FORMAN CHRISTIAN COLLEGE (A CHARTERED UNIVERSITY)

COMP360-C

Spring’21

Final Project

Submitted by:

Abdullah Qureshi 221-437659

Mohsin Zahid 22-10234

Muhammad Muzafar 23-10903

Submitted to:

Ma’am Aasia Khanum

Date:

15th July 2021

Apple Stock Price Prediction

Data source:

API Yahoo

Dataset Name:

AAPL(Apple Inc. (AAPL) Stock Historical Prices &Data)

Variables in dataset:

- Date
- Stock Opening Price
- Stock Closing Price
- Number of shares traded
- Maximum Stock Price
- Minimum Stock Price
- Volume

Input variables:

- Date

Output variable:

- Predicted Stock Closing Price
- Actual stock Closing price
- Root mean square error.
- Accuracy of predicted stock price
- Closing stock price of given date by user

- Actual Closing stock price on that day

Background:

Apple Inc. is an American multinational technology company that specializes in consumer electronics, computer software, and online services. Apple is the world's largest technology company by revenue and, since January 2021, the world's most valuable company.

We are going to implement the model to predict the stock prices of apple by using machine learning methods like:

- **Deep Neural Networks** (DNN):
- **Recurrent Neural Networks (RNN)**
- **Long Short-Term Memory (LSTM)**

We will train our testing data using train data with ratio 80 and 20 percent respectively.

Then we will create LSTM model.

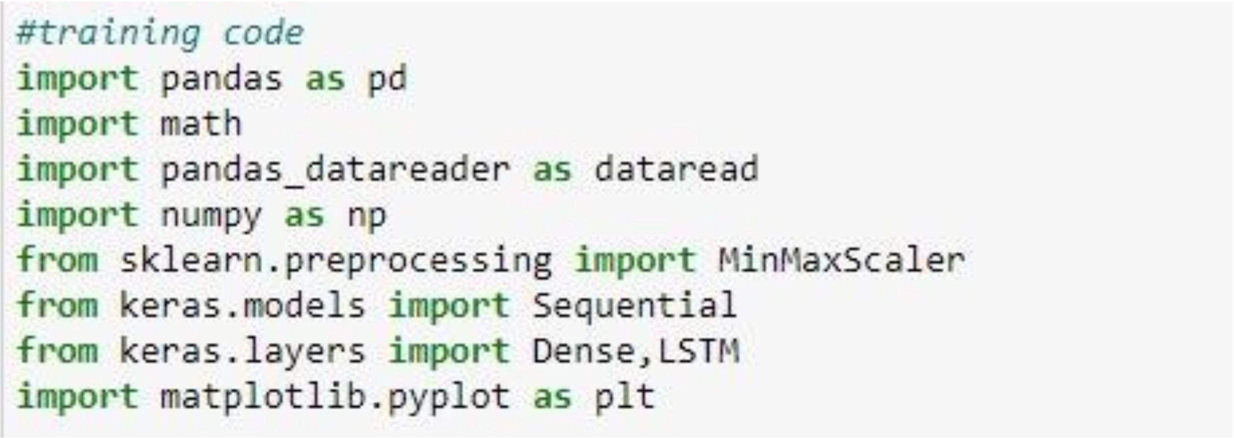
Used jupyter notebook for this project.

We used pandas_datareader library to extract dataset of apple stock from yahoo.

Library Used:

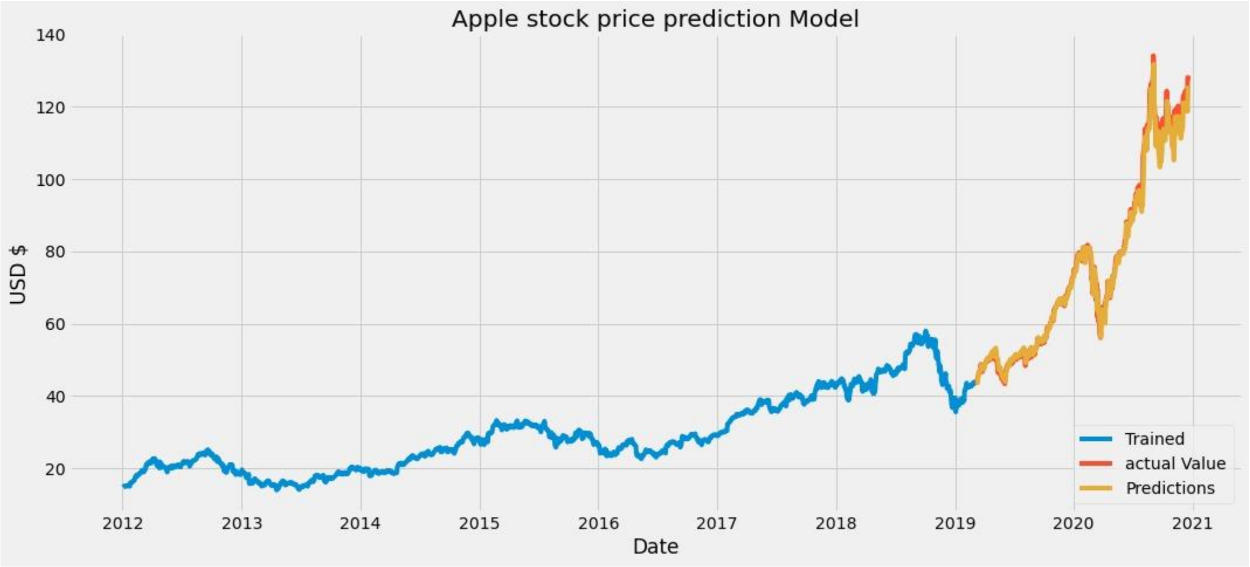
1 of the most important library which is required in this project:

Tensorflow

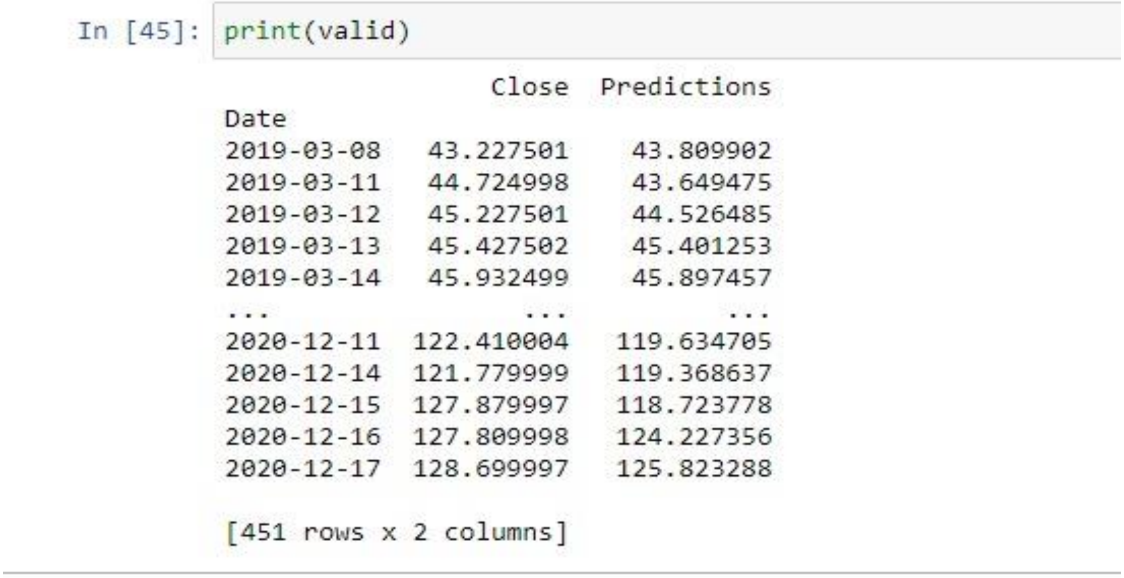


```
#training code
import pandas as pd
import math
import pandas_datareader as dataread
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense,LSTM
import matplotlib.pyplot as plt
```

Data after applying LSTM with prediction prices visualization using graph as you can see predictions are so close to actual values of apple closing stock prices.



Actual Value and Predicted Values by our model:

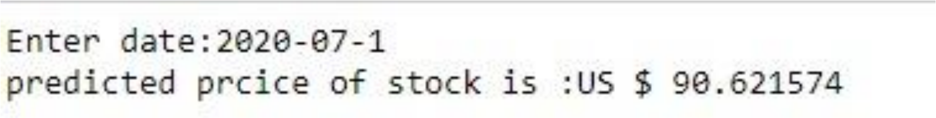


```
In [45]: print(valid)

                 Close  Predictions
Date
2019-03-08   43.227501    43.809902
2019-03-11   44.724998    43.649475
2019-03-12   45.227501    44.526485
2019-03-13   45.427502    45.401253
2019-03-14   45.932499    45.897457
...                ...          ...
2020-12-11  122.410004   119.634705
2020-12-14  121.779999   119.368637
2020-12-15  127.879997   118.723778
2020-12-16  127.809998   124.227356
2020-12-17  128.699997   125.823288

[451 rows x 2 columns]
```

Input and Predicted Value:

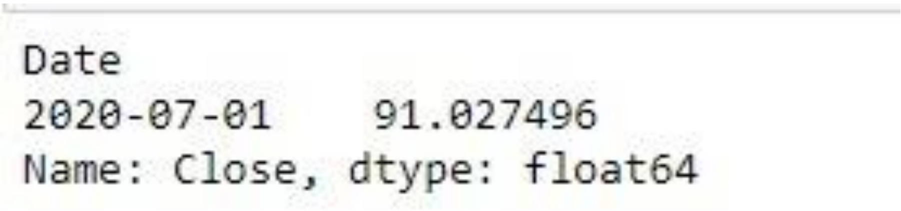
Input given by user

Prediction by our model



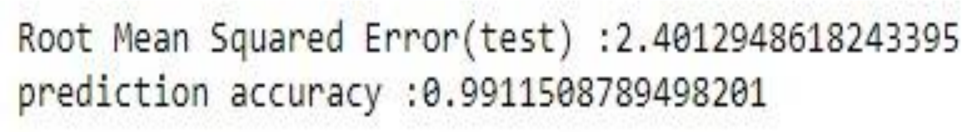
```
Enter date:2020-07-1
predicted prcice of stock is :US $ 90.621574
```

Actual Stock Price on the given day:



```
Date
2020-07-01    91.027496
Name: Close, dtype: float64
```

Prediction Accuracy:



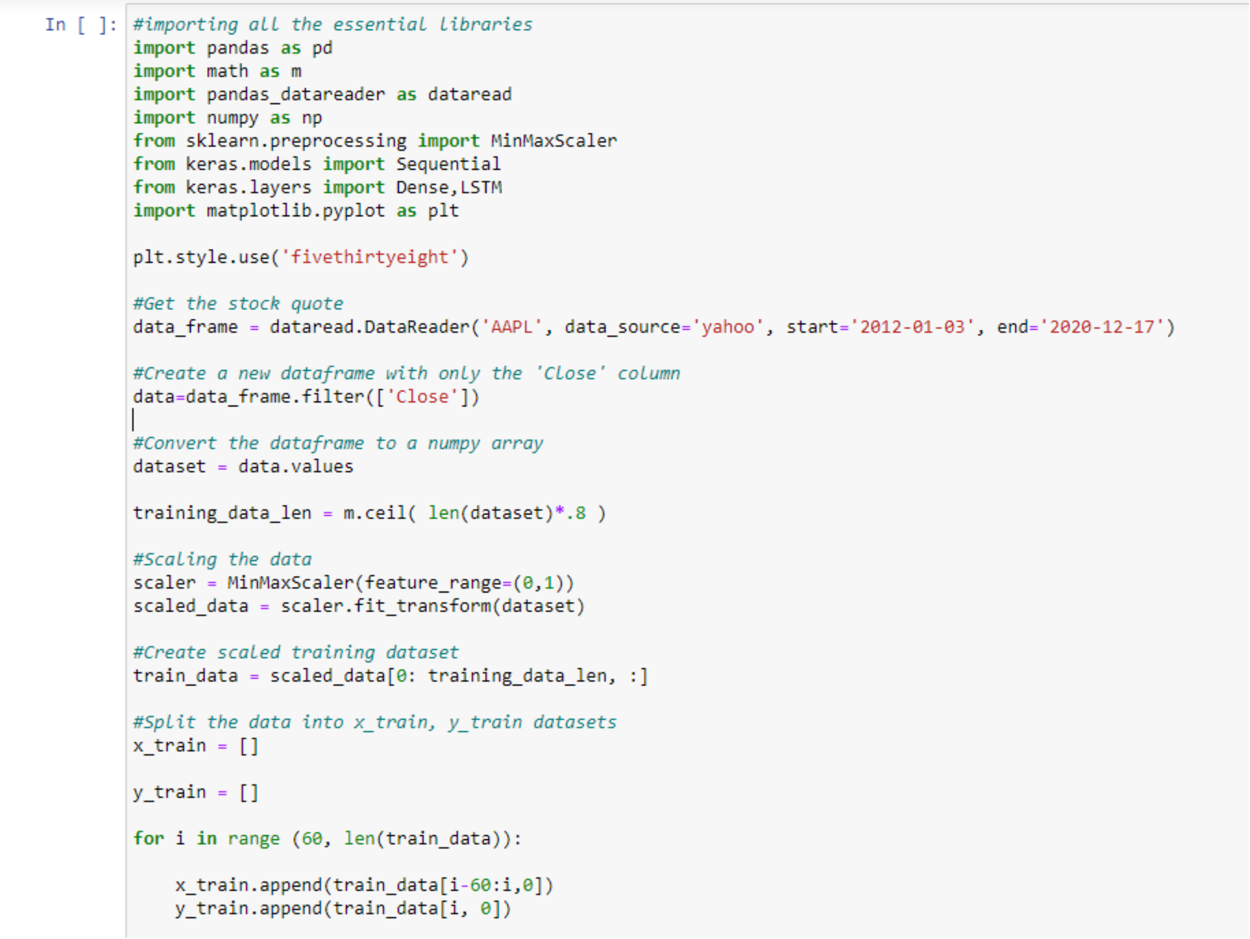
```
Root Mean Squared Error(test) :2.4012948618243395
prediction accuracy :0.9911508789498201
```

Link for YouTube Video Demonstrating the software:

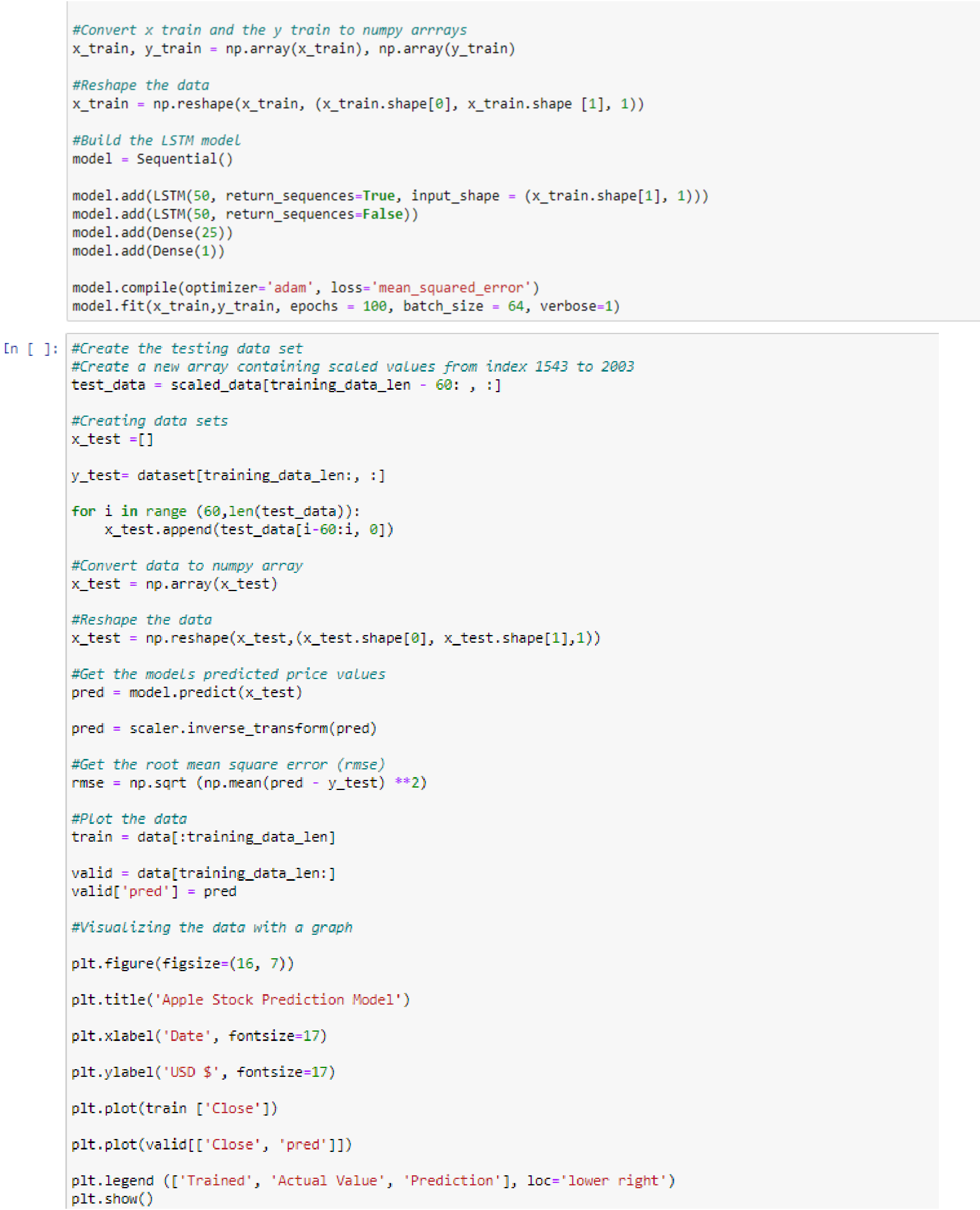
https://youtu.be/7e34Lk86Xo8

Python Code:

Notebook file and python file attached in zip file

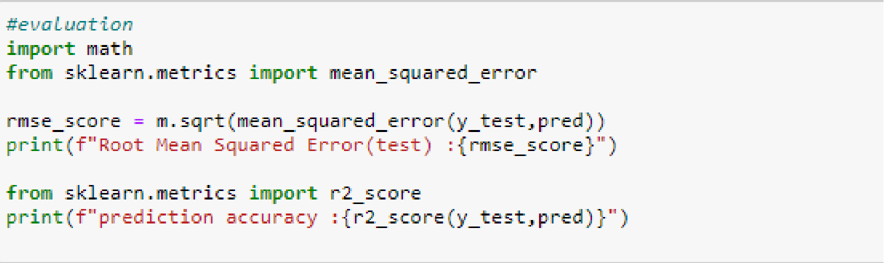


```
In [ ]: #importing all the essential libraries
        import pandas as pd
        import math as m
        import pandas_datareader as dataread
        import numpy as np
        from sklearn.preprocessing import MinMaxScaler
        from keras.models import Sequential
        from keras.layers import Dense,LSTM
        import matplotlib.pyplot as plt

        plt.style.use('fivethirtyeight')

        #Get the stock quote
        data_frame = dataread.DataReader('AAPL', data_source='yahoo', start='2012-01-03', end='2020-12-17')

        #Create a new dataframe with only the 'Close' column
        data=data_frame.filter(['Close'])

        #Convert the dataframe to a numpy array
        dataset = data.values

        training_data_len = m.ceil( len(dataset)*.8 )

        #Scaling the data
        scaler = MinMaxScaler(feature_range=(0,1))
        scaled_data = scaler.fit_transform(dataset)

        #Create scaled training dataset
        train_data = scaled_data[0: training_data_len, :]

        #Split the data into x_train, y_train datasets
        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])
```

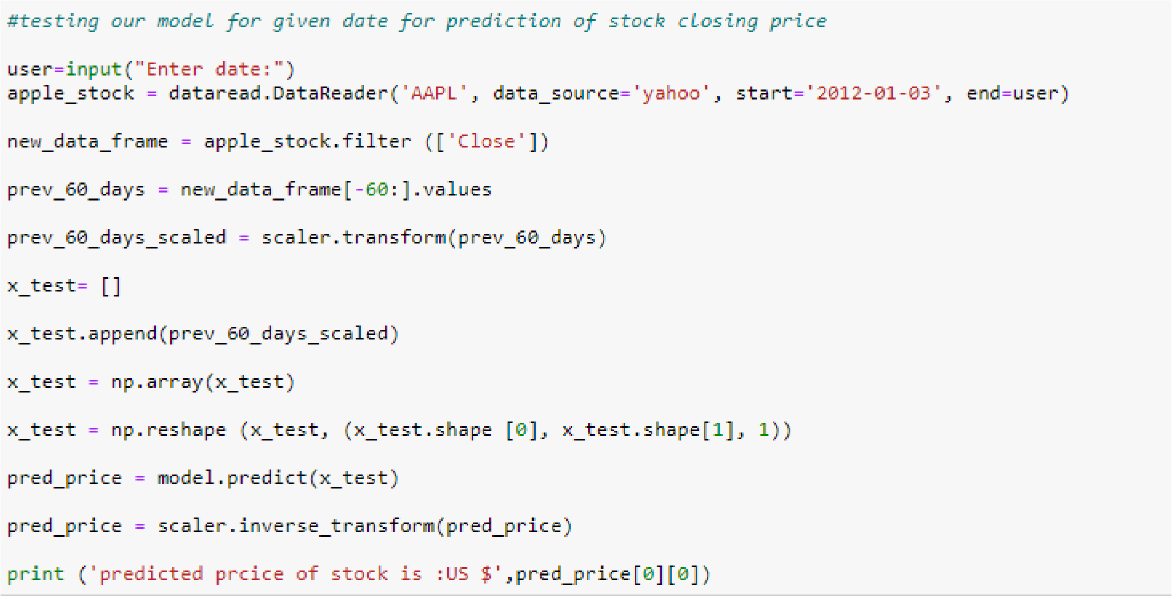


```
#Convert x train and the y train to numpy arrrays
x_train, y_train = np.array(x_train), np.array(y_train)

#Reshape the data
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape [1], 1))

#Build the LSTM model
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))

model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(x_train,y_train, epochs = 100, batch_size = 64, verbose=1)
```

```
In [ ]: #Create the testing data set
        #Create a new array containing scaled values from index 1543 to 2003
        test_data = scaled_data[training_data_len - 60: , :]

        #Creating data sets
        x_test =[]

        y_test= dataset[training_data_len:, :]

        for i in range (60,len(test_data)):
            x_test.append(test_data[i-60:i, 0])

        #Convert data to numpy array
        x_test = np.array(x_test)

        #Reshape the data
        x_test = np.reshape(x_test,(x_test.shape[0], x_test.shape[1],1))

        #Get the models predicted price values
        pred = model.predict(x_test)

        pred = scaler.inverse_transform(pred)

        #Get the root mean square error (rmse)
        rmse = np.sqrt (np.mean(pred - y_test) **2)

        #Plot the data
        train = data[:training_data_len]

        valid = data[training_data_len:]
        valid['pred'] = pred

        #Visualizing the data with a graph

        plt.figure(figsize=(16, 7))

        plt.title('Apple Stock Prediction Model')

        plt.xlabel('Date', fontsize=17)

        plt.ylabel('USD $', fontsize=17)

        plt.plot(train ['Close'])

        plt.plot(valid[['Close', 'pred']])

        plt.legend (['Trained', 'Actual Value', 'Prediction'], loc='lower right')
        plt.show()
```



```
#evaluation
import math
from sklearn.metrics import mean_squared_error

rmse_score = m.sqrt(mean_squared_error(y_test,pred))
print(f"Root Mean Squared Error(test) :{rmse_score}")

from sklearn.metrics import r2_score
print(f"prediction accuracy :{r2_score(y_test,pred)}")
```

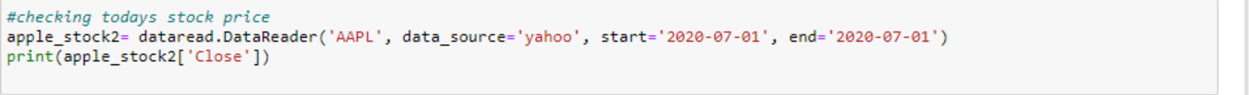


```
#testing our model for given date for prediction of stock closing price

user=input("Enter date:")
apple_stock = dataread.DataReader('AAPL', data_source='yahoo', start='2012-01-03', end=user)

new_data_frame = apple_stock.filter (['Close'])

prev_60_days = new_data_frame[-60:].values

prev_60_days_scaled = scaler.transform(prev_60_days)

x_test= []

x_test.append(prev_60_days_scaled)

x_test = np.array(x_test)

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

pred_price = model.predict(x_test)

pred_price = scaler.inverse_transform(pred_price)

print ('predicted prcice of stock is :US $',pred_price[0][0])
```

Conclusion

Our model performed good at predicting the Apple Stock price using a Stacked LSTM model. This entire notebook can be reused in any stock price prediction.

THE END 😋



```
#checking todays stock price
apple_stock2= dataread.DataReader('AAPL', data_source='yahoo', start='2020-07-01', end='2020-07-01')
print(apple_stock2['Close'])
```