**Sports Analysis Dashboard**

Major Project Report

Submitted to the Centurion University

In partial fulfillment of requirements for the award of degree

**Bachelor of Technology in**

**Electronics and Communication Engineering**

by

G. Sumanth

211801131001

****

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT (CUTM-AP), VIZIANAGARAM,**

**535003**

****

**DEPARTMENT OF ELECTRONIC AND COMMUNICATION ENGINEERING**

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**VIZIANAGARAM CAMPUS**

## BONAFIDE CERTIFICATE

Certified that this project report **Sports Analysis Dashboard** is the bonafide work of “G. SUMANTH (211801131001)”, who carried out the project work under my supervision. This is to further certify to the best of my knowledge, that this project has not been carried out earlier in this institute and the university.

**External Examiner** **HEAD OF THE DEPARTMENT**

Mr. K. JOGINAIDU

Assistant Professor Department of ECE

**ABSTRACT**

Sports analytics has become a pivotal component in optimizing team performance, player strategies, and fan engagement. The *Sports Analysis Dashboard* is an interactive and comprehensive data visualization tool built using Python. It leverages libraries such as Matplotlib, Seaborn, and Plotly to present dynamic insights into sports performance metrics and trends.

This project aims to provide an intuitive interface for analyzing player statistics, team performance, and game outcomes. Using real-time or historical data, the dashboard supports various sports like cricket, football, and basketball, offering customizable visualizations, including heatmaps, bar charts, line graphs, and scatter plots.

Advanced techniques such as data preprocessing, feature extraction, and correlation analysis ensure data accuracy and relevance. The dashboard is tailored for coaches, analysts, and fans to identify patterns, predict outcomes, and make informed decisions. Its scalable design supports future integration with machine learning models for predictive analytics.

By combining Python's robust data manipulation capabilities with interactive visualization, this tool highlights the potential of data-driven decision-making in sports, fostering innovation and competitiveness in the field.

**INDEX**

**S.NO TITLE PAGE**

**Abstract 3**

**List of Figures 6**

**1 Introduction 7**

**2 Existing and Proposed System 8**

**3 System Requirements 8**

**3.1 Hardware Requirements 8**

**3.2 Software Requirements 8**

1. **Methodology 9**

**4.1 Modular Design 9**

**4.1.1 Data Acquisition 9**

**4.1.2 Data Processing 10**

**4.1.3 Implementation 10**

**4.1.4 Analysis 10**

**4.1.5 Visualization 11**

**4.2 What is Python 11**

**4.3.Modules 12**

**4.3.1 Pandas 12**

**4.3.2 Plotly. Express 13**

**4.3.3 folium 13**

1. **Data set 14**

**6. Implementation 15**

1. **Results 19**
2. **Conclusion 24**
3. **References 25**

**LIST OF FIGURES**

# **FIG. NO. TITLE PAGE NO**

**1 Match\_Name vs Innings\_Id 19**

**2 Innings\_Id vs Sixes 20**

**3 Innings\_Id vs Runs 20**

**4 Match\_Name vs Runs 21**

**5 Match\_Name vs Innings\_Id 21**

**6 Fours vs Match\_Name 22**

**7 Runs vs Match\_Name 22**

**1.INTRODUCTION**

Sports analytics is transforming the way games are played, analyzed, and experienced. With the exponential growth of data availability, analyzing sports statistics has evolved from manual record-keeping to sophisticated data-driven methodologies. This document outlines the development of a *Sports Analysis Dashboard*, a powerful tool designed to provide comprehensive insights into sports performance through Python-based data visualization.

The dashboard utilizes advanced data visualization libraries such as Matplotlib, Seaborn, and Plotly to present key performance indicators, game statistics, and trends in an interactive and user-friendly manner. It caters to a broad audience, including coaches, players, analysts, and sports enthusiasts, enabling them to identify patterns, predict outcomes, and optimize strategies.

By processing and visualizing both real-time and historical data, the dashboard supports various sports, offering customizable charts, graphs, and heatmaps to highlight critical insights. This tool not only simplifies complex datasets but also fosters informed decision-making in competitive scenarios.

The document explores the design, functionality, and impact of the Sports Analysis Dashboard, emphasizing the integration of technology and analytics in modern sports. It highlights the significance of Python's data handling and visualization capabilities in driving innovation and enhancing performance in the sports domain.

Data visualization is a cornerstone of sports analytics, transforming raw data into actionable insights. In the fast-paced world of sports, the ability to quickly interpret complex data can make the difference between winning and losing. Visualization techniques like heatmaps, trend lines, and performance matrices enable stakeholders to identify patterns, assess player efficiency, and monitor team dynamics effectively.

**2.Existing and Proposed System**

Current sports analytics tools are often limited by their lack of interactivity and adaptability. Many rely on static reports and manual analysis, making it challenging to extract meaningful insights efficiently. These tools often struggle with handling large datasets, lack customization options, and fail to provide real-time updates. Additionally, they are either proprietary or expensive, restricting their accessibility to a broader audience

The *Sports Analysis Dashboard* addresses these gaps by providing an interactive, real-time, and customizable platform for sports data visualization. Built using Python, it offers features such as dynamic visualizations, real-time data integration, and the ability to tailor analytics to specific sports or metrics. The dashboard supports multiple sports, including cricket, football, and basketball, ensuring versatility and scalability.

1. **SYSTEM REQUIREMENTS**

**3.1. Hardware** **Requirements:**

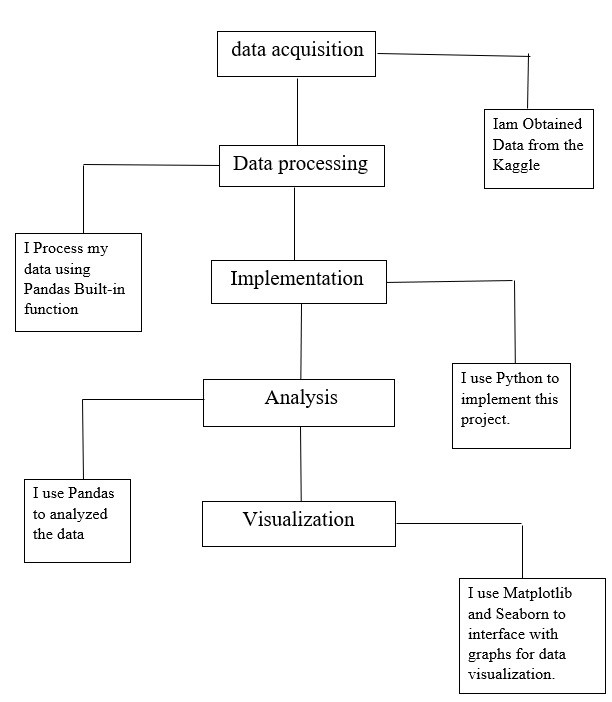
1. Processor-Pentium IV Processor
2. Speed-1.1GHZ
3. RAM-256MB
4. Storage-20GB

**3.2. Software Requirements:**

1. Operating system -XP/7/8/8.1/10
2. IDLE used-Python

**4. METHODOLOGY**

**4.1. Modular Design**



#### **4.1.1. Data Acquisition**

For collecting the raw data I used online sources like Kaggle, GitHub, Google Scholar, etc. Data acquisition is theprocess of measuring physical world conditions and phenomena such as electricity, sound,temperature and pressure. This is done through the use of various sensors which sample the environment’s analog signals and transform them to digital signals using an analog-to-digital converter.

#### **4.1.2. Data Processing**

Now the acquisited data is processed in excel for filtering the garbage values. I also computed the data using mathematical calculations in excel for a better result. Data preprocessing is the process oftransforming raw data into a useful, understandable format**.** Real-world or raw data usually has inconsistent formatting, human errors, and can also be incomplete. Data preprocessing resolves such issues and makes datasets more complete and efficient to perform data analysis.

#### **4.1.3. Implementation**

After processing the data, I imported the data in python environment for analysis. Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

#### **4.1.4. Analysis**

After implementation the data has to analyze for visualization. In analysis stage the user can clearly analyze the data in the form of rows and columns for a clear understanding.Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains.

#### **4.1.5. Visualization**

After analysis we come to the visualization stage where we used some python built-in-modules such as plotly where I produced interactive graphs for a better representation of the data.Data Visualization is used to communicate information clearly and efficiently to users by the usage of information graphics such as tables and charts. It helps users in analyzing a large amount of data in a simpler way. It makes complex data more accessible, understandable, and usable.

**4.2**. **What is Python?**

* Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.
* Python uses garbage collection and has dynamic typing. It supports a variety of programming paradigms, including procedural, objectoriented, and functional programming as well as structured programming (especially this). Due to its extensive standard library, it is frequently referred to as a "batteries included" language.
* Python was created by Guido van Rossum in the late 1980s to replace the ABC programming language, and it was originally made available as Python 0.9.0 in 1991.
* Python frequently causes programmers to fall in love due to the enhanced productivity it offers. The edit-test-debug cycle is extraordinarily quick because there is no compilation step.
* Python programs are simple to debug since a segmentation failure is never caused by a bug or incorrect input. Instead, the interpreter raises an exception when it finds a mistake. The interpreter prints a stack trace if the application doesn't catch the exception.
* Setting breakpoints, evaluating arbitrary expressions, inspecting local and global variables, stepping through the code one line at a time, and other features are all possible with a source level debugger. Python's ability to perform introspection is demonstrated by the debugger, which is developed in Python.

### **4.3. Modules**

For this project I’ve used the following modules:-

1.Pandas

2.Plotly. Express

3.Plotly. Graph

4.folium

#### **4.3.1. Pandas**

For the purpose of manipulating and analyzing data, the Python programming language has a software package called pandas. It includes specific data structures and procedures for working with time series and mathematical tables. It is free software distributed under the BSD license's three clauses. The word is derived from "panel data," a phrase used in econometrics to refer to data sets that contain observations for the same persons throughout a range of time periods. Python data analysis is a play on words in the name of the thing. When Wes McKinney worked as a researcher at AQR Capital from 2007 to 2010, he began creating the pandas that would eventually become famous.

**4.3.2. Plotly Express**

An interactive, open-source plotting toolkit for Python, plotly provides over 40 different chart types for a variety of statistical, financial, geographic, scientific, and three-dimensional use-cases.

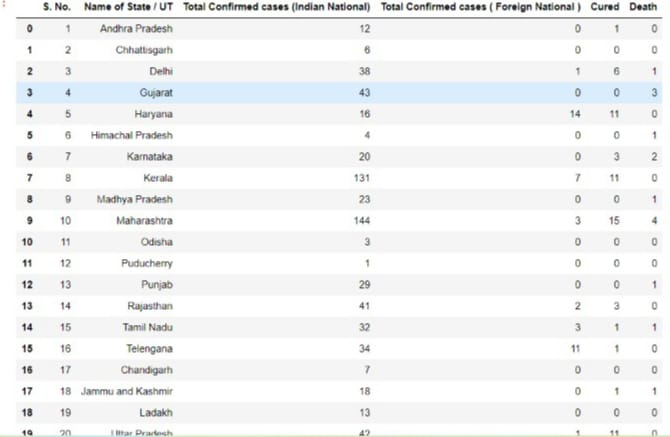
Plotly a python tool that is built on top of the Plotly JavaScript library (plotly.js), allows users to create stunning interactive web-based visualizations that can be viewed in Jupyter notebooks, saved to standalone HTML files, or used as a component of web applications that are entirely written in Python and served using Dash. To distinguish it from the JavaScript library, the plotly Python library is sometimes called "plotly.py."

**4.3.3Folium:**

Folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the leaflet.js library. Manipulate your data in Python, then visualize it in on a Leaflet map via folium. folium makes it easy to visualize data that’s been manipulated in Python on an interactive leaflet map. Folium is a powerful data visualization library in Python that was built primarily to help people visualize geospatial data. With Folium, one can create a map of any location in the world. Folium is actually a python wrapper for leaflet.js which is a javascript library for plotting interactive maps.

**5.DATASET**

So for performing any visualizations the main thing is the data. These data can be of any type i.e. raw data, filtered data, selective data etc. A data set (sometimes spelled dataset) is a group of data. In the case of tabular data, a data set relates to one or more database tables, where each row refers to a specific record in the corresponding data set and each column to a specific variable.



**6. IMPLEMENTATION**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the data (Replace 'world\_cup\_data.csv' with your dataset file)

data = pd.read\_csv('world\_cup\_data.csv')

# Scatter Plot: Runs vs. Wickets

plt.figure(figsize=(8, 6))

sns.scatterplot(x='Runs', y='Wickets', data=data, color='blue', s=100)

plt.title('Scatter Plot: Runs vs Wickets', fontsize=16)

plt.xlabel('Runs', fontsize=12)

plt.ylabel('Wickets', fontsize=12)

plt.grid(True)

plt.show()

# Pie Chart: Percentage of World Cups Won by Top Teams

top\_teams = data['Winner'].value\_counts().head(5)

plt.figure(figsize=(8, 8))

plt.pie(top\_teams, labels=top\_teams.index, autopct='%1.1f%%', startangle=140, colors=sns.color\_palette('pastel'))

plt.title('Pie Chart: Percentage of World Cups Won by Top Teams', fontsize=16)

plt.show()

# Bar Plot: Total Matches Played in Each World Cup

plt.figure(figsize=(10, 6))

sns.barplot(x='Year', y='Matches', data=data, palette='coolwarm')

plt.title('Bar Plot: Matches Played in Each World Cup', fontsize=16)

plt.xlabel('Year', fontsize=12)

plt.ylabel('Matches', fontsize=12)

plt.xticks(rotation=45)

plt.show()

# Violin Plot: Distribution of Runs for Each Year

plt.figure(figsize=(12, 6))

sns.violinplot(x='Year', y='Runs', data=data, palette='muted')

plt.title('Violin Plot: Distribution of Runs Over the Years', fontsize=16)

plt.xlabel('Year', fontsize=12)

plt.ylabel('Runs', fontsize=12)

plt.xticks(rotation=45)

plt.show()

# Box Plot: Runs Scored by Teams

plt.figure(figsize=(10, 6))

sns.boxplot(x='Winner', y='Runs', data=data, palette='Set2')

plt.title('Box Plot: Runs Scored by Winning Teams', fontsize=16)

plt.xlabel('Team', fontsize=12)

plt.ylabel('Runs', fontsize=12)

plt.xticks(rotation=45)

plt.show()

# Distribution Plot: Runs Scored

plt.figure(figsize=(8, 6))

sns.histplot(data['Runs'], kde=True, bins=15, color='purple')

plt.title('Distribution Plot: Runs Scored in World Cups', fontsize=16)

plt.xlabel('Runs', fontsize=12)

plt.ylabel('Frequency', fontsize=12)

plt.show()

# Histogram: Wickets Taken

plt.figure(figsize=(8, 6))

plt.hist(data['Wickets'], bins=10, color='orange', edgecolor='black')

plt.title('Histogram: Wickets Taken in World Cups', fontsize=16)

plt.xlabel('Wickets', fontsize=12)

plt.ylabel('Frequency', fontsize=12)

plt.show()

# Correlation Heatmap: Numerical Features

plt.figure(figsize=(10, 8))

correlation = data.corr()

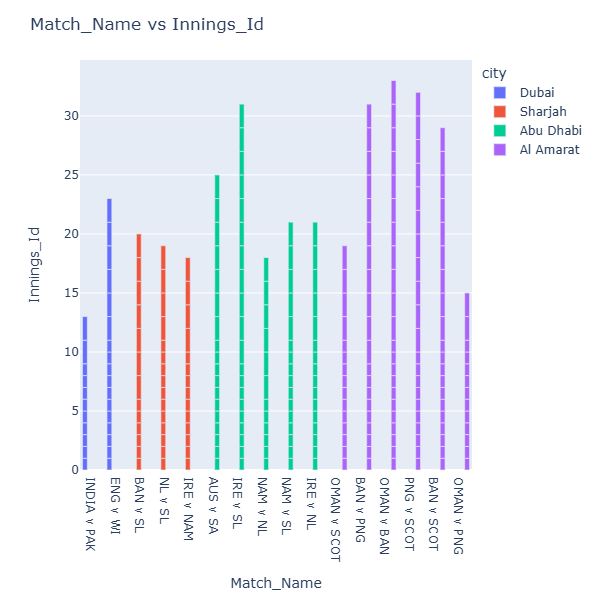
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

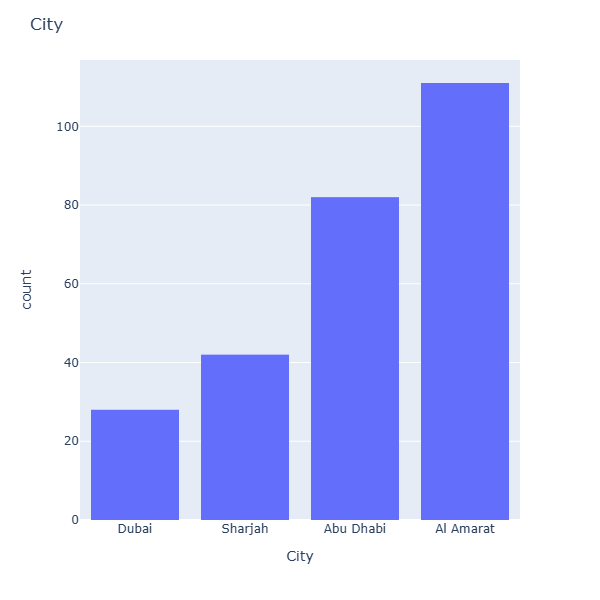
plt.title('Correlation Heatmap', fontsize=16)

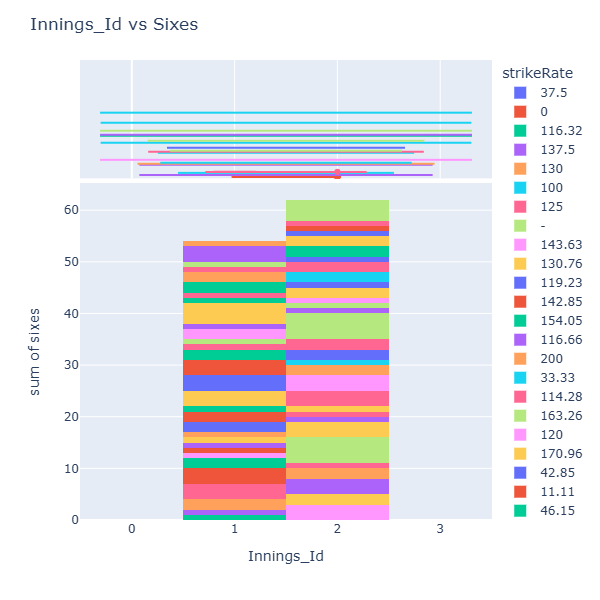
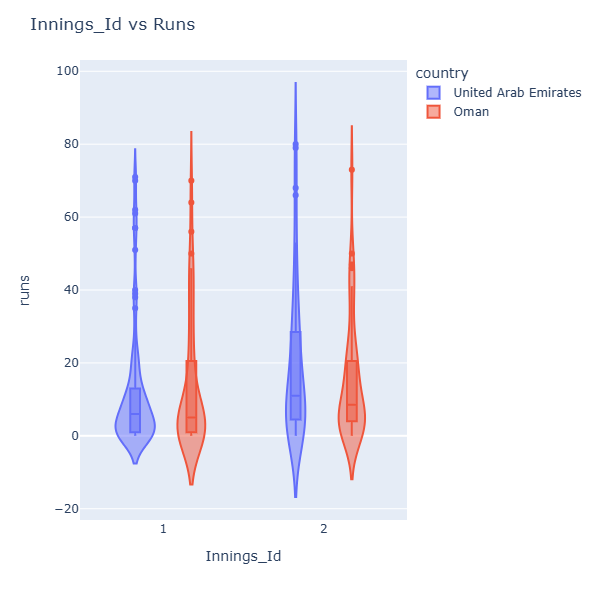
plt.show()

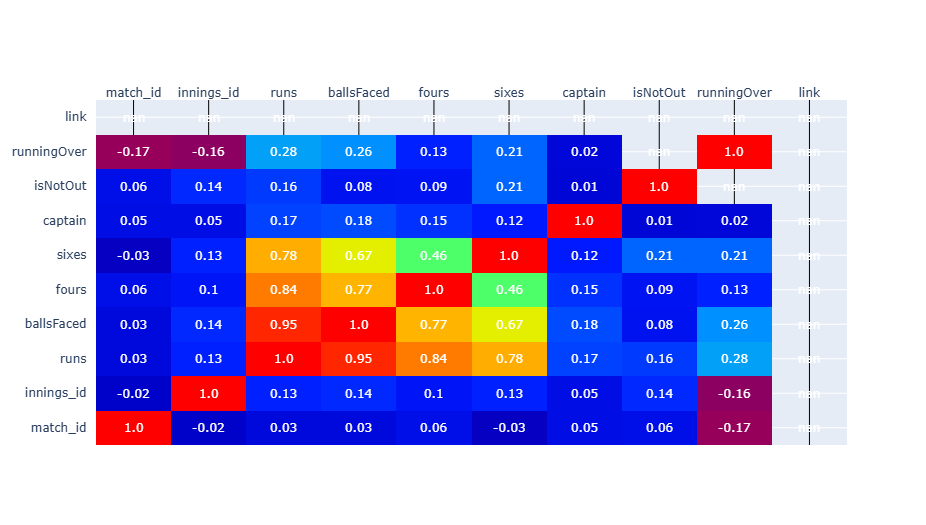
**7. RESULTS**

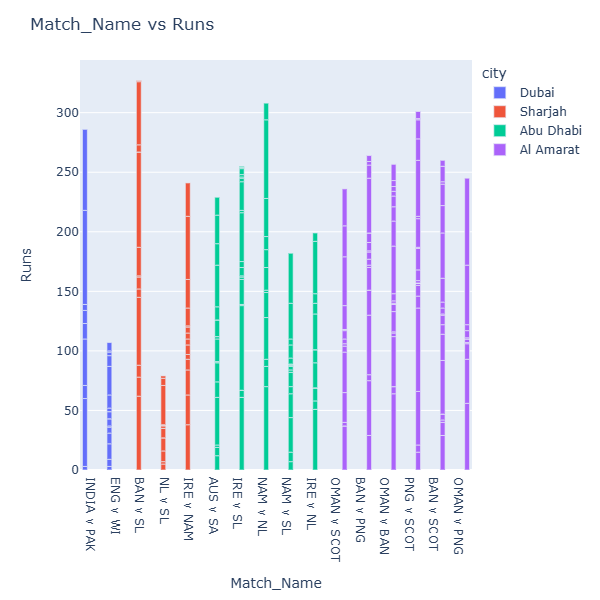
**Match\_Name vs Innings\_Id**

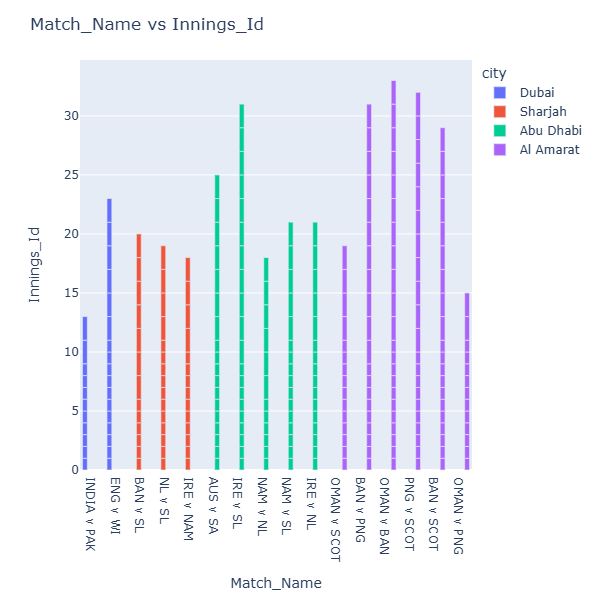


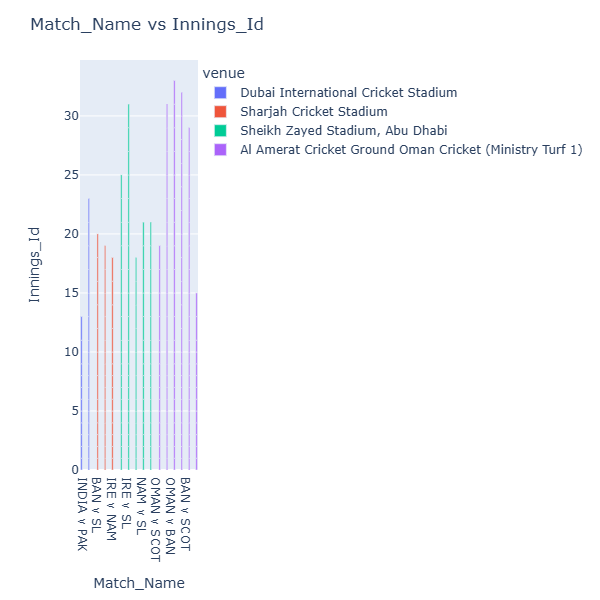
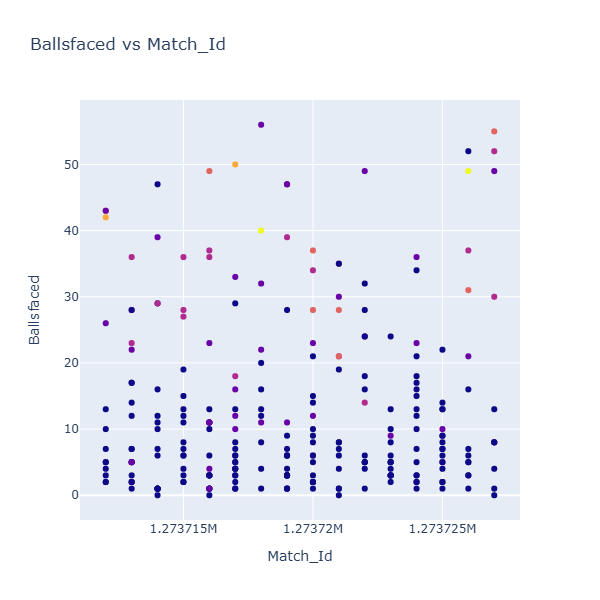
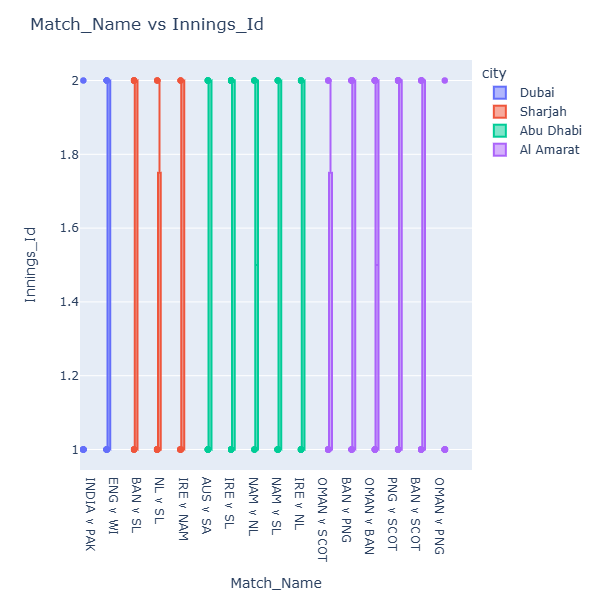
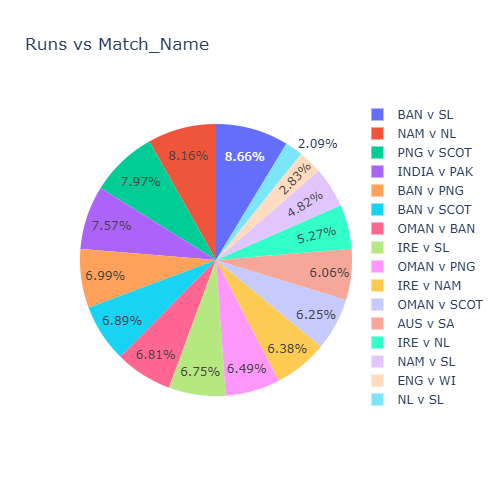
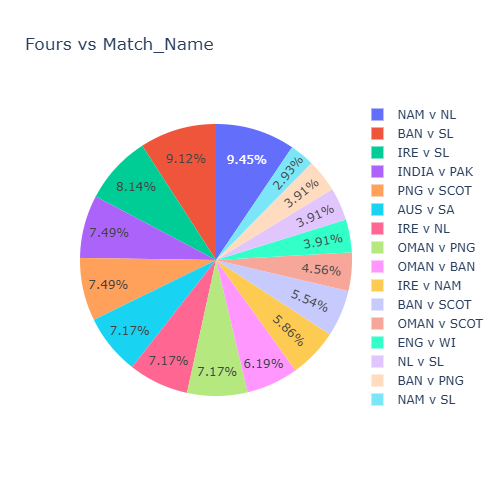










**8.CONCLUSION**

The *Sports Analysis Dashboard* represents a significant advancement in the field of sports analytics, offering an interactive, scalable, and user-friendly solution for analyzing complex data. By leveraging Python's robust data processing and visualization capabilities, the dashboard addresses the limitations of existing systems, such as lack of customization, real-time updates, and affordability.

This tool empowers coaches, analysts, and fans to gain actionable insights, optimize strategies, and enhance decision-making in various sports contexts. Its versatility across multiple sports and metrics ensures wide applicability, while its open-source nature makes it accessible to a diverse audience.

Additionally, the dashboard fosters innovation by simplifying the analysis of large datasets, making advanced analytics approachable for all stakeholders. Its adaptability ensures it can grow with the demands of the sports industry, incorporating features like live game updates and trend forecasting.

As sports analytics continues to evolve, the *Sports Analysis Dashboard* sets a strong foundation for future developments, including machine learning integration and predictive analytics. This innovative approach not only enhances performance and engagement but also underscores the transformative role of data-driven methodologies in modern sports.

**9.REFERENCES**

1. International Journal of Recent Technology and Engineering

(IJRTE) ISSN: 2277-3878, Volume-7 Issue-4S, November 2018

1. International Journal of Engineering Science InventionISSN (Online):

2319 – 6734, ISSN (Print): 2319 – 6726 [www.ijesi.org](https://jovian.ai/outlink?url=http%3A%2F%2Fwww.ijesi.org) Volume 3 Issue 6ǁ June 2014 ǁ PP.07-11

1. <https://coed.dypvp.edu.in/educational-resurgence-journal/documents/jan-2021/35-40.pdf>
2. <https://medium.com/codex/covid-19-analysis-with-python-b898181ea627>

5.DATA SET FROM KAGGLE

6. W3School (<https://www.w3schools.com/python>)

7.Plotly (<https://plotly.com/python/>)

8.Pandas (<https://pandas.pydata.org/>)