

Forecasting Greatness: Olympic Medal Predictions based on PRE Model

Summary

The Olympic Games, inspired by the ancient Greek Games, is the world's premier international multi-sport event. Held every four years, it features summer and winter editions, bringing together athletes from around the globe to compete in a wide range of sports. For every sports lover, the Stimulation of Olympic Medals is a very interesting topic, while it's crucial for countries to adjust their strategy. **In this paper, we will analyze the data of the Summer Olympic Games from 1896 to 2024, and use the data to simulate the results of the 2028 and 2032 Olympics using different methods from 'PRE' model mostly based on machine learning.**

First, we've examined, cleaned and transformed the raw data, setting the ice sports aside, combining various teams of one country, and mapping the countries which no longer exist to the current existing country. Then, we turn to machine learning, select the independent variables to use by **Correlation Coefficient Matrix** and **Principal Component Analysis**, split the data into training and testing parts, and use **eXtreme Gradient Boosting** methods to train features and target matrix. **Kolmogorov-Smirnov Test** is used to ensure the effectiveness of the model, while history data are reprocessed using feature importance and data in recent years are given higher weights to improve the accuracy of the model. In the end, we use the model to predict the results of the 2028 and 2032 Olympics, and the results are shown in multiple forms.

After that, we encode country labels, create feature dataset of each country, and filter out data for countries that won medals in 1896 to construct training and testing datasets of the feature of 'First Win Country'. **Random Forest** is used to train the model, and the results of the top ten countries most likely to win their first medal in 2028 are shown in the form of a bar chart.

By analyzing the data given, we find that some specific sports and events play a significant role in the medal tally of some specific countries, e.g: long distance race for Kenya. We calculate the **proportion** of medals from these events, further estimate their importance, and finally come to the result of the extent to which choosing these events impact countries performance in the medal list.

Finally, we combined methods used above and created a comprehensive prediction model called '**PRE**' model, naming after the primary methods we use. The model is specifically-tuned to calculate **Legendary Index** so as to detect **Great Couch Effect** using data of US gymnastics team coached by Bela Karolyi and Marta Karolyi. Lang Ping is successfully detected as a legendary coach, and we've found many more great teams and athletes, even controversial results(2024 Male Fencing, Italy) and decline of the great ranks(2024 Tennis, China).

Greatness is not born, but made. We hope that our model may help countries' Olympic committees to accommodate strategies and achieve better results in the future Olympic Games.

Keywords: Olympic Games; Data Analysis; Stimulation; Machine Learning

Contents

1	Introduction	2
2	Assumptions and Notations	3
3	Data Preprocessing	4
3.1	Data Cleaning	4
3.2	Country Mapping	5
4	Task I : Predicting the Medal of the 2028 LA Olympics	6
4.1	Model Establishment	6
4.2	Result Analysis	7
4.3	Prediction of 2028 Los Angeles Olympics	9
5	Task II : Key Events	9
5.1	Key Events	9
6	Task III : Predicting First Medal Countries	10
6.1	Problem Analysis	10
6.2	Model Establishment	10
6.3	Result Analysis	11
7	Task IV : Great Coach Effect	12
7.1	Problem Analysis	12
7.2	Model Establishment : Legendary Index	13
7.3	Lang Ping: The Great Coach	14
7.4	Three Countries to invest in ‘great’ coach	15
8	Other Insights	16
8.1	Try to predict the medal of the 2032 Olympics	16
8.2	Greatness and the strategy of the country Olympic committees	16
9	Strength and Weakness	17
9.1	Strength	17
9.2	Weakness	17
9.3	Further Discussions	18
9.3.1	Possible Optimization	18
9.3.2	Challenges in practical application	18
	Appendices	19
	Appendix A Core Code for Predicting 2028 Olympics Medal	19
	Appendix B Core Code for Great Athletes and Key Events	22
	Appendix C Core Code for Predicting First Medal Countries	22
	Appendix D Core Code for Great Coach Effect	23

1 Introduction

Background and Literature Review

The Olympic Games, often simply referred to as the Olympics, are the world's foremost international multi-sport events. With a history that dates back over 2,000 years to ancient Greece, the modern Olympics were revived in 1896 by Pierre de Coubertin. The Olympics are held every four years, with the Summer Olympics and the Winter Olympics alternating every two years. The Summer Olympics feature a vast array of sports, from athletics and swimming, which are considered the cornerstones of the Games, to more specialized sports like fencing, badminton, and gymnastics. Athletes from around the globe gather to compete at the highest level, showcasing their extraordinary skills, determination, and physical prowess.

The prediction of Olympic medal counts has long been a topic of interest among sports enthusiasts, statisticians, and researchers. Understanding how to forecast the number of medals a country or athlete might win is not only a matter of curiosity but also has implications for sports management, marketing, and national pride.

Traditional approaches are typically made closer to the start of an upcoming Olympic Games when information about the current athletes scheduled to compete becomes available. This approach allows for a more accurate assessment of a country's or athlete's medal prospects. For example, the virtual medal table forecast by Nielsen [2] provides a more real-time and data-driven prediction. By incorporating current athlete performance, injuries, and recent competition results, these modern models can better capture the dynamic nature of sports.

However, these data may be concealed and intentionally modified by countries' Olympic committees, which may produce misleading predictions. As a result, our model will be based on the historical data of the Olympic Games, which is more reliable and less likely to be manipulated. Research in this area has also explored the use of advanced statistical techniques. Machine learning algorithms, for instance, have been employed to analyze large datasets encompassing a wide range of variables related to athletes, sports, and countries. These algorithms can identify complex patterns and relationships that might not be apparent through traditional statistical methods.

In conclusion, the field of predicting Olympic medal counts has evolved significantly over the years. While historical contemporary methods still provide some basis for understanding trends, the focus has shifted towards historical data. The use of advanced statistical techniques and an increased awareness of external factors have improved the accuracy of these predictions. However, there is still room for further research, particularly in the use of machine learning methods and elements that can't be captured directly by historical data, such as the Great Coach Effect. As the Olympics continue to evolve, so too will the methods and models used to predict the medal counts, ensuring that this remains a vibrant and relevant area of study.

Restatement of the Problem

Considering the background, in this paper we are required to solve the following problems:

- **Task 1:** Develop a model for medal counts for each country, both **Gold** and **Total**, and use the model to predict various countries' performance in the 2028 Olympics. The model also

includes estimates of the precision and measures of how well the model performs.

- **Task 2:** Develop a model for prediction of when a country will win its **first** medal in the Olympics, and the probability of winning it in the coming next LA Olympics. Also, we will evaluate this model.
- **Task 3:** Develop a model for estimation of the **"Great Coach"** effect, and further using this model to identify three countries suitable for imitating this strategy.
- **Task 4:** Calculating the **"Great Athlete"** effect, referring to the phenomenon that some great athletes won a large number of the (Gold) medals of a certain sport or event, and sometimes his/her country's medal tally greatly depend on him/her during his/her athlete career. We will also explain how this can inform country Olympic committees.

2 Assumptions and Notations

Assumptions and Justifications

To simplify problem, make it convenient to construct simulation model and ensure it's effectiveness, following basic assumptions are made, each of which is properly justified.

- **Data Preprocess and Country Mapping.**

Data is cleaned and preprocessed to make it more suitable. We set the ice sports aside, combine various teams of one country, and map the countries which no longer exist to the current existing country. It's a rather difficult task since countries have changed a lot during the past century, but it's necessary to stick to the current world map.

- **Relationship between Medal counts and Historical data.**

Most data provided is examined to be accurate and reliable, although some may not be the same as the data provided from other sources. Few of the data is examined and changed since it's too faraway from reality. Medal counts are also assumed to be related to multiple variables: years, gender, athletes and number of sports, disciplines, events, etc.

- **Countries that will possibly take part in the 2028 Olympics.**

Countries that have taken part in 2024 Olympics will also take part in 2028 Olympics. There's no reason for them not to attend the Olympics, and it's impossible to estimate medal of new countries if they've never taken part in the Olympics using historical data. Russia is also assumed to participate in the 2028 Olympics, whether the athletes will be able to compete under the Russian flag or not, their country is called Russia.

- **Possible Sports in 2028.**

There's said to be five new sports in the 2028 Olympics, but we set them aside because we don't have enough data to predict the medal counts of these events. Similarly, we set aside the sports that will be removed from the Olympics in 2028, and assume that most sports are like that in 2024 or their most common appearance in history (Such as gymnastics).

Other details will be stated in the following sections when we discuss the specific models.

Notations

Symbols	Definitions	Symbols	Definitions
h	aaaaaaaaaaaaaaaaaaaaa	a	aaaaaaaaaaaaaaaaaaaaa
k			
c_p			
ρ			

Define the Main parameters. Other Specific notations will be listed and explained later.

3 Data Preprocessing

3.1 Data Cleaning

- Garbled Code



Figure 1: bad data



Figure 2: bad text

We carefully examine the data and find that there are some garbled codes in the data, which may be caused by the encoding of the data. They are replaced by the correct text.

- Null, Duplicated Values and Renaming

Listing 1: Data Cleaning

```
data.fillna(0, inplace=True)
data.drop_duplicates(inplace=True)
athletes.rename(columns={'Team': 'Country'}, inplace=True)
```

Team are renamed to Country in the athletes data so that there won't be Germany-1.

- Data Splitting

Listing 2: Data Splitting

```
data.fillna(0, inplace=True)
data.drop_duplicates(inplace=True)
athletes.rename(columns={'Team': 'Country'}, inplace=True)
```

- Remove unnecessary data

Listing 3: Remove Ice Sports

```
ice_sports = ['Figure Skating', 'Ice Hockey']
programs = programs[~programs['Sport'].isin(ice_sports)]
athletes = athletes[~athletes['Sport'].isin(ice_sports)]
```

Listing 4: Remove Program from 1906

```
# Remove program from the year 1906
program = program[program['Year'] != 1906]
```

3.2 Country Mapping

Listing 5: Country Mapping

```
country_mapping = {# map due to the change of country in history
    'Soviet Union': 'Russia', 'Unified Team': 'Russia',
    'West Germany': 'Germany', 'East Germany': 'Germany',
    'Yugoslavia': 'Serbia',
    'Bohemia': 'Czech Republic', 'Czechoslovakia': 'Czech Republic',
    'Virgin Islands': 'United States',
}
# NOC mapping is similar. Only part of the country_mapping is shown here. There're
  ↳ also map between NOC and country.
athletes['NOC'] = athletes['NOC'].replace(noc_mapping)
medals['NOC'] = medals['NOC'].replace(country_mapping)
```

Primary processed data: The ‘Top 10s’

Top 10 Countries by Medal Count

NOC	Gold	Silver	Bronze	Total
United States	1105	880	780	2765
Russia	588	487	478	1553
Germany	457	475	503	1435
Great Britain	299	339	343	981
France	240	278	299	817
China	303	226	198	727
Italy	230	205	228	663
Australia	182	192	226	600
Japan	189	162	193	544
Hungary	187	162	182	531

Figure 3: top 10 countries

Top 10 Athletes by Medal Count

Name	Gold	Silver	Bronze	Total
Michael li	23	3	2	28
Larysa (diriy-)	9	5	4	18
Nikolay Andrianov	7	5	3	15
Charles Jr.	9	4	2	15
John Jr.	9	1	3	13
Borys Shakhlin	7	4	2	13
Edoardo Mangiarotti	6	5	2	13
Takashi Ono	5	4	4	13
Sawao Kato	8	3	1	12
Dara -minas)	4	4	4	12

Figure 4: top 10 athletes

Model Overview

In order to solve those problems, we will proceed as follows:

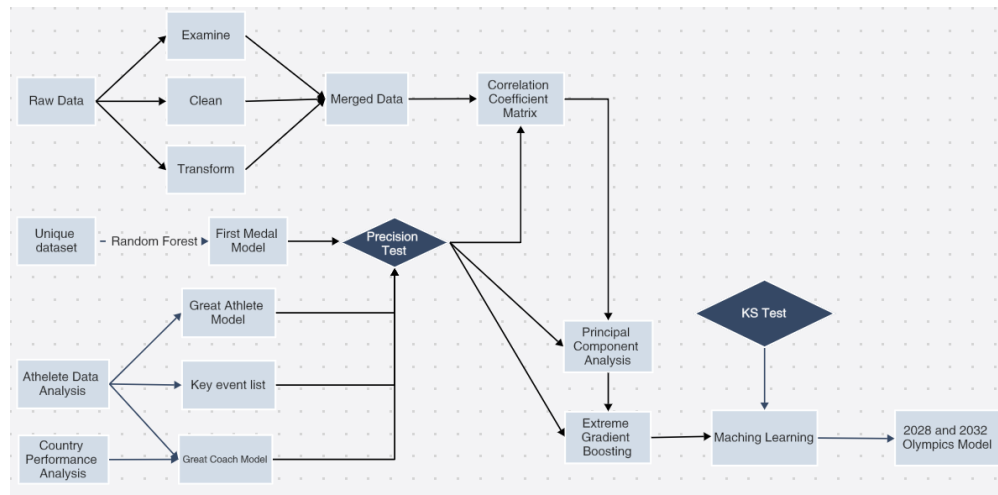


Figure 5: Flow chart of our work

- **Presenting our model.** In order to investigate the problem deeper, we divide our task into four models: **Predicting the Medal of the 2028 LA Olympics**, **Predicting First Medal Countries**, **Great Coach Effect**, **Key Events** and **Great Athletes**.
- **Data Processing.**First, We examine, clean and transform the data given and merge them. Then the data is put into the Correlation Coefficient Matrix for further machine learning.
- **First Medal Model.**Next, We use **Random Forest** to train the model, giving predicition of first medal countries.
- **Performance Analysis.**Then, we calculate the performance of Athletes and countries (on certain sports/events), forming the **Key Event Model** and the **Great Coach Model**.
- **Machine Learning.** Finally, we use **KS Test** to ensure the precision of the final model.

4 Task I : Predicting the Medal of the 2028 LA Olympics

4.1 Model Establishment

Since we want to predict future medal list on the basis of past data, we merge the given data and do some preprocessing. We combine the data with the same **Year** and **NOC** together.

We use **Correlation Coefficient Matrix** and **Principal Component Analysis**, split the data into training and testing parts, and use **Extreme Gradient Boosting methods** to train features and target matrix.

We also use **KS test** to ensure the effectiveness of the model.After the model is established, we use it to predict the 2028 and 2032 Olympics.

What's important is that we add the factor of whether being a host city/country or not in to the training model so that the final result can take this factor into consideration. For example, in the coming LA Olympics, the USA is likely to gain more medals due to the **"hosting effect"**.

4.2 Result Analysis

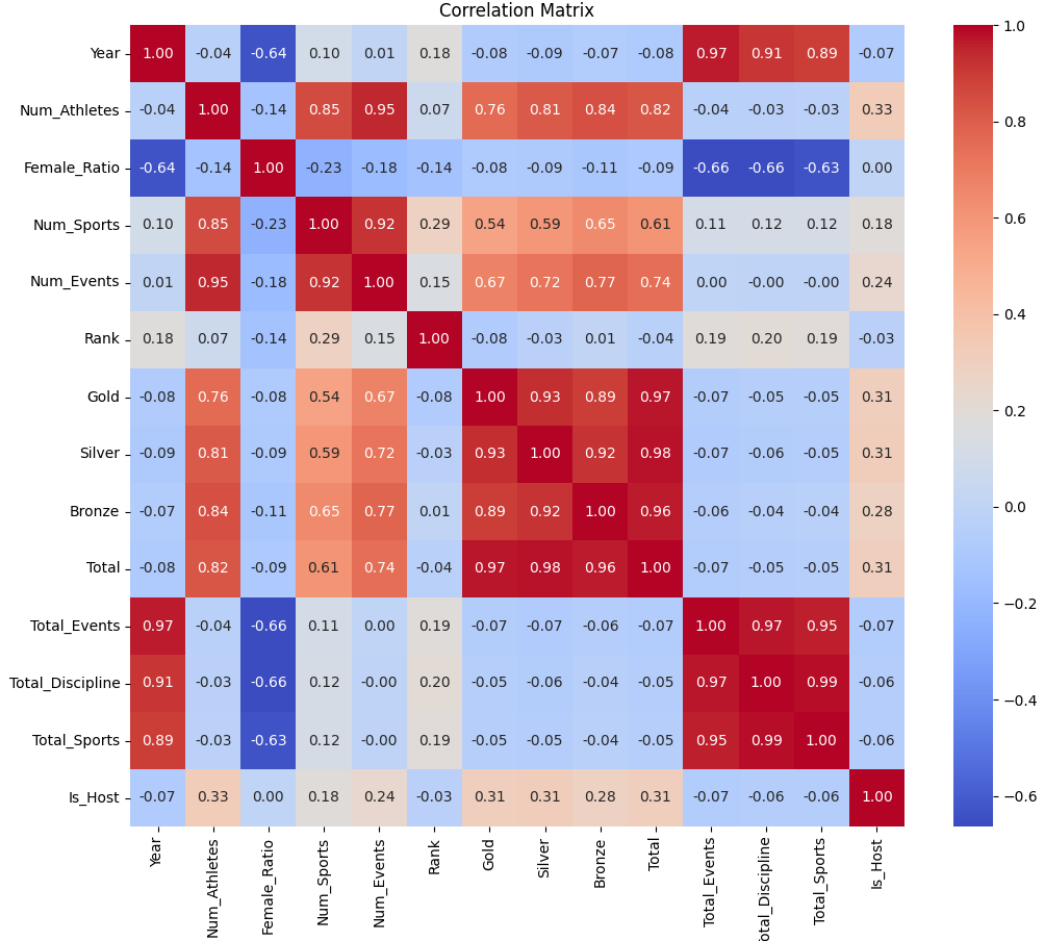


Figure 6: Correlation Matrix

According to the results graphs following, we can see that our model is of good accuracy, and have a good performance on test set.

From Figure2, we can see that the prediction of XGBoost outperformed the prediction of Random Forest Predictions, which is one of the reason for our choice of model.

As to the specific principle of XGBoost:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

The loss function and the objective function are as follows:

$$L(\hat{y}_i^{(t-1)}, y_i) \quad \text{loss function } y_i \text{ prediction from the first to the } t\text{-1th tree } \hat{y}_i^{(t-1)}$$

The complete objective function (loss function plus regularization term) can be represented as:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n L(\hat{y}_i^{(t-1)}, y_i) + \Omega(f_t)$$

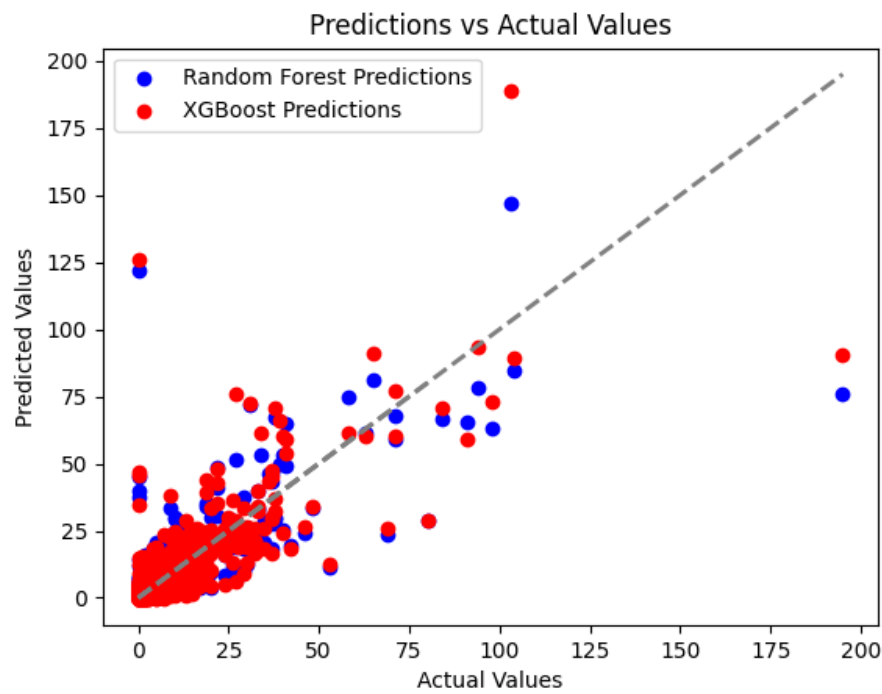


Figure 7: Precision of the estimation

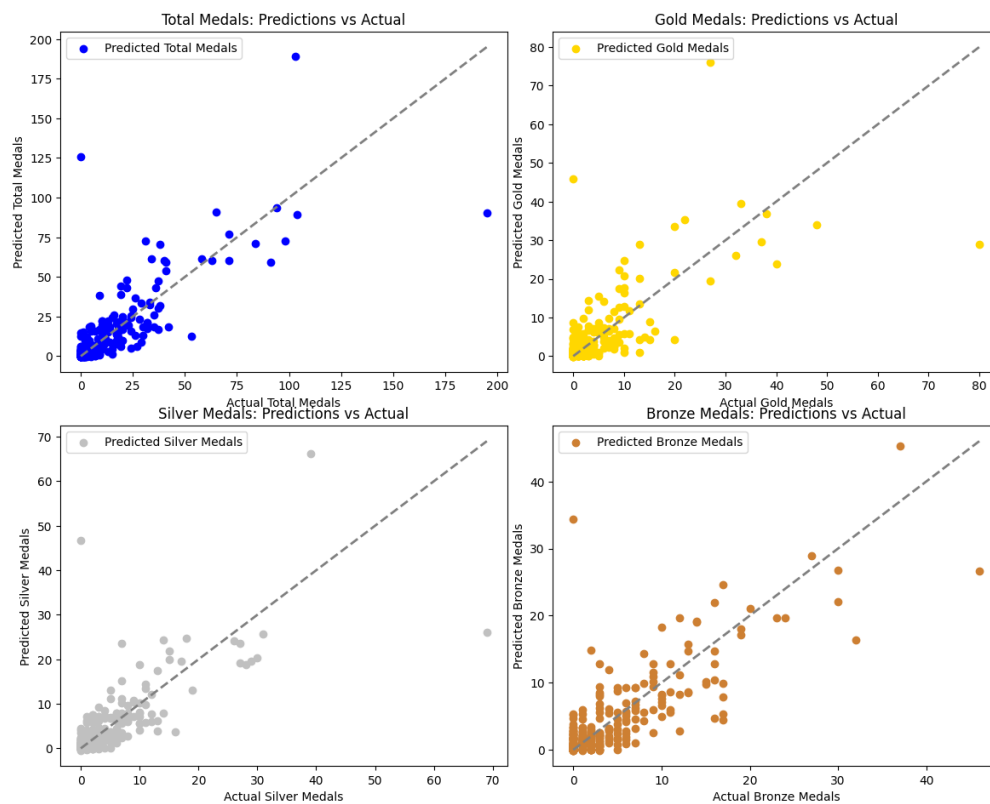


Figure 8: Precision of the estimation(Gold, Silver, Bronze and Total)

```

Random Forest Predictions: [[6.00000000e-01 2.50000000e-01 1.90000000e-01 1.60000000e-01]
[1.02000000e+00 1.70000000e-01 3.50000000e-01 5.00000000e-01]
[6.19000000e+00 1.28000000e+00 2.64000000e+00 2.27000000e+00]
...
[1.05000000e+00 7.00000000e-02 3.90000000e-01 5.90000000e-01]
[4.85714286e-03 4.85714286e-03 0.00000000e+00 0.00000000e+00]
[2.99000000e+00 7.30000000e-01 1.27000000e+00 9.90000000e-01]]
XGBoost Predictions: [[ 5.1203884e-02  3.2490236e-03  1.3189411e-04  5.9093786e-03]
[-2.0074600e-02  1.5166138e-02  2.2303849e-03  3.3099253e-02]
[ 6.2406716e+00  1.2808298e+00  3.1602407e+00  2.2505367e+00]
...
[ 5.8379167e-01 -5.4412475e-03  2.2964623e-02  3.4039971e-01]
[ 2.9888877e-04  1.2590903e-03 -3.9039849e-05 -5.8104652e-05]
[ 3.4480846e+00  9.0951747e-01  1.3775554e+00  1.0207008e+00]]
K-S Test Statistic: [0.3      0.4953125  0.390625  0.4265625], P-Value: [8.25480975e-26 1.16773287e-71 5.95698276e-44 1.29633612e-52]
Random Forest MSE: 27.695270017868456, R^2: 0.6306700795983087
XGBoost MSE: 30.791641235351562, R^2: 0.5817188024520874
XGBoost Total Medals - MSE: 86.20039085781896, R^2: 0.6263195818297187
XGBoost Gold Medals - MSE: 18.183467512401762, R^2: 0.45440816844255594
XGBoost Silver Medals - MSE: 11.543446173187839, R^2: 0.5503359404587108
XGBoost Bronze Medals - MSE: 7.239262598410515, R^2: 0.6958110484049175

```

Figure 9: Specific result during training and estimation

where $\Omega(f_t)$ is the model complexity of the t -th tree, which can be expressed as:

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

where T is the number of nodes in the tree, w_j is the weight of each node, and γ and λ are regularization parameters.

4.3 Prediction of 2028 Los Angeles Olympics

Table 1: Predicted Medal Table of 2028 Los Angeles Olympics

Rank	NOC	Gold	Silver	Bronze	Total
1	United States	49	46	44	139
2	China	38	30	29	97
3	Russia	28	25	22	75
4	United Kingdom	27	23	20	70
5	Japan	22	19	17	58
6	Australia	17	15	13	45
7	Italy	15	13	12	40
8	Netherlands	13	12	11	36
9	France	14	12	9	35
10	Germany	15	12	8	35

5 Task II : Key Events

5.1 Key Events

Calculation Methods

To identify key events, we will use the proportion of medals from these events among the whole medal tally as a main factor.

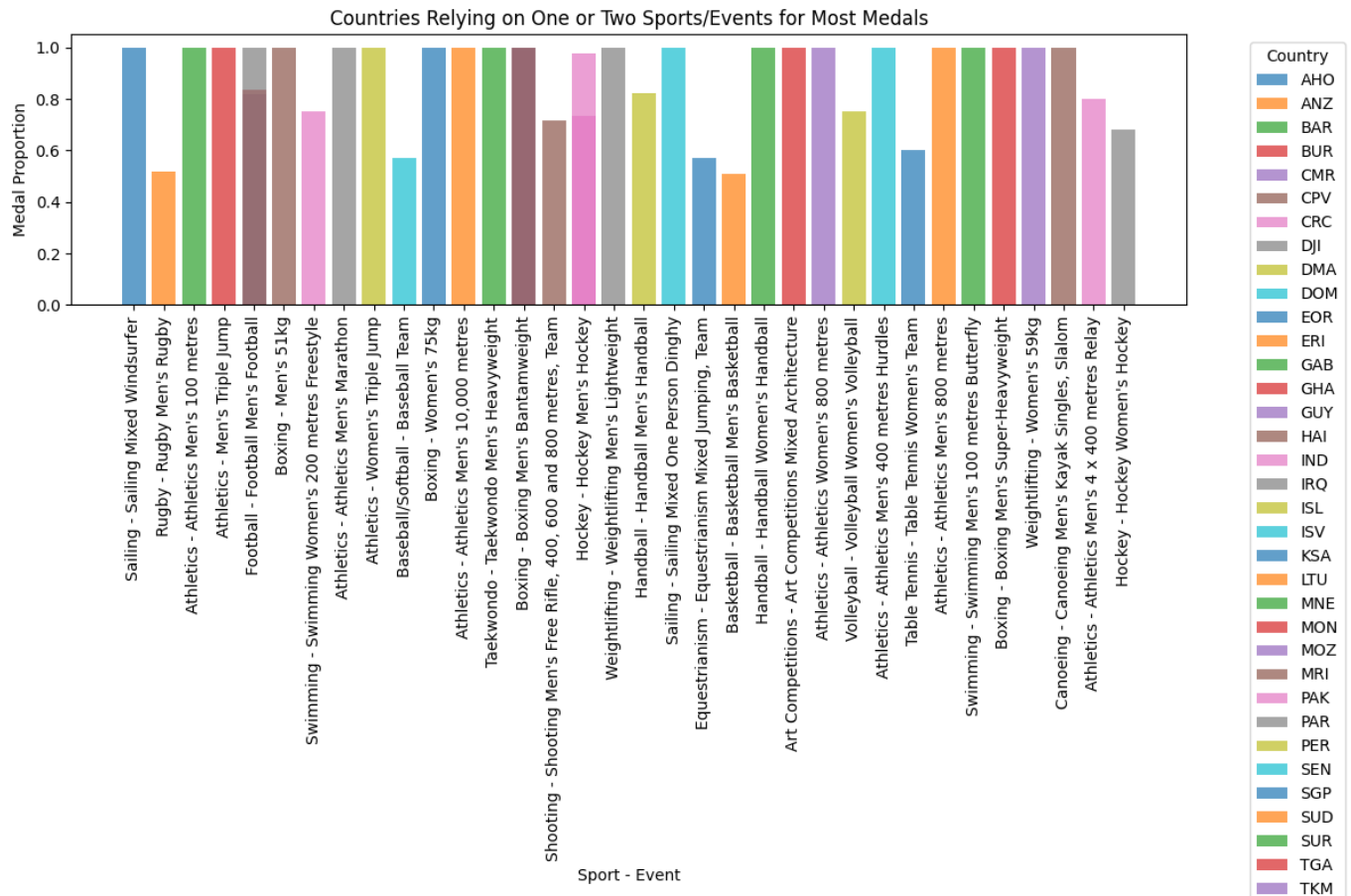


Figure 10: Countries with biased medal distribution

From this graph, we can see that many countries rely on one or two sports/events for their medal tally. This can be inspiring for the countries' Olympics committees.

6 Task III : Predicting First Medal Countries

6.1 Problem Analysis

More than 60 countries have still yet to win an Olympic medal.

We are required to predict how many countries will win their first medal in the 2028 Los Angeles Olympics, their probability of winning a medal, and their winning rate compared to each other.

6.2 Model Establishment

To establish the model(Codes are in the appendix):

Firstly, we filter out those countries that won medals in 1896. Then we create a feature set for each country, encode the country name, and use the countries' first medal winning data split as the training set and testing set to train the model using Random Forest Classifier. Secondly, we predict the probability of 'no medal' country winning a medal in the 2028 Los Angeles Olympics.

Finally, we calculate the winning rate of those country and compare them.

6.3 Result Analysis

Number of countries that have not won a medal: 79

This value may be slightly more than the real number of countries that have not won a medal, as some countries which didn't win a medal in the past may have vanished from the earth due to historical or political reasons, and some countries may be the same country but have different country names which are not detected by the 'country map' preproccession. However, the calculation is rather accurate. From the information on the internet, there are more than 60 or 70 countries that have not won a medal in the Olympics.

If we assume that a probability of winning a medal greater than 0.3 is a good chance of winning a medal, then 7 countries have a good chance of winning a medal in the 2028 Los Angeles Olympics. The definition of good chance varies from person to person, and the threshold can be adjusted according to the actual situation. A probability distribution histogram is also provided in **Figure 11**, which can indicate some interesting factors.

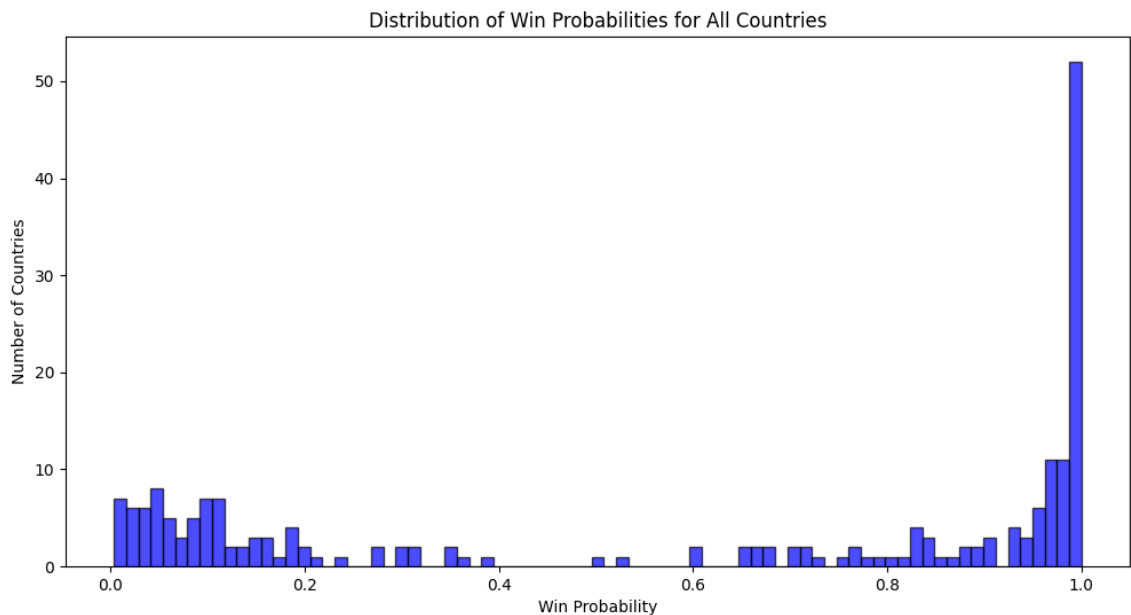


Figure 11: Distribution of Win Probabilities for All Countries

The winning probability of those countries shows a distribution trend of the opposite of gaussian distribution : countries with low and high probability are more than countries with medium probability, and the trend is like a curve which is low in the middle and high at both ends, and the winning probability of some countries is extremely high. That can be explained in real world, as countries which have not got a medal are more likely to be new countries participating in the Olympics or countries that have attended a lot with still no medal, which is very likely to win a medal in the future.

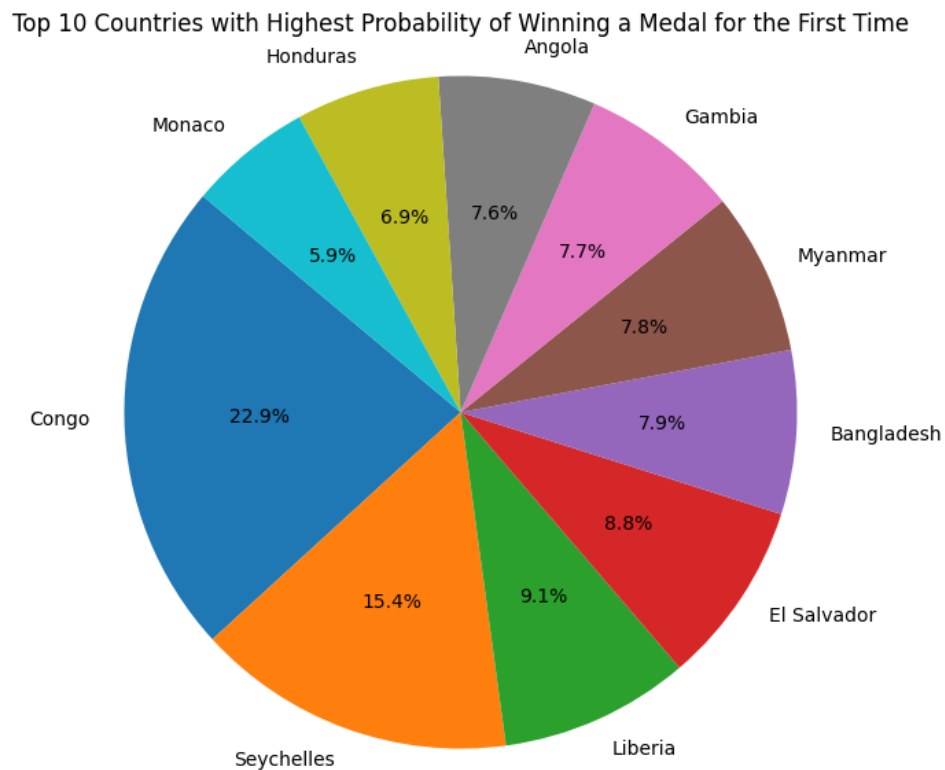


Figure 12: First Medal Results

We've also drawn a pie chart to show the winning odds of top 10 countries in **Figure 12**.

Congo has a extremely high winning probability of 90.28% , while Liberia has a decent winning probability of 35.97%. Those two countries are typical African countries which have long strived to win a medal but failed due to various reasons. We sincerely hope that they will get their first medal in the 2028 Los Angeles Olympics.

7 Task IV : Great Coach Effect

7.1 Problem Analysis

According to the reference:

Two possible examples of this include Lang Ping[2], who coached volleyball teams from both the U.S. and China to championships, and the sometimes-controversial gymnastics coach, Béla Károlyi[3], who coached Romania and then the U.S. women's teams with great success.

Bela and Marta instructed the US gym team in the following years:

1981: They defected to the US and began coaching gymnastics soon after.

1984: They coached Mary Lou Retton, who won America's first gold medal in the Olympic all-around.

1991: They trained Kim Zmeskal, who became the world champion.

1996: Marta coached the "Magnificent Seven" member Kerri Strug to win the Olympic team gold.

2000: Bela was named national team coordinator.

2001 - 2016: Marta served as the coordinator of the US women's gymnastics national team.

Since there's too little data of Lang Ping, we decide to use Bela and Marta as an example to train data. We use comprehensive methods developed by previous tasks and establish the last part of our 'PRE' model : **Legendary Index**.

7.2 Model Establishment : Legendary Index

We rearrange the data by Sport, NOC, and Sex.

Since we do not have specific information on which events or disciplines Bela and Marta coached, we only consider sports, gender, and country. We assign a value of 1 if a great coach is coaching and 0 if not. Then we split the data into training, testing dataset and train our model by year according to specific parameters. Finally, We use the model to calculate Legendary Index to indicate greatness.

Listing 6: Part of the Model

```

from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import accuracy_score
legendary_years
= [1976, 1981, 1984, 1989, 1996, 2000, 2004, 2008, 2012, 2016, 2020]
us_women_gymnastics.loc[:, 'Legendary_Coach']
= us_women_gymnastics['Year'].apply(lambda x: 1 if x in legendary_years else 0)
...
params = {
    'objective': 'reg:squarederror',
    'learning_rate': 0.01,
    'max_depth': 6,
    'min_child_weight': 1,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'n_estimators': 1000,
    'random_state': 42,
    'early_stopping_rounds': 10
}

```

```

model = xgb.XGBRegressor(**params)
model.fit(X_train, y_train,
          eval_set=[(X_train, y_train), (X_test, y_test)],
          verbose=False)
...
medal_counts0 =
    ↪ group.groupby('Year')['Medal'].value_counts().unstack(fill_value=0)
medal_counts0 = medal_counts0.reindex(columns=['Gold', 'Silver',
    ↪ 'Bronze', 'No medal'], fill_value=0)
medal_counts0['Total'] = medal_counts0[['Gold', 'Silver',
    ↪ 'Bronze']].sum(axis=1)
medal_counts0 = medal_counts0[['Total', 'Gold', 'Silver', 'Bronze', 'No
    ↪ medal']].reset_index()

medal_counts0 = medal_counts0.reset_index()

medal_counts0['Sport'] = sport
medal_counts0['NOC'] = noc
medal_counts0['Sex'] = sex

legendary_coach_results = pd.DataFrame()

for (sport, noc, sex), group in grouped_all_medal_counts:
    group['Legendary_Index'] = model.predict(group[['Year', 'Total', 'Gold',
    ↪ 'Silver', 'Bronze', 'No medal']])

legendary_threshold = 0.68

group['Predicted_Legendary_Coach'] = group['Legendary_Index'] >
    ↪ legendary_threshold

result = group[['Year', 'Sport', 'NOC', 'Sex', 'Legendary_Index',
    ↪ 'Predicted_Legendary_Coach']]

```

7.3 Lang Ping: The Great Coach

Fortunately, We've found Lang Ping in the Predicted Great Coach.

Besides Lang Ping, other great achievements are also detected. Although we don't have enough time to check every achievements with the information of the internet, every data we have checked is really great achievement.

Chinese Table Tennis and Gymnastics are selected as the most famous examples. 1996, 2000, 2008 and 2020 are great years for Chinese Male Table Tennis team, while 2024 is not although they got the gold medal.

We've even found interesting data of controversial results caused by the penalty awarded by

8 Other Insights

8.1 Try to predict the medal of the 2032 Olympics

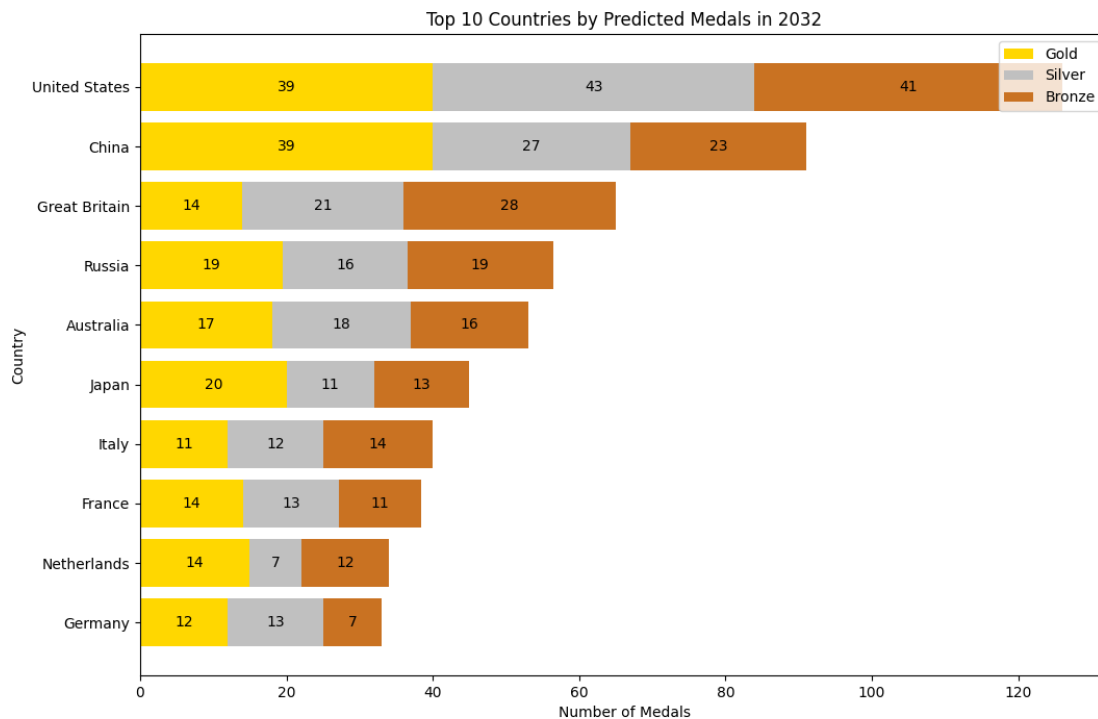


Figure 14: Predicted Medal Counts for the 2032 Olympics

The prediction of 2032 Olympics is rather difficult because of the uncertainty of the future and almost no information of the 2032 Olympics. In **Figure 14**, we've tried to make some prediction with our comprehensive PRE model, taking all effects into account. Although the prediction may not be accurate enough, it can still serve as a reference.

8.2 Greatness and the strategy of the country Olympic committees

During the estimation of whether a country's achievement can be due to **Great Coaches**, we set a parameter of **legendary** to evaluate the successness of a team on a certain event. Combined with the **Great Athletes** mentioned before, we can now tell apart whether it's the great athlete(s) or the great coach that mainly contribute to great achievements. Also, we can exclude these factors to more explicitly discuss other elements which influence various countries' performance. The graph above shows a part of the calculation result. We find out that a high **legendary index** is of higher probability be due to **Great Coach** when a **Great Athlete** is absent.

As to the strategy of the country Olympic committees, we give the calculation result as the suggestion that a **legendary index** higher than 0.68 means a high potential of future success. It means that the committees can invest more in this certain sport/event, for example, hiring a **Great**

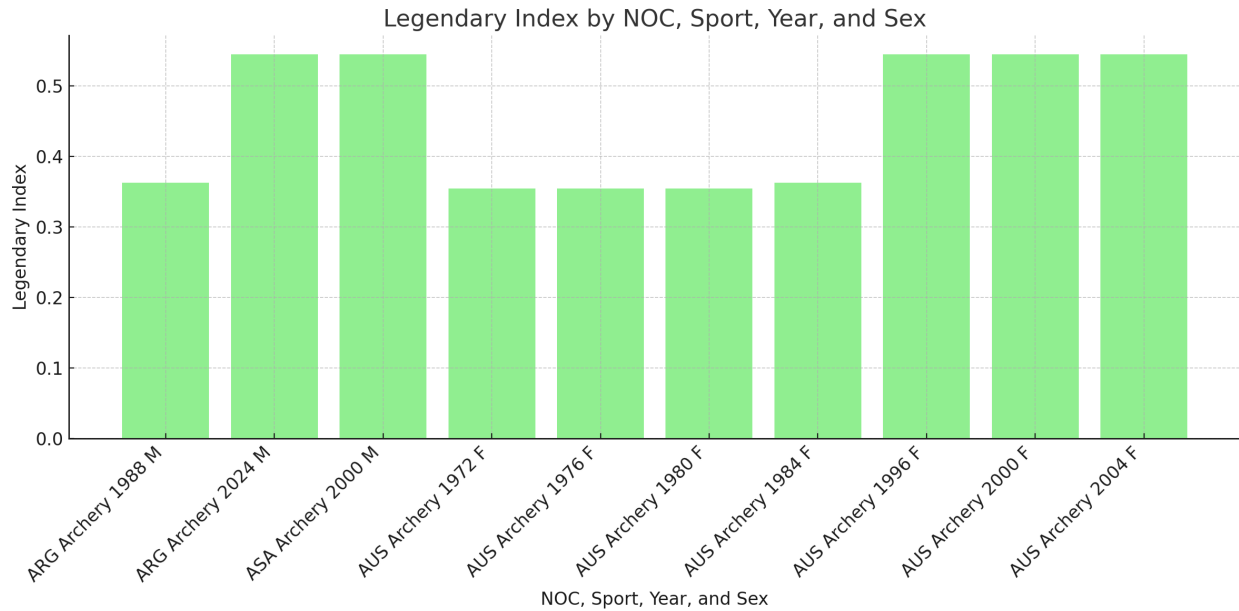


Figure 15: Legendary index of certain events

Coach or cultivating **Great Athletes**, and hope for great achievements.

9 Strength and Weakness

9.1 Strength

- We analyze the problem based on historical data with a relatively long period, so that the model we established is of great validity.
- Our model is fairly robust due to our careful cleaning of raw data and careful selection of parameter according to real Olympic events.
- Via Programming software, we simulate the 2028 LA Olympic results. The outcome is vivid for us to understand the predicition.
- We come up with testing methods to ensure precision, like KS test. Hence an overall correctness can be within reach.
- We take multiple factors into consideration during modeling, for instance, female ratio and the continuity of participation, which add to the accuracy of our prediction model

9.2 Weakness

- We choose specific values for some peculiar parameters, for instance, we assume the proportion of female athletes of the USA team in 2028. Those values may not be reasonable in reality.

- Although we investigate a lot in the given data, the model is so complicated that need to be studied further with more than these data. For example, the population of a country will impact the number of potential athletes, and further influence the country's future performance in the Olympics.
- Limited to data amount, **Great Coach Model** is of certain sensitivity, because we can only get access to data of one or two Olympics before the great coach start tutoring.

9.3 Further Discussions

9.3.1 Possible Optimization

Due to lack of data, our model only use past data with few factors taken into account, such as year, sex and event, while neglecting other factors such as a country's population, GDP and willingness for medals. To be more accurate, we can add more data into our model.

Also, we can adapt more nonlinear models, besides methods like **Random Forest**, and compare their results to be more precise on prediction.

9.3.2 Challenges in practical application

This model is potentially useful for country Olympic committees for arrangement strategies, but it only cares about data of performance, but training and taking part in the Olympics is a complex system, countless factors such as money investment and training period need to be considered and balanced when deciding whether to invest in a certain event or not.

Thus, this model can be combined with data analysis on other factors to fully cover the real situation, and contribute to practical application.

References

- [1] All team members. *Github Repository(Open after the Contest)*. <https://github.com/Code-Test-Source/MCM2025/>. 2025.
- [2] Nielsen. *2024 virtual-medal-table-forecast*. <https://www.nielsen.com/news-center/2024/virtual-medal-table-forecast/>. 2024.
- [3] *Olympics.com Biography, Lang Ping*. <https://olympics.com/en/athletes/ping-lang>. 2024.
- [4] *paris-2024-olympics-federscherma-and-coni-at-the-lunge-after-filippo-macchis-silver-medal*. <https://sportal.eu/2024/07/29/paris-2024-olympics-federscherma-and-coni-at-the-lunge-after-filippo-macchis-silver-medal/>. 2024.
- [5] *USA Gymnastics Hall of Fame*. <https://usagym.org/halloffame/inductee/coaching-team-bela-martha-karolyi/>. 2024.
- [6] *USWNT Head Coach Emma Hayes Named The Best FIFA Women's Coach of 2024*. <https://www.ussoccer.com/stories/2024/12/girma-horan-and-naeher-named-to-the-best-fifa-womens-11-hayes-named-the-best-fifa-womens-coach-of-the-year>. 2024.

Appendices

For all Code, find them in the Github Repository in the first Reference.

Appendix A Core Code for Predicting 2028 Olympics Medal

Input Python source:

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from scipy.stats import ks_2samp
import numpy as np
import matplotlib.pyplot as plt
import chardet
from sklearn.metrics import mean_squared_error, r2_score
import seaborn as sns
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
country_mapping = {
    'Soviet Union': 'Russia',
    ...,
    'Libya': 'Liberia',
}
noc_mapping = {
    'URS': 'RUS',
    ...
    'EOR': 'RUS',
}
country_codes = {
    'EOR': 'Russia',
    'ROC': 'Russia',
    'AFG': 'Afghanistan',
    'ALB': 'Albania',
    'ALG': 'Algeria',
    'AND': 'Andorra',
    'ANG': 'Angola',
    'ANT': 'Antigua and Barbuda',
    ...
    'VIE': 'Vietnam',
    'VIN': 'Saint Vincent and the Grenadines',
    'YEM': 'Yemen',
    'ZAM': 'Zambia',
    'ZIM': 'Zimbabwe'
}
athletes['NOC'] = athletes['NOC'].replace(noc_mapping)
```

```

medal_counts['NOC'] = medal_counts['NOC'].replace(country_mapping)
# Split Host in hosts.csv into City and Country
hosts[['City', 'NOC']] = hosts['Host'].str.split(',', expand=True)
hosts['NOC'] = hosts['NOC'].str.strip()
# Map NOC in athletes.csv to countries
athletes['NOC'] = athletes['NOC'].map(country_codes).fillna(athletes['NOC'])
# Preprocess athletes data
athletes['Sex'] = athletes['Sex'].map({'M': 1, 'F': 0})
athletes_agg = athletes.groupby(['Year', 'NOC']).agg({
    'Name': lambda x: x.nunique(),
    'Sex': lambda x: x.mean(),
    'Sport': lambda x: x.nunique(),
    'Event': lambda x: x.nunique()
}).reset_index()
athletes_agg.rename(columns={'Name': 'Num_Athletes', 'Sex': 'Female_Ratio',
    ↳ 'Sport': 'Num_Sports', 'Event': 'Num_Events'}, inplace=True)
# Convert 'Year' column to int in medal_counts
medal_counts['Year'] = medal_counts['Year'].astype(int)
# Merge athletes_agg and medal_counts
data = pd.merge(athletes_agg, medal_counts, on=['Year', 'NOC'], how='left')
# Read specific rows and columns from programs.csv
programs_sum = pd.read_csv(programs_file_path, skiprows=lambda x: x not in [0,
    ↳ 72, 73, 74], usecols=range(4, programs.shape[1]))
# Transform the data into the required format
programs_sum = programs_sum.transpose().reset_index()
programs_sum.columns = ['Year', 'Total_Events', 'Total_Discipline',
    ↳ 'Total_Sports']
# Convert 'Year' column to int in programs_sum
programs_sum['Year'] = programs_sum['Year'].astype(int)
# Merge programs_sum with data on Year
data = pd.merge(data, programs_sum, on='Year', how='left')
# Determine if the country is the host for each year
data['Is_Host'] = data.apply(lambda row: 1 if row['NOC'] in hosts[hosts['Year']
    ↳ == row['Year']]['NOC'].values else 0, axis=1)
data = data.fillna(0)
# Prepare features and target with additional variables
X = data[['Year', 'Is_Host', 'Num_Athletes', 'Female_Ratio', 'Num_Sports',
    ↳ 'Num_Events', 'Total_Events', 'Total_Discipline', 'Total_Sports']]
y = data[['Total', 'Gold', 'Silver', 'Bronze']].apply(pd.to_numeric,
    ↳ errors='coerce')
# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.19,
    ↳ random_state=42)
# Random Forest model
rf_model = RandomForestRegressor(n_estimators=1000, random_state=42)
rf_model.fit(X_train, y_train)
rf_predictions = rf_model.predict(X_test)

```

```

# Create new variable matrix for XGBoost
X_train_rf = X_train.copy()
X_test_rf = X_test.copy()
for i, col in enumerate(['Total', 'Gold', 'Silver', 'Bronze']):
    X_train_rf[f'RF_Predictions_{col}'] = rf_model.predict(X_train)[: , i]
    X_test_rf[f'RF_Predictions_{col}'] = rf_predictions[: , i]
# XGBoost model
xgb_model = XGBRegressor(n_estimators=1000, random_state=42)
xgb_model.fit(X_train_rf, y_train)
xgb_predictions = xgb_model.predict(X_test_rf)
# K-S Test
ks_stat, p_value = ks_2samp(y_test, xgb_predictions)
# Predict medals for 2028 for all countries
future_data_all = data[(data['Year'] == 2024) & (data['NOC'] != 'France') |
    ↳ ((data['Year'] == 2020) & (data['NOC'] == 'France')) | ((data['Year'] ==
    ↳ 2016) & (data['NOC'] == 'Russia'))].copy()
future_data_all['Year'] = 2032
# Predict using Random Forest
rf_predictions_all = rf_model.predict(future_data_all[['Year', 'Is_Host',
    ↳ 'Num_Athletes', 'Female_Ratio', 'Num_Sports', 'Num_Events',
    ↳ 'Total_Events', 'Total_Discipline', 'Total_Sports']])
future_data_all['RF_Predictions_Total'] = rf_predictions_all[:, 0]
future_data_all['RF_Predictions_Gold'] = rf_predictions_all[:, 1]
future_data_all['RF_Predictions_Silver'] = rf_predictions_all[:, 2]
future_data_all['RF_Predictions_Bronze'] = rf_predictions_all[:, 3]
# Ensure future_data_all has the same columns as X_train_rf
future_data_rf_all = future_data_all[['Year', 'Is_Host', 'Num_Athletes',
    ↳ 'Female_Ratio', 'Num_Sports', 'Num_Events', 'Total_Events',
    ↳ 'Total_Discipline', 'Total_Sports',
    ↳ 'RF_Predictions_Total', 'RF_Predictions_Gold',
    ↳ 'RF_Predictions_Silver',
    ↳ 'RF_Predictions_Bronze']]
# Predict using XGBoost
future_predictions_all = xgb_model.predict(future_data_rf_all)
# Aggregate predicted medal counts for each country
future_data_all['Predicted_Gold'] = future_predictions_all[:, 1]
future_data_all['Predicted_Silver'] = future_predictions_all[:, 2]
future_data_all['Predicted_Bronze'] = future_predictions_all[:, 3]
# Aggregate medal counts for each country
future_medal_totals = future_data_all.groupby('NOC')[['Predicted_Gold',
    ↳ 'Predicted_Silver', 'Predicted_Bronze']].sum().reset_index()
# Sort by predicted total medals and save to CSV
future_medal_totals['Predicted_Total'] = future_medal_totals['Predicted_Gold'] +
    ↳ future_medal_totals['Predicted_Silver'] +
    ↳ future_medal_totals['Predicted_Bronze']
future_medal_totals_sorted =
    ↳ future_medal_totals.sort_values(by='Predicted_Total', ascending=False)

```

Appendix B Core Code for Great Athletes and Key Events

Input Python source:

```

medal_data = data[data['Medal'] != 'No medal']
medal_counts = medal_data.groupby(['NOC', 'Sport',
    ↪ 'Event'])['Medal'].count().reset_index(name='Medal Count')
total_medals_per_country = medal_counts.groupby('NOC')['Medal
    ↪ Count'].sum().reset_index(name='Total Medals')
medal_data_with_totals = pd.merge(medal_counts, total_medals_per_country,
    ↪ on='NOC')
medal_data_with_totals['Medal Proportion'] = medal_data_with_totals['Medal
    ↪ Count'] / medal_data_with_totals['Total Medals']
threshold = 0.5
reliant_countries = medal_data_with_totals[medal_data_with_totals['Medal
    ↪ Proportion'] > threshold]
reliant_countries_top_sports = reliant_countries.groupby('NOC').apply(lambda x:
    ↪ x.nlargest(2, 'Medal Proportion')).reset_index(drop=True)

```

Appendix C Core Code for Predicting First Medal Countries

Input Python source:

```

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
country_mapping = {
    'Soviet Union': 'Russia',
    ...
    'ROC': 'Russia',
}
data['NOC'] = data['NOC'].replace(country_mapping)
# Filter out the data for countries that won medals in 1896
data_1896_winners = data[(data['Year'] == 1896) & (data['Total'] >
    ↪ 0)]['NOC'].unique()
data = data[~data['NOC'].isin(data_1896_winners)]
# Create a feature set for each country
country_features = data.groupby('NOC').agg({
    'Year': ['count', 'min', 'max'],
    'Female_Ratio': 'mean',
    'Num_Athletes': 'sum',
    'Num_Sports': 'sum',
    'Num_Events': 'sum',
    'Total': 'sum'
}).reset_index()
country_features.columns = ['NOC', 'Participations', 'First_Year', 'Last_Year',
    ↪ 'Total_Athletes', 'Female_Ratio', 'Total_Sports', 'Total_Events',
    ↪ 'Total_Medals']

```

```

# Add a column for whether the country has won a medal
country_features['Has_Won'] = country_features['Total_Medals'] > 0
# Encode the NOC column
le = LabelEncoder()
country_features['NOC'] = le.fit_transform(country_features['NOC'])
# Split the data into training and testing sets
X = country_features.drop(columns=['Has_Won', 'Total_Medals'])
y = country_features['Has_Won']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    ↪ random_state=42)
# Train a RandomForestClassifier
clf = RandomForestClassifier(n_estimators=10000, random_state=42)
clf.fit(X_train, y_train)
# Predict the probability of winning a medal for each country
country_features['Win_Probability'] = clf.predict_proba(X)[: , 1]
# Print the countries with the highest probability of winning a medal for the
    ↪ first time
first_time_winners = country_features[country_features['Has_Won'] ==
    ↪ False].sort_values(by='Win_Probability', ascending=False)
# Decode the NOC column back to country names
first_time_winners['NOC'] = le.inverse_transform(first_time_winners['NOC'])
# Plot a pie chart for the top 10 countries with the highest probability of
    ↪ winning a medal for the first time
top_10_countries = first_time_winners.head(10)
# Calculate the number of countries that have not won a medal
num_countries_no_medal = country_features[country_features['Has_Won'] ==
    ↪ False].shape[0]
# Calculate the number of countries that have a probability of winning a medal
    ↪ greater than 0.3 and have not won a medal
num_countries_prob_above_3 =
    ↪ first_time_winners[first_time_winners['Win_Probability'] > 0.3].shape[0]

```

Appendix D Core Code for Great Coach Effect

Input Python source:

```

from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import accuracy_score
us_women_gymnastics = df[(df['NOC'] == 'USA') & ((df['Sport'] == 'Gymnastics'))
    ↪ & (df['Sex'] == 'F')]
legendary_years = [1976, 1981, 1984, 1989, 1996, 2000, 2004, 2008, 2012, 2016,
    ↪ 2020]
us_women_gymnastics.loc[:, 'Legendary_Coach'] =

```



```

    ↪ us_women_gymnastics['Year'].apply(lambda x: 1 if x in legendary_years else
    ↪ 0)
medal_counts = us_women_gymnastics.groupby(['Year',
    ↪ 'Legendary_Coach'])['Medal'].value_counts().unstack(fill_value=0)
medal_counts['Total'] = medal_counts[['Gold', 'Silver', 'Bronze']].sum(axis=1)
medal_counts = medal_counts[['Total', 'Gold', 'Silver', 'Bronze', 'No medal']]
medal_counts = medal_counts.reset_index()
sports_list = programs_df['Sport'].unique()
all_medal_counts = pd.DataFrame()
for sport in sports_list:
    sport_data = df[df['Sport'] == sport]
    grouped_data = sport_data.groupby(['NOC', 'Sex'])
    for (noc, sex), group in grouped_data:
        medal_counts0 =
            ↪ group.groupby('Year')['Medal'].value_counts().unstack(fill_value=0)
        medal_counts0 = medal_counts0.reindex(columns=['Gold', 'Silver',
            ↪ 'Bronze', 'No medal'], fill_value=0)
        medal_counts0['Total'] = medal_counts0[['Gold', 'Silver',
            ↪ 'Bronze']].sum(axis=1)
        medal_counts0 = medal_counts0[['Total', 'Gold', 'Silver', 'Bronze', 'No
            ↪ medal']].reset_index()
        medal_counts0 = medal_counts0.reset_index()
        medal_counts0['Sport'] = sport
        medal_counts0['NOC'] = noc
        medal_counts0['Sex'] = sex
        all_medal_counts = pd.concat([all_medal_counts, medal_counts0],
            ↪ ignore_index=True)
X = medal_counts[['Year', 'Total', 'Gold', 'Silver', 'Bronze', 'No medal']]
y = medal_counts['Legendary_Coach']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    ↪ random_state=42)
params = {
    'objective': 'reg:squarederror',
    'learning_rate': 0.01,
    'max_depth': 6,
    'min_child_weight': 1,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'n_estimators': 1000,
    'random_state': 42,
    'early_stopping_rounds': 10
}
model = xgb.XGBRegressor(**params)
model.fit(X_train, y_train,
        eval_set=[(X_train, y_train), (X_test, y_test)],
        verbose=False)
grouped_all_medal_counts = all_medal_counts.groupby(['Sport', 'NOC', 'Sex'])

```

```
legendary_coach_results = pd.DataFrame()
for (sport, noc, sex), group in grouped_all_medal_counts:
    group['Legendary_Index'] = model.predict(group[['Year', 'Total', 'Gold',
    ↳ 'Silver', 'Bronze', 'No medal']])
    legendary_threshold = 0.68
    group['Predicted_Legendary_Coach'] = group['Legendary_Index'] >
    ↳ legendary_threshold
    result = group[['Year', 'Sport', 'NOC', 'Sex', 'Legendary_Index',
    ↳ 'Predicted_Legendary_Coach']]
    legendary_coach_results = pd.concat([legendary_coach_results, result],
    ↳ ignore_index=True)
```

Report on Use of AI

1. OpenAI ChatGPT (Dec 14, 2024 version, ChatGPT-4o)

Query1: <insert the exact wording you input into the AI tool>

Output: <insert the complete output from the AI tool>

2. Github Copilot (2024 version, ChatGPT-4o, o1(preview))

Query1: <insert the exact wording you input into the AI tool>

Output: <insert the complete output from the AI tool>

3. Deepseek (2025 version)

Query1: Find some references about Medal prediction in the Olympic Games

Output: Here are some references related to medal prediction in the Olympic Games, focusing on methodologies, data-driven approaches, and key insights from academic research and practical applications:

1. Academic Papers on Medal Prediction

1.1. “Predicting Olympic Medal Counts: The Effects of Economic Development on Olympic Success”

- **Authors:** Daniel K. N. Johnson, Ayfer Ali
- **Journal:** *Social Science Quarterly* (2004)
- **Summary:** This paper explores the relationship between a country’s economic development (GDP, population) and its Olympic medal count. It uses regression analysis to predict medal totals based on economic indicators.
- **Key Insight:** Economic factors (GDP per capita, population size) are strong predictors of Olympic success.
- **Link:** DOI:10.1111/j.0038-4941.2004.08501009.x

1.2. “A Machine Learning Approach to Predicting Olympic Medal Counts”

- **Authors:** S. S. R. Abidi, M. Manzoor
 - **Conference:** *International Conference on Machine Learning and Applications* (2018)
 - **Summary:** This study applies machine learning algorithms (e.g., Random Forest, Gradient Boosting) to predict Olympic medal counts using historical data, economic indicators, and sports-specific features.
 - **Key Insight:** Machine learning models outperform traditional regression models in accuracy, especially when incorporating sport-specific data.
 - **Link:** IEEE Xplore
-

1.3. “The Role of Host Nation Advantage in Olympic Success”

- **Authors:** Andrew B. Bernard, Meghan R. Busse
 - **Journal:** *The Review of Economics and Statistics* (2004)
 - **Summary:** This paper investigates the “host nation advantage” and its impact on medal counts. It uses historical data to quantify the boost in performance for host countries.
 - **Key Insight:** Host nations typically win an additional 1.3% of total medals, with the effect lasting for two Olympic cycles.
 - **Link:** DOI:10.1162/003465304323023723
-

2. Data-Driven Approaches

2.1. “Gracenote Olympic Medal Predictions”

- **Source:** Gracenote Sports
 - **Summary:** Gracenote uses a proprietary algorithm to predict Olympic medal counts based on historical performance, recent results, and athlete rankings. Their predictions are widely cited in the media.
 - **Key Insight:** The model incorporates both individual athlete performance and team dynamics.
 - **Link:** Gracenote Sports
-

2.2. “Nielsen’s Olympic Performance Predictions”

- **Source:** Nielsen Sports
 - **Summary:** Nielsen uses a combination of historical data, athlete rankings, and economic indicators to predict medal counts for each country.
 - **Key Insight:** The model highlights the importance of recent performance trends in predicting future success.
 - **Link:** Nielsen Sports
-

3. Practical Applications and Case Studies

3.1. “Predicting Olympic Medal Counts Using Bayesian Hierarchical Models”

- **Authors:** Andrew Gelman, Jennifer Hill
- **Book:** *Data Analysis Using Regression and Multilevel/Hierarchical Models* (2006)
- **Summary:** This book includes a case study on predicting Olympic medal counts using Bayesian hierarchical models, which account for country-level and sport-level variability.

- **Key Insight:** Hierarchical models provide more accurate predictions by capturing the nested structure of Olympic data (e.g., athletes within countries, sports within events).
 - **Link:** Cambridge University Press
-

3.2. “The Economics of the Olympics: Predicting Medal Counts”

- **Authors:** Stefan Szymanski
 - **Journal:** *Journal of Sports Economics* (2010)
 - **Summary:** This paper examines the economic and demographic factors that influence Olympic success, using regression analysis to predict medal counts.
 - **Key Insight:** Population size and GDP are the most significant predictors, but sport-specific investments also play a role.
 - **Link:** DOI:10.1177/1527002510368101
-

4. Open Data Sources for Medal Prediction

4.1. Olympic Medal Dataset

- **Source:** Kaggle
 - **Summary:** A comprehensive dataset containing historical Olympic medal counts, athlete information, and event results.
 - **Link:** Kaggle Olympic Dataset
-

4.2. World Bank Economic Indicators

- **Source:** World Bank
- **Summary:** Provides GDP, population, and other economic indicators that can be used as predictors in medal count models.
- **Link:** World Bank Data

4. Doubao (2025 version)

Query1: The year Bela and Marta instructed the us gym team

Output: Bela and Marta instructed the US gym team in the following years:

- **1981:** They defected to the US and began coaching gymnastics soon after.
- **1984:** They coached Mary Lou Retton, who won America’s first gold medal in the Olympic all - arounds.
- **1991:** They trained Kim Zmeskal, who became the world champion.
- **1996:** Marta coached the ”Magnificent Seven” member Kerri Strug to win the Olympic team gold.
- **2000:** Bela was named national team coordinator.
- **2001 - 2016:** Marta served as the coordinator of the US women’s gymnastics national team.