MGS 616 PREDICTIVE ANALYTICS FINAL REPORT

NEW YORK STOCK EXCHANGE DATA

INTRODUCTION

The data used in this project is from the New York Stock Exchange, dated from 2010 to 2016, sourced from Kaggle.

Research Question: How accurately can we predict the "Closing" stock price of a company based on its "Open", "Low", and "High" prices? How can we leverage this model?

METHODS

The dataset is preprocessed first to ensure it is suitable for building the models. The first step of preprocessing is to filter out the stock price records of Amazon. Then the duplicate entries are removed, if any, and NULL values are checked for.

	open	close	low	high	volume
count	1762.000000	1762.000000	1762.000000	1762.000000	1.762000e+03
mean	337.875664	337.899058	333.969688	341.464438	4.607596e+06
std	189.294231	189.109339	187.654696	190.525796	3.091557e+06
min	105.930000	108.610001	105.800003	111.290001	9.844000e+05
25%	192.962494	193.377506	190.284997	195.532501	2.741550e+06
50%	282.500000	282.915008	279.869995	285.074997	3.890700e+06
75%	398.425003	398.014999	393.799988	402.082496	5.384450e+06
max	845.789978	844.359985	840.599976	847.210022	4.242110e+07

Fig. 1: Descriptive Statistics of Dataset

The following step is to check for correlation. Fig. 2 shows the "Open", "High", "Low" prices and "Volume" have a strong positive correlation with the "Close" price and amongst each other. Thus, to reduce complexity, "Open", "High", "Low" prices are chosen, dropping off "Volume".

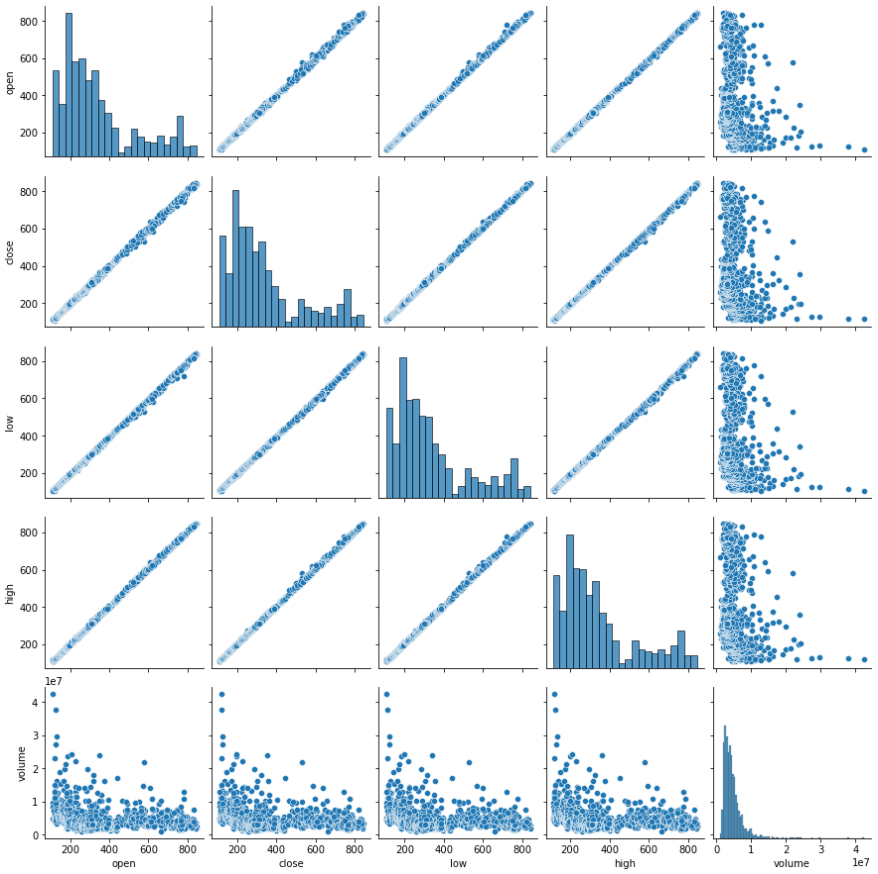


Fig. 2: Correlation Plots

After finalizing the variables, the dataset is split Training data with **80 percent** of the records and into Testing data with the remaining **20 percent.**

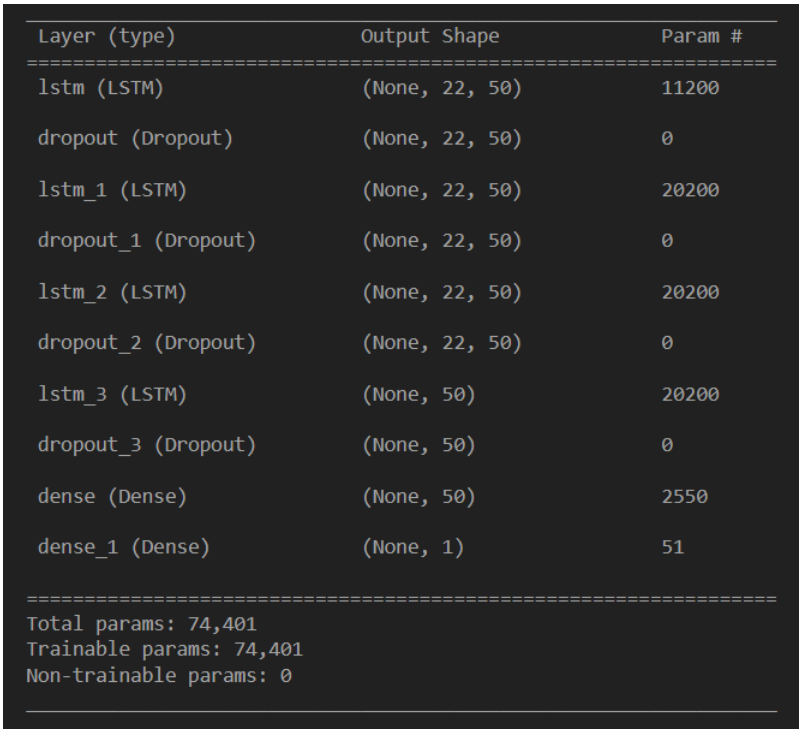
Two models have been dealt with, a naive approach, i.e., Linear Regression and Long-Short Term Memory model, which is an exclusive Recurrent Neural Network.

LINEAR REGRESSION

An instance of a simple Linear Regression is created, and the training data is fed into the regressor to fit the model along a linear regression line. The testing data is now used to validate the training by passing the test variables and consequently obtaining the values on the fitted line. This predicted value is then checked with the actual test values and calculated for accuracy and error.

LONG SHORT-TERM MEMORY (LSTM)

LSTM is a specific RNN that is used to satisfy the prediction requirement.



Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 22, 50)	11200
dropout (Dropout)	(None, 22, 50)	0
lstm_1 (LSTM)	(None, 22, 50)	20200
dropout_1 (Dropout)	(None, 22, 50)	0
lstm_2 (LSTM)	(None, 22, 50)	20200
dropout_2 (Dropout)	(None, 22, 50)	0
lstm_3 (LSTM)	(None, 50)	20200
dropout_3 (Dropout)	(None, 50)	0
dense (Dense)	(None, 50)	2550
dense_1 (Dense)	(None, 1)	51

Total params: 74,401
Trainable params: 74,401
Non-trainable params: 0

Fig.3: Layers of the LSTM model

Each record of the linear data that we make use of can be represented in the form of a linear expression, where the coefficients are represented as weights of a node in the LSTM. Through multiple times of processing (epochs), we gradually adjust the weights so as to fit them into the linear relationship such that when the input variable values are fed in, the most accurate output is yielded (back-propagation). LSTM has a memory cell, to remember the cell state which helps in efficient processing. The epoch given in this instance is **100**. The activation function is the **Linear activation function.** Dropout Layers are used to drop a few weights by deactivating them so that the model doesn't memorize the values thereby avoiding overfitting.

RESULTS

LINEAR REGRESSION

The **Train** and **Test score** is **0.99** and **0.99**. The **Mean Absolute Error (MAE)** is **1.56**.

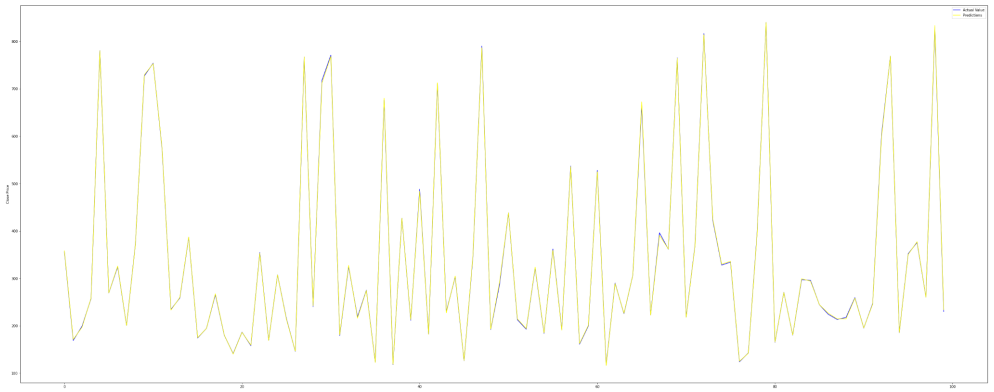


Fig.4: Actual Test values and Predicted Test values comparison (LR)

The train accuracy score might first give off the impression that the model could be overfitted. But the test accuracy score proves that the model is efficient in predicting new data as well. Fig 4 proves how prominently the actual and the predicted test values are overlapped with no overfitting.

LONG SHORT-TERM MEMORY (LSTM)

The **Train** and **Test score** is **0.86** and **0.32**. The **RMSE** for **Train data** is **17.09** while for **Test data RMSE** is **67.27**.

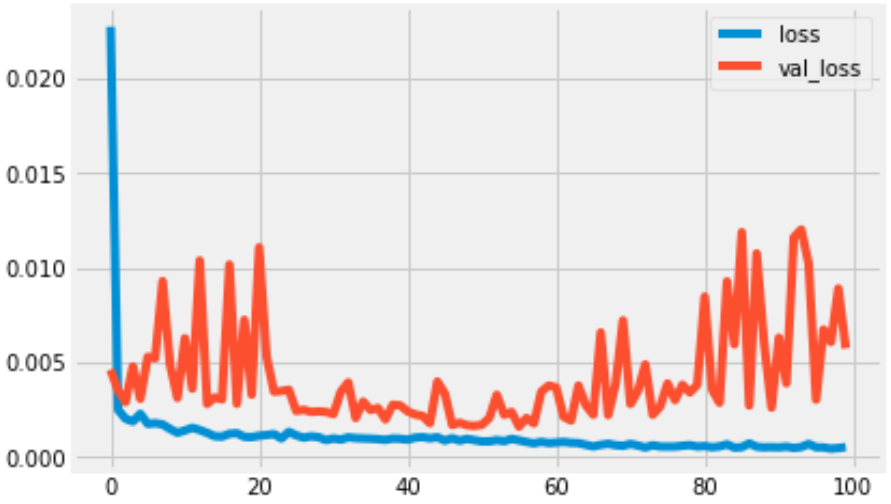


Fig. 5: Train and Validation Loss

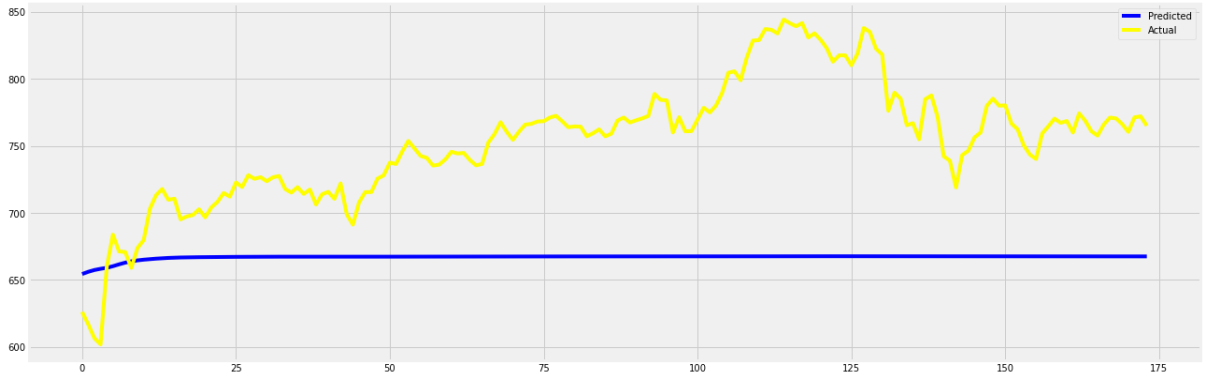


Fig.6: Actual Test values and Predicted Test values comparison (LSTM)

The LSTM model has poor accuracy scores i.e. the test accuracy is way lesser than the Train accuracy which is a clear indication of overfitting. The losses from Fig. 5 prove the same. Fig. 6 shows that the prediction is centered around a certain value, maybe, has arisen due to a bias from overfitting.

CONCLUSION

It is concluded that the basic, naive approach itself is good at predicting the Closing Stock Prices compared to the complex LSTM method. The efficiency of the LSTM can be improved by adding more Dropout Layers to reduce overfitting. Additional LSTM layers can also be added to arrive at more accurate weight values. To increase the training time, epochs can also be increased. Accurately predicting the stock market helps investors make better monetary decisions, manage their risks by identifying potential losses and reducing their vulnerability to market turbulence. It also helps companies understand their own value better and know how their financial performance is comprehended and received by the public. Hence, there is an inevitable reason for putting the Linear Regression model into practical use. It can be deployed as an application (mobile/web) where ,when the input variables are provided, the best closing price is predicted with which the users can assess the market situation. The cost of building such an app is optimal given that prolific benefits can be achieved on account of the model's efficiency.