

**An Industrial Oriented Mini Project (CS755PC)**

on

**“ DATA ANALYTICS ON OLYMPIC DATASET- ATHLETE BMI  
DETECTION”**

**Submitted**

in partial fulfilment of the requirements for

the award of the degree of

**Bachelor of Technology**

in

**Computer Science and Engineering (Data Science)**

by

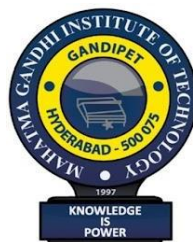
**KARNATI MANIDHEEPA**

**H.T.No: 21261A6723**

Under the guidance of

**Ms. D Deepika**

(Assistant Professor)



**DEPARTMENT OF EMERGING TECHNOLOGIES**

**MAHATMA GANDHI INSTITUTE OF TECHNOLOGY**

Gandipet, Hyderabad- 500075, Telangana (India) .

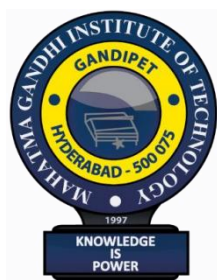
2024-2025

**MAHATMA GANDHI INSTITUTE OF TECHNOLOGY**  
**(AUTONOMOUS)**

**(Affiliated to Jawaharlal Nehru Technological University Hyderabad)**

**GANDIPET, HYDERABAD TELANGANA- 500075 (INDIA)**

**CERTIFICATE**



This is to certify that the project entitled “ **DATA ANALYTICS ON OLYMPIC DATASET-ATHLETE BMI DETECTION**” is being submitted by **KARNATI MANIDHEEPA** bearing **Roll No: 21261A6723** in partial fulfilment of the requirements for the award of **Bachelor of Technology in Computer Science and Engineering (Data Science)** is a record of bonafide work carried out by him/her under our guidance and supervision.

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

**Mentor**

**Ms. D Deepika**

Assistant Professor

Department of ET

**Head of the Department**

**Dr. M. Rama Bai**

Professor

Department of ET

**EXTERNAL EXAMINER**

## **DECLARATION**

This is to certify that the work reported in this project titled “**DATA ANALYTICS ON OLYMPIC DATASET-ATHLETE BMI DETECTION**” is a record of work done by me in the Department of Emerging Technologies, Mahatma Gandhi Institute of Technology, Hyderabad.

No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred to in the text. The report is based on the work done entirely by me and not copied from any other source.

**KARNATI MANIDHEEPA**

**(21261A6723)**

## ACKNOWLEDGEMENT

I would like to express my sincere thanks to **Dr. G. ChandraMohan Reddy, Principal, MGIT**, for providing the working facilities in college.

I wish to express our sincere thanks and gratitude to **Dr. M. Rama Bai, Professor and HoD**, Department of Emerging Technologies, MGIT, for all the timely support and valuable suggestions during the period of project.

I am extremely thankful to our Project Coordinator **Mr. Mallela Srikanth (Assistant Professor)**, and **Ms. Karuna Verma (Assistant Professor)**, Department of Emerging Technologies, MGIT, for their encouragement and support throughout the project.

I am extremely thankful and indebted to my internal guide **Ms. D. Deepika , Assistant Professor**, Department of Emerging Technologies, for her constant guidance, encouragement and moral support throughout the project.

Finally, I would also like to thank all the faculty and staff of the ET Department who helped me directly or indirectly, for completing this project.

**KARNATI MANIDHEEPA**  
**(21261A6723)**

# TABLE OF CONTENTS

TOPIC	PAGE NO.
Certificate	i
Declaration	ii
Acknowledgement	iii
List of Figures	vi
Abstract	vii
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement	1
1.2 Existing System	1
1.3 Proposed System	2
1.4 Project Scope & Objectives	3
1.5 System Requirements	3
1.5.1 Software Requirements	
1.5.2 Hardware Requirements	
<b>2. LITERATURE SURVEY</b>	<b>4</b>
<b>3. DATA ANALYTICS ON OLYMPIC DATASET- ATHLETE BMI DETECTION</b>	<b>5</b>
3.1 Algorithms	6
3.2 Required Modules/Libraries/Framework	8
3.3 Installation	11
3.4 Dataset	12
3.5 Home	13
3.6 Exploratory Data Analytics	15
3.7 Data Pre-Processing	16
3.8 Trends	17
3.9 Prediction	21
<b>4. CONCLUSION AND FUTURE SCOPE</b>	<b>23</b>
<b>5. REFERENCES</b>	<b>24</b>



## LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE
Figure 1.1	Existing System	2
Figure 3.1	Architecture of proposed system	5
Figure 3.2	Graph representing Linear Regression	6
Figure 3.3	Linear Regression function	6
Figure 3.4	Cost Function(J)	7
Figure 3.5	Anaconda Navigator Window	8
Figure 3.6	New Environment on Anaconda Navigator	9
Figure 3.7	Home Page	11
Figure 3.8	Exploratory Data Analytics	12
Figure 3.9	Data Preprocessing	15
Figure 3.10	Relationship b/w height & weight	15
Figure 3.11	Approximate no. of Male & Female participants	16
Figure 3.12	Participation trend in Summer and Winter	16
Figure 3.13	Women participation over the years	17
Figure 3.14	No. of medals won by Male & Female Athletes	17
Figure 3.15	Athletes with most medals	18
Figure 3.16	Countries with most medals & find whether given country is in the zero-medal list.	18
Figure 3.17	Prediction	19

## **ABSTRACT**

Athletes' physical fitness plays a crucial role in determining their performance and suitability for specific sports, with Body Mass Index (BMI) serving as a vital indicator. The "Data Analytics on Olympic dataset-Athlete BMI Detection" project aims to assess the compatibility of athletes for various sports by analyzing their BMI, ensuring they meet the necessary fitness levels required for optimal performance. The study addresses the growing need for objective and data-driven decision-making in sports selection, highlighting the role of BMI as a benchmark for categorizing athletes and optimizing their potential contributions to their respective fields.

The project employs advanced machine learning algorithms to analyze BMI data and predict athletes' suitability for specific sports disciplines. By combining predictive and descriptive analytics, the system identifies athletes who meet predefined fitness thresholds, ensuring that only those who meet the required standards are selected for further consideration. The descriptive analysis visualizes the distribution and trends of BMI within the dataset, helping to uncover patterns and outliers. The dataset includes BMI and demographic data, which enables comprehensive analysis and correlation studies to refine the predictions, ultimately improving the accuracy of athlete assessments.

By integrating machine learning techniques, the project offers a significant advantage for coaches, sports committees, and fitness professionals, enabling them to make data-driven decisions. It not only aids in athlete selection but also provides valuable insights for fitness management, allowing for the tailoring of training regimens to enhance performance. Moreover, the findings from the analysis contribute to broader sports science research, deepening the understanding of BMI's impact on athletic performance and advancing methodologies used in talent scouting and fitness evaluations.



# **1. INTRODUCTION**

The Modern Olympic Games or Olympics are leading international sports events featuring summer and winter sports competitions in which thousands of athletes from around the world participate in a variety of competitions. The 'modern Olympics' comprises all the Games from Athens 1986 to Rio 2016. The Olympic Games are considered the world's foremost sports competition with more than 200 nations participating.

The Olympics is more than just a quadrennial multi-sport world championship. It is a lens through which to understand the global history, including shifting geopolitical power dynamics, women empowerment and evolving values of society.

Therefore, it is very much essential to analyze the Olympics data to determine the various relationships between the athletes involved and their participation in the events.

## **1.1 PROBLEM STATEMENT**

The problem revolves around knowing the trends and relationship between the attributes of participated athletes. For this purpose, the Athlete events dataset containing a total of 15 attributes, describing about the Athlete object has been considered.

It also includes a basic predictive analysis on BMI values of athletes and their concerned sport. For this purpose Athlete BMI dataset has been imported. The results must be embedded into an application (or) interface.

## **1.2 EXISTING SYSTEM**

Usually, the analytics part is done using Jupyter Notebook, Google colab and other tools. In those environments, cells are present which contains the code that produces output on run command. The figure 1.1 represents the existing system.

A serious drawback is the lack of a proper interface, that would make results even more appealing to look at.

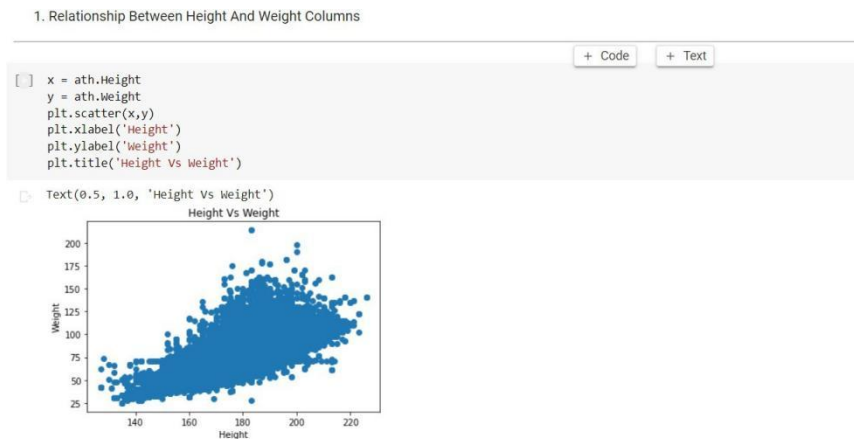


Figure1.1 Existing system

## DRAWBACKS OF EXISTING SYSTEM

- No easy interface
- Code Visibility
- Not appealing to the users

## 1.3 PROPOSED SYSTEM

A simple application (or) Graphical User Interface that would act as an intermediate between the results and the users.

For the code Python is used and for building the application streamlit(python) has been used and for analytics libraries such as Numpy, pandas, matplotlib etc(python) have been implemented accordingly.

The proposed system would have five main components:

Home, Exploratory Data Analysis, Data preprocessing, Trends and Prediction.

## 1.4 PROJECT SCOPE & OBJECTIVES

### SCOPE

- The project is an interactive application that would help to know about the Olympics more.
- It's especially useful to the Olympics managing authorities.
- It can lay foundations to build more Data Analytics applications that would easy to present the results and user-friendly.

## **OBJECTIVES**

- How weight of an athlete is dependent on his/her height?
- The total number of medals won by Male and female athletes.
- Determining the participation trend in the Summer and Winter Seasons
- Which Countries have the most medals?
- Name the athletes with most medals.
- Determine the countries winning the most gold medals in a specific year.
- Predict the weight of an athlete, given the height?
- Predict the sport a person is apt for, depending on his/her BMI values.
- Analyze women participation over the years.

## **1.5 SYSTEM REQUIREMENTS**

### **1. Hardware requirements**

- Processor : Pentium V (or) Higher
- RAM : 1GB
- Space on hard disk : minimum 512MB

### **2. Software requirements**

- Web browser/engine: Google chrome (or) IE.
- Python libraries:(matplotlib, plotly, numpy, pandas, re, sklearn,seaborn)  
Anaconda environment
- PC running with windows 7 (or) more and streamlit framework

## 2. LITERATURE SURVEY

[1] **Yamunathangam D., Kirthicka G., Shahanas Parveen** published a paper titled ‘Performance Analysis in Olympic Games using Exploratory Data Analysis Techniques’. This paper focuses on applying advanced exploratory data analysis (EDA) techniques to evaluate the performance of athletes in the Olympic Games. The study leverages visualization tools to uncover trends and patterns in Olympic data, such as medal distribution, country-wise performance, and athlete statistics over the years. By utilizing effective visualization methods, the study provides valuable insights into historical Olympic performances, offering a user-friendly approach to understanding complex datasets.

[2] **Saul Buentello** published a paper ‘120 Years of Olympic Games — How to Analyze and Visualize the History with R’. This study presents a comprehensive analysis of 120 years of Olympic history using R programming. It includes detailed visualization techniques such as heatmaps, bar charts, and time-series plots to represent key metrics, including athlete participation, medal counts, and country-wise trends. The author focuses on leveraging R’s capabilities for statistical computing and data visualization, creating compelling graphical representations to make the data accessible.

[3] **Rahul Pradhan, Karthik Agrawal, Anubhav Bag** published a paper ‘Analyzing Evolution of the Olympics by Exploratory Data Analysis using R’. This paper explores the evolution of the Olympics using EDA techniques in R, with a focus on trends in athlete demographics, event popularity, and country-level performance. The authors provide a detailed methodology for data analysis, including data cleaning and feature extraction, to derive meaningful insights. The findings are presented using clear, structured visualizations that highlight patterns and anomalies in the Olympic data. Documentation ensures reproducibility, making it useful for practitioners and researchers.

### 3. DATA ANALYTICS ON OLYMPIC DATASET-ATHLETE BMI DETECTION

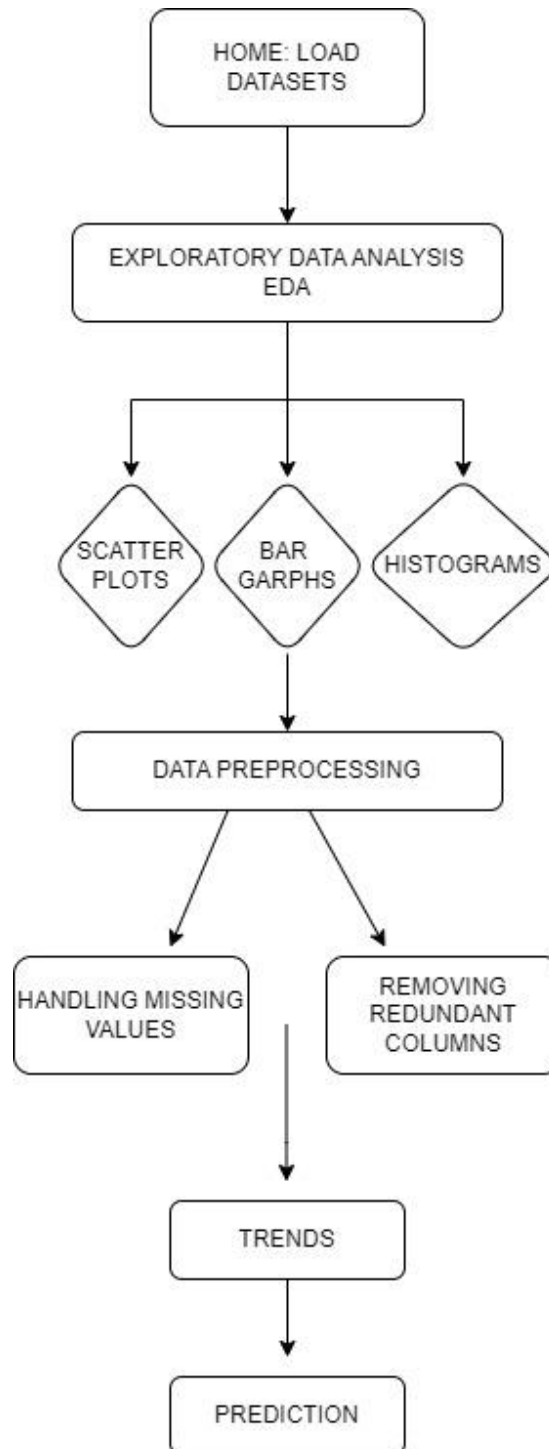


Figure 3.1 Data Analytics on olympic dataset

### 3.1 Algorithm & Methodologies

#### ➤ Linear regression:

- Figure 3.2 represents the graph of linear regression.
- Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task.
- Regression models a target prediction value based on independent variables.
- It is mostly used for finding out the relationship between variables and forecasting.
- Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.
- Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x).
- So, this regression technique finds out a linear relationship between x (input) and y (output). Hence, the name is Linear Regression.
- In the Figure 3.2, X (input) is the work experience and Y (output) is the salary of a person.
- The regression line is the best fit line for our model.

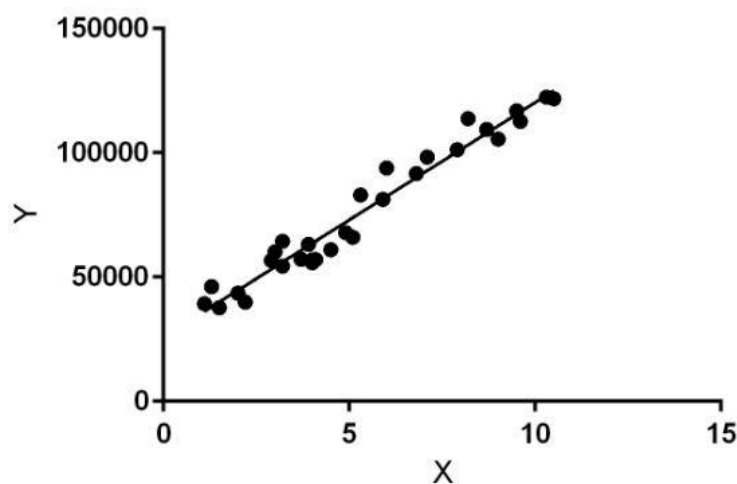


Figure 3.2 Graph representing Linear regression

Hypothesis for linear regression:

$$y = \theta_1 + \theta_2 \cdot x$$

Figure 3.3 Linear Regression function

In the figure 3.3, while training the model we are given :

x: input training data (univariate – one input variable(parameter))  
y: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best  $\theta_1$  and  $\theta_2$  values.

$\theta_1$ : intercept

$\theta_2$ : coefficient of x

Once we find the best  $\theta_1$  and  $\theta_2$  values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

How to update  $\theta_1$  and  $\theta_2$  values to get the best fit line ?

**Cost Function (J):**

By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the  $\theta_1$  and  $\theta_2$  values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y). The figure 3.4 represents the cost function(J).

$$\text{minimize } \frac{1}{n} \sum_{i=1}^n (\text{pred}_i - y_i)^2$$
$$J = \frac{1}{n} \sum_{i=1}^n (\text{pred}_i - y_i)^2$$

Figure 3.4 Cost Function (J)

Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE)

between predicted y value (pred) and true y value (y).

**Gradient Descent:**

To update  $\theta_1$  and  $\theta_2$  values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random  $\theta_1$  and  $\theta_2$  values and then iteratively updating the values, reaching minimum cost.

## 3.2 Required Modules/ Libraries/ Framework:

- Numpy
- Pandas
- Streamlit
- Matplotlib
- Plotly
- Scikit-Learn

## 3.3 INSTALLATION:

### 3.3.1 ANACONDA NAVIGATOR

Create a New Python Environment:

- Go to the Environments tab on the left side bar as shown in Figure 3.5.

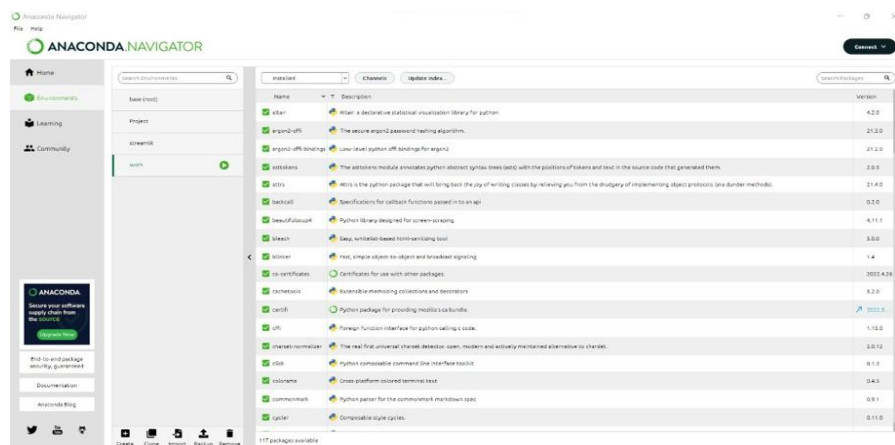


Figure 3.5 Anaconda Navigator

- Click on Create button at the bottom to create a new environment.
- Name the environment Olympics\_env and select **Python 3.7** version as shown in Figure 3.6 .



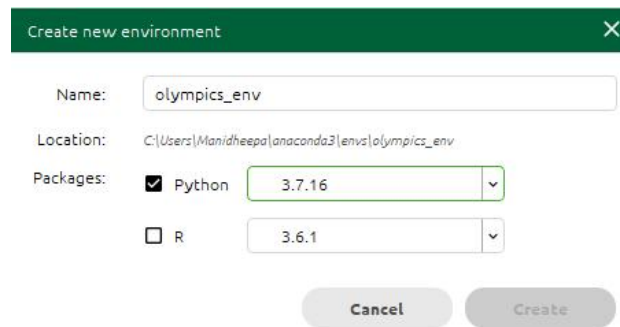


Figure 3.6 Create New Environment in Anaconda Navigator

- Click create to setup the Environment.
- Activate the Environment:
  - In Anaconda Navigator, click the **Play** button next to the environment name and select **Open Terminal**.
  - Activate the Environment using the command:  
**conda activate olympics\_env**
- Install the required Libraries:
  - Run the following commands one-by-one in the terminal:  
**pip install streamlit**  
**pip install pandas**  
**pip install numpy**  
**pip install matplotlib**  
**pip install plotly**  
**pip install seaborn**  
**pip install scikit-learn**
  - Or you can install all the libraries together using the following command:  
**pip install streamlit pandas numpy matplotlib plotly seaborn scikit-learn**
- Download the datasets:
  - Datasets are imported in csv file format
  - We will be using two datasets, Athlete Event dataset and the Athlete BMI dataset.
  - Download the **Athlete Events dataset** from Kaggle and place it in a directory on your system.
  - It contains total of 277116 row tuples, mapped over 15 attributes.
  - Each row corresponds to an individual athlete competing in an individual Olympic Event.

- The **Athlete BMI dataset** has been manually created by taking three attributes:
  - **Athlete name**
  - **Athlete BMI**
  - **Athlete concerned sport (represented in the form of integers)**
- The sport and their integer values are as follows:
  - **Marathon -1**
  - **Basketball -2**
  - **Rugby -3**
  - **Shot put- 4**
- The Dataset consists of 20 tuples.
- Setup Streamlit Application:
  - Open any Text editor (**VSCode, Notepad++ ,etc.**) and write the code.
  - Save the code with .py extension such as **app.py** in the same folder where you have the two datasets.
- Run the Streamlit Application:
  - Open the Terminal from Anaconda Navigator or use a system terminal.
  - Navigate to the folder where app.py is located using the terminal :  
**cd path/to/your/project**
  - Run the Streamlit application:  
**streamlit run app.py**

The Application contains 5 pages, namely:

1. Home
2. Exploratory Data Analysis
3. Data Preprocessing
4. Trends
5. Prediction

### 3.4 HOME:

- In figure 3.7, Home contains the basic introduction on the Olympics application, along with its logo of five connected circles.
- It also contains the navigation bar linking the five pages.

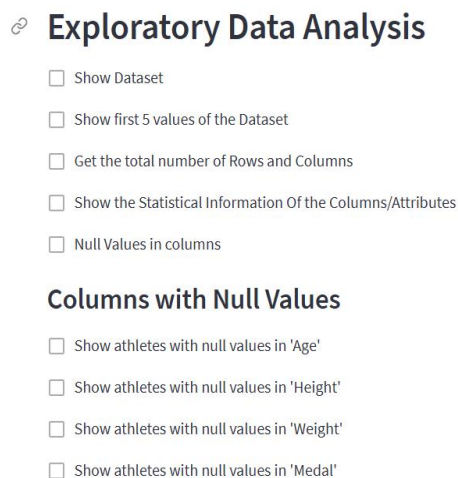


Figure 3.7 Home page

- There are two load buttons:
  1. **Load Athlete Events dataset-**
    - On clicking this button, it loads the dataset and prints the message ‘Loaded Successfully’.
    - It also presents the users with a flowchart that guides them during the exploration of the application.
  2. **Load Athlete BMI dataset-**
    - Body Mass Index (BMI) is the value derived from the mass(weight) and height of a person.
    - BMI is defined as the body mass divided by the square of body height, and is expressed in units of  $\text{kg/m}^2$  resulting from mass in kg and height in metres.
    - The BMI values of athletes and corresponding sport are used to predict the appropriate sport for a testcase.
    - The button on click gives a flowchart that guides users to the prediction page.

### 3.5 EXPLORATORY DATA ANALYSIS:

- EDA is an approach to analyse data using visualization techniques.
- It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations.
- The next two pages Data processing and trends are also a part of EDA.
- For simple reasoning, let's assume this page would just explore the dataset(Athlete Events) and provide statistical information to the user.
- In figure 3.8, The EDA page contains 9 checkboxes:



**Exploratory Data Analysis**

- ☐ Show Dataset
- ☐ Show first 5 values of the Dataset
- ☐ Get the total number of Rows and Columns
- ☐ Show the Statistical Information Of the Columns/Attributes
- ☐ Null Values in columns

**Columns with Null Values**

- ☐ Show athletes with null values in 'Age'
- ☐ Show athletes with null values in 'Height'
- ☐ Show athletes with null values in 'Weight'
- ☐ Show athletes with null values in 'Medal'

Figure 3.8 Exploratory Data Analysis

- The show dataset checkbox, when clicked presents the user with the entire dataset which can be viewed using scroll bars.
- The show first five values of dataset, returns the head of the dataset. For this, `df.head()` has been used.
- The third checkbox, Get total Rows and columns , uses the shape function to return total number of data tuples and attributes.
- The show statistical information, shows the information of column/attributes, when clicked uses the `df.describe()` function to present the user with statistical information such as Count, Mean, Standard Deviation, 25%, 50%, 75%, Max & Min of the numeric attributes in the dataset.

- The NULL values in columns checkbox gives the total NaN values in the dataset. For this, `df.isnull().sum()` operation has been performed.
- The Athletes with NULL values in age, checkbox displays the athletes with NULL values in age. For this, `is.null()` operation has been performed.
- The Athletes with NULL values in height, displays the athletes with NULL values in height. For this, `is.null()` operation has been performed.
- The Athletes with NULL values in weight, displays the athletes with NULL values in weight. For this, `is.null()` operation has been performed.
- The Athletes with NULL values in medal, displays the athletes with NULL values in medals. For this, `is.null()` operation has been performed.

### **3.6 DATA PRE-PROCESSING:**

From the above exploration, it was clear that the dataset has these issues:

1. Presence of NaN values in the columns Age, Height, Weight and Medal which preclude efficient data analysis.
  2. The medal column should be converted to Numeric type for analysis part.
  3. The deletion of attribute 'Games' on an account of redundancy. Since it is a concatenation of Year and Season columns.
- From these observations it is clear that data cleaning should be performed.
  - Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistency in the data.
  - The various methods to handle missing values include:
    - Ignoring the tuple
    - Manually filling the missing values
    - Using global constant to fill the missing value
    - Use attribute mean for numeric values and attribute mode for categorical values.
  - In the project, we considered replacing NaN with mean values of Age, Height and Medal.

- NaN values of Medals are replaced with 0 which indicates no medal and also the Medal column has been converted to numeric form, with 1 representing Gold, 2 representing Silver and 3 with Bronze.
- As in Figure 3.8, the page consists of 7 checkboxes.
- The first three removes NaN values in Age, Height and Weight , the fourth performs operations on the Medal column. One can check the updated NaN values using the fifth checkbox.

## Data Preprocessing

- ☐ Remove Null Values in Age Column
- ☐ Remove Null Values in Height Column
- ☐ Remove Null Values in Weight Column
- ☐ Convert Medals to Numeric and Remove Null Values
- ☐ Remove redundant 'Games' column

Figure 3.9 Data Preprocessing

### 3.7 TRENDS:

- Trends determine the relationship between attributes and are helpful in answering questions presented in the Objectives session.
- Consider the query ‘**Analyze the relationship between height and weight of an Athlete**’, for this purpose a simple scatter plot is used to check the correlation between the two attributes.

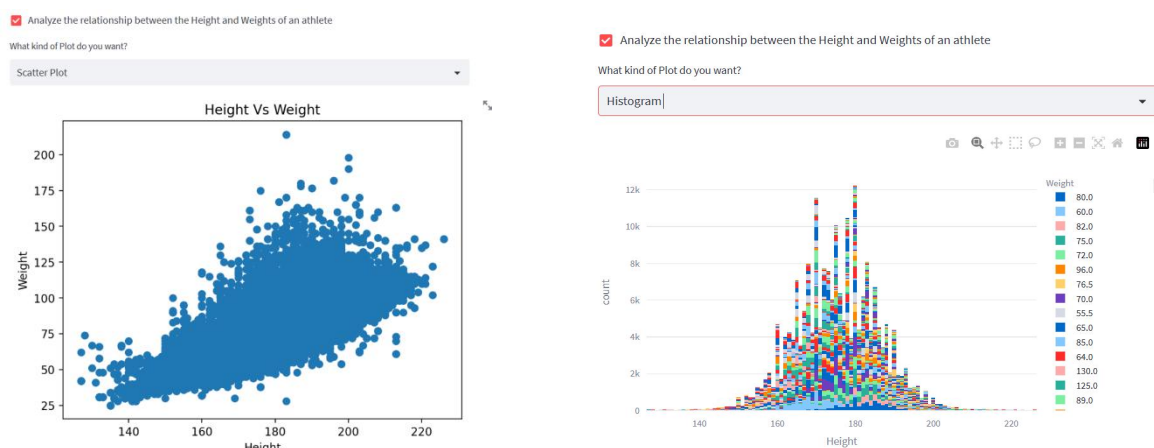


Figure 3.10 Relationship b/w height vs Weight

- Consider the query “**Approximate no. of Males and Females participated in the Olympics**”, the answer to which is found using the bargraph from matplotlib, with x-axis representing the gender and y-axis representing the count.

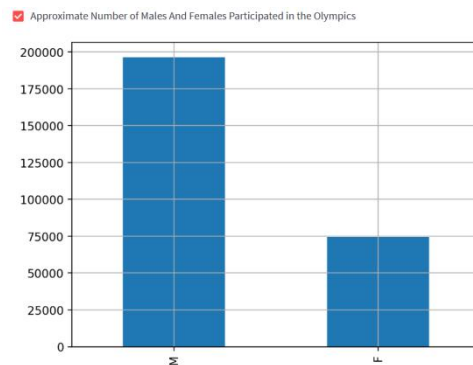


Figure 3.11 Approximate no. of Male and Female Participants

- Consider '**Determining the participation trend in the Summer and Winter seasons**'.
  - The answer is found using the histogram from plotly express, with x-axis representing season and y-axis representing the count.
  - Result: Summer Olympics see more participation than Winter.

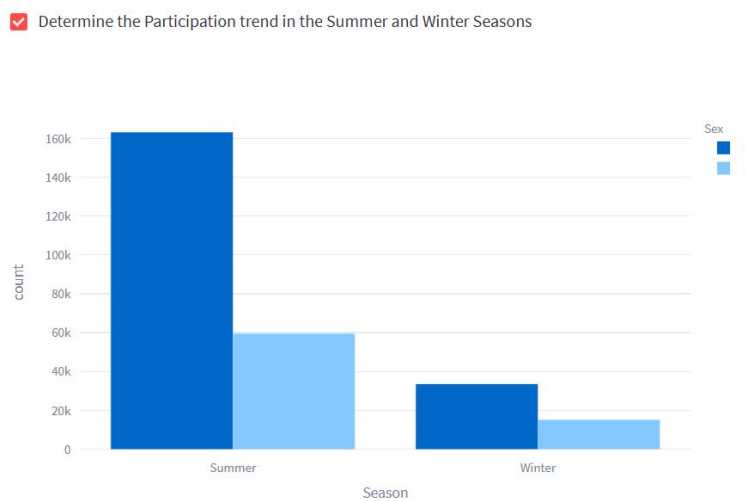


Figure 3.12 Participation trend in Summer and Winter

- Consider the query **‘Women participation over the years’**.
  - The answer is illustrated in the form of histogram from plotly express.
  - The x-axis representing the year and y-axis representing the count
  - Result: The graph clearly shows that the women participation is increasing over years.

✓ Women Participation over the years

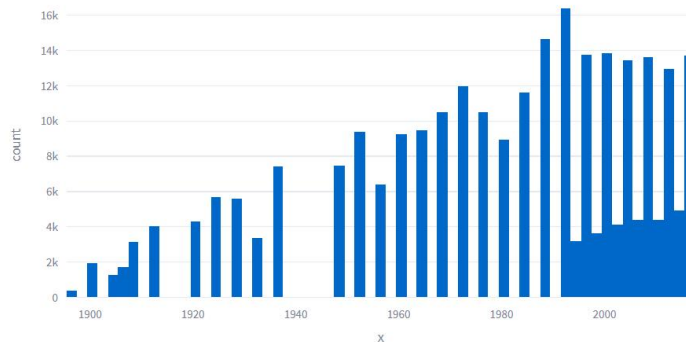


Figure 3.13 Women participation over the years

- The query **‘No. of Medals won by Male and Female Athletes’**.

✓ Number of Medals Won by M and F

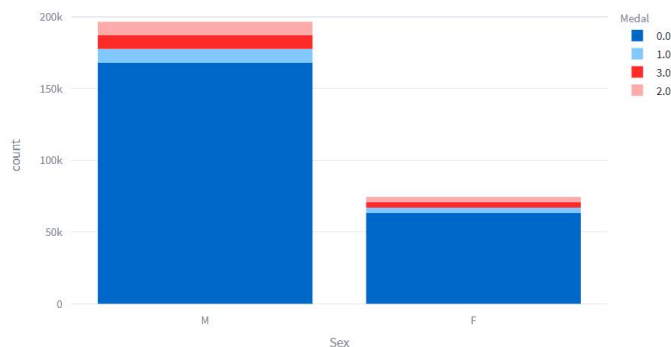


Figure 3.14 No. of medals won by Male and Female Athletes

- Consider the image, showing medals won at each levels ( 0 being no medal, 1 gold, 2 silver, 3 bronze)
- Result: Men have won 9625 gold, 9524 Silver, 9381 Bronze whereas Women have won 3747 Gold, 3771 Silver, 3735 Bronze.



- Consider the query ‘Athletes with most medals’.

☒ Athletes with Most Medals

Name	Total
Michael Fred Phelps, II	28
Larysa Semenivna Latynina (Diriy-)	18
Nikolay Yefimovich Andrianov	15
Edoardo Mangiarotti	13
Borys Anfiyanovych Shakhlin	13
Takashi Ono	13
Ole Einar Bjrndalen	13
Sawao Kato	12
Paavo Johannes Nurmi	12
Aleksey Yuryevich Nemov	12

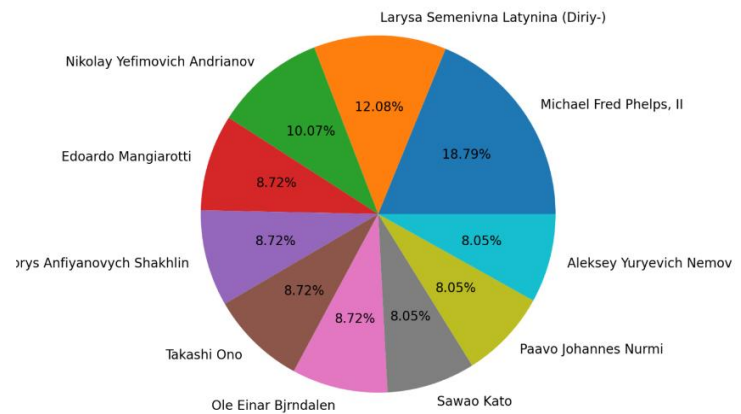


Figure 3.15 Athletes with most medals

- ‘Countries with most medals’ the answer be found using get dummies() function from the pandas. Given the list 10 countires with most medals, this checkbox is also used to find out whether a country is in 0 medal list.

☒ Countries with Most Medals

	0
0	United States
1	Soviet Union
2	Germany
3	Great Britain
4	France
5	Italy
6	Sweden
7	Australia
8	Canada
9	Hungary


Find whether your country is in the Zero-Medal list?

Enter

Figure 3.16 Countries with most medals & find whether given country is in the zero-medal list.

### 3.8 PREDICTION:

- In the Figure 3.15, The prediction page allows the users to input relevant data and generate predictions based on the trained model.
- In this context, predictions could relate to BMI classification, sport-specific predictions based on historical data trends in the dataset ( Athlete\_bmi\_dataset).
- The page will most likely include a form where users can input specific details such as:
  - Athlete Height
  - BMI value
- The predictions deal with, predicting the weight of an athlete based on the given input 'Height' and the suitable sport for an athlete based on the BMI value.

 **Prediction**

**Predict the Weight of an athlete with his/her Height**

Enter the Height

0.00 - +

Predict Weight

**Predict the suitable Sport with the BMI values**

☐ Show Athlete BMI Dataset

Enter BMI

0.00 - +

Results

#### 3.17 Prediction

- It also consists of a checkbox, '**Show Athlete BMI dataset**' which when clicked displays the dataset.

## 4. CONCLUSION & FUTURE SCOPE

### Conclusion

The Project provides valuable insights into athlete performance and trends based on historical data. By employing ML models, Data preprocessing, and visualization techniques, we have been able to extract meaningful patterns from the Olympic athlete dataset, such as performance trends, BMI distributions and sport-specific statistics.

The interactive application built using Streamlit offers users a seamless experience in exploring the dataset, generating predictions, and understanding the correlation between athletes' physical attributes and their sporting outcomes.

The project successfully demonstrates how historical data can be leveraged to make data-driven predictions and decisions, further reinforcing the importance of analytics in sports industry. The implemented features- ranging from Exploratory data analysis to trend visualization and prediction- highlight the potential of data science in enhancing our understanding of sports performance.

### Future scope

- **Real-Time Data Integration** : The application can be extended to include real-time data streaming from live sports events.
- **Integration with Sports Performance Platforms:** The project could be integrated with existing sports performance platforms or wearable technology to track athletes' real-time physical metrics and provide predictive insights based on live data feeds.
- **Mobile App Integration:** Developing a mobile version of the application can increase accessibility, enabling athletes, coaches, and sports analysts to use the predictive model on the go.

## **5. References**

- [1] S. Arampatzis and D. Tselios, "Performance Analysis in Olympic Games Using Exploratory Data Analysis Techniques".**
- [2] J. Smith and R. Jones, "120 Years of Olympic Games — How to Analyze and Visualize the History".**
- [3] P. A. Doe, "Analyzing Evolution of the Olympics by Exploratory Data Analysis Using R".**
- [4] S. Arampatzis and D. Tselios, "Performance Analysis in Olympic Games Using Exploratory Data Analysis Techniques".**
- [5] B. Liang and K. Wang, "Using Data Mining Techniques to Predict Athlete Performance in Olympic Games".**
- [6] F. Lopez and T. Nguyen, "Exploratory Analysis and Visualizations of Olympic Data Using Python Libraries".**

## Appendix:

### ❖ CSV file of Athlete BMI dataset:

"Athlete","BMI","Sport"  
"Joe Kovacs","40","4"  
"Patty Mills","24","2"  
"Ryan Crouser","35.9","4"  
"Richie Mccaw","30.6","3"  
"Goran Dragic","23.8","2"  
"Brigid Kosgei","17.3","1"  
"Seth Curry","23.8","2"  
"Heather Moyce","22.6","3"  
"Zerseney Tadese","21.1","1"  
"Fernando Portugal","28.1","3"  
"Kyrie Irving","24.9","2"  
"Eliud Kipchoge","18.6","1"  
"Valerie Adams","32.2","4"  
"David Harvey","27.8","3"  
"Tom Walsh","35.1","4"  
"Lelisa Desisa","20.1","1"  
"Ivanka Khristovia","30.4","4"  
"Santiago Gomez","25.2","3"  
"Abdi Nageeye","19.8","1"  
"Bruce Brown","24.7","2"

### ❖ Code for the application:

```
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import time

# Title of the application
st.title("Olympics Dataset Analytics") # need to change the font

# Sidebar Navigation
nav = st.sidebar.radio("Navigation", ['Home', 'Exploratory Data Analysis',
'Data Preprocessing', 'Trends', 'Prediction'])
```

```

# Home Section
if nav == 'Home':

    st.image("https://www.google.com/url?sa=i&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FMixed_teams_at_the_Olympics&psig=A0vVaw25bsFBQ42iAMuPQj0n47RT&ust=1728638512531000&source=images&cd=vfe&opi=89978449&ved=0CBQQjRxqFwoTCLD9yLW-g4kDFQAAAAAdAAAAABAE", width=800)
    st.write("The main theme of this app is to perform data analytics on Olympic datasets (namely Athlete Events and Athlete BMI datasets)...")

    # Load Athlete Events dataset button
    if st.button('Load Main/Athlete Events dataset'):
        st.write('Loaded successfully! Can perform analysis on it by following this flowchart')
        st.graphviz_chart('''
        digraph {
            Home -> ExploratoryDataAnalysis
            ExploratoryDataAnalysis -> DataPreprocessing
            DataPreprocessing -> Trends
            Trends -> Prediction
            DataPreprocessing -> Prediction
        }
        ''')

    # Load Athlete BMI dataset button
    if st.button('Load Athlete BMI dataset'):
        st.write('Loaded successfully! Can perform predictions on it')
        st.graphviz_chart('''
        digraph {
            Home -> Prediction
        }
        ''')

# Load the athlete dataset
data = pd.read_csv('C:/Users/Manidheepa/Downloads/athlete_events_dataset.csv')
p = pd.DataFrame(data)

# Exploratory Data Analysis Section
if nav == 'Exploratory Data Analysis':
    st.header('Exploratory Data Analysis')

    # Show Dataset
    if st.checkbox("Show Dataset"):
        st.write(p)

    # Show First 5 Values of the Dataset
    if st.checkbox("Show first 5 values of the Dataset"):
        st.dataframe(p.head())

```

```

# Get the total number of Rows and Columns
if st.checkbox("Get the total number of Rows and Columns"):
    st.write(p.shape)

# Show the Statistical Information Of the Columns/Attributes
if st.checkbox("Show the Statistical Information Of the
Columns/Attributes"):
    st.write(p.describe())

# Check for Null Values in columns
if st.checkbox("Null Values in columns"):
    d = pd.DataFrame(p.isnull().sum()).transpose()
    st.write(d)

# Display null values in each column
st.subheader("Columns with Null Values")
null_columns = p.columns[p.isnull().any()] # Identify columns with
null values
null_counts = p[null_columns].isnull().sum()

for col in null_columns:
    # Checkbox for each column with null values
    if st.checkbox(f"Show athletes with null values in '{col}'"):
        null_athletes = p[p[col].isnull()]['Name'] # List of
athletes with null in this column
        st.write(f"Athletes with null values in '{col}':")
        st.write(null_athletes.reset_index(drop=True)) # Display
athlete names in Streamlit app

# Data Preprocessing Section
if nav == 'Data Preprocessing':
    st.header('Data Preprocessing')

# Remove Null Values in Age Column
if st.checkbox("Remove Null Values in Age Column"):
    p['Age'] = p['Age'].fillna(p.Age.mean())
    st.dataframe(p)

# Remove Null Values in Height Column
if st.checkbox("Remove Null Values in Height Column"):
    p['Height'] = p['Height'].fillna(p.Height.mean())
    st.dataframe(p)

# Remove Null Values in Weight Column
if st.checkbox("Remove Null Values in Weight Column"):
    p['Weight'] = p['Weight'].fillna(p.Weight.mean())
    st.dataframe(p)

```

```

# Convert Medals to Numeric datatype and Remove Null Values
if st.checkbox('Covert Medals to Numeric datatype and Remove Null
Values'):
    p['Medal'] = p.Medal.replace({'Gold': 1, 'Silver': 2, 'Bronze':
3})
    p['Medal'] = p['Medal'].fillna(0)
    st.write(p)

# Check Updated Null Values
if st.checkbox("Updated Null Values"):
    d = pd.DataFrame(p.isnull().sum()).transpose()
    st.write(d)

# Remove Redundant column
if st.checkbox("Remove redundant column"):
    p = p.drop(['Games'], axis=1)
    st.write(p)

# Show Final Dataset
if st.button("Final Dataset"):
    st.write(p)

# Trends Section
if nav == 'Trends':
    st.header('Trends')

# Analyze the relationship between the Height and Weights of an
athlete
if st.checkbox("Analyze the relationship between the Height and
Weights of an athlete"):
    graph = st.selectbox("What kind of Plot do you want?", ['Scatter
Plot', 'Histogram'])

    if graph == 'Histogram':
        Figure= px.histogram(p, x=p.Height, color=p.Weight)
        st.write(fig)

    if graph == 'Scatter Plot':
        plt.scatter(p['Height'], p['Weight'])
        plt.xlabel('Height')
        plt.ylabel('Weight')
        plt.title('Height Vs Weight')
        st.set_option('deprecation.showPyplotGlobalUse', False)
        st.pyplot()

# Approximate Number of Males And Females Participated in the
Olympics
if st.checkbox('Approximate Number of Males And Females Participated
in the Olympics'):

```



```

p['Sex'].value_counts().plot.bar(p['Sex'])
st.set_option('deprecation.showPyplotGlobalUse', False)
plt.grid()
st.pyplot()

# Determine the Participation trend in the Summer and Winter Seasons
if st.checkbox("Determine the Participation trend in the Summer and
Winter Seasons"):
    Figure= px.histogram(p, x=p.Season, color=p.Sex, barmode="group")
    st.write(fig)

# Women Participation over the years
if st.checkbox('Women Participation over the years'):
    y = p[p['Sex'] == 'F']['Sex']
    Figure= px.histogram(y, x=p.Year)
    st.write(fig)

# Number of Medals Won by Male and Female athletes
if st.checkbox('Number of Medals Won by M and F'):
    p['Medal'] = p.Medal.replace({'Gold': 1, 'Silver': 2, 'Bronze':
3}))
    p['Medal'] = p['Medal'].fillna(0)
    Figure= px.histogram(p, x=p.Sex, color=p.Medal)
    st.write(fig)

# Athletes with Most Medals
if st.checkbox("Athletes with Most Medals"):
    p['Medal'] = p.Medal.replace({'Gold': 1, 'Silver': 2, 'Bronze':
3}))
    p['Medal'] = p['Medal'].fillna(0)
    df = p[['Medal']]
    df = pd.get_dummies(df.Medal)
    df = df.drop([0], axis=1)
    df['Name'] = p['Name']
    df['Total'] = df[1] + df[2] + df[3]
f=df.groupby(df['Name'])['Total'].sum().sort_values(ascending=False).head
(10)
    x = pd.DataFrame(f)
    st.write(x)
    Figure= plt.figure()
    ax = fig.add_axes([0, 0, 1, 1])
    ax.axis('equal')
    Team = list(f.index.values)
    Count_of_Medal = f
    ax.pie(Count_of_Medal, labels=Team, autopct='%1.2f%%')
    plt.show()
    st.pyplot()

# Countries with Most Medals

```

```

if st.checkbox("Countries with Most Medals"):
    p['Medal'] = p.Medal.replace({'Gold': 1, 'Silver': 2, 'Bronze':
3}))
    p['Medal'] = p['Medal'].fillna(0)
    df = p[['Medal']]
    df = pd.get_dummies(df.Medal)
    df = df.drop([0], axis=1)
    df['Team'] = p['Team']
    df['Total'] = df[1] + df[2] + df[3]
    f =
df.groupby(df['Team'])['Total'].sum().sort_values(ascending=False)
    k = f.head(10)
    k = list(k.index.values)
    k = pd.DataFrame(k)
    st.write(k)

    # Check for Zero Medal List
    st.subheader("Find whether your country is in the Zero-Medal
list?")
    x = st.text_input('Enter')
    if st.checkbox('Show'):
        if x not in f.index.values:
            st.write('Country not listed in the dataset, so be
optimistic about your country winning a medal')
        else:
            if f[x] != 0:
                st.write('Your country has won at least 1 Medal, so
chill')
            else:
                st.write('Sorry to break it to you, your country is
in the Zero-Medal list')

    # Countries winning the most Gold medals in a specific year
    if st.checkbox("Countries winning the most Gold medals in a specific
year"):
        number = st.number_input('Insert the Leap Year', 1896.00, 2016.00,
step=4.00)
        st.write('The current Year is ', int(number))
        max_year = int(number)

        if max_year not in p.Year:
            st.write("Enter a valid year")
        else:
            if st.button('Show'):
                if max_year not in p.Year:
                    st.write("Enter a valid year")
                else:
                    team_list = p[(p.Year == max_year) & (p.Medal ==
'Gold')].Team

```

```

        if len(team_list) != 0:
            sns.barplot(x=team_list.value_counts().head(),
y=team_list.value_counts().head().index)
            st.set_option('deprecation.showPyplotGlobalUse',
False)

            st.pyplot()
        else:
            st.write('Enter valid year')

# Observations
if st.checkbox('Observations'):
    a = st.text_area("Observations")
    st.write(a)

# Prediction Section
if nav == 'Prediction':
    st.header('Prediction')

    # Prediction for Weight based on Height
    st.subheader('Predict the Weight of an athlete with his/her Height')
    model = LinearRegression()
    p['Height'] = p['Height'].fillna(p.Height.mean())
    p['Weight'] = p['Weight'].fillna(p.Weight.mean())
    x = p['Height']
    x = np.array(x).reshape(-1, 1)
    y = p['Weight']
    y = np.array(y).reshape(-1, 1)
    x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2)
    model.fit(x_train, y_train)
    t = st.number_input('Enter the Height')
    t = np.array(t).reshape(-1, 1)
    d = model.predict(t)
    if st.button('Predict Weight'):
        st.write(d)

    # Prediction for suitable Sport based on BMI values
    st.subheader('Predict the suitable Sport with the BMI values')
    data
=pd.read_csv('C:/Users/Manidheepa/Downloads/athlete_bmi_dataset.csv')
    k = pd.DataFrame(data)

    # Show Athlete BMI Dataset
    if st.checkbox("Show Athlete BMI Dataset"):
        st.dataframe(k)
        st.write("Note:")
        q = {"Value": ['1', '2', '3', '4'],
            "Corresponding Sport": ['Marathon', 'Basketball', 'Rugby',
'Shot Put']}

```

```

m = pd.DataFrame(q)
st.write(m)

model1 = LinearRegression()
j = k['BMI']
j = np.array(j).reshape(-1, 1)
l = k['Sport']
l = np.array(l).reshape(-1, 1)
j_train, j_test, l_train, l_test = train_test_split(j, l,
test_size=0.3)
model1.fit(j_train, l_train)
t = st.number_input('Enter BMI')
t = np.array(t).reshape(-1, 1)
d = model1.predict(t)
d = d * 10
d = np.round(d)

if st.button('Results'):
    my_bar = st.progress(0)
    for percent_complete in range(100):
        time.sleep(0.001)
        my_bar.progress(percent_complete + 1)

    if d in range(0, 12):
        st.write('Definitely Marathon')
    elif d in range(12, 19):
        st.write('Marathon, But also suitable for Basketball')
    elif d in range(19, 23):
        st.write('Definitely Basketball')
    elif d in range(23, 27):
        st.write('Basketball, But also suitable for Rugby')
    elif d in range(27, 33):
        st.write('Definitely Rugby')
    elif d in range(33, 38):
        st.write('Rugby, can also try shot put')
    else:
        st.write('Opt for Shot Put')

```

