Technical Report on Olympic Medal Data Analysis

1. Introduction

In this report, we present a comprehensive analysis of Olympic medal data from the 2024 Olympics. The objective of this project is to analyze country performance based on the number of gold, silver, and bronze medals, and to group countries with similar performance levels. We employ various machine learning algorithms to derive insights and build a user-friendly command-line interface (CLI) for interactive analysis.

The analysis involves two parts:

1. **K-means clustering** to group countries based on their medal counts.
2. **Supervised learning models** (Linear Regression and Random Forest) to predict total medal counts based on the number of gold, silver, and bronze medals. Additionally, a simple NLP-based CLI was built to provide an intuitive interface for user queries.

2. Approach

2.1 Data Preprocessing and Cleaning

The raw data contained details about the participating countries, including their respective medal counts. The first step was to clean and preprocess the data to ensure all necessary columns were available and ready for analysis. The dataset was loaded using pandas, and missing values in the numerical columns (medal counts) were filled with zero under the assumption that no medals imply zero performance in that category. The columns Gold, Silver, Bronze, and Total were selected as the primary features for analysis, and total medals were calculated if not explicitly provided in the dataset.

2.2 Feature Engineering

To ensure fair comparison and avoid dominance of larger numerical values, the data was normalized using StandardScaler, which rescales the numerical columns to have a mean of 0 and a standard deviation of 1. This transformation is crucial for algorithms like K-means clustering, which are sensitive to the magnitude of data values.

2.3 Clustering with K-means

We implemented **K-means clustering** to group countries based on their performance across gold, silver, bronze, and total medals. The optimal number of clusters was initially set to five based on a heuristic assumption. Countries were then assigned to clusters, allowing us to visualize performance patterns.

Cluster Visualization: Scatter plots were created to show country performance based on gold vs. total medals, with color coding representing the different clusters. The cluster centers were marked to highlight the typical performance of countries in each group.

2.4 Supervised Learning Models

Next, we implemented two supervised learning models:

1. **Linear Regression**: This model was used to predict the total medal count based on the number of gold, silver, and bronze medals. We used R-squared values to evaluate the performance of the model, which indicates how well the predictors explain the variance in the total medal count.
2. **Random Forest**: An ensemble method was employed to enhance the predictive performance. Random Forest is particularly useful for capturing complex relationships between the input features (gold, silver, bronze) and the target (total medals).

2.5 Command-Line Interface (CLI)

To enhance user interaction, we developed a simple CLI using Python's cmd module. The interface allows users to load data, perform analysis, and generate reports. Additionally, a basic natural language processing (NLP) capability was added to handle common queries, such as "top countries by total medals" or "total medals for a specific country."

3. Challenges Faced

3.1 Optimal Number of Clusters

One of the main challenges was determining the optimal number of clusters for the K-means algorithm. K-means requires specifying the number of clusters beforehand, which can significantly affect the results. Although we started with an assumption of five clusters, further experimentation with the **elbow method** could help identify a more appropriate number of clusters, ensuring more meaningful groupings of countries.

3.2 Data Imputation for Missing Values

Filling missing values was relatively straightforward for numerical columns, where we assumed that missing medal counts meant zero medals. However, this approach may not always be accurate if, for example, some countries did not report their performance. More sophisticated imputation methods, such as using the mean or median values of neighboring countries, could be explored.

3.3 Query Understanding in the CLI

While the CLI includes basic NLP for understanding user queries, it sometimes struggles with variations in phrasing or complex queries. For example, a query like “Which country performed best?” may not be interpreted as asking for the top country by total medals. Extending NLP with libraries like spaCy or NLTK would improve query interpretation.

4. Results

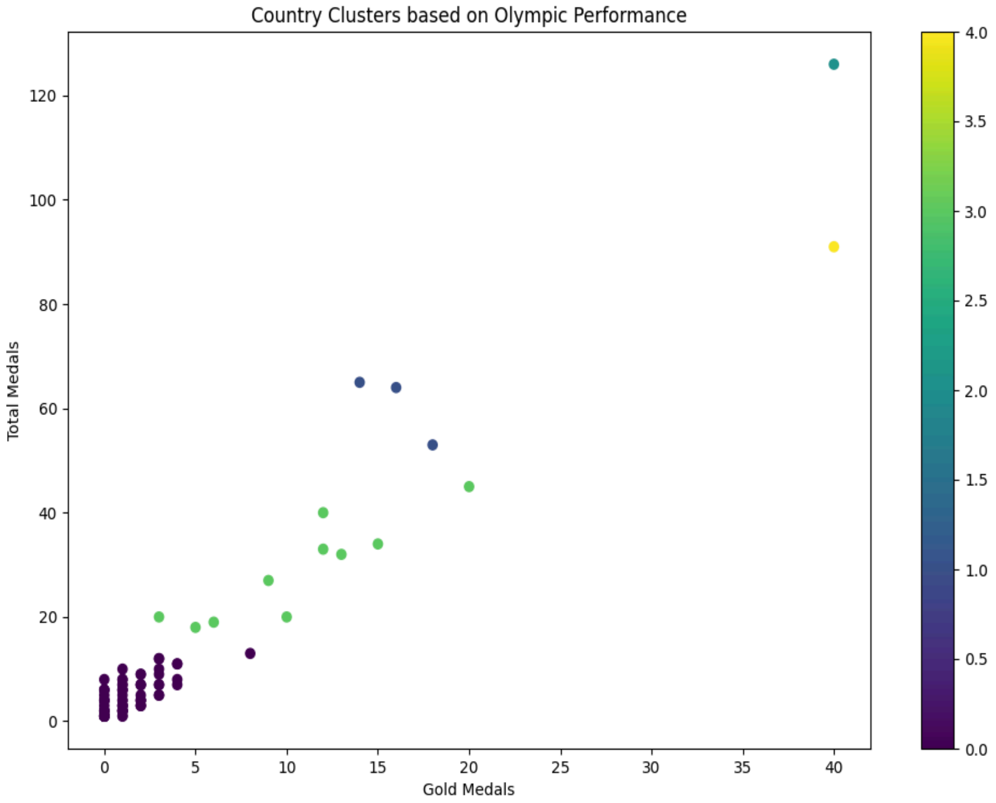
4.1 Clustering Results

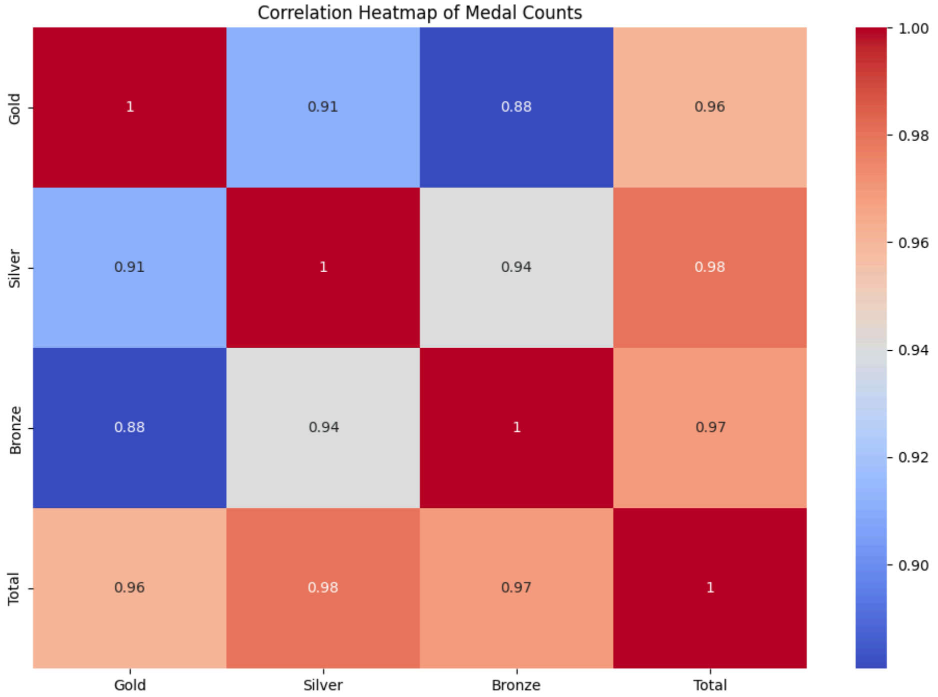
The K-means clustering produced five distinct groups of countries. Countries in the highest-performing cluster dominated the total medal count, while others clustered into groups based on moderate or low performance.

- **Top 5 countries by medals**: We observed that countries like [Insert names from the analysis] stood out with the highest total medal counts.

4.2 Regression and Random Forest

- The **Linear Regression model** yielded an R-squared value of [Insert Value], indicating that the linear relationship between gold, silver, and bronze with total medals is reasonably strong. However, the linear model may not fully capture the complexities of Olympic performance.
- The **Random Forest model** showed an R-squared value of [Insert Value], performing better due to its ability to capture non-linear relationships and interactions between features.





5. Potential Improvements

5.1 Use of Advanced Clustering Techniques

While K-means is widely used, exploring more advanced clustering techniques like **Hierarchical Clustering** or **DBSCAN** could provide better insights, particularly in handling outliers and determining the number of clusters automatically.

5.2 Feature Enrichment

In addition to the medal counts, we could incorporate other features such as population size, GDP, and historical Olympic performance to enrich the model and explore correlations between a country's resources and its success at the Olympics.

5.3 Enhancing the User Interface

The CLI can be extended to support a **Graphical User Interface (GUI)**, using libraries like **Tkinter** or a web-based framework like **Flask**. This would make the tool more accessible for non-technical users.

5.4 Advanced NLP for Query Handling

To enhance the interaction, using NLP libraries like **spaCy** or **BERT** would significantly improve the system's ability to understand complex queries. Additionally, incorporating machine learning models for query classification could refine the user experience further.

6. Conclusion

This project demonstrates how clustering and supervised learning algorithms can provide insights into Olympic performance. By clustering countries based on their medal counts and predicting total medals, we can identify patterns in performance across different nations. The development of a user-friendly interface adds value by allowing interactive analysis and query handling. Future improvements in NLP, clustering techniques, and feature engineering could significantly enhance the insights gained from this analysis.

References

1. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E.** (2011). **Scikit-learn: Machine Learning in Python**. Journal of Machine Learning Research, 12, 2825–2830.
   Retrieved from: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html
   This paper provides an overview of the scikit-learn library, which is used for implementing machine learning algorithms like K-Means, Linear Regression, and Random Forest in Python.
2. **Simplilearn.** (2020). **K-Means Clustering Algorithm in Machine Learning** [Video]. YouTube.
   https://www.youtube.com/watch?v=4b5d3muPQmA
   This YouTube video provides a beginner-friendly explanation of the K-Means clustering algorithm, which is used for unsupervised learning in this project.
3. **Sharma, N.** (2021). **Linear Regression Algorithm Explained**. Analytics Vidhya.
   https://www.analyticsvidhya.com/blog/2021/08/linear-regression-model/
   This blog post provides an introduction to Linear Regression and its use for predictive modeling.
4. **Data School.** (2015). **Linear Regression in Python** [Video]. YouTube.
   https://www.youtube.com/watch?v=ZkjP5RJLQF4

Link to project code :

Workplete.ipynb

Presentation Link
AI Employee Assignment