Stock Closing price prediction

Name:	**Kunal Shaw**
Registration No./Roll No.:	19353
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	Feb 2, 2022
Date of Submission:	April 24, 2022

1 Introduction

Prediction of Closing Price of stocks is of great interest for day traders, swing traders and active investors around the globe. Correctly predicting closing price of stock can make good sum of wealth from the market. For the same here I am using data of two years of 88 companies of different sectors to predict the closing price of stock any particular day. Data consist of four features: "Open": Opening price of the Stock "High": Highest price reached by the Stock in a day "Close": Price of Stock when Stock Exchange got closed. "Volume": Number of Shares Traded in that particular day. All the features are numerical. No categorical feature.

2 Methods

From our data , we have done feature selection to find the best features for model training among [‘Open’ , ‘High’ , ’Low’ , ‘Volume’] to predict [‘Close’] price of the stock. In feature selection, we have calculate Mutual Information regression of all the feature. From the above information we can observe that the [‘Open’ , ‘High’ , ’Low’] are relevant feature for model training. Forward feature selection is also used as a feature selection method. This is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model. Using this method we observe that all four features should be used to train the model.

As we have highly correlated features and to avoid over fitting of our data we have taken all the features of our data i.e [’Open’, ’High’, ’Low’, ’Volume’].

Model Training

Now, data is ready to be trained into various regression model. At first we have split the data into 4 parts X_train, X_test, y_test, y_train. The data is divided into train:test = 80:20 ratio. The size of training data is 78185 and size of test data is 19547. We are using 5 different regression model to train the model and then compare their results over test data to select the best model.

1. Linear Regression

Table 1: Mutual information of all the features

Feature 0=[‘Open’]	4.102847
Feature 1=[‘High’]	4.473270
Feature 2=[‘Low’]	4.443530
Feature 3=[‘Volume’]	0.396575

2. SVM Regression
3. Decision Tree Regression
4. Random Forest Regression
5. KNN Regression

Hyper Parameter Tuning

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters. However, there is another kind of parameters, known as Hyper parameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn. For hyper parameter tuning we have used GridSearchCV which is library function of sklearn's model selection package. This also help us to set the number of cross-validation . Let discuss the methodology of every regression algorithm to get the best model

Linear Regression

Linear Regression fits a linear model with coefficients w = (w1, . . . , wp) to minimize the residual sum of squares between the observed targets in the data-set, and the targets predicted by the linear approximation. Here we have used Ordinary least squares Linear Regression. In this algorithm we have done 10 fold cross- validation with scoring parameter as negative mean square error . This algorithm don't have any hidden parameter to tune , so we have applied GridSearchCV to train the model.

SVM Regression

The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points. As this algorithm is computationally very expensive so we have manually selected values of the parameter. C=0.8, epsilon=0.2, kernel='poly'. With this parameter we have trained our model

Decision Tree Regression

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values. During Hyper Parameter tuning , parameter have been tuned with many values. The best model parameters are Criterion: friedman_mse , max_depth: 20, max_feature: auto, max_leaf_nodes: None, min_samples_leaf: 3, spliter: best 5 fold cross- validation with scoring criteria of mean squared error is used to train the model.

Random Forest Regression

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. During hyperparameter tunning the best parameter selected are bootstrap: True, max_depth: 20, max_features: 'auto', n_estimators: 13. The value of n_estimator is an important parameter, which the number of decision tree created in random forest classifier. In our model , 13 decision trees gives best model. 3 fold cross- validation(as this algorithm is computationally expensive with higher cv values) with scoring criteria of mean squared error is used to train the model

KNN Regression

Here, regression is based on k-nearest neighbors. The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set. Using GridSearchCV for hyper parameter tuning the values of parameter best for the algorithm are algorithm: 'brute', n_neighbors: 2, Here the most important parameter is the number of neighbours , 2 is best for our model. 3 fold cross- validation(as this algorithm is computationally expensive with higher cv values) with scoring criteria of mean squared error is used to train the model.

Github Link

Explain the proposed methods in detail and write different methods that you have explored for experimental analysis in brief and with proper references [?]. However, no need to write the existing method (e.g., SVM, k-means etc.) in detail.

Discuss the idea of parameter tuning with proper references [?, ?]. You may also refer about other experimental settings here e.g., the tools[1] used to implement the classifiers [?, ?].

3 Evaluation Criteria

The data is trained with all the algorithm and hence models are generated. The performance of the model is evaluated based on the predicted value of close price of stock with the actual close price of the test data. From all the scoring criteria like Mean Absolute Error, Max Error, RMSE, Median Absolute Error, MAPE , we have taken Mean Square Error(MSE) to comapre various model performance

$$MSE = (\tfrac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2$$

n : number of datapoints.
yi : True value of Closing Price.
xi : Predicted Value of Closing Price.

4 Analysis of Results

The performance of all the models is evaluated and mentioned in the table. As we can observe that linear regression has highest MAE value over Random Forest , KNN, SVR or Decision Regressor. As the features are highly correlated with each other therefore 1

Table 2: The performance of all the models is evaluated and mentioned in the table

Model	MAE	RMSE	MAPE	Scoring=MAE
Random Forest Regression	4.3687	72.938	0.00195	0.99997835
KNN Rrgressor	46.4371	146.849	0.5338	0.99995644
Linear Regressor	6.344	72.589	0.003701	0.99998927
Support Vector Regressor	2493.9149	22375.73201	15500244462853	0.01231
Decision Tree Regression	9.8195	114.299	155134995093.683	0.99997358

5 Discussions and Conclusion

Here we will tune the SVM with proper hyper parameter as it is computational very costly. Also we will try to use other advanced regression model like ANN and LSTM. Also futher observation of performance metric will also have to done.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC