# CHAPTER 1

INTRODUCTION

**1.1 Motivation**

One of the best courses that we have in the Computer Science and Engineering is Mini Project. This is the course where a person can develop the skills in the interested area. Boys are always attracted to games. This becomes a motivation as well as our project arena to think. Moreover, data analyst on football are hired in Champion Leagues becomes a major motivation. As the football game is more familiar to us, it becomes very easy to get basic knowledge on that field.

**1.2 Objective**

The objective is to predict the winner of the match through data analytics. Winner is predicted based on the criteria such as individual skills, team statistics and twitter data. As individual skills are always not sufficient to predict the ultimate decision, we are getting the twitter data and analyzing it. The critics can use this analysis before the match and predict the result. The losing team can change the team formation or other parameters so that he can form better team to defeat the opponent.

# CHAPTER 2

LITERATURE SURVEY

Various data mining techniques have been employed to predict game results in recent years, such as artificial neural networks, decision trees, bayesian method, logistic regression, and Support Vector Machine(SVM) and fuzzy methods. A review of the related literature showed that the choice of technique to a large extent depends on the parameters of the system. Supervised learning uses labelled data to train a model. Two types of taxonomy exists, regression , an algorithm that is meant for interval labels and ,classification, an algorithm for class labels. SVM could provide a learning method that is used for both regression and classification. Here, a non-probabilistic binary linear classifier takes a set of input data and predicts, for each given input, which of the two possible classes comprises the input In this project, the hyper plane classified the data sets into their respective classes.

An SVM is largely characterized by the choice of its kernel and SVM's thus link the problems that are designed for with a large body of existing work on kernel based methods. Here, a linear kernel is used for the binary classification of input data set.

# CHAPTER 3

HARDWARE AND SOFTWARE SPECIFICATION

**3.1 Hardware requirements:**

For both Developer and User:

* + Processor : ARMv7 processor or higher
  + RAM : 512 MB or higher
  + Internal Memory : 1 GB or higher

**3.2**  **Software requirements:**

For Developer:

* + Development environment : Enthought canopy, XAMPP
  + Operating System : Windows
  + Programming language : python, R.

For user:

* Operating System : Windows, Linux.
* Web browser : Google Chrome, Mozilla, etc.

## **USERS OF THE SYSTEM**

Administrator

Viewers

**3.4 FUNCTIONAL FEATURES OF WEBSITE:**

* The administrator could add, delete and update player details
* The admin can add more leagues.
* The users can select two teams to predict the output.

**CHAPTER 4**

**SYSTEM ANALYSIS**

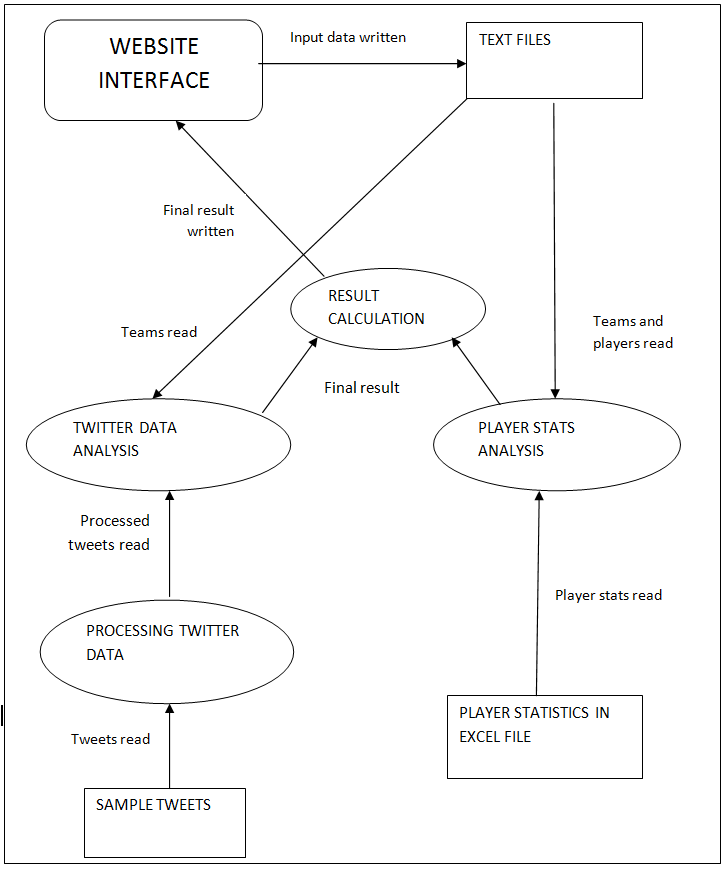


Figure 4.1: Block Diagram

The basic block diagram is represented above. This block diagram involves five modules.

Website Interface module

Player Statistics Calculation Module

Twitter data calculation Module

Total result Calculation module

Pre-processing twitter data module

The website interface module handles with the user giving the input. This input had to be transferred into the files from which data can be read.

Player Statistics module involves identifying the dominant team based on the player statistics.

Twitter data module handles with the processing of the twitter data to find out which team has the most support before the match.

Pre-processing twitter data involves removing unnecessary words and data from the twitter data and producing only the required words.

Total result calculation combines the output of player statistics and the output of twitter data calculation into a single output and is shown to the user in the website interface.

FILES:

Text files- This is where the data that should be given as the input is written.

Excel files- This is where the data of the player statistics are stored.

Sample tweets- This is where the tweets of teams are stored.

**CHAPTER 5**

**SYSTEM DESIGN**

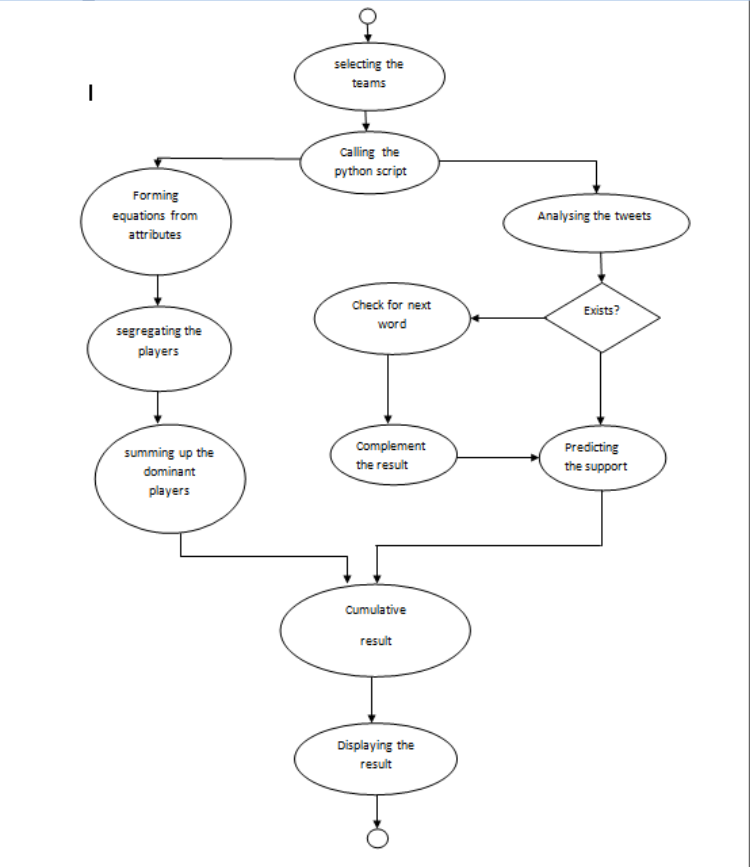
****

Figure 5.1: Flow Chart

The flow chart above describes the basic working of the system.

**STEP 1:** From the website interface the teams are selected.

**STEP 2**: From the website interface the players for both the teams are selected.

**STEP 3**: Calling the python script.

**STEP 4**: Forming the equations for player positions.

**STEP 5**: Classifying the players based on the result obtained from the equations using Support Vector Machine

**STEP 6**: summing up the results of the players of each team and identifying the dominant team.

**STEP 7**: Storing the result.

**STEP 8**: Analyzing the twitter data that is stored.

**STEP 9**: Predicting the support based on the positive and negative words in the tweets.

**STEP 10**: Storing the result.

**STEP 11**: Summing up the results in step 7 and step 10.

**STEP 12**: displaying the final output in the website.

There are text files where necessary information are written and read when required.

There are also excel files in which the player statistics data are stored.

There are also text files where the sample tweets are downloaded and stored.

**CHAPTER 6**

**IMPLEMENTATION**

Our mini project basically contains three parts. The first part involves analyzing the player statistics. The second part involves analyzing the twitter data. The third part involves developing a website.

**6.1 ANALYSING PLAYER STATISTICS:**  
LANGUAGE USED: Python

CLASSIFIER USED: SVM

For implementing the analysis of player statistics a few in-built packages are needed. The packages needed are scikit\_learn and openpyxl. After downloading the packages and installing coding process can begin.

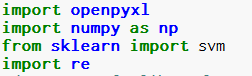


Figure 6.1: Required Packages

For analyzing the player statistics the different positions involved in football should be known. For every match each team contains 11 players represented using 11 positions. There will be three forward players, three mid-fielders, four defense players and a goal-keeper. So for every position a list is declared so that the data can be stored in the list.

The various attributes are:

Ball control Balance Acceleration

Dribbling Heading Stamina

Marking Finishing GK diving

Tackle Long shots GK Reactions

Interceptions FK accuracy

Crossing Penalties

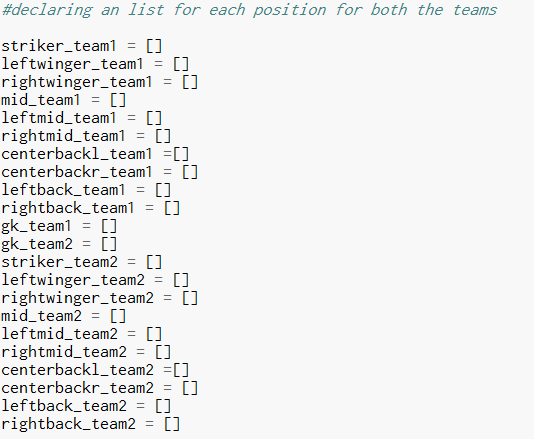


Figure 6.2: Declaring list

So the data is obtained from the excel sheet and stored in this list. There are various attributes for each player which is out of hundred. So basically all these values are obtained from the excel sheet and stored in the respective list.

There are various teams in any league. From the website which is developed, the two teams are selected. The names of the team are stored in a file named as team.txt . So the file is read and the two teams that have to be analyzed are stored in a list.

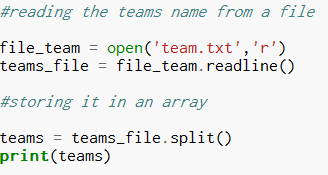


Figure 6.3: teams name read

After storing the names of the teams in a list the teams respective excel sheets where data is stored should be opened. In order to open this excel sheet we use the excel package.

So comparing the teams with the list we can open the excel sheets of the respective team.

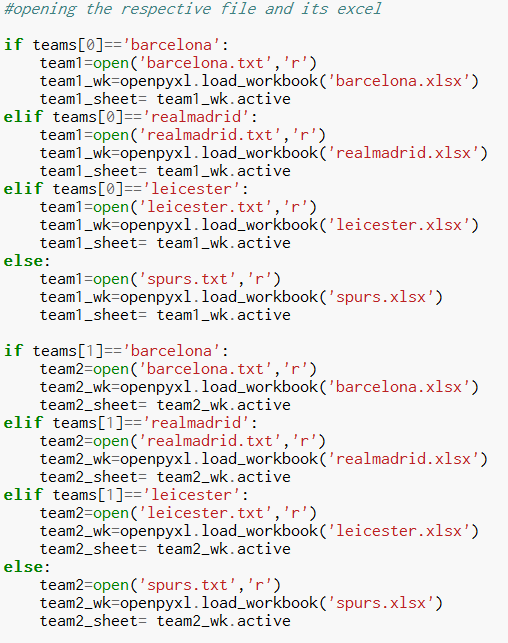


Figure 6.4: opening excel files

Also the players from each team who are going to play the match will be obtained from the website. The names of the players are stored in the respective text files. So while opening the excel files the respective text files are also opened. So on reading the text files and the player names are stored in an array.

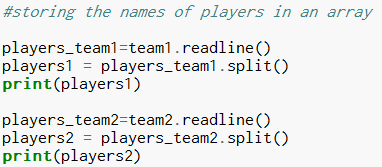


Figure 6.5: storing player names

We have to match the names of the players obtained and the names in the excel sheet to obtain the data for each player. Two arrays named store1 and store2 are declared for each team. These arrays will hold the row position of each player. By using this values stored in the array we can obtain the data from the excel sheet.

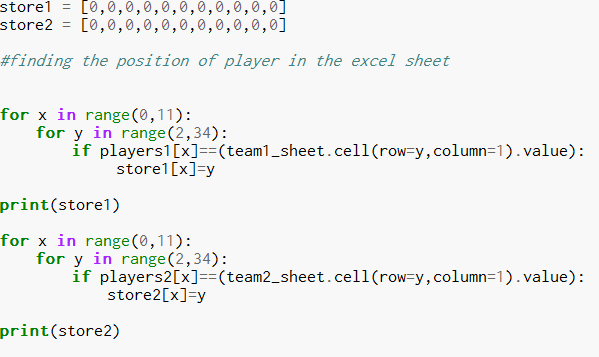


Figure 6.6: finding the position

After performing all the above operations, the data from the excel sheet be stored for each player position for both the teams. So the data is obtained from the excel sheet based on the row position stored and the necessary column values required.

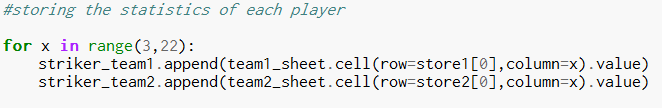


Figure 6.7: storing the statistics

Similarly the data should be obtained for all the positions for both the teams and stored in the list which was declared previously.

After obtaining the data and storing it in a list, these data have to be substituted in an equation and a two point values have to be generated. The equations should be generated for each position for eleven players. The same equation cannot be used for all the players because for different players the importance of attributes differs. For example for a striker defending skills are not required whereas for a defender these skills are very important. Based on the criteria different equations are generated for each player position. But we the value that is returned after substituting in the equation should be almost same (weight should be given same for every player).

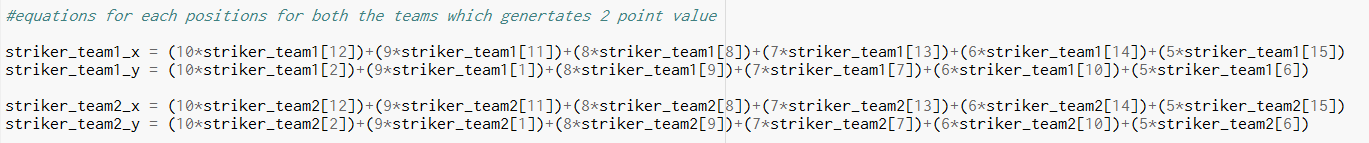


Figure 6.8: equations

These equations are generated for each position and a two point value is generated.

Now the two points are out of four thousand five hundred each. So the proportionate values for every point have to be identified and stored which can be used to classify the data.

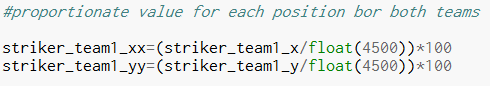


Figure 6.9: proportionate value

Now every position will have a two point value out of hundred each. These data can be used to classify the data. Classification process is performed for each category such as forward, mid, defense. So the two point values are stored in an array and classified using SVM. So each position for both teams will either have a value one or value zero.

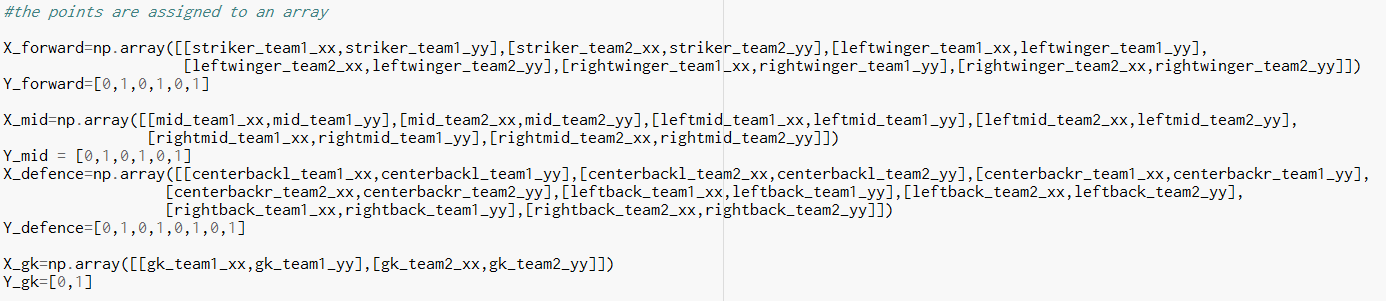


Figure 6.10: assigning the points

By this the players are categorized. Now using SVM we can classify the data.

For classification in-built functions which are available in SVM are used. This is why the packages were imported for SVM. The package that was used is scikit\_learn. Now fit() will classify the data and predict() will predict the class in which the data belongs.

The resultant values are stored in a integer value which can take either a zero or a one.

Based on this the more dominant player and less dominant player can be identified.

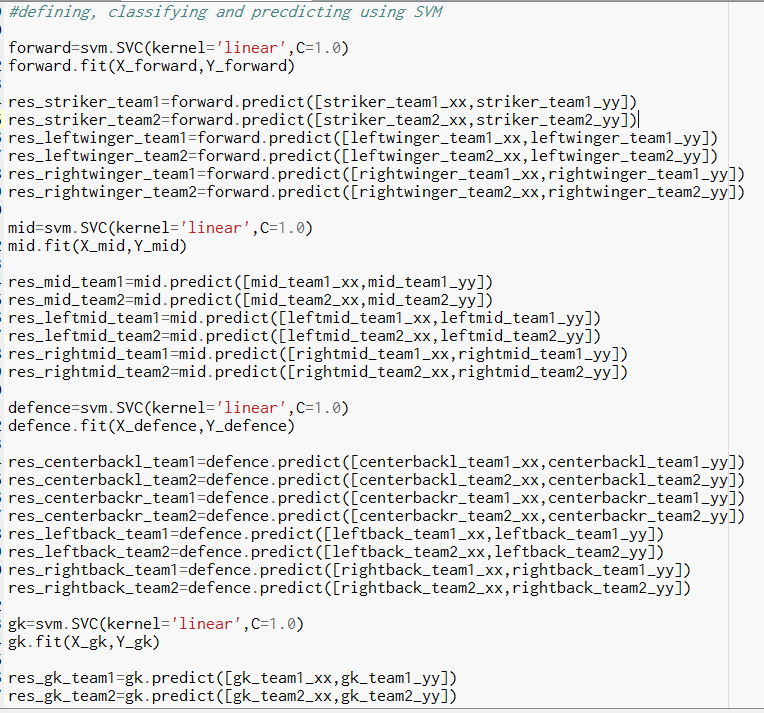


Figure 6.11: SVM classification

Sometimes a problem may arise such as two strikers from different teams belong to the same group and two wingers may belong to the other group. When this problem arises the two players of the same group have to be internally compared and re-allocated to the groups.

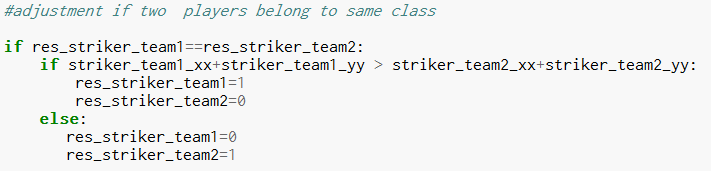


Figure 6.12: adjustment

The two point value is summed up and then the players with higher value is allocated in class one whereas the player with lower value is allocated in class zero.

After identifying where each player belongs a final team value has to be identified. For this to be performed a final equation is generated. The final classified value which is either zero or one is substituted in this equation and a value is obtained.

These values proportionate values have to be identified. So this will give the percentage of chance to win based on the player statistics. The sum of the values will be equal to hundred.

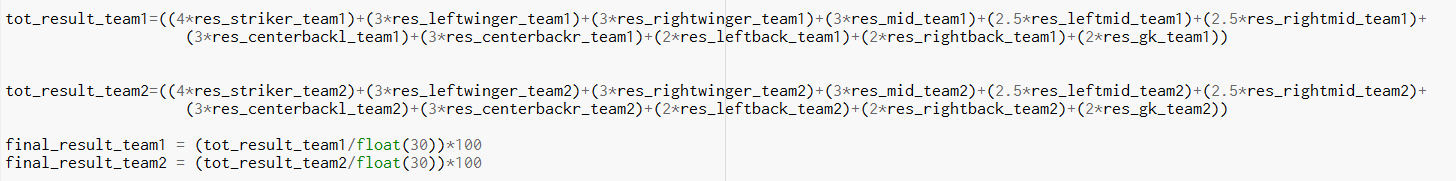


Figure 6.13: final result

**6.2 SENTIMENTAL ANALYSIS:**

How this analysis helps our project?

We cannot totally depend on the statistics. Assume we depend only on the statistics, then every time the team which has the higher points would win. That is not the case, we have in real time situations. It may be the situation when the key player in the team is out of form. This creates a major impact on the decision. To get that type of information, we are analyzing the twitter data. To make the prediction more accurate, we are using the twitter data.

**Downloading the Twitter Data:**

We are downloading the data through the R programming language. To download the twitter data, we are in need of the access code. That is the tokens that are generated during the API creation in twitter. By including these tokens in the R language, we can get the required data of that particular hashtag.

**Analyzing the twitter data:**

The data that we got is raw data, that includes user name, repeated words, lots of punctuations, their own colloquial languages. To extract the information that we needed, we are using some preprocessing techniques.

Preprocessing techniques are:

1. split tweet into words
2. replace two or more with two occurrences
3. strip punctuation
4. check if the word stats with an alphabet
5. ignore if it is a stop word
6. Convert www.\* or https?://\* to URL
7. Convert @username to AT\_USER
8. Remove additional white spaces
9. Replace #word with word

Once we completed these preprocesses, we will get the exact data. That is data is ready for analysis. We are preparing two files that is list of positive words and negative words. Once we get the original data, we can compare the data with the list of words in the files. And determining the fan’s support stands on which side.

Through these analysis, it contributes a part of the result. As our result is a cumulative answer, it becomes a part of it.

**6.3 WEBSITE DEVELOPMENT**

Before beginning the prediction analysis, there are certain things that are to be done. First, the two teams are to be selected, the twitter data are downloaded and then the starting 11,ie. the players who are in the squad for the match are selected for both the teams. For this, website is used as an user interface to do the above said things.

Basic html is used to develop the website and the created website is styled using Cascaded Style Sheets(CSS). For any website to interact with the user and the user's actions, there has to be a base for client-server communication. The browser will act as a client and there has to be a server to respond to the browsers needs.

**SERVER-SIDE SCRIPTING**

In this scenario, PHP: Hypertext Preprocessor(PHP) is used as a scripting language between the server and the client. That is, PHP acts as a server-side scripting language. Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for each user's (client's) request to the website. Server-side scripting is often used to provide a customized interface for the user. These scripts may assemble client characteristics for use in customizing the response based on those characteristics, the user's requirements, access rights, etc. Server-side scripting also enables the website owner to hide the source code that generates the interface, whereas with client-side scripting, the user has access to all the code received by the client. Server-side scripts are completely processed by the servers instead of clients. When clients request a page containing server-side scripts, the applicable server processes the scripts and returns an HTML page to the client. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks.

**Why PHP?**

* It is easy to learn and implement in a short period of time
* It is platform independent. ie. it can be run on any platforms like unix, linux, mac os, windows.
* It supports almost all the web servers available.
* It uses its own memory space and thus decreases the loading time and workload from the server. The processing speed is fast and web applications like Ecommerce, CRM, CMS and Forums are also developed faster by it.
* It has multiple layers of security to prevent threats and malicious

attacks.

**WEB SERVER**

The web-pages are only static(can be accessed only in the local machine),if not hosted in a web server. Also, PHP is a server-side scripting language, it requires a server to link webpages. In this project, we used a open source web server called XAMPP. This runs a server in a local machine.

**REASONS FOR CHOOSING XAMPP**

i) Hosting a website requires a domain. Domain space are not free, they are bought for a certain price. But XAMPP is open source and completely free.

ii) XAMPP runs an apache server in the local system, which acts as a server for the website.

iii) XAMPP is cross platform and can be used in any environment.

iv) XAMPP supports all the scripting languages.

**TEAM SELECTION:**

This page shows the teams that are included in this project. Two teams are to be selected. The page consists of basic html tags such as:

i) <div> tag:

The div tag used to represent related operations within a same class or

name. The specific tags are identified by the class or name.

ii) <form> tag:

The form tag is specialized tag used in html to process user inputs.

The <form > tag can contain other tags such as <input>, <table>, <button>,

etc.

**PAGE HIGHLIGHTS:**

One of the best highlights in these pages is that, we used video as a background for the web-pages. The web-pages are styled using CSS to enhance the look of the web-pages.

<video> tag:

This tag embeds a video to that html page. The video embedded is then styled to be the background of the page.

<!DOCTYPE html>

<html>

<head>

<title>TEAM SELECTION</title>

<meta charset="utf-8">

</head>

<body>

<div class="background">

<video id="background-video" preload="auto" autoplay="true" loop="loop" muted="muted" >

<source src="The Most INSANE Free Kicks Eve.mp4" type="video/mp4">

Video not supported

</video>

</div>

<form class="team\_select" action="asa.php" method="post">

<input type="checkbox" id="barca" name="teams[ ]" value="<team\_name>">

<input type="submit" name="submit" value="Submit"/>

</form>

</body>

</html>

**PLAYER SELECTION:**

The following PHP code retrieves the user selection from the form tags and then writes the input in appropriate .txt files. The contents of the files are then used by the python script for execution. The following code snippet is used to write the player selection for match between barcelona and realmadrid.

<?php

if(isset($\_POST['Submit'])){

if(!empty($\_POST['barca'])){

$fp\_barca = fopen("barcelona.txt", "w");

file\_put\_contents("barcelona.txt"," ");

fclose($fp\_barca);

$fp\_barca = fopen(("barcelona.txt", "a");

foreach ($\_POST['barca'] as $team\_barca) {

fwrite($fp\_barca, $team\_barca);

$team\_barca=$team\_barca." ";

}

fclose($fp\_barca);

}

if (!empty($\_POST['real'])) {

$fp\_real = fopen("realmadrid\_selected.txt","a");

foreach ($\_POST['real'] as $team\_real) {

fwrite($fp\_real, $team\_real);

$team\_real=$team\_real." ";

//fwrite($fp\_real,$space);

}

fclose($fp\_real);

}

header('Location:/call\_python.php');

}

?>

**CALLING PYTHON:**

One of the main aspects of scripting languages is that, another scripting languages can be called from a scripting language. The scenario here is that, the web-browser acts as a client and XAMPP acts as the server. But in this instance where the php has to call a python script, the PHP script acts as a client and the python script acts as a server. The PHP makes a request to the python script and the python script returns the output after its execution.

$command = escapeshellcmd('C:/studies/xampp/htdocs/

miniproj/pythonfinalchanged.py');

$output = shell\_exec($command);

The above php code stub is used to call python from the browser. The python script is then executed. The python script uses the files that we created during team selection and player selection.

**CHAPTER 7**

**SYSTEM TESTING**

**7.1 FUNCTIONALITY TESTING:**

**FUNCTIONAL TESTING FOR PLAYER STATISTICS:**

Test case 1:

INPUT:

Teams:

Barcelona, Real Madrid

Players Team1:

Lionel Messi, Luis Suarez, Munir, Dani Romera, Salva Chamorro, Neymar, Sandro, Camara, Arda Turan, Aleix Vidal, Ivan Rakitic.

Players Team2:

Karim Benzema, Jese, Borja Mayoral, Mariano, James Rodriguez, Isco, Mateo Kovacic, Matin Odegaard, Marcos Llorente, Javi Munoz, Lucas Vazquez.

OUTPUT:

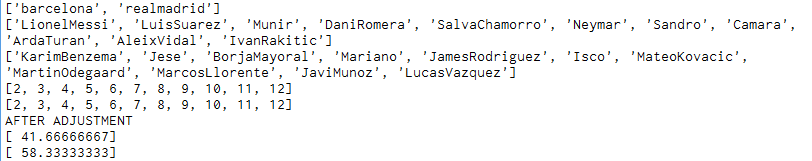


Figure 7.1: sample output-1

Test case 2:

INPUT:

Teams:

Leicester, Spurs

Players Team1:

Jamie Vardy, Shinji Okazaki, Leonarda Ulloa, N’GoloKante, Marc Albrington, Demarai Gray, Riyad Maharez, Daniel Amartey, Danny Drinkwater, Robert Huth, Christian Fuchs.

Players Team2:

Harry Kane, Nacer Chadli, Clinton N’jlie, Nathan Oduwa, Anthony Georgiou, Cy Goddard, Eric Dier, Christian Eriksen, Dele Alli, Nabil Bentaleb, Harry Winks.

OUTPUT:

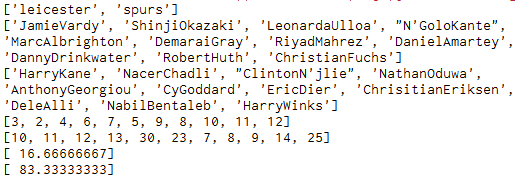


Figure 7.2: sample output-2

On analyzing the test cases, the required output is obtained. The main problem that arose was the player that is being written in the files should be without spaces. Every player name should be separated by a space but not within the player name. There was an error when the player name that is being written is misspelled. So the name will not be matched properly with the excel file and no data will be read. So it should be noted that the player name that is being written should be same as that of the players name in the excel file.

**FUNCTIONALITY TESTING FOR TWITTER DATA ANALYSIS:**

Test Case 1: Analysis between Real Madrid and Barcelona:

Sample Input:

Teams[0]=Barcelona

Teams[1]=Real Madrid

Sample Output:

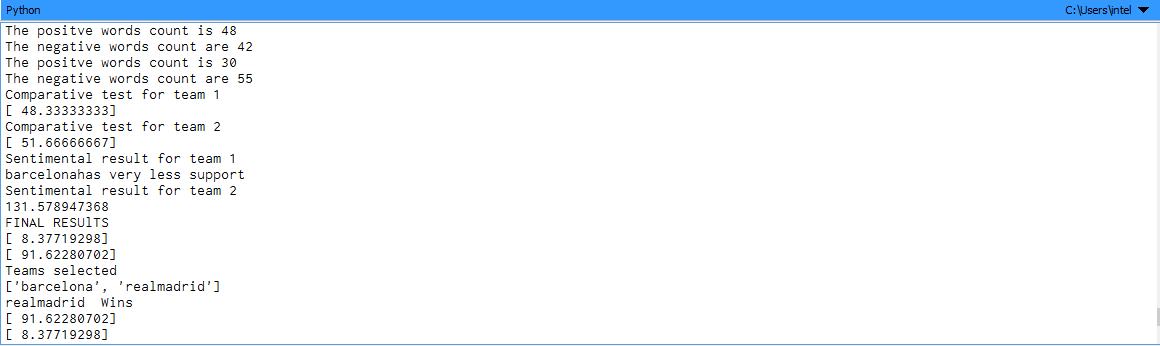


Figure 7.3: sample output-3

Sample Input:

Test Case 2: Analysis between Spurs and Leicester:

Teams[0]=Spurs

Teams[1]=Leicester

OUTPUT:

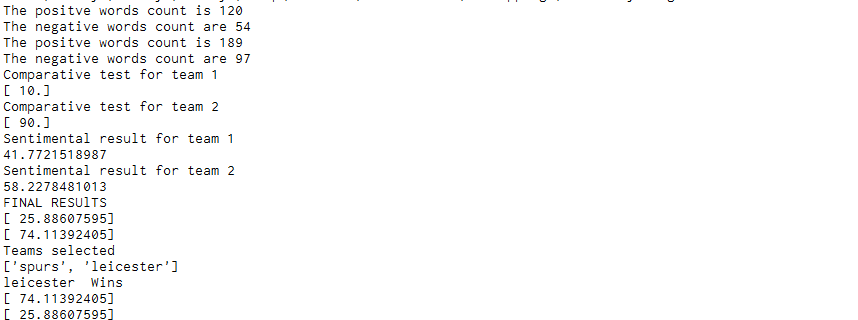
****

Figure 7.4: sample output-4

**Challenges that are solved:**

File pointers to open the tweets file that is we require two pointers to open two different files and processing them with same techniques.

When the negative tweets exceed the positive tweets, we are displaying their less support.

# 7.2 CODE TESTING:

Code testing is used to detect the unused header files, unused variables that is to detect the garbage memory to make it more efficient. There are separate header files to do these operations like Unittest, pytest, docs. These header files are working only for previous versions of python. The IDE (Enthought Canopy) , we are using is latest version and this itself provides the testing results of garbage memory.



Figure 7.5: Code testing

Through these facilities provided by Canopy, we produced the efficient code such

way that garbage values doesn’t acquire more memory.

# 

**CHAPTER 8**

**CONCLUSION**

The proposed application is very useful for the football fans. The application is significant in predicting the result. The proposed application is adequate to solve the curiosity of the fans and other betters. The players and the teams are updated and kept in track. The application can be further improved by adding more parameters. The application can be further enhanced if it is added with features like by giving sign in options to users, all teams can be added and the user can select any two teams and view the result, and classifying the teams according to the leagues. This application is a huge treat for the football fans, betters, coaches and other critics.

**APPENDIX**

1. **Code for Comparative test**

import openpyxl

import numpy as np

from sklearn import svm

import re

#import matplotlib.pyplot as plt

#from matplotlib import style

#style.use("ggplot")

#declaring an array for each position for both the teams

striker\_team1 = []

leftwinger\_team1 = []

rightwinger\_team1 = []

mid\_team1 = []

leftmid\_team1 = []

rightmid\_team1 = []

centerbackl\_team1 =[]

centerbackr\_team1 = []

leftback\_team1 = []

rightback\_team1 = []

gk\_team1 = []

gk\_team2 = []

striker\_team2 = []

leftwinger\_team2 = []

rightwinger\_team2 = []

mid\_team2 = []

leftmid\_team2 = []

rightmid\_team2 = []

centerbackl\_team2 =[]

centerbackr\_team2 = []

leftback\_team2 = []

rightback\_team2 = []

#reading the teams name from a file

file\_team = open('team.txt','r')

teams\_file = file\_team.readline()

#storing it in an array

teams = teams\_file.split()

print(teams)

#opening the respective file and its excel

if teams[0]=='barcelona':

team1=open('barcelona.txt','r')

team1\_wk=openpyxl.load\_workbook('barcelona.xlsx')

team1\_sheet= team1\_wk.active

elif teams[0]=='realmadrid':

team1=open('realmadrid.txt','r')

team1\_wk=openpyxl.load\_workbook('realmadrid.xlsx')

team1\_sheet= team1\_wk.active

elif teams[0]=='leicester':

team1=open('leicester.txt','r')

team1\_wk=openpyxl.load\_workbook('leicester.xlsx')

team1\_sheet= team1\_wk.active

else:

team1=open('spurs.txt','r')

team1\_wk=openpyxl.load\_workbook('spurs.xlsx')

team1\_sheet= team1\_wk.active

if teams[1]=='barcelona':

team2=open('barcelona.txt','r')

team2\_wk=openpyxl.load\_workbook('barcelona.xlsx')

team2\_sheet= team2\_wk.active

elif teams[1]=='realmadrid':

team2=open('realmadrid.txt','r')

team2\_wk=openpyxl.load\_workbook('realmadrid.xlsx')

team2\_sheet= team2\_wk.active

elif teams[1]=='leicester':

team2=open('leicester.txt','r')

team2\_wk=openpyxl.load\_workbook('leicester.xlsx')

team2\_sheet= team2\_wk.active

else:

team2=open('spurs.txt','r')

team2\_wk=openpyxl.load\_workbook('spurs.xlsx')

team2\_sheet= team2\_wk.active

#storing the names of players in an array

players\_team1=team1.readline()

players1 = players\_team1.split()

print(players1)

players\_team2=team2.readline()

players2 = players\_team2.split()

print(players2)

store1 = [0,0,0,0,0,0,0,0,0,0,0]

store2 = [0,0,0,0,0,0,0,0,0,0,0]

#finding the position of player in the excel sheet

for x in range(0,11):

for y in range(2,34):

if players1[x]==(team1\_sheet.cell(row=y,column=1).value):

store1[x]=y

print(store1)

for x in range(0,11):

for y in range(2,34):

if players2[x]==(team2\_sheet.cell(row=y,column=1).value):

store2[x]=y

print(store2)

#storing the statistics of each player

for x in range(3,22):

striker\_team1.append(team1\_sheet.cell(row=store1[0],column=x).value)

striker\_team2.append(team2\_sheet.cell(row=store2[0],column=x).value)

for x in range(3,22):

leftwinger\_team1.append(team1\_sheet.cell(row=store1[1],column=x).value)

leftwinger\_team2.append(team2\_sheet.cell(row=store2[1],column=x).value)

for x in range(3,22):

rightwinger\_team1.append(team1\_sheet.cell(row=store1[2],column=x).value)

rightwinger\_team2.append(team2\_sheet.cell(row=store2[2],column=x).value)

for x in range(3,22):

mid\_team1.append(team1\_sheet.cell(row=store1[3],column=x).value)

mid\_team2.append(team2\_sheet.cell(row=store2[3],column=x).value)

for x in range(3,22):

leftmid\_team1.append(team1\_sheet.cell(row=store1[4],column=x).value)

leftmid\_team2.append(team2\_sheet.cell(row=store2[4],column=x).value)

for x in range(3,22):

rightmid\_team1.append(team1\_sheet.cell(row=store1[5],column=x).value)

rightmid\_team2.append(team2\_sheet.cell(row=store2[5],column=x).value)

for x in range(3,22):

centerbackl\_team1.append(team1\_sheet.cell(row=store1[6],column=x).value)

centerbackl\_team2.append(team2\_sheet.cell(row=store2[6],column=x).value)

for x in range(3,22):

centerbackr\_team1.append(team1\_sheet.cell(row=store1[7],column=x).value)

centerbackr\_team2.append(team2\_sheet.cell(row=store2[7],column=x).value)

for x in range(3,22):

leftback\_team1.append(team1\_sheet.cell(row=store1[8],column=x).value)

leftback\_team2.append(team2\_sheet.cell(row=store2[8],column=x).value)

for x in range(3,22):

rightback\_team1.append(team1\_sheet.cell(row=store1[9],column=x).value)

rightback\_team2.append(team2\_sheet.cell(row=store2[9],column=x).value)

for x in range(3,22):

gk\_team1.append(team1\_sheet.cell(row=store1[10],column=x).value)

gk\_team2.append(team2\_sheet.cell(row=store1[10],column=x).value)

#equations for each positions for both the teams which genertates 2 point value

striker\_team1\_x = (10\*striker\_team1[12])+(9\*striker\_team1[11])+(8\*striker\_team1[8])+(7\*striker\_team1[13])+(6\*striker\_team1[14])+(5\*striker\_team1[15])

striker\_team1\_y = (10\*striker\_team1[2])+(9\*striker\_team1[1])+(8\*striker\_team1[9])+(7\*striker\_team1[7])+(6\*striker\_team1[10])+(5\*striker\_team1[6])

striker\_team2\_x = (10\*striker\_team2[12])+(9\*striker\_team2[11])+(8\*striker\_team2[8])+(7\*striker\_team2[13])+(6\*striker\_team2[14])+(5\*striker\_team2[15])

striker\_team2\_y = (10\*striker\_team2[2])+(9\*striker\_team2[1])+(8\*striker\_team2[9])+(7\*striker\_team2[7])+(6\*striker\_team2[10])+(5\*striker\_team2[6])

leftwinger\_team1\_x = (10\*leftwinger\_team1[6])+(9\*leftwinger\_team1[11])+(8\*leftwinger\_team1[1])+(7\*leftwinger\_team1[8])+(6\*leftwinger\_team1[14])+(5\*leftwinger\_team1[15])

leftwinger\_team1\_y = (10\*leftwinger\_team1[10])+(9\*leftwinger\_team1[12])+(8\*leftwinger\_team1[2])+(7\*leftwinger\_team1[7])+(6\*leftwinger\_team1[9])+(5\*leftwinger\_team1[13])

leftwinger\_team2\_x = (10\*leftwinger\_team2[6])+(9\*leftwinger\_team2[11])+(8\*leftwinger\_team2[1])+(7\*leftwinger\_team2[8])+(6\*leftwinger\_team2[14])+(5\*leftwinger\_team2[15])

leftwinger\_team2\_y = (10\*leftwinger\_team2[10])+(9\*leftwinger\_team2[12])+(8\*leftwinger\_team2[2])+(7\*leftwinger\_team2[7])+(6\*leftwinger\_team2[9])+(5\*leftwinger\_team2[13])

rightwinger\_team1\_x = (10\*rightwinger\_team1[6])+(9\*rightwinger\_team1[11])+(8\*rightwinger\_team1[1])+(7\*rightwinger\_team1[8])+(6\*rightwinger\_team1[14])+(5\*rightwinger\_team1[15])

rightwinger\_team1\_y = (10\*rightwinger\_team1[10])+(9\*rightwinger\_team1[12])+(8\*rightwinger\_team1[2])+(7\*rightwinger\_team1[7])+(6\*rightwinger\_team1[9])+(5\*rightwinger\_team1[13])

rightwinger\_team2\_x = (10\*rightwinger\_team2[6])+(9\*rightwinger\_team2[11])+(8\*rightwinger\_team2[1])+(7\*rightwinger\_team2[8])+(6\*rightwinger\_team2[14])+(5\*rightwinger\_team2[15])

rightwinger\_team2\_y = (10\*rightwinger\_team2[10])+(9\*rightwinger\_team2[12])+(8\*rightwinger\_team2[2])+(7\*rightwinger\_team2[7])+(6\*rightwinger\_team2[9])+(5\*rightwinger\_team2[13])

mid\_team1\_x = (10\*mid\_team1[7])+(9\*mid\_team1[9])+(8\*mid\_team1[1])+(7\*mid\_team1[8])+(6\*mid\_team1[12])+(5\*mid\_team1[6])

mid\_team1\_y = (10\*mid\_team1[2])+(9\*mid\_team1[13])+(8\*mid\_team1[14])+(7\*mid\_team1[10])+(6\*mid\_team1[11])+(5\*mid\_team1[15])

mid\_team2\_x = (10\*mid\_team2[7])+(9\*mid\_team2[9])+(8\*mid\_team2[1])+(7\*mid\_team2[8])+(6\*mid\_team2[12])+(5\*mid\_team2[6])

mid\_team2\_y = (10\*mid\_team2[2])+(9\*mid\_team2[13])+(8\*mid\_team2[14])+(7\*mid\_team2[10])+(6\*mid\_team2[11])+(5\*mid\_team2[15])

leftmid\_team1\_x = (10\*leftmid\_team1[7])+(9\*leftmid\_team1[9])+(8\*leftmid\_team1[13])+(7\*leftmid\_team1[1])+(6\*leftmid\_team1[12])+(5\*leftmid\_team1[10])

leftmid\_team1\_y = (10\*leftmid\_team1[6])+(9\*leftmid\_team1[8])+(8\*leftmid\_team1[2])+(7\*leftmid\_team1[14])+(6\*leftmid\_team1[11])+(5\*leftmid\_team1[15])

leftmid\_team2\_x = (10\*leftmid\_team2[7])+(9\*leftmid\_team2[9])+(8\*leftmid\_team2[13])+(7\*leftmid\_team2[1])+(6\*leftmid\_team2[12])+(5\*leftmid\_team2[10])

leftmid\_team2\_y = (10\*leftmid\_team2[6])+(9\*leftmid\_team2[8])+(8\*leftmid\_team2[2])+(7\*leftmid\_team2[14])+(6\*leftmid\_team2[11])+(5\*leftmid\_team2[15])

rightmid\_team1\_x = (10\*rightmid\_team1[7])+(9\*rightmid\_team1[9])+(8\*rightmid\_team1[13])+(7\*rightmid\_team1[1])+(6\*rightmid\_team1[12])+(5\*rightmid\_team1[10])

rightmid\_team1\_y = (10\*rightmid\_team1[6])+(9\*rightmid\_team1[8])+(8\*rightmid\_team1[2])+(7\*rightmid\_team1[14])+(6\*rightmid\_team1[11])+(5\*rightmid\_team1[15])

rightmid\_team2\_x = (10\*rightmid\_team2[7])+(9\*rightmid\_team2[9])+(8\*rightmid\_team2[13])+(7\*rightmid\_team2[1])+(6\*rightmid\_team2[12])+(5\*rightmid\_team2[10])

rightmid\_team2\_y = (10\*rightmid\_team2[6])+(9\*rightmid\_team2[8])+(8\*rightmid\_team2[2])+(7\*rightmid\_team2[14])+(6\*rightmid\_team2[11])+(5\*rightmid\_team2[15])

centerbackl\_team1\_x = (10\*centerbackl\_team1[4])+(9\*centerbackl\_team1[3])+(8\*centerbackl\_team1[13])+(7\*centerbackl\_team1[7])+(6\*centerbackl\_team1[15])+(5\*centerbackl\_team1[10])

centerbackl\_team1\_y = (10\*centerbackl\_team1[5])+(9\*centerbackl\_team1[11])+(8\*centerbackl\_team1[9])+(7\*centerbackl\_team1[1])+(6\*centerbackl\_team1[6])+(5\*centerbackl\_team1[8])

centerbackl\_team2\_x = (10\*centerbackl\_team2[4])+(9\*centerbackl\_team2[3])+(8\*centerbackl\_team2[13])+(7\*centerbackl\_team2[7])+(6\*centerbackl\_team2[15])+(5\*centerbackl\_team2[10])

centerbackl\_team2\_y = (10\*centerbackl\_team2[5])+(9\*centerbackl\_team2[11])+(8\*centerbackl\_team2[9])+(7\*centerbackl\_team2[1])+(6\*centerbackl\_team2[6])+(5\*centerbackl\_team2[8])

centerbackr\_team1\_x = (10\*centerbackr\_team1[4])+(9\*centerbackr\_team1[3])+(8\*centerbackr\_team1[13])+(7\*centerbackr\_team1[7])+(6\*centerbackr\_team1[15])+(5\*centerbackr\_team1[10])

centerbackr\_team1\_y = (10\*centerbackr\_team1[5])+(9\*centerbackr\_team1[11])+(8\*centerbackr\_team1[9])+(7\*centerbackr\_team1[1])+(6\*centerbackr\_team1[6])+(5\*centerbackr\_team1[8])

centerbackr\_team2\_x = (10\*centerbackr\_team2[4])+(9\*centerbackr\_team2[3])+(8\*centerbackr\_team2[13])+(7\*centerbackr\_team2[7])+(6\*centerbackr\_team2[15])+(5\*centerbackr\_team2[10])

centerbackr\_team2\_y = (10\*centerbackr\_team2[5])+(9\*centerbackr\_team2[11])+(8\*centerbackr\_team2[9])+(7\*centerbackr\_team2[1])+(6\*centerbackr\_team2[6])+(5\*centerbackr\_team2[8])

leftback\_team1\_x = (10\*leftback\_team1[4])+(9\*leftback\_team1[3])+(8\*leftback\_team1[13])+(7\*leftback\_team1[7])+(6\*leftback\_team1[1])+(5\*leftback\_team1[10])

leftback\_team1\_y = (10\*leftback\_team1[5])+(9\*leftback\_team1[6])+(8\*leftback\_team1[9])+(7\*leftback\_team1[11])+(6\*leftback\_team1[15])+(5\*leftback\_team1[8])

leftback\_team2\_x = (10\*leftback\_team2[4])+(9\*leftback\_team2[3])+(8\*leftback\_team2[13])+(7\*leftback\_team2[7])+(6\*leftback\_team2[1])+(5\*leftback\_team2[10])

leftback\_team2\_y = (10\*leftback\_team2[5])+(9\*leftback\_team2[6])+(8\*leftback\_team2[9])+(7\*leftback\_team2[11])+(6\*leftback\_team2[15])+(5\*leftback\_team2[8])

rightback\_team1\_x = (10\*rightback\_team1[4])+(9\*rightback\_team1[3])+(8\*rightback\_team1[13])+(7\*rightback\_team1[7])+(6\*rightback\_team1[1])+(5\*rightback\_team1[10])

rightback\_team1\_y = (10\*rightback\_team1[5])+(9\*rightback\_team1[6])+(8\*rightback\_team1[9])+(7\*rightback\_team1[11])+(6\*rightback\_team1[15])+(5\*rightback\_team1[8])

rightback\_team2\_x = (10\*rightback\_team2[4])+(9\*rightback\_team2[3])+(8\*rightback\_team2[13])+(7\*rightback\_team2[7])+(6\*rightback\_team2[1])+(5\*rightback\_team2[10])

rightback\_team2\_y = (10\*rightback\_team2[5])+(9\*rightback\_team2[6])+(8\*rightback\_team2[9])+(7\*rightback\_team2[11])+(6\*rightback\_team2[15])+(5\*rightback\_team2[8])

gk\_team1\_x = (gk\_team1[17])

gk\_team1\_y = (gk\_team1[16]+gk\_team1[18])/2

gk\_team2\_x = (gk\_team2[17])

gk\_team2\_y = (gk\_team2[16]+gk\_team2[18])/2

#proportionate value for each position bor both teams

striker\_team1\_xx=(striker\_team1\_x/float(4500))\*100

striker\_team1\_yy=(striker\_team1\_y/float(4500))\*100

striker\_team2\_xx=(striker\_team2\_x/float(4500))\*100

striker\_team2\_yy=(striker\_team2\_y/float(4500))\*100

leftwinger\_team1\_xx=(leftwinger\_team1\_x/float(4500))\*100

leftwinger\_team1\_yy=(leftwinger\_team1\_y/float(4500))\*100

leftwinger\_team2\_xx=(leftwinger\_team2\_x/float(4500))\*100

leftwinger\_team2\_yy=(leftwinger\_team2\_y/float(4500))\*100

rightwinger\_team1\_xx=(rightwinger\_team1\_x/float(4500))\*100

rightwinger\_team1\_yy=(rightwinger\_team1\_y/float(4500))\*100

rightwinger\_team2\_xx=(rightwinger\_team2\_x/float(4500))\*100

rightwinger\_team2\_yy=(rightwinger\_team2\_y/float(4500))\*100

mid\_team1\_xx=(mid\_team1\_x/float(4500))\*100

mid\_team1\_yy=(mid\_team1\_y/float(4500))\*100

mid\_team2\_xx=(mid\_team2\_x/float(4500))\*100

mid\_team2\_yy=(mid\_team2\_y/float(4500))\*100

leftmid\_team1\_xx=(leftmid\_team1\_x/float(4500))\*100

leftmid\_team1\_yy=(leftmid\_team1\_y/float(4500))\*100

leftmid\_team2\_xx=(leftmid\_team2\_x/float(4500))\*100

leftmid\_team2\_yy=(leftmid\_team2\_y/float(4500))\*100

rightmid\_team1\_xx=(rightmid\_team1\_x/float(4500))\*100

rightmid\_team1\_yy=(rightmid\_team1\_y/float(4500))\*100

rightmid\_team2\_xx=(rightmid\_team2\_x/float(4500))\*100

rightmid\_team2\_yy=(rightmid\_team2\_y/float(4500))\*100

centerbackl\_team1\_xx=(centerbackl\_team1\_x/float(4500))\*100

centerbackl\_team1\_yy=(centerbackl\_team1\_y/float(4500))\*100

centerbackl\_team2\_xx=(centerbackl\_team2\_x/float(4500))\*100

centerbackl\_team2\_yy=(centerbackl\_team2\_y/float(4500))\*100

centerbackr\_team1\_xx=(centerbackr\_team1\_x/float(4500))\*100

centerbackr\_team1\_yy=(centerbackr\_team1\_y/float(4500))\*100

centerbackr\_team2\_xx=(centerbackr\_team2\_x/float(4500))\*100

centerbackr\_team2\_yy=(centerbackr\_team2\_y/float(4500))\*100

leftback\_team1\_xx=(leftback\_team1\_x/float(4500))\*100

leftback\_team1\_yy=(leftback\_team1\_y/float(4500))\*100

leftback\_team2\_xx=(leftback\_team2\_x/float(4500))\*100

leftback\_team2\_yy=(leftback\_team2\_y/float(4500))\*100

rightback\_team1\_xx=(rightback\_team1\_x/float(4500))\*100

rightback\_team1\_yy=(rightback\_team1\_y/float(4500))\*100

rightback\_team2\_xx=(rightback\_team2\_x/float(4500))\*100

rightback\_team2\_yy=(rightback\_team2\_y/float(4500))\*100

gk\_team1\_xx=gk\_team1\_x

gk\_team1\_yy=gk\_team1\_y

gk\_team2\_xx=gk\_team2\_x

gk\_team2\_yy=gk\_team2\_y

#the points are assigned to an array

X\_forward=np.array([[striker\_team1\_xx,striker\_team1\_yy],[striker\_team2\_xx,striker\_team2\_yy],[leftwinger\_team1\_xx,leftwinger\_team1\_yy],

[leftwinger\_team2\_xx,leftwinger\_team2\_yy],[rightwinger\_team1\_xx,rightwinger\_team1\_yy],[rightwinger\_team2\_xx,rightwinger\_team2\_yy]])

Y\_forward=[0,1,0,1,0,1]

X\_mid=np.array([[mid\_team1\_xx,mid\_team1\_yy],[mid\_team2\_xx,mid\_team2\_yy],[leftmid\_team1\_xx,leftmid\_team1\_yy],[leftmid\_team2\_xx,leftmid\_team2\_yy],

[rightmid\_team1\_xx,rightmid\_team1\_yy],[rightmid\_team2\_xx,rightmid\_team2\_yy]])

Y\_mid = [0,1,0,1,0,1]

X\_defence=np.array([[centerbackl\_team1\_xx,centerbackl\_team1\_yy],[centerbackl\_team2\_xx,centerbackl\_team2\_yy],[centerbackr\_team1\_xx,centerbackr\_team1\_yy],

[centerbackr\_team2\_xx,centerbackr\_team2\_yy],[leftback\_team1\_xx,leftback\_team1\_yy],[leftback\_team2\_xx,leftback\_team2\_yy],

[rightback\_team1\_xx,rightback\_team1\_yy],[rightback\_team2\_xx,rightback\_team2\_yy]])

Y\_defence=[0,1,0,1,0,1,0,1]

X\_gk=np.array([[gk\_team1\_xx,gk\_team1\_yy],[gk\_team2\_xx,gk\_team2\_yy]])

Y\_gk=[0,1]

#defining, classifying and precdicting using SVM

forward=svm.SVC(kernel='linear',C=1.0)

forward.fit(X\_forward,Y\_forward)

res\_striker\_team1=forward.predict([striker\_team1\_xx,striker\_team1\_yy])

res\_striker\_team2=forward.predict([striker\_team2\_xx,striker\_team2\_yy])

res\_leftwinger\_team1=forward.predict([leftwinger\_team1\_xx,leftwinger\_team1\_yy])

res\_leftwinger\_team2=forward.predict([leftwinger\_team2\_xx,leftwinger\_team2\_yy])

res\_rightwinger\_team1=forward.predict([rightwinger\_team1\_xx,rightwinger\_team1\_yy])

res\_rightwinger\_team2=forward.predict([rightwinger\_team2\_xx,rightwinger\_team2\_yy])

mid=svm.SVC(kernel='linear',C=1.0)

mid.fit(X\_mid,Y\_mid)

res\_mid\_team1=mid.predict([mid\_team1\_xx,mid\_team1\_yy])

res\_mid\_team2=mid.predict([mid\_team2\_xx,mid\_team2\_yy])

res\_leftmid\_team1=mid.predict([leftmid\_team1\_xx,leftmid\_team1\_yy])

res\_leftmid\_team2=mid.predict([leftmid\_team2\_xx,leftmid\_team2\_yy])

res\_rightmid\_team1=mid.predict([rightmid\_team1\_xx,rightmid\_team1\_yy])

res\_rightmid\_team2=mid.predict([rightmid\_team2\_xx,rightmid\_team2\_yy])

defence=svm.SVC(kernel='linear',C=1.0)

defence.fit(X\_defence,Y\_defence)

res\_centerbackl\_team1=defence.predict([centerbackl\_team1\_xx,centerbackl\_team1\_yy])

res\_centerbackl\_team2=defence.predict([centerbackl\_team2\_xx,centerbackl\_team2\_yy])

res\_centerbackr\_team1=defence.predict([centerbackr\_team1\_xx,centerbackr\_team1\_yy])

res\_centerbackr\_team2=defence.predict([centerbackr\_team2\_xx,centerbackr\_team2\_yy])

res\_leftback\_team1=defence.predict([leftback\_team1\_xx,leftback\_team1\_yy])

res\_leftback\_team2=defence.predict([leftback\_team2\_xx,leftback\_team2\_yy])

res\_rightback\_team1=defence.predict([rightback\_team1\_xx,rightback\_team1\_yy])

res\_rightback\_team2=defence.predict([rightback\_team2\_xx,rightback\_team2\_yy])

gk=svm.SVC(kernel='linear',C=1.0)

gk.fit(X\_gk,Y\_gk)

res\_gk\_team1=gk.predict([gk\_team1\_xx,gk\_team1\_yy])

res\_gk\_team2=gk.predict([gk\_team2\_xx,gk\_team2\_yy])

print "TEAM1 STRIKER",res\_striker\_team1

print "TEAM2 STRIKER",res\_striker\_team2

print "TEAM1 LEFT WINGER",res\_leftwinger\_team1

print "TEAM2 LEFT WINGER",res\_leftwinger\_team2

print "TEAM1 RIGHT WINGER",res\_rightwinger\_team1

print "TEAM2 RIGHT WINGER",res\_rightwinger\_team2

print "TEAM1 CENTER MID",res\_mid\_team1

print "TEAM2 CENTER MID",res\_mid\_team2

print "TEAM1 LEFT MID",res\_leftmid\_team1

print "TEAM2 LEFT MID",res\_leftmid\_team2

print "TEAM1 RIGHT MID",res\_rightmid\_team1

print "TEAM2 RIGHT MID",res\_rightmid\_team2

print "TEAM1 CENTERAL LEFT DEFENDER",res\_centerbackl\_team1

print "TEAM2 CENTRAL LEFT DEFFENDER",res\_centerbackl\_team2

print "TEAM1 CENTERAL RIGHT DEFENDER",res\_centerbackr\_team1

print "TEAM2 CENTRAL RIGHT DEFFENDER",res\_centerbackr\_team2

print "TEAM1 LEFT DEFENDER",res\_leftback\_team1

print "TEAM2 LEFT DEFENDER",res\_leftback\_team2

print "TEAM1 RIGHT DEFENDER",res\_rightback\_team1

print "TEAM2 RIGHT DEFENDER",res\_rightback\_team2

print "TEAM1 GOAL KEEPER",res\_gk\_team1

print "TEAM2 GOAL KEEPER",res\_gk\_team2

#adjustment if two players belong to same class

if res\_striker\_team1==res\_striker\_team2:

if striker\_team1\_xx+striker\_team1\_yy > striker\_team2\_xx+striker\_team2\_yy:

res\_striker\_team1=1

res\_striker\_team2=0

else:

res\_striker\_team1=0

res\_striker\_team2=1

if res\_leftwinger\_team1==res\_leftwinger\_team2:

if leftwinger\_team1\_xx+leftwinger\_team1\_yy > leftwinger\_team2\_xx+leftwinger\_team2\_yy:

res\_leftwinger\_team1=1

res\_leftwinger\_team2=0

else:

res\_leftwinger\_team1=0

res\_leftwinger\_team2=1

if res\_rightwinger\_team1==res\_rightwinger\_team2:

if rightwinger\_team1\_xx+rightwinger\_team1\_yy > rightwinger\_team2\_xx+rightwinger\_team2\_yy:

res\_rightwinger\_team1=1

res\_rightwinger\_team2=0

else:

res\_rightwinger\_team1=0

res\_rightwinger\_team2=1

if res\_mid\_team1==res\_mid\_team2:

if mid\_team1\_xx+mid\_team1\_yy > mid\_team2\_xx+mid\_team2\_yy:

res\_mid\_team1=1

res\_mid\_team2=0

else:

res\_mid\_team1=0

res\_mid\_team2=1

if res\_leftmid\_team1==res\_leftmid\_team2:

if leftmid\_team1\_xx+leftmid\_team1\_yy > leftmid\_team2\_xx+leftmid\_team2\_yy:

res\_leftmid\_team1=1

res\_leftmid\_team2=0

else:

res\_leftmid\_team1=0

res\_leftmid\_team2=1

if res\_rightmid\_team1==res\_rightmid\_team2:

if rightmid\_team1\_xx+rightmid\_team1\_yy > rightmid\_team2\_xx+rightmid\_team2\_yy:

res\_rightmid\_team1=1

res\_rightmid\_team2=0

else:

res\_rightmid\_team1=0

res\_rightmid\_team2=1

if res\_centerbackl\_team1==res\_centerbackl\_team2:

if centerbackl\_team1\_xx+centerbackl\_team1\_yy > centerbackl\_team2\_xx+centerbackl\_team2\_yy:

res\_centerbackl\_team1=1

res\_centerbackl\_team2=0

else:

res\_centerbackl\_team1=0

res\_centerbackl\_team2=1

if res\_centerbackr\_team1==res\_centerbackr\_team2:

if centerbackr\_team1\_xx+centerbackr\_team1\_yy > centerbackr\_team2\_xx+centerbackr\_team2\_yy:

res\_centerbackr\_team1=1

res\_centerbackr\_team2=0

else:

res\_centerbackr\_team1=0

res\_centerbackr\_team2=1

if res\_leftback\_team1==res\_leftback\_team2:

if leftback\_team1\_xx+leftback\_team1\_yy > leftback\_team2\_xx+leftback\_team2\_yy:

res\_leftback\_team1=1

res\_leftback\_team2=0

else:

res\_leftback\_team1=0

res\_leftback\_team2=1

if res\_rightback\_team1==res\_rightback\_team2:

if rightback\_team1\_xx+rightback\_team1\_yy > rightback\_team2\_xx+rightback\_team2\_yy:

res\_rightback\_team1=1

res\_rightback\_team2=0

else:

res\_rightback\_team1=0

res\_rightback\_team2=1

print("AFTER ADJUSTMENT")

print "TEAM1 STRIKER",res\_striker\_team1

print "TEAM2 STRIKER",res\_striker\_team2

print "TEAM1 LEFT WINGER",res\_leftwinger\_team1

print "TEAM2 LEFT WINGER",res\_leftwinger\_team2

print "TEAM1 RIGHT WINGER",res\_rightwinger\_team1

print "TEAM2 RIGHT WINGER",res\_rightwinger\_team2

print "TEAM1 CENTER MID",res\_mid\_team1

print "TEAM2 CENTER MID",res\_mid\_team2

print "TEAM1 LEFT MID",res\_leftmid\_team1

print "TEAM2 LEFT MID",res\_leftmid\_team2

print "TEAM1 RIGHT MID",res\_rightmid\_team1

print "TEAM2 RIGHT MID",res\_rightmid\_team2

print "TEAM1 CENTERAL LEFT DEFENDER",res\_centerbackl\_team1

print "TEAM2 CENTRAL LEFT DEFFENDER",res\_centerbackl\_team2

print "TEAM1 CENTERAL RIGHT DEFENDER",res\_centerbackr\_team1

print "TEAM2 CENTRAL RIGHT DEFFENDER",res\_centerbackr\_team2

print "TEAM1 LEFT DEFENDER",res\_leftback\_team1

print "TEAM2 LEFT DEFENDER",res\_leftback\_team2

print "TEAM1 RIGHT DEFENDER",res\_rightback\_team1

print "TEAM2 RIGHT DEFENDER",res\_rightback\_team2

print "TEAM1 GOAL KEEPER",res\_gk\_team1

print "TEAM2 GOAL KEEPER",res\_gk\_team2

tot\_result\_team1=((4\*res\_striker\_team1)+(3\*res\_leftwinger\_team1)+(3\*res\_rightwinger\_team1)+(3\*res\_mid\_team1)+(2.5\*res\_leftmid\_team1)+(2.5\*res\_rightmid\_team1)+

(3\*res\_centerbackl\_team1)+(3\*res\_centerbackr\_team1)+(2\*res\_leftback\_team1)+(2\*res\_rightback\_team1)+(2\*res\_gk\_team1))

tot\_result\_team2=((4\*res\_striker\_team2)+(3\*res\_leftwinger\_team2)+(3\*res\_rightwinger\_team2)+(3\*res\_mid\_team2)+(2.5\*res\_leftmid\_team2)+(2.5\*res\_rightmid\_team2)+

(3\*res\_centerbackl\_team2)+(3\*res\_centerbackr\_team2)+(2\*res\_leftback\_team2)+(2\*res\_rightback\_team2)+(2\*res\_gk\_team2))

final\_result\_team1 = (tot\_result\_team1/float(30))\*100

final\_result\_team2 = (tot\_result\_team2/float(30))\*100

print(final\_result\_team1)

print(final\_result\_team2)

1. **Sentimental Analysis:**
   1. **Downloading the twitter data:**

library(twitteR)

library(httr)

requestURL = "https://api.twitter.com/oauth/request\_token"

accessURL = "https://api.twitter.com/oauth/access\_token"

authURL = "https://api.twitter.com/oauth/authorize"

consumerKey = "Plwiyg7BNKzxRUB59AUmCEDY3"

consumerSecret = "urZh7hhCBy4zgAKu0LfZibReVcfQXXKTFxjeIe1OiyqiLWFStC"

accessToken = "282920491-wyg0wcCC0hHfKeQordNeZfZL3eN1HI3cfTFzTkPz"

accessSecret = " ISAiOXeJ5EWpSEs4kjet9TcUo3YUVQp54TlsDarelBnis"

setup\_twitter\_oauth(consumerKey,consumerSecret,accessToken,accessSecret)

r\_stats <- searchTwitter("#leicester", n=500)

#head( r\_stats )

r\_stats.df <- do.call(rbind, lapply(r\_stats, as.data.frame))

write.csv(r\_stats.df, "E:/Mini Project/R Language/sampletweet70.csv")

* 1. **Sentimental Analysis code:**

#initialize preportions

preportions = []

def replaceTwoOrMore(s):

pattern = re.compile(r"(.)\1{1,}", re.DOTALL)

return pattern.sub(r"\1\1", s)

#end

def getwords(wordsfile):

preportions = []

preportions.append('AT\_USER')

preportions.append('URL')

fp = open(wordsfile, 'r')

line = fp.readline()

while line:

word = line.strip()

preportions.append(word)

line = fp.readline()

fp.close()

return preportions

#end

def getdata(t):

tweetp = []

words = t.split()

for w in words:

w = replaceTwoOrMore(w)

w = w.strip('\'"?,.')

val = re.search(r"^[a-zA-Z][a-zA-Z0-9]\*$", w)

if(w in preportions or val is None):

continue

else:

tweetp.append(w.lower())

return tweetp

#end

def tdata(t):

t = t.lower()

t = re.sub('((www\.[^\s]+)|(https?://[^\s]+))','URL',t)

t = re.sub('@[^\s]+','AT\_USER',t)

t = re.sub('[\s]+', ' ', t)

t = re.sub(r'#([^\s]+)', r'\1', t)

t = t.strip('\'"')

return t

countpos=0

countneg=0

posw = [line.rstrip('\n') for line in open('E:/Mini Project/poswords.txt')]

negw = [line.rstrip('\n') for line in open('E:/Mini Project/negwords.txt')]

st = open('E:/Mini Project/R Language/preportions.txt', 'r')

preportions = getwords('E:/Mini Project/R Language/preportions.txt')

if (teams[0]=="barcelona" and teams[1]=="realmadrid"):

choice=1;

fp = open('E:/Mini Project/R Language/barca.txt', 'r')

line = fp.readline()

fp1 = open('E:/Mini Project/R Language/real.txt', 'r')

line1 = fp1.readline()

elif(teams[0]=="spurs" and teams[1]=="leicester"):

choice=2;

fp = open('E:/Mini Project/R Language/spurstweet.txt', 'r')

line = fp.readline()

fp1 = open('E:/Mini Project/R Language/leicestertweet.txt', 'r')

line1 = fp1.readline()

elif(teams[0]=="barcelona" and teams[1]=="leicester"):

fp = open('E:/Mini Project/R Language/barca.txt', 'r')

line = fp.readline()

fp1 = open('E:/Mini Project/R Language/leicestertweet.txt', 'r')

line1 = fp1.readline()

elif(teams[0]=="realmadrid" and teams[1]=="spurs"):

fp = open('E:/Mini Project/R Language/real.txt', 'r')

line = fp.readline()

fp1 = open('E:/Mini Project/R Language/spurstweet.txt', 'r')

line1 = fp1.readline()

elif(teams[0]=="realmadrid" and teams[1]=="leicester"):

fp = open('E:/Mini Project/R Language/real.txt', 'r')

line = fp.readline()

fp1 = open('E:/Mini Project/R Language/leicestertweet.txt', 'r')

line1 = fp1.readline()

else:

fp = open('E:/Mini Project/R Language/barca.txt', 'r')

line = fp.readline()

fp1 = open('E:/Mini Project/R Language/spurstweet.txt', 'r')

line1 = fp1.readline()

while line:

trimdata = tdata(line)

tweetp = getdata(trimdata)

print tweetp

line = fp.readline()

for i in tweetp:

for j in posw:

if i==j:

countpos+=1

for i in tweetp:

for j in negw:

if i==j:

countneg+=1

#end loop

countpost=0

countnegt=0

while line1:

trimdata = tdata(line1)

tweetp = getdata(trimdata)

print tweetp

line1 = fp1.readline()

for i in tweetp:

for j in posw:

if i==j:

countpost+=1

for i in tweetp:

for j in negw:

if i==j:

countnegt+=1

#end loop

fp.close()

fp1.close()

print "The positve words count is %d" %countpos

print "The negative words count are %d" %countneg

print "The positve words count is %d" %countpost

print "The negative words count are %d" %countnegt

t\_barca=countpos-countneg

t\_real=countpost-countnegt

ft\_barca=(t\_barca/float(t\_barca+t\_real))\*100

ft\_real=(t\_real/float(t\_barca+t\_real))\*100

if(ft\_real<0):

print teams[1]+"has very less support"

print "Comparative test for team 1"

print(final\_result\_team1)

print "Comparative test for team 2"

print(final\_result\_team2)

print "Sentimental result for team 1"

if(ft\_barca < 0):

print teams[0]+"has very less support"

else:

print(ft\_barca)

print "Sentimental result for team 2"

if(ft\_real<0):

print teams[1]+"has very less support"

else:

print(ft\_real)

**3. Code for Website:**

**Homepage:**

<!DOCTYPE html>

<html>

<head>

<title>Prediction Analysis</title>

<link rel="stylesheet" href="landing\_styles.css">

</head>

<body >

<div class="background">

<video id="background-video" preload="auto" autoplay="true" loop="loop" muted="muted" >

<source src="joined-all.mp4" type="video/mp4">

Video not supported

</video>

</div>

<form class="content" action="new\_table.html" method="post">

<h1>Football Prediction</h1>

<button onclick="teams.php" >Enter</button>

</form>

</body>

</html>

Team selection page:

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

.background{

position: fixed;

z-index: -1000;

width: 100%;

height: 100%;

overflow: hidden;

top: 0;

left: 0;

}

#background-video{

position: absolute;

top: 0;

left: 0;

min-height: 100%;

min-width: 100%;

}

.team\_select{

position: absolute;

width: 100%;

min-height: 100%;

z-index: 1000;

background-color: rgba(0,0,0,0.7);

}

.team\_select{

text-align: center;

font-size: 25px;

text-transform: uppercase;

font-weight: 100;

color: #fff;

line-height: 2;

padding-top: 5%;

margin-bottom: 10px;

}

.teams:hover{

width: 250px;

height: 190px;

}

</style>

<meta charset="utf-8">

</head>

<body>

<div class="background">

<video id="background-video" preload="auto" autoplay="true" loop="loop" muted="muted" >

<source src="The Most INSANE Free Kicks Eve.mp4" type="video/mp4">

Video not supported

</video>

</div>

<form class="team\_select" action="asa.php" method="post">

<table align="center" cellspacing="1" width="500px" height="300px">

<tr>

<td>

<img class="teams" src="barca.jpg" ></br>

<input type="checkbox" id="barca" name="teams[]" value="barcelona">

</td>

<td>

<img class="teams" src="real\_madrid.jpg" ></br>

<input type="checkbox" id="real\_madrid" name="teams[]" value="realmadrid">

</td>

</tr>

<tr>

<td>

<img class="teams" src="leicester\_city.png" ></br>

<input type="checkbox" id="leicester" name="teams[]" value="leicester">

</td>

<td>

<img class="teams" src="tottenham.jpg" ></br>

<input type="checkbox" id="tottenham" name="teams[]" value="spurs">

</td>

</tr>

</table>

<input type="submit" name="submit" value="Submit"/>

</form>

</body>

</html>

**WRITING THE SELECTED TEAMS INTO TXT FILE**:

<?php

if(isset($\_POST['submit'])){

if (!empty($\_POST['teams'])) {

$fp = fopen("team.txt","w");

file\_put\_contents("team.txt", " ");

fclose($fp);

$fp = fopen("team.txt","a");

foreach ($\_POST['teams'] as $selected) {

$selected = $selected." ";

fwrite($fp, $selected);

}

fclose($fp);

$countbr=0;

$countbs=0;

$countbl=0;

$countrs=0;

$countrc=0;

$countsc=0;

foreach ($\_POST['teams'] as $direct) {

if($direct=='spurs' || $direct=='leicester'){

$countsc++;

}

elseif($direct=='barcelona' || $direct=='realmadrid'){

$countbr++;

}

elseif($direct=='barcelona' || $direct=='spurs'){

$countbs++;

}

if($direct=='barcelona' || $direct=='leicester'){

$countbl++;

}

if($direct=='realmadrid' || $direct=='leicester'){

$countrc++;

}

if($direct=='realmadrid' || $direct=='spurs'){

$countrs++;

}

else

header('Location:/miniproj/new\_table.html');

}

if($countbr>=2){

header('Location:/miniproj/barca\_vs\_real.php');

}

elseif ($countbs>=2) {

header('Location:/miniproj/barca\_vs\_spurs.php');

}

elseif ($countbl>=2) {

header('Location:/miniproj/barca\_vs\_city.php');

}

elseif ($countrc>=2) {

header('Location:/miniproj/real\_vs\_city.php');

}

elseif ($countrs>=2) {

header('Location:/miniproj/real\_vs\_spurs.php');

}

else{

if($countsc>=2){

header('Location:/miniproj/spurs\_vs\_city.php');

}

else

header('Location:/miniproj/new\_table.html');

}

}

}

?>

**Player selection:**

1) barcelona versus real

<!DOCTYPE html>

<html>

<head>

<title>Player select </title>

<style type="text/css">

.background{

position: fixed;

z-index: -1000;

width: 100%;

height: 100%;

overflow: hidden;

top: 0;

left: 0;

}

#background-video{

position: absolute;

top: 0;

left: 0;

min-height: 100%;

min-width: 100%;

}

.team\_select{

position: absolute;

width: 100%;

min-height: 100%;

z-index: 1000;

background-color: rgba(0,0,0,0.7);

}

.team\_select{

color: #fff;

}

</style>

</head>

<body>

<?php

if(isset($\_POST['Submit'])){

if(!empty($\_POST['barca'])){

$fp\_barca = fopen("barcelona.txt", "w");

file\_put\_contents("barcelona.txt", " ");

fclose($fp\_barca);

$fp\_barca = fopen("barcelona.txt", "a");

foreach ($\_POST['barca'] as $team\_barca) {

$team\_barca=$team\_barca." ";

fwrite($fp\_barca, $team\_barca);

//fwrite($fp\_barca,$space);

}

fclose($fp\_barca);

}

if (!empty($\_POST['real'])) {

$fp\_real = fopen("realmadrid.txt","w");

file\_put\_contents("realmadrid.txt", " ");

fclose($fp\_real);

$fp\_real = fopen("realmadrid.txt", "a");

foreach ($\_POST['real'] as $team\_real) {

$team\_real=$team\_real." ";

fwrite($fp\_real, $team\_real);

//fwrite($fp\_real,$space);

}

fclose($fp\_real);

}

header('Location:/miniproj/call\_python.php');

}

?>

<div class="background">

<video id="background-video" preload="auto" autoplay="true" loop="loop" muted="muted" >

<source src="Cristiano Ronaldo.mp4" type="video/mp4">

Video not supported

</video>

</div>

<form class="team\_select" action="barca\_vs\_real.php" method="post">

<table align="left" cellspacing="50" >

<tr>

<td>

<input type="checkbox" name="barca[]" value="LionelMessi">Lionel Messi(CF)(94)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="LuisSuarez">Luis Suarez(ST)(90)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Munir">Munir(ST)(74)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="DaniRomera">Dani Romera(ST)(62)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="SalvaChamorro">Salva Chamorro(ST)(62)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Neymar">Neymar(LW)(90)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Sandro">Sandro(LW)(72)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Camara">Camara(LW)(66)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="ArdaTuran">Arda Turan(RM)(83)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="AleixVidal ">Aleix Vidal(RM)(79)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="IvanRakitic">Ivan Rakitic(RCM)(85)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Rafinha">Rafinha(CM)(79)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="SergiRoberto">Sergi Roberto(CDM)(78)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Gerard">Gerard(CM)(68)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="SergiSamper">Sergi Samper(CDM)(70)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="WilfridJauresKaptoum">Wilfrid Jaures Kaptoum(CM)(62)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="RobertGoncalves">Robert Goncalves(CAM)(62)

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="SergioBusquets">Sergio Busquets(CDM)(86)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Iniesta">Iniesta(LCM)(88)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Douglas">Douglas(RB)(72)

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="DaniAlves">Dani Alves(RB)(84)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Pique">Pique(RCB)(85)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="BorjaLopez">Borja Lopez(CB)(67)

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="JeremyMathieu">Jeremy Mathieu(CB)(82)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="ThomasVermaelen">Thomas Vermaelen(CB)(80)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="MarcBartra">Marc Bartra(CB)(81)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="JavierMascharano">Javier Mascharano(LCB)(83)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="JordiAlba">Jordi Alba(LB)(84)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Adriano">Adriano(LB)(77)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="MoiDelgado">Moi Delgado(LB)(61)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="JordiMasip">Jordi Masip(GK)(71)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="ClaudioBravo">Claudio Bravo(GK)(83)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="barca[]" value="Marc-AndreterStegen">Marc-Andre ter Stegen(GK)(82)

</td>

</tr>

</table>

<table align="right" cellspacing="50" >

<tr>

<td>

Karim Benzema(ST)(88)<input type="checkbox" name="real[]" value="KarimBenzema">

</td>

</tr>

<tr>

<td>

Jese(ST)(80)<input type="checkbox" name="real[]" value="Jese">

</td>

</tr>

<tr>

<td>

Borja Mayoral(ST)(66)<input type="checkbox" name="real[]" value="BorjaMayoral">

</td>

</tr>

<tr>

<td>

Mariano(ST)(64)<input type="checkbox" name="real[]" value="Mariano">

</td>

</tr>

<tr>

<td>

James Rodriguez(CAM)(87)<input type="checkbox" name="real[]" value="JamesRodriguez">

</td>

</tr>

<tr>

<td>

Isco(CAM)(85)<input type="checkbox" name="real[]" value="Isco">

</td>

</tr>

<tr>

<td>

Mateo Kovacic(CM)(78)<input type="checkbox" name="real[]" value="MateoKovacic">

</td>

</tr>

<tr>

<td>

Martin Odegaard(CAM)(69)<input type="checkbox" name="real[]" value="MartinOdegaard">

</td>

</tr>

<tr>

<td>

Marcos Llorente(CM)(68)<input type="checkbox" name="real[]" value="MarcosLlorente">

</td>

</tr>

<tr>

<td>

Javi Munoz(CAM)(65)<input type="checkbox" name="real[]" value="JaviMunoz">

</td>

</tr>

<tr>

<td>

Lucas Vazquez(RM)(79)<input type="checkbox" name="real[]" value="LucasVazquez">

</td>

</tr>

<tr>

<td>

Gareth Bale(RM)(87)<input type="checkbox" name="real[]" value="GarethBale">

</td>

</tr>

<tr>

<td>

Luka Modric(RCM)(87)<input type="checkbox" name="real[]" value="LukaModric">

</td>

</tr>

<tr>

<td>

Toni Kroos(LCM)(87)<input type="checkbox" name="real[]" value="ToniKroos">

</td>

</tr>

<tr>

<td>

Lazo(LM)(65)<input type="checkbox" name="real[]" value="Lazo">

</td>

</tr>

<tr>

<td>

Cristiano Ronaldo(LW)(93)<input type="checkbox" name="real[]" value="CristianoRonaldo">

</td>

</tr>

<tr>

<td>

Enzo Fernandez(LM)(63)<input type="checkbox" name="real[]" value="EnzoFernandez">

</td>

</tr>

<tr>

<td>

Casemiro(CDM)(79)<input type="checkbox" name="real[]" value="Casemiro">

</td>

</tr>

<tr>

<td>

Lucas Torro(CDM)(67)<input type="checkbox" name="real[]" value="LucasTorro">

</td>

</tr>

<tr>

<td>

Carvajal(RB)(82)<input type="checkbox" name="real[]" value="Carvajal">

</td>

</tr>

<tr>

<td>

Pepe(RCB)(84)<input type="checkbox" name="real[]" value="Pepe">

</td>

</tr>

<tr>

<td>

Alvaro Tejero(RB)(61)<input type="checkbox" name="real[]" value="AlvaroTejero">

</td>

</tr>

<tr>

<td>

Danilo(RB)(80)<input type="checkbox" name="real[]" value="Danilo">

</td>

</tr>

<tr>

<td>

Raphael Varane(CB)(82)<input type="checkbox" name="real[]" value="RaphaelVarane">

</td>

</tr>

<tr>

<td>

Philipp Lienhart(CB)(64)<input type="checkbox" name="real[]" value="PhilippLienhart">

</td>

</tr>

<tr>

<td>

Sergio Ramos(LCB)(87)<input type="checkbox" name="real[]" value="SergioRamos">

</td>

</tr>

<tr>

<td>

Marcelo(LB)(83)<input type="checkbox" name="real[]" value="Marcelo">

</td>

</tr>

<tr>

<td>

Nacho Fernandez(LCB)(77)<input type="checkbox" name="real[]" value="NachoFernandez">

</td>

</tr>

<tr>

<td>

Abner(LB)(63)<input type="checkbox" name="real[]" value="Abner">

</td>

</tr>

<tr>

<td>

Arbeloa(LB)(65)<input type="checkbox" name="real[]" value="Arbeloa">

</td>

</tr>

<tr>

<td>

Kiko Casilla(GK)(80)<input type="checkbox" name="real[]" value="KikoCasilla">

</td>

</tr>

<tr>

<td>

Ruben Yanez(GK)(65)<input type="checkbox" name="real[]" value="RubenYanez">

</td>

</tr>

<tr>

<td>

Keylor Navas(GK)(82)<input type="checkbox" name="real[]" value="KeylorNavas">

</td>

</tr>

</table>

<input type="submit" name="Submit" />

</form>

</body>

</html>

**2) spurs versus leicester:**

<!DOCTYPE html>

<html>

<head>

<title>Player select </title>

<style type="text/css">

.background{

position: fixed;

z-index: -1000;

width: 100%;

height: 100%;

overflow: hidden;

top: 0;

left: 0;

}

#background-video{

position: absolute;

top: 0;

left: 0;

min-height: 100%;

min-width: 100%;

}

.team\_select{

position: absolute;

width: 100%;

min-height: 100%;

z-index: 1000;

background-color: rgba(0,0,0,0.7);

}

.team\_select{

color: #fff;

}

</style>

</head>

<body>

<?php

if(isset($\_POST['Submit'])){

if(!empty($\_POST['spurs'])){

$fp\_spurs = fopen("spurs.txt", "w");

file\_put\_contents("spurs.txt", " ");

fclose($fp\_spurs);

$fp\_spurs = fopen("spurs.txt", "as");

foreach ($\_POST['spurs'] as $team\_spurs) {

$team\_spurs=$team\_spurs." ";

fwrite($fp\_spurs, $team\_spurs);

//fwrite($fp\_barca,$space);

}

fclose($fp\_spurs);

}

if (!empty($\_POST['leicester'])) {

$fp\_city = fopen("leicester.txt","w");

file\_put\_contents("leicester.txt", " ");

fclose($fp\_city);

$fp\_city = fopen("leicester.txt","a");

foreach ($\_POST['leicester'] as $team\_city) {

$team\_city=$team\_city." ";

fwrite($fp\_city, $team\_city);

//fwrite($fp\_real,$space);

}

fclose($fp\_city);

}

header('Location:/miniproj/call\_python.php');

}

?>

<div class="background">

<video id="background-video" preload="auto" autoplay="true" loop="loop" muted="muted" >

<source src="Cristiano Ronaldo.mp4" type="video/mp4">

Video not supported

</video>

</div>

<form class="team\_select" action="spurs\_vs\_city.php" method="post">

<table align="left" cellspacing="50" >

<tr>

<td>

<input type="checkbox" name="spurs[]" value="HarryKane">Harry Kane(ST)(85)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="NacerChadli">Nacer Chadli(LM)(79)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="ClintonN'jlie">Clinton N'jlie(LW)(72)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="NathanOduwa">Nathan Oduwa(LM)(69)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="AnthonyGeorgiou">Anthony Georgiou(LM)(60)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="CyGoddard">Cy Goddard(LM)(62)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="EricDier">Eric Dier(CDM)(80)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="ChrisitianEriksen">Chrisitian Eriksen(CDM)(80)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="DeleAlli">Dele Alli(CAM)(82)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="NabilBentaleb">Nabil Bentaleb(CM)(78)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="HarryWinks">Harry Winks(RM)(67)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="TomCarroll">Tom Carroll(CM)(75)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="RyanMason">Ryan Mason(CAM)(77)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="LukeAmos">Luke Amos(CM)(60)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="EmmanuelSonupe">Emmanuel Sonupe(CM)(61)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="HarryWinks">Harry Winks(RM)(67)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="WilliamMiller">William Miller(RM)(63)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="JoshOnomah">Josh Onomah(RM)(74)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="HeungMinSon">Heung Min Son(RM)(79)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="HarryWinks">Harry Winks(RM)(67)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="BenDavies">Ben Davies(LB)(79)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="KyleWalker-Peters">Kyle Walker- Peters(LB)(62)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="JanVetonghen">Jan Vetonghen(LCB)(82)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="DannyRose">Danny Rose(LB)(80)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="CameronCarte-Vickers">Cameron Carte-Vickers(CB)(65)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="ChristisnMaghoma">Christisn Maghoma(CB)(63)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="KevinWimmer">Kevin Wimmer(CB)(80)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="KyleWalker">Kyle Walker(RB)(79)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="TobyAlderweireld">Toby Alderweireld(RCB)(83)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="KieranTrippier">Kieran Trippier(RB)(77)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="LukeMcGee">Luke McGee(GK)(66)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="HugoLloris">Hugo Lloris(GK)(85)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="MochaelVorm">Mochael Vorm(GK)(80)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="ClaudioBravo">Claudio Bravo(GK)(83)

</td>

</tr>

<tr>

<td>

<input type="checkbox" name="spurs[]" value="MarcAndreterStegen">Marc Andre-ter Stegen(GK)(82)

</td>

</tr>

</table>

<table align="right" cellspacing="50" >

<tr>

<td>

Jamie Vardy(ST)(79)<input type="checkbox" name="leicester[]" value="JamieVardy">

</td>

</tr>

<tr>

<td>

Shinji Okazaki(ST)(76)<input type="checkbox" name="leicester[]" value="ShinjiOkazaki">

</td>

</tr>

<tr>

<td>

Leonarda Ulloa(ST)(73)<input type="checkbox" name="leicester[]" value="LeonardaUlloa">

</td>

</tr>

<tr>

<td>

N'Golo Kante(LCM)(83)<input type="checkbox" name="leicester[]" value="N'GoloKante">

</td>

</tr>

<tr>

<td>

Marc Albrighton(LM)(77)<input type="checkbox" name="leicester[]" value="MarcAlbrighton">

</td>

</tr>

<tr>

<td>

Demarai Gray(LM)(72)<input type="checkbox" name="leicester[]" value="DemaraiGray">

</td>

</tr>

<tr>

<td>

Riyad Mahrez(RM)(85)<input type="checkbox" name="leicester[]" value="RiyadMahrez">

</td>

</tr>

<tr>

<td>

Daniel Amartey(CM)(75)<input type="checkbox" name="leicester[]" value="DanielAmartey ">

</td>

</tr>

<tr>

<td>

Danny Drinkwater(RCM)(78)<input type="checkbox" name="leicester[]" value="DannyDrinkwater">

</td>

</tr>

<tr>

<td>

Robert Huth(LCB)(78)<input type="checkbox" name="leicester[]" value="RobertHuth">

</td>

</tr>

<tr>

<td>

Christian Fuchs(LB)(76)<input type="checkbox" name="leicester[]" value="ChristianFuchs">

</td>

</tr>

<tr>

<td>

Danny Simpson(RB)(72)<input type="checkbox" name="leicester[]" value="DannySimpson">

</td>

</tr>

<tr>

<td>

Wes Morgan(RCB)(78)<input type="checkbox" name="leicester[]" value="WesMorgan">

</td>

</tr>

<tr>

<td>

Kasper Schmeichel(GK)(78)<input type="checkbox" name="leicester[]" value="KasperSchmeichel">

</td>

</tr>

<tr>

<td>

Jeffrey Schlupp(LB)(75)<input type="checkbox" name="leicester[]" value="JeffreySchlupp">

</td>

</tr>

<tr>

<td>

RonRobert Zieler(GK)(82)<input type="checkbox" name="leicester[]" value="RonRobertZieler">

</td>

</tr>

<tr>

<td>

Luis Hernanez(CB)(79)<input type="checkbox" name="leicester[]" value="LuisHernanez">

</td>

</tr>

<tr>

<td>

Andy King(CM)(73)<input type="checkbox" name="leicester[]" value="AndyKing">

</td>

</tr>

<tr>

<td>

Nampalys Mendy(CDM)(84)<input type="checkbox" name="leicester[]" value="NampalysMendy">

</td>

</tr>

<tr>

<td>

Carvajal(RB)(82)<input type="checkbox" name="leicester[]" value="Carvajal">

</td>

</tr>

<tr>

<td>

Pepe(RCB)(84)<input type="checkbox" name="leicester[]" value="Pepe">

</td>

</tr>

<tr>

<td>

Alvaro Tejero(RB)(61)<input type="checkbox" name="leicester[]" value="AlvaroTejero">

</td>

</tr>

<tr>

<td>

Danilo(RB)(80)<input type="checkbox" name="leicester[]" value="Danilo">

</td>

</tr>

<tr>

<td>

Raphael Varane(CB)(82)<input type="checkbox" name="leicester[]" value="RaphaelVarane">

</td>

</tr>

<tr>

<td>

Philipp Lienhart(CB)(64)<input type="checkbox" name="leicester[]" value="PhilippLienhart">

</td>

</tr>

<tr>

<td>

Sergio Ramos(LCB)(87)<input type="checkbox" name="leicester[]" value="SergioRamos">

</td>

</tr>

<tr>

<td>

Marcelo(LB)(83)<input type="checkbox" name="leicester[]" value="Marcelo">

</td>

</tr>

<tr>

<td>

NachoFernandez(LCB)(77)<input type="checkbox" name="leicester[]" value="NachoFernandez">

</td>

</tr>

<tr>

<td>

Abner(LB)(63)<input type="checkbox" name="leicester[]" value="Abner">

</td>

</tr>

<tr>

<td>

Arbeloa(LB)(65)<input type="checkbox" name="leicester[]" value="Arbeloa">

</td>

</tr>

<tr>

<td>

Kiko Casilla(GK)(80)<input type="checkbox" name="leicester[]" value="KikoCasilla">

</td>

</tr>

<tr>

<td>

Ruben Yanez(GK)(65)<input type="checkbox" name="leicester[]" value="RubenYanez">

</td>

</tr>

<tr>

<td>

Keylor Navas(GK)(82)<input type="checkbox" name="leicester[]" value="KeylorNavas">

</td>

</tr>

</table>

<input type="submit" name="Submit" />

</form>

</body>

</html>

**Call python:**

<!DOCTYPE html>

<html>

<head>

<title>Call\_python</title>

</head>

<body>

<?php

//echo "good";

$pythonpath = 'C:/Python33/python.exe';

//$filepath = "C:\\studies\\xampp\\htdocs\\output.txt";

$pyscript = 'C:/studies/xampp/htdocs/pythonfinalchanged.py';

//$cmd = "$pythonpath $pyscript $filepath";

$command = escapeshellcmd('C:/studies/xampp/htdocs/miniproj/pythonfinalchanged.py');

$outpt = shell\_exec($command);

//echo $outpt;

/\*foreach ($output as $arr) {

echo $arr;

}

\*/

header('Location:/miniproj/result.html');

?>

</body>

</html>

**Display result:**

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

.background{

position: fixed;

z-index: -1000;

width: 100%;

height: 120%;

overflow: hidden;

top: 0;

left: 0;

}

#background-video{

position: absolute;

top: 0;

left: 0;

min-height: 100%;

min-width: 100%;

}

/\*.team\_select{

position: absolute;

width: 100%;

min-height: 100%;

z-index: 1000;

background-color: rgba(0,0,0,0.7);

}\*/

</style>

</head>

<body>

<?php

$fp = fopen("results.txt", "r");

//echo '<p align="center">'.show\_ad("location=top468x60").'</p>';

$message = nl2br(file\_get\_contents("results.txt"));

echo "<p style='position: absolute;color:white; font-size: 22px;text-align:center;top: 10%;left: 50%; margin-left: -170px;

width: 25%;' align=center>$message</p> ";

fclose($fp);

$fp = fopen("results.txt","w");

file\_put\_contents("results.txt", " ");

fclose($fp);

//echo '<div style="text-align: center;>$message</div>';

//echo $message;

?>

<div class="background">

<img id="background-video" src="Real-Madrid-HD-Picture-Wallpapers (1).jpeg" >

</div>

<form class="team\_select">

<p style="text-align:center;color:red;font-size:30px" >The results of the match: </p>

</form>

</body>

</html>

Display result page :

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

button{

position: absolute;

font-family:helvetica;

background: #587286;

margin:auto;

display:block;

text-align: center;

top: 20%;

left: 50%;

margin-left: -125px;

width: 25%;

}

.background{

position: fixed;

z-index: -1000;

width: 100%;

height: 120%;

overflow: hidden;

top: 0;

left: 0;

}

#background-video{

position: absolute;

top: 0;

left: 0;

min-height: 100%;

min-width: 100%;

}

</style>

</head>

<body >

<div class="background">

<img id="background-video" src="Real-Madrid-HD-Picture-Wallpapers (1).jpeg" >

</div>

<form action="/miniproj/result.php" method="post">

<button onclick="result.php" >Show results</button>

</form>

</body>

</html>

**REFERENCE**