|  |  |
| --- | --- |
|  |  |
| Tesla Stock price Prediction |  |
|  |  |
|  | SubjectData Mining |
|  | ABUBAKARSIDDIQUE(BAIM-s20-006)Hassan Razzaq(BDSM-s20-001)Abdulrafay(BAIM-s20-005)Arshi Iftikhar |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | INSTRUCTIONSImplementation of Stock Price Prediction in Python 1. Importing Modules  2. Loading and Preparation of Data  3. Understanding the Data  3.1 Getting Unique Stock Names  3.2 Extracting Data for a specific stock name  3.3 Visualizing the stock data  4. Creating a new Data frame and Training data  5. Building LSTM Model  6. Compiling the Model  7. Testing the model on testing data  8. Error Calculation  9. Make Predictions  10. The Actual vs Predicted Values For the project, we will be using basic modules likeNumPy,pandas, andmatplotlib.In addition to this, we will be using some submodules of karas to create and build our model properly.We would also require the math module for basic calculation and preprocessing module of sk learn to handle the data in a better and simpler way. | |  |

**1. Importing Modules**

First step is to import all the necessary modules in the project.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense, LSTM

import math

from sklearn.preprocessing import MinMaxScaler

For the project, we will be using basic modules like numpy, pandas, and matplotlib. In addition to this, we will be using some submodules of keras to create and build our model properly.

We would also require the math module for basic calculation and preprocessing module of sklearn to handle the data in a better and simpler way.

**2. Loading and Preparation of Data**

For the project we will be using the all\_stocks\_5yrs csv file which includes stock data for 5 years and has seven columns which are listed below.

Date – Format of date is: “yy-mm-dd”

Open – Price of the stock at open market

High – Highest price reached in the day

Low – Lowest price reached in the day

Close – Price of the stock at the close market

Volume – Number of shares traded

Name – The name of the stock ticker

1

2

data=pd.read\_csv("all\_stocks\_5yr..csv")

data.head()

The head function displays first five rows of the dataset.



**3. Understanding the Data**

**3.1 Getting Unique Stock Names**

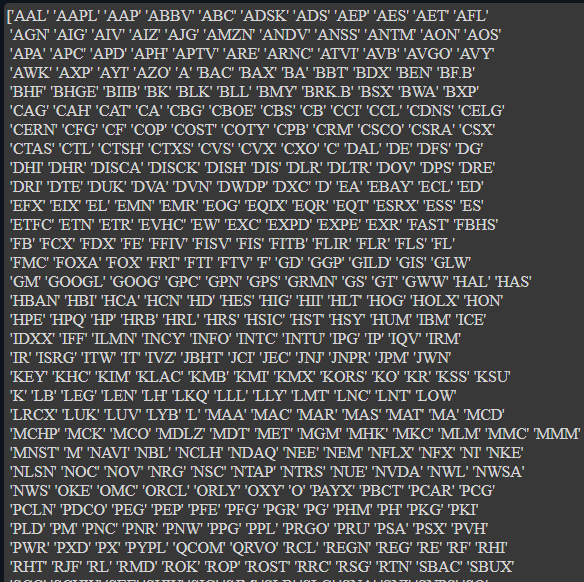
From the whole dataset, we will first extract all the unique stock ticks name with the help of unique function. In the dataset, we have 444 different stock names.

1

2

all\_stock\_tick\_names = data['Name'].unique()

print(all\_stock\_tick\_names)



**3.2 Extracting Data for a specific stock name**

We will try to understand how the stock data works by taking an input of a stock name from the user and collecting all data of that particular stock name.

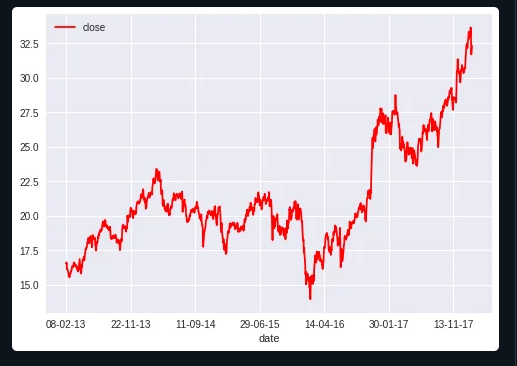
To visualize the data we will be first plotting the date vs close market prices for the FITB stock for all the data points.

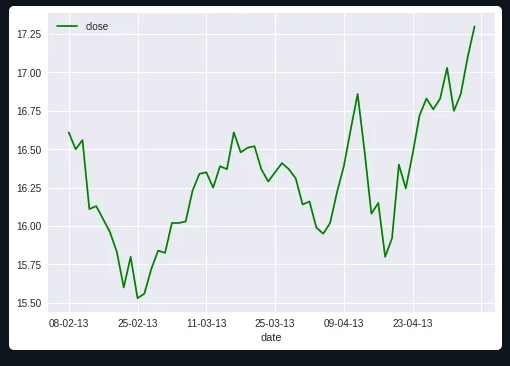
**3.3 Visualizing the stock data**

To visualize the data we will be first plotting the date vs close market prices for the FITB stock for all the data points.

To make the visualization simpler, we would be plotting the same plot but for only the first 60 data points.

plt.show()kk





**4. Creating a new Dataframe and Training data**

To make our study easier we will only consider the closing market price and predict the closing market price using Python. The whole train data preparation is shown in the steps below. Comments are added for your reference.

**5. Building LSTM Model**

The LSTM model will have two LSTM layers with 50 neurons and two Dense layers, one with 25 neurons and the other with one neuron.

**6. Compiling the Model**

The LSTM model is compiled using the mean squared error (MSE) loss function and the adam optimizer.

**7. Testing the model on testing data**

The code below will get all the rows above the training\_data\_len from the column of the closing price. Then convert the x\_test data set into the NumPy arrays so that they can be used to train the LSTM model.

As the LSTM model is expecting the data in 3-dimensional data set, using reshape() function we will reshape the data set in the form of 3-dimension.

Using the predict() function, get the predicted values from the model using the test data. And scaler.inverse\_transform() function is undoing the scaling.

**8. Error Calculation**

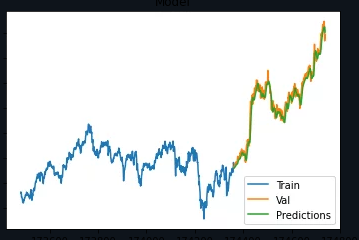
RMSE is the root mean squared error, which helps to measure the accuracy of the model.

The lower the value, the better the model performs. The 0 value indicates the model’s predicted values match the actual values from the test data set perfectly.

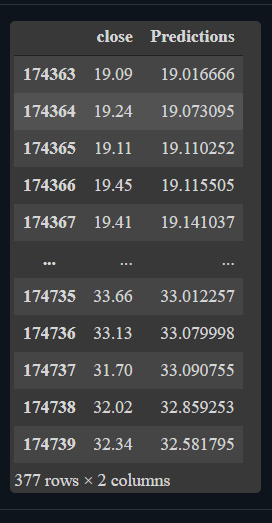
rmse value we received was 0.6505512245089267 which is decent enough.

**9. Make Predictions**

The final step is to plot and visualize the data. To visualize the data we use these basic functions like title, label, plot as per how we want our graph to look like.



**10. The Actual vs Predicted Values**

****

**Conclusion**

#### Today we learned how to predict stock prices using an LSTM model! And the values for actual (close) and predicted (predictions) prices match quite a lot.

**Thank you for reading!**