An Exploration and Analysis of Biological and Demographic Information of Summer Olympic Athletes From 1960 to 2016

Katie Manzione

May 2021

Abstract

This study investigated the commonalities and demographic backgrounds between Summer Olympic athletes who attended the Olympic Games and who received a medal, grouped by their specific sports, using a variety of exploratory data analysis methods and logistic regression models. We also attempted to classify the instance of an athlete receiving a medal at the Games using both Logistic Regression with a Bayes Classifier and Support Vector Machines (SVM) with and without an unbalanced data correction. The models were evaluated and compared based on the following metrics: overall error rate, precision, sensitivity, and specificity. We used the 120 years of Olympic History: Athletes and Results data set which was scraped from existing data from sports-reference.com by Randi Griffin, of Northeastern University, in May 2018. We considered Olympic athletes from only the Summer Olympic Games in 1960 and after. We further grouped the athletes by individual sports to account for the confounding of overall team skill narrowed to cycling, diving, fencing, gymnastics, shooting, swimming, weightlifting, and wrestling. The main objective of this study was to determine if there is a most common type of athlete, based on given physical characteristics by sport, that appears at the Games or that tends to medal given certain predictor variables through running various Logistic Regression Models for each sport. We also attempted to determine if there are particular demographic or background predictors that may impact the frequency of those who attend the Games through conducting multiple Two-Sample Tests for Equality of Proportions.

1 Introduction

The modern Olympic Games has historically been the largest, most prestigious athletic competition in the world, first taking place in Athens in 1896 [1, 2]. In 1896, the modern Games first consisted of 280 athletes from only 13 nations competing across 43 different events [1]. Over time, the Games has significantly grown, as it had more recently featured 11238 athletes from 207 countries across 306 different events in 2016 [2]. These athletes are considered to be the best in the world for their respective events. Thus, the question arises of if there is a trend across the athletes that appear at the Olympic Games and then subsequently medal in any characteristics they may exhibit.

This study was conducted to determine if certain biological and/or demographic attributes of athletes contribute to or have an effect on the athlete winning a medal. The basis of this research question is to answer whether or not there is a most common "type" of athlete, by event, that tends to medal at the Summer Olympic Games. Additionally, this study aimed to classify the instance of an athlete receiving a medal at the Games through the use of a Logistic Regression with a Bayes Classifier and different Support Vector Machines (SVM), with and without an unbalanced data correction

2 Data

2.1 Subjects

Athletes from only the Summer Olympic Games from 1960 to 2016 were included in the data set, which totaled to be 166706 athletes. Before 1960, the Games were primarily male-dominated, so sub-setting the data to Olympic Games 1960 and after allowed us to focus on a population that was more equal in gender proportions. Additionally, all entries with any missing entries were omitted from the data. We then further narrowed the data set observations to consider only individualized sports, sports where the athletes primarily compete in their event independently. This included Cycling (n = 7775), Diving (n = 2011), Fencing (n = 6537), Gymnastics (n = 18271), Shooting (n = 7260), Swimming (n = 18776), Weightlifting (n = 2974), and Wrestling (n = 5186). The purpose of including only these individual sports was to account for possible confounding of overall team skill, whereas a less skilled, outlying individual, in comparison to the rest of their team, could still win a medal due to their team winning a medal.

Note that each data observation is one event or instance of an athlete competing because there are athletes that appear at the games and compete in multiple events. The information contained within each observation includes, the sport, specified gender, age, height, weight, the country the athlete is represented, and the medal type won (either none, bronze, silver, gold).

2.2 Exploratory Data Analysis

After initial exploration of the data set, we found that there were many empty or missing data entries prior to approximately 1960. This could potentially be attributed to a lack of advanced data collection technologies and procedures in the late 1800s and early 1900s. Additionally, after exploration of the gender proportions, it was evident that there were vastly different proportions of males and females participating in the games prior to approximately 1960 as well. Thus all subsequent analyses were conducted with a narrowed data set, only including data from the years 1960 to 2016 and with all observations with missing entries omitted.

2.3 Data Manipulation

We re-coded our response variable of "Medal" to an indicator variable of medal won (0 = No, 1 = Yes). This was done so that we would not have to distinguish between the type of medal won, since our response variable of interest was overall only if the athlete had won a medal of any type or not.

As a result of our interest in the potential effects of an athlete coming from a developed country versus a non-developed country, we created a new variable, "Developed", in the data set as an indicator of whether the athlete is representing a developed country at the Games (0 = No, 1 = Yes). Countries that are considered to be developed are deemed, by the United Nations, to have high growth and high security economies [3].

3 Methods

These data were adapted from the online data set "120 years of Olympic History: Athletes and Results" [4]. These data were initially scraped from existing data from sports-reference.com by Randi Griffin [4], of Northeastern University, in May 2018. Data import, manipulation, tidying, and visualization were performed with the use of R package tidyverse [5]. P-values of less than 0.05 were deemed as statistically significant.

3.1 Analysis of Proportions

The differences of proportions between the following groups were analyzed: 1. differences in proportions between athletes from not developed or developed countries, and 2. differences in proportions between athletes who won a medal from not developed or developed countries. We investigated these proportions because since developed countries tend to have more resources due to their more developed economies, we hypothesized that there would be higher proportions of athletes representing developed countries as opposed to non-developed. These proportions were evaluated through the use of a Two-Sample Test for Equality of Proportions with Continuity Correction for each of the specified sports.

3.2 Logistic Regression Analyses

Logistic regression models were built and analyzed to determine the predictor variable(s) that have significant effects on an athlete winning a medal. Eight different models were built, one for each sport, and each contained the factors of "Age", "Height", "Weight", and "Developed", with the response of "Medal". The models were built using a generalized linear regression model with a binomial family.

3.3 Classification Methods

The instance of an athlete receiving a medal at the Games in each of the eight specified sports in this analysis was modeled through four different classification models. These included 1. Logistic Regression with a Bayes Classifier, 2. Support Vector Machine (SVM) with a linear kernel, 3. SVM with a polynomial kernel, and 4. SVM with a radial kernel. The models were evaluated and compared based on the following metrics: overall error rate, precision, sensitivity, and specificity.

These classification models were initially built on the raw data. However, since the models did not predict any instance of an athlete winning a medal, because the proportion of athletes winning a medal is so small compared to all participants, we decided to continue the analysis with an unbalanced data correction. In this study, we utilized the Generation of Synthetic Data by Randomly Over Sampling Examples (ROSE) approach to aids in our attempt of binary classification in the presence of these rare classes of winning a medal [6]. For these classification methods, the data formula used to generate the synthetic data included our familiar "Medal" variable as our response, and only the "Height" and "Weight" variables as our factors, and the probability of the minority class examples in the resulting data set to be the default of $p = 0.5$. The classification models in this analysis were then subsequently built on the synthetic data.

4 Results

4.1 Analysis of Proportions

The analysis of proportions between athletes from not developed or developed countries found that all eight sports had statistically significant differences between the proportions of athletes representing developed countries and those representing non-developed countries (all $p << 0.05$). However, weightlifting and wrestling contradict our initial hypothesis, having significantly larger proportions of athletes representing non-developed countries, whereas the remaining six sports support our initial hypothesis having significantly larger proportions of athletes representing developed countries.

The analysis of proportions between athletes who won a medal from not developed or developed countries found that cycling, diving, fencing, gymnastics,

swimming, weightlifting, and wrestling had statistically significant differences between the proportions of athletes who won medals representing developed countries and those representing non-developed countries ($p < 0.05$). Shooting was the only sport that did not have a statistically significant difference in the proportions. Among the significant differences, cycling, fencing, and swimming all had significantly larger proportions of athletes representing developed countries, whereas the remaining four sports had significantly larger proportions of athletes representing non-developed. Building off our initial hypothesis, cycling, fencing, and swimming are all costly sports, thus developed countries may have more financial resources to spend on their athletes to help them excel in more expensive sports in comparison to non-developed countries.

Proportions results of the Two-Sample Test for Equality of Proportions with Continuity Correction for each of the specified sports for 1. differences in proportions between athletes from not developed or developed countries, and 2. differences in proportions between athletes who won a medal from not developed or developed countries are shown in Tables 1 and 2, respectively.

Table 1: 2-sample test for equality of proportions with continuity correction for proportions of athletes who are from not developed or developed countries

Sport	Proportions		p
	Not Developed	Developed	
Cycling	0.355	0.645	< 0.0001***
Diving	0.452	0.548	< 0.0001***
Fencing	0.375	0.625	< 0.0001***
Gymnastics	0.359	0.641	< 0.0001***
Shooting	0.470	0.530	< 0.0001***
Swimming	0.394	0.606	< 0.0001***
Weightlifting	0.574	0.426	< 0.0001***
Wrestling	0.543	0.457	< 0.0001***

**indicates a significant value at $\alpha = 0.01$
***indicates a significant value at $\alpha = 0.001$

Table 2: 2-sample test for equality of proportions with continuity correction for proportions of athletes who recieved a medal from not developed or developed countries

Sport	Proportions		p
	Not Developed	Developed	
Cycling	0.219	0.781	< 0.0001***
Diving	0.552	0.448	< 0.0157*
Fencing	0.333	0.667	< 0.0001***
Gymnastics	0.540	0.459	0.0004**
Shooting	0.475	0.525	0.1264
Swimming	0.232	0.768	< 0.0001***
Weightlifting	0.643	0.357	< 0.0001***
Wrestling	0.557	0.443	< 0.0001***

*indicates a significant value at $\alpha = 0.05$
**indicates a significant value at $\alpha = 0.01$
***indicates a significant value at $\alpha = 0.001$

4.2 Logistic Regression Analysis

Overall, the logistic regression models show there is no single variable across all the analyzed sports that have a significant effect on an athlete winning a medal. We suspect this is due to significant variables, in terms of their effect on an athlete winning a medal, depending on the nature of the given sport.

The odds ratios of the logistic regression model coefficients for each of the eight sports are shown in Table 3. Statistically significant coefficients are noted. Note that all odds ratios greater than one represent a greater odds of association between the predictor and instance of winning a medal, whereas all odds ratios less than one represent a lower odds of association between the predictor and instance of winning a medal

Table 3: Odds ratio coefficients from the logistic regression models for each sport

Event	Intercept	Age	Height	Weight	Developed
Cycling	0.018***	1.011	0.989	1.044***	1.956***
Diving	0.001***	1.034*	1.052**	0.938***	0.650**
Fencing	0.023***	1.008	1.011	0.999	1.212*
Gymnastics	0.752	1.043***	0.990	0.977**	0.467***
Shooting	0.341	0.965***	0.994	1.010*	0.972
Swimming	0.000***	1.002***	1.028***	1.002	2.234***
Weightlifting	84.504***	0.975*	0.962***	1.016***	0.738**
Wrestling	0.2739	1.014	0.995	1.005	0.924**

*indicates a significant value at $\alpha = 0.05$
**indicates a significant value at $\alpha = 0.01$
***indicates a significant value at $\alpha = 0.001$

4.3 Classification Methods

The four metrics used to determine the effectiveness of each classification model are overall error rate, precision, sensitivity, and specificity. The overall error rate was determined by dividing the total incorrect predictions made by the model by the total predictions made. Precision was determined by dividing the true positives by the sum of the true positives and false positives. Sensitivity is the ability of the model to correctly identify athletes who did medal, i.e. the true positive rate. Specificity is the ability of the model to correctly identify athletes who did not medal, i.e. the true negative rate.

All the classification models, by sport, had similar precision rates. Of the four metrics, our main metric of interest was the error rate, where the goal is for the chosen model to have the smallest rate of error. Across all models for each of the sports, the polynomial SVM had the smallest error rate. The only exception is for shooting, but even then the error rates between the linear and polynomial SVMs for shooting are very close. Within all the models, we see that there is a trade-off between sensitivity and specificity, where when we see high sensitivity, we see low specificity. While considering all the metrics and placing our main interest in the error rate, we determined that overall for each sport, the polynomial SVM ranked highest in its ability to classify the instance of an athlete receiving a medal at the Games.

The results and values of the four metrics used to evaluate the classification models for each sport are shown in Table 4.

Table 4: Table of error, precision, sensitivity, and specificity rates for each of the classification methods

Event	Method	Error	Precision	Sensitivity	Specificity
Cycling	Bayes	0.362	0.917	0.655	0.489
	Linear SVM	0.457	0.906	0.548	0.507
	*Polynomial SVM	0.131	0.896	0.966	0.031
	Radial SVM	0.347	0.913	0.678	0.444
Diving	Bayes	0.344	0.912	0.683	0.421
	Linear SVM	0.396	0.932	0.603	0.614
	*Polynomial SVM	0.131	0.910	0.948	0.175
	Radial SVM	0.319	0.905	0.721	0.333
Fencing	Bayes	0.379	0.799	0.696	0.340
	Linear SVM	0.365	0.799	0.719	0.317
	*Polynomial SVM	0.225	0.792	0.970	0.037
	Radial SVM	0.379	0.794	0.703	0.314
Gymnastics	Bayes	0.441	0.945	0.560	0.546
	Linear SVM	0.397	0.940	0.613	0.461
	*Polynomial SVM	0.104	0.931	0.959	0.022
	Radial SVM	0.473	0.956	0.523	0.587
Shooting	Bayes	0.167	0.921	0.896	0.078
	*Linear SVM	0.077	0.923	1	0
	*Polynomial SVM	0.111	0.922	0.961	0.0260
	Radial SVM	0.449	0.924	0.559	0.448
Swimming	Bayes	0.445	0.876	0.561	0.514
	Linear SVM	0.432	0.877	0.579	0.499
	*Polynomial SVM	0.180	0.860	0.945	0.053
	Radial SVM	0.369	0.874	0.666	0.410
Weightlifting	Bayes	0.343	0.810	0.740	0.335
	Linear SVM	0.322	0.801	0.790	0.246
	*Polynomial SVM	0.248	0.800	0.916	0.120
	Radial SVM	0.291	0.805	0.835	0.222
Wrestling	Bayes	0.303	0.806	0.821	0.193
	*Linear SVM	0.257	0.805	0.898	0.109
	*Polynomial SVM	0.252	0.803	0.911	0.084
	Radial SVM	0.282	0.802	0.862	0.131

*indicates chosen model

5 Discussion

From this study, we saw that there are significant differences in proportions between athletes from developed and athletes from non-developed countries participating in the Olympic Games. This was the same for all athletes who won a medal, with the exception of shooting. We subsequently determined that there was no single variable across all the analyzed sports that had a significant effect on an athlete winning a medal because significant variables by sports depend on the nature of the sport itself. Lastly, we were able to determine that a polynomial support vector machine was the most effective model in being able to classify an athlete winning a medal.

Limitations of this study include that since we were limited to only complete data observations, we were limited to looking at only a smaller time range of data. This study may be extended to an analysis of individual sports within the Winter Olympic Games.

References

[1] History.com Editors. *The Olympic Games*.

[2] International Olympic Committee. *Rio 2016*.

[3] United Nations. *Country Classification: Data sources, country classifications and aggregation methodology*, 2014.

[4] Rgriffin. *120 Years of Olympic History: Athletes and Results*.

[5] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019.

[6] Nicola Lunardon, Giovanna Menardi, and Nicola Torelli. ROSE: a Package for Binary Imbalanced Learning. *R Journal*, 6(1):82–92, 2014.

Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(readr)
library(dplyr)
library(knitr)
library(tidyverse)
library(ggplot2)
library(ggpubr)
setwd("~/Desktop")
olympic <- read_csv("olympicdata.csv")
olympic<- olympic[,c("Sex", "Age", "Height", "Weight", "Team", "Games",
"Year", "Sport", "Medal")]
olympic$Games <- gsub("\\d", "", olympic$Games)
olympic$Games <- trimws(olympic$Games)
olympic <- subset(olympic, Games == "Summer")
olympic$Medal[is.na(olympic$Medal)] <- 0
olympic <- olympic %>%
  mutate(Medal = recode(Medal, "Gold" = "1", "Silver" = "1", "Bronze" = "1"))
olympic$Medal <- as.numeric(olympic$Medal)
olympic <- na.omit(olympic)
olympic$Medal <- as.factor(olympic$Medal)
#olympic$Year <- as.factor(olympic$Year)

olympic <- olympic %>%
  mutate(Developed = Team) %>%
  mutate(Developed = recode(Developed, "Canada" = "1", "France"= "1",
"Germany"= "1", "Italy"= "1", "Japan"= "1", "Great Britain"= "1", "United
States"= "1", "Iceland"= "1", "Norway"= "1", "Switzerland"= "1", "Austria"=
"1", "Belgium"= "1", "Denmark"= "1", "Finland"= "1", "Greece"= "1",
"Ireland"= "1", "Luxembourg"= "1", "Netherlands"= "1", "Portugal"= "1",
"Spain"= "1", "Sweden"= "1", "Australia" = "1", "Japan" = "1", "New Zealand"
= "1", "Bulgaria" = "1", "Croatia"= "1", "Cyprus" = "1", "Estonia" = "1",
"Czech Republic" = "1", "Hungary" = "1", "Latvia" = "1", "Lithuania" = "1",
"Malta"  = "1", "Poland" = "1", "Romania" = "1", "Slovakia" = "1", "Slovenia"
= "1"))
olympic$Developed[olympic$Developed != "1"] <- 0
head(olympic)
nrow(olympic)
olympic$Sport <- as.factor(olympic$Sport)
sportList <- c("Cycling", "Diving", "Fencing", "Gymnastics", "Shooting",
"Swimming", "Weightlifting", "Wrestling")


cycling <- subset(olympic, Sport == sportList[1])
cycling <- cycling[which(cycling$Year > 1960),]

diving <- subset(olympic, Sport == sportList[2])
diving <- diving[which(diving$Year > 1960),]
```

```
fencing <- subset(olympic, Sport == sportList[3])
fencing <- fencing[which(fencing$Year > 1960),]

gymnastics <- subset(olympic, Sport == sportList[4])
gymnastics <- gymnastics[which(gymnastics$Year > 1960),]

shooting <- subset(olympic, Sport == sportList[5])
shooting <- shooting[which(shooting$Year > 1960),]

swimming <- subset(olympic, Sport == sportList[6])
swimming <- swimming[which(swimming$Year > 1960),]

weightlifting <- subset(olympic, Sport == sportList[7])
weightlifting <- weightlifting[which(weightlifting$Year > 1960),]

wrestling <- subset(olympic, Sport == sportList[8])
wrestling <- wrestling[which(wrestling$Year > 1960),]


my_olympic <- rbind(cycling, diving, fencing, gymnastics, shooting, swimming,
weightlifting, wrestling)
olympic_developed1 <- ggplot(my_olympic) +
  geom_bar(aes(x=Year, fill=as.factor(Developed))) +
  ggtitle("Plot of Country Classification by Time")
olympic_developed1

olympic_developed2 <- ggplot(my_olympic) +
  geom_bar(aes(x=Medal, fill=as.factor(Developed)))+
  ggtitle("Barplot of Medal by Developed")
olympic_developed2
cycling_gender <- ggplot(cycling)+
  geom_bar(aes(x=Year, fill=as.factor(Sex))) +
  ggtitle("Cycling by Gender")

diving_gender <- ggplot(diving)+
  geom_bar(aes(x=Year, fill=as.factor(Sex))) +
  ggtitle("Diving by Gender")

fencing_gender <- ggplot(fencing)+
  geom_bar(aes(x=Year, fill=as.factor(Sex))) +
  ggtitle("Fencing by Gender")

gymnastics_gender <- ggplot(gymnastics)+
  geom_bar(aes(x=Year, fill=as.factor(Sex))) +
  ggtitle("Gymnastics by Gender")

shooting_gender <- ggplot(shooting)+
  geom_bar(aes(x=Year, fill=as.factor(Sex))) +
```

```
  ggtitle("Shooting by Gender")

swimming_gender <- ggplot(swimming)+
  geom_bar(aes(x=Year, fill=as.factor(Sex))) +
  ggtitle("Swimming by Gender")

weightlifting_gender <- ggplot(weightlifting)+
  geom_bar(aes(x=Year, fill=as.factor(Sex))) +
  ggtitle("Weightlifting by Gender")

wrestling_gender <- ggplot(wrestling)+
  geom_bar(aes(x=Year, fill=as.factor(Sex))) +
  ggtitle("Wrestling by Gender")
ggarrange(cycling_gender, diving_gender, fencing_gender, gymnastics_gender,
shooting_gender, swimming_gender, weightlifting_gender, wrestling_gender,
ncol=2, nrow=2)
cycling_medal <- ggplot(cycling)+
  geom_bar(aes(x=Medal, fill=as.factor(Developed))) +
  ggtitle("Cycling Medals")

diving_medal <- ggplot(diving)+
  geom_bar(aes(x=Medal, fill=as.factor(Developed))) +
  ggtitle("Diving Medals")

fencing_medal <- ggplot(fencing)+
  geom_bar(aes(x=Medal, fill=as.factor(Developed))) +
  ggtitle("Fencing Medals")

gymnastics_medal <- ggplot(gymnastics)+
  geom_bar(aes(x=Medal, fill=as.factor(Developed))) +
  ggtitle("Gymnastics Medals")

shooting_medal <- ggplot(shooting)+
  geom_bar(aes(x=Medal, fill=as.factor(Developed))) +
  ggtitle("Shooting Medals")

swimming_medal <- ggplot(swimming)+
  geom_bar(aes(x=Medal, fill=as.factor(Developed))) +
  ggtitle("Swimming Medals")

weightlifting_medal <- ggplot(weightlifting)+
  geom_bar(aes(x=Medal, fill=as.factor(Developed))) +
  ggtitle("Weightlifting Medals")

wrestling_medal <- ggplot(wrestling)+
  geom_bar(aes(x=Medal, fill=as.factor(Developed))) +
  ggtitle("Wrestling Medals")
ggarrange(cycling_medal, diving_medal, fencing_medal, gymnastics_medal,
shooting_medal, swimming_medal, weightlifting_medal, wrestling_medal, ncol=2,
```

```
nrow=2)
cycling_height <- ggplot(cycling)+
  geom_density(aes(x=Height, col=as.factor(Medal))) +
  ggtitle("Cycling Height")

diving_height <- ggplot(diving)+
  geom_density(aes(x=Height, col=as.factor(Medal))) +
  ggtitle("Diving Height")

fencing_height <- ggplot(fencing)+
  geom_density(aes(x=Height, col=as.factor(Medal))) +
  ggtitle("Fencing Height")

gymnastics_height <- ggplot(gymnastics)+
  geom_density(aes(x=Height, col=as.factor(Medal))) +
  ggtitle("Gymnastics Height")

shooting_height <- ggplot(shooting)+
  geom_density(aes(x=Height, col=as.factor(Medal))) +
  ggtitle("Shooting Height")

swimming_height <- ggplot(swimming)+
  geom_density(aes(x=Height, col=as.factor(Medal))) +
  ggtitle("Swimming Height")

weightlifting_height <- ggplot(weightlifting)+
  geom_density(aes(x=Height, col=as.factor(Medal))) +
  ggtitle("Weightlifting Height")

wrestling_height <- ggplot(wrestling)+
  geom_density(aes(x=Height, col=as.factor(Medal))) +
  ggtitle("Wrestling Height")
ggarrange(cycling_height, diving_height, fencing_height, gymnastics_height,
shooting_height, swimming_height, weightlifting_height, wrestling_height,
ncol=2, nrow=2)
cycling_height_time <- ggplot(cycling)+
  geom_point(aes(x=Year, y = Height, col=as.factor(Medal))) +
  ggtitle("Cycling Height Over Time")

diving_height_time <- ggplot(diving)+
  geom_point(aes(x=Year,y=Height, col=as.factor(Medal))) +
  ggtitle("Diving Height Over Time")

fencing_height_time <- ggplot(fencing)+
  geom_point(aes(x=Year,y=Height, col=as.factor(Medal))) +
  ggtitle("Fencing Height Over Time")

gymnastics_height_time <- ggplot(gymnastics)+
  geom_point(aes(x=Year,y=Height, col=as.factor(Medal))) +
```

```
  ggtitle("Gymnastics Height Over Time")

shooting_height_time <- ggplot(shooting)+
  geom_point(aes(x=Year,y=Height, col=as.factor(Medal))) +
  ggtitle("Shooting Height Over Time")

swimming_height_time <- ggplot(swimming)+
  geom_point(aes(x=Year,y=Height, col=as.factor(Medal))) +
  ggtitle("Swimming Height Over Time")

weightlifting_height_time <- ggplot(weightlifting)+
  geom_point(aes(x=Year,y=Height, col=as.factor(Medal))) +
  ggtitle("Weightlifting Height Over Time")

wrestling_height_time <- ggplot(wrestling)+
  geom_point(aes(x=Year,y=Height, col=as.factor(Medal))) +
  ggtitle("Wrestling Height Over Time")
ggarrange(cycling_height_time, diving_height_time, fencing_height_time,
gymnastics_height_time, shooting_height_time, swimming_height_time,
weightlifting_height_time, wrestling_height_time, ncol=2, nrow=2)
cycling_weight <- ggplot(cycling)+
  geom_density(aes(x=Weight, col=as.factor(Medal))) +
  ggtitle("Cycling Weight")

diving_weight <- ggplot(diving)+
  geom_density(aes(x=Weight, col=as.factor(Medal))) +
  ggtitle("Diving Weight")

fencing_weight <- ggplot(fencing)+
  geom_density(aes(x=Weight, col=as.factor(Medal))) +
  ggtitle("Fencing Weight")

gymnastics_weight <- ggplot(gymnastics)+
  geom_density(aes(x=Weight, col=as.factor(Medal))) +
  ggtitle("Gymnastics Weight")

shooting_weight <- ggplot(shooting)+
  geom_density(aes(x=Weight, col=as.factor(Medal))) +
  ggtitle("Shooting Weight")

swimming_weight <- ggplot(swimming)+
  geom_density(aes(x=Weight, col=as.factor(Medal))) +
  ggtitle("Swimming Weight")

weightlifting_weight <- ggplot(weightlifting)+
  geom_density(aes(x=Weight, col=as.factor(Medal))) +
  ggtitle("Weightlifting Weight")

wrestling_weight <- ggplot(wrestling)+
```

```
  geom_density(aes(x=Weight, col=as.factor(Medal))) +
  ggtitle("Wrestling Weight")
ggarrange(cycling_weight, diving_weight, fencing_weight, gymnastics_weight,
shooting_weight, swimming_weight, weightlifting_weight, wrestling_weight,
ncol=2, nrow=2)
cycling_weight_time <- ggplot(cycling)+
  geom_point(aes(x=Year, y = Weight, col=as.factor(Medal))) +
  ggtitle("Cycling Weight Over Time")

diving_weight_time <- ggplot(diving)+
  geom_point(aes(x=Year,y=Weight, col=as.factor(Medal))) +
  ggtitle("Diving Weight Over Time")

fencing_weight_time <- ggplot(fencing)+
  geom_point(aes(x=Year,y=Weight, col=as.factor(Medal))) +
  ggtitle("Fencing Weight Over Time")

gymnastics_weight_time <- ggplot(gymnastics)+
  geom_point(aes(x=Year,y=Weight, col=as.factor(Medal))) +
  ggtitle("Gymnastics Weight Over Time")

shooting_weight_time <- ggplot(shooting)+
  geom_point(aes(x=Year,y=Weight, col=as.factor(Medal))) +
  ggtitle("Shooting Weight Over Time")

swimming_weight_time <- ggplot(swimming)+
  geom_point(aes(x=Year,y=Weight, col=as.factor(Medal))) +
  ggtitle("Swimming Weight Over Time")

weightlifting_weight_time <- ggplot(weightlifting)+
  geom_point(aes(x=Year,y=Weight, col=as.factor(Medal))) +
  ggtitle("Weightlifting Weight Over Time")

wrestling_weight_time <- ggplot(wrestling)+
  geom_point(aes(x=Year,y=Weight, col=as.factor(Medal))) +
  ggtitle("Wrestling Weight Over Time")
ggarrange(cycling_weight_time, diving_weight_time, fencing_weight_time,
gymnastics_weight_time, shooting_weight_time, swimming_weight_time,
weightlifting_weight_time, wrestling_weight_time, ncol=2, nrow=2)
cycling_glm <- glm(Medal ~ Age + Height + Weight + Developed, data=cycling,
family = "binomial")
summary(cycling_glm)
diving_glm <- glm(Medal ~ Age + Height + Weight + Developed, data=diving,
family = "binomial")
summary(diving_glm)
fencing_glm <- glm(Medal ~ Age + Height + Weight + Developed, data=fencing,
family = "binomial")
summary(fencing_glm)
gymnastics_glm <- glm(Medal ~ Age + Height + Weight + Developed,
```

```
data=gymnastics, family = "binomial")
summary(gymnastics_glm)
shooting_glm <- glm(Medal ~ Age + Height + Weight + Developed, data=shooting,
family = "binomial")
summary(shooting_glm)
swimming_glm <- glm(Medal ~ Age + Height + Weight + Developed, data=swimming,
family = "binomial")
summary(swimming_glm)
weightlifting_glm <- glm(Medal ~ Age + Height + Weight + Developed,
data=weightlifting, family = "binomial")
summary(weightlifting_glm)
wrestling_glm <- glm(Medal ~ Age + Height + Weight + Developed,
data=wrestling, family = "binomial")
summary(wrestling_glm)
round(exp(coef(cycling_glm)),3)
round(exp(coef(diving_glm)),3)
round(exp(coef(fencing_glm)),3)
round(exp(coef(gymnastics_glm)),3)
round(exp(coef(shooting_glm)),3)
round(exp(coef(swimming_glm)),3)
round(exp(coef(weightlifting_glm)),3)
round(exp(coef(wrestling_glm)),3)
cycling_prop <- prop.test(x = c(length(which(cycling$Developed ==
0)),length(which(cycling$Developed == 1))), n =
c(length(cycling$Developed),length(cycling$Developed)), alternative =
"two.sided")
cycling_prop
diving_prop <- prop.test(x = c(length(which(diving$Developed ==
0)),length(which(diving$Developed == 1))), n =
c(length(diving$Developed),length(diving$Developed)), alternative =
"two.sided")
diving_prop
fencing_prop <- prop.test(x = c(length(which(fencing$Developed ==
0)),length(which(fencing$Developed == 1))), n =
c(length(fencing$Developed),length(fencing$Developed)), alternative =
"two.sided")
fencing_prop
gymnastics_prop <- prop.test(x = c(length(which(gymnastics$Developed ==
0)),length(which(gymnastics$Developed == 1))), n =
c(length(gymnastics$Developed),length(gymnastics$Developed)), alternative =
"two.sided")
gymnastics_prop
shooting_prop <- prop.test(x = c(length(which(shooting$Developed ==
0)),length(which(shooting$Developed == 1))), n =
c(length(shooting$Developed),length(shooting$Developed)), alternative =
"two.sided")
shooting_prop
swimming_prop <- prop.test(x = c(length(which(swimming$Developed ==
0)),length(which(swimming$Developed == 1))), n =
c(length(swimming$Developed),length(swimming$Developed)), alternative =
```

```
"two.sided")
swimming_prop
weightlifting_prop <- prop.test(x = c(length(which(weightlifting$Developed ==
0)),length(which(weightlifting$Developed == 1))), n =
c(length(weightlifting$Developed),length(weightlifting$Developed)),
alternative = "two.sided")
weightlifting_prop
wrestling_prop <- prop.test(x = c(length(which(wrestling$Developed ==
0)),length(which(wrestling$Developed == 1))), n =
c(length(wrestling$Developed),length(wrestling$Developed)), alternative =
"two.sided")
wrestling_prop
cyclingMedal <- subset(cycling, Medal == 1)
divingMedal <- subset(diving, Medal == 1)
fencingMedal <- subset(fencing, Medal == 1)
gymnasticsMedal <- subset(gymnastics, Medal == 1)
shootingMedal <- subset(shooting, Medal == 1)
swimmingMedal <- subset(swimming, Medal == 1)
weightliftingMedal <- subset(weightlifting, Medal == 1)
wrestlingMedal <- subset(wrestling, Medal == 1)
cycling_propMedal <- prop.test(x = c(length(which(cyclingMedal$Developed ==
0)),length(which(cyclingMedal$Developed == 1))), n =
c(length(cyclingMedal$Developed),length(cyclingMedal$Developed)), alternative
= "two.sided")
cycling_propMedal
diving_propMedal <- prop.test(x = c(length(which(divingMedal$Developed ==
0)),length(which(divingMedal$Developed == 1))), n =
c(length(divingMedal$Developed),length(divingMedal$Developed)), alternative =
"two.sided")
diving_propMedal
fencing_propMedal <- prop.test(x = c(length(which(fencingMedal$Developed ==
0)),length(which(fencingMedal$Developed == 1))), n =
c(length(fencingMedal$Developed),length(fencingMedal$Developed)), alternative
= "two.sided")
fencing_propMedal
gymnastics_propMedal <- prop.test(x =
c(length(which(gymnasticsMedal$Developed ==
0)),length(which(gymnasticsMedal$Developed == 1))), n =
c(length(gymnasticsMedal$Developed),length(gymnasticsMedal$Developed)),
alternative = "two.sided")
gymnastics_propMedal
shooting_propMedal <- prop.test(x = c(length(which(shootingMedal$Developed ==
0)),length(which(shootingMedal$Developed == 1))), n =
c(length(shootingMedal$Developed),length(shootingMedal$Developed)),
alternative = "two.sided")
shooting_propMedal
swimming_propMedal <- prop.test(x = c(length(which(swimmingMedal$Developed ==
0)),length(which(swimmingMedal$Developed == 1))), n =
c(length(swimmingMedal$Developed),length(swimmingMedal$Developed)),
alternative = "two.sided")
```

```
swimming_propMedal
weightlifting_propMedal <- prop.test(x =
c(length(which(weightliftingMedal$Developed ==
0)),length(which(weightliftingMedal$Developed == 1))), n =
c(length(weightliftingMedal$Developed),length(weightliftingMedal$Developed)),
alternative = "two.sided")
weightlifting_propMedal
wrestling_propMedal <- prop.test(x = c(length(which(wrestlingMedal$Developed
== 0)),length(which(wrestlingMedal$Developed == 1))), n =
c(length(wrestlingMedal$Developed),length(wrestlingMedal$Developed)),
alternative = "two.sided")
wrestling_propMedal
library(tidyverse) # data manip
library(ISLR) # data
library(GGally) # pairs plots
library(e1071) #svm
nrow(diving)
0.7*nrow(diving)
0.3*nrow(diving)
diving_train <- cycling[1:1304,]
diving_test <- cycling[1305:1862,]
0.7*nrow(fencing)
nrow(fencing)
nrow(fencing) - 3867
fencing_train <- fencing[1:3867,]
fencing_test <- fencing[3868:5523,]
0.7*nrow(gymnastics)
0.3*nrow(gymnastics)
nrow(gymnastics)
gymnastics_train <- gymnastics[1:10981,]
gymnastics_test <- gymnastics[10982:15686,]
0.7*nrow(shooting)
0.3*nrow(shooting)
nrow(shooting)
shooting_train <- shooting[1:4665,]
shooting_test <- shooting[4666:6663,]
0.7*nrow(swimming)
0.3*nrow(swimming)

swimming_train <- swimming[1:12520,]
swimming_test <- swimming[12521:17885,]
0.7*nrow(weightlifting)
0.3*nrow(weightlifting)

weightlifting_train <- weightlifting[1:1892,]
weightlifting_test <- weightlifting[1893:2702,]
0.7*nrow(wrestling)
0.3*nrow(wrestling)

wrestling_train <- wrestling[1:3253,]
```

```r
wrestling_test <- wrestling[3254:4647,]
cplot <- ggplot(cycling) +
  geom_point(aes(x=Weight, y=Height, col = as.factor(Medal)))
cplot

0.7*nrow(cycling)
0.3*nrow(cycling)

cycling_train <- cycling[1:5026,]
cycling_test <- cycling[5027:7181,]

7181 - 5026

log.reg.model_cyc <- glm(Medal ~ Height +Weight, data = cycling_train, family
= "binomial")
summary(log.reg.model_cyc)

#get the estimated y value (0/1) for each point in the grid
yhat_cyc <- predict(log.reg.model_cyc, newdata = cycling_test, type =
"response")
yhat_cyc <- (yhat_cyc > 0.5)*1
cycling_test$yhat <- yhat_cyc
summary(yhat_cyc)

#confMat_log <- table(actual=cycling_test$Medal, predicted=cycling_test$yhat)
#(confMat_log[1,2] + confMat_log[2,1])/dim(cycling_test)[1]



#cyc_fig <- ggplot()+ geom_tile(aes(x = Weight, y = Height, fill =
as.factor(yhat)), data = cycling_test, alpha = .5) + labs(fill = "Predicted
Y") + geom_point(data = cycling, aes(x = Weight, y = Height, col =
as.factor(Medal))) + labs(col = "Observed Y")
#cyc_fig


svmOut <-svm(Medal ~ Height , data = cycling_train, kernel = "linear", degree
= 2)
summary(svmOut)
yhat_grid3 <- predict(svmOut, newdata = cycling_test, type = "response")
#yhat_grid3 <- (yhat_grid3 > 0.5)*1
cycling_test$yhat3 <- yhat_grid3
summary(yhat_grid3)

#confMat_svmLIN <- table(actual=cycling_test$Medal,
predicted=cycling_test$yhat)
#(confMat_svmLIN[1,2] + confMat_svmLIN[2,1])/dim(cycling_test)[1]
#(confMat_svmLIN[1,1])/(confMat_svmLIN[1,1]+confMat_svmLIN[2,1])
#(confMat_svmLIN[1,1])/(confMat_svmLIN[1,1]+confMat_svmLIN[1,2])
```

```r
#(confMat_svmLIN[2,2])/(confMat_svmLIN[2,2]+confMat_svmLIN[2,1])

#cyc_fig2 <- ggplot()+ geom_tile(aes(x = Weight, y = Height, fill =
as.factor(yhat3)), data = cycling_test, alpha = .5) + labs(fill = "Predicted
Y") + geom_point(data = cycling, aes(x = Weight, y = Height, col =
as.factor(Medal))) + labs(col = "Observed Y")
#cyc_fig2


#work around: over-sampling-> sample more of the unbalanced class
#resample to 1000...on avg. 50/50 split to train
#canvas-->unbalanced classification issue slide (diff weight to fit the
model...) -->adding weight to minor category


wt =ifelse(cycling_train$Medal==0, 1, 1000)
fit_logit_wt_cyc <-glm(Medal ~ Height + Weight, data =
cycling_train,family=binomial,weights = wt)
#Log-odd ratio
pred_logit_wt_cyc<-predict(fit_logit_wt_cyc,newdata=cycling_test)

# Classify a sample as 0 when its predicted log-odd ratio is negative
y_hat_wt=ifelse(pred_logit_wt_cyc>=0, 1, 1000)
summary(y_hat_wt)

mean(y_hat_wt!=cycling_test$Medal)

cycling_test$yhat_wt <- y_hat_wt


cyc_fig_wt <- ggplot()+ geom_tile(aes(x = Weight, y = Height, fill =
as.factor(yhat_wt)), data = cycling_test, alpha = .5) + labs(fill =
"Predicted Y") + geom_point(data = cycling, aes(x = Weight, y = Height, col =
as.factor(Medal))) + labs(col = "Observed Y")
cyc_fig_wt



svmOut_wt <-svm(Medal ~ Height , data = cycling_train, kernel = "linear",
degree = 2, weight = wt)
summary(svmOut_wt)
yhat_grid_svmwt <- predict(svmOut_wt, newdata = cycling_test)
summary(yhat_grid_svmwt)
y_hat_wt=ifelse(pred_logit_wt_cyc>=0, 1, 1000)

#yhat_grid3 <- (yhat_grid3 > 0.5)*1
cycling_test$yhat3 <- yhat_grid3
summary(yhat_grid3)
```

```r
cyc_fig2 <- ggplot()+ geom_tile(aes(x = Weight, y = Height, fill =
as.factor(yhat3)), data = cycling_test, alpha = .5) + labs(fill = "Predicted
Y") + geom_point(data = cycling, aes(x = Weight, y = Height, col =
as.factor(Medal))) + labs(col = "Observed Y")
cyc_fig2
library(rpart)
library(ROSE)
#imb <- rpart(Medal ~ Height + Weight, data = cycling_train)
#pred.treeimb <- predict(imb, newdata = cycling_test)
#accuracy.meas(cycling_test$Medal, pred.treeimb[,2])

#data_balanced_over <- ovun.sample(Medal ~ Height + Weight, data =
cycling_train, method = "over",N = nrow(cycling))$data
#table(data_balanced_over$Medal)

data.rose1 <- ROSE(Medal ~ Height + Weight , data = cycling_train, seed =
1)$data
table(data.rose1$Medal)

#data.rose2 <- ROSE(Medal ~ Height + Weight, data = cycling_test, seed =
1)$data
#table(data.rose2$Medal)

#cycling_rose <- rbind(data.rose1, data.rose2[,1:3])

#Precision = TP / (TP + FP) out of observations labeled as positive, how many
are actually labeled positive
#Sensitivity = TP / (TP + FN) how many observations of positive class are
labeled correctly.
#Specificity = TN / (TN + FP) how many observations of negative class are
labeled correctly.


# Bayes Classifier

log.reg.model_cyc.rose <- glm(Medal ~ Height +Weight, data = data.rose1,
family = "binomial")
#summary(log.reg.model_cyc.rose)

#get the estimated y value (0/1) for each point in the grid
yhat_cyc.logRose <- predict(log.reg.model_cyc.rose, newdata = cycling_test,
type = "response")
yhat_cyc.logRose <- (yhat_cyc.logRose > 0.5)*1
cycling_test$yhat.logRose <- as.factor(yhat_cyc.logRose)
summary(cycling_test$yhat.logRose)

confMat_logRose <- table(actual=cycling_test$Medal,
predicted=cycling_test$yhat.logRose)
```

```
(confMat_logRose[1,2] + confMat_logRose[2,1])/dim(cycling_test)[1]
(confMat_logRose[1,1])/(confMat_logRose[1,1]+confMat_logRose[2,1])
(confMat_logRose[1,1])/(confMat_logRose[1,1]+confMat_logRose[1,2])
(confMat_logRose[2,2])/(confMat_logRose[2,2]+confMat_logRose[2,1])
#error rate is 0.361949
#precision is 0.9173913
#sensitivity is 0.6552795
#specificity is 0.4887892



# Linear Rose SVM

svmOut_rose <-svm(Medal ~ Height +Weight, data = data.rose1, kernel =
"linear", degree = 2)
summary(svmOut_rose)
yhat_grid_rosesvm <- predict(svmOut_rose, newdata = cycling_test, type =
"response")
#yhat_grid3 <- (yhat_grid3 > 0.5)*1
cycling_test$yhat_rosesvm <- yhat_grid_rosesvm
summary(yhat_grid_rosesvm)


confMat_roseLINsvm<-table(actual=cycling_test$Medal,
predicted=cycling_test$yhat_rosesvm)
(confMat_roseLINsvm[1,2] + confMat_roseLINsvm[2,1])/dim(cycling_test)[1]
(confMat_roseLINsvm[1,1])/(confMat_roseLINsvm[1,1]+confMat_roseLINsvm[2,1])
(confMat_roseLINsvm[1,1])/(confMat_roseLINsvm[1,1]+confMat_roseLINsvm[1,2])
(confMat_roseLINsvm[2,2])/(confMat_roseLINsvm[2,2]+confMat_roseLINsvm[2,1])
#error rate is 0.4566125
#precision is 0.9058219
#sensitivity is 0.547619
#specificity is 0.5067265



# Polynomial Rose SVM

svmOut_rose2 <-svm(Medal ~ Height +Weight, data = data.rose1, kernel =
"polynomial", degree = 2)
summary(svmOut_rose2)
yhat_grid_rosesvm2 <- predict(svmOut_rose2, newdata = cycling_test, type =
"response")
#yhat_grid3 <- (yhat_grid3 > 0.5)*1
cycling_test$yhat_rosesvm2 <- yhat_grid_rosesvm2
summary(yhat_grid_rosesvm2)


confMat_rosePOLYsvm<-table(cycling_test$Medal, cycling_test$yhat_rosesvm2)
```

```
(confMat_rosePOLYsvm[1,2] + confMat_rosePOLYsvm[2,1])/dim(cycling_test)[1]
(confMat_rosePOLYsvm[1,1])/(confMat_rosePOLYsvm[1,1]+confMat_rosePOLYsvm[2,1]
)
(confMat_rosePOLYsvm[1,1])/(confMat_rosePOLYsvm[1,1]+confMat_rosePOLYsvm[1,2]
)
(confMat_rosePOLYsvm[2,2])/(confMat_rosePOLYsvm[2,2]+confMat_rosePOLYsvm[2,1]
)
#error rate is 0.1308585
#precision is 0.8962536
#sensitivity is 0.9658385
#specificity is 0.03139013



# Polynomial Rose SVM

svmOut_rose3 <-svm(Medal ~ Height +Weight, data = data.rose1, kernel =
"radial", degree = 2)
summary(svmOut_rose3)
yhat_grid_rosesvm3 <- predict(svmOut_rose3, newdata = cycling_test, type =
"response")
#yhat_grid3 <- (yhat_grid3 > 0.5)*1
cycling_test$yhat_rosesvm3 <- yhat_grid_rosesvm3
summary(yhat_grid_rosesvm3)


confMat_roseRADsvm<-table(cycling_test$Medal, cycling_test$yhat_rosesvm3)
(confMat_roseRADsvm[1,2] + confMat_roseRADsvm[2,1])/dim(cycling_test)[1]
(confMat_roseRADsvm[1,1])/(confMat_roseRADsvm[1,1]+confMat_roseRADsvm[2,1])
(confMat_roseRADsvm[1,1])/(confMat_roseRADsvm[1,1]+confMat_roseRADsvm[1,2])
(confMat_roseRADsvm[2,2])/(confMat_roseRADsvm[2,2]+confMat_roseRADsvm[2,1])
#error rate is 0.3466357
#precision is 0.9134682
#sensitivity is 0.6775362
#specificity is 0.4439462



#contingency table (confusion matrix) --> actual medal/predicted medal

#lmitation -> two groups are not linearly separted

#cyc_fig_rosesvm <- ggplot()+ geom_tile(aes(x = Weight, y = Height, fill =
as.factor(yhat_rosesvm)), data = cycling_test, alpha = .5) + labs(fill =
"Predicted Y") + geom_point(data = cycling, aes(x = Weight, y = Height, col =
as.factor(Medal))) + labs(col = "Observed Y")
#cyc_fig_rosesvm
```

```
#gfig2 <- ggplot()+ geom_tile(aes(x = x1, y = x2, fill = as.factor(yhat2)),
data = xgrid, alpha = .5) + labs(fill = "Predicted Y") + geom_point(data =
mydf, aes(x = x1, y = x2, col = as.factor(y))) + labs(col = "Observed Y")
#gfig2
```