Stocks Closing Price Prediction Using Machine Learning

Name:	**Shukla Shivam Ashokbhai**
Registration No./Roll No.:	19301
Institute/University Name:	IISER Bhopal
Program/Stream:	e.g., EECS
Problem Release date:	February 02, 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

For the given regression problem, I have used five ML regression algorithms:
Linear Regression, KNN, Decision Tree, Random Forest and Artificial Neural Networks. I have used two methods of feature selection i.e Correlation Feature Selection and Mutual Info Regression. As the number of features are only four so,I have a randomly choose two feature at time and trained them on different models and observed the change in error. Like choosing first ["Open","Low"] then ["Open","High"] and so on.

For hyper-parameter tuning I have used Grid Search with cross-validation with 10 partitions on all the above five algorithms to find the best parameters which minimizes the error. During Tuning I have first randomly gave the grid of parameters to each model and then got the best out of them , again I make new grid of parameters depending upon the previous best parameters obtained by grid-search and then run the grid search to got new set of parameter which can minimize the error further. I have iteratively performed this process until there is no further decrease in the error.

Further I have tried to made an experimental method to minimize the error further using two best model obtained after Hyper-parameter tuning.
Method:
let best model be M1 and M2
step1: train the M1 and M2 with there best set of parameters and get y1 and y2 as prediction on training set.
step2: Combine y1 and y1 and form an Data-set X = [y1,y2]
step3: Train this X as feature and target value from original data on one of the best models M1 or M2.

step4: Get get the prediction on test set.
step5: Compare the result with best model score.

From all above process model/method with there best parameters will be used for prediction on test set for submission.

GitHub line here

3 Evaluation Criteria

Mean Average Percentage Error (MAPE): MAPE is the percentage error.

$$MAPE = \frac{1}{n}\Sigma_{i=1}^{n}|\frac{A_i-F_i}{A_i}|$$

Ai : True value of Closing Price.
Fi : Predicted Value of Closing Price.

MAPE is the better way to describe the error in this model. For different companies stocks closing price vary, they are not in some range some may have price of 800 and some have price 100. For this mean squared error(MSE), or mean absolute error(MAE) will not give intuitive estimate of error. For example MAE of 80 for 800 is 10 percent, but for other company which has price 100 its 80 percent so we can't use MAE or MSE for Evaluation. On other hand MAPE gives proper estimate of error irrespective of variation in closing price of company.

4 Analysis of Results

Feature Selection:
From both classical methods i.e Correlation feature selection the Mutual Info Regression the scores for the features ["Open","High","Low"] where comparatively high than ["Volume"]. From randomly

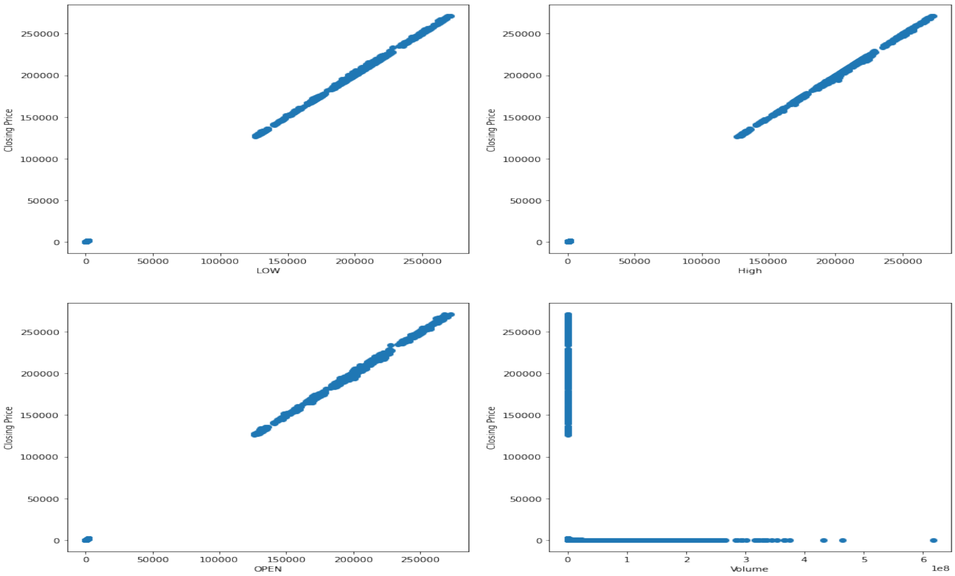

Figure 1: Scatter Plot of All the features with Closing Price(Target)

Dropping the features I got the following results.

	Models				
Feature Selected	Linear	Decision Tree	Random Forest	KNN	ANN
("Open","Low")	0.010	0.007	0.006	0.60	0.01
("Open","High")	0.009	0.007	0.006	0.60	0.010
("High","Volume")	0.009	0.007	0.006	0.55	68.605
("Low","High")	0.004	0.006	0.0.005	0.500	0.050
("All features")	0.004	0.005	0.0.004	0.492	0.050

Table 1: Feature Selection

All the values in the table are in MAPE (Mean Absolute Percentage Error)

From above we can see that we are getting the best result when all the feature is selected. I have than tuned the hyper-parameters of all the above models using grid search and got the results as:

	Linear	Decision Tree	Random Forest	KNN	ANN
MAPE	0.0037	0.0079	0.0039	0.5035	0.618

Table 2: Error After Hyper-Parameter Tuning

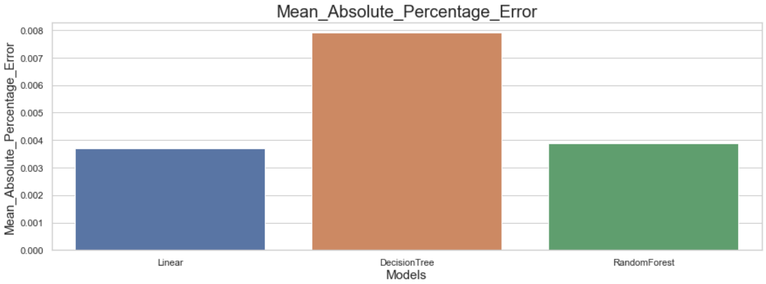

Figure 2: Error for three optimum models

The best Parameter for all the models:

Linear:LinearRegression(normalize=True)

Decision Tree:DecisionTreeRegressor(ccp-alpha=0.0009,criterion='absolute-error',max-depth=10,splitter = "best")

Random Forest: RandomForestRegressor(max-depth=20, max-features=None, n-estimators=400)

KNN: KNeighborsRegressor(algorithm = "auto",leaf-size=1, n-neighbors=1)

ANN: MLPRegressor(alpha=0.001, hidden-layer-sizes=[45, 25, 15],learning-rate-init=0.01)

From here we can conclude that Linear Regression and Random Forest are the two best models which least MAPE. Futher we have choose M1 = Linear Regression and M2 = Random Forest Regression and Implemented the error reducing method which was discussed in Method Section.
Results of the method:
When Linear Regression is choose in Step3:

```
MAPE: 0.003796
```

When Random Forest is choose in Step3:

```
MAPE: 0.004144
```

Here, we can conclude that this method to reduce the error does not worked and we got the nearly same result which we got in Linear Regression. Hence, from all above process of training and tuning of different models with various methods we conclude that **Linear Regression** is the best model for our problem with **MAPE of 0.0037**.

5 Discussions and Conclusion

From the above findings we can infer that all the complex regression model of Machine Learning gives low performance than simple linear regression. Linear regression gives best results than all other regression algorithm and it is less computationally expensive than all other models. We can Conclude that features of the data has high linear correlation with close price.

In general Artificial Neural Network gives best results on any regression problem but here I had tried many set of hyper-parameters but didn't get the results close to any of the optimal model. For further work we can design the ANN/Deep-learning model which may give more accuracy than Current Best Linear Regression. Significant findings and scopes of future works may be explained here in few sentences.