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 that we would like to predict is the price. Here there are three types of algorithms to compare and the paper 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 machines or Neural network as they are used more these days.**

I. Introduction

This project concerns closing prices of stocks, therefore day trading is 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 prefers a simplified model of the stock market. Here, the independent variables are the features and the dependent variable is that which predicts the prices. 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 regression to predict price changes and evaluate different models success by withholding 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 6-month period can be predicted by using one or more organizations quarterly financial reports in the form of datasets 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. Related Work

Prediction of stock market returns is an important issue in finance. Artificial neural networks have been in stock market prediction during the last decade. Studies were performed for the prediction of stock market values. Nowadays, artificial neural network (ANN) have been popularly applied to finance problems such as stock price index prediction, bankruptcy prediction and corporate bond classification. ANN was used in the stock market prediction during the last decade. One of the first projects was by Kimoto and friends who had used ANN predicted buying and selling signals with an overall prediction rate of 63Md. Rafiul Hassan and Baikunthu Nath used Hidden Markov models(HMM) approach to forecasting stock price for interrelated markets. HMM was used for pattern recognition and classification problems because of its proven suitability for modeling dynamic systems. The advantage of the HMM was strong statistical foundation. Its able to handle the new data robustly and computationally efficient to develop and evaluate similar patterns. Further, the development of the of hybrid system using AI paradigms with HMM improve the accuracy and efficiency of forecast the stock market. Fazel Zarandi M.H, Razaee B, TurksenI.B and Neshat E used a type-2 fuzzy rule based expert system is developed for stock price analysis. The purposed type-2 fuzzy model applies the technical and fundamental indexes as the input variables. The model used for the stock price prediction of an automotive manufacturer in Asia. The output membership values were projected onto the input spaces to generate the next membership values of input variables and tuned by genetic algorithm. The type-1 method was used for interference and to increasing the robustness of the system. This method was used to robustness, flexibility and error minimization. It is used to forecast more profitable trading in the stock markets.

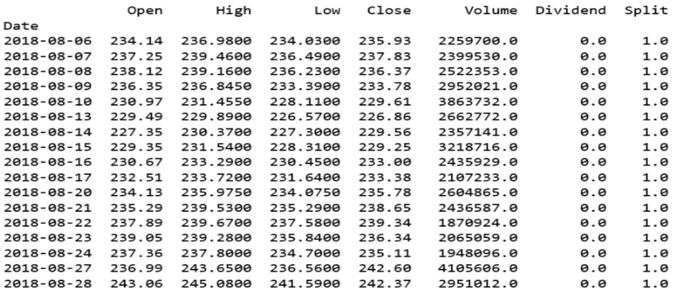
IV. Solutions

A. Dataset Description

The data set 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 convert these CSV files into pandas Data Frames in order that were indexed by date. The actual use of the training data set up to 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 data frame 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, we have to reshape it. 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 algorithms will increase and it will show a clear idea that the values will always create the state that is required best to run the algorithms and increasing the value of the C will delay the result and the required output will also change valuation of the learning algorithm. The support vector regression with radial basis function will provide the really good accuracy since it forms the bell shaped curve. The gamma parameter sets the width of the bell shaped curve. The larger the value of gamma the narrower will be the bell. Small values of gamma yield the wide bells. We have also used the random foresting method to find the stock prices as decision trees can be used for various learning applications. Random forest overcomes the problem by training multiple decision trees on different subspaces of the features space at the cost of the slightly increased bias.

B. 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 4 months is taken and the prices are of closed stock. Here the use of the data frames with converted adjacent close columns of the stock data frames into a list are shown. 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. The linear regression model is fit to training data and made prediction using testing set. Then accuracy of prediction was checked using the testing data. The values of split training and data set initially were started with 20% for test size and 80% for train size. However, the achieved accuracy was below 20% and sometimes even below 0%, i.e, negative accuracy was achieved. In order to optimize the performance of the linear regression model several possible combinations of test and train sizes were used until the achieved accuracy was more than 70%.Hence, the train size was chosen to be 75% and test size was chosen to be 25%.



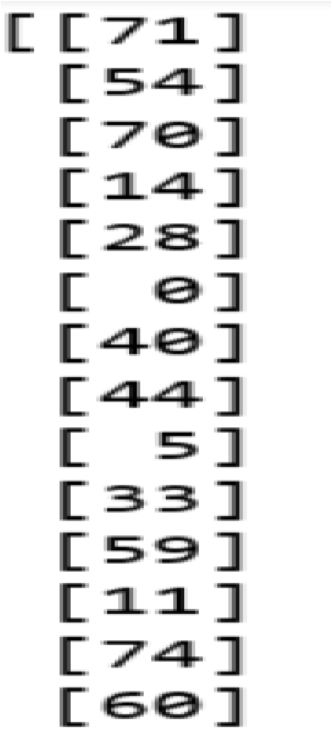
```
              Open      High       Low   Close     Volume  Dividend  Split
Date
2018-08-06  234.14  236.9800  234.0300  235.93  2259700.0       0.0    1.0
2018-08-07  237.25  239.4600  236.4900  237.83  2399530.0       0.0    1.0
2018-08-08  238.12  239.1600  236.2300  236.37  2522353.0       0.0    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
2018-08-20  234.13  235.9750  234.0750  235.78  2604865.0       0.0    1.0
2018-08-21  235.29  239.5300  235.2900  238.65  2436587.0       0.0    1.0
2018-08-22  237.89  239.6700  237.5800  239.34  1870924.0       0.0    1.0
2018-08-23  239.05  239.2800  235.8400  236.34  2065059.0       0.0    1.0
2018-08-24  237.36  237.8000  234.7000  235.11  1948096.0       0.0    1.0
2018-08-27  236.99  243.6500  236.5600  242.60  4105606.0       0.0    1.0
2018-08-28  243.06  245.0800  241.5900  242.37  2951012.0       0.0    1.0
```

Fig. 1. Portion of the dataset imported from Quandl

C. Random Forest Regressor

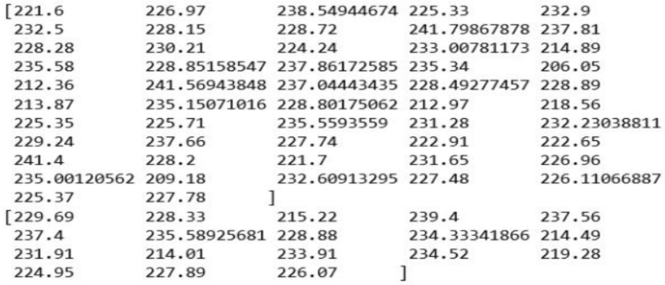
Random forest algorithm is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest it 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..



```
[[71]
 [54]
 [70]
 [14]
 [28]
 [ 0]
 [40]
 [44]
 [ 5]
 [33]
 [59]
 [11]
 [74]
 [60]]
```

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

4) 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.

In the case of Random forest regressor, several values of n_estimators ranging from 500 to 1000 were used. For each value of n_estimators the random forest regressor algorithm was using scikit-learn library. It was observed that, the lower the value for n_estimators, the less accurate the results were found. Consequently, when n_estimators is equal to 1000, the accuracy of regression algorithm was more than 95% which was the highest achieved accuracy among all the other algorithms

D. 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. SVRs represent this optimal hyper plane with support vectors. The sparse solution and good generalization of the SVR lend themselves to adaptation to regression problems. SVR 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 multi-objective 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 SVR, 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 is 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

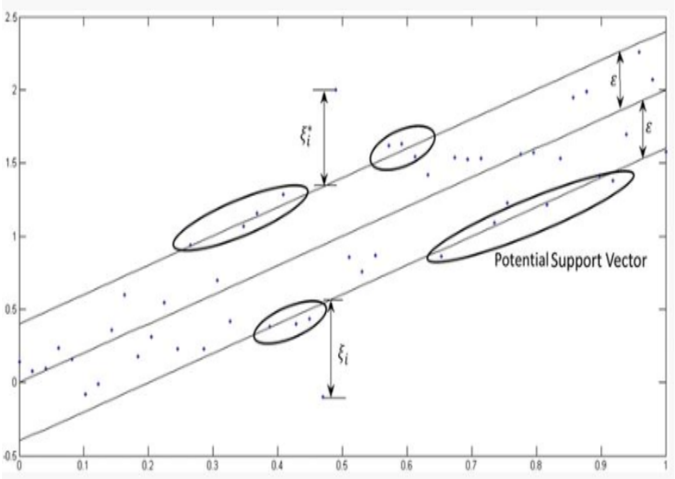


Fig. 4. 1-D Linear SVR

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. For Linear Support Vector Regression, the value of count was increased from 500 to 1000 expecting for better accuracy and it was found that accuracy was not more than 68.8% .It changes the number of counts never gave an output that matched with the expected accuracy.

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. 5. Mathematical Expression for Support Vector Regression

For Polynomial Support Vector Regression, changing the number of counts exponentially increased the processing time of the algorithm. It provided better results compared to less number of counts that were processed earlier. For the final results C =1000 and degree = 2 were set.

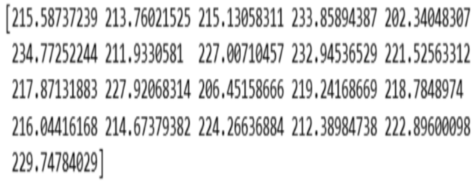
For Radial basis function (RBF) Support Vector Regression, this is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. For approximations the support vector machines and other models employing the kernel trick doesn't scale well to large numbers of features in the input space, several approximations to the RBF kernel (and similar kernels) have been introduced. Typically, these take the form of a function z that maps a single vector to a vector of higher dimensionality, approximating the kernel .

$$\langle z(\mathbf{x}), z(\mathbf{x}') \rangle \approx \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle = K(\mathbf{x}, \mathbf{x}')$$

Fig. 6. Mathematical Expression for RBF Support Vector Regression

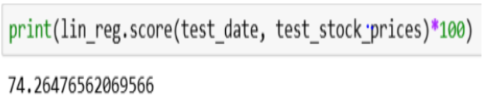
V. Results

The following figures below represent the outputs and the associated statistical results for the Linear Regression model: The advantages of Linear Regression are as follows:



```
[215.58737239 213.76021525 215.13058311 233.85894387 202.34048307
 234.77252244 211.9330581  227.00710457 232.94536529 221.52563312
 217.87131883 227.92068314 206.45158666 219.24168669 218.7848974
 216.04416168 214.67379382 224.26636884 212.38984738 222.89600098
 229.74784029]
```

Fig. 7. Predicted Stock Prices using Linear Regression



```
print(lin_reg.score(test_date, test_stock_prices)*100)

74.26476562069566
```

Fig. 8. Accuracy of the predicted stock prices using Linear Regression

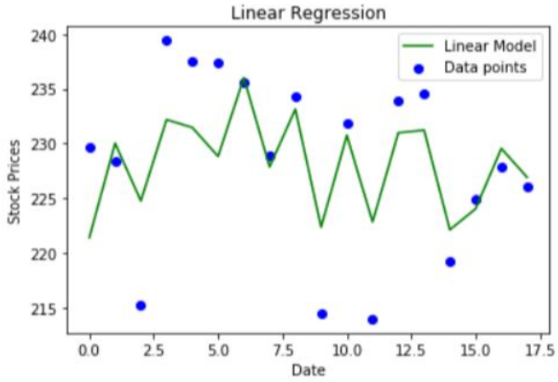


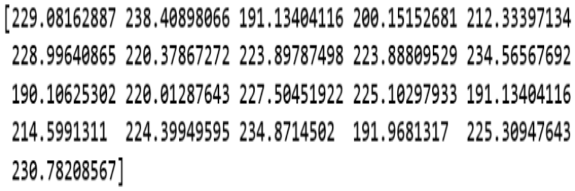
Fig. 9. Plot of predicted stock prices using Linear Regression

1) Space complexity is very low it just needs to save the weights at the end of training. hence it's a high latency algorithm.
2) It is very simple to understand.
3) Good interpretability.
4) Feature importance is generated at the time model building. With the help of hyper-parameter lambda,it can handle features selection hence we can achieve dimensionality reduction.

Similarly, the disadvantages are as follows:

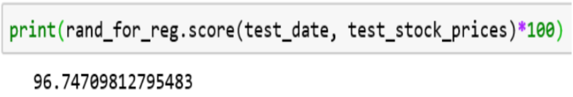
1) The algorithm assumes data is normally distributed in real they are not.
2) Before building model multi-collinearity should be avoided.
3) Prone to outliers.

The results for the Random Forest Regressor model are displayed below: There are many advantages to use random forest algorithm:



```
[229.08162887 238.40898066 191.13404116 200.15152681 212.33397134
 228.99640865 220.37867272 223.89787498 223.88809529 234.56567692
 190.10625302 220.01287643 227.50451922 225.10297933 191.13404116
 214.5991311  224.39949595 234.8714502  191.9681317  225.30947643
 230.78208567]
```

Fig. 10. Predicted stock prices using Random Forest Regressor



```
print(rand_for_reg.score(test_date, test_stock_prices)*100)

  96.74709812795483
```

Fig. 11. Accuracy of predicted stock prices using Random Forest Regressor

1) 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.
2) 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:

1) 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.
2) Due to their complexity, they require much more time to train than other comparable algorithms.

Linear SVR: The following are the advantages of Support Vector Regression:

1) It works really well with clear margin of separation.
2) It is effective in high dimensional spaces.
3) It is effective in cases where number of dimensions is greater than the number of samples.
4) It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

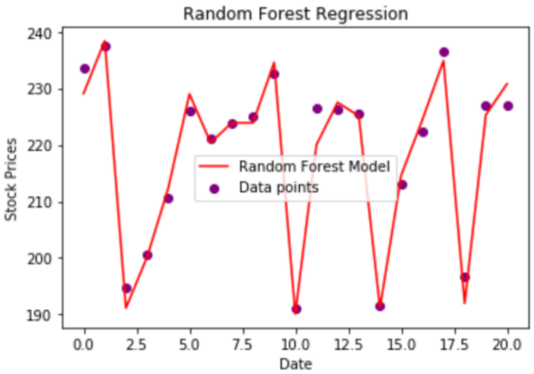
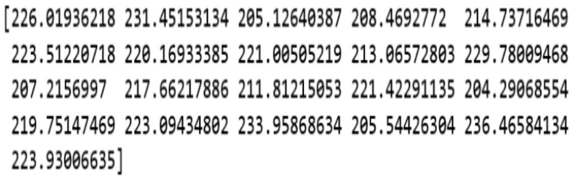
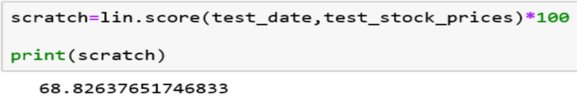


Fig. 12. Plot of predicted stock prices using Random Forest Regressor



```
[226.01936218 231.45153134 205.12640387 208.4692772  214.73716469
 223.51220718 220.16933385 221.00505219 213.06572803 229.78009468
 207.2156997  217.66217886 211.81215053 221.42291135 204.29068554
 219.75147469 223.09434802 233.95868634 205.54426304 236.46584134
 223.93006635]
```

Fig. 13. Predicted stock prices using Linear SVR



```
scratch=lin.score(test_date,test_stock_prices)*100
print(scratch)
    68.82637651746833
```

Fig. 14. Accuracy of predicted stock priceS using Linear SVR

The disadvantages of Support Vector Machines are:

1) It does not perform well, when we have large data set because the required training time is higher.
2) It also does not perform very well, when the data set has more noise i.e. target classes are overlapping.
3) SVM does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library

VI. Future Research Direction

The future works talks about working on random forest such that, its efficiency increases up to 98%. Further, the paper will focus more on using neural network approach to build a constant network so as to take care of more complex stock price predictions. The changes and the approach followed for this paper have taken its own process of time and this makes it more productive. In the near future, the concept of neural network will be considered more, in hope that it creates a significant results with more advance and faster algorithm.

VII. Conclusion and Comparisons

The final report provides us with the knowledge of all the three algorithms which are used here. The first algorithm talks about the linear regression and showed us how the training data

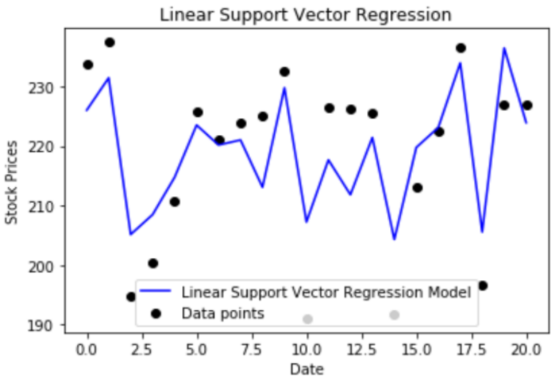
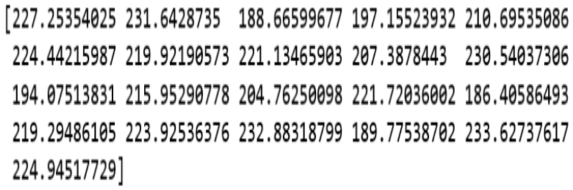
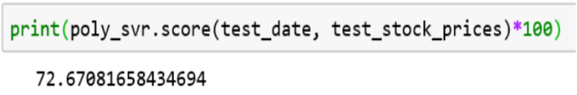


Fig. 15. Plot of predicted stock prices using Linear SVR



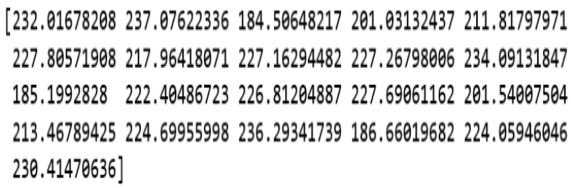
```
[227.25354025 231.6428735  188.66599677 197.15523932 210.69535086
 224.44215987 219.92190573 221.13465903 207.3878443  230.54037306
 194.07513831 215.95290778 204.76250098 221.72036002 186.40586493
 219.29486105 223.92536376 232.88318799 189.77538702 233.62737617
 224.94517729]
```

Fig. 16. Predicted stock prices using Polynomial SVR



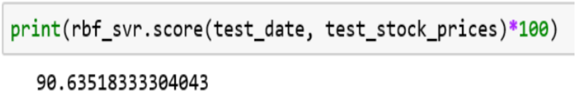
```
print(poly_svr.score(test_date, test_stock_prices)*100)

    72.67081658434694
```

Fig. 17. Accuracy of predicted stock priceS using Polynomial SVR



```
[232.01678208 237.07622336 184.50648217 201.03132437 211.81797971
 227.80571908 217.96418071 227.16294482 227.26798006 234.09131847
 185.1992828  222.40486723 226.81204887 227.69061162 201.54007504
 213.46789425 224.69955998 236.29341739 186.66019682 224.05946046
 230.41470636]
```

Fig. 19. Predicted stock prices using RBF SVR



```
print(rbf_svr.score(test_date, test_stock_prices)*100)

    90.63518333304043
```

Fig. 20. Accuracy of predicted stock prices using RBF SVR

is to be used in order to find maximum accuracy for the result. The potential graph shows this with accuracy of 74.5% which is fairly good considering the amount of the scattered data. The value of the training data provides us with the changes in the learning rate that affect the results.The Support Vector Regression method has three parts namely Polynomial, Radial Basis Function, Linear. The Linear Support Vector Regression provides an accuracy 68.8% while Support Vector Regression Polynomial with accuracy of 72.6% and finally Support Vector Machines Radial Basis Function provides the accuracy of 90.6%. The time taken by Poly SVR was most due to C= 1000, degree = 2 for the reason that the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models. Intuitively, the polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these. In the context of regression analysis, such combinations are known as interaction features. The (implicit) feature space of a polynomial kernel is equivalent to that of polynomial regression, but without the combinatorial blow up in the number of parameters to be learned. When the input features are binary-valued (booleans), then the features correspond to logical conjunctions of input features.Hence, it took much more time to show the results than any other algorithm. On the other hand the Support Vector Regression-Radial Basis Function could work only with the first point connection with the Training data and Testing data and hence, even after changing variables of Gamma and C it provides the results about 90%. Here, if it is required to maintain a really good result, it is better to keep the results for Gamma = 0.1. The final algorithm that requires most attention in our paper is Random Forest, Since the approach of this algorithm makes it more efficient and keeps the rate of algorithm in a certain minimum time frame. The efficiency for random foresting is 96.7% which is due to decision tree approach in the algorithm.

VIII. Contribution

1) Ayushi Chaturvedi: Coded the Linear Regression model using Scikit-learn and prepared presentation slides
2) Abrar Alam: Primary research, prohect proposal, data pre-processing and coded the Random Forest Model using Scikit-learn, conversion of mid-stage, as well final reports to LaTex using Overleaf
3) Shreyansh Sharma: Coded the SVR models using the Scikit-learn and mid-stage, as well as final reports

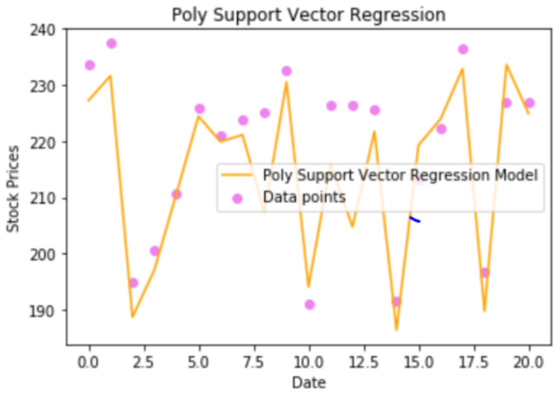


Fig. 18. Plot of predicted stock prices using Polynomial SVR

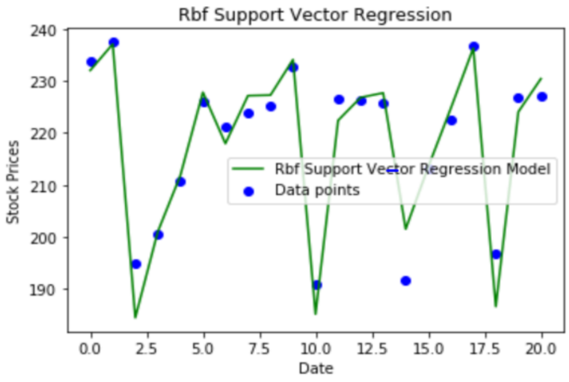


Fig. 21. Plot of predicted stock prices using RBF SVR

IX. 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.