STOCK PRICE PREDICTION SYSTEM

A

Project Report

Submitted In Partial Fulfillment of the Requirements

for the Award Of

Bachelor of Technology

Under Guidance Of

RAHUL SHARMA

MICROSOFT & HPE CERTIFIED TECHNICAL TRAINER

Project Carried Out

At

ARDENT®

"Drives You to Industry"

Ardent Computes Pvt. Ltd.

(An ISO 9001:2015 Certified)

SDF Building, Module No: 132,

Ground Floor Sector V,

GP Block, Kolkata- 700091

Submitted By-

RAVNISH SINGH

ACKNOWLEDGEMENT

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project work.

Our heartful thanks to **Avick Kr. Dey** for providing us the opportunity to develop the project at **Ardent Computech Pvt. Ltd.**

We would like to show our greatest appreciation to **Technical Trainer at Ardent,** Durgapur. We always feel motivated and encouraged every time by his valuable advice and constant inspiration; without his encouragement and guidance this project would not have materialized. We also want to thank them for sharing their pearls of wisdom with us during this project.

Words are inadequate in offering our thanks to the other trainees**, project assistants** and other members at **Ardent Computes Pvt. Ltd.** for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project

Self Certificate

This is to certify that the dissertation/project proposal entitled **"Stock Price Prediction System"** is done by **Ravnish Singh, Yash Goswami and Sauham Joshi** is an authentic work carried out for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** under the guidance of **our technical trainer**. The matter embodied in this project work has not been submitted earlier for award of any degree to the best of my knowledge and belief.

Name of the Students:

1. **RAVNISH SINGH**
2. **YASH GOSWAMI**
3. **SAUHAM JOSHI**

Certificate by Guide

This is to certify that this project entitled **"Stock Price Prediction System"** submitted in partial fulfillment of the degree of **Bachelor of Technology (B.Tech)** by **Shri Vaishnav Vidyapeeth Vishwavidyalaya** done by **Ravnish Singh, Yash Goswami and Sauham Joshi.**

Is an authentic work carried out under my guidance & best of our knowledge and belief.

Signature of Student

Date:

Signature of the Guide

Date:

TABLE OF CONTENTS

1. ARDENT COMPUTECH PVT.LTD.

Ardent Computech Private Limited is an ISO 9001-2008 certified Software Development Company in India. It has been operating independently since 2003. It was recently merged with ARDENT TECHNOLOGIES.

Ardent Technologies

ARDENT TECHNOLOGIES is a Company successfully providing its services currently in UK, USA, Canada and India. The core line of activity at ARDENT TECHNOLOGIES is to develop customized application software covering the entire responsibility of performing the initial system study, design, development, implementation and training. It also deals with consultancy services and Electronic Security systems. Its primary clientele includes educational institutes, entertainment industries, resorts, theme parks, service industry, telecom operators, media and other business houses working in various capacities.

Ardent Collaborations

ARDENT COLLABORATIONS, the Research Training and Development Department of ARDENT COMPUTECH PVT LTD is a professional training Company offering IT enabled services & industrial trainings for B-Tech, MCA, BCA, MSc and MBA fresher's and experienced developers/programmers in various platforms. Summer Training / Winter Training / Industrial training will be provided for the students of B.TECH, M.TECH, MBA, MCA and BCA only. Deserving candidates may be awarded stipends, scholarships and other benefits, depending on their performance and recommendations of the mentors.

Associations

Ardent is an ISO 9001:2008 company.

It is affiliated to National Council of Vocational Training (NCVT), Directorate General of Employment & Training (DGET), Ministry of Labor & Employment, and Government of India.

2. ABSTRACT

Accurate prediction of stock market returns is a very challenging task due to volatile and non-linear nature of the financial stock markets. We attempt to build a scalable model to perform this analysis. With the introduction of artificial intelligence and increased computational capabilities, programmed methods of prediction have proved to be more effective in predicting stock prices. In this work, **LSTM** (Long Short Term Memory) which is a part of Recurrent Neural Network (**RNN**) has been utilized for predicting the closing price for TESLA company. The financial data: Open, High, Low , Close, Volume and Adjacent Close prices of stock are used for creating new variables which are used as inputs to the model. The models are evaluated using standard strategic indicator: RMSE . The low value of this indicator show that the model is efficient in predicting stock closing price.

3. INTRODUCTION

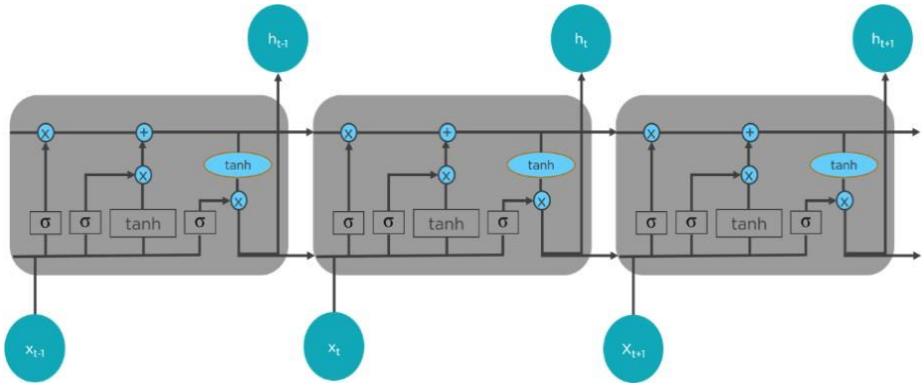
A stock market is a public market where we can buy and sell shares for publicly listed companies. The stocks, also known as equities, represent ownership in the company. The stock exchange is the mediator that allows the buying and selling of shares.

Stock market is characterized as dynamic, unpredictable and non-linear in nature. Predicting stock prices is a challenging task as it depends on various factors including but not limited to political conditions, global economy, company's financial reports and performance etc. Thus, to maximize the profit and minimize the losses, techniques to predict values of the stock in advance by analyzing the trend over the last few years, could prove to be highly useful for making stock market movements. Traditionally, two main approaches have been proposed for predicting the stock price of an organization. Technical analysis method uses historical price of stocks like closing and opening price, volume traded, adjacent close values etc. of the stock for predicting the future price of the stock. The second type of analysis is qualitative, which is performed on the basis of external factors like company profile, market situation, political and economic factors, textual information in the form of financial new articles, social media and even blogs by economic analyst. Now a days, advanced intelligent techniques based on either technical or fundamental analysis are used for predicting stock prices. Particularly, for stock market analysis, the data size is huge and also non-linear. To deal with this variety of data efficient model is needed that can identify the hidden patterns and complex relations in this large data set. Machine learning techniques in this area have proved to improve efficiencies by 60-86 percent as compared to the past methods.

4. ALGORITHMS

For our project, we focused on one main algorithm for stock market prediction:

4.1 Long Short Term Memory (LSTM) : LSTMs are a type of Recurrent Neural Network for learning long-term dependencies. It is commonly used for processing and predicting time-series data.



LSTMs have a chain-like structure. Generally RNNs (Recurrent Neural Networks) have a single neural network layer and LSTMs, on the other hand, have four interacting layers communicating extraordinarily.

LSTMs work in a three-step process :-

- The first step in LSTM is to decide which information is to be omitted from the cell in that particular time step. It is decided with the help of a sigmoid function. It looks at the previous state (ht-1) and the current input xt and computes the function.
- There are two functions in the second layer. The first is the sigmoid function, and the second is the tanh function. The sigmoid function decides which values to let through (0 or 1). The tanh function gives the weightage to the values passed, deciding their level of importance from -1 to 1.
- The third step is to decide what will be the final output. First, we need to run a sigmoid layer which determines what parts of the cell state make it to the output. Then, we must put the cell state through the tanh function to push the values between -1 and 1 and multiply it by the output of the sigmoid gate.

4.2 Convolutional Neural Network (CNN) - A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-

engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

In deep learning, a **convolutional neural network (CNN/ConvNet)** is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

5. PROBLEM STATEMENT

Stock market analysis is divided into two parts – Fundamental Analysis and Technical Analysis.

- Fundamental Analysis involves analyzing the company's future profitability on the basis of its current business environment and financial performance.
- Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market.

But here we will mainly deal with the technical analysis. There are multiple variables in the dataset – date, open, high, low, close, adjacent close and volume.

- The columns Open and Close represent the starting and final price at which the stock is traded on a particular day.
- High, Low and Adjacent close represents the maximum, minimum, and closing price of the share for the day.
- Volume is the number of shares bought or sold in the day.

6. BACKGROUND DATASET

- The dataset includes 10 year data from 06/29/2010 to 03/17/2017.
- The dataset (Tesla's stock prices) used to train the model is collected from Kaggle.
- The data contains information about the stock such as High, Low, Open, Close, Adjacent close and Volume.
- There are no null values present in any of the cells.
- Only the day-wise closing price of the stock has been extracted.

.

7. WHAT IS ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

The word Artificial Intelligence comprises of two words “Artificial” and “Intelligence”. Artificial refers to something which is made by human or non natural thing and Intelligence means ability to understand or think. There is a misconception that Artificial Intelligence is a system, but it is not a system .AI is implemented in the system. There can be so many definition of AI, one definition can be “It is the study of how to train the computers so that computers can do things which at present human can do better.”Therefore It is a intelligence where we want to add all the capabilities to machine that human contain.

8. SYSTEM ANALYSIS

8.1 IDENTIFICATION OF NEED

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studies to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The System is viewed as a whole and the input to the system are identified. The outputs from the organization are traced to the various processes. System analysis is concerned with becoming aware of the problem ,identifying the relevant and Decisional variables, analysis and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem area are identified. The designer now function as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

8.2 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose the organization for the amount of work, effort and time spent on it : Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various area that were considered very carefully during the feasibility study of this project such as Technical, Economic and operational feasibilities.

Technical Feasibility: This project is technically feasible as all it has got to do to extract tweets, is to get the proper credentials from the Developer's console provided by Twitter. After the credentials are obtained, i.e. the Access Token Key, Access Token Secret Key, Consumer Key and Consumer Secret key,Twitter gives us access to its tweets. Hence we get a sufficiently large dataset to conduct sentiment analysis. Also the range of tweets obtained is limited to 300 tweets per page,which ensures that the results do not go out of bounds. Thus, this is technically feasible.

Economic Feasibility : This project work is economically feasible as it does not take into account any additional costs. Whatever data is extracted, it is done without any charges. Twitter provides free use of this data that is non-encrypted and publicly available for analysis purpose.Hence, this work is economically feasible as well.

Operational Feasibility : This is operationally feasible as well. As already mentioned, it takes in 300 tweets per page as that is the limit set by Twitter. Therefore it is operationally feasible as well. The system won't hang when getting the results.

8.3 WORK FLOW

- First we have to import python libraries like - numpy as np for mathematical calculation, pandas as pd for and matplotlib.pylpot as plt for plotting of 2D graphs.
- Next we have to load the training dataset i.e 'Tesla.csv'.
- Then we will use the open price stock column to train the model.
- Next we have to normalize the dataset by importing MinMaxScaler.
- Then we have to create x_train and y_train data structures.
- Next we have to reshape the data.
- After reshaping, we have to build the model by importing important libraries and adding different layers to LSTM.
- Then we need to fit or train the model.
- After that we have to extract the actual stockprices.
- Then we have to prepare the input for the model.
- Finally, we will make predictions of adjusted close date of stock price against the actual closing date.
- And at last we will plot the actual and predicted stock prices.

8.4 HARDWARE AND SOFTWARE REQUIREMENTS

❖ Hardware Requirements

Standard computer with at least i3 processor

Standard computer with 2GB of RAM

Standard computer with 100GB of free space

❖ Software Requirements

python 3.7

Anaconda-3.5

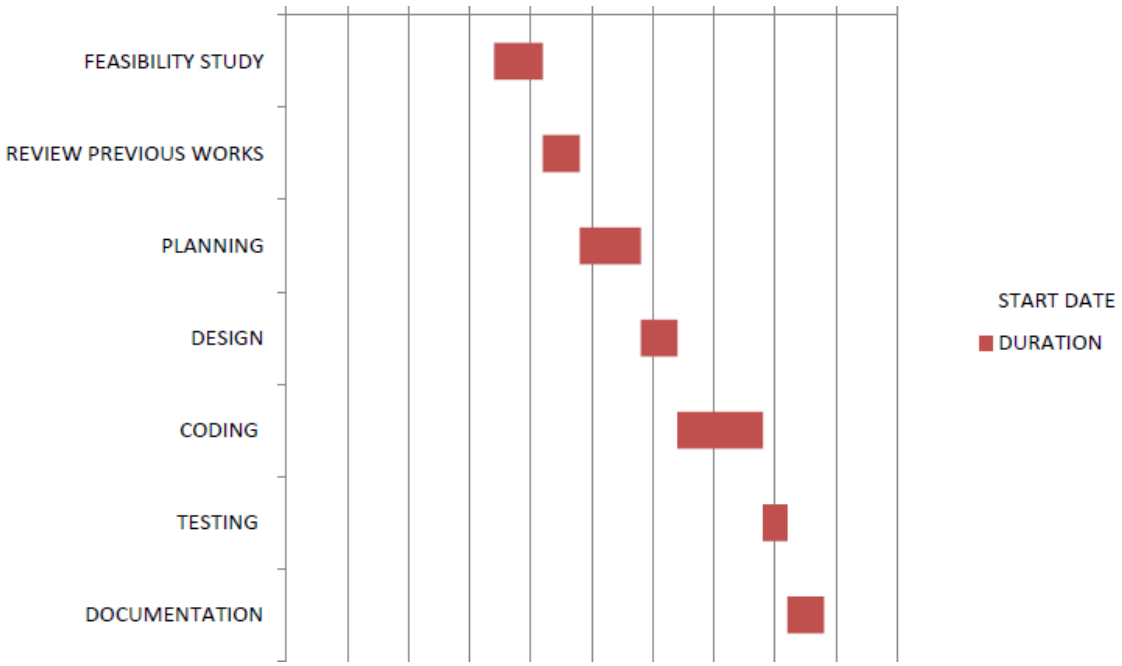
Ms Office

Draw.io

9. SYSTEM DESIGN

9.1 GANTT CHART

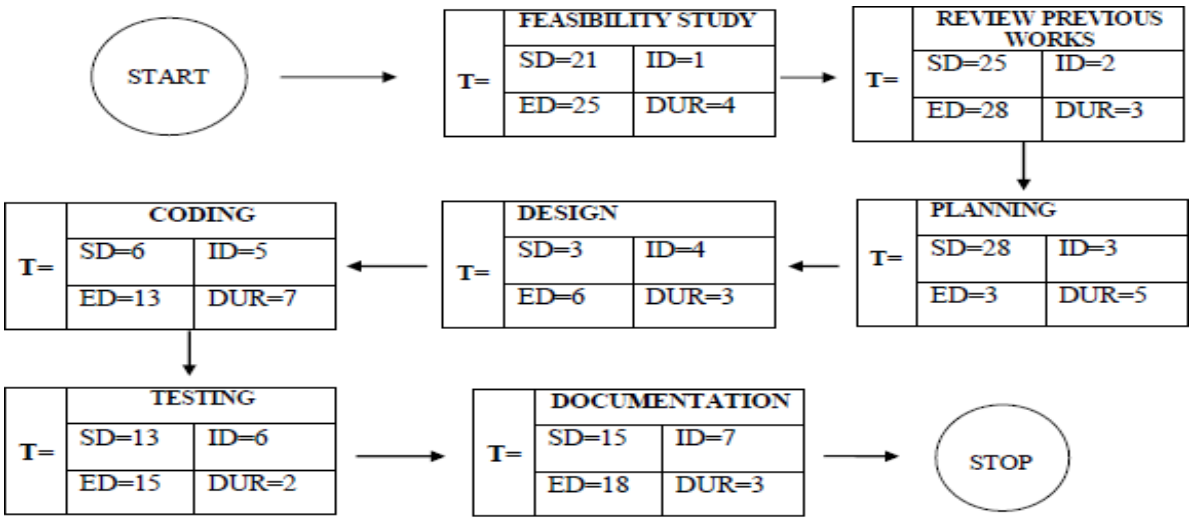
1	TASK	START DATE	END DATE	DURATION
2	FEASIBILITY STUDY	03-Jul-21	06-Jul-21	4
3	REVIEW PREVIOUS WORKS	07-Jul-21	11-Jul-21	5
4	PLANNING	12-Jul-21	16-Jul-21	5
5	DESIGN	17-Jul-21	20-Jul-21	4
6	CODING	21-Jul-21	27-Jul-21	7
7	TESTING	28-Jul-21	30-Jul-21	3
8	DOCUMENTATION	31-Jul-21	03-Aug-21	4



GANTT CHART

9.2 PERT CHART:

1	TASK	START DATE	END DATE	DURATION
2	FEASIBILITY STUDY	03-Jul-21	06-Jul-21	4
3	REVIEW PREVIOUS WORKS	07-Jul-21	11-Jul-21	5
4	PLANNING	12-Jul-21	16-Jul-21	5
5	DESIGN	17-Jul-21	20-Jul-21	4
6	CODING	21-Jul-21	27-Jul-21	7
7	TESTING	28-Jul-21	30-Jul-21	3
8	DOCUMENTATION	31-Jul-21	03-Aug-21	4



PERT CHART DIAGRAM

9.3 SEQUENCE DIAGRAM

Sequence diagrams can be useful reference diagrams for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how tasks are moved between objects or components of a process.
- Plan and understand the detailed functionality of an existing or futurescenario.

9.3.1 Popular Sequence Diagram Uses:

- **Usage Scenario** – A usage scenario is a diagram of how your system could potentially be used. It's a great way to make sure that you have worked through the logic of every usage scenario for the system.
- **Method Logic** - Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to Usage Scenario - A usage scenario is a diagram of how your system could potentially be used. It's a great explore the logic of any function, procedure, or complex process.
- **Service Logic -** If you consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.

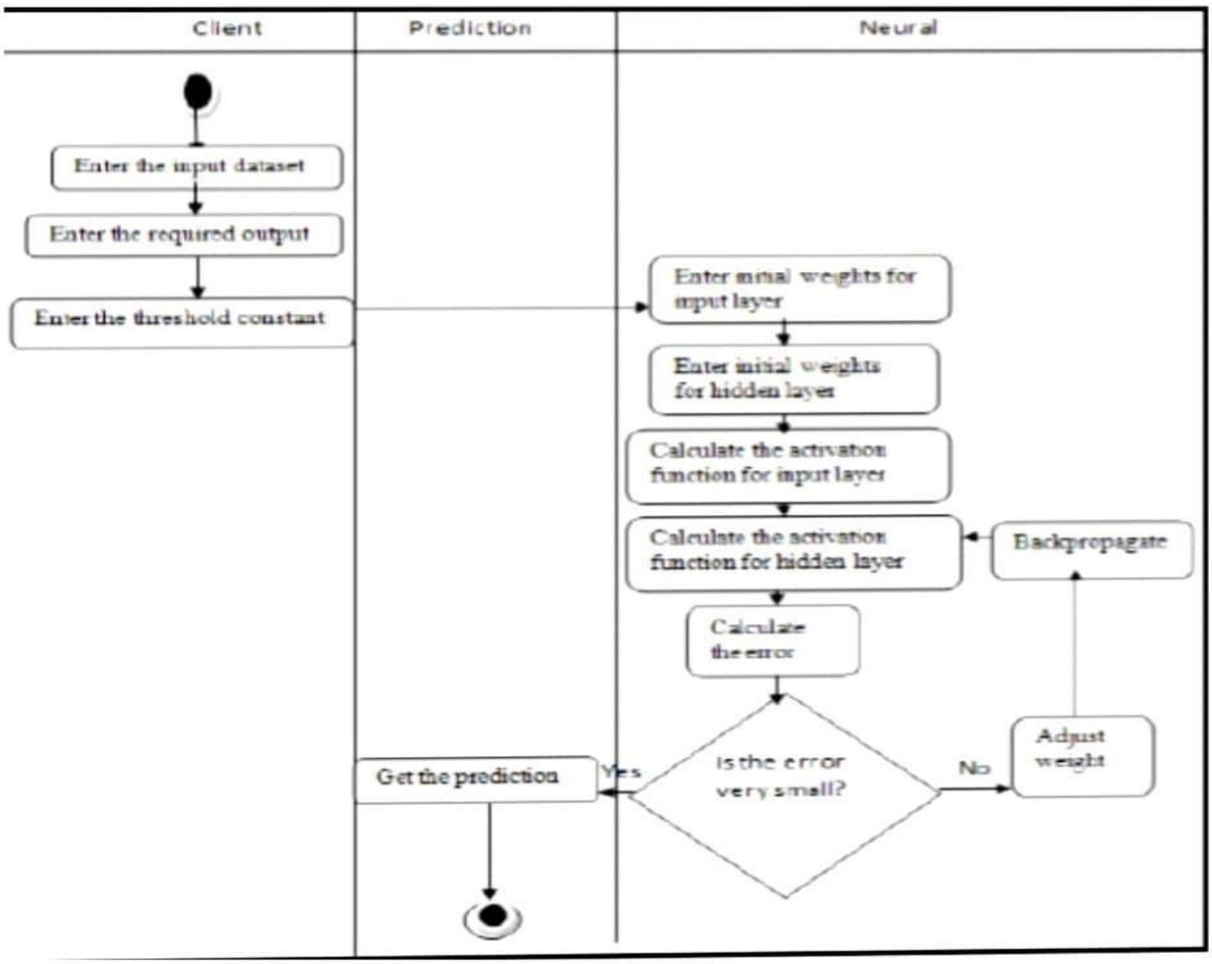


Fig:Sequence Diagram

9.4 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Where to Use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

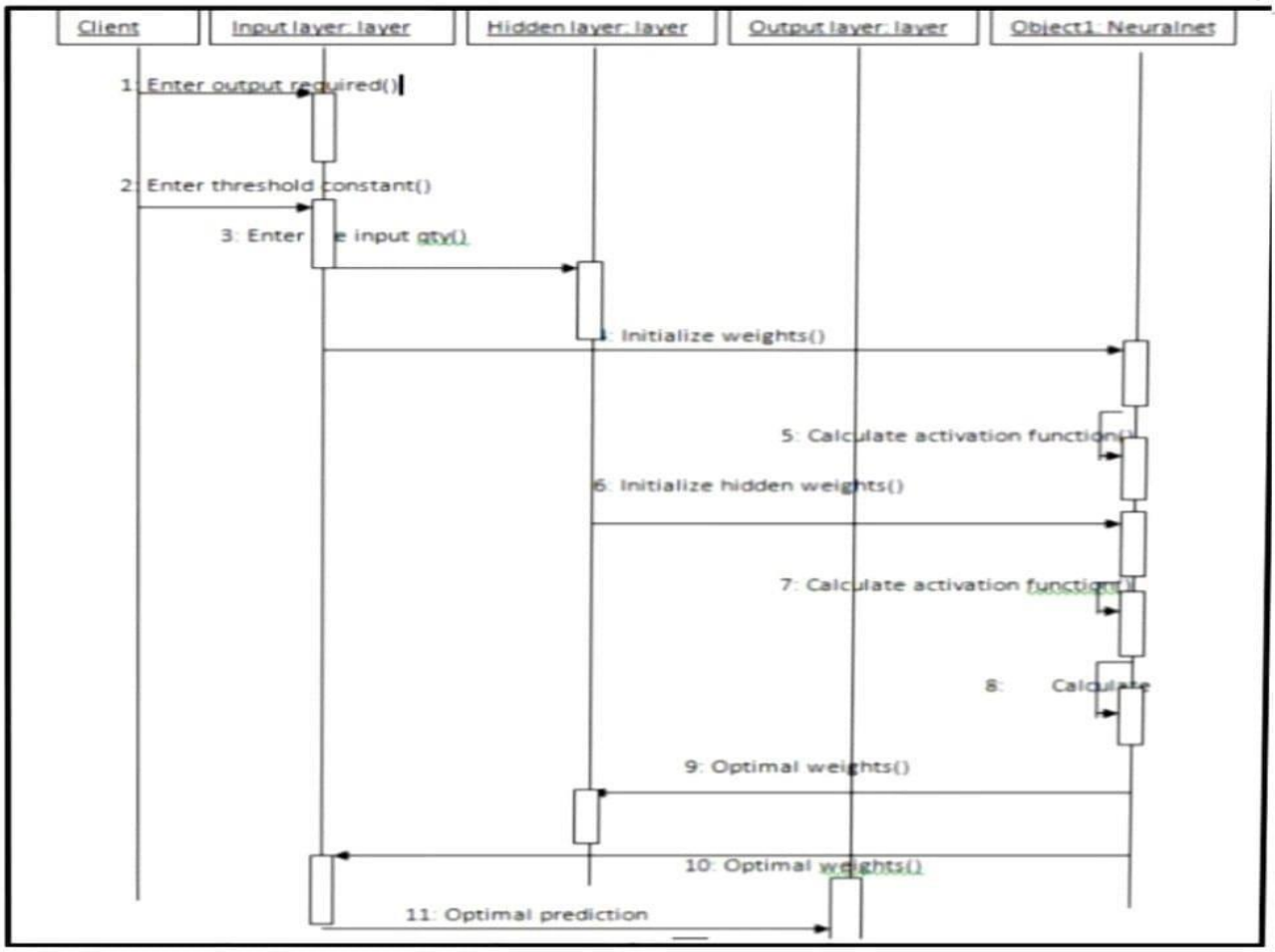


Fig:Activity Diagram

9.5. USE CASE DIAGRAM

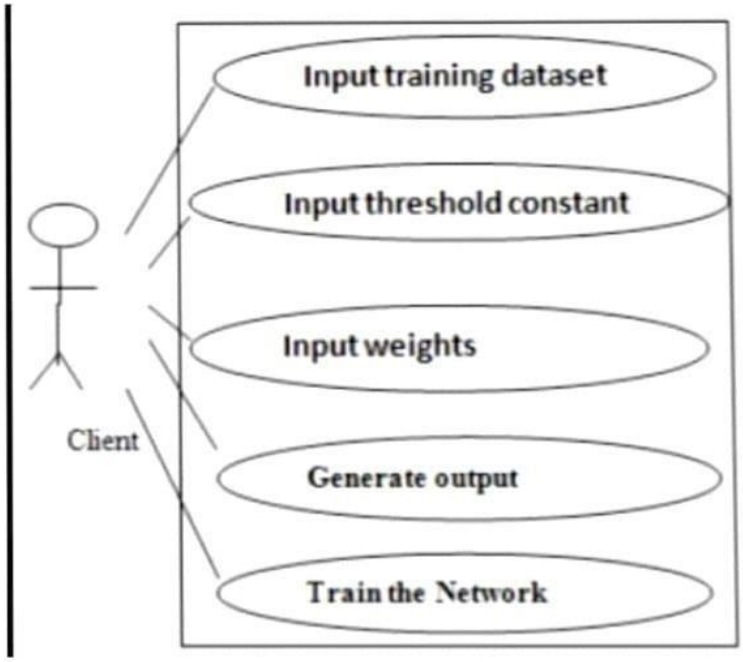


Fig:Use Case Diagram

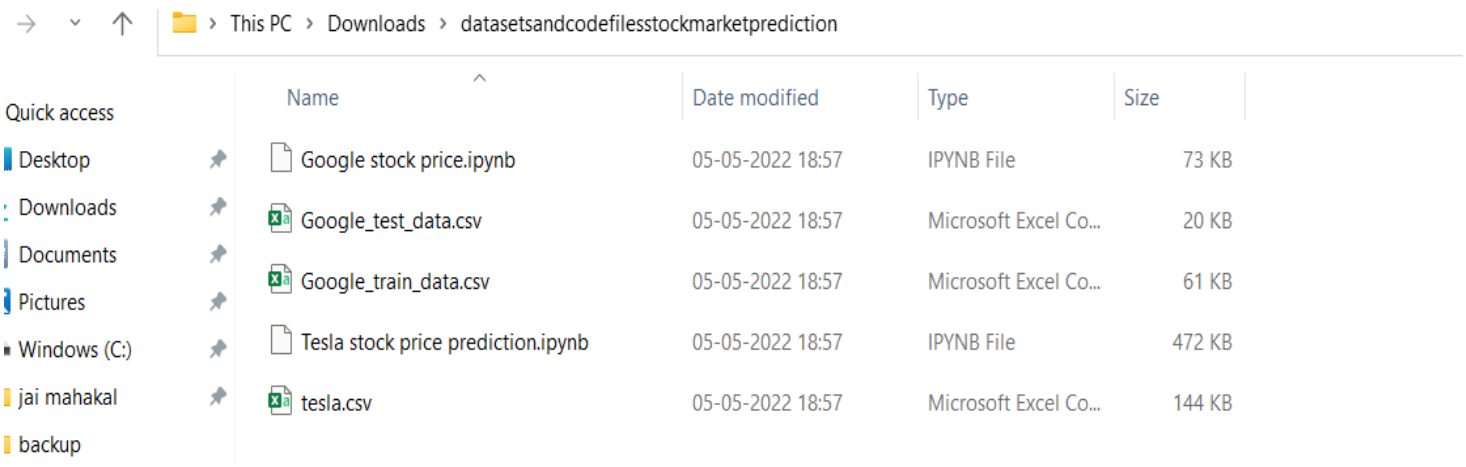
10. USER INTERFACE DESIGN

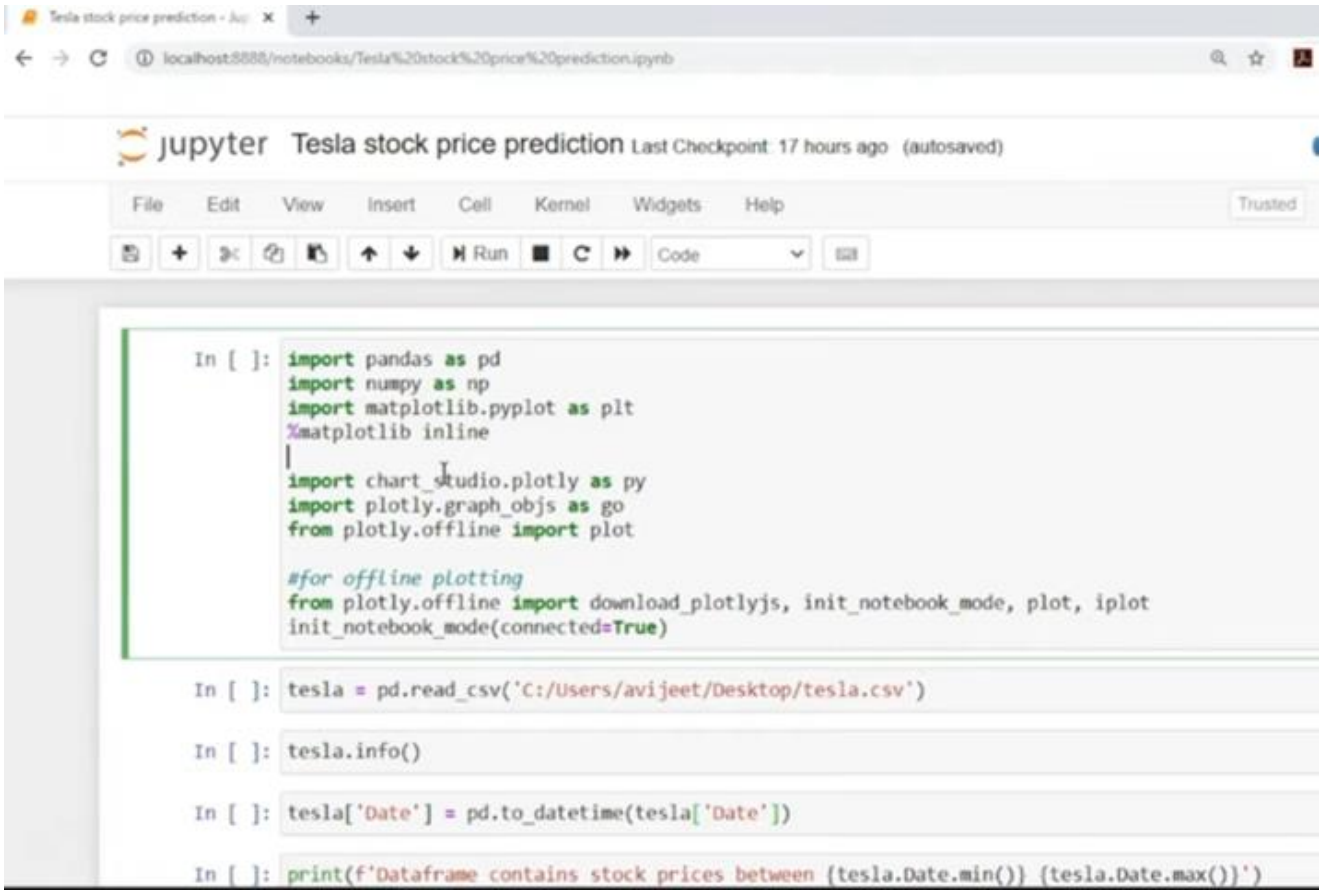
1. The design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centereddesign).

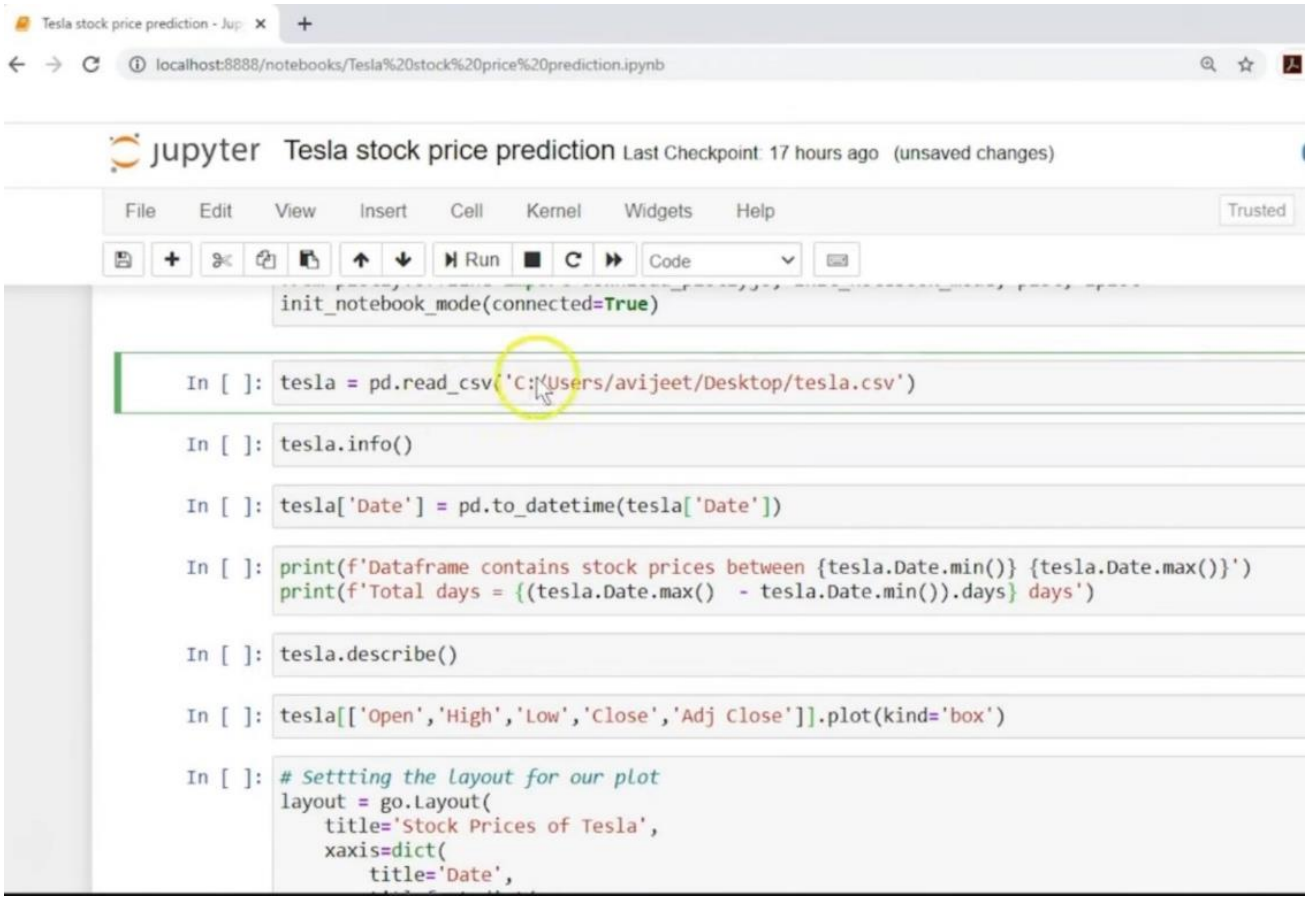
Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

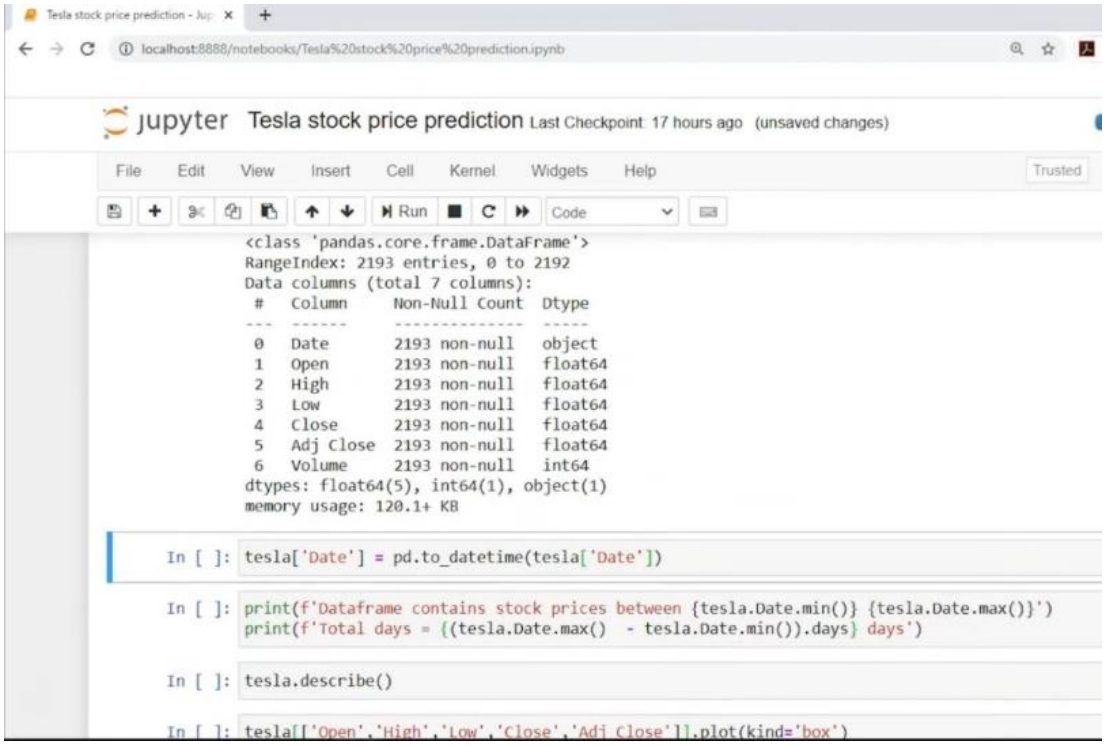
SNAPSHOTS

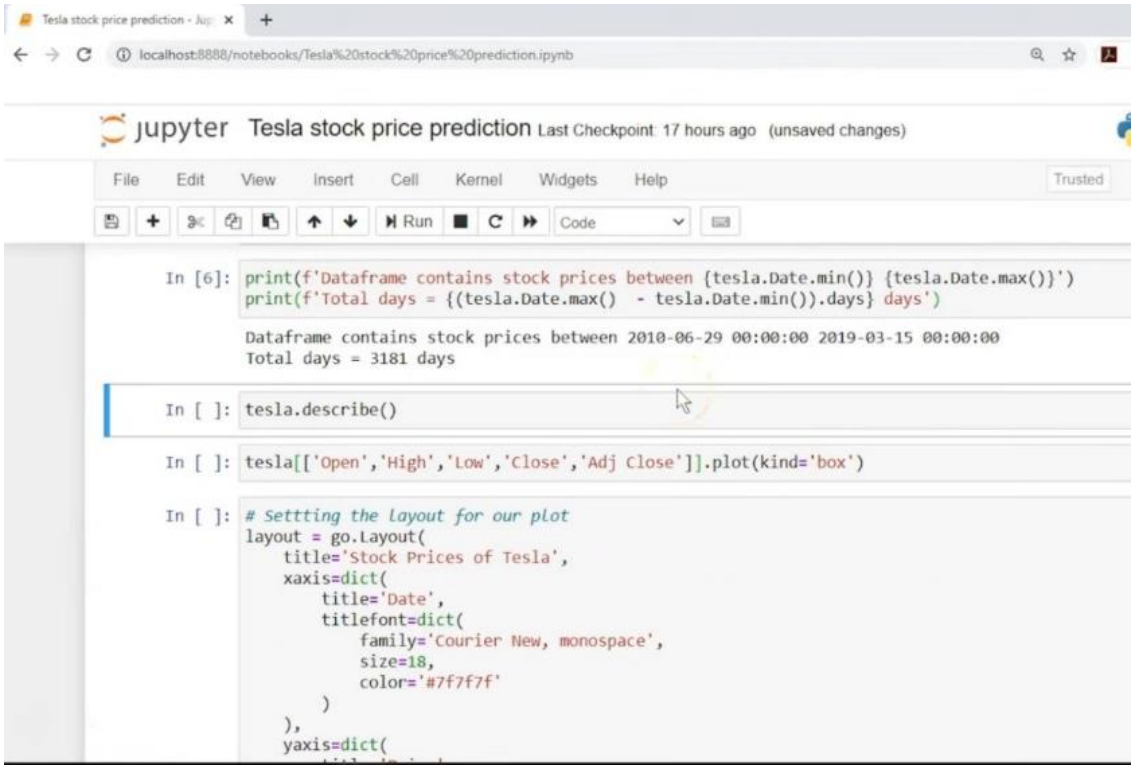
EXPERIMENTAL RESULTS Dataset Used:

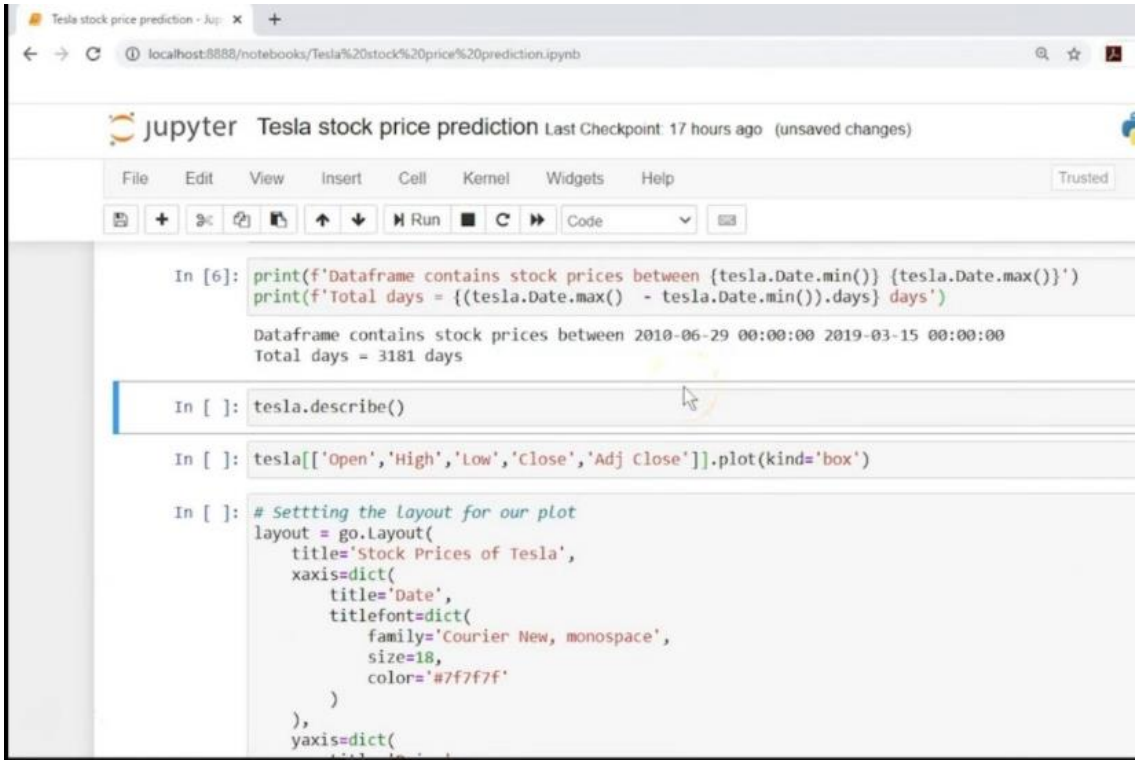


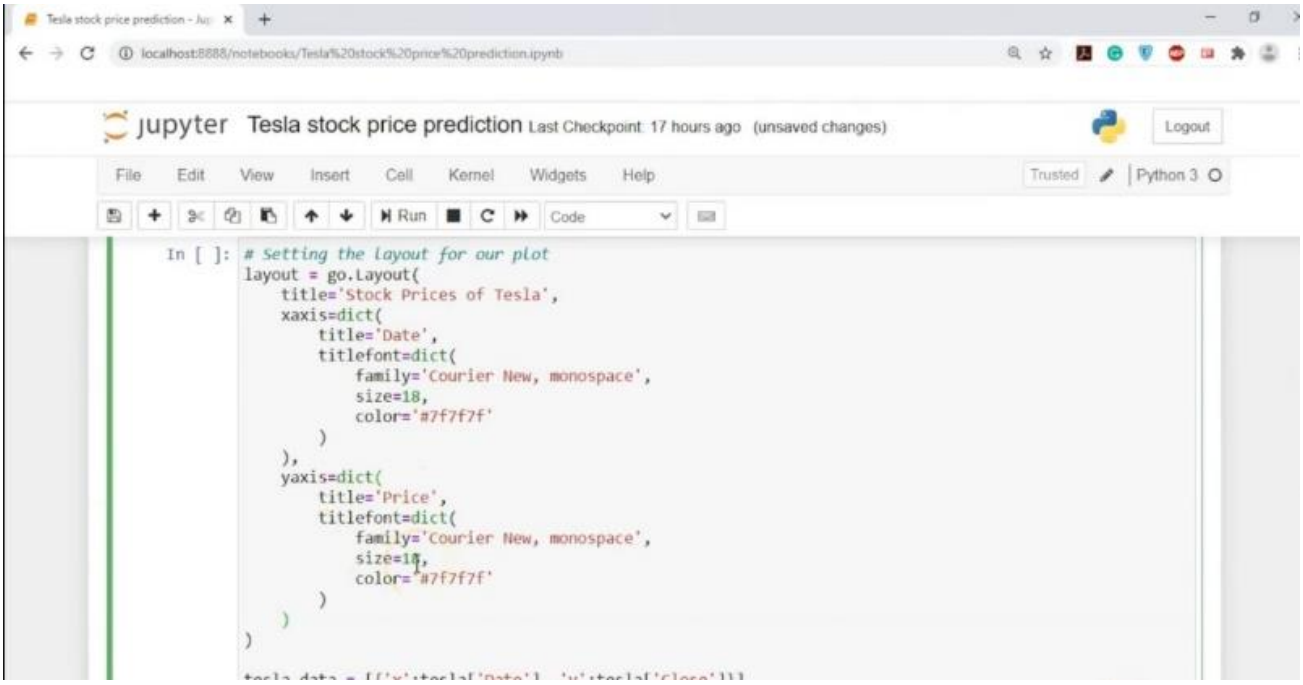


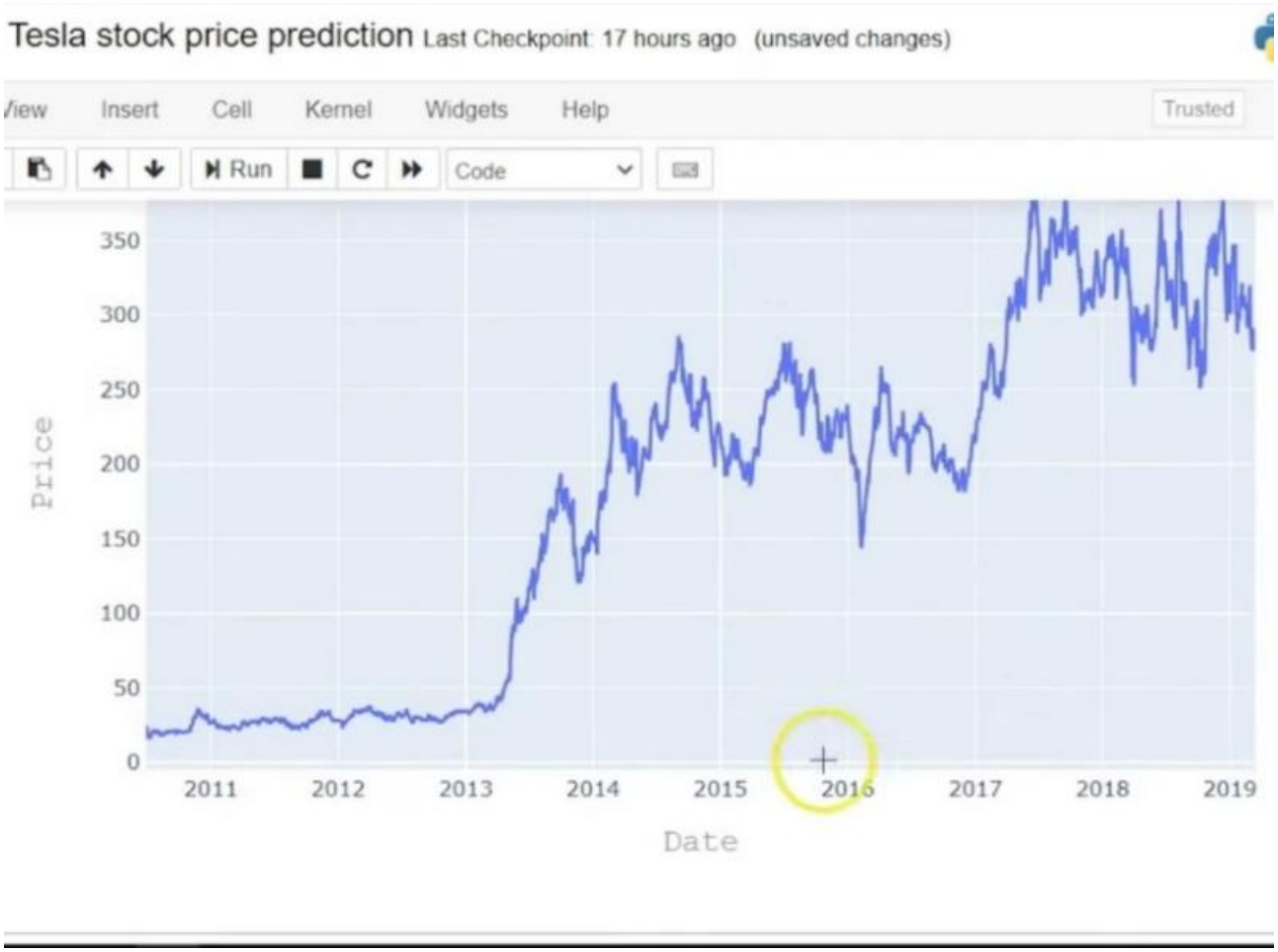

Tesla stock price prediction - Jup
localhost:8888/notebooks/Tesla%20stock%20price%20prediction.ipynb
jupyter Tesla stock price prediction Last Checkpoint: 17 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Run Code
init_notebook_mode(connected=True)
In [ ]: tesla = pd.read_csv('C:/Users/avijeet/Desktop/tesla.csv')
In [ ]: tesla.info()
In [ ]: tesla['Date'] = pd.to_datetime(tesla['Date'])
In [ ]: print(f'Dataframe contains stock prices between {tesla.Date.min()} {tesla.Date.max()}')
print(f'Total days = {(tesla.Date.max() - tesla.Date.min()).days} days')
In [ ]: tesla.describe()
In [ ]: tesla[['Open','High','Low','Close','Adj Close']].plot(kind='box')
In [ ]: # Settting the layout for our plot
layout = go.Layout(
title='Stock Prices of Tesla',
xaxis=dict(
title='Date',


Tesla stock price prediction - Jup
localhost:8888/notebooks/Tesla%20stock%20price%20prediction.ipynb
jupyter Tesla stock price prediction Last Checkpoint: 17 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Run Code
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2193 entries, 0 to 2192
Data columns (total 7 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 Date 2193 non-null object
1 Open 2193 non-null float64
2 High 2193 non-null float64
3 Low 2193 non-null float64
4 Close 2193 non-null float64
5 Adj Close 2193 non-null float64
6 Volume 2193 non-null int64
dtypes: float64(5), int64(1), object(1)
memory usage: 120.1+ KB
In [ ]: tesla['Date'] = pd.to_datetime(tesla['Date'])
In [ ]: print(f'Dataframe contains stock prices between {tesla.Date.min()} {tesla.Date.max()}')
print(f'Total days = {(tesla.Date.max() - tesla.Date.min()).days} days')
In [ ]: tesla.describe()
In [ ]: tesla[['Open','High','Low','Close','Adj Close']].plot(kind='box')

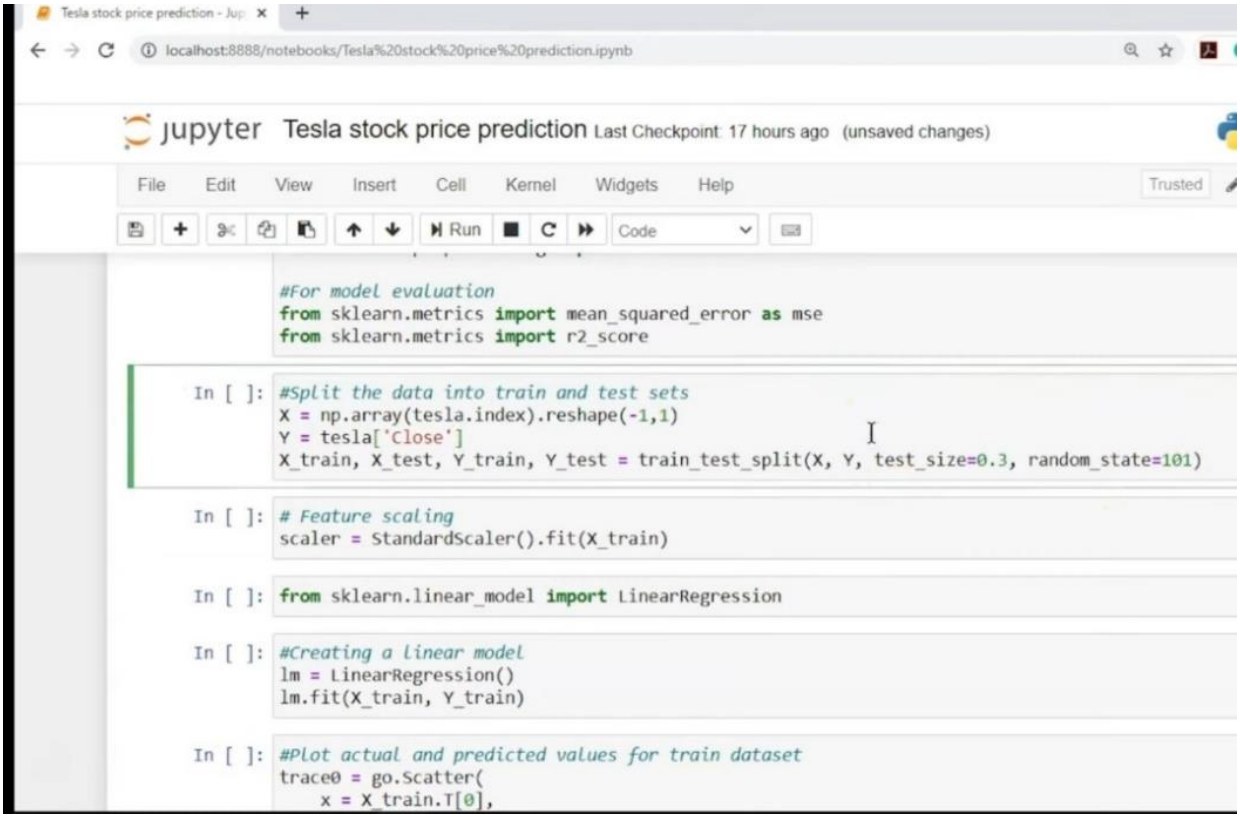

Tesla stock price prediction - Jup
localhost:8888/notebooks/Tesla%20stock%20price%20prediction.ipynb
jupyter Tesla stock price prediction Last Checkpoint: 17 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Run Code
In [6]: print(f'Dataframe contains stock prices between {tesla.Date.min()} {tesla.Date.max()}')
print(f'Total days = {(tesla.Date.max() - tesla.Date.min()).days} days')
Dataframe contains stock prices between 2010-06-29 00:00:00 2019-03-15 00:00:00
Total days = 3181 days
In [ ]: tesla.describe()
In [ ]: tesla[['Open','High','Low','Close','Adj Close']].plot(kind='box')
In [ ]: # Settting the layout for our plot
layout = go.Layout(
title='Stock Prices of Tesla',
xaxis=dict(
title='Date',
titlefont=dict(
family='Courier New, monospace',
size=18,
color='#7f7f7f'
)
),
yaxis=dict(

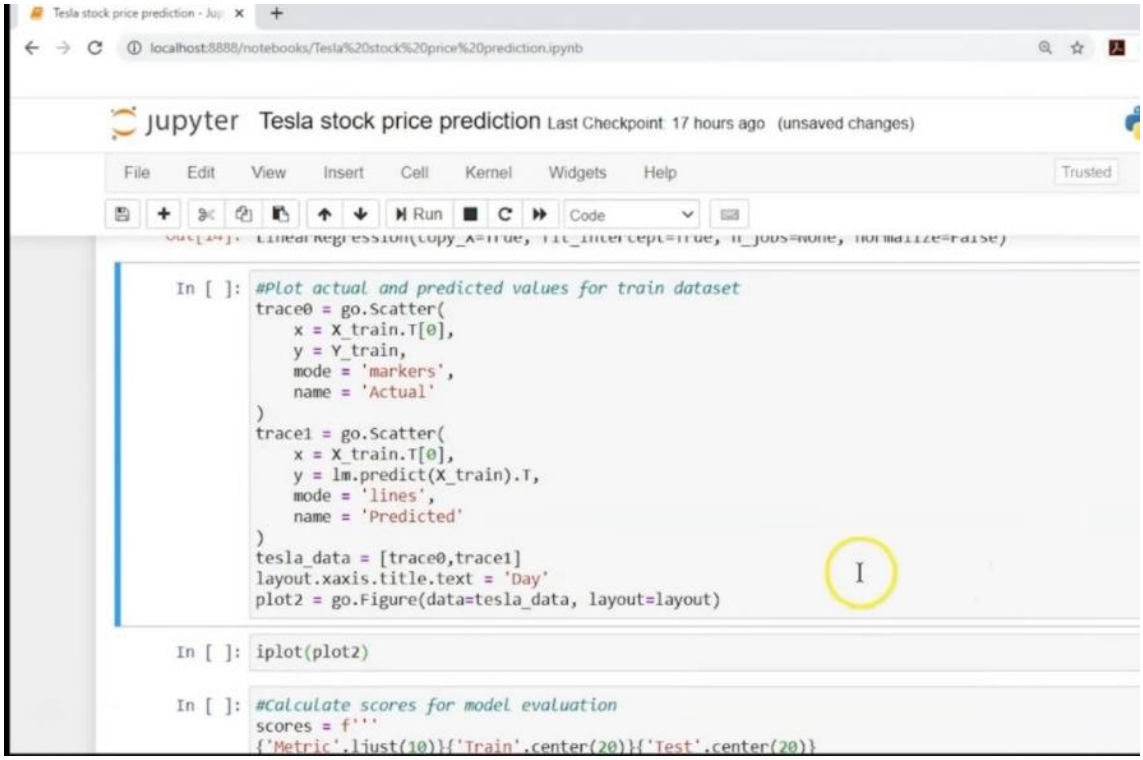

Tesla stock price prediction - Jup
localhost:8888/notebooks/Tesla%20stock%20price%20prediction.ipynb
jupyter Tesla stock price prediction Last Checkpoint: 17 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Run Code
In [6]: print(f'Dataframe contains stock prices between {tesla.Date.min()} {tesla.Date.max()}')
print(f'Total days = {(tesla.Date.max() - tesla.Date.min()).days} days')
Dataframe contains stock prices between 2010-06-29 00:00:00 2019-03-15 00:00:00
Total days = 3181 days
In [ ]: tesla.describe()
In [ ]: tesla[['Open','High','Low','Close','Adj Close']].plot(kind='box')
In [ ]: # Settting the layout for our plot
layout = go.Layout(
title='Stock Prices of Tesla',
xaxis=dict(
title='Date',
titlefont=dict(
family='Courier New, monospace',
size=18,
color='#7f7f7f'
)
),
yaxis=dict(

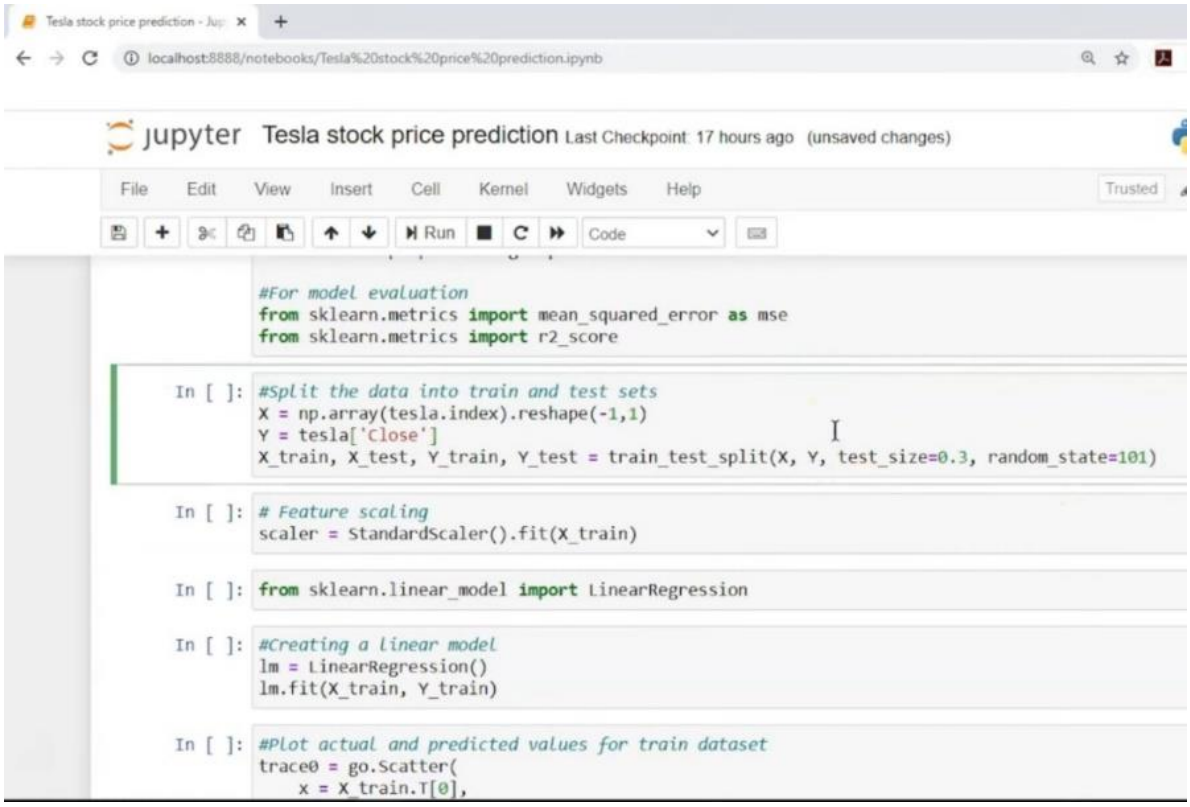

Tesla stock price prediction - Jup
localhost:8888/notebooks/Tesla%20stock%20price%20prediction.ipynb
jupyter Tesla stock price prediction Last Checkpoint: 17 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [ ]:
# Setting the layout for our plot
layout = go.Layout(
title='Stock Prices of Tesla',
xaxis=dict(
title='Date',
titlefont=dict(
family='Courier New, monospace',
size=18,
color='#7f7f7f'
)
),
yaxis=dict(
title='Price',
titlefont=dict(
family='Courier New, monospace',
size=18,
color='#7f7f7f'
)
)
)


Tesla stock price prediction Last Checkpoint: 17 hours ago (unsaved changes)
View Insert Cell Kernel Widgets Help
Trusted
Run
Code
350
300
250
200
150
100
50
0
Price
2011
2012
2013
2014
2015
2016
2017
2018
2019
Date

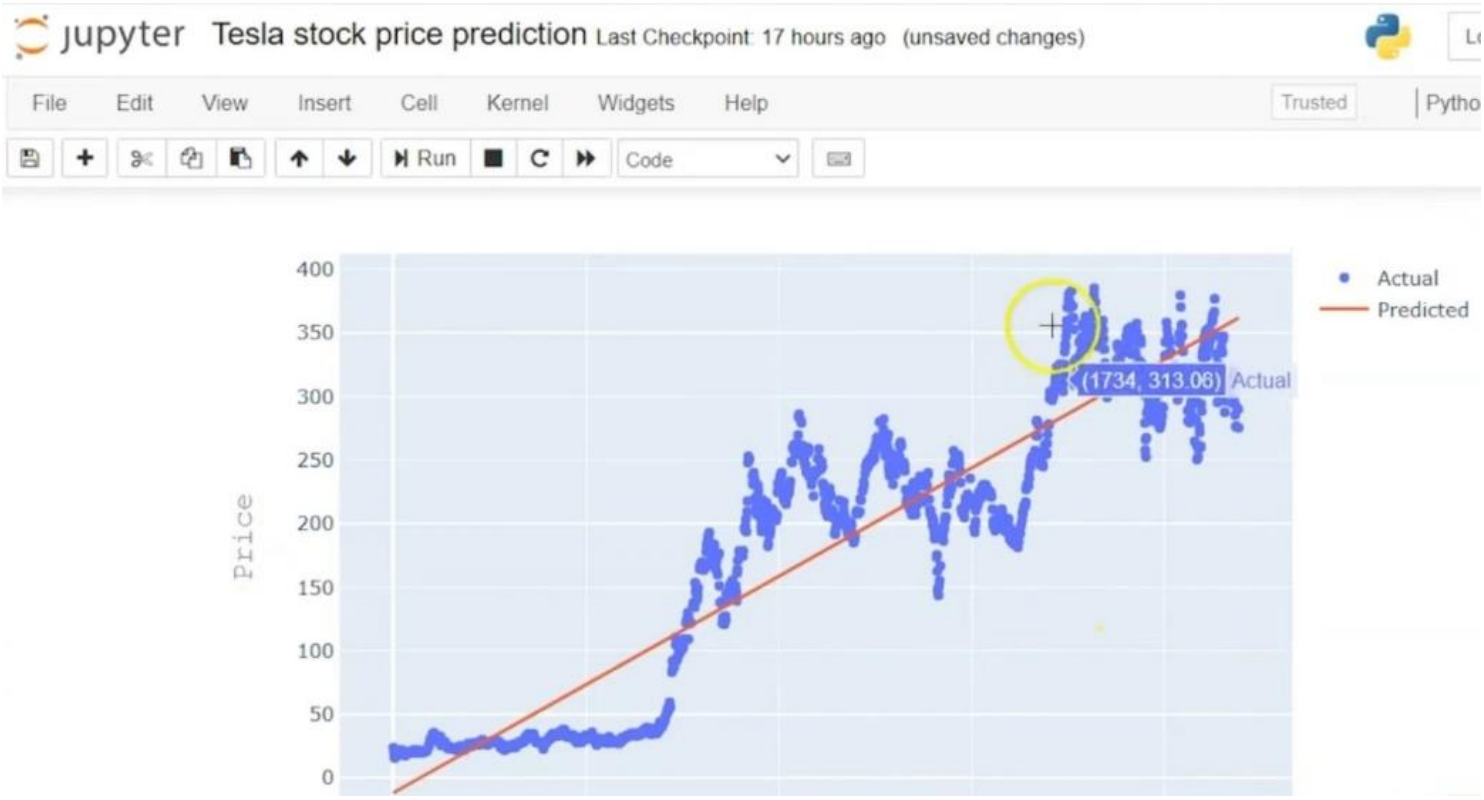
Training the model






Tesla stock price prediction - Jup
localhost:8888/notebooks/Tesla%20stock%20price%20prediction.ipynb
jupyter Tesla stock price prediction Last Checkpoint: 17 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Run
Code
#For model evaluation
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import r2_score
In [ ]:
#Split the data into train and test sets
X = np.array(tesla.index).reshape(-1,1)
Y = tesla['Close']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=101)
In [ ]:
# Feature scaling
scaler = StandardScaler().fit(X_train)
In [ ]:
from sklearn.linear_model import LinearRegression
In [ ]:
#Creating a linear model
lm = LinearRegression()
lm.fit(X_train, Y_train)
In [ ]:
#Plot actual and predicted values for train dataset
trace0 = go.Scatter(
x = X_train.T[0],

OUTPUT OF TRAINED MODEL



11. IMPLEMENTATION AND TESTING

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the "WHY" and "HOW" product validation. It should be through enough to be useful but not so through that no one outside the test group will read it.

- ## 11.1 Introduction

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the userrequirements.

The main purpose of testing is to detect errors and error prone areas in a system.Testing must be through well planned.A partially tested system is to detect errors and error prone areas in a system.Testing must be through well planned.A partially tested system is as bad as an untested system.And the price of an untested and under tested system is high.

11.2 Objectives Of Testing

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

11.3 Process Overview

The following represents the overall flow of the testing process:

- Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
- Identify which particular test(s) will be used to test each module.
- Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.

• 11.4 Test Cases

A test case is a document that describe an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular such as test case identifier, test condition, input data.

Requirement expected results. The process of developing test cases can help find problems in the requirements or design of an application since it requires completely thinking through the operations of the application.

• 11.5 Testing Steps

• Unit Testing

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

• Integration Testing

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface.

- **11.6 Validation**

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

a) The function or performance characteristics confirm to specification and are accepted.

b) A deviation from specification is uncovered and a deficiency lists is created.

c) Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

Tested By:	Nikita Sarkar
Test Type	Unit Testing
Test CaseNumber	1
TestCaseName	Prediction of Stock Price
Test Case Description	The user should enter any particular date. The system then predicts the closing price of the stock on that particular date. Now the predicted price can be compared with the actual closing price.

Item(s) to be tested	
1	Comparison of actual closing price and predicted price for determining the error.

Specifications	
Input	**Expected Output/Result**
1) Actual closing stock price of the day.	1) Output contains the predicted closing stock price of a particular day.

- **11.7 White Box Testing**

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

11.8 Black box testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

11.9 System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity.

11.10 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

11.11 User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.
This is done in regard to the following point:

1. Input Screen Design.
2. Output Screen Design.
3. Format of reports and other outputs.

• 11.12 Integration Testing

Software testing is always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

- Whether the software matches with the specification (i.e.process base) to verify the product.
- Whether this software in one client what wants (i.e. product base) to validate the product.
- Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was test to find any unexpected behaviour if exists. Later all the module was integrated and flat file was generated.

• 11.13 Functional Testing

These are the points concerned during the stress test:

- Nominal input: character is in putted in the place of digits and the system has to flash the message "Dataerror"
- Boundary value analysis: exhaustive test cases have designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

12. SYSTEM SECURITY MEASURES

- **12.1 Database Security**

System security measure is meant to be provided to make your system reliable and secured from unauthorized user may create threats to the system. So you should follow some security measures. We have used security levels in database level at system level.

- **12.2 System Security**

If we talk about the system security in our proposed system we have implemented with the help of maintain the session throughout the system's use. Once a user has logged out than he/she will not be able to perform any task before signing back again.

A high level of authentic login is given to the system so this is a very tedious task to enter without authorization and authentication.

13. COST ESTIMATION

The **Constructive Cost Model** (COCOMO) is a procedural **software cost estimation model** developed by **Barry W. Boehm**. Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMO additionally accounts for the influence of individual project phases.

Types of COCOMO

1. **Basic COCOMO**

   Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC, KLOC).
   COCOMO applies to three classes of software projects:

   - Organic projects - small teams with good experience working with less than rigidrequirements.

Semi-detached projects - medium teams with mixed experience working with a mix of rigid and less.

   - than rigid requirements.
   - Embedded projects - developed within a set of tight constraints. It is also combination of organic and semi-detached projects. (Hardware, software, operational, ...)

   The basic COCOMO equations take the form: Effort Applied (E) = $a*(KLOC)^b$[man-months]

   Development Time (D) = $c*(\text{Effort Applied})^d$ [months]

   People required (P) = Effort Applied / Development Time [count]

   Where,

   KLOC is the estimated number of delivered lines (expressed in thousands) of code for project. The coefficients a, b, c and d are given in the following table:

Software project	**a**	**b**	**c**	**d**
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Basic COCOMO is good for quick estimate of software costs. However, it does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and so on.

2. Intermediate COCOMO

Intermediate COCOMO computes software development effort as function of program size and a set of cost drivers that include subjective assessment of product, hardware, personnel and project attributes. This extension considers a set of four cost drivers, each with a number of subsidiary attributes: -

Product attributes:

1) Required software reliability
2) Size of application database
3) Complexity of the product

Hardware attributes:

1) Run-time performance constraints
2) Memory constraints
3) Volatility of the virtual machine environment
4) Required turnabout time

Personnel attributes

1) Analyst capability
2) Software engineering capability
3) Applications experience
4) Virtual machine experience
5) Programming language experience

Project attributes

1) Use of software tools
2) Application of software engineering methods
3) Required development schedule

Each of the 15 attributes receives a rating on a six-point scale that ranges from very low to extra high (in importance or value). An effort multiplier from the table below applies to the rating. The product of all effort multipliers results in an effort adjustment factor (EAF). Typical values for EAF range from 0.9 to 1.4. The Intermediate COCOMO formula now takes the form:

$E=a*(KLOC)^b(EAF)$

Where,

E is the effort applied in person-months, KLOC is the estimated number of thousands of delivered lines of code for the project, and EAF is the factor calculated above. The coefficient a and the exponent b is given in the table:

Software project	**a**	**B**
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

The Development time D calculation uses P in the same way as in the Basic COCOMO.

3. Detailed COCOMO

Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process. The detailed model uses different effort multipliers for each cost driver attribute. These Phase Sensitive effort multipliers are each to determine the amount of effort required to complete each phase. In detailed COCOMO, the whole software is divided in different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort.

In detailed COCOMO, the effort is calculated as function of program size and a set of cost drivers given according to each phase of software life cycle. A Detailed project schedule is never static. The Six phases of detailed COCOMO are: -

- Plan and requirement.
- System design.
- Detailed design.
- Module code and test.
- Integration and test.
- Cost Constructive model

Detailed cost estimation of Movie Recommender System

So by considering all the facts, if we calculate the cost, it will be like;

The OES is having 2.1 Kilo Lines of Code. According to the COCOMO model the comparison is:

Mode	Project Size	Innovation	Deadline	Development Environment
Organic	Typically 2-50 KLOC	Little	Not tight	Familiar and in house
Semi-detached	Typically 50-300 KLOC	Medium	Medium	Medium
Embedded	Typically Over 300 KLOC	Significant	Tight	Complex hardware/Customer interface is required

The project has 2.1 KLOC, so it's under **organic** category.

Putting the facts in the formulas,

Effort Applied (E) = $a*(KLOC)^b$ [person-months]

= $2.4*(2.1)^{1.05}$ PM

= **5 PM**

Development Time (D) = $c*(\text{Effort Applied})^d$ [months]

= $2.5*(5)^{0.38}$ M

= **4.6 M**

14. FUTURE SCOPE AND FURTHER ENHANCEMENTS

The popularity of stock market trading is growing rapidly, which is encouraging researchers to find out new methods for the prediction using new techniques. The forecasting technique is not only helping the researchers but it also helps investors and any person dealing with the stock market. In order to help predict the stock indices, a forecasting model with good accuracy is required. In this work, we have used one of the most precise forecasting technology using Long Short-Term Memory unit which helps investors, analysts or any person interested in investing in the stock market by providing them a good knowledge of the future situation of the stock market.

15. CONCLUSION

Predicting stock market returns is a challenging task due to consistently changing stock values which are dependent on various parameters which form complex patterns. The dataset available on the website consists of only few features like high, low, open, close, adjacent close value of stock prices, volume of shares traded etc., which are not sufficient enough. To obtain higher accuracy in the predicted price value new variables have been created using the existing variables. LSTM (Long Short Term Memory) is used for predicting the next day closing price of the stock and for a analysis. The analysis based on MSE (Mean Squared Error) values clearly indicate that LSTM gives better prediction of stock prices. For future work, deep learning models could be developed which financial news articles along with financial parameters such as a closing price, traded volume, profit and loss statements etc., for possibly better results.

16. BIBLIOGRAPHY

- Masoud, Najeb MH. (2017) "The impact of stock market performance upon economic growth." International Journal of Economics and Financial Issues **3 (4)** : 788–798.
- Lawrence Sirovich and Michael Kirby.
- Takeo Kanade, volume 47.Birkhauser Basel, 1977.

- Hur, Jung, Manoj Raj, and Yohanes E. Riyanto. (2006) "Finance and trade: A cross-country empirical analysis on the impact of fifinancial development and asset tangibility on international trade." World Development **34 (10)** : 1728-1741.

- Li, Lei, Yabin Wu, Yihang Ou, Qi Li, Yanquan Zhou, and Daoxin Chen. (2017) "Research on machine learning algorithms and feature extraction for time series." IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)