Prediction of Closing Prices of Various Stocks

Name:	**Eshan Kale**
Registration No./Roll No.:	17088
Institute/University Name:	IISER Bhopal
Program/Stream:	Maths
Problem Release date:	January 19, 2022
Date of Submission:	April 24, 2022

1 Introduction

The data we have consists of Open, Low, High, Close and Volume of multiple stocks across different. Each row corresponds to data of one day. Our aim is to create a model which predicts the closing price so that we will be able to minimize the loss and maximize the profit. We will try to analyse the dataset and apply various algorithms to predict the closing price.

First I checked for any NaN values and remove them. I added two new columns "$TradedValue$" $= Open \times Volume$ and "$HL\ Percent$" $= \frac{(High-Low)}{High}$. I created a copy of our dataset adding two the columns. After applying some algorithms it was found that there was not much change after adding the two datasets. I started by finding correlation matrix.

Table 1: Correlation Matrix

	Open	High	Low	Volume	Traded Value	HL Percent	Close
Open	1	0.999988	0.999988	-0.054682	0.070580	-0.055169	0.999974
High	0.999988	1	0.999987	-0.054687	0.070740	-0.055021	0.999989
Low	0.999988	0.999987	1	-0.054680	0.070541	-0.055321	0.999988
Volume	-0.054682	-0.054687	-0.054680	1	0.629466	0.161610	-0.054684
Traded Value	0.070580	0.070740	0.070541	0.629466	1	0.148275	0.070689
HL Percent	-0.055169	-0.055021	-0.055321	0.161610	0.148275	1	-0.055158
Close	0.999974	0.999989	0.999988	-0.054684	0.070689	-0.055158	1

As I can see that Open, High, Low and Close are highly correlated. Also, since these values are so close, we will have high accuracy. To distinguish between the wrong predictions and correct predictions better, I defined some new metrics and evaluated the models.

2 Methods

I applied Linear Regression, Ridge Regression and Lasso Regression with the help of sklearn [1]. Since Linear Regression did not have many parameters, hyper parameter was done for Ridge and Lasso Regression only. For these 3 techniques I tried both datasets. It was found that there was not much difference in the results. After that I applied Random Forest Regression. Since the computation time was higher I was not able to apply Grid Search.

I then applied KNeighbors Regressor. First I used it on multiple values of K and then plotted the graph of K versus mean squared error. It was observed that the plot was not smooth. This was happening because Volume had extremely high values and this was affecting the results. I scaled the data and reapplied the algorithms. There was a slight difference in the previous algorithms. As for KNeighbors Regressor, I got the following graph.

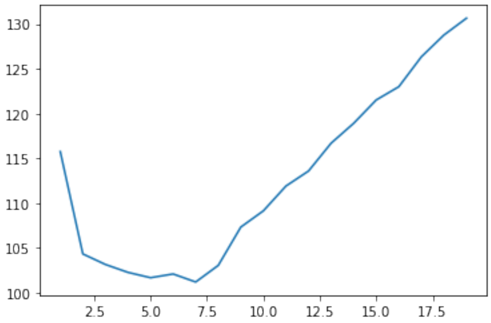

Figure 1: Number of Neighbors vs. Mean Squared Error

I then tried to use Artificial Neural Networks with help of Keras [2]. I first started with no hidden layer, then I went onto add multiple hidden layers and changed the number of units in each layer. I also changed the optimizers, learning rates for different optimizers and I tried different activations functions for each layer. Github link for this project is included here.

3 Evaluation Criteria

As mentioned previously I defined new metrics Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Bias Error (MBE) (Refer [3]). Here n denotes the number of samples, A_i is the actual value of ith sample and P_i is the predicted value of ith sample.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(A_i - P_i)^2}{n}}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - P_i|}{A_i}$$

$$MBE = \frac{1}{n}\sum_{i=1}^{n}|(A_i - P_i)|$$

I evaluated the models on basis of these metrics and r2-score.

For an ideal prediction, all these three values should be zero and r2-score should be 1. Also none of these newly defined metrics can be negative.

4 Analysis of Results

I have not included the result of Neural Networks since none of the results were good. The results were highly based on the initial state. The model yielded good results less than 20% of times. The results had r2 score either 0.98 or -3.17. The following table shows best results for each model.

Table 2: Performance Of Different Regressors

Model	Number of features	r2-score	RMSE	MAPE	MBE
Linear	4	0.9999899	0.00311011353	0.00021106	0.00025577
Linear	6	0.9999899	0.003110196629	0.00022861	0.00025765
Ridge	4	0.9999899	0.003110085880	0.00021453	0.00025614
Ridge	6	0.9999899	0.003110169117	0.00022504	0.00025724
Lasso	4	0.9999899	0.003110085880	0.00021453	0.00025614
Lasso	6	0.9999899	0.003110169117	0.00022504	0.00025724
Random Forest	4	0.9999815	0.00421508004	0.00026429	0.0003658
KNeighbours	4	0.9999789	101.2136821	0.03763835	10.35174269

Here if the number of features is 4 then the model is fitted on dataset with features ”Open”, ”High”, ”Low” and ”Volume”. Otherwise the features ”Traded Value” and ”HL Percent” are also included.

As we can see, the r2-score is not a good measure and the variants of linear regression have descent result with the best being linear regression with just 4 features. Note that we need extremely accurate data since the stock prices may not change much. Also MAPE is more important in our evaluations as compared to other metrics because, MAPE considers the percentage difference. A $2 error will have more significance for stock with price $20 than a $200 stock.

5 Discussions and Conclusion

We determined that Linear Regression actually gave best results. Incorrect predictions for Artificial Neural Networks might be due to wrong convergence since we were able to get good predictions sometimes due to good initialized vectors. From this we can conclude that the data was actually not good for ANN. Also, although KNeighbors had good r2-score, the other metrics were highly deviated.

We were able to determine that the traded value and HL percent did not have much effect. Since we did not know any information about which sector the stock belonged to or how the stocks are arranged (date), much information was missing. We can get better results by restricting the stocks to single sector and possibly include date of the stock.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[2] François Chollet. keras. `https://github.com/fchollet/keras`, 2015.

[3] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar. Stock closing price prediction using machine learning techniques. *Procedia computer science*, 167:599–606, 2020.