PROFFESIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfilment of the requirements for the award

of Bachelor of Technology Degree in

Information Technology

By

SMRITHII.R

(Reg no:38120079)

DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600119

JANUARY 2021

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.
Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344
www.sathyabama.ac.in

ISO 9001:2015

SCHOOL OF COMPUTING

BONAFIDE CERITFICATE

This is to certify that this project Report is the Bonafide work of who carried out the project entitled "STOCK PREDICTION USING PYTHON" under our supervision from August 2020 to January 2021.

Internal Guide

Dr.P.Jeyanthi M.E., Ph.D.,

Head of the Department

Dr.R.Subhashini M.E., Ph.D.,

Submitted for viva voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I hereby declare that the project report entitled **"STOCK PREDICTION USING PYTHON"** done by me under the guidance of Dr.P. Jeyanthi M.E., Ph.D., at Sathyabama Institute of Science and Technology (Deemed to be university), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -600119 is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:21/01/2021

Place: Chennai SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

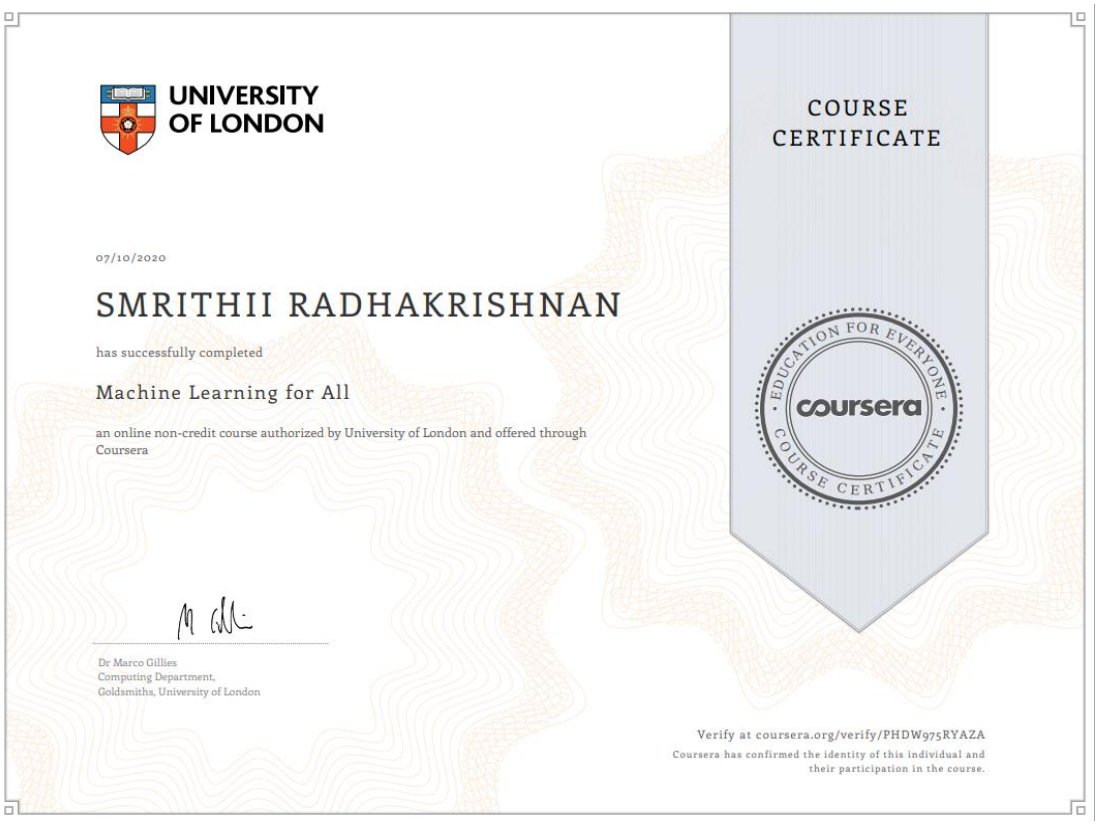
I am pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing** and **Dr. R. SUBHASHINI M.E., Ph.D., Head of the Department, Department of Information Technology** for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. P. JEYANTHI M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE


UNIVERSITY OF LONDON
COURSE CERTIFICATE
07/10/2020
SMRITHII RADHAKRISHNAN
has successfully completed
Machine Learning for All
an online non-credit course authorized by University of London and offered through Coursera
EDUCATION FOR EVERYONE
coursera
COURSE CERTIFICATE
Dr Marco Gillies
Computing Department,
Goldsmiths, University of London
Verify at coursera.org/verify/PHDW975RYAZA
Coursera has confirmed the identity of this individual and their participation in the course.

ABSTRACT

STOCK PREDICTION USING PYTHON

In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python. we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. A machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBRIEVATIONS

ABBRIEVATIONS	EXPANSION
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
LSTM	LONG SHORT TERM MEMORY

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Basically, quantitative traders with a lot of money from stock markets buy stocks derivatives and equities at a cheap price and later on selling them at high price. The trend in a stock market prediction is not a new thing and yet this issue is kept being discussed by various organizations. There are two types to analyze stocks which investors perform before investing in a stock, first is the fundamental analysis, in this analysis investors look at the intrinsic value of stocks, and performance of the industry, economy, political climate etc. to decide that whether to invest or not. On the other hand, the technical analysis it is an evolution of stocks by the means of studying the statistics generated by market activity, such as past prices and volumes. In the recent years, increasing prominence of machine learning in various industries have enlightened many traders to apply machine learning techniques to the field, and some of them have produced quite promising results. Develop a financial data predictor program in which there will be a dataset storing all historical stock prices and data will be treated as training sets for the program. The main purpose of the prediction is to reduce uncertainty associated to investment decision making. Stock Market follows the random walk, which implies that the best prediction you can have about tomorrow's value is today's value. Indisputably, the forecasting stock indices is very difficult because of the market volatility that needs accurate forecast model. The stock market indices are highly fluctuating and it effects the investor's belief. Stock prices are considered to be a very dynamic and susceptible to quick changes because of underlying nature of the financial domain and in part because of the mix of a known parameters (Previous day's closing price, P/E ratio etc.) and the unknown factors (like Election Results, Rumours etc.). There has been numerous attempts to predict stock price with Machine Learning.

1.3 OUTLINE OF THE PROJECT

This project aims to bring a stock prediction which gives accurate values, from not making the traders or stockists get deceived and which gives the values in a faster manner without too much time in delay.

CHAPTER 2

AIM AND SCOPE

2.1 AIM:

The main objective of this project is to bring out the most accurate stock value using python with the help of machine learning

2.2 PROBLEM STATEMENT:

It is hard for the traders or Stockists to predict the value in a manual method. Hence, a Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**.** Machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem.

2.3 SCOPE:

This project will investigate how machine learning techniques can be used to find the accuracy of stock price predictions. This aims to develop a financial data predictor program in which there will be a dataset storing all historical stock prices and data will be treated as training sets for the program. The main purpose of the prediction is to reduce uncertainty associated to investment decision making. Stock Market follows the random walk, which implies that the best prediction you can have about tomorrow's value is today's value. Indisputably, the forecasting stock indices is very difficult because of the market volatility that needs accurate forecast mode.

CHAPTER 3
SYSTEM ANALYSIS

3.1 GENERAL:

3.1.1 MACHINE LEARNING:

Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own.

3.1.2 PYTHON:

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

3.1.3 IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

IDE stands for Integrated Development Environment. It’s a coding tool which allows you to write, test, and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools. And even though the IDE is a strictly defined concept, it’s starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs. The IDE used in this project is Anaconda Navigator.

3.1.4 USER INTERFACE

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse

and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

3.2 SOFTWARES USED

3.2.1 CODE EDITORS USED

ANACONDA

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is an integrated development environment developed by data scientists and is best suitable for Machine learning and Data science.

THE JUPYTER NOTEBOOK

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning

3.2.2 PROGRAMMING LANGUAGES USED

PYTHON

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

HTML:

Html is a markup language used for designing webpages. In our case, I have used for designing the table for dataset.

3.2.3 LIBRARIES AND FRAMEWORKS USED

PANDAS

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

KERAS

Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers, Francois Chollet. It is made user-friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

MATPLOTLIB

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. It is better to represent the data through the graph where we can analyse the data more efficiently and make the specific decision according to data analysis.

NUMPY:

NumPy stands for numeric python which is a python package for the computation and processing of the multidimensional and single dimensional array elements.

3.3 ALGORITHM

CLUSTERING ALGORITHM:

Clustering is a Machine Learning technique that involves the grouping of data points. Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group. In theory, data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields.

ASSOCIATION RULE LEARNING ALGORITHM:

Association rule learning is a rule-based machine learning method or algorithm for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.[1]

REINFORCEMENT LEARNING:

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

3.4 PROJECT DESCRIPTION

The project aims at displaying the correct or accurate values of the stock without giving any approximate value and also by ensuring the stockists or traders from not getting deceived. Therefore, the values would be given in the dataset. The main idea of this project is to predict the prices of the stock before describing it on the graph. The table of the dataset is displayed using html. The project focuses on the use of Regression and LSTM based Machine learning to predict stock values. Factors considered are open, close, low, high and volume.

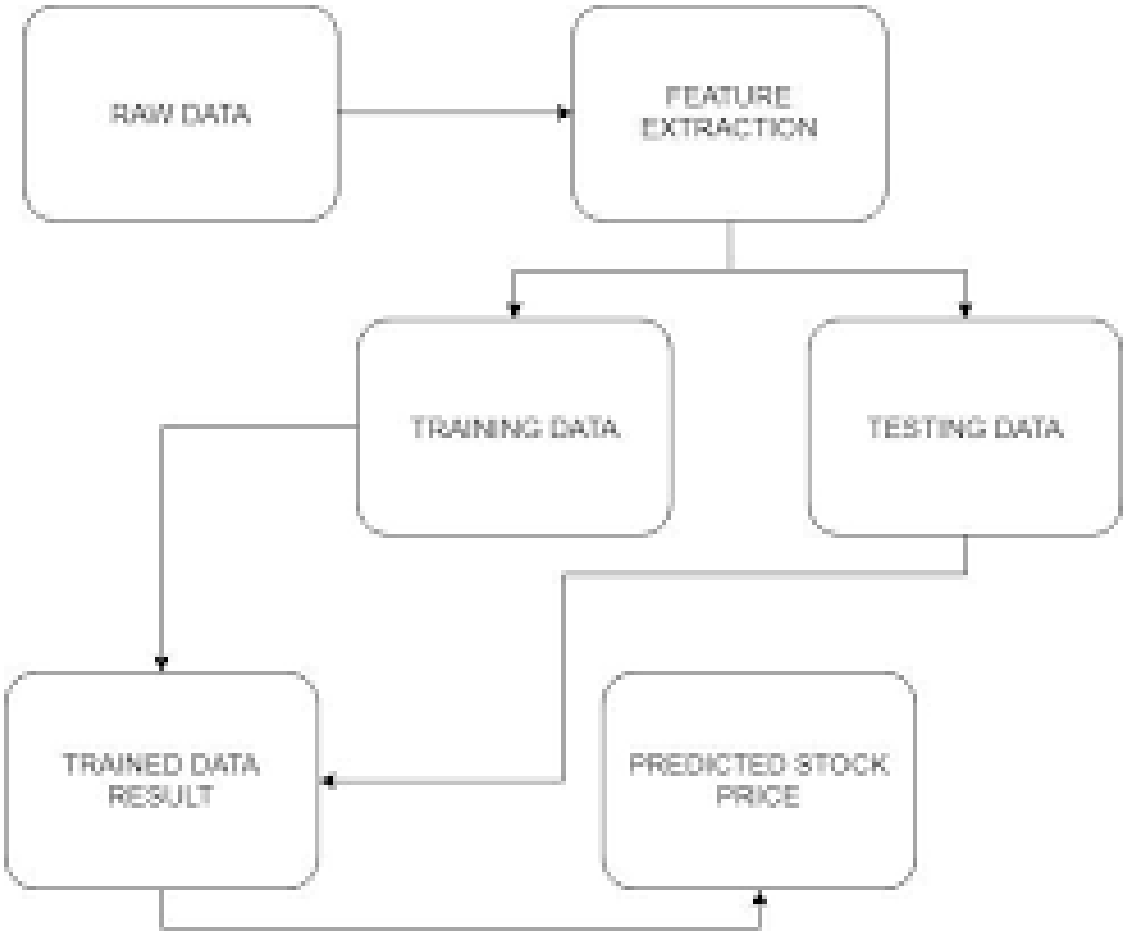


FIG 3.1 DATA FLOW

CHAPTER 4

RESULTS AND DISCUSSION

4.1 SYSTEM REQUIREMENTS

- PROCESSOR: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
- RAM: 8GB
- SYSTEM TYPE: 64 bit Operating System
- STORAGE:500 GB

4.2 RESULT

As a result, using the clustering and asssociative algorithms the datas are being displayed on the dataset. Thus, this shows the fluctuations on the graph, whether they are high or low. Based on the predictions by the result will be displayed. This is machine leearning’s technique of displaying the output with python.

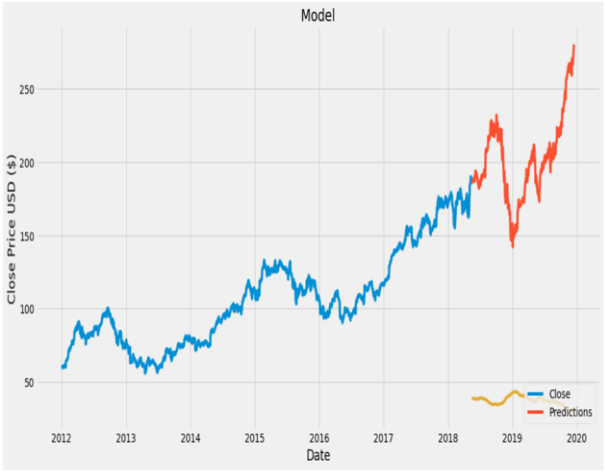


FIG 4.1GRAPH

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This project is undertaken using machine learning and evaluates the performance by using clustering algorithm, regression method and associative algorithms and also using LSTM method. In our proposed model, all the methods are involved. The implementation of above system would help in better prediction of the stock prices or values, than assumptions or manual methods.

5.2 FUTURE WORK

This project describes prediction of stock using python and machine learning. In future the traders or stockists can determine the values using the computerised method unlike the normal method.

5.3 REFERENCES

[1] Krishna Kumar Singh, Dr, Priti Dimri, Madhu Rawat:" Dept. of Computer Science, GBPEC, Ghurdauri, "Green Database Model for Stock Market: a Case Study of Indian Stock Market".

[2] Ashish Sharma, Dinesh Bhuriya, Upendra Singh: Women's Polytechnic, Indore " Survey of Stock Market Prediction Using Machine Learning Approach", International Conference on Electronics, Communication and Aerospace Technology ICECA 2017.

[3] Carol Hargreaves, Yi Hao: " Does the use of Technical & Fundamental Analysis improve Stock Choice? : A Data Mining Approach applied to the Australian Stock Market"

[4] K. jae Kim, "Financial time series forecasting using support vector machines," Neurocomputing, vol. 55, 2003.

APPENDIX

A.SCREENSHOTS

Getting the stock quote

Date	High	Low	Open	Close	Volume	Adj Close
2012-01-03	58.928570	58.428570	58.485714	58.747143	75555200.0	50.857235
2012-01-04	59.240002	58.468571	58.571430	59.062859	65005500.0	51.130558
2012-01-05	59.792858	58.952858	59.278572	59.718571	67817400.0	51.698215
2012-01-06	60.392857	59.888573	59.967144	60.342857	79573200.0	52.238651
2012-01-09	61.107143	60.192856	60.785713	60.247143	98506100.0	52.155792
...	...	...	...	...	...	...
2019-12-11	271.100006	268.500000	268.809998	270.769989	19689200.0	269.399658
2019-12-12	272.559998	267.320007	267.779999	271.459991	34327600.0	270.086151
2019-12-13	275.299988	270.929993	271.459991	275.149994	33396900.0	273.757477
2019-12-16	280.790009	276.980011	277.000000	279.859985	32046500.0	278.443604
2019-12-17	281.769989	278.799988	279.570007	280.410004	28539600.0	278.990875

2003 rows × 6 columns

Visualizing the closing price history

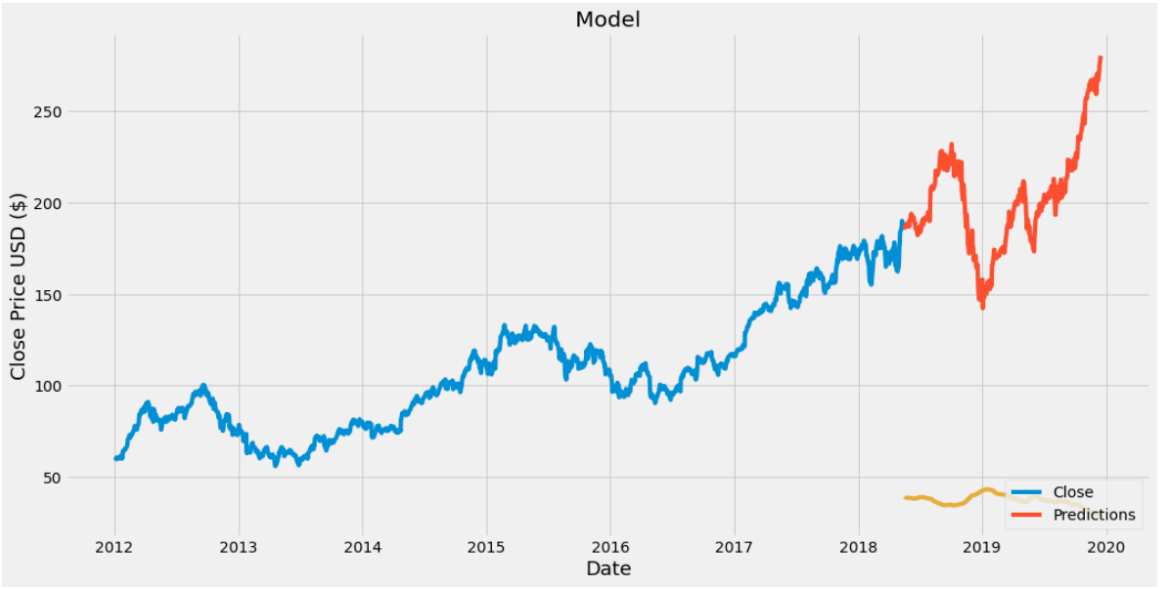
Scaling the data

```
array([[ 58.74714279],
       [ 59.06285858],
       [ 59.71857071],
       ...,
       [275.1499939 ],
       [279.85998535],
       [280.41000366]])
```

Creating the scaled training dataset

```
[array([0.01316509, 0.01457064, 0.01748985, 0.02026915, 0.01984303,
       0.02080338, 0.02036454, 0.01962679, 0.01862193, 0.02173194,
       0.02453668, 0.02367172, 0.01893356, 0.02345548, 0.01900352,
       0.03569838, 0.03440731, 0.0360927 , 0.03973694, 0.04194384,
       0.0417594 , 0.0410789 , 0.04397903, 0.04670744, 0.04979839,
       0.05479095, 0.0652785 , 0.06543749, 0.07127594, 0.07563885,
       0.06814046, 0.07102789, 0.07097066, 0.07906688, 0.07791571,
       0.08004628, 0.08387497, 0.08600558, 0.09214292, 0.09661397,
       0.09790501, 0.09835659, 0.09071194, 0.08886753, 0.08914103,
       0.09632775, 0.09835024, 0.10269405, 0.11293358, 0.12659476,
       0.12403805, 0.1240444 , 0.13392141, 0.13701233, 0.13481179,
       0.13280207, 0.13070964, 0.13766105, 0.14243103, 0.14442805])]
[0.13949272033425864]
```

Visualize the data



Showing the prediction

Date	Close	Predictions
2018-05-17	186.990005	38.912300
2018-05-18	186.309998	38.769295
2018-05-21	187.630005	38.662605
2018-05-22	187.160004	38.579830
2018-05-23	188.360001	38.519722
...	...	...
2019-12-11	270.769989	30.044834
2019-12-12	271.459991	30.000523
2019-12-13	275.149994	29.941971
2019-12-16	279.859985	29.859503
2019-12-17	280.410004	29.741396

400 rows × 2 columns

B. SOURCE CODE

#import libraries

import math

```
import pandas_datareader as web
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras import Dropout
import matplotlib as plt
plt.style.use(“fivethirtyeight”)
# Get the stock quote
df = web. DataReader('AAPL' start='2012-01-01', end='2019-12-17')
#Show the data
Df
#Visualize the closing price history
plt.figure(figsize=(16,8))
plt.title('Close Price History')
plt.plot(df['Close'])
plt.xlabel('Data', fontsize=18)
plt.ylabel('Close Price USD ($)', fontsize=18)
plt.show()
#Create a new dataframe width only the 'Close','Date' columns
New_data = df.filter(['Close')
#Conver the dataframe to a numpy array
dataset = new_data.values
#get the number of rows to train the model on
training_data_len = math.ceil(len(dataset) * .8)
training_data_len
1603
#Scale the data
scaler = MinMaxScaler(feature)
final_dataset=new_data.values
final_dataset
#Create the training dataset
#Create the scaled training dataset
train_data = final_dataset[0:training_data_len , :]
```

```
valid_data = final_dataset[training_data_len:,:]
#Split the data into x_train and y_train datasets
scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(final_dataset)
x_train,y_train = [],[]
for i in range(60, len(train_data)):
  x_train.append(scaled_data[i-60:i,0])
  y_train.append(scaled_data[i, 0])
  if i<= 60:
    print(x_train)
    print(y_train)
    print()
#Build the LSTM model
lstm_model=Sequential()
lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train.sh
ape[1],1)))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(1))
inputs_data=new_data[len(new_data)-len(valid_data)-60:].values
inputs_data=inputs_data.reshape(-1,1)
inputs_data=scaler.transform(inputs_data)
X_test=[]
for i in range(60,inputs_data.shape[0]):
X_test.append(inputs_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))
predicted_closing_price=lstm_model.predict(X_test)
predicted_closing_price=scaler.inverse_transform(predicted_closing_price)
train_data=new_data[:training_data_len]
valid_data=new_data[training_data_len:]
valid_data['Predictions']=predicted_closing_price
#visualize the data
plt.figure(figsize=(16,8))
plt.title('Model')
```

```
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close Price USD ($)' , fontsize=18)
plt.plot(train_data["Close"])
plt.plot(valid_data[['Close',"Predictions"]])
plt.legend(['Close','Predictions'],loc='lower right')
#show the valid and prediction
valid
```

FOR DISPLAYING DATA IN DATASET

```
"cells": [
 {
  "cell_type": "code",
  "execution_count": 25,
  "metadata": {
   "colab": {},
   "colab_type": "code",
   "id": "9_GyN4Z7c0FR"
  },
  "outputs": [],
  "source": [
   "#import libraries\n",
   "import math\n",
   "import pandas_datareader as web\n",
   "import numpy as np\n",
   "import pandas as pd\n",
   "from sklearn.preprocessing import MinMaxScaler\n",
   "from keras.models import Sequential\n",
   "from keras.layers import Dense, LSTM,Dropout\n",
   "import matplotlib.pyplot as plt\n",
   "plt.style.use('fivethirtyeight')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 26,
```

```
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/",
  "height": 431
 },
 "colab_type": "code",
 "id": "s9jtdFoNeAIi",
 "outputId": "3ad68ef8-5abf-4e99-fc9f-f043ffbb1019"
},
"outputs": [
 {
  "data": {
   "text/html": [<html>
    "<div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
    "    <tr style=\"text-align: right;\">\n",
    "      <th></th>\n",
    "      <th>High</th>\n",
    "      <th>Low</th>\n",
    "      <th>Open</th>\n",
```

```
"      <th>Close</th>\n",
"      <th>Volume</th>\n",
"      <th>Adj Close</th>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>Date</th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>2012-01-03</th>\n",
"      <td>58.928570</td>\n",
"      <td>58.428570</td>\n",
"      <td>58.485714</td>\n",
"      <td>58.747143</td>\n",
"      <td>75555200.0</td>\n",
"      <td>50.857235</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2012-01-04</th>\n",
"      <td>59.240002</td>\n",
"      <td>58.468571</td>\n",
"      <td>58.571430</td>\n",
"      <td>59.062859</td>\n",
"      <td>65005500.0</td>\n",
"      <td>51.130558</td>\n",
"    </tr>\n",
"    <tr>\n",
```

```
"      <th>2012-01-05</th>\n",
"      <td>59.792858</td>\n",
"      <td>58.952858</td>\n",
"      <td>59.278572</td>\n",
"      <td>59.718571</td>\n",
"      <td>67817400.0</td>\n",
"      <td>51.698215</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2012-01-06</th>\n",
"      <td>60.392857</td>\n",
"      <td>59.888573</td>\n",
"      <td>59.967144</td>\n",
"      <td>60.342857</td>\n",
"      <td>79573200.0</td>\n",
"      <td>52.238651</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2012-01-09</th>\n",
"      <td>61.107143</td>\n",
"      <td>60.192856</td>\n",
"      <td>60.785713</td>\n",
"      <td>60.247143</td>\n",
"      <td>98506100.0</td>\n",
"      <td>52.155792</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>...</th>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
```

```
"    </tr>\n",
"    <tr>\n",
"      <th>2019-12-11</th>\n",
"      <td>271.100006</td>\n",
"      <td>268.500000</td>\n",
"      <td>268.809998</td>\n",
"      <td>270.769989</td>\n",
"      <td>19689200.0</td>\n",
"      <td>269.399658</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2019-12-12</th>\n",
"      <td>272.559998</td>\n",
"      <td>267.320007</td>\n",
"      <td>267.779999</td>\n",
"      <td>271.459991</td>\n",
"      <td>34327600.0</td>\n",
"      <td>270.086151</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2019-12-13</th>\n",
"      <td>275.299988</td>\n",
"      <td>270.929993</td>\n",
"      <td>271.459991</td>\n",
"      <td>275.149994</td>\n",
"      <td>33396900.0</td>\n",
"      <td>273.757477</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2019-12-16</th>\n",
"      <td>280.790009</td>\n",
"      <td>276.980011</td>\n",
"      <td>277.000000</td>\n",
"      <td>279.859985</td>\n",
```

```
"      <td>32046500.0</td>\n",
"      <td>278.443604</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2019-12-17</th>\n",
"      <td>281.769989</td>\n",
"      <td>278.799988</td>\n",
"      <td>279.570007</td>\n",
"      <td>280.410004</td>\n",
"      <td>28539600.0</td>\n",
"      <td>278.990875</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"<p>2003 rows × 6 columns</p>\n",
"</div>"
],
</table>

"                 High        Low  ...      Volume   Adj Close\n",
"Date                              ...                  \n",
"2012-01-03   58.928570   58.428570  ...  75555200.0   50.857235\n",
"2012-01-04   59.240002   58.468571  ...  65005500.0   51.130558\n",
"2012-01-05   59.792858   58.952858  ...  67817400.0   51.698215\n",
"2012-01-06   60.392857   59.888573  ...  79573200.0   52.238651\n",
"2012-01-09   61.107143   60.192856  ...  98506100.0   52.155792\n",
"...                ...         ...  ...         ...         ...\n",
"2019-12-11  271.100006  268.500000  ...  19689200.0  269.399658\n",
"2019-12-12  272.559998  267.320007  ...  34327600.0  270.086151\n",
"2019-12-13  275.299988  270.929993  ...  33396900.0  273.757477\n",
"2019-12-16  280.790009  276.980011  ...  32046500.0  278.443604\n",
"2019-12-17  281.769989  278.799988  ...  28539600.0  278.990875\n",
"\n",
"[2003 rows x 6 columns]"
```

```
    ]
   },
   "execution_count": 26,
   "metadata": {
    "tags": []
   },
   "output_type": "execute_result"
  }
 ],
 "source": [
  "# Get the stock quote\n",
  "df = web.DataReader('AAPL', data_source='yahoo', start='2012-01-01', end='2019-12-17')\n",
  "#Show the data\n",
  "df"
 ]
},
{
 "cell_type": "code",
 "execution_count": 27,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 34
  },
  "colab_type": "code",
  "id": "DFqq5OlceqSJ",
  "outputId": "f395481a-e5c8-466d-e989-43169fb8c87d"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(2003, 6)"
```

```
    ]
   },
"execution_count": 27,
   "metadata": {
    "tags": []
   },
   "output_type": "execute_result"
  }
 ],

  "df.shape"
 ]
},
{
 "cell_type": "code",
 "execution_count": 28,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 558
  },
  "colab_type": "code",
  "id": "Sio4e55yexKe",
  "outputId": "a14199b4-3122-4e5b-8e4f-47cf23844bd4"
 },
 </html>
```