841 222.85713111 230.99021682 231.22942522 222.1395059
           224.05317313 229.5549664  226.92367396]
```

Fig. 4. Predicted stock prices

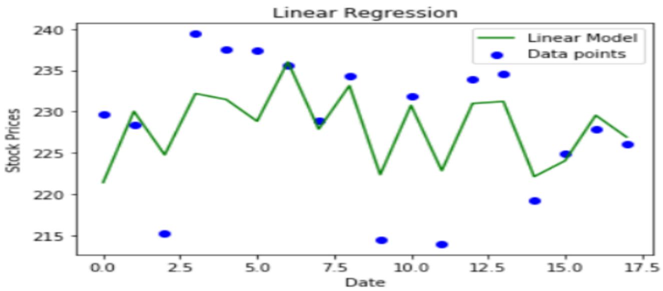


Fig. 5. Linear Regression Plot

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote. There are many advantages to use random forest algorithm:

- The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of ”the crowd”; therefore the overall biasedness of the algorithm is reduced.
- This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.

There are disadvantages also with this algorithm:

- A major disadvantage of random forests lies in their complexity. They required much more computational resources, owing to the large number of decision trees joined together.
- Due to their complexity, they require much more time to train than other comparable algorithms.

C. Support Vector Regression

The regression problem is a generalization of the classification problem, in which the model returns a continuous-valued output, as opposed to an output from a finite set. In other words, a regression model estimates a continuous-valued multivariate function. The optimization problem entails finding the maximum margin separating the hyperplane, while correctly classifying as many training points as possible. SVMs represent this optimal hyperplane with support vectors. The sparse solution and good generalization of the SVM lend themselves to adaptation to regression problems. SVM generalization to SVR is accomplished by introducing an -insensitive region around the function, called the -tube. This tube reformulates the optimization problem to find the tube that best approximates the continuous-valued function, while balancing model complexity and prediction error. More specifically, SVR is formulated as an optimization problem by first defining a convex -insensitive loss function to be minimized and finding the flattest tube that contains most of the training instances. Hence, a multiobjective function is constructed from the loss function and the geometrical properties of the tube. Then, the convex optimization, which has a unique solution, is solved, using appropriate numerical optimization algorithms. The hyperplane is represented in terms of support vectors, which are training samples that lie outside the boundary of the tube. As in SVM, the support vectors in SVR are the most influential instances that affect the shape of the tube, and the training and test data are assumed to be independent and identically distributed (iid), drawn from the same fixed but unknown probability distribution function in a supervised-learning context. SVR adopts an -insensitive loss function, penalizing predictions that are farther than from the desired output. The value of determines the width of the tube; a smaller value indicates a lower tolerance for error and also affects the number of support vectors and, consequently, the solution sparsity. The smaller the value of , the more points that lie outside the tube and hence the greater the number of support vectors. SVR is a useful and flexible technique, helping the user to deal with the limitations pertaining to distributional properties of underlying variables, geometry of the data and the common problem of model overfitting. The choice of kernel function is critical for SVR modeling.

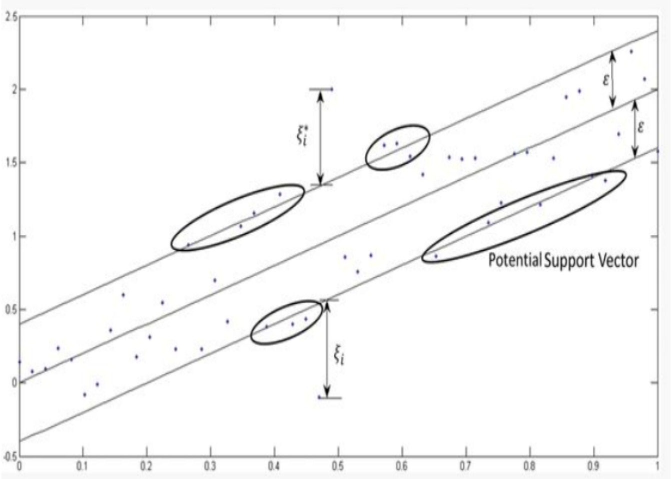
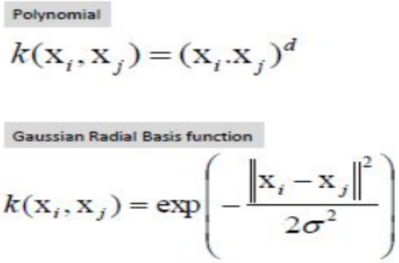


Fig. 6. 1D Linear SVR



Polynomial

$$k(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}_i . \mathrm{x}_j)^d$$

Gaussian Radial Basis function

$$k(\mathrm{x}_i, \mathrm{x}_j) = \exp\left(-\frac{\left\| \mathrm{x}_i - \mathrm{x}_j \right\|^2}{2\sigma^2} \right)$$

Fig. 7. Mathematical Expression for SVR

V. Result

The mid semester report provides us with the knowledge of all the three algorithms which are being used here. The first algorithm talks about the linear regression and showed us how the training data is being used in order to find maximum accuracy for the result. The potential graph shows this with an accuracy of 55.6% which is fairly good considering the amount of the scattered data. In the final report, The Comparison will be more specific as, the comparison will be done with three algorithms and the final result will be generated.

VI. References

References

[1] P.K. Sahoo and K. Charlapally *Stock Price Prediction Using Regression Analysis*, Volume.6, Issue.3, pp.1655-1659. International Journal of Scientific Engineering Research, March 2015.

[2] S. Madge and S. Bhatt *Predicting Stock Price Direction Using Support Vector Machines*. Independent Work Report, Spring 2015.

[3] L. Nunno *Stock Price Prediction Using Linear and Polynomial Regression Models*

[4] Bean Coder *Predicting Google's Stock Price Using Linear Regression* Bean Coder, 2018. http://beancoder.com/linear-regression-stock-prediction/

[5] F. Gharehchopogh, T. Bonab and S. Khaze *A LINEAR REGRESSION APPROACH TO PREDICTION OF STOCK MARKET TRADING VOLUME: A STUDY*, Vol.4, No.3 International Journal of Managing Value and Supply Chains (IJMVSC), September 2013.

[6] M. Awad and R. Khanna *Support Vector Regression*, pp. 67-80 Efficient Learning Machines, 2015.

[7] Bayowa *Simple Linear Programming vs Support Vector Regression* Advantages of Support Vector Regression over Simple Linear Regression, 2017. https://rpubs.com/linkonabe/SLSvsSVR.