PREDICTION OF AMAZON STOCKS USING DEEP LEARNING (LSTM)

Aniruddha Kashinath Shelke and Sumaiya Shaikh

Department of Statistics, V.P.M.'s B.N.Bandodkar College of Science, Thane-400601

Abstract: Amazon is the most valuable company and the founder of Amazon Mr. Jeff Bezos is in the elite list of wealthy peoples around the world before starting Amazon Mr.Jeff Bezos used to work as a vice-president in a investment firm in New York , he saw some serious potential of the growth of internet in the future and thought of using that potential of internet to implement in his business, further he listed 20-25 products whom which he can sell online and then he finalised books, after finalising books he went to seattle to look out for some software engineers and book distribution centers, In November 1994 he bought a garage and started his business with 2 programmers. Mr . Bezos took financial help of his wife and registered his company by the name CADABRA further he went to name his company as Amazon, and on 16th July 1995 he launched the website Amazon.com, the prices of book were less/cheap as compared to book store, at that time YAHOO ! used to publish What's Cool list, 3 days later after launching the Amazon.com website it got featured in that list and in the 1st week orders worth

$12,438 were placed. Amazon used the beep sound functionality (when any orders were placed a beep sound was produced), as time passed amazon got more bigger with their website, warehouses, fundings, further the company became a global brand and an e-commerce website which is able to sell any product .

OBJECTIVES :

The research project objectives are proposed as follows :

Objective 1**:** The main objective of the research is to predict Amazon stock values using LSTM model .

Objective 2 **:** Importing the data and checking for null values Visualising the dataset , Training and testing certain amount of dataset , using optimizer for a small amount of batch size at lastpredicting the results.

METHODOLOGY :

Data collection : The research is based on secondary data which has been collected from the following website :

https://www.kaggle.com/kaushiksuresh147/faang-fbamazonapplenetflixgooglehttps://www.kaggle.com/kaushiksuresh147/faang-fbamazonapplenetflixgoogle-stocks?select=AMZN+Historical+Data.csvstocks?select=AMZN+Historical+Data.csvthe data set contains 6 columns are as follows :

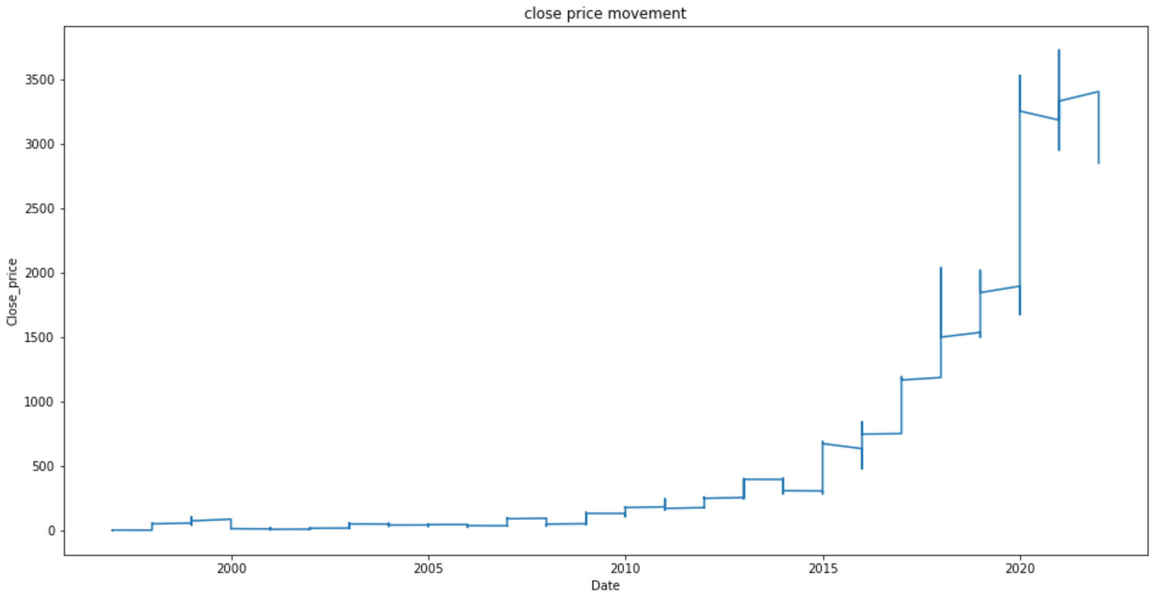
1. Date : Date of observation.
2. Open : Opening price of the respective day.
3. High : Highest price of the respective day.
4. Low : Lowest price of the respective day.
5. Close : Closing price of the respective day.
6. Volume : Volume of transactions on the respective day.

The data till 21-Jan-2022 was included in the research work.

From the following source the data was imported in jupyter notebook using pandas packages and was visualized using data visualising library matplotlib.pyplot.

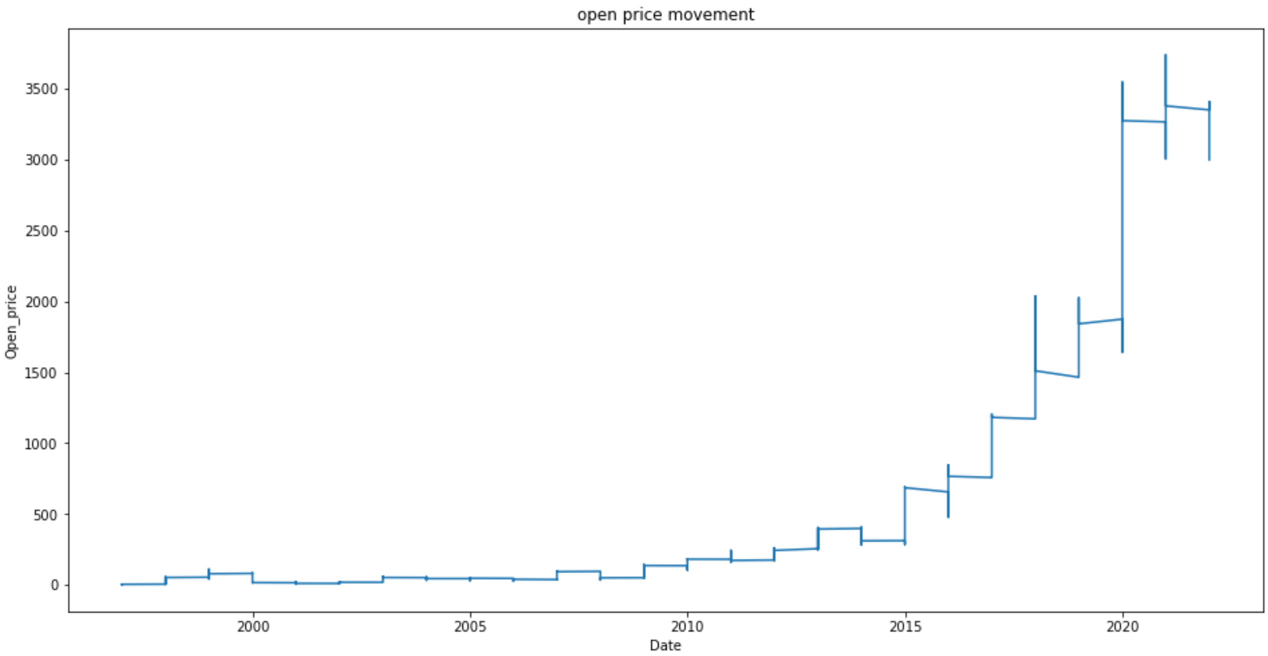
ANALYSIS:

The close price movement chart shows how the prices were changed over the years the chart was visualized using (matplotlib.pyplot) library in python.



On X-axis the the scale has a 5 year interval.

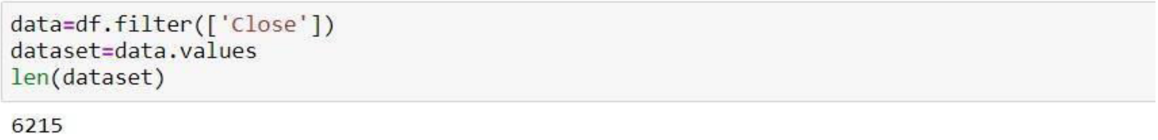
The open price movement chart shows how the prices were changed over the years the chart was visualized using (matplotlib.pyplot) library in python.



On X-axis the the scale has a 5 year interval.

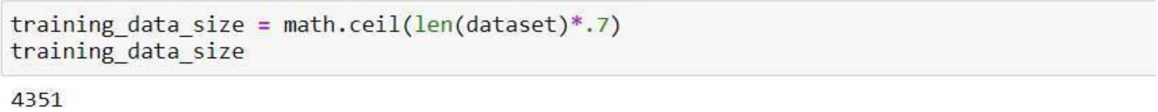
The data is divided in to two parts that’s training and testing
70% of the dataset is used for training purpose and rest 30% was used for testing.

The data is divided in to two parts that’s training and testing.



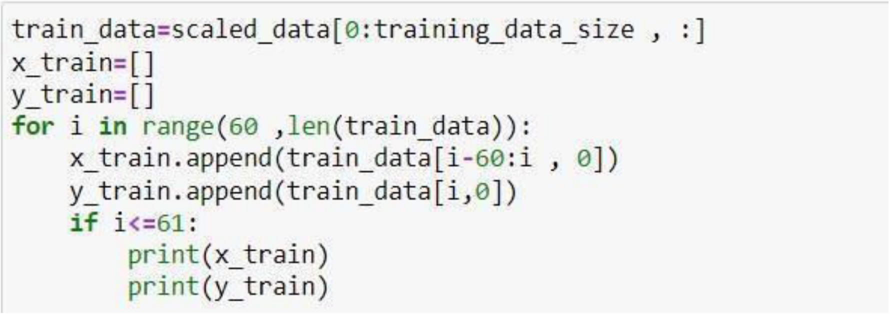
```
data=df.filter(['Close'])
dataset=data.values
len(dataset)

6215
```



```
training_data_size = math.ceil(len(dataset)*.7)
training_data_size

4351
```

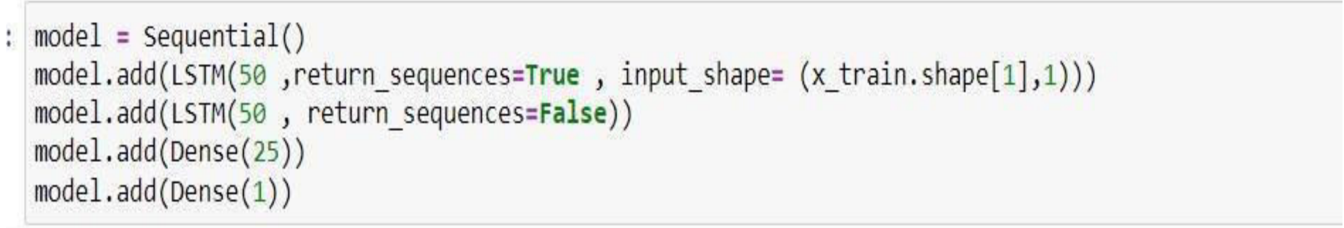


```
train_data=scaled_data[0:training_data_size , :]
x_train=[]
y_train=[]
for i in range(60 ,len(train_data)):
    x_train.append(train_data[i-60:i , 0])
    y_train.append(train_data[i,0])
    if i<=61:
        print(x_train)
        print(y_train)
```

For testing the dataset LSTM Model was used basically LSTM stands for Long Short Term Memory.LSTM networks were designed specifically to overcome the long-term dependency problem faced by recurrent neural networks RNNs . LSTMs have feedback connections which make them different to more traditional feedforward neural networks. This property enables LSTMs to process entire sequences of data (e.g. time series) without treating each point in the sequence independently, but rather, retaining useful information about previous data in the sequence to help with the processing of new data points. As a result, LSTMs are particularly good at processing sequences of data .An LSTM network can learn this pattern that exists every unit of time. It doesn't just use the previous prediction but rather retains a longer-term context which helps it overcome the long-term dependency problem faced by other models. It is worth noting that this is a very simplistic example, but when the pattern is separated by much longer period of time. Firstly, at a basic level, the output of an LSTM at a particular point in time is dependant on three things:

1] The current long-term memory of the network — known as the cell state.

2] The output at the previous point in time — known as the previous hidden state.
3] The input data at the current time stamp which is the output of the previous stage.



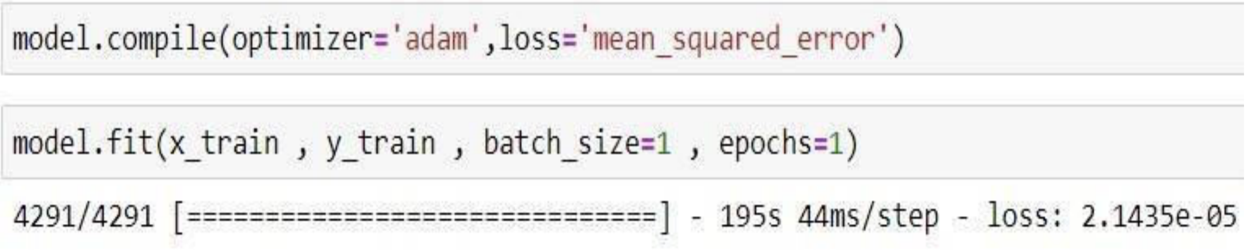
```
: model = Sequential()
  model.add(LSTM(50 ,return_sequences=True , input_shape= (x_train.shape[1],1)))
  model.add(LSTM(50 , return_sequences=False))
  model.add(Dense(25))
  model.add(Dense(1))
```

This LSTM model consists of 4 layers the first is an input layer and the last is an output layer.

Using keras library the optimizer adam was used.

ADAM OPTIMIZER: Adam optimizer is an adaptive moment estimation , the optimizer updates the learning rate for each network weight individually , In adam instead of adapting learning rates based on first moments i.e mean it also uses the second moment . the adam optimizer has a faster running time and low memory requirements than any other optimizer /optimization algorithm .

Generally a batch size of 32,64 or 108 is recommended but for this research work the batch size of 1 was preferred.



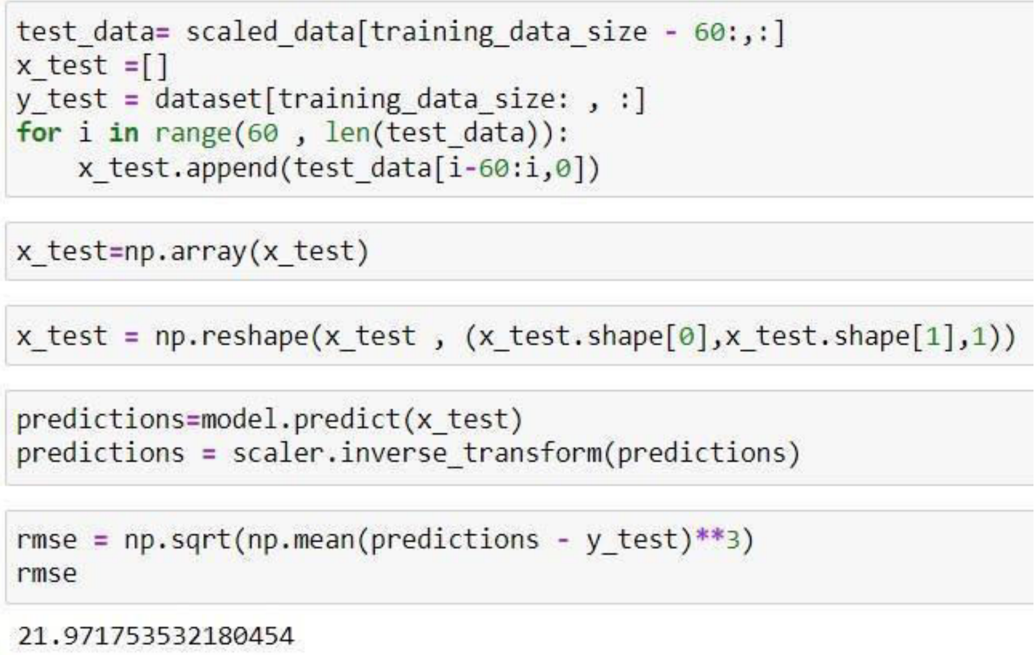
```
model.compile(optimizer='adam',loss='mean_squared_error')
```

```
model.fit(x_train , y_train , batch_size=1 , epochs=1)
```

```
4291/4291 [==============================] - 195s 44ms/step - loss: 2.1435e-05
```

TESTING THE DATA :

As mentioned earlier 30% of the data set was used for testing purpose



```
test_data= scaled_data[training_data_size - 60:,:]
x_test =[]
y_test = dataset[training_data_size: , :]
for i in range(60 , len(test_data)):
    x_test.append(test_data[i-60:i,0])
```

```
x_test=np.array(x_test)
```

```
x_test = np.reshape(x_test , (x_test.shape[0],x_test.shape[1],1))
```

```
predictions=model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
```

```
rmse = np.sqrt(np.mean(predictions - y_test)**3)
rmse
```

```
21.971753532180454
```

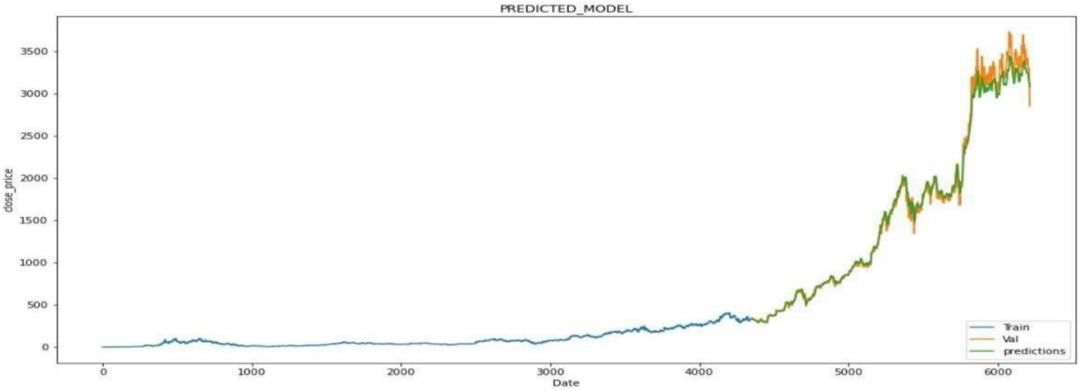
The data was scaled using MinMaxScaler and the testing was done for 60 time stamps.

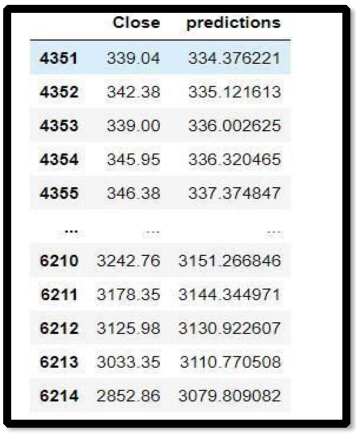
EXPECTED OUTCOMES :

Both of the datasets which were trained and tested are merged which gave us the final result of predicted closing price

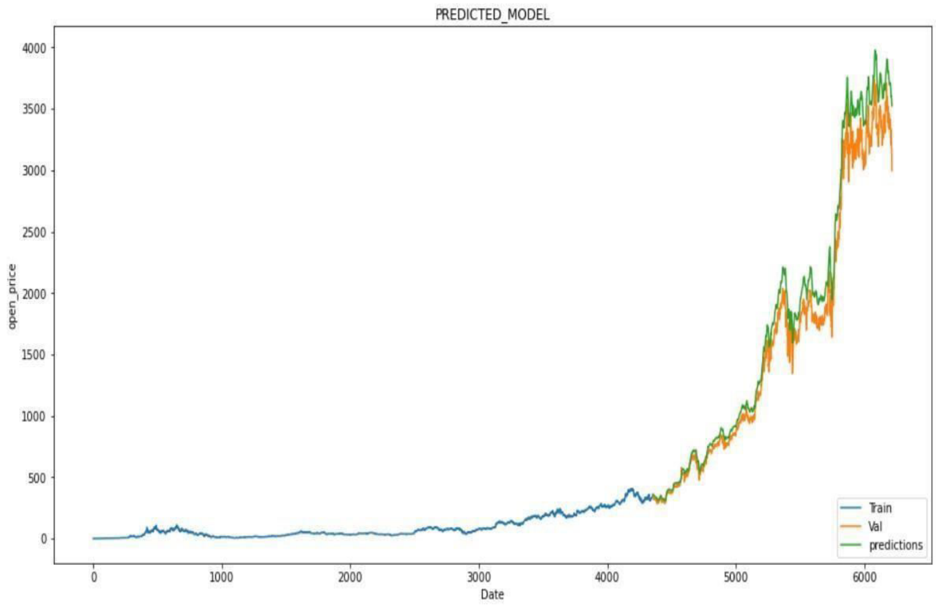
The predicted model came out with an accuracy of 78%.

Here are the Actual values and Predicted values for Close price :

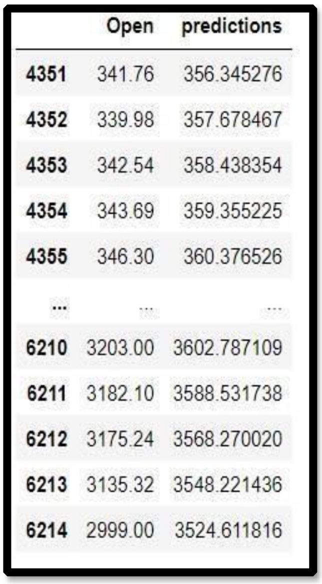




	Close	predictions
4351	339.04	334.376221
4352	342.38	335.121613
4353	339.00	336.002625
4354	345.95	336.320465
4355	346.38	337.374847
...	...	...
6210	3242.76	3151.266846
6211	3178.35	3144.344971
6212	3125.98	3130.922607
6213	3033.35	3110.770508
6214	2852.86	3079.809082



Here are the Actual values and Predicted values



	Open	predictions
4351	341.76	356.345276
4352	339.98	357.678467
4353	342.54	358.438354
4354	343.69	359.355225
4355	346.30	360.376526
...	...	...
6210	3203.00	3602.787109
6211	3182.10	3588.531738
6212	3175.24	3568.270020
6213	3135.32	3548.221436
6214	2999.00	3524.611816

The same was applied for Open price movement and the conclusion made was the predicted values deviates less for the initial phase but shows more deviation for the later

BENEFITS: By using the concept of Data visualisation, Graph Deep learning and Regression Models it becomes easier to explain what’s happening in the stock market and how the prices are affected at any time of the day. So anyone from non-statistics/non commerce background is able to understand the concept and is able to pass a decision.

FUTURE SCOPE:

The Model can be more optimized using ADAGRAD optimizer and more advanced Deep learning techniques to improve the accuracy.

SWOC ANALYSIS:

- **STRENGTH :** As Keras is regarded as the most powerful library for deep learning purpose. This LSTM Model can predict the closing price for any stock with an accuracy above 80%.

- **WEAKNESS :** When it comes to the open price prediction there's an immense amount of variation for the later stages.
- **OPPORTUNITY :**Not only for stock price predicton but this model can also be used for making a prediction for a whole new topic e.g . (Predicting COVID-19 cases , Time Series Prediction ,etc.).
- **CHALLENGES :**For an individual from a non-commerce background it becomes very challenging for him/her to understand the concept of Optimizer futher when to use it (i.e whats's the required batch size , total epochs). It takes a lot of research work to study a particular Optimizer and to grasp the overall concept which is a time draining process.

CONCLUSION :

The data was imported and no null values were found and was visualized using matplotlib.pyplot and was analysed further.LSTM model with 4 layers was imposed for prediction.Adam optimizer was used for a batch size of 1 and the trained and tested sets were merged to predict the prices for open as well as close movement.

REFERENCES :

1. ADAM OPTIMIZER : https://keras.io/api/optimizers/adam/https://www.youtube.com/watch?v=tuU59-G1PgU
2. LSTM : https://www.youtube.com/watch?v=S8tpSG6Q2H0