**Documentation: Olympic Performance Analysis Project**

**Introduction**

This project analyzes the historical performance of countries in the Olympics, focusing on medal trends over the years. By visualizing these trends, we can identify which nations dominate the Olympics and how their performances have evolved.

The dataset used is *"120 Years of Olympic History"* from Kaggle, which contains athlete details, events, and medal information from 1896 to 2016.

**Steps for Beginners**

**Step 1: Setting Up**

1. **Install Python Libraries**:
   * Install Python and ensure packages pandas, matplotlib, and seaborn are installed. Use:

bash

Copy code

pip install pandas matplotlib seaborn

1. **Download the Dataset**:
   * Go to Kaggle.
   * Download the dataset and save it as athlete\_events.csv in your working directory.

**Step 2: Algorithm**

1. **Import Required Libraries**:
   * Load the necessary libraries: pandas for data manipulation, matplotlib and seaborn for visualization.
2. **Load the Dataset**:
   * Read the dataset into a pandas DataFrame.
3. **Data Cleaning**:
   * Remove rows with missing medal information (NaN values in the "Medal" column).
   * Ensure the "Year" column is in numeric format for easier grouping.
4. **Group and Aggregate Data**:
   * Group data by Year and NOC (country code) to calculate the total medals won by each country in each Olympic year.
5. **Visualize Trends**:
   * Plot medal trends for selected countries over time using a line plot.
6. **Generate Insights**:
   * Identify top-performing countries by total medals.
   * Save visuals for presentations or reports.

**Complete Algorithm**

1. **Setup and Import**:
   * Import the libraries:

python

Copy code

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

1. **Data Loading**:
   * Load the dataset:

python

Copy code

df = pd.read\_csv('athlete\_events.csv')

1. **Data Cleaning**:
   * Remove missing medal entries and format the "Year" column:

python

Copy code

df\_medals = df.dropna(subset=['Medal'])

df\_medals['Year'] = df\_medals['Year'].astype(int)

1. **Grouping Data**:
   * Group medals by Year and NOC:

python

Copy code

medal\_trends = df\_medals.groupby(['Year', 'NOC'])['Medal'].count().reset\_index()

1. **Visualize Data**:
   * Plot medal trends for selected countries:

python

Copy code

plt.figure(figsize=(12, 8))

top\_countries = ['USA', 'RUS', 'CHN', 'GBR']

for country in top\_countries:

country\_data = medal\_trends[medal\_trends['NOC'] == country]

plt.plot(country\_data['Year'], country\_data['Medal'], label=country)

plt.title('Olympic Medal Trends', fontsize=16)

plt.xlabel('Year')

plt.ylabel('Medals')

plt.legend(title='Country')

plt.grid(True)

plt.savefig('olympic\_trends.png')

plt.show()

1. **Insights**:
   * Find the top 5 countries:

python

Copy code

top\_5\_countries = df\_medals['NOC'].value\_counts().head(5)

print(top\_5\_countries)

**For Beginners: Key Notes**

* **What is a DataFrame?**: It’s like an Excel sheet where rows are observations and columns are variables.
* **What is Grouping?**: Grouping helps aggregate data (e.g., summing medals for a country per year).
* **Why Clean Data?**: Incomplete data (missing values) can lead to incorrect analysis.

**Visualization Explanation**

* **Line Plot**: Shows the number of medals won by each country over time.
* **Countries Compared**: USA, RUS, CHN, GBR (feel free to customize).
* **Customization**: Change colors, add labels, or focus on specific years.

**Key Takeaways**

* This project introduces basic data handling and visualization.
* Perfect for understanding how to manipulate and analyze real-world datasets.
* Opportunity to expand by using machine learning for future medal predictions!