The Winning Formula: Modeling Athlete and Country Success at the Olympics

2024-12-06

Introduction

What makes home advantage an "advantage"? Is it the crowd cheering your name instead of your rival's, the familiarity with the arena, the comfort of hometown cuisine, or simply the presence of your support system? The debate around home advantage in sports, whether it truly exists and how significantly it impacts outcomes, has always been controversial yet captivating. In this project, we dive into the world of Olympic events to explore whether athletes competing at home and host countries gain an edge, as well as to identify other key factors that influence Olympic outcomes.

We aim to identify key predictors influencing two facets of Olympic performance: (a) individual athlete success in winning medals and (b) a country's overall standing in the medal rankings. Our analysis focuses on understanding how factors like home advantage, prior participation, and player demographics influence these outcomes. The methodology involves building classification models to predict medal outcomes at the athlete level and ranking predictions at the country level. By interpreting model coefficients, assessing variable importance, and visualizing key findings, we aim to provide a comprehensive understanding of what drives success at the Olympics.

Data Description

The dataset used in this analysis contains biographical details of athletes and their participation records from Olympic Games held between 1896 and 2016. Key data includes each athlete's demographic information, participation in different Olympic Games, events, and medals won. To facilitate predictive modeling, extensive feature engineering was performed, resulting in two distinct branches of data: **athlete-level features** and **country-level features**. For detailed explanations on how these features were engineered, please refer to the corresponding code sections and the Appendix.

Athlete-Level Features

The original dataset provided key variables for each athlete:

- **Name**: The athlete's name.
- **Sex**: Gender of the athlete (Male/Female).
- **Age**: Age of the athlete during the Olympic event.
- **Height**: Height in centimeters.
- **Weight**: Weight in kilograms.
- **NOC**: National Olympic Committee (country) to which the athlete belongs.
- **Games**: The year and season of the Olympics (e.g., "2016 Summer").
- **Sport**: The sport in which the athlete competed.
- **Event**: The specific event in which the athlete participated.

- **Medal**: Type of medal won (Gold, Silver, Bronze, None).

In addition to these original datapoints, new features were engineered:

- **home_advantage**: A binary variable indicating whether the athlete competed in their home country (1 if yes, 0 otherwise).
- **lag_games_played**: Number of Olympic Games the athlete had previously participated in.
- **lag_has_medal**: A binary feature indicating whether the athlete won a medal in the previous Olympics (1 if yes, 0 otherwise).
- **lag_total_medal_score**: The athlete's medal score in the previous Olympics, calculated as a weighted sum (Gold = 3, Silver = 2, Bronze = 1).
- **lag_aggregated_medal_score**: The cumulative total medal score achieved by the athlete in all previous Olympic Games up to the current event.

Country-Level Features

To analyze country-level performance, the dataset was aggregated by **NOC** and **Year**, generating the following features:

- **avg_age**: Average age of athletes from the country.
- **avg_height**: Average height of athletes from the country.
- **avg_weight**: Average weight of athletes from the country.
- **percent_medalists_lag**: Percentage of athletes from the country who were medalists in the previous Olympics.
- **events_participated**: Total number of events the country participated in during the current Olympics.
- **sports_participated**: Total number of sports the country participated in.
- **home_advantage**: A binary variable indicating whether the country hosted the Olympic Games (1 if yes, 0 otherwise).
- **male_count/female_count**: Count of male and female athletes representing the country.
- **rank**: The country's ranking based on total medals won.
- **lag_total_medals**: Number of medals won by the country in the previous Olympics.
- **lag_gold/silver/bronze_medals**: Number of gold, silver, and bronze medals won by the country in the previous Olympics.
- **lag_weighted_medals**: Weighted medal count for the previous Olympics, with weights assigned as Gold = 3, Silver = 2, Bronze = 1.
- **avg_total_medal_last_5**: Average total medals won by the country in the past five Olympic Games.
- **avg_gold/silver/bronze_last_5**: Average count of gold, silver, and bronze medals in the past five Olympics.
- **avg_rank_last_5**: Average rank of the country in the medal table for the last five Olympics.
- **medals_per_events_lag**: Ratio of medals won to events participated in during the previous Olympics.
- **medals_per_sports_lag**: Ratio of medals won to sports participated in during the previous Olympics.
- **rank_bin**: Response variable for country rank prediction, created by dividing countries into bins of 5 based on their rankings.

Training & Testing Data Split

The data was split into training and testing sets, with 20% reserved for the hold-out testing set to evaluate model performance. For individual-level predictions, an additional dummified version of the dataset was created for use in linear discriminant analysis.

Model Selection & Methodology

In this project, we employ two separate predictive modeling approaches to analyze success in the Olympic Games at both the individual and national levels. The primary focus of these models is not just prediction but also understanding how different factors influence success. Thus, while accuracy on the 20% hold-out test set is considered, the emphasis is on interpretability and inference, rather than solely optimizing predictive power.

Player-Level Models

For predicting the outcomes of individual athletes per event (i.e., the type of medal an athlete may win in that event), two models were developed: Linear Discriminant Analysis (LDA) and Multinomial Logistic Regression. Although the multinomial logistic regression demonstrated superior accuracy, both models contribute to a deeper understanding of the factors impacting athlete performance.

Linear Discriminant Analysis (LDA)

The LDA model was built using a dummified version of the data, which converts categorical predictors into numerical format, making it suitable for LDA. The response variable is the athlete's medal type ("Gold," "Silver," "Bronze," or "None"). The predictors include:

- Season: Whether the event took place during the Summer or Winter Olympics.
- Home Advantage: Whether the athlete competed in their home country.
- Sex: Athlete gender.
- Age, Height, Weight: Biographical details of the athlete.
- Lagged Performance Metrics: Including "lag_games_played", "lag_has_medal", and "lag_aggregated_medal_score".

The model demonstrated an accuracy of 0.85 on the hold-out test set.

Multinomial Logistic Regression

The Multinomial Logistic Regression model was built using the same set of features as the LDA, but with the original (undummified) dataset. Unlike LDA, this model allowed for the inclusion of interaction terms, particularly examining how home advantage interacts with other factors like Sex and Season. The goal was to capture the non-linear effects and nuanced relationships between predictors and the type of medal won.

The accuracy of the Multinomial Logistic Regression model on the hold-out set was 0.854, slightly outperforming LDA. Given its higher accuracy and the richer interpretability from model coefficients, the Multinomial Logistic Regression was chosen as the final model for player-level analysis.

Country-Level Models

To predict the ranking of countries in the Olympic Games, a Random Forest Classifier was employed. The response variable is rank_bin, which categorizes countries into rank groups of five (e.g., Top 5, Top 10, etc.). This approach allows for the identification of countries' general standing without the need for precise ranking, simplifying the problem into classification bins.

Random Forest Model

The Random Forest model utilized a wide set of features derived from the aggregated country-level data, including:

- Demographic Metrics: avg_age, avg_height, avg_weight of athletes representing the country.
- Performance Metrics: lag_total_medals, lag_gold_count, lag_silver_count, lag_bronze_count, and lag_weighted_medals (previous Olympics performance).
- Recent Trends: Rolling averages for medals over the last five Olympics ("avg_total_medals_last_5," "avg_gold_count_last_5," etc.).
- Participation Metrics: events_participated, sports_participated, and the number of athletes by gender (male_count and female_count).
- Home Advantage: Whether the country was the host nation for the event.
- Additional Metrics: percent_medalists_lag (percentage of returning medalists), medals_per_events_lag, and medals_per_sport_lag (providing insights into efficiency in medal-winning per event and per sport).

The Random Forest model was constructed with 500 trees (ntree) and a mtry value of 5 (number of features considered at each split). The model achieved an accuracy of 0.698 on the hold-out test set. A subsequent grid search for optimal hyperparameters was performed, but no configuration surpassed the initial model. Thus, the original model was retained.

Results & Interpretation

Multinomial Logistic Regression for Predicting Athlete Medal Outcomes

The accuracy of the multinomial logistic regression model on the hold-out test set is **0.854**. While this accuracy is impressive, it largely stems from correct predictions for the majority "None" class (i.e., athletes not winning medals). Predicting which medal an athlete will win (if they do win) remains challenging, as medal-winning is a relatively rare event. Despite this, we can extract meaningful insights into how different factors impact individual performances based on significant predictors.

Below, I summarize the key findings from the model's results (For a full summary, refer to the stargazer table in the Appendix):

1. Home Advantage (`home_advantage`)

- **Bronze**: -0.032 (not significant)
- **Silver**: 0.307
- **Gold**: 0.755

Home advantage has a significant positive effect on winning Silver and Gold medals. Specifically, competent athletes competing in their home country are more likely to win Silver or Gold, with a stronger effect observed for Gold medals. The coefficient for Bronze is small and not statistically significant, suggesting home advantage does not significantly influence Bronze medal outcomes.

2. Lag Games Played (`lag_games_played`)

- **Bronze**: -0.087
- **Silver**: -0.175
- **Gold**: -0.371

The number of **previous Olympic games played** by an athlete has a negative effect on their likelihood of winning a medal, with the impact being stronger for Gold. This is counterintuitive to the common belief that experience benefits performance. Instead, the results suggest that athletes with multiple prior Olympic participations may face challenges in achieving new medals, potentially due to age-related performance declines.

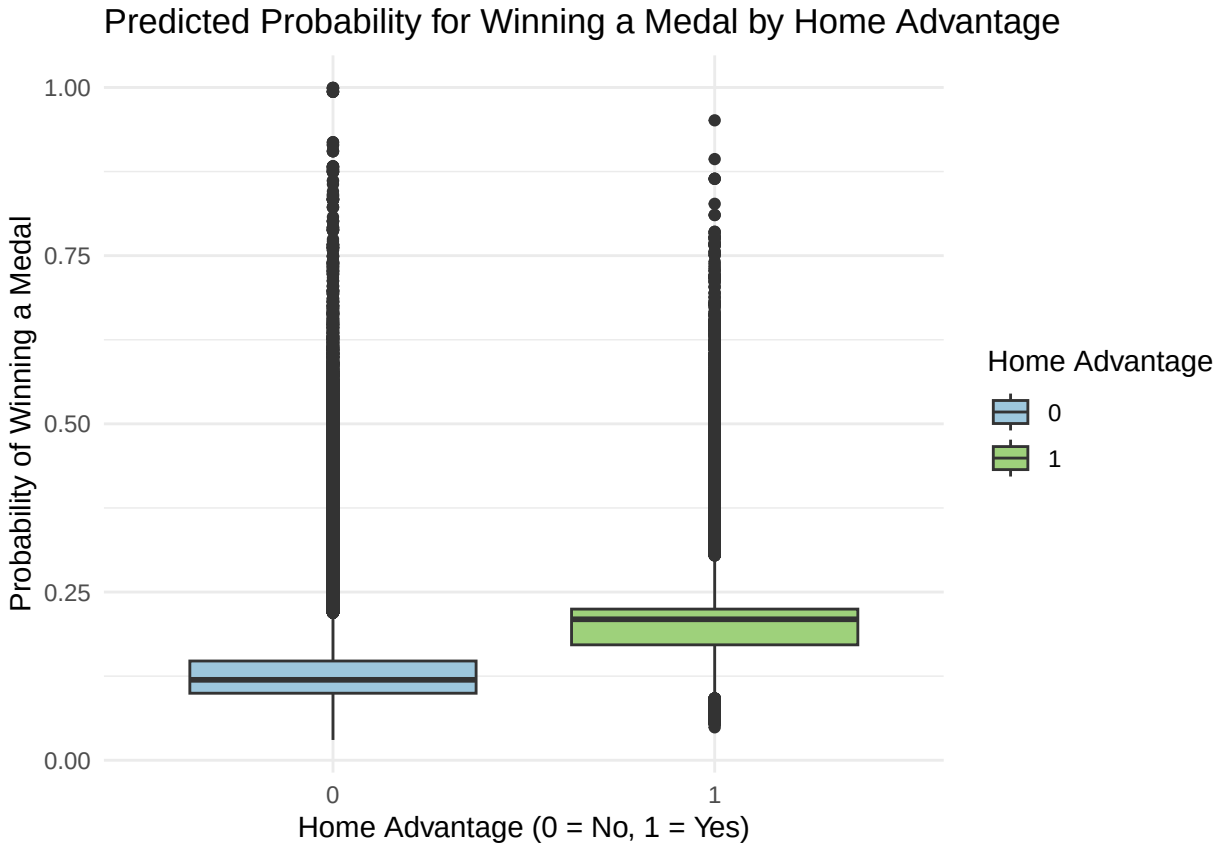
3. Lag Has Medal (`lag_has_medal`)

- **Bronze**: 1.063
- **Silver**: 1.310
- **Gold**: 1.612

Winning a medal in a **previous Olympics** has a strong positive effect on the likelihood of winning a medal in the current Olympics, with the effect being most substantial for Gold medals. This suggests that prior medalists are more likely to continue their success, highlighting the significance of prior achievement as an indicator of current performance.

4. Interactions Involving Home Advantage

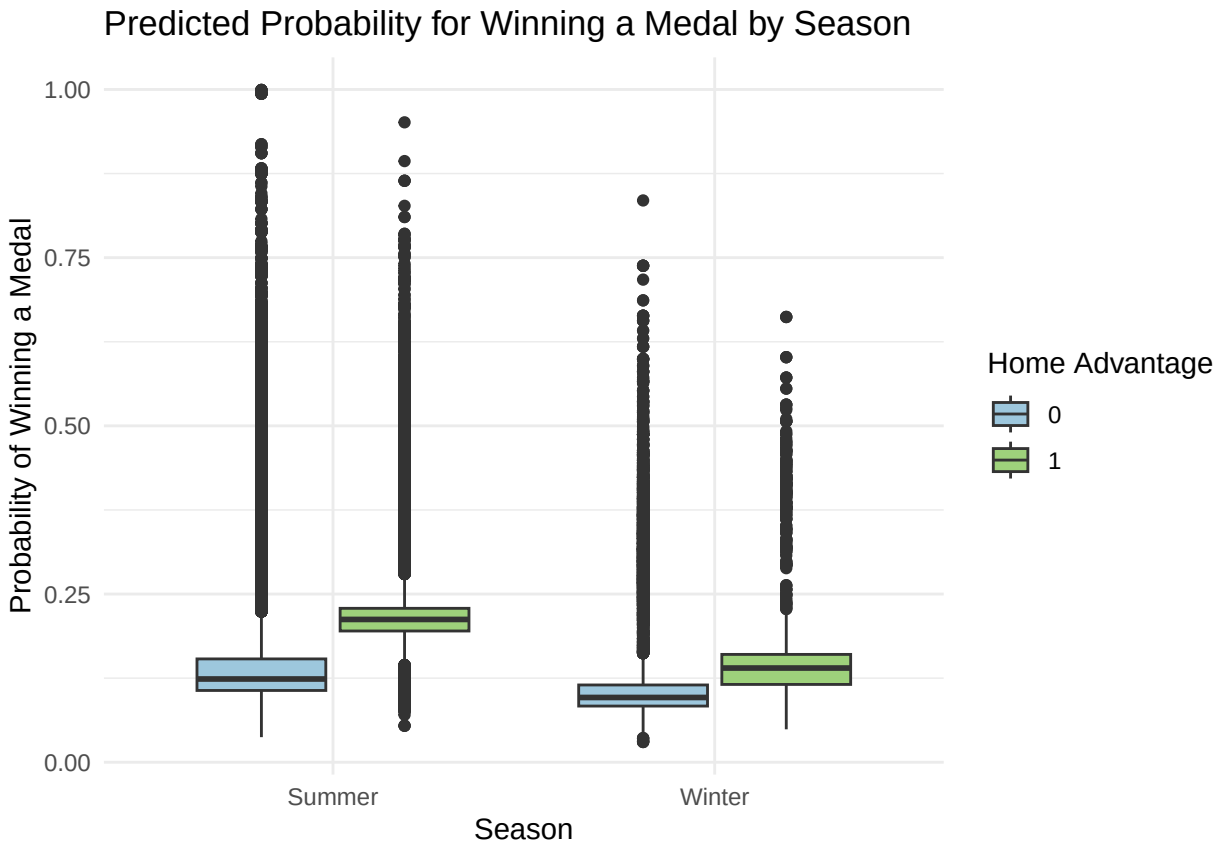
Based on the training model, I included predicted winning probabilities in the dataset and visualized the effect of **home advantage**. It is evident that athletes with home advantage have a **higher average winning probability** compared to those without it. This effect is consistent across different athlete groups, although the magnitude of the effect varies by season, sex, and specific sports events which we will explore in the following section.



Home Advantage: Season Interaction (Defaut = Summer)

- **Bronze**: -0.294
- **Silver**: -0.147
- **Gold**: -0.184

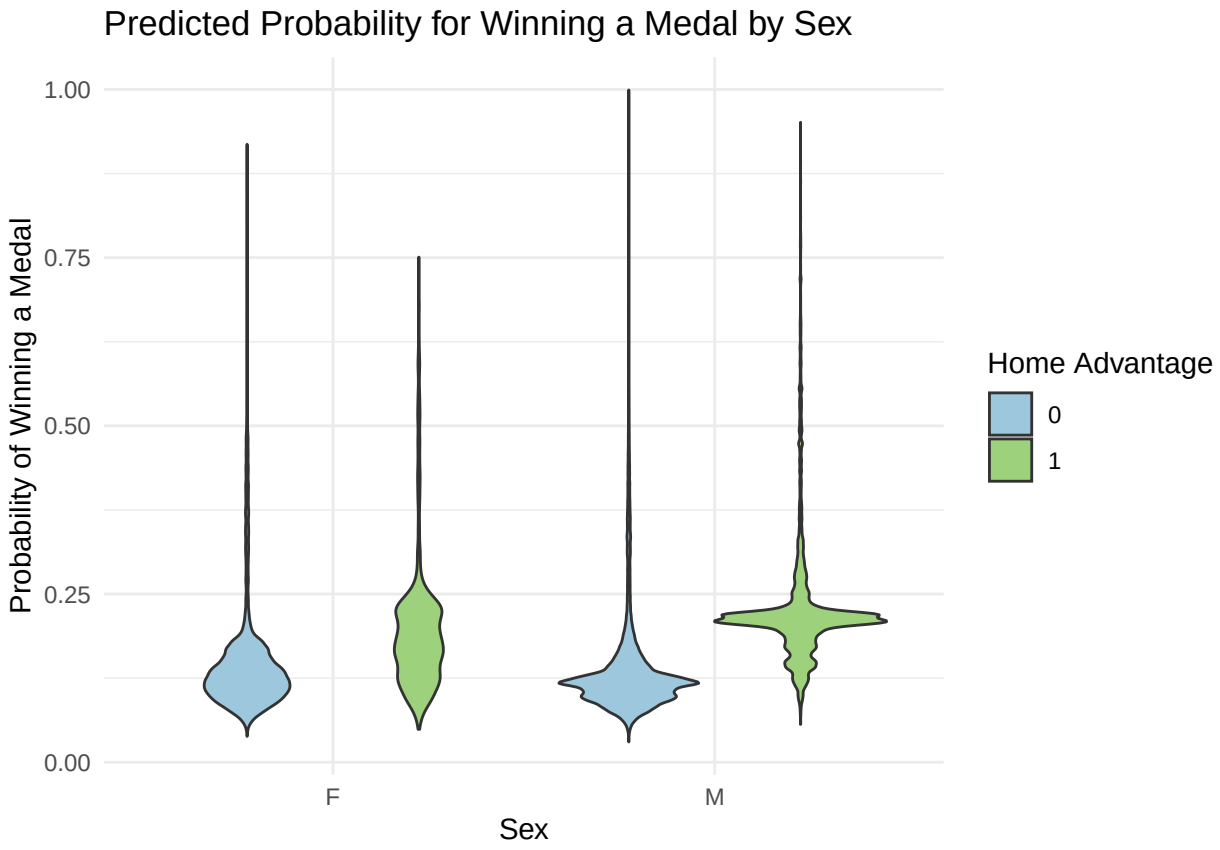
The interaction between **home advantage** and competing in the **Winter Olympics** has a negative impact across all medal categories, indicating that home advantage is less beneficial for athletes in the Winter Olympics compared to the Summer Olympics. This result is further demonstrated in the visualization of predicted winning probabilities, where the boost from home advantage is notably larger for Summer events than for Winter events.



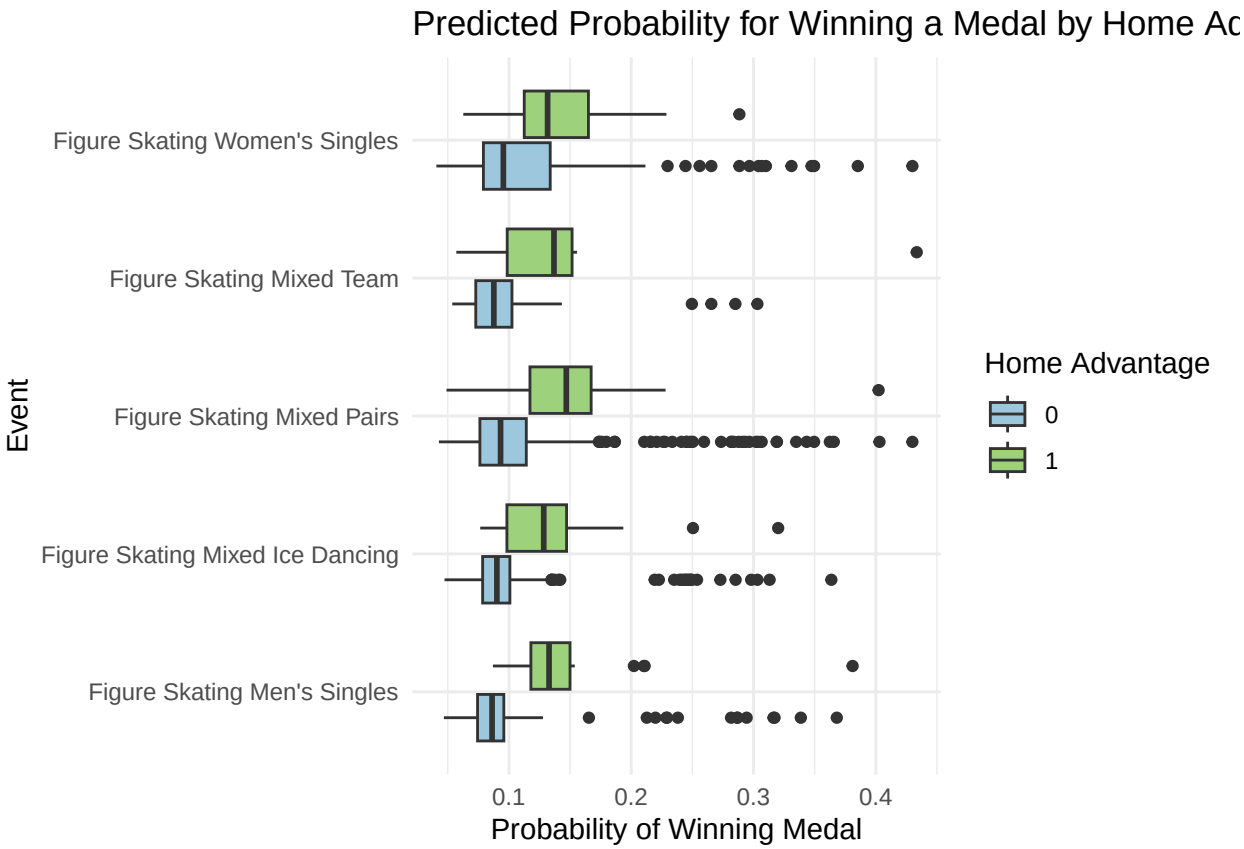
Home Advantage: Sex Interaction (Default = Female)

- **Bronze**: 0.537
- **Silver**: 0.378
- **Gold**: 0.115 (not significant)

The interaction between **home advantage** and being **male** is positive for Bronze and Silver, suggesting that male athletes tend to benefit more from home advantage compared to female athletes. However, for Gold medals, the coefficient is small and not statistically significant, implying no substantial difference in home advantage effects based on sex. The violin plot shows that the distribution of winning probabilities for males with home advantage generally shifts upwards compared to females with home advantage.



Home Advantage: Figure Skating The effect of **home advantage** on winning probabilities is analyzed for different events within **figure skating** (my favorite sport) as an example of how home advantage might vary across events. The analysis reveals that home advantage provides a significant boost in winning probability across all events, with the smallest boost observed in **ice dancing** and the largest boost in **pairs figure skating**.



Random Forest Model for Country Ranking

The random forest model achieved an accuracy of **0.698** on the hold-out test set. While this accuracy is not exceptionally high due to the inherent uncertainty in predicting country rankings—driven by changes in participation, development, and external factors—it provides valuable insights. The variable importance plot highlights key predictors that significantly influence a country's ranking in the Olympics.

Key Insights from Variable Importance

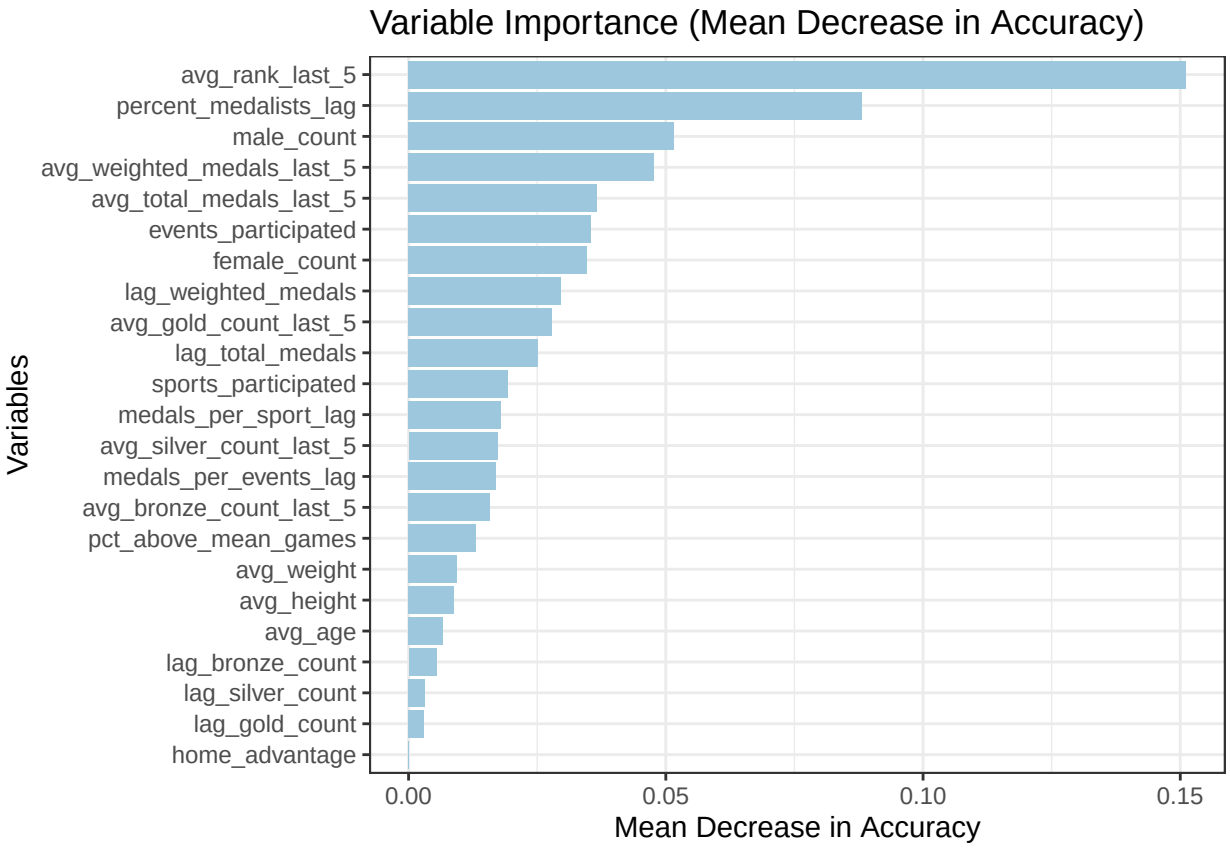
- `avg_rank_last_5`:
  The most important predictor in the model, indicating that a country's average ranking over the last five Olympics is the strongest determinant of its current ranking. This reflects the **consistent dominance** of historically successful Olympic nations and underscores the importance of long-term performance trends.

- `percent_medalists_lag`:
  The proportion of medalists in the previous Olympics significantly correlates with current performance, suggesting that recent success in producing medalists translates into a higher likelihood of sustained success. This aligns with findings from the **individual medal prediction model**, where being a previous medalist strongly influences individual outcomes. As country rankings aggregate individual results, this relationship is expected and logical.

- `avg_weighted_medals_last_5` and `avg_total_medals_last_5`:
  These features summarize a country's **historical medal performance**. They reflect a nation's competence in sports development, encompassing infrastructure, training systems, and resource allocation. These factors capture the broader capabilities of a country in fostering high-performing athletes.

- **`events_participated` and `sports_participated`**:
  The **breadth of participation** across events and sports positively impacts rankings. Countries with broader representation in diverse events have more opportunities to secure medals, contributing to higher overall rankings.

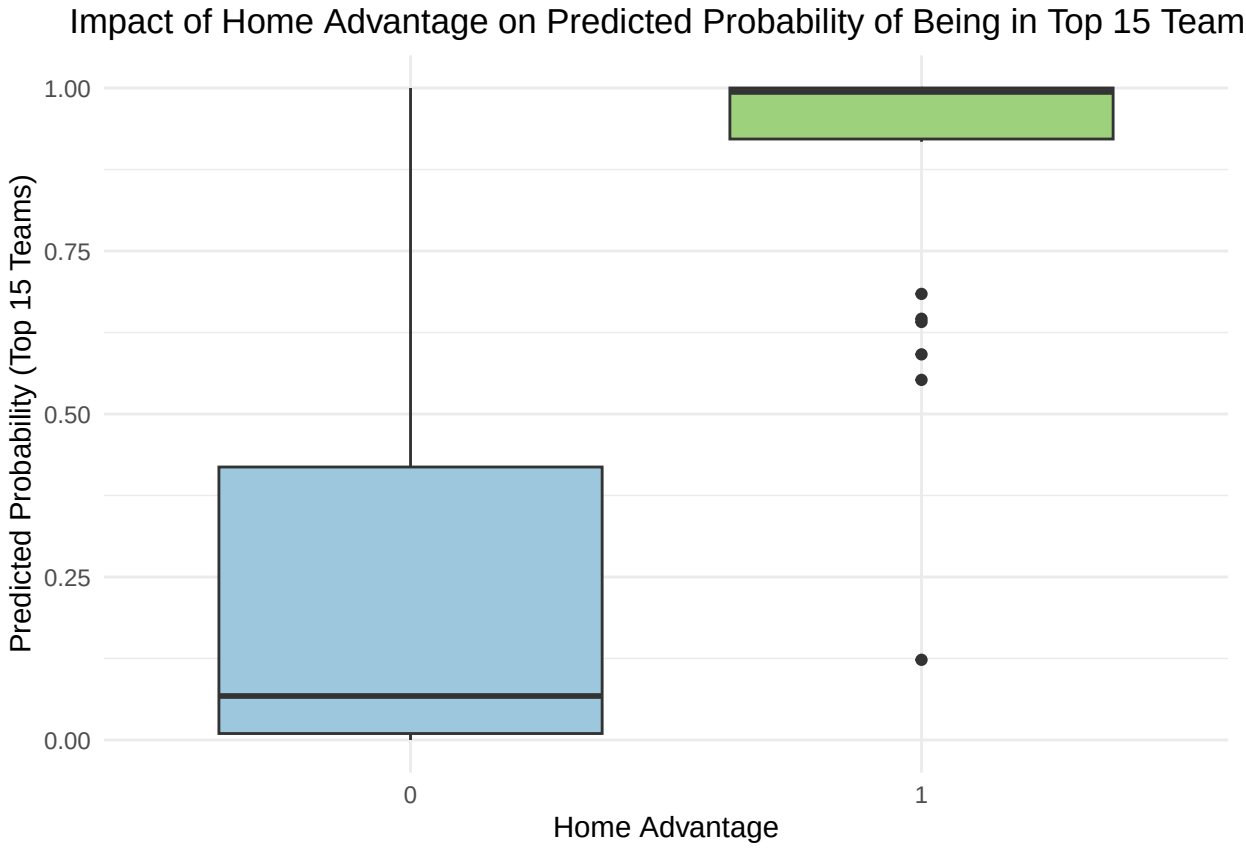
- **`home_advantage`**:
  Surprisingly, home advantage is the lowest-ranked feature, indicating that hosting the Olympics has a relatively small direct effect on predicting a country's rank compared to historical trends and team composition. However, this variable warrants a deeper investigation.



Home Advantage and Being a Top 15 Team

Although the variable importance metric suggests home advantage has limited predictive value for overall rankings, further analysis reveals its significant influence on the **probability of being a top 15 team**.

To examine this, I calculated the predicted probability of being in the top 15 for teams with and without home advantage. The results indicate that **hosting the Olympics almost guarantees a top 15 finish**. But why doesn't this make home advantage a crucial predictor of country rankings overall?



Deeper Analysis: Hosting Nations in the Dataset After data wrangling, I created a table summarizing the 23 countries that hosted the Olympics in the dataset. Key findings include:

- **17 out of 23 hosting nations** were ranked in the top 15 teams in more than **60% of the Olympics** they participated in.
- Some nations, including **France, Britain, Germany, and the United States**, have consistently been in the top 15 in **every Olympic Games** they participated in. These countries significantly boost the overall probability of a hosting nation being in the top 15.

Why Home Advantage Appears Less Important for Rankings Hosting the Olympics requires **substantial investment** and often holds **diplomatic significance**, meaning that the countries chosen to host are often economically advanced nations. These countries typically have:

1. **Strong sports training pipelines**: A well-established infrastructure to develop and support top-tier athletes.
2. **Cultural enthusiasm for sports**: A societal emphasis on athletic achievement.
3. **Consistent Olympic success**: Many hosting countries were already top-performing nations, independent of home advantage.

Thus, while home advantage appears to strongly influence the probability of being in the top 15, its effect on overall ranking prediction diminishes because hosting countries tend to be strong performers irrespective of hosting.

Conclusion

This project explores the diverse factors influencing Olympic success at both individual and national levels, shedding light on what separates winners from the rest.

Key Findings

- **Home Advantage**:
  - At the individual level, home advantage significantly boosts the likelihood of winning Silver or Gold medals, with minimal impact on Bronze outcomes.
  - At the national level, hosting the Olympics nearly guarantees a top 15 finish, despite home advantage having limited direct influence on rankings. This is because hosting nations often possess strong sports infrastructure and a history of success.
- **Historical Performance**:
  Consistency is critical. Prior success, such as past medals or rankings, is the strongest predictor of future outcomes, highlighting the importance of sustained investment in sports training and infrastructure.
- **Breadth of Participation**:
  Countries with broader participation across events and sports achieve better rankings, as greater representation increases medal opportunities.

Limitations

This analysis has inherent limitations. While the model provides valuable insights, more advanced machine learning techniques could improve accuracy, albeit at the cost of interpretability. Additionally, external factors like geopolitical events or economic shifts are not explicitly modeled but may influence outcomes. Lastly, the models capture correlations but cannot fully account for the complexity of athletic competition.

Appendix:

Feature Engineering Details

Athlete-Level Features

1. **home_advantage**: Created by geocoding the hosting city using the `gggeoencoder` package and mapping the resulting country to the NOC code.
2. **lag_games_played**: For each athlete, the number of games played before the current Olympic event was calculated using `row_number() - 1` to create a lagged value.
3. **lag_has_medal**: Constructed by grouping athletes by their unique ID and Olympic year, then checking if they had won any medals (`Medal != None`). This feature was lagged using the `lag()` function with a default of 0.
4. **lag_total_medal_score**: Medals were assigned scores (Gold = 3, Silver = 2, Bronze = 1), and the sum was taken for each Olympic year. The lag of this score was used as a feature.
5. **lag_aggregated_medal_score**: Created using a cumulative sum (`cumsum()`) of `lag_total_medal_score` to track the athlete's total medal score up to the current event.

Country-Level Features

1. **avg_age, avg_height, avg_weight**: These were calculated by taking the mean of athlete attributes grouped by country (NOC) and year.
2. **percent_medalists_lag**: Calculated as the percentage of athletes who were medalists in the previous Olympic Games for each country.
3. **events_participated, sports_participated**: Derived by counting distinct events and sports participated in by each country for a given year.
4. **lag_total_medals, lag_gold/silver/bronze_medals**: Number of medals won in the previous Olympics, derived using the `lag()` function on the respective medal counts.
5. **avg_total_medal_last_5, avg_gold/silver/bronze_last_5**: Calculated using rolling averages (`zoo::rollapply`) over the last five Olympic Games for each country.
6. **medals_per_events_lag, medals_per_sports_lag**: Computed by dividing the lagged medal counts by the respective number of events or sports participated in.
7. **rank_bin**: Countries were grouped into bins of five based on their rankings in the medal tally for classification purposes.

Multinomial Logistic Regression Output Summary

Codes

```
# library(tidyverse)
# library(tidyr)
# library(readr)
# library(tidygeocoder)
# library(caret)
# library(zoo)
# library(nnet)
# library(MASS)
# library(randomForest)
# library(ggplot2)
#
# olympic_data <- read_csv("Dataset 2 - Olympic events data.csv")
#
# ############################## Geocoding and Host Information ##############################
#
# # Geocode unique cities to identify host countries
# unique_cities <- olympic_data %>%
#   dplyr::select(City) %>%
#   distinct()
#
# geocoded_cities <- unique_cities %>%
#   mutate(address = City) %>%
#   geocode(address = address, method = "osm", full_results = TRUE) %>%
#   mutate(Country = sub(".*, ([^,]+)$", "\\1", display_name)) %>%
#   dplyr::select(City, Country)
#
# # Add country information to the main dataset
# data <- olympic_data %>%
#   left_join(geocoded_cities, by = "City")
#
```

Multinomial Logistic Regression Results

	Estimate	Std. Error	z value	Pr(>|z|)
Bronze:(Intercept)	-6.161	0.245	-25.123	0
Bronze:home_advantage	-6.628	0.248	-26.721	0
Bronze:SeasonWinter	-7.290	0.247	-29.499	0
Bronze:SexM	-0.032	0.083	-0.381	0.703
Bronze:Age	0.307	0.073	4.205	0.00003
Bronze:Height	0.755	0.063	11.892	0
Bronze:Weight	-0.277	0.030	-9.364	0
Bronze:lag_games_played	-0.289	0.030	-9.512	0
Bronze:lag_has_medal	-0.271	0.031	-8.829	0
Bronze:lag_aggregated_medal_score	-0.386	0.026	-14.579	0
Bronze:home_advantage:SeasonWinter	-0.431	0.027	-16.018	0
Bronze:home_advantage:SexM	-0.406	0.027	-14.856	0
Silver:(Intercept)	-0.004	0.002	-2.364	0.018
Silver:home_advantage	-0.001	0.002	-0.815	0.415
Silver:SeasonWinter	-0.006	0.002	-2.993	0.003
Silver:SexM	0.016	0.002	9.495	0
Silver:Age	0.018	0.002	10.432	0
Silver:Height	0.021	0.002	12.510	0
Silver:Weight	0.011	0.001	9.354	0
Silver:lag_games_played	0.012	0.001	9.947	0
Silver:lag_has_medal	0.014	0.001	11.483	0
Silver:lag_aggregated_medal_score	-0.087	0.017	-5.074	0.00000
Silver:home_advantage:SeasonWinter	-0.175	0.018	-9.740	0
Silver:home_advantage:SexM	-0.371	0.020	-18.643	0
Gold:(Intercept)	1.063	0.042	25.418	0
Gold:home_advantage	1.310	0.040	33.042	0
Gold:SeasonWinter	1.612	0.038	42.015	0
Gold:SexM	0.073	0.009	8.522	0
Gold:Age	0.101	0.008	13.249	0
Gold:Height	0.158	0.007	23.499	0
Gold:Weight	-0.294	0.112	-2.633	0.008
Gold:lag_games_played	-0.147	0.099	-1.486	0.137
Gold:lag_has_medal	-0.184	0.092	-2.009	0.044
Gold:lag_aggregated_medal_score	0.537	0.091	5.901	0
Gold:home_advantage:SeasonWinter	0.378	0.081	4.700	0.00000
Gold:home_advantage:SexM	0.115	0.071	1.623	0.105

Figure 1: Multinomial Logistic Regression Summary

```
# # Map NOC codes to host countries
# noc_mapping <- tibble(
#   Country = c(
#     "España", "United Kingdom", "België / Belgique / Belgien", "France", "Canada",
#     "Norge", "United States", "Suomi / Finland", "Australia", "Sverige", "   ",
#     " ", "Italia", " ", "Brasil", "E  ", "Österreich",
#     "Bosna i Hercegovina /            ", "México", "Deutschland",
#     "  ", "Nederland", "Schweiz/Suisse/Svizzera/Svizra"
#   ),
#   NOC_host = c(
#     "ESP", "GBR", "BEL", "FRA", "CAN", "NOR", "USA", "FIN",
#     "AUS", "SWE", "RUS", "JPN", "ITA", "CHN", "BRA", "GRE",
#     "AUT", "BIH", "MEX", "GER", "KOR", "NED", "SUI"
#   )
# )
#
# # Integrate host country information and calculate home advantage
# data <- data %>%
#   left_join(noc_mapping, by = "Country") %>%
#   mutate(
#     home_advantage = as.integer(NOC == NOC_host)
#   ) %>%
#   dplyr::select(-Country)
#
# ############################### Player-Level Features ###############################
#
# # Number of games played by each player
# player_games <- olympic_data %>%
#   distinct(ID, Games) %>%
#   group_by(ID) %>%
#   summarize(games_played = n(), .groups = "drop")
#
# # Lagged games played for each player
# player_games_lag <- olympic_data %>%
#   distinct(ID, Games) %>%
#   group_by(ID) %>%
#   arrange(Games) %>%
#   mutate(lag_games_played = row_number() - 1) %>%
#   ungroup()
#
# # Lagged player medal info
# player_medals_lag <- olympic_data %>%
#   mutate(Medal = replace_na(Medal, "None")) %>%
#   group_by(ID, Games) %>%
#   summarize(has_medal = as.integer(any(Medal %in% c("Gold", "Silver", "Bronze"))), .groups = "drop") %
#   group_by(ID) %>%
#   arrange(Games) %>%
#   mutate(lag_has_medal = lag(has_medal, default = 0)) %>%
#   ungroup()
#
# # Lagged player medal scores
# player_scores_lag <- olympic_data %>%
#   mutate(
```

```
#     medal_score = case_when(
#       Medal == "Gold" ~ 3,
#       Medal == "Silver" ~ 2,
#       Medal == "Bronze" ~ 1,
#       TRUE ~ 0
#     )
#   ) %>%
#   group_by(ID, Games) %>%
#   summarize(total_medal_score = sum(medal_score, na.rm = TRUE), .groups = "drop") %>%
#   group_by(ID) %>%
#   arrange(Games) %>%
#   mutate(
#     lag_total_medal_score = lag(total_medal_score, default = 0),
#     lag_aggregated_medal_score = cumsum(lag(total_medal_score, default = 0))
#   ) %>%
#   ungroup()
#
# # Merge lagged features into main dataset
# data <- data %>%
#   mutate(
#     Medal = replace_na(Medal, "None"),
#     Medal = factor(Medal, levels = c("None", "Bronze", "Silver", "Gold"))
#   ) %>%
#   left_join(player_games, by = "ID") %>%
#   left_join(player_games_lag, by = c("ID", "Games")) %>%
#   left_join(player_medals_lag, by = c("ID", "Games")) %>%
#   left_join(player_scores_lag, by = c("ID", "Games"))
#
# # Fill missing player-level features
# data <- data %>%
#   group_by(NOC) %>%
#   mutate(
#     Age = ifelse(is.na(Age), mean(Age, na.rm = TRUE), Age),
#     Height = ifelse(is.na(Height), mean(Height, na.rm = TRUE), Height),
#     Weight = ifelse(is.na(Weight), mean(Weight, na.rm = TRUE), Weight)
#   ) %>%
#   ungroup() %>%
#   drop_na()
#
# ############################## Country-Level Features ###############################
#
# # Unique medals
# unique_medals <- data %>%
#   distinct(NOC, Year, Event, Medal, .keep_all = TRUE)
#
# # Aggregate country-level features
# country_features <- unique_medals %>%
#   group_by(NOC, Year) %>%
#   summarize(
#     total_medals = sum(Medal %in% c("Gold", "Silver", "Bronze")),
#     gold_count = sum(Medal == "Gold"),
#     silver_count = sum(Medal == "Silver"),
#     bronze_count = sum(Medal == "Bronze"),
```

```
#     weighted_medals = sum(case_when(
#       Medal == "Gold" ~ 3,
#       Medal == "Silver" ~ 2,
#       Medal == "Bronze" ~ 1,
#       TRUE ~ 0
#     )),
#     avg_age = mean(Age, na.rm = TRUE),
#     avg_height = mean(Height, na.rm = TRUE),
#     avg_weight = mean(Weight, na.rm = TRUE),
#     percent_medalists_lag = mean(lag(has_medal, default = 0), na.rm = TRUE),
#     events_participated = n_distinct(Event),
#     sports_participated = n_distinct(Sport),
#     home_advantage = first(home_advantage),
#     male_count = sum(Sex == "M", na.rm = TRUE),
#     female_count = sum(Sex == "F", na.rm = TRUE),
#     .groups = "drop"
#   ) %>%
#   arrange(desc(Year), desc(total_medals)) %>%
#   group_by(Year) %>%
#   mutate(rank = dense_rank(desc(total_medals))) %>%
#   ungroup() %>%
#   mutate(across(where(is.numeric), round, 3))
#
# # Lagged country-level features
# country_features <- country_features %>%
#   group_by(NOC) %>%
#   arrange(Year) %>%
#   mutate(
#     lag_total_medals = lag(total_medals, default = 0),
#     lag_gold_count = lag(gold_count, default = 0),
#     lag_silver_count = lag(silver_count, default = 0),
#     lag_bronze_count = lag(bronze_count, default = 0),
#     lag_weighted_medals = lag(weighted_medals, default = 0),
#     lag_events_participated = lag(events_participated, default = 1),
#     lag_sports_participated = lag(sports_participated, default = 1),
#     avg_total_medals_last_5 = zoo::rollapply(total_medals, width = 5, FUN = mean, align = "right", fi
#     avg_gold_count_last_5 = zoo::rollapply(gold_count, width = 5, FUN = mean, align = "right", fill =
#     avg_silver_count_last_5 = zoo::rollapply(silver_count, width = 5, FUN = mean, align = "right", fi
#     avg_bronze_count_last_5 = zoo::rollapply(bronze_count, width = 5, FUN = mean, align = "right", fi
#     avg_weighted_medals_last_5 = zoo::rollapply(weighted_medals, width = 5, FUN = mean, align = "righ
#     avg_rank_last_5 = zoo::rollapply(rank, width = 5, FUN = mean, align = "right", fill = 0, na.rm =
#     medals_per_events_lag = round(lag_total_medals / lag_events_participated, 3),
#     medals_per_sport_lag = round(lag_total_medals / lag_sports_participated, 3)
#   ) %>%
#   ungroup()
#
# # Mean games played per country
# mean_games_played <- data %>%
#   group_by(NOC) %>%
#   summarize(mean_games = round(mean(games_played, na.rm = TRUE), 0), .groups = "drop")
#
# # Pct of players with games above mean
# pct_above_mean_games <- data %>%
```

```
#   group_by(NOC, Year) %>%
#   summarize(
#     pct_above_mean_games = round(mean(games_played > mean(mean_games_played$mean_games), na.rm = TRUE
#     .groups = "drop"
#   )
#
# country_features <- country_features %>%
#   left_join(pct_above_mean_games, by = c("NOC", "Year")) %>%
#   group_by(NOC) %>%
#   mutate(
#     avg_age = ifelse(is.na(avg_age), mean(avg_age, na.rm = TRUE), avg_age),
#     avg_height = ifelse(is.na(avg_height), mean(avg_height, na.rm = TRUE), avg_height),
#     avg_weight = ifelse(is.na(avg_weight), mean(avg_weight, na.rm = TRUE), avg_weight)
#   ) %>%
#   ungroup() %>%
#   drop_na() %>%
#   mutate(
#     rank_bin = ceiling(rank / 5),
#     rank_bin = as.factor(rank_bin)
#   )
#
# ############################### Train/Test Splits ###############################
#
# # Country-level features split
# set.seed(2024)
# train_index_cf <- createDataPartition(country_features$rank, p = 0.8, list = FALSE)
# train_cf <- country_features[train_index_cf, ]
# test_cf <- country_features[-train_index_cf, ]
#
# # Player-level features split
# data_model <- data
# train_index_ind <- createDataPartition(data_model$lag_total_medal_score, p = 0.8, list = FALSE)
# train_ind <- data_model[train_index_ind, ]
# test_ind <- data_model[-train_index_ind, ]
#
# ############################### Data Transformation for Modeling ################################
#
# dummies <- dummyVars(
#   Medal ~ home_advantage * Season + home_advantage * Sex + Age +
#     Height + Weight + lag_games_played + lag_has_medal + lag_aggregated_medal_score,
#   data = train_ind
# )
#
# train_dummified <- data.frame(predict(dummies, newdata = train_ind))
# train_dummified$Medal <- train_ind$Medal
#
# test_dummified <- data.frame(predict(dummies, newdata = test_ind))
# test_dummified$Medal <- test_ind$Medal
#
# ############################### Linear Discriminant Analysis (LDA) ################################
#
# lda_fit <- lda(Medal ~ ., data = train_dummified)
#
```

```
# test_dummified <- test_dummified %>%
#   mutate(lda_pred = predict(lda_fit, newdata = test_dummified)$class)
#
# lda_conf_matrix <- table(Predicted = test_dummified$lda_pred, Actual = test_dummified$Medal)
# lda_accuracy <- sum(diag(lda_conf_matrix)) / sum(lda_conf_matrix)
# print(paste("Accuracy of the LDA Model on Hold Out Set:", round(lda_accuracy, 3)))
#
# ############################### Multinomial Logistic Regression ################################
#
# lr_medal_fit <- multinom(
#   Medal ~ home_advantage * Season + home_advantage * Sex + Age +
#     Height + Weight + lag_games_played + lag_has_medal + lag_aggregated_medal_score,
#   data = train_ind
# )
#
# test_ind <- test_ind %>%
#   mutate(lr_medal_pred = predict(lr_medal_fit, newdata = test_ind))
#
# conf_matrix <- table(Predicted = test_ind$lr_medal_pred, Actual = test_ind$Medal)
# accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
# print(paste("Accuracy of the Multinomial Logistic Regression on Hold Out Set:", round(accuracy, 3)))
#
# ############################### Predicting Olympics Country Ranking ################################
#
# rf_ranking <- randomForest(
#   rank_bin ~ avg_age + avg_height + avg_weight + percent_medalists_lag +
#     events_participated + sports_participated + home_advantage + male_count +
#     female_count + lag_total_medals + lag_gold_count + lag_silver_count +
#     lag_bronze_count + lag_weighted_medals + avg_total_medals_last_5 +
#     avg_gold_count_last_5 + avg_silver_count_last_5 + avg_bronze_count_last_5 +
#     avg_weighted_medals_last_5 + avg_rank_last_5 + medals_per_events_lag +
#     medals_per_sport_lag + pct_above_mean_games,
#   data = train_cf,
#   importance = TRUE,
#   ntree = 500,
#   mtry = floor(sqrt(ncol(train_cf) - 1))
# )
#
# ############################### Random Forest Predictions and Evaluation ################################
#
# test_cf <- test_cf %>%
#   mutate(rf_rank_bin_pred = predict(rf_ranking, newdata = test_cf))
#
# conf_matrix <- table(Predicted = test_cf$rf_rank_bin_pred, Actual = test_cf$rank_bin)
# rf_rank_accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
# print(paste("Random Forest Accuracy on Hold Out Set:", round(rf_rank_accuracy, 3)))
#
# ############################### Model Summary and P-values ################################
#
# summary(lr_medal_fit)
# z_values <- summary(lr_medal_fit)$coefficients / summary(lr_medal_fit)$standard.errors
# p_values <- 2 * (1 - pnorm(abs(z_values)))
# print(round(p_values, 3))
```

```
#
# ############################## Predicted Probabilities ##############################
#
# train_ind <- train_ind %>% mutate(pred_class = predict(lr_medal_fit))
# class_probs <- as.data.frame(predict(lr_medal_fit, type = "probs"))
# colnames(class_probs) <- c("None_Prob", "Bronze_Prob", "Silver_Prob", "Gold_Prob")
#
# train_ind <- cbind(train_ind, class_probs)
#
# ############################## Visualization of Predictions by Home Advantage ##########################
#
# ggplot(train_ind) +
#   geom_boxplot(aes(
#     x = factor(home_advantage),
#     y = (Gold_Prob + Silver_Prob + Bronze_Prob),
#     fill = factor(home_advantage)
#   )) +
#   scale_fill_manual(values = c("0" = "#9DC7DD", "1" = "#9ED17B")) +
#   labs(
#     title = "Predicted Probability for Winning a Medal by Home Advantage",
#     x = "Home Advantage (0 = No, 1 = Yes)",
#     y = "Probability of Winning a Medal",
#     fill = "Home Advantage"
#   ) +
#   theme_minimal()
#
# ############################## Visualization by Season ##############################
#
# ggplot(train_ind) +
#   geom_boxplot(aes(
#     x = factor(Season),
#     y = (Gold_Prob + Silver_Prob + Bronze_Prob),
#     fill = factor(home_advantage)
#   )) +
#   scale_fill_manual(values = c("0" = "#9DC7DD", "1" = "#9ED17B")) +
#   labs(
#     title = "Predicted Probability for Winning a Medal by Season",
#     x = "Season",
#     y = "Probability of Winning a Medal",
#     fill = "Home Advantage"
#   ) +
#   theme_minimal()
#
# ############################## Visualization by Sex ##############################
#
# ggplot(train_ind) +
#   geom_violin(aes(
#     x = factor(Sex),
#     y = (Gold_Prob + Silver_Prob + Bronze_Prob),
#     fill = factor(home_advantage)
#   )) +
#   scale_fill_manual(values = c("0" = "#9DC7DD", "1" = "#9ED17B")) +
#   labs(
```

```
#     title = "Predicted Probability for Winning a Medal by Sex",
#     x = "Sex",
#     y = "Probability of Winning a Medal",
#     fill = "Home Advantage"
#   ) +
#   theme_minimal()
#
# ############################## Visualization for Figure Skating ##############################
#
# figure_skating_data <- train_ind %>%
#   filter(Sport == "Figure Skating", Event != "Figure Skating Men's Special Figures")
#
# ggplot(figure_skating_data) +
#   geom_boxplot(aes(
#     x = Event,
#     y = (Gold_Prob + Silver_Prob + Bronze_Prob),
#     fill = factor(home_advantage)
#   )) +
#   coord_flip() +
#   scale_fill_manual(values = c("0" = "#9DC7DD", "1" = "#9ED17B")) +
#   labs(
#     title = "Predicted Probability for Winning a Medal by Home Advantage in Figure Skating Events",
#     x = "Event",
#     y = "Probability of Winning Medal",
#     fill = "Home Advantage"
#   ) +
#   theme_minimal()
#
# ############################## Variable Importance ##############################
#
# var_importance <- rf_ranking$importance
# var_importance_df <- data.frame(
#   Variable = rownames(var_importance),
#   MeanDecreaseAccuracy = var_importance[, "MeanDecreaseAccuracy"],
#   MeanDecreaseGini = var_importance[, "MeanDecreaseGini"]
# )
#
# ggplot(var_importance_df, aes(x = reorder(Variable, MeanDecreaseAccuracy), y = MeanDecreaseAccuracy))
#   geom_bar(stat = "identity", fill = "#9DC7DD") +
#   coord_flip() +
#   theme_bw() +
#   labs(
#     title = "Variable Importance (Mean Decrease in Accuracy)",
#     x = "Variables",
#     y = "Mean Decrease in Accuracy"
#   )
#
# ## Confusion Matrix Visualization
# # conf_matrix_df <- as.data.frame(as.table(conf_matrix))
# # colnames(conf_matrix_df) <- c("Predicted", "Actual", "Freq")
# #
# # ggplot(conf_matrix_df, aes(x = Predicted, y = Actual, fill = Freq)) +
# #   geom_tile(color = "white") +
```

```
# #   scale_fill_gradient(low = "white", high = "blue") +
# #   geom_text(aes(label = Freq), color = "black", size = 5) +
# #   theme_bw() +
# #   labs(
# #     title = "Confusion Matrix",
# #     x = "Predicted",
# #     y = "Actual"
# #   )
#
# ############################### Probability of Being in Top 15 Teams ###############################
#
# train_cf <- train_cf %>%
#   mutate(
#     rf_top15_prob = predict(rf_ranking, type = "prob")[, 1] +
#       predict(rf_ranking, type = "prob")[, 2] +
#       predict(rf_ranking, type = "prob")[, 3]
#   )
#
# ggplot(train_cf) +
#   geom_boxplot(aes(
#     x = factor(home_advantage),
#     y = rf_top15_prob,
#     fill = factor(home_advantage)
#   )) +
#   scale_fill_manual(values = c("0" = "#9DC7DD", "1" = "#9ED17B")) +
#   labs(
#     title = "Impact of Home Advantage on Predicted Probability of Being in Top 15 Teams",
#     x = "Home Advantage",
#     y = "Predicted Probability (Top 15 Teams)",
#     fill = "Home Advantage"
#   ) +
#   theme_minimal() +
#   theme(
#     plot.title = element_text(hjust = 0.5),
#     legend.position = "none"
#   )
#
# ############################### Host Countries and Top 15 Frequency ###############################
#
# train_cf_hosts <- train_cf %>%
#   filter(NOC %in% noc_mapping$NOC_host) %>%
#   mutate(Top_15 = ifelse(rf_top15_prob >= 0.5, 1, 0)) %>%
#   group_by(NOC) %>%
#   summarize(
#     Total = n(),
#     Top_15_Count = sum(Top_15),
#     Top_15_Percentage = round((Top_15_Count / Total) * 100, 2),
#     .groups = "drop"
#   ) %>%
#   arrange(desc(Top_15_Percentage))
#
# train_cf_hosts
```