

SENTIMENT ANALYSIS ON ROMAN URDU DATASET

FINAL REPORT

Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Registration No.

Name

Roll No.

11701711

Subhadip Mondal

RKMO71A13

11701716

Smaranika Datta

RKMO71A14

11702551

Mohammad Abuzar

RKMO71A15

11703127

Amrita Chaudri

RKMO71A16

Under the guidance of

Dr. Aditya Khamparia



School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

NOV 2019

ACKNOWLEDGMENT

Place: - Lovely Professional University

Date: - 11th November, 2019

We would like to thank Dr. Aditya Khamparia for assigning us with this project. Through the project we are able to grasp more technical and have a hands-on practical experience with python and some machine learning algorithms. Through it we are able to learn how a project is created and how necessary and crucial technical knowledge is. We are really grateful to the faculty that has provided us with the necessary guidelines.

Registration No.	Name
11701711	Subhadip Mondal
11701716	Smaranika Datta
11702551	Mohammad Abuzar
11703127	Amrita Chaudri

DECLARATION STATEMENT

Place: - Lovely Professional University

Date: - 11th November, 2019

This is to declare that this report has been written by us. No part of the report is copied from the other sources. All information included from other sources has been duly acknowledged. We aver that if any part of the report is found to be copied, we will take full responsibility for it.

Name : Subhadip Mondal

Reg No. : 11701711

Signature :

Name : Smaranika Datta

Reg No. : 11701716

Signature :

Name : Mohammad Abuzar

Reg No. : 11702551

Signature :

Name : Amrita Chaudri

Reg No. : 11703127

Signature :

TABLE OF CONTENTS

1. INTRODUCTION

- 1.1. Introduction
- 1.2. Uses
- 1.3. Special Button

2. METHODOLOGY

- 2.1. Requirements
- 2.2. Functionality
- 2.3. Implementation
- 2.4. Debugging
- 2.5. Execution
- 2.6. Working

3. TECHNOLOGY

- 3.1. Python 3.7
- 3.2. Machine Learning
- 3.3. Natural Language Toolkit
- 3.4. Pandas
- 3.5. Numpy
- 3.6. Matplotlib
- 3.7. Sklearn
- 3.8. GUI
- 3.9. IDE
- 3.10. Tkinter
- 3.11. Widgets classes
- 3.12. Option with values

4. WORK DIVISION

- 4.1. Sentiment Analysis Home page and User Interface
- 4.2. Sentiment Analysis About Developer page.
- 4.3. Sentiment Analysis Data Cleaning and User Interface
- 4.4. Sentiment Analysis Model Creation

5. REFERENCES

LIST OF TABLES

<i>Table No.</i>	<i>Name</i>
4.1	Work Division
4.2	Gantt Chart
4.3	Flow Chart

LIST OF FIGURES

<i>Fig No.</i>	<i>Name</i>
1.3.1.	About Developer
1.3.2.	User Interface
1.3.3.	Return to Home
1.3.4	Help Center
1.3.4.1.	Help Instruction
1.3.5.	Check Accuracy
1.3.5.1.	Model Accuracy value
1.3.6.	Confusion Matrix button
1.3.6.1.	Confusion Matrix
1.3.7.	View Original data button
1.3.7.1.	View Original data
1.3.8.	Predict Sentiment button
1.3.9.	EXIT

CHAPTER 1

INTRODUCTION

1.1. Introduction

We had performed sentiment analysis on Roman Urdu Dataset using Python, NLTK and Sklearn. We had used machine learning techniques to perform this task we will be doing classification on a dataset.

1.2. Uses

Model created will help the user to predict the sentiment (POSITIVE or NEGATIVE or NEUTRAL) of their review or comment.

All the button used in this project perform some special kind of operations.

- Predict the sentiment of user input comment
- Know about developer
- Go to home page
- View actual data
- View confusion matrix
- View accuracy score
- Get help
- Exit from the program

Button used in this project are like EXIT, OK, PREDICT etc.

1.3. Special Button

Sentiment Analysis on Roman Urdu Dataset project uses some special buttons.

All the button used in this program are

- KNOW ABOUT DEVELOPER
- CLICK TO TAKE USER INPUT
- RETURN TO HOME PAGE
- HELP CENTER
- VIEW ACCURACY SCORE
- VIEW CONFUSION MATRIX
- VIEW ORIGINAL DATA
- PREDICT SENTIMENT
- OK
- EXIT

KNOW ABOUT DEVELOPER



Fig 1.3.1

Code we used in our program...

```
Button(home_window, image = about_us_dev, font="halston 20 italic",  
cursor="hand2", command = About_Dev).grid(row = 1, column = 0)
```

This button is defined in the program named "home.py".

This button tells us about the developer.

On clicking this button, we will go to the program name "about_us.py" from "home.py".

CLICK TO TAKE USER INPUT



Fig 1.3.2

Code we used in our program...

```
Button(home_window, image = take_UI, font="halston 20 italic",  
cursor="hand2", command = Predict_Input).grid(row = 1, column = 1)
```

This button is defined in the program named "home.py".

This button help us to take user input.

On clicking this button, we will go to the program name “about_us.py” from “home.py”.

RETURN TO HOME PAGE



Fig 1.3.3

Code we used in our program...

```
Button(about_us_window5, image=home_button, cursor="hand2",  
command=home_page).grid(row = 2, column = 0, columnspan = 3, pady=10)
```

This button is defined in the program named “about_us.py”.

This button helps us to go to home page.

On clicking this button, we will go to the program name “home.py” from “about_us.py”.

HELP CENTER



Fig 1.3.4

Code we used in our program...

```
Button(user_input_window, image = help_image, cursor="hand2", command =  
Take_Help).grid(row = 2, column = 2, padx =5, pady = 5)
```

This button is defined in the program named “PredictUserInput.py”.

On clicking this button, user will get an helpful instruction to take input.

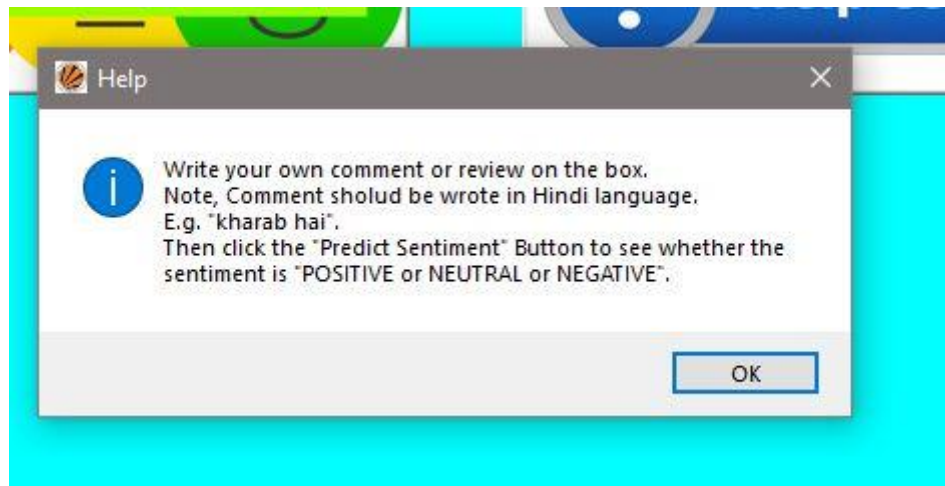


Fig 1.3.4.1

VIEW ACCURACY SCORE



Fig 1.3.5

Code we used in our program...

```
Button(user_input_window, image = accuracy_image, cursor="hand2",
command = View_Model_Accuracy).grid(row = 2, column = 0, padx =5, pady =
5)
```

This button is defined in the program named "PredictUserInput.py".
On clicking this button, we will check the accuracy we get from the model.

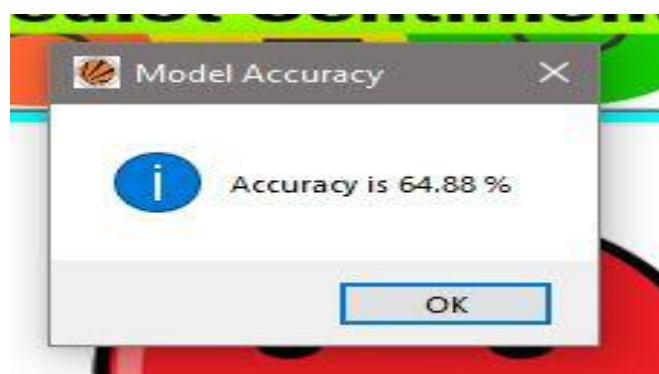


Fig 1.3.5.1

VIEW CONFUSION MATRIX



Fig 1.3.6

Code we used in our program...

```
Button(user_input_window, image = confusion_image, cursor="hand2",
command = Visualize_C_Matrix).grid(row = 1, column = 0, padx =5, pady = 5)
```

This button is defined in the program named "PredictUserInput.py".
On clicking this button, we can view the confusion matrix.

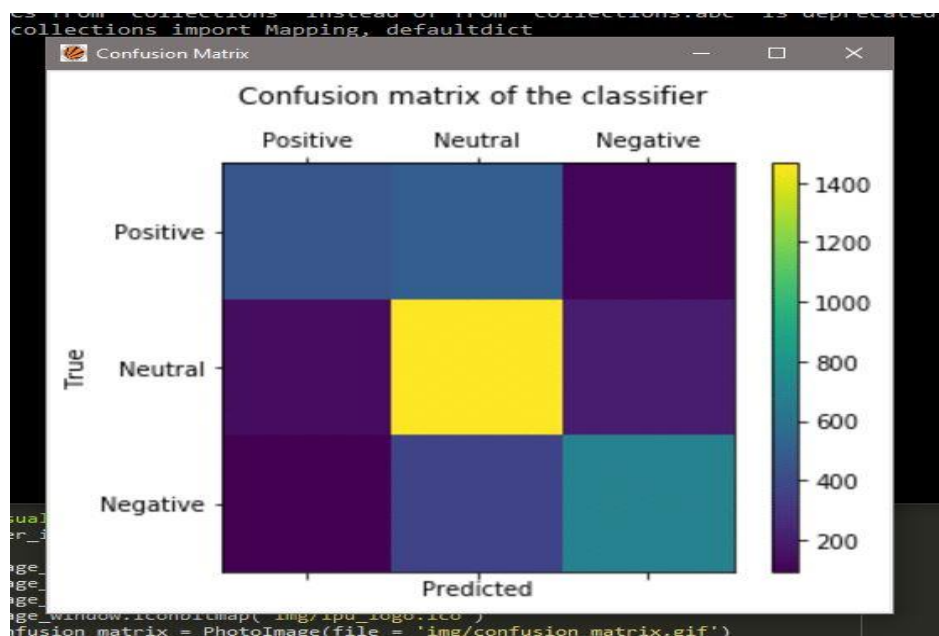


Fig 1.3.6.1

VIEW ORIGINAL DATA



Fig 1.3.7

Code we used in our program...

```
Button(user_input_window, image = data_image, cursor="hand2", command =
See_Main_Data).grid(row = 0, column = 0, padx =5, pady = 5)
```

This button is defined in the program named “PredictUserInput.py”.
On clicking this button, we can check the data in visualize manner.

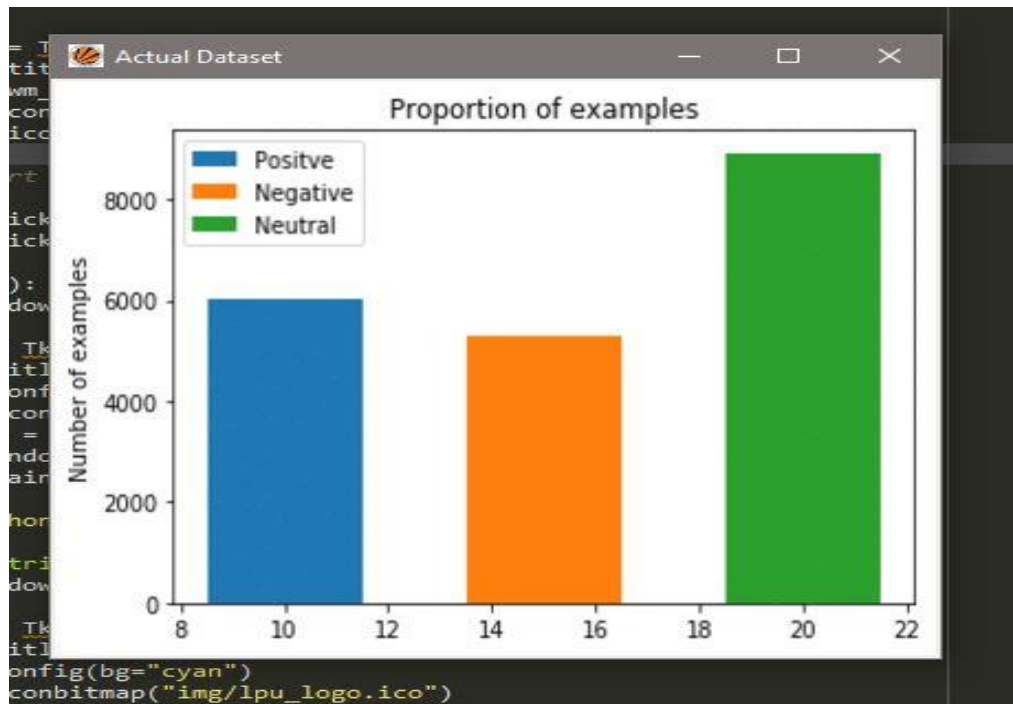


Fig 1.3.7.1

PREDICT SENTIMENT



Fig 1.3.8

Code we used in our program...

```
Button(user_input_window, image = predict_image, cursor="hand2", command  
= Predict_User_Input).grid(row = 2, column = 1)
```

This button is defined in the program named “PredictUserInput.py”.
On clicking the button, user will able to know the sentiment.

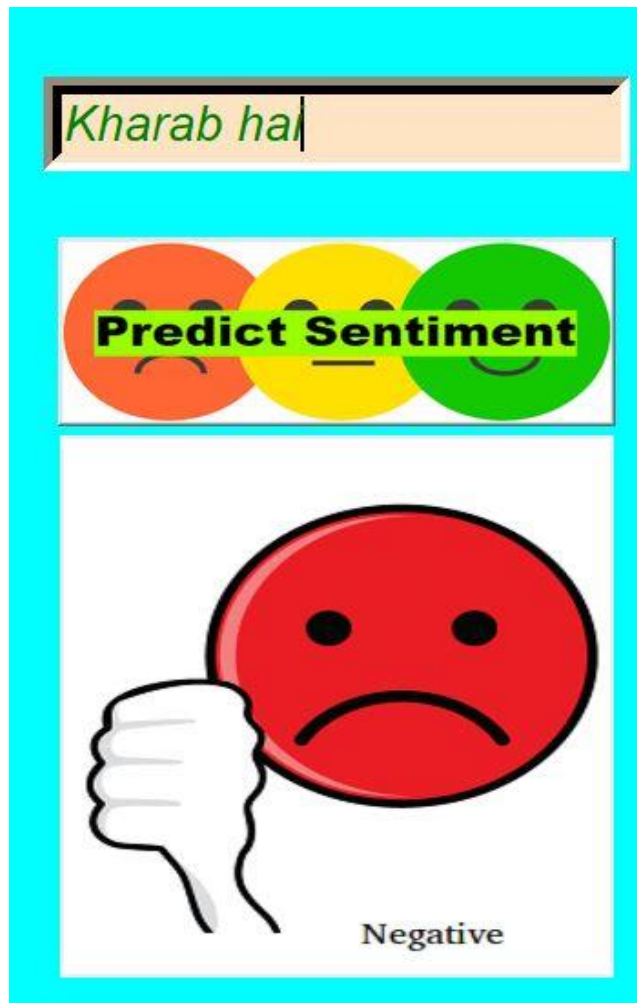


Fig 1.3.8.1

OK

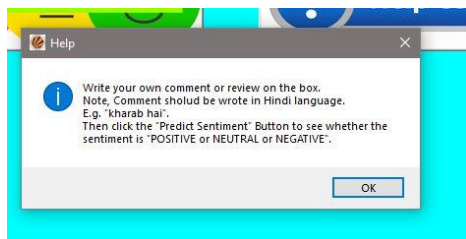


Fig 1.3.9

Code we used in our program...

```
from tkinter import messagebox
messagebox.showinfo('Help', 'Write your own comment or review on the
box.\nNote, Comment sholud be wrote in Hindi language.\nE.g. "kharab
hai".\nThen click the "Predict Sentiment" Button to see whether the sentiment
is "POSITIVE or NEUTRAL or NEGATIVE".')
```

This button is defined in the program named “PredictUserInput.py”.

EXIT



Fig 1.3.10

Code we used in our program...

```
Button(user_input_window, image = exit_image, cursor="hand2", command =  
exit_window).grid(row = 1, column = 3, padx =5, pady = 5)
```

This button is defined in the program named “about_us.py”, “home.py” and “PredictUserInput.py”.

On clicking this button, we will exit from the program.

CHAPTER 2

METHODOLOGY

2.1. Requirements

First step is gathering all the information required of project. Learn all the technology that maybe used in the project. The functions and the modules are the necessary for the development of the project.

2.2. Functionality

In this step, we will discuss the different function used in the program and how they are working.

The different types of function are used are...

- `exit_window()`
- `About_Dev()`
- `Predict_Input()`
- `home_page()`
- `See_Main_Data()`
- `Visualize_C_Matrix()`
- `View_Model_Accuracy()`
- `Take_Help()`
- `Predict_User_Input()`

Now we will discuss working of each of the function.

`exit_window()`

This function is defined in the program named “home.py”, “about_us.py” and “PredictUserInput.py”

Code we used is...

```
def exit_window():  
    home_window.destroy()
```

This function help us to exit from program.

`About_Dev()`

This function is defined in the program named “home.py”

Code we used is...

```
def About_Dev():  
    home_window.destroy()
```

```
os.system('python about_us.py')
```

This function gives us all information about the developers those who made this project.

Predict_Input()

This function is defined in the program named "home.py"

Code we used is...

```
def Predict_Input():  
    home_window.destroy()  
    os.system('python PredictUserInput.py')
```

This function destroys the current window or file and land us to another python file.

home_page()

This function is defined in the program named "PredictUserInput.py" and "about_us.py"

Code we used is...

```
def home_page():  
    about_us_window.destroy()  
    os.system("python home.py")
```

This function destroys the current window or file and land us to another python file.

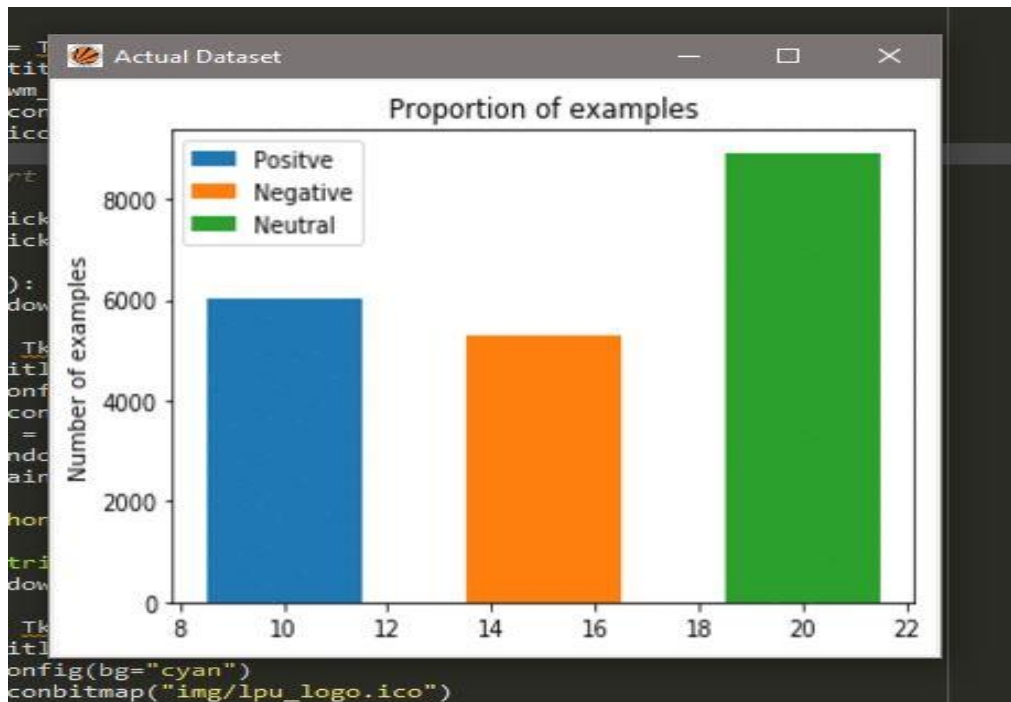
See_Main_Data()

This function is defined in the program named "PredictUserInput.py"

Code we used is...

```
def See_Main_Data():  
    user_input_window.destroy()  
  
    image_window = Tk()  
    image_window.title("Actual Dataset")  
    image_window.config(bg="cyan")  
    image_window.iconbitmap("img/lpu_logo.ico")  
    actual_dataset = PhotoImage(file = 'img/actual_data.gif')  
    Label(image_window, image = actual_dataset).grid(row = 0, column = 0)  
    image_window.mainloop()  
  
    os.system('python PredictUserInput.py')
```

This function displays the original data in graphical manner.



Visualize_C_Matrix()

This function is defined in the program named "PredictUserInput.py"

Code we used is...

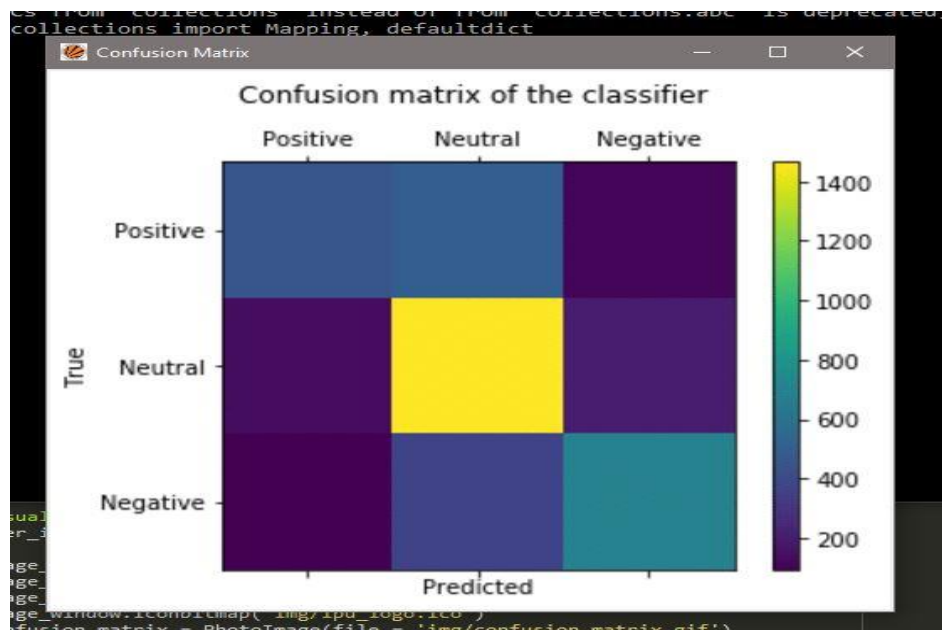
```
def Visualize_C_Matrix():
    user_input_window.destroy()

    image_window = Tk()
    image_window.title("Confusion Matrix")
    image_window.config(bg="cyan")
    image_window.iconbitmap("img/lpu_logo.ico")
    confusion_matrix = PhotoImage(file = 'img/confusion_matrix.gif')
    Label(image_window, image = confusion_matrix).grid(row = 0, column =
0)

    image_window.mainloop()

    os.system('python PredictUserInput.py')
```

This function displays the confusion matrix in graphical manner.



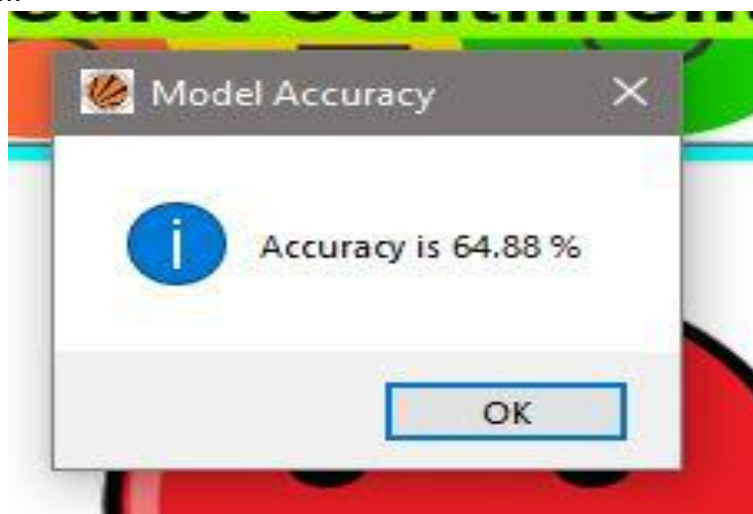
View_Model_Accuracy()

This function is defined in the program named "PredictUserInput.py"

Code we used is...

```
def View_Model_Accuracy():
    file = open('AccuracyPercentage', 'rb')
    Accuracy = pickle.load(file)
    file.close()
    from tkinter import messagebox
    messagebox.showinfo('Model Accuracy', 'Accuracy is ' + Accuracy)
```

This function load the pickle file and show the accuracy of the machine learning model.



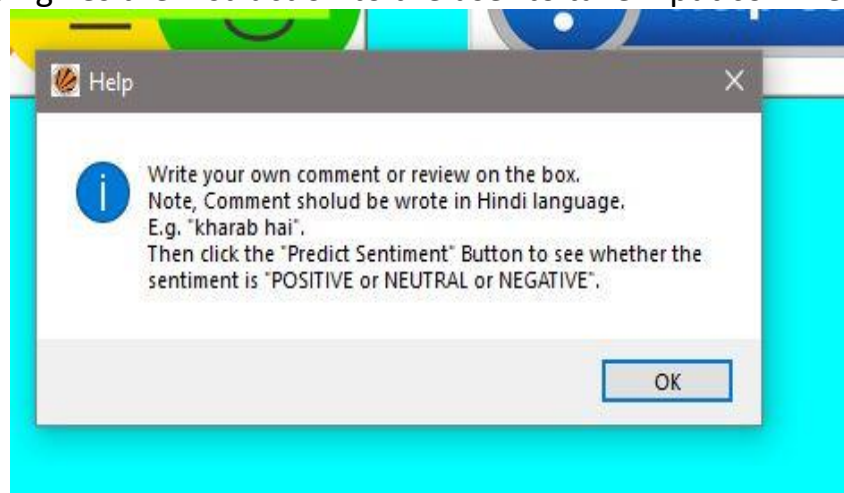
Take_Help()

This function is defined in the program named "PredictUserInput.py"

Code we used is...

```
def Take_Help():  
    from tkinter import messagebox  
    messagebox.showinfo('Help', 'Write your own comment or review on  
the box.\nNote, Comment sholud be wrote in Hindi language.\nE.g. "kharab  
hai".\nThen click the "Predict Sentiment" Button to see whether the sentiment  
is "POSITIVE or NEUTRAL or NEGATIVE".')
```

This function gives the instruction to the user to take input comment.



Predict_User_Input()

This function is defined in the program named "PredictUserInput.py"

This function gives an interface to take user input comment and then clean the data. After that our machine learning model will predict the sentiment according to that.

Enter Comment / Review in Hindi



2.3. Implementation

This step is to implementing the project by coding. This is most important part of the program. A particular care should be done in order to get error free program.

2.4. Debugging

This step is to check the project for any errors in the code. And linking of the module take place after checking individual modules and functions.

2.5. Execution

This step is to executing the project and simple testing.

2.6. Working

Module used in this project are

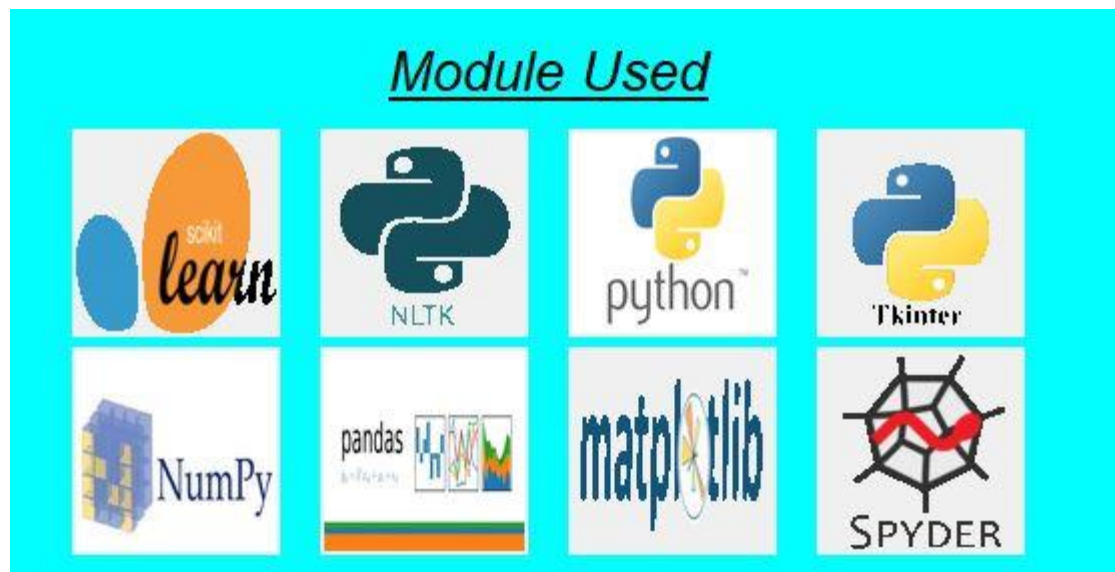


Fig 2.6.1

We have used tkinter library to develop GUI for the project.

We have used nltk to tokenize the sentence.

We have used numpy for numeric operation.

We have used pandas for data structures and operations for manipulating numerical tables and time series.

We have used matplotlib for plotting the graph.

We have used sklearn for selecting most appropriate classifier.

We import sys library to know the current version we are using.

We import messagebox from tkinter to show information.

We import os for linking a program with another program.

The detail description about the working of the project is discussing now. On running the program name "home.py" we will see this interface

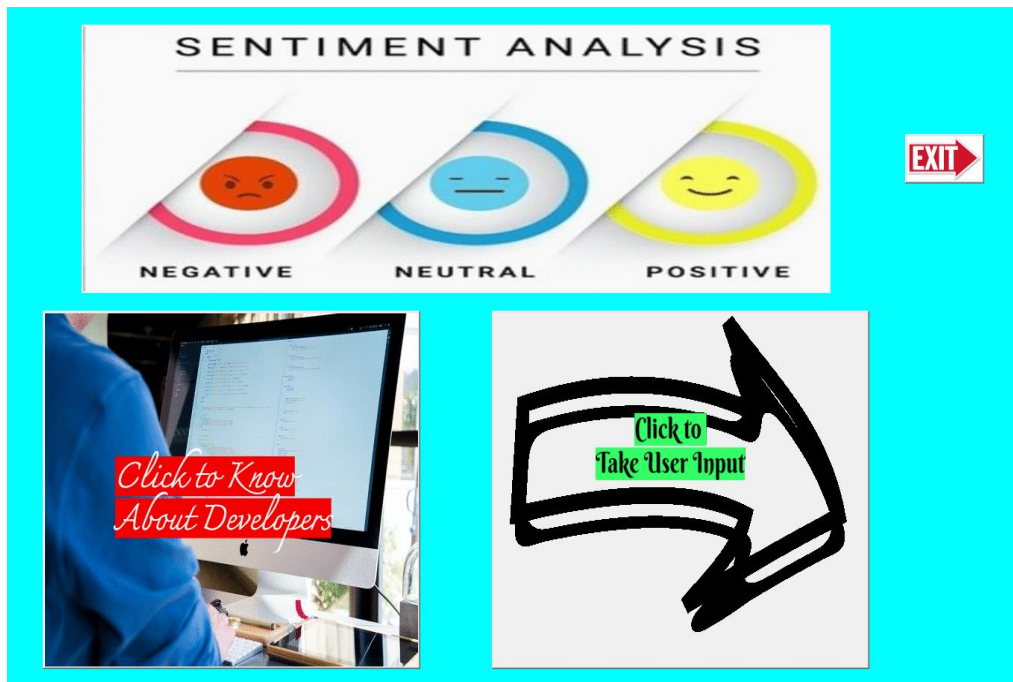


Fig 2.6.2

User can get the two interfaces. One is to check about the developer and other is to take comment or review as input.

On clicking the button click to know about developer, user landed to the file called “about_us.py”. Interface is

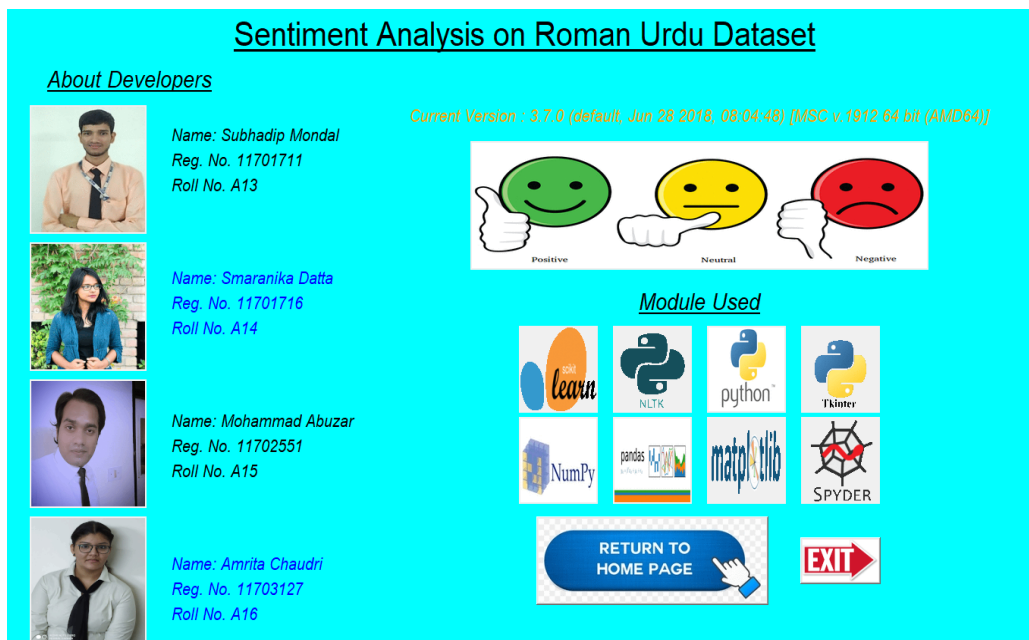


Fig 2.6.3

Then the user click the return to home page button and landed to the previous page.

After that on clicking the button called “Take User Input”, they landed to the file called “PredictUserInput.py”. The user interface is like this

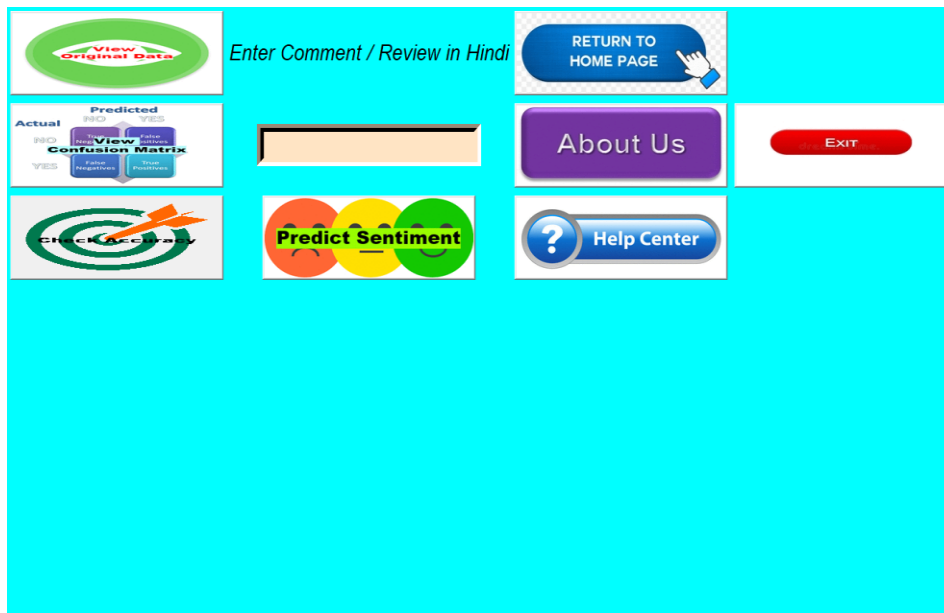
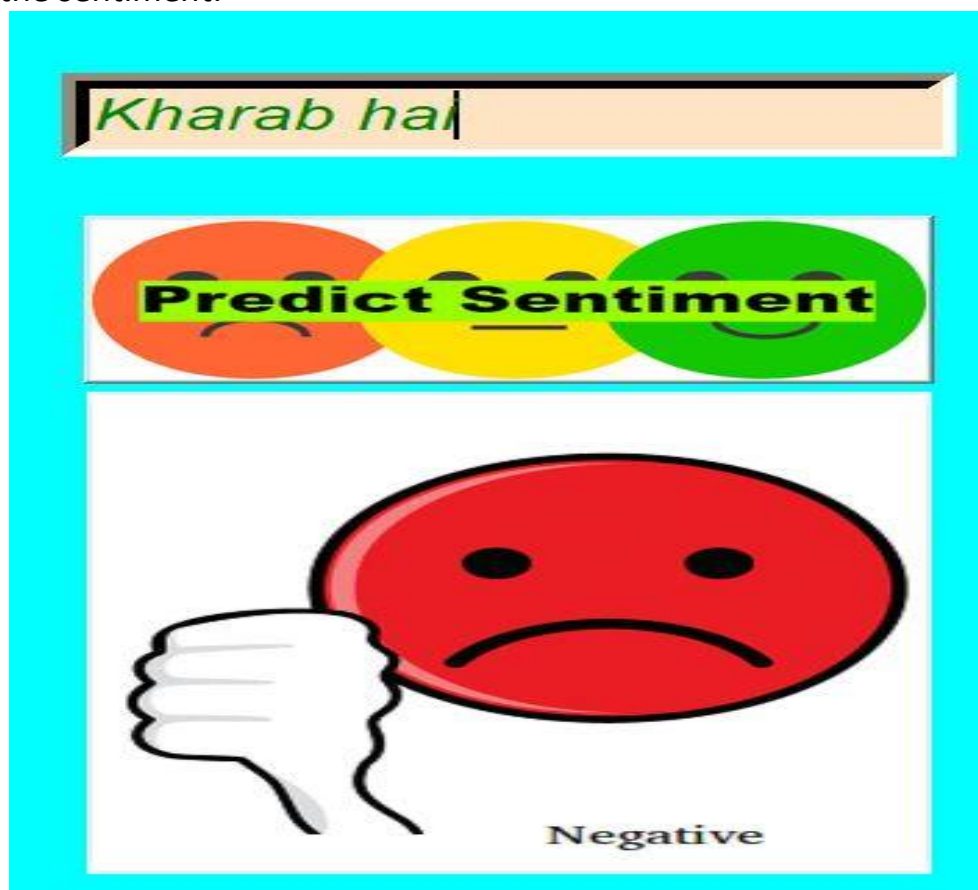


Fig 2.6.4

User can type their comment on the box and click the predict sentiment button to see the sentiment.



They can use other buttons according to their requirements. Then click the EXIT button to exit from the program.

3.1. Python 3.7

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

3.2. Machine Learning

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning.

3.3. Natural Language Toolkit

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

3.4. Pandas

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.

3.5. Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

3.6. Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for

embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

3.7. Sklearn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines and others.

3.8. GUI

Short for Graphical User Interface, a GUI (pronounced as either G-U-I or gooey) allows the use of icons or other visual indicators to interact with electronic devices, rather than using only text via the command line.

A GUI uses icons, and menus to carry out commands, such as opening, deleting, and moving files.

3.9. IDE

Integrated development environment (IDE) used is Spyder. It provides code analysis, a graphical debugger, an integrated unit tester etc.

3.10. Tkinter

Python offers multiple options for developing GUI. Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications.

3.11. Widgets classes

Tkinter's GUI classes define common GUI widgets such as button, labels, check buttons, frames, canvases and other. We will use some of them in our project.

3.12. Option with values

Widgets classes contains option with their values to change the interface of the widgets classes means appearance of the window. Some of the option are width, height, fg, bg, side, row, column, text, onvalue, offvalue, variable, expand, fill, file, image, command and so on.

CHAPTER 4

WORK DIVISION

We divide the project into four modules. Each module is designed by one of us.

Work Division

S. No.	Student Name	Module Name
1.	Mohammad Abuzar	Sentiment Analysis Home page and User Interface
2.	Smaranika Datta	Sentiment Analysis About Developer page
3.	Subhadip Mondal	Sentiment Analysis Data Cleaning and User Interface
4.	Amrita Chaudri	Sentiment Analysis Model Creation

Table 4.1

Module-1

Module Name: -Sentiment Analysis Home page and User Interface.

Member Name: - Mohammad Abuzar

Description: -He created the program named as “home.py” and “PredictUserInterface.py”.

Widgets and option used are...

- Button-text, fg, bg, command
- Pack-side, fill, anchor, padx, pady
- Photoimage-file
- Label-text, fg, bg, text

Module-2

Module Name: -Sentiment Analysis About Developer page.

Member Name: - Smaranika Datta

Description: -She created the about developer page of program named as “about_us.py”.

Widgets and option used are...

- Button-text, fg, bg, command
- Frame- bg, cursor
- Pack-side, fill, anchor, padx, pady
- Photoimage-file
- Label-text, fg, bg, text

Module-3

Module Name: -Sentiment Analysis Data Cleaning and User Interface.

Member Name: - Subhadip Mondal

Description: -He created the program named as “MyModel.py” and “PredictUserInterface.py”.

Library, Widgets and option used are...

- Pandas
- Numpy
- Matplotlib
- Sklearn
- NLTK
- Title
- Frame-bg, cursor
- Pack-fill, expand, padx, pady
- Grid-row, column, rowspan, colspan, padx, pady, ipadx, ipady
- PhotoImage-file
- Button-text, padx, pady, ipadx, ipady, command
- Label-text, fg, bg, font

Module-4

Module Name: -Sentiment Analysis Model Creation.

Member Name: - Amrita Chaudri

Description: -She created program named as “MyModel.py”.

Library used are...

- Pandas
- Numpy
- Matplotlib
- Sklearn
- Machine Learning
- SVM

Gantt Chart

Each module is/will complete with in the given date as mention below.

	1 st -20 th Sept	20 th Sept -10 th Oct	10 th - 20 th Oct	20 th Oct – 1 st Nov	1 st - 5 th Nov	5 th – 10 th Nov
Research Synopsis						

GUI
Data Preprocessing
Creating Model
User Interface
Demo Video
Final Report



Table 4.2

Flow Chart

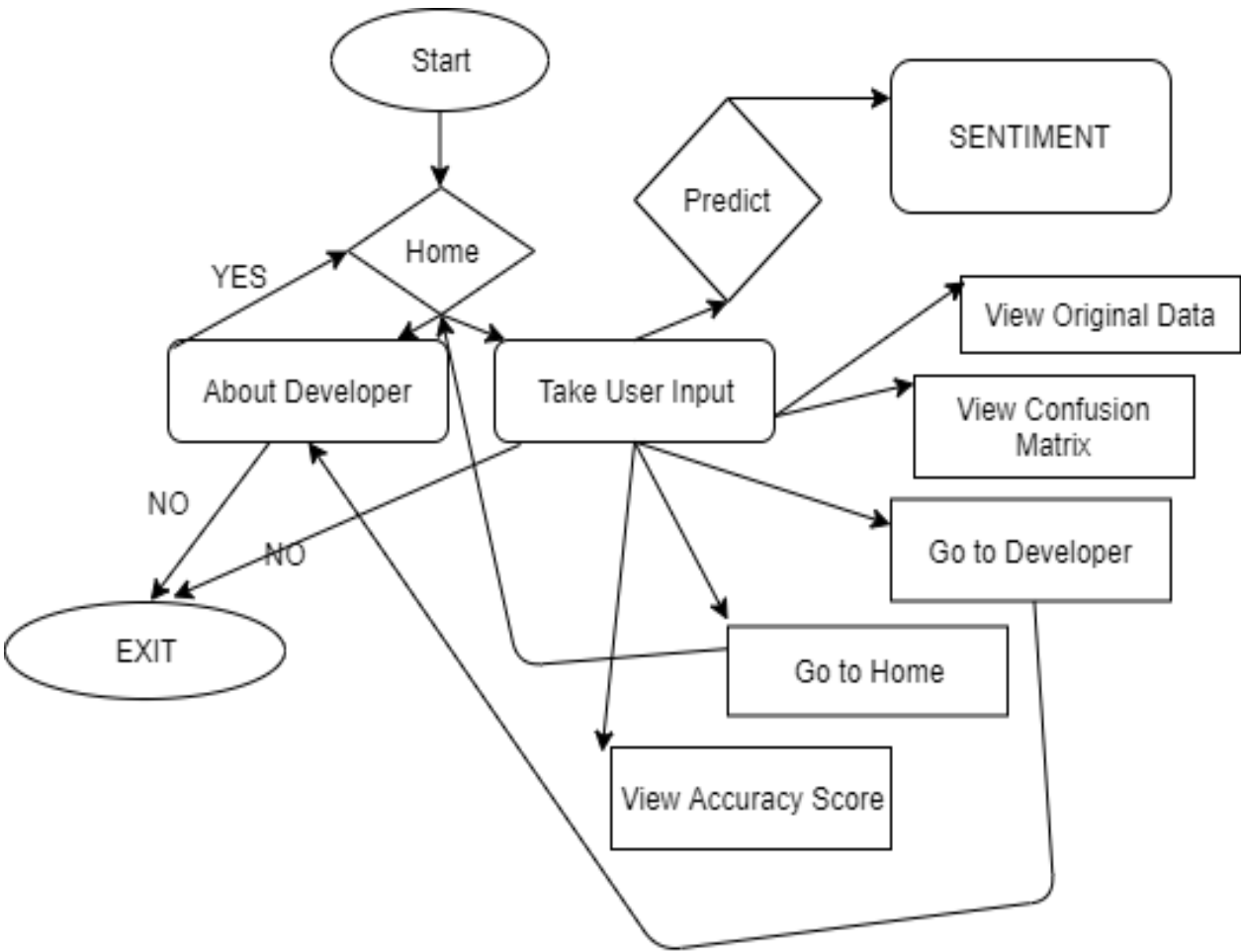


Table 4.3

CHAPTER 5

REFERENCES

We went through different website and learn those concept that will used in making this project.

Some of the website are...

- <https://www.google.com/>
- <https://www.python.org/>
- <https://numpy.org/>
- <https://pandas.pydata.org/>
- <https://matplotlib.org/>
- <https://www.nltk.org/>
- <https://scikit-learn.org/>
- We followed the text book named “Principles of Soft Computing” by S. N. Deepa and S. N. Sivanandam.