A

Major Project

On

SPORTS ANALYSIS USING DATA ANALYTICS AND DATA VISUALIZATION

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

A. Vineeth Reddy(167R1A05J6)

T. Praveen Kumar(167R1A05P2)

Bonup Dheeraj(167R1A05J7)

Under the Guidance of

T. VASAVI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE ANDENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401

2017-21

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**SPORTS ANALYSIS USING DATA ANALYTICS AND DATA VISUALIZATION**" being submitted by **Vineeth Reddy, Praveen Kumar** and **Bonup Dheeraj** bearing the roll number **167R1A05J6, 167R1A05P2** and **167R1A05J7** respectively, in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2020-21.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Ms. T. Vasavi
Assistant Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voce Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

Ms. T. Vasavi, Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. J. Narasimha Rao, Mr. B. P. Deepak Kumar, Mr. K. Murali, Dr. Suwarna Gothane and Mr. B. Ramji**, for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman **Sri. Ch. Gopal Reddy** for his encouragement throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

A. VINEETH REDDY(167R1A05J6)

T.PRAVEEN KUMAR(167R1A05P2)

BONUP DHEERAJ(167R1A05J7)

ABSTRACT

The Federation International de Football Association (FIFA) is a non-profit organization which describes itself as an international governing body of association football, futsal, beach soccer, and football. It is the highest governing body of football. The FIFA association football World cup is played every 4 years. It is the most watched and followed scheduled event in the world.

The objective is to develop a software to organize and reflect on recent advances and challenges in the field of sports data visualization. Age, nation, international reputation and position are some of the attributes. Here, we analyze current research contributions through the lens of three categories of sports data: box score data (data containing statistical summaries of a sporting event such as a game), tracking data (data about in-game actions and trajectories), and meta-data (data about the sport and its participants but not necessarily a given name).

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
Fig. 3.1	Block diagram	8
Fig. 3.2	Use Case Diagram	9
Fig. 3.3	Class Diagram	9
Fig. 3.4	Activity Diagram	10
Fig. 3.5	Sequence Diagram	10
Fig. 6.1	General Report of Messi	28
Fig. 6.2	General report of Cristiano Ronaldo	28
Fig. 6.3	General Report of De Gea	29
Fig. 6.4	General Report of K. De Bruyne	29
Fig. 6.5	Most Preferred Foot of the Players	30
Fig. 6.6	Different nations Participants	30
Fig. 6.7	Different Body Types	31
Fig. 6.8	Distribution of Overall Scores	31
Fig. 6.9	Remuneration of the Players	32
Fig. 6.10	International Reputation of the Player	32
Fig. 6.11	Comparison of Position and Players	33
Fig. 6.12	Player Skill Moves	33
Fig. 6.13	Work rates of Players	34
Fig. 6.14	Height of the Players	34
Fig. 6.15	Speciality Scores of Players	35
Fig. 6.16	Player’s Potential Scores	35
Fig. 6.17	Player Overall Scores	36
Fig. 6.18	Age of Players	36
Fig. 6.19	International Reputation and Rating	37
Fig. 6.20	Weight of Player	37

LIST OF SCREENSHOTS

FIGURE NO	FIGURE NAME	PAGE NO
Fig. 4.2.1	Generate Reports of Players	16
Fig. 4.2.2	Details of Players	17
Fig. 4.2.3	Right Footballer Stats	17
Fig. 4.2.4	Footballer Stats Analysis	18
Fig. 4.2.5	Analysis	18
Fig. 4.2.6	Insights	19
Fig. 4.2.7	Stats of Footballer and Country	19
Fig. 4.2.8	General information of Players	20
Fig. 4.2.9	Important data of Club Players	20
Fig. 4.2.10	Clubs Information	21
Fig. 5.1	Name Error	23
Fig. 5.2	Recovering of Name Error	23
Fig. 5.3	Type Error	24
Fig. 5.4	Recovering of Type Error	24
Fig. 5.5	Indentation Error	25
Fig. 5.6	Recovering of Indentation Error	25
Fig. 5.7	Syntax Error	26
Fig. 5.8	Recovering Syntax Error	26
Fig. 5.9	File Not Found Error	27
Fig. 5.10	Recovering File Not Found Error	27

TABLE OF CONTENTS

ABSTRACT i

LIST OF FIGURES ii

LIST OF SCREENSHOTS iii

1. INTRODUCTION

1.1 About Project 1

1.2 Existing System with Drawbacks 2

1.3 Proposed System with Features 2

2. SYSTEM ANALYSIS

2.1 Hardware and Software Requirements 3

2.2 Functional requirements 3

2.3 Non-Functional requirements 4

2.4 Module Description 5

3. ARCHITECTURE

3.1 Block Diagram 8

3.2 Use Case Diagram 9

3.3 Class Diagram 9

3.4 Activity Diagram 10

3.5 Sequence Diagram 10

4. IMPLEMENTATION

4.1 Technology Used 11

4.2 Frontend Design 16

5. TESTING 22

6. RESULTS 28

7. CONCLUSION 38

8. FUTURE SCOPE 39

9. BIBLIOGRAPHY 40

1. INTRODUCTION

1.1 About project

Sports analysis has been a trend that has grown a lot and remains extremely popular. The knowledge about sports analytics and these years remains very popular. Enhancement of the viewer experience in football plus the insights from predictive models is the perfect match to bring more spectators to sports and keep those who follows. Currently this is a wealth of data on clubs, players and sports games. The content visualization have promoted the consumption of these content. As a result, sports data analysis is becoming extensive and diversified.

The objective into develop a software to organize and reflect on recent advances and challenges in the field of sports data visualization. Age, nation, international reputation are some of the attributes. Prediction of the sports and the performance of the teams in the future. Big data analysis can be useful to collect and analyse data to predict the future requirements by applying machine learning techniques. Machine learning models use past data to recognize patterns followed by applying new data to predict an outcome.

The system is intended to collect football related data, such as information from clubs, players, matches and competition venues and apply data analytics mechanisms to extract relevant information. The statistical technologies and techniques lead to an improvement in data collection and decision making competitive sports. We propose a visual analytics work flow for interactive feature visualization enabling experts to understand the characteristics of situations. This enables drastically improved data collection in many application areas, With data visualization, large data and records can be quickly and conveniently analysed without having to go through piece by piece. Data presented in this manner are in the form of charts, graphs and images.

1.2 EXISTING SYSTEM :

1. No feature engineering.

2. Data visualization is not having analysis among the features.

3. Statistically analyzing is not done on positions.

4. Querying upon the data only which is available in the dataset.

1.3 LIMITATIONS OF EXISTING SYSTEM:

1. The traditional analysis is unable to predict the future outcomes out of the analysis and the past events that occurred.

2. In the traditional system, data visualization techniques are not used.

3. The data representation need to be updated in the traditional system.

4. The statistical analysis is not properly initiated in the existing system.

5. The feature engineering techniques need to be implemented

1.4 PROPOSED SYSTEM WITH FEATURES:

1. Using feature engineering segregating different features of players and defining new sports attributes to define the players reports.

2. Data visualization consisting of various charts, graphs and analysis is done among the features.

3. Statistical analysis of different skills required to play on different positions of football field.

4. Querying upon the data to find out high end results and perform qualification analysis.

2. SYSTEM ANALYSIS

2.1 HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirements:

Processor	:	I3 (Intel core processor)
RAM	:	4GB (min)
Hard Disk Capacity	:	40 GB (min)

Software Requirements:

Operating System	:	Windows XP/7+
Programming Language	:	Python
Environment	:	Jupyter Notebook(1.1.0)

2.2 FUNCTIONAL REQUIREMENTS:

Functional requirements are associated with specific functions, tasks or behaviour the system must support. It define what a system is supposed to do.

Functional requirements describe a system's capabilities and services. They are:

- Data analysis
- Data visualization

Data analysis:

Data analysis is a process of inspecting, cleansing, transforming and modelling data with the goal of discovering useful information, informing conclusion and supporting decision making. Here we analyse the datasets to summarize their main characteristics often with visual methods. Data analysis in sports has reached an important level.

Data visualization:

It is graphical representation of data by using visual elements like charts, maps and graphs. Here, we have the knowledge on sports domain and attributes.

2.3NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviours. It defines how a system is supposed to be. Non-functional requirements describe the properties of capabilities and the desired level of services. They are:

- Performance
- Scalability
- Reliability
- Security
- Usability

Performance:

Performance defines how fast the system or its particular piece responds to certain users' actions under certain workload. In most cases, this metric explains how much a user must wait before the target operation happens.

Scalability:

Scalability assesses the highest workloads under which the system will still meet the performance requirements.

Reliability:

Reliability is a quality attribute that specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions. Traditionally, it's expressed as a probability percentage.

Security:

Security assures that all data inside the system or its part will be protected against malware attacks or unauthorized access.

Usability:

Usability is yet another classical non-functional requirement that addresses a simple question: How hard is it to use the product? Defining these requirements isn't as easy as it seems.

2.4 MODULE DESCRIPTION:

1. **Data collection:**

It is the process of systematic approach to gathering and measuring information from a variety of sources to get a complete and accurate. The goal for all data collection is to capture quality evidence that allows analysis to lead to the formulation of convincing and credible answers to the questions that have been posted. Here we have collected the data through the kaggle.com and dataset is named as FIFA 19.

2. **Feature engineering:**

It is the process of using domain knowledge to extract features from raw data via data mining techniques. It includes image, audio, and textual data, but could just as easily include tabular data with millions of attributes. It is automatically reducing dimensionality of these types of observations into a much smaller set that can be

modelled. Likewise we have reduced the attributes of players with one word like Defending, General, Rating and Shooting etc.

3. **Exploratory of Data Analysis:**

It is a process of inspecting, cleansing, transforming and modelling data with the goal of discovering useful information, informing conclusion and supporting decision-making. The purpose of data analysis is to extract useful information from data and taking the decision based upon the data analysis. This is nothing but analysing our past or future and making decisions based on it can be analysed by charts and graphs.

4. **Data visualization:**

It is used as combination of manual methods and automated tools such as data visualizations, charts, and initial reports. It is about describing the data by means of statistical and visualization techniques. We have Explored the data in order to bring important aspects of that data into focus for further analysis. More importantly, it helps build a familiarity with the existing information that makes finding better answers much simpler.

5. **Results:** We train the data by using previous data analysis to predict the future outcomes. The findings are reported based on the methodology. The graphs are charts are generated in-order to analyse the data.

3. ARCHITECTURE

3.1 BLOCK DIAGRAM:

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams.

Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction.

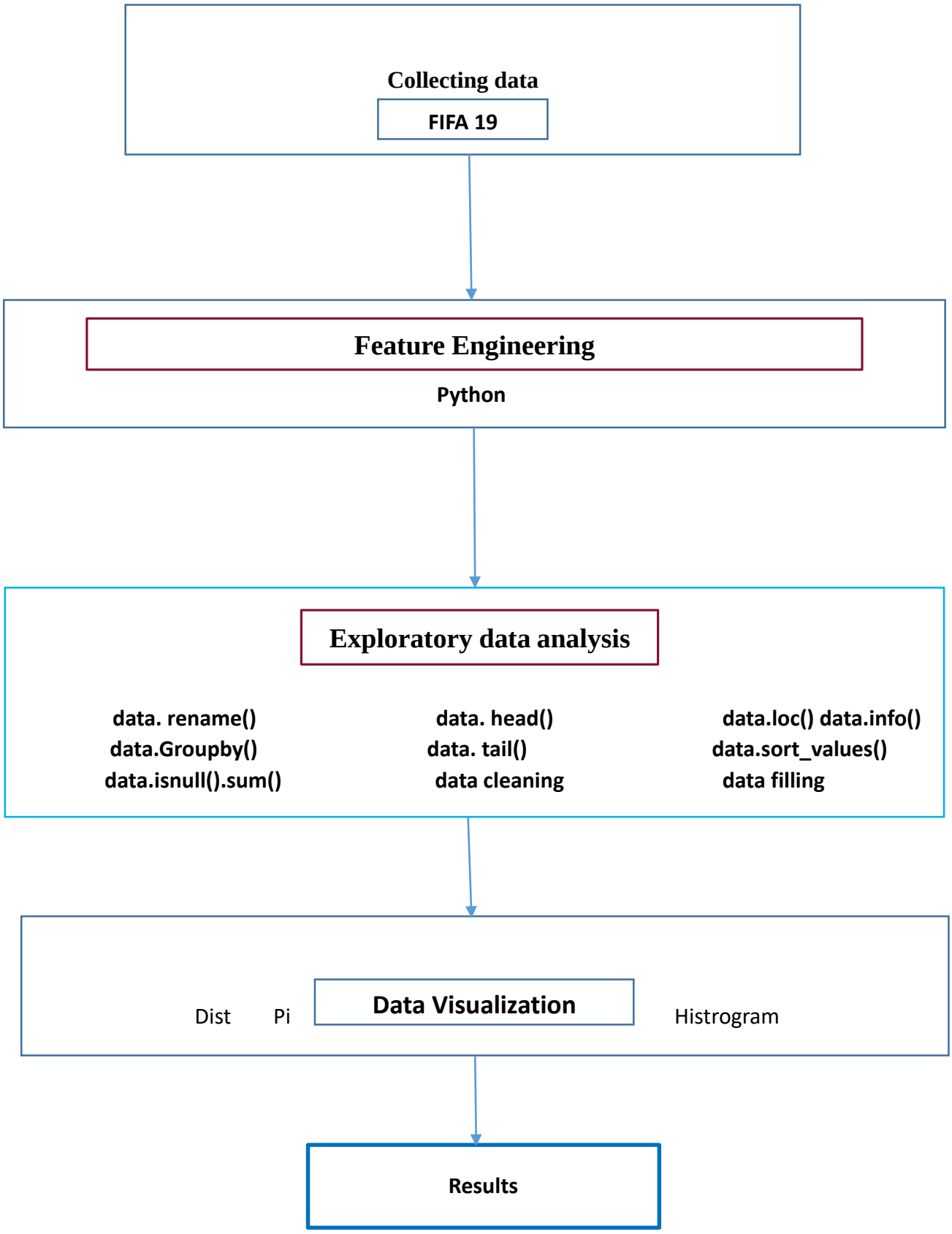

Figure 3.1 Block Diagram

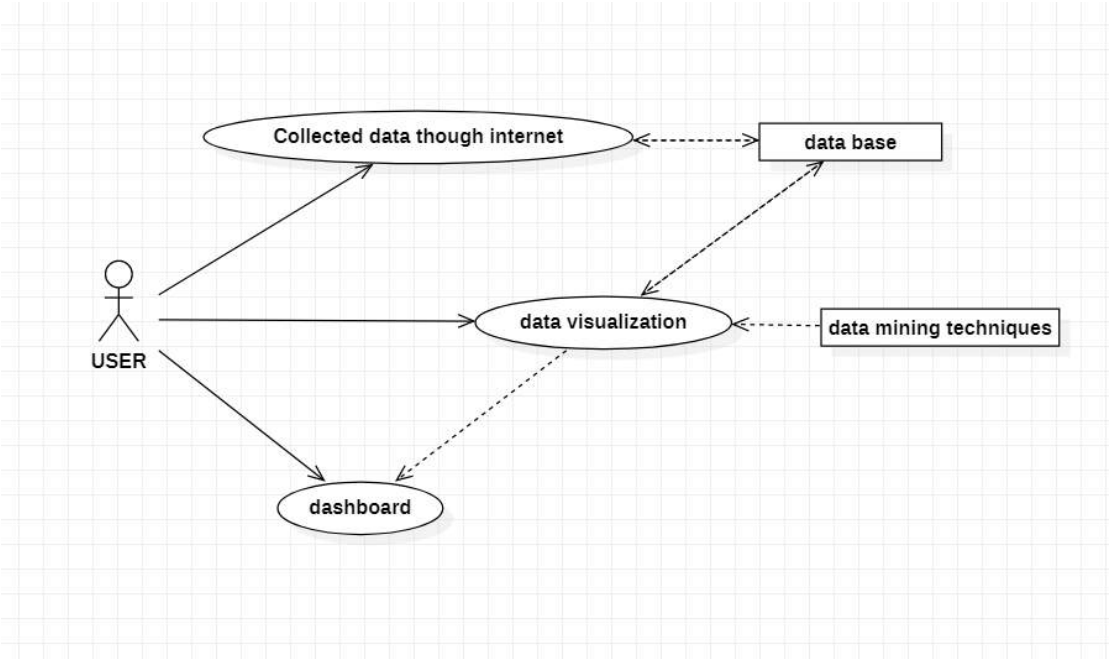

Fig. 3.2 Use Case Diagram

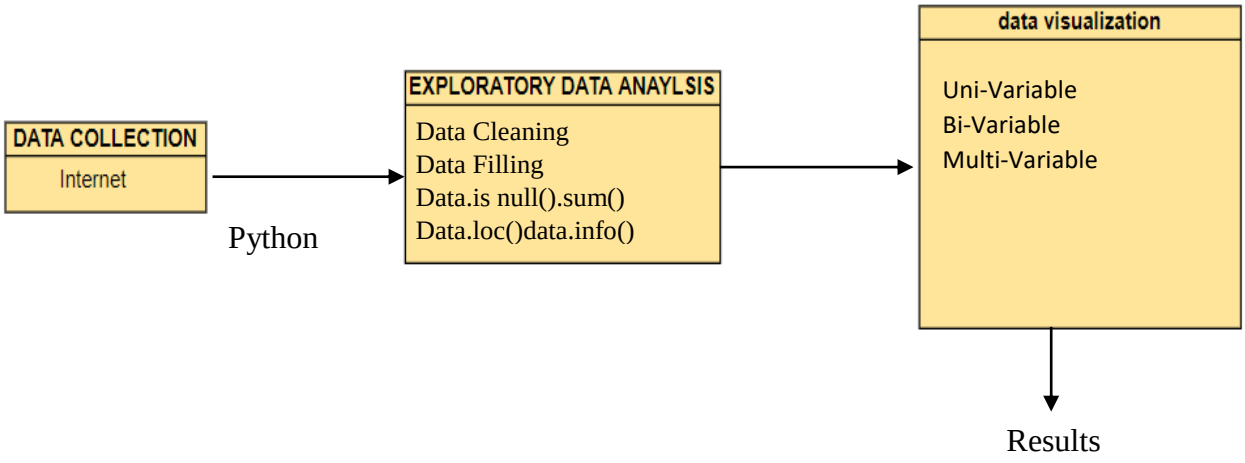

Fig. 3.3 Class Diagram

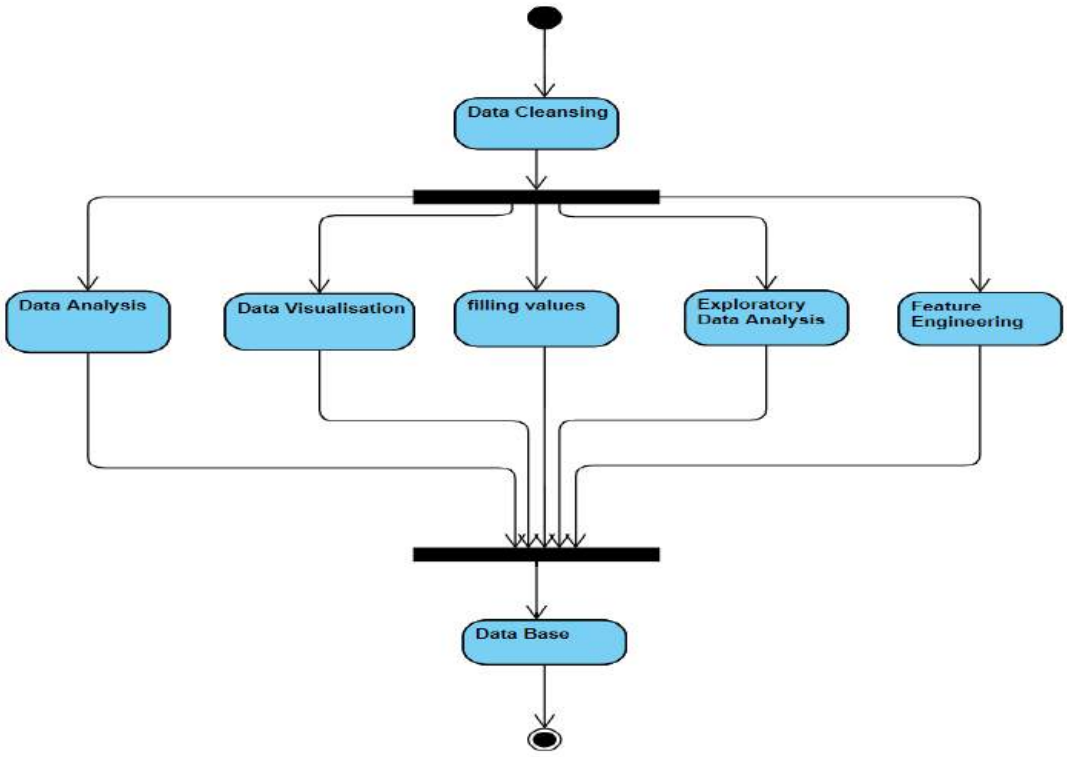

Fig. 3.4 Activity Diagram

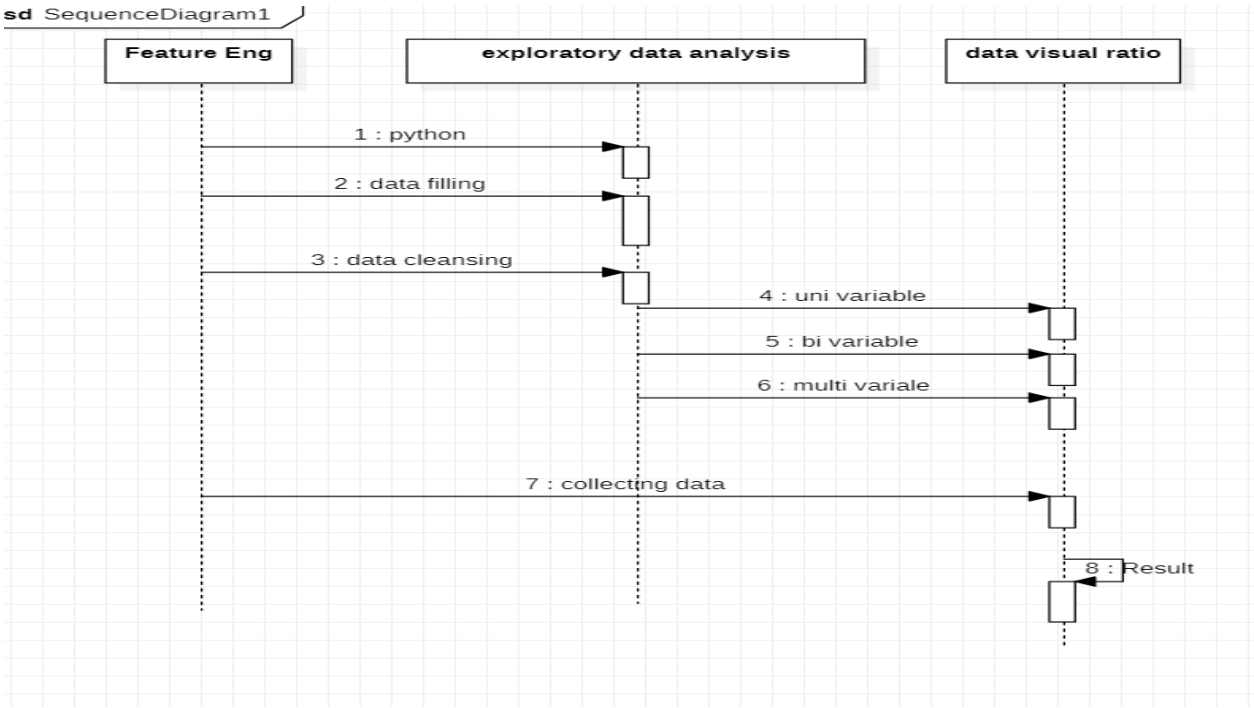

Fig. 3.5 Sequence Diagram

4. IMPLEMENTATION

4.1 TECHNOLOGY USED:

Python:

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. Python can be treated in a procedural way, an object-orientated way or a functional way .Python was designed for readability, and has some similarities to the English language with influence from mathematics. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).It has a simple syntax similar to the English language .It has syntax that allows developers to write programs with fewer lines than some other programming languages .Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Data science:

Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining and big data.

Data science is a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyse actual phenomena" with data. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, and information science.

Data science is an interdisciplinary field focused on extracting knowledge from data sets, which are typically large (see big data). The field encompasses analysis, preparing data for analysis, and presenting findings to inform high-level decisions in an organization. As such, it incorporates skills from computer science, mathematics, statistics, information visualization, graphic design, and business. Statistician Nathan Yau, drawing on Ben Fry, also links data science to human-computer interaction: users should be able to intuitively control and explore data.

Machine learning:

It is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or in-feasible to develop a conventional algorithm for effectively performing the task.

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. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

SOFTWARE USED:

Jupyter Notebook:

The Jupyter Notebook application allows you to create and edit documents that display the input and output in which name comes from the core supported programming languages that it supported: Julia, Python and R. Jupyter ships with the IPython Kernel ,which allows you to write your programs in python ,but there are currently over 100 other kernels that you can also use. It is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Jupyter notebook is maintained by the people at Project Jupyter.

LIBRARIES USED:

1. NumPy:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional data.

Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

2. Pandas:

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the python programming language. Using Pandas we can create table format data structures which has column names and column names.

With Pandas we can write data from Pandas data-frame to csv or excel file and also we store data to pandas data-frame from csv or excel file. This makes flexible while working with large datasets in Machine learning. Pandas provides various built in functions. Which makes it easy to access and retrieve data from datasets

3. Matplotlib:

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Pip install matplotlib running this python shells installs matplotlib From matplotlib import pyplot as plt or import matplotlib. Pyplot as plt are used to importing matplotlib.

Some of the basic graphs of Matplotlib are:

- Bar Graph
- Scatter plot
- Histogram.
- Box plot
- Pie chart

4. Seaborn:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn tries to make it easy to switch between different visual representations that can be parameterized with the same data set-oriented API. Built on top of Matplotlib. Has Fewer lines of code. Categorical data is represented in x-axis and values correspond to them represented through y-axis. Used for visualizing both uni-variate or bi-variate distributions

pip install seaborn Command to install seaborn. Import seaborn as sns is used to import it

Some of the basic graphs are:

- Count Plot
- Histogram
- Dist Plot
- Scatter plot
- Violin Plot
- Swarm Plot
- Heat Map

5. Ipywidgets:

Ipywidgets are interactive HTML widgets for Jupyter notebooks, JupyterLab and the IPython kernel.

Notebooks come alive when interactive widgets are used. Users gain control of their data and can visualize changes in the data.

Learning becomes an immersive, fun experience. Researchers can easily see how changing inputs to a model impact the results. We hope you will add ipywidgets to your notebooks, and we're here to help you get started.

A demonstration notebook provides an overview of the core interactive widgets, including:

- Sliders
- progress bars
- text boxes
- toggle buttons and check-boxes
- display area
- and more

4.2 FRONT END DESIGN:

Figure 4.2.1 Generate Reports of players

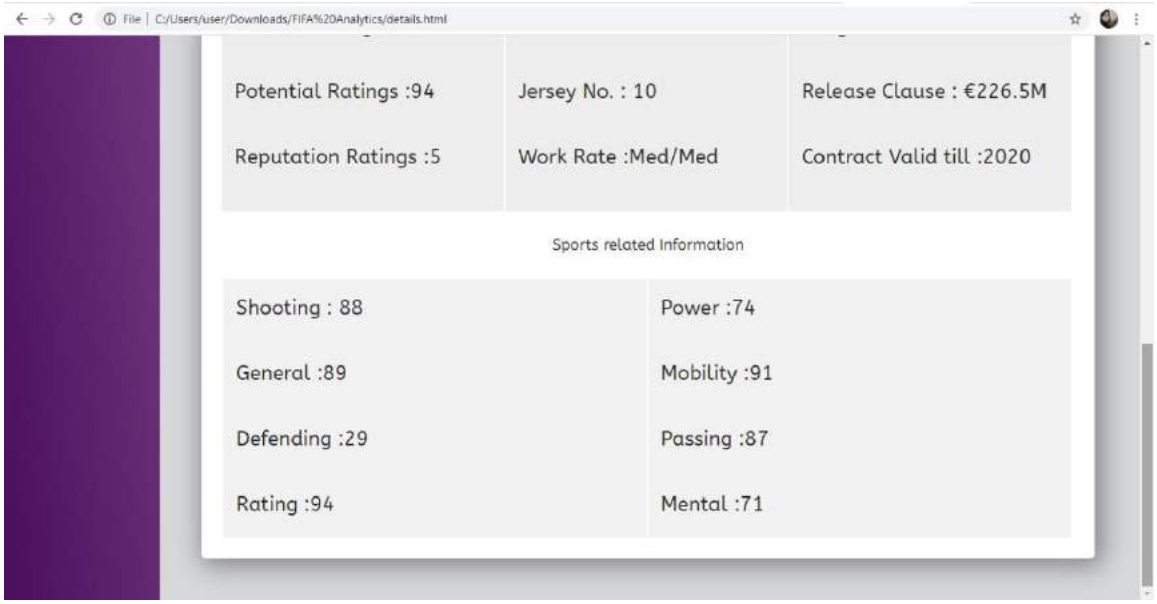

Figure 4.2.2 Details of player

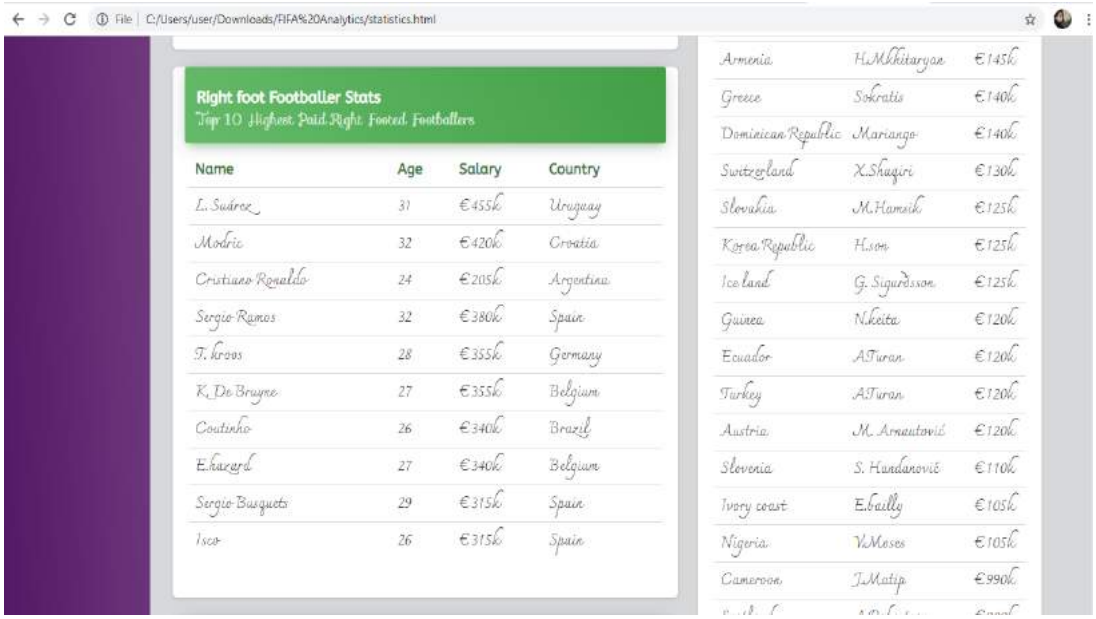

Figure 4.2.3 Right foot Footballer Stats

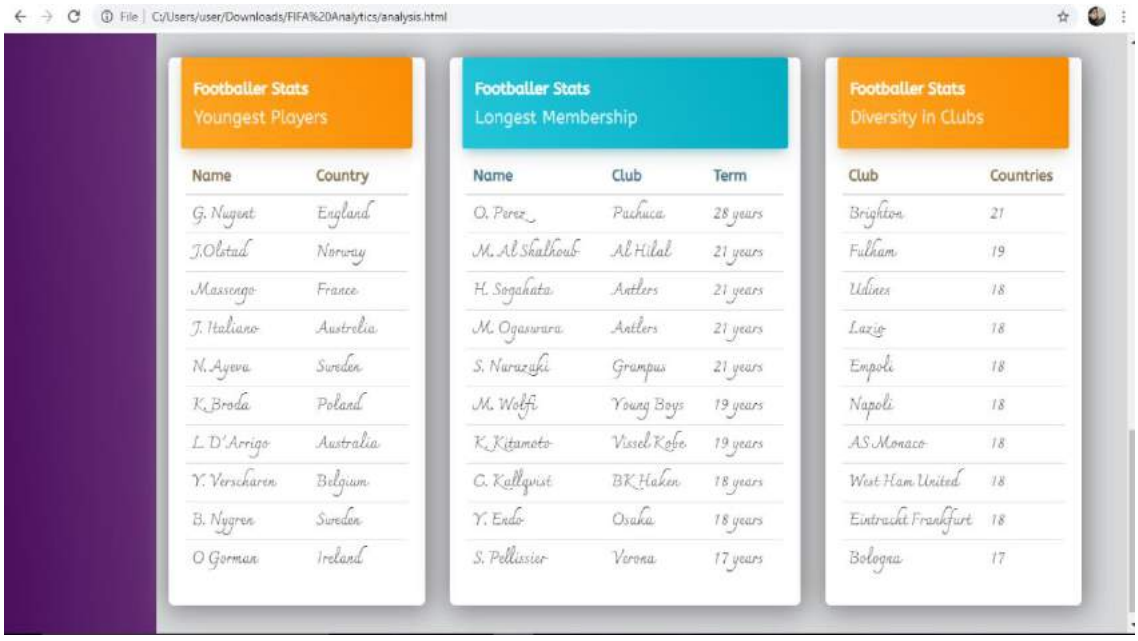

Figure 4.2.4 Footballer Stats analysis

Figure 4.2.5 Analysis

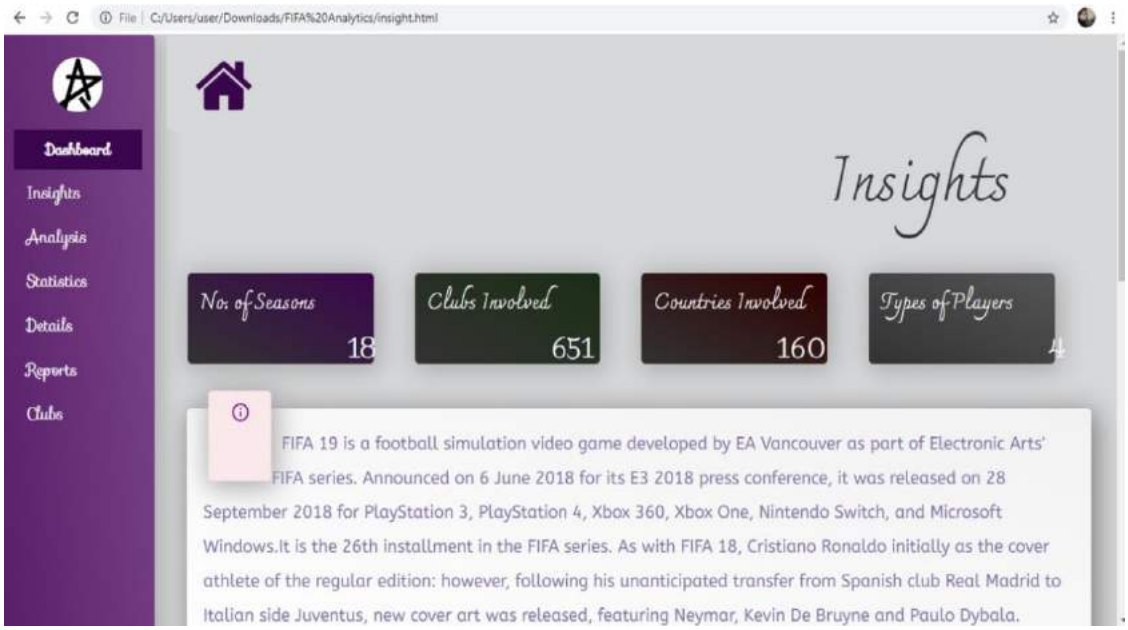

Figure 4.2.6 Insights

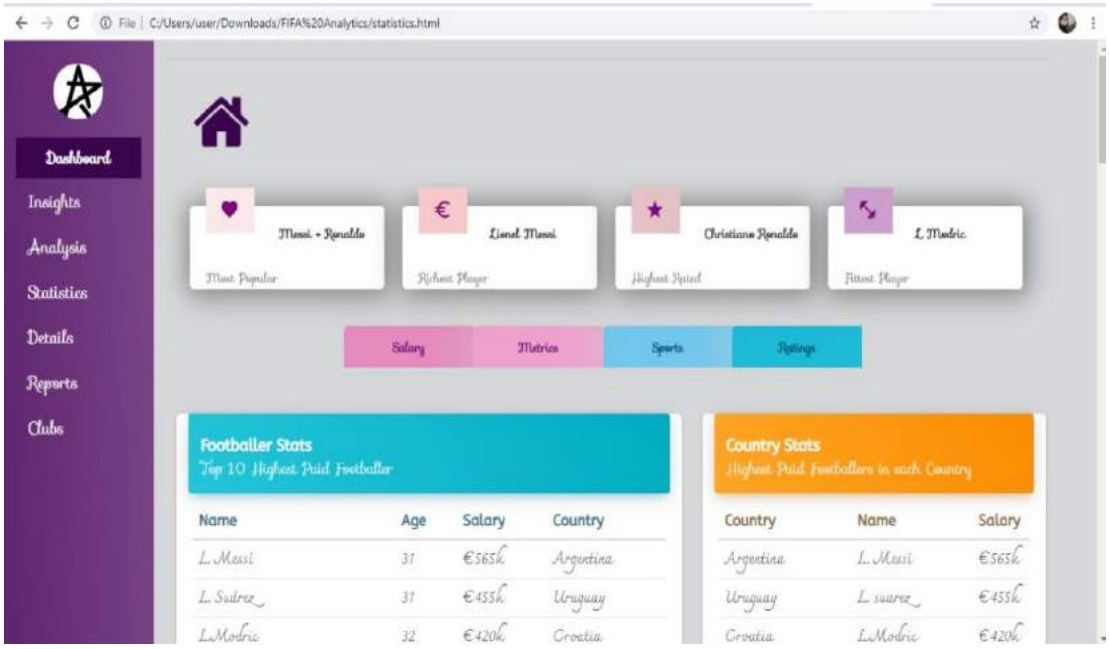

Figure 4.2.7 Stats of footballer and country

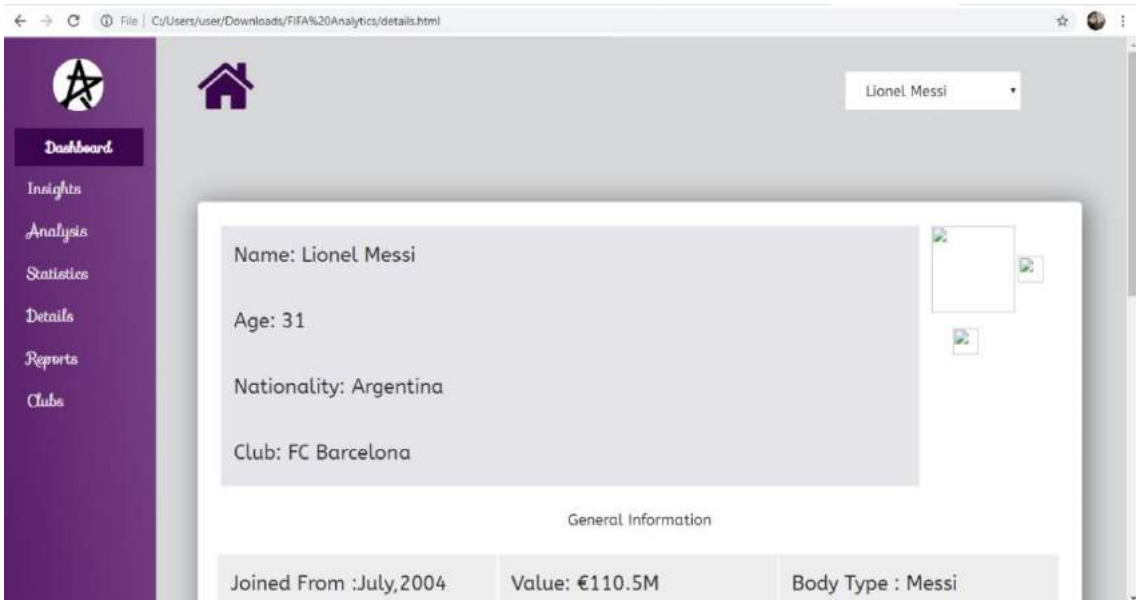

Figure 4.2.8 General information of players

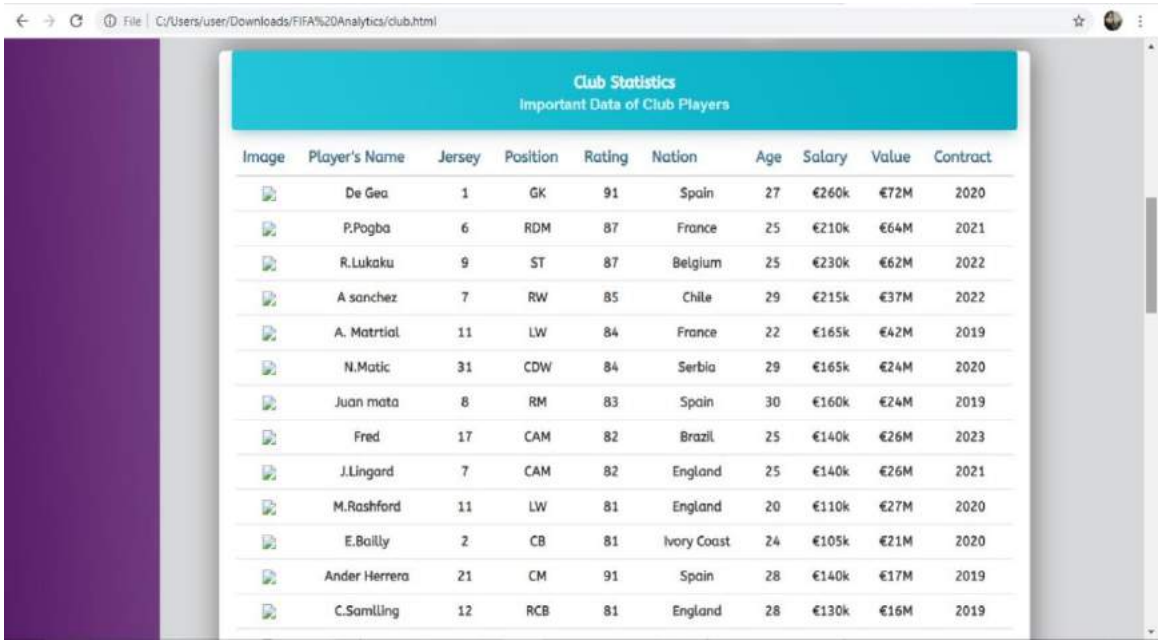

Figure 4.2.9 Important Data of Club Players

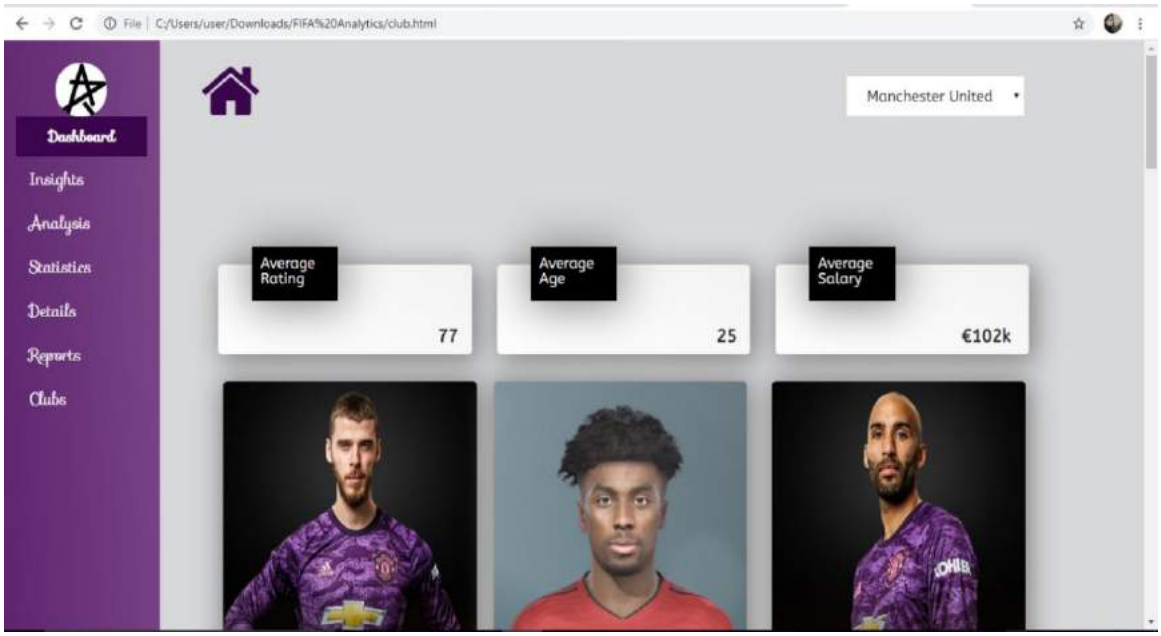

Figure 4.2.10 Clubs information

5.TESTING

Testing presents an interesting anomaly for the software engineer. During earlier software engineering activities, the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test cases that are intended to "demolish" the software that has been built. In fact, testing is the one step in the software process that could be viewed (psychologically, at least) as destructive rather than constructive. Software engineers are by their nature constructive people. Testing requires that the developer discard preconceived notions of the "correctness" of software just developed and overcome a conflict of interest that occurs when errors are uncovered.

If testing is conducted successfully (according to the objectives stated previously), it will uncover errors in the software. As a secondary benefit, testing demonstrates that software functions appear to be working according to specification, that behavioral and performance requirements appear to have been met. In addition, data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole. But testing cannot show the absence of errors and defects, it can show only that software errors and defects are present. It is important to keep this (rather gloomy) statement in mind as testing is being done. Some of the test cases performed are name error, type error, indentation error, syntax error and file not found error.

Name-Error: A Name Error means that Python tried to use a variable or function name. The most common cause is a simple misspelling of the function being used.

Usual causes:

- A mistyped variable or function name.
- Using a variable before it is defined.
- The name was intended to be enclosed in quotes.

Figure 5.1 Name Error

Figure 5.2 After recovering the Name Error the correct output is generated

TypeError:A Type Error occurs when an operation or function is applied to an object of inappropriate type. It is a kind of fault that occurs during the hypothesis testing process when a null hypothesis is rejected, even though it is accurate and should not be rejected.

```
In [26]: # The Longest membership in the club

         import datetime

         now = datetime.datetime.now('s')
         data['Join_year'] = data.Joined.dropna().map(lambda x: x.split(',')[1].split(' ')[1])
         data['Years_of_member'] = (data.Join_year.dropna().map(lambda x: now.year - int(x))).astype('int')
         membership = data[['Name', 'Club', 'Years_of_member']].sort_values(by = 'Years_of_member', ascending = False).head(10)
         membership.set_index('Name', inplace=True)
         membership

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         <ipython-input-26-05978868254f> in <module>
               3 import datetime
               4
         ----> 5 now = datetime.datetime.now('s')
               6 data['Join_year'] = data.Joined.dropna().map(lambda x: x.split(',')[1].split(' ')[1])
               7 data['Years_of_member'] = (data.Join_year.dropna().map(lambda x: now.year - int(x))).astype('int')

         TypeError: tzinfo argument must be None or of a tzinfo subclass, not type 'str'
```

Figure 5.3 Type Error

```
In [60]: # The Longest membership in the club

         import datetime

         now = datetime.datetime.now()
         data['Join_year'] = data.Joined.dropna().map(lambda x: x.split(',')[1].split(' ')[1])
         data['Years_of_member'] = (data.Join_year.dropna().map(lambda x: now.year - int(x))).astype('int')
         membership = data[['Name', 'Club', 'Years_of_member']].sort_values(by = 'Years_of_member', ascending = False).head(10)
         membership.set_index('Name', inplace=True)
         membership
```

Out[60]:

Name	Club	Years_of_member
O. Pérez	Pachuca	28
M. Al Shalhoub	Al Hilal	21
H. Sogahata	Kashima Antlers	21
M. Ogasawara	Kashima Antlers	21
S. Narazaki	Nagoya Grampus	20
M. Wölfli	BSC Young Boys	19
K. Kitamoto	Vissel Kobe	19
C. Källqvist	BK Häcken	18
Y. Endo	Gamba Osaka	18

Figure 5.4 After recovering the Type Error the correct output is generated

IndendationError: Whenever you have a situation with code inside of a statement (such as the code that defines the main() function), that 'inside' code must be indented,

and must be indented consistently. Like mentioned , this error primarily occurs because of space or tab errors in your code.

```
In [13]: # plotting a correlation heatmap

         plt.rcParams['figure.figsize'] = (30, 20)
         #sns.heatmap(data_selected[['Age', 'Nationality', 'Overall', 'Potential', 'Club', 'Value',
                             'Wage', 'Special', 'Preferred Foot', 'International Reputation', 'Weak Foot',
                             'Skill Moves', 'Work Rate', 'Body Type', 'Position', 'Height', 'Weight',
                             'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribbling',
                             'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
                             'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
                             'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
                             'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
                             'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
                             'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause']].corr(), annot = True)

         plt.title('Histogram of the Dataset', fontsize = '10')
         plt.show()

           File "<ipython-input-13-e43fba9f67e7>", line 5
             'Wage', 'Special', 'Preferred Foot', 'International Reputation', 'Weak Foot',
             ^
         IndentationError: unexpected indent
```

Figure 5.5 Indentation Error

```
In [43]: # plotting a correlation heatmap

         plt.rcParams['figure.figsize'] = (30, 20)
         sns.heatmap(data_selected[['Age', 'Nationality', 'Overall', 'Potential', 'Club', 'Value',
                             'Wage', 'Special', 'Preferred Foot', 'International Reputation', 'Weak Foot',
                             'Skill Moves', 'Work Rate', 'Body Type', 'Position', 'Height', 'Weight',
                             'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribbling',
                             'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
                             'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
                             'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
                             'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
                             'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
                             'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause']].corr(), annot = True)

         plt.title('Histogram of the Dataset', fontsize = 30)
         plt.show()
```

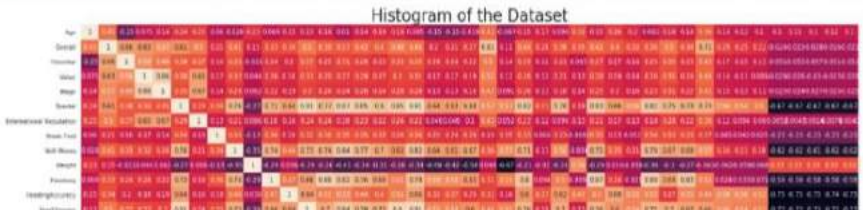

Figure 5.6 After recovering the Indentation Error the correct output is generated

Syntax-Error: A syntax error is an error in the source code of a program. These are small grammatical mistakes, sometimes limited to a single character. For example extra bracket at the end of the function.

```
In [14]: # Every Nations' Player and their Weights

some_countries = ['England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'Italy', 'Columbia')
data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Weight']]

plt.rcParams['figure.figsize'] = (12, 7)
ax = sns.violinplot(x = data_countries['Nationality'], y = data_countries['Weight'], palette = 'colorblind')
ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
ax.set_ylabel(ylabel = 'Weight in lbs', fontsize = 9)
ax.set_title(label = 'Distribution of Weight of players from different countries', fontsize = 20)
plt.show()

  File "<ipython-input-14-aab452a3ec56>", line 3
    some_countries = ['England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'Italy', 'Columbia')
                                                                                                        ^
SyntaxError: invalid syntax
```

Figure 5.7 Syntax Error

```
In [47]: # Every Nations' Player and their Weights

some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'Italy', 'Columbia')
data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Weight']]

plt.rcParams['figure.figsize'] = (12, 7)
ax = sns.violinplot(x = data_countries['Nationality'], y = data_countries['Weight'], palette = 'colorblind')
ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
ax.set_ylabel(ylabel = 'Weight in lbs', fontsize = 9)
ax.set_title(label = 'Distribution of Weight of players from different countries', fontsize = 20)
plt.show()

/opt/conda/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensiona
l indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array inde
x, `arr[np.array(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

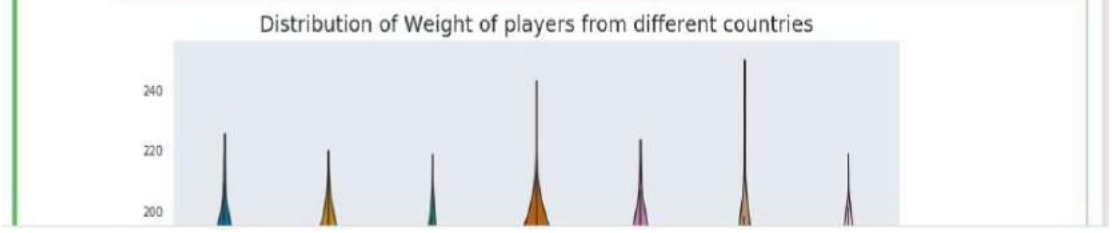

Figure 5.8 After recovering the Syntax Error the correct output is generated

File-Not-Found-Error: This error is raised when the path of file doesn't lead to it or when the file doesn't exists.

Figure 5.9 File Not Found Error

Figure 5.10 After recovering the File Not Found Error the correct output is generated

6. RESULTS

Polar Graph:

Figure 6.1 General report of Messi

Figure 6.2 General report of Cristiano Ronaldo

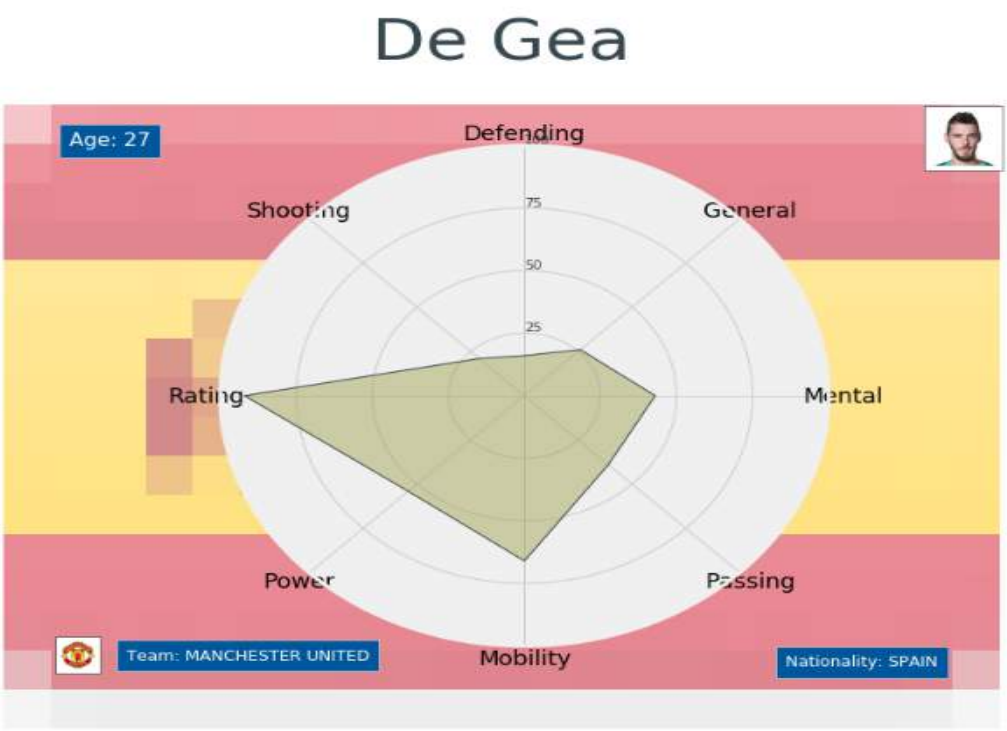

Figure 6.3 General report of De Gea

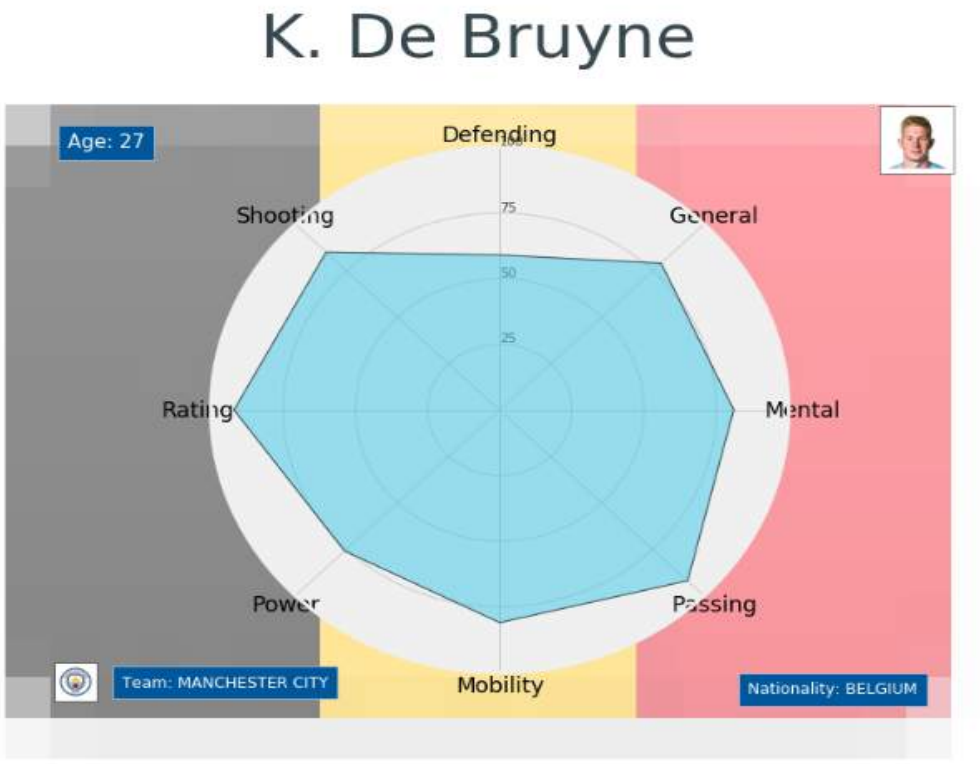

Figure 6.4 General report of K. De Bruyne

Bar Graph:

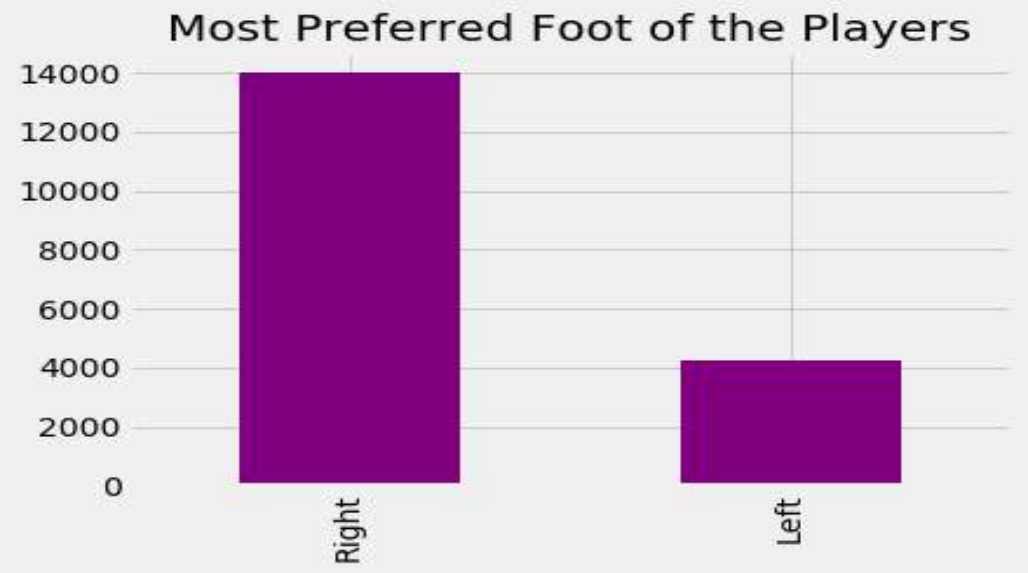

Figure 6.5 Most preferred Foot of the Players

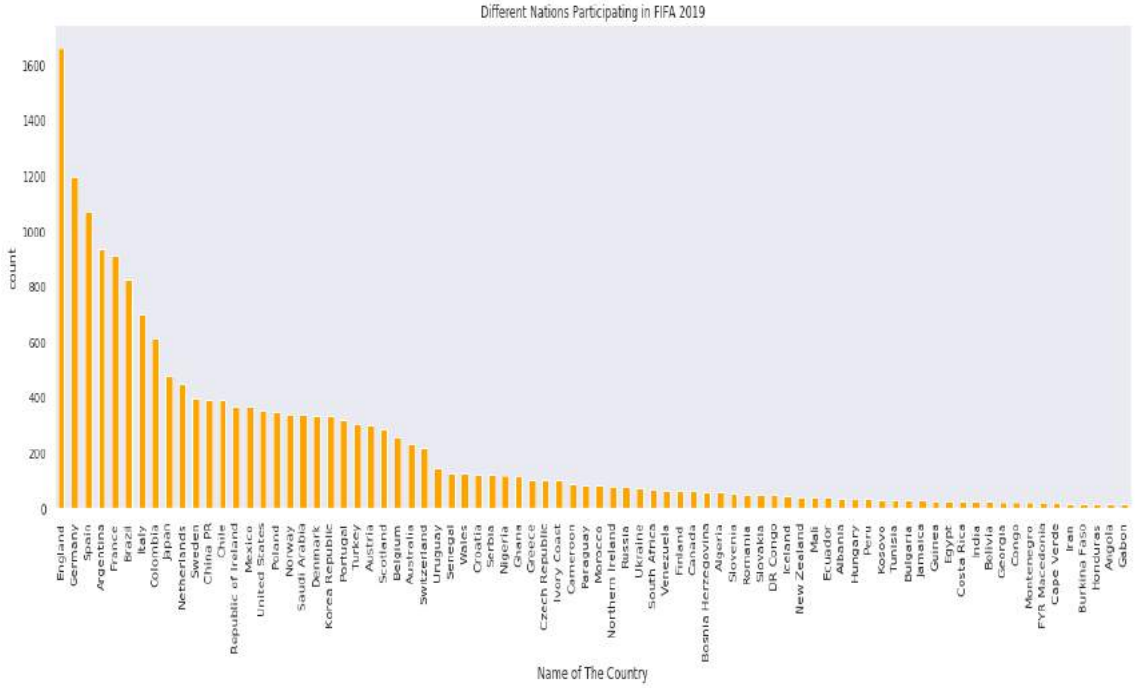

Figure 6.6 Different Nations participation

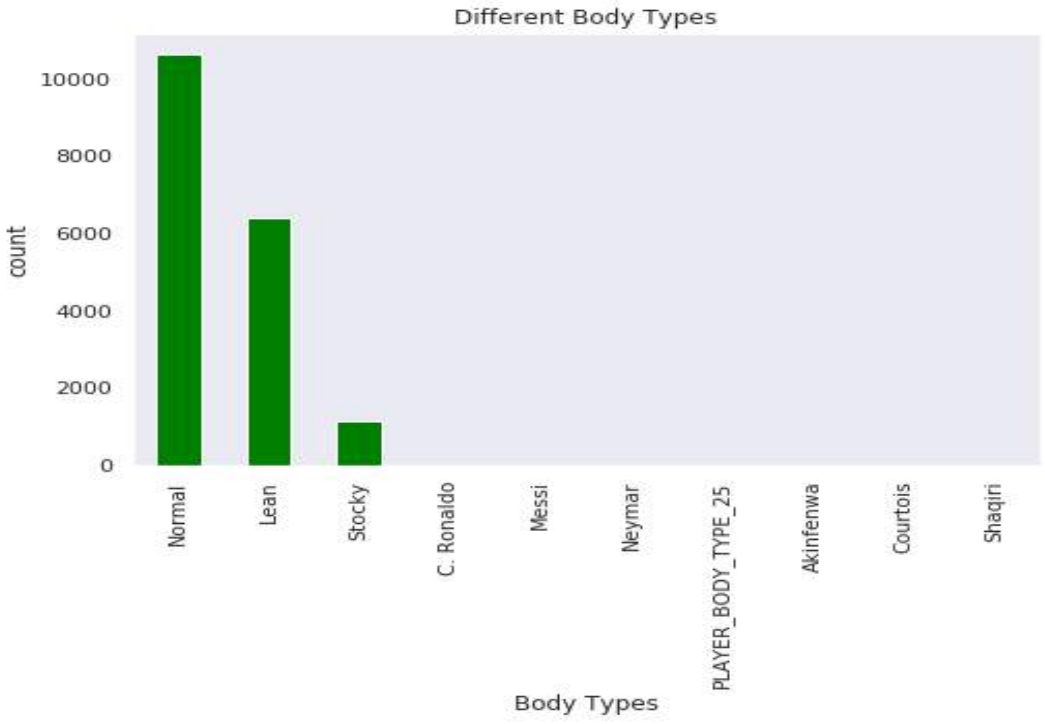

Figure 6.7 Different Body Types

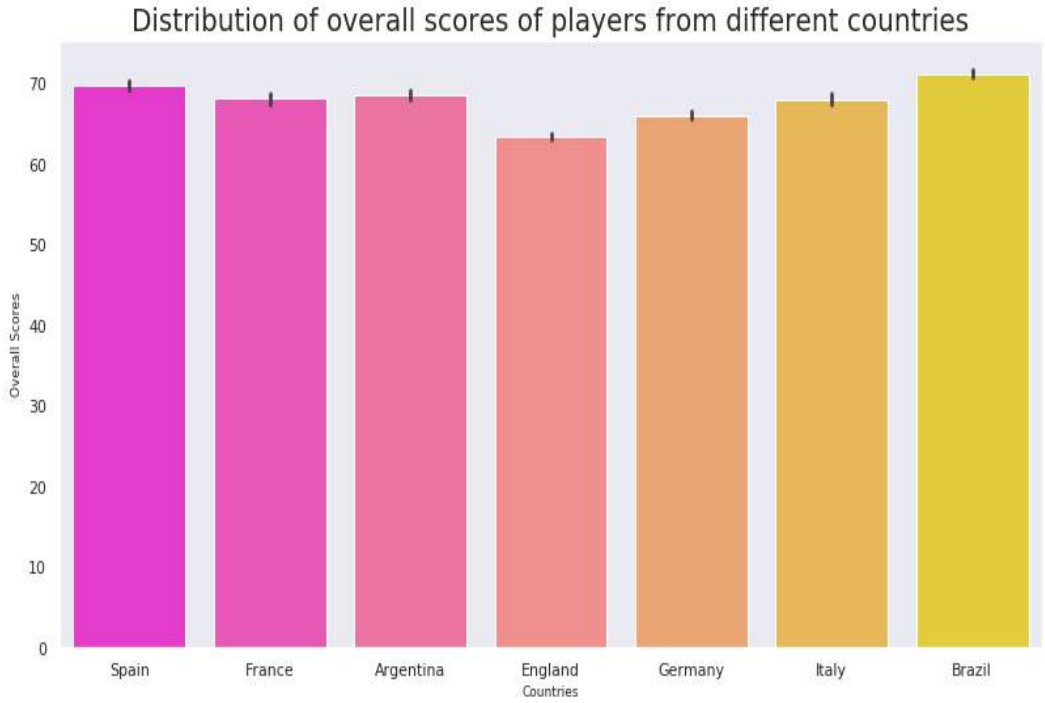

Figure 6.8 Distribution of overall scores of Players

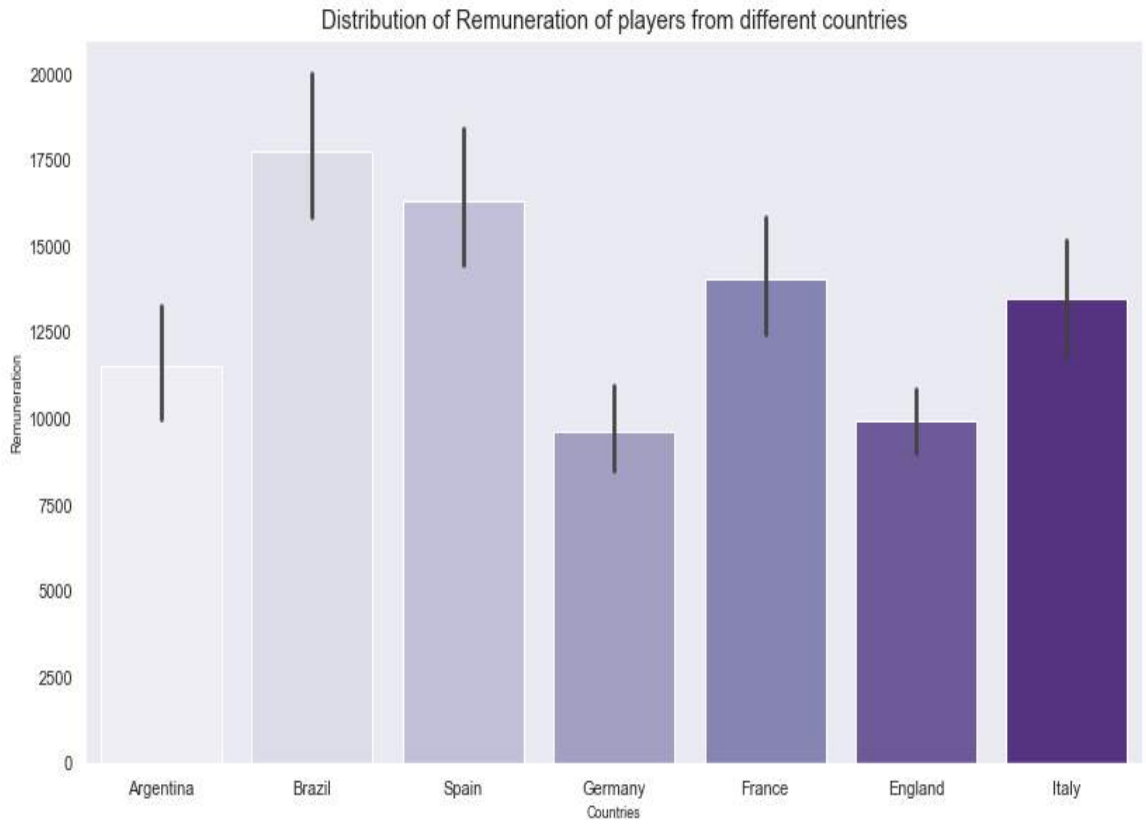

Figure 6.9 Remuneration of Players

Pie Chart:

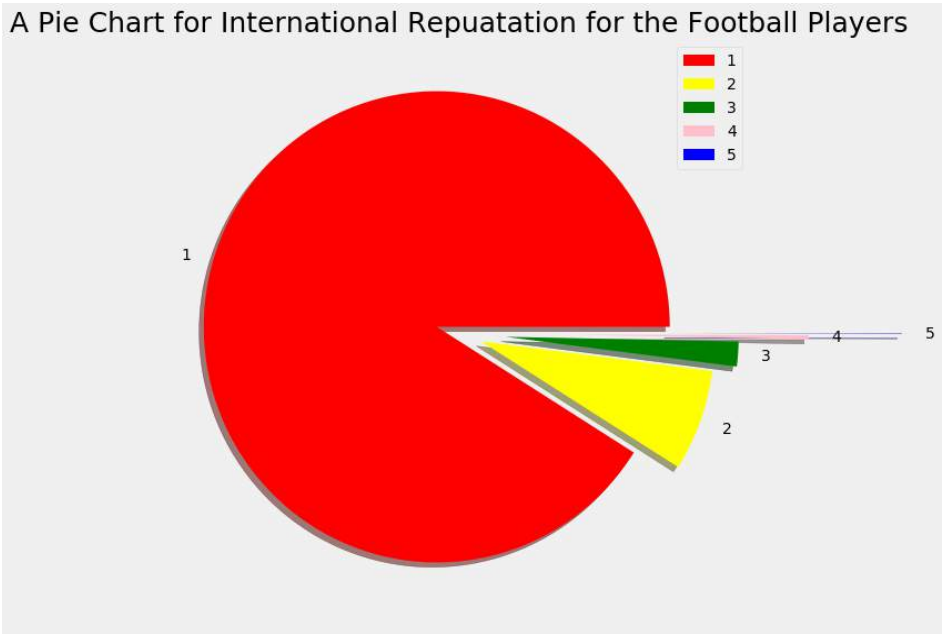

Figure 6.10 International Reputation for Player

Count plot:

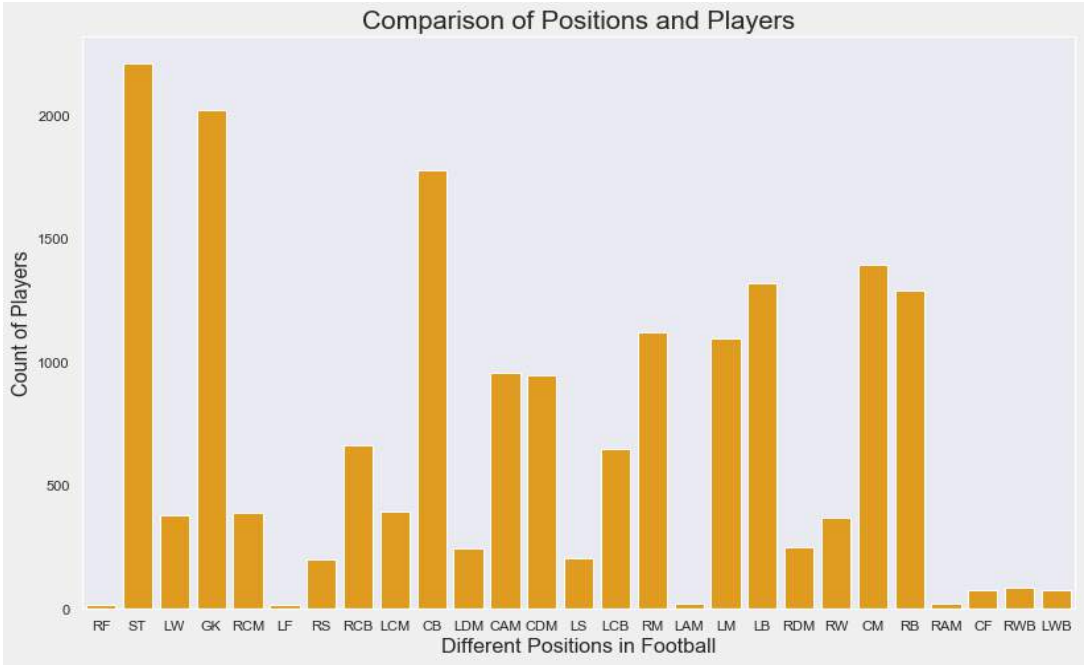

Figure 6.11 Comparison of Positions and Players

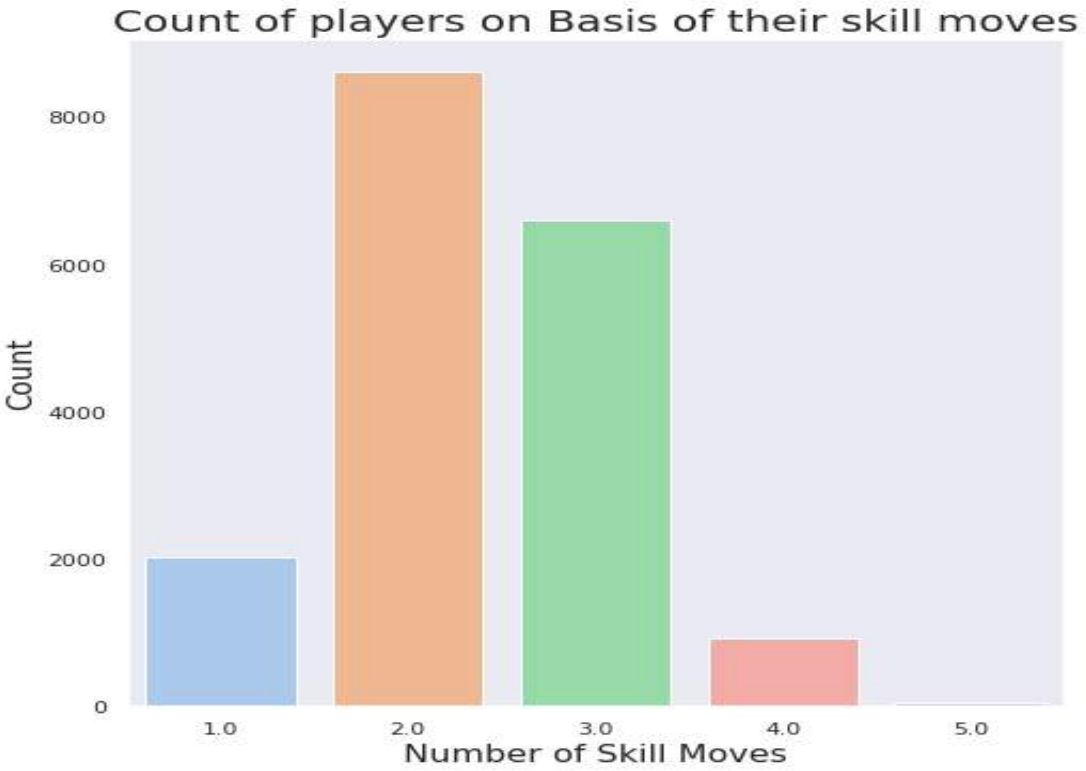

Figure 6.12 Players skill moves

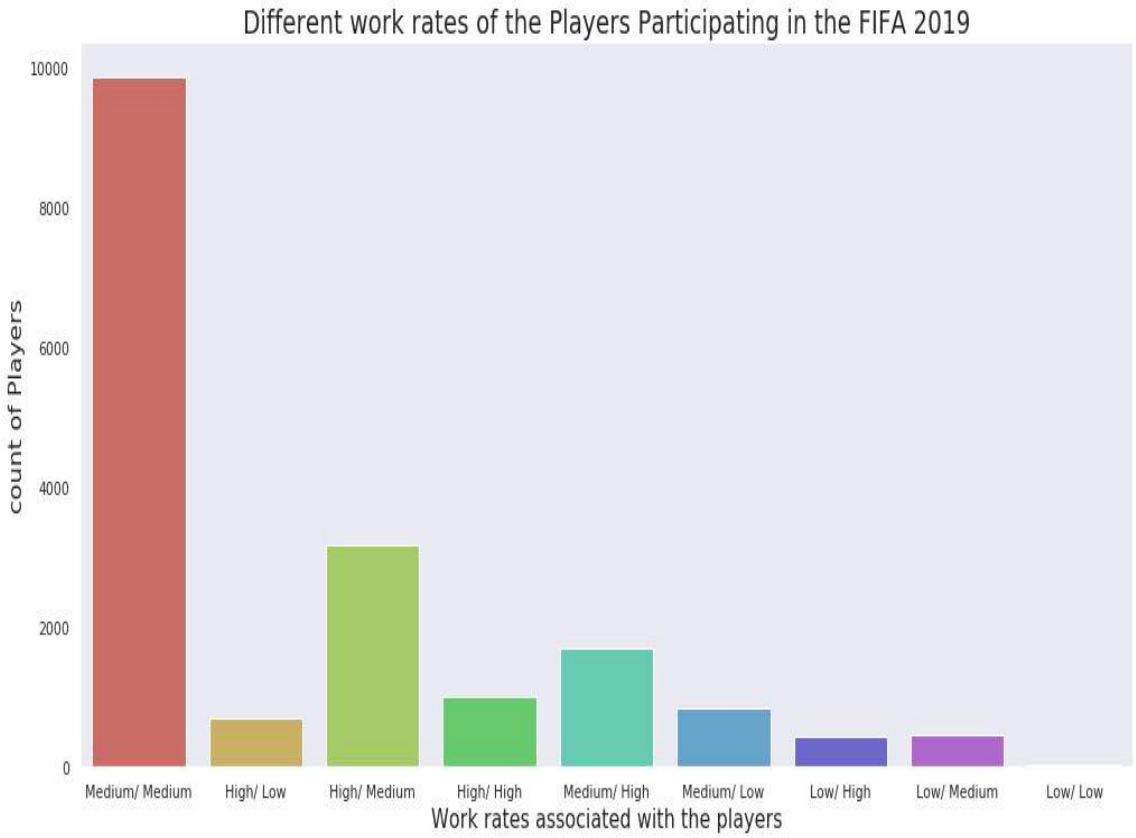

Figure 6.13 Workrates of Players

Dist plot:

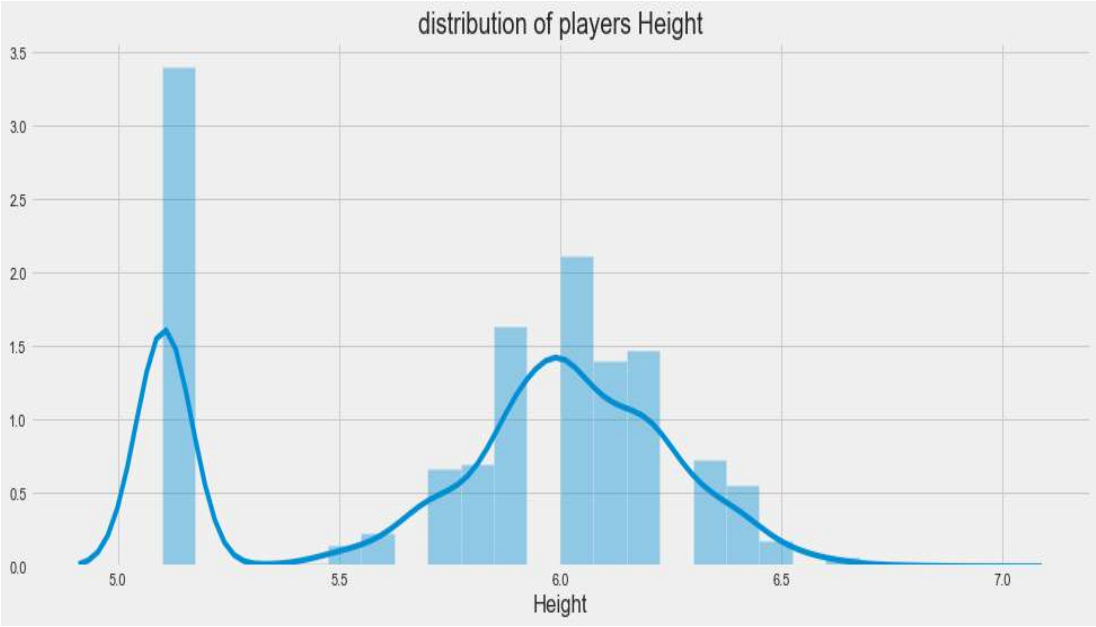

Figure 6.14 Height of the Players

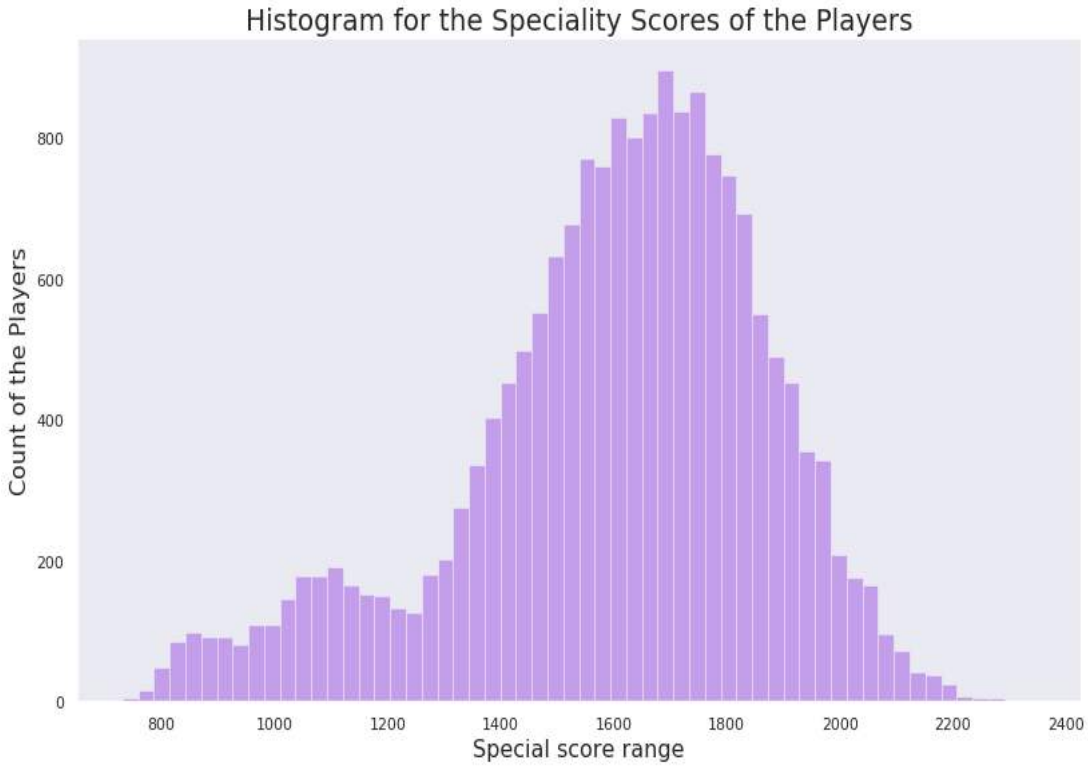

Figure 6.15 Speciality Scores of Players

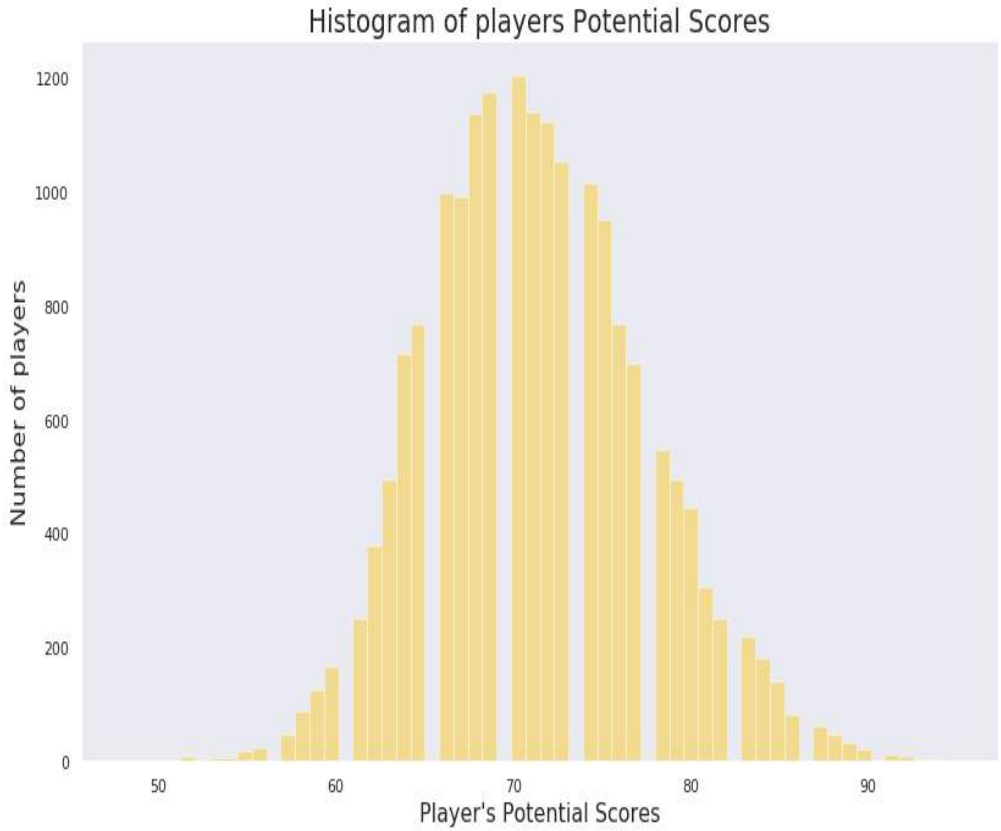

Figure 6.16 Player's Potential Scores

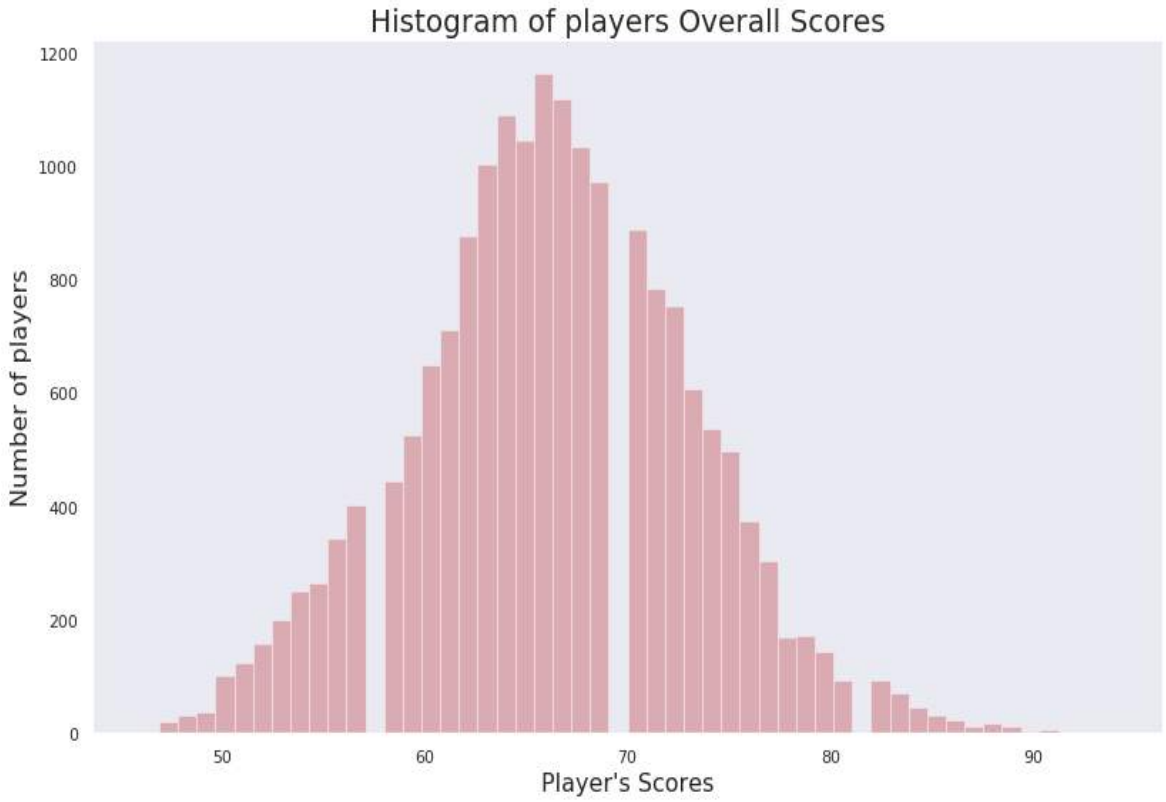

Figure 6.17 Player Overall Scores

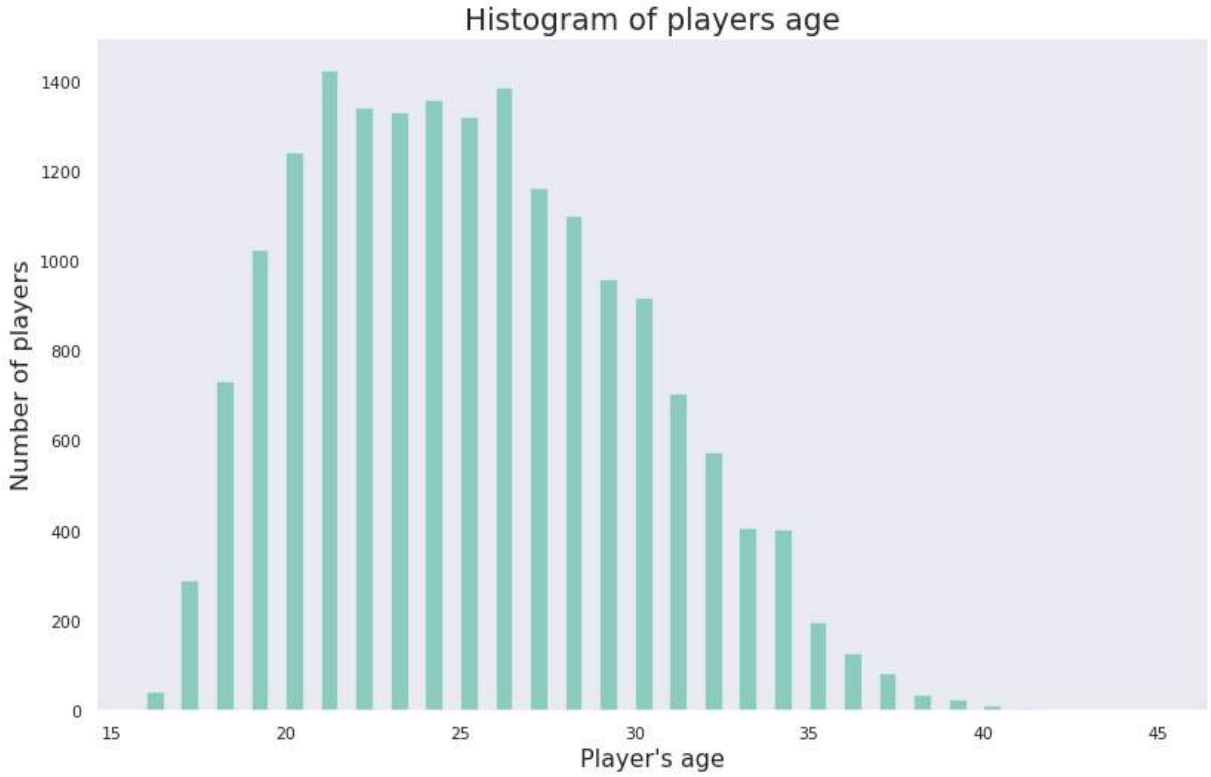

Figure 6.18 Age of Players

Bubble Plot:

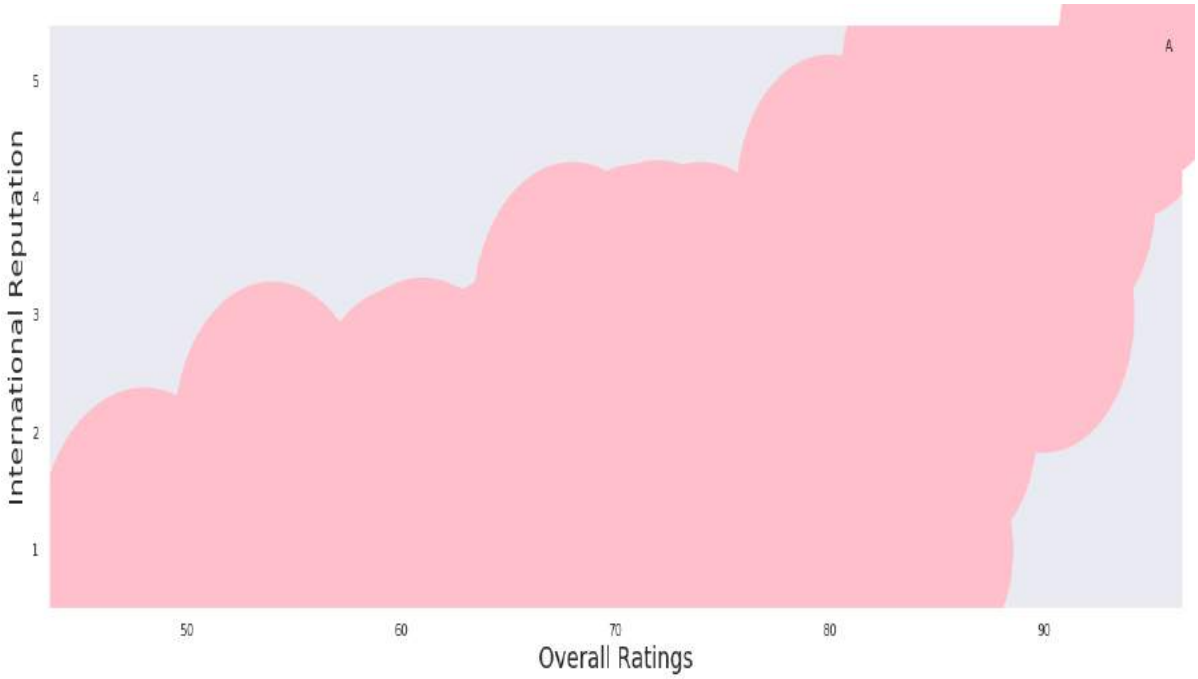

Figure 6.19 International Reputation and Rating

Violin plot:

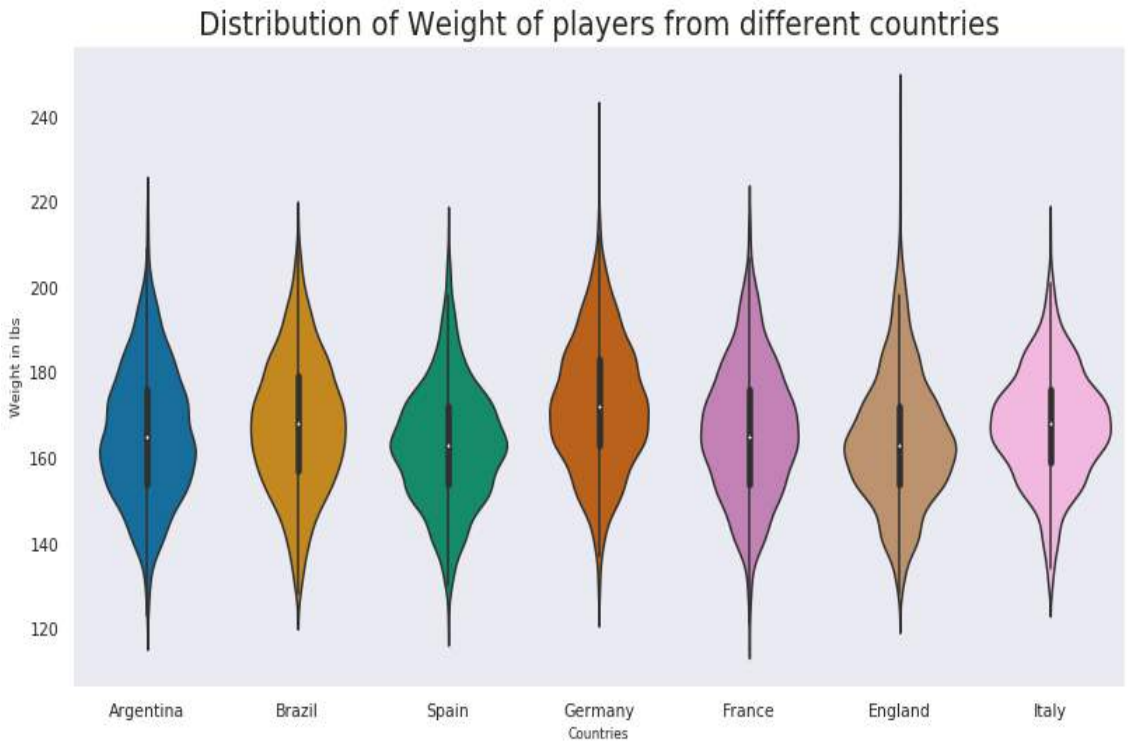

Figure 6.20 Weight of Players

7. CONCLUSION

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources – that would generate a proper System .While making the system, an eye has been kept on making it as user-friendly, and as flexible as possible. We conclude the Project with a High level Discussion of sports visualization research informed by our analysis.

8. FUTURE SCOPE AND ENHANCEMENT

The scope of the project includes that what all future enhancements can be done in the analyzing and predicting the data by using the algorithms such as recommendation (content) and clustering(k means).

- In future, artificial intelligence algorithms may be used in order to predict the structure of the game.
- Data-set can be revised whenever required.

BIBILIOGRAPHY

[1]. NumPy: https://python.org.in

[2] .Pandas, https://www.learnpython.org/en/Pandas_Basics

[3]. Matplotlib https://realpython.com

[4]. Dataset from source https://www.kaggle.com/datasets

[5]. sklearn, https://scikit-learn.org/stable/

[6] https://w3schools.com/machine learning

Text Book:

Core Python Programming , Wesley j.Chun, Second Edition,Pearson.