Cover Page

The Project Title

Exploring, Analysing and Predicting the Medal Winning Rate For History of Olympics Data (1896-2016)

Author's Name(s)

Lakshmi Prasanna Gundabolu (G01223268)

Course Project Professor's Name

Dr. DuoDuo Liao,Ph.D.

University Name, Course Number and Name

George Mason University

AIT-614 Big Data Essentials

Date

May 10, 2020

AIT-614 Final Project Report

Lakshmi Prasanna Gundabolu

Professor: Duoduo Liao, Ph.D.

PROJECT TITLE

Exploring, Analysing and Predicting the Medal Winning Rate for History of Olympics Data (1896-2016)

AIT-614 Final Project Report

Lakshmi Prasanna Gundabolu

Professor: Duoduo Liao, Ph.D.

Abstract

The cutting edge Olympic Games are driving worldwide games including summer and winter sports rivalries in which a huge number of competitors from around the globe take an interest in an assortment of rivalries.The first olympic games were held in 1896 and the Rio olympics, the recent games were held in 2016. The Olympic games are viewed as the world's premier games rivalry with in excess of 200 countries taking an interest. The Olympic Games are held at regular intervals, with the Summer and Winter Games exchanging by happening like clockwork yet two years separated. The Olympic Games are held at regular intervals, with the Summer and Winter Games exchanging by happening like clockwork yet two years separated.

In this project we preprocessed the dataset(checking for missing values, duplicate values and deleting some attributes which are not useful) using Big Data Tool PySpark, Performed some visualizations and Implemented Correlation Analysis to show which factors are influencing the Athlete's Performance and what would be the important factors we need to consider for predicting the winning rate of athlete. We Performed k-Means and Hierarchical Clustering techniques for identifying groups in a dataset and to get the structure of data we're dealing with. We built a Logistic regression model for predicting how likely an athlete is to win a particular medal and a Linear Regression Model to predict the Winning Rate for either of the medals(Gold, Silver, and Bronze). Additionally, We also made use of Apriori rules to know in which season events are happening most with respect to the medal winning rate. Furthermore, We performed Principal Component Analysis for reducing the dimensionality of Height Attribute and displaying how data points in that attribute are separating. Finally, We performed Inferential Statistics by doing hypothesis testing and also found confidence intervals for numerical attributes in the dataset. Our proposed model helps in analysing the Olympic History from the past 120 years and also helps in predicting how likely the athlete wins the medal.

Introduction

The dataset is about modern Olympic Games, including all the games from Athens 1896 to Rio 2016 which provides the information about athletes from different nations, different sports and events.In this project, our goal is to shed light on major patterns in Olympic history. For example, answering questions like How many athletes, sports, and nations are there, Where do most athletes come from, Who wins medals,What are the characteristics of the athletes e.g., gender and physical size.

The full dataset consists of 271,116 rows and 15 columns. The dataset analysed here spans both summer and winter Olympics held between 1896 and 2016, it contains details of every event held and the athletes who competed. Each athlete has a unique ID number, with some athletes participating at multiple events and across different Olympics. The Winter and Summer Games were held in the same year up until 1992. After that, they staggered them such that Winter Games occurred on a four year cycle starting with 1994, then Summer in 1996, then Winter in 1998, and so on. In such a way we have the entire data set spanning the data of the games held in both the seasons.

Our project delves into data from 120 years of Olympic history and seeks to provide an interesting and insightful analysis of this rich dataset.

Project and its Scope

The scope of this project is Predicting the probability of Winning rate of a Medal based on the historical data given in the dataset about the athletes. The Motivation of this project is preprocessing the data, analyzing it, visualizing and performing different Machine Learning Techniques like Regression for prediction and Clustering for grouping. By fitting the Logistic Regression Model to the dataset we found the chance of winning either of the medals for each athlete.

We performed Linear regression for Predicting whether the athlete wins the medal or not. By performing K-Means Clustering Technique we analysed the height and weight distribution of players who won the medal and who lost the medal. With Hierarchical Clustering we grouped the Gold, Silver, and Bronze Medal Winners who are having the same age. The Main Purpose of performing Principal Component Analysis is reducing the dimensions of Height Attribute to know how similar height of athletes are being seperated into a single dimension. We also used Apriori Rules to know in which season most medals are won by the athletes. Finally, in this project we performed Inferential Statistics by conducting a hypothesis test for predicted value to determine the probability that a given condition is true and computing the p-value to know whether the null hypothesis is true or alternative is true.

Background and Related Work:

- Some related work was proposed on real-time Exploratory data analysis of the Olympics participation data set, to understand the impact of historical world events on the international games.

  Source: Shaswat Rungta

  Srungta. (n.d.). srungta/120-years-of-olympics.

  Retrieved from https://github.com/srungta/120-years-of-olympics

- Most of them focused on visualization, analyzing and drawing some key insights from the dataset.

  Source: Jagajithmk. (2019, March 9). jagajithmk/Tableau-Visualization-120-years-of-Olympic-history.

  Retrieved from https://github.com/jagajithmk/Tableau-Visualization-120-years-of-Olympic-history

- They also performed linear regression to predict the winning rate of the medals for players.

  Source: ShubhamKmr. (n.d.). ShubhamKmr/120-years-of-Olympic-sports-history-and-their-results.

  Retrieved from https://github.com/ShubhamKmr/120-years-of-Olympic-sports-history-and-their-results

Objectives

The main goal of our project is the likelihood of a participant winning a medal (any medal i.e gold, silver or bronze) in an event based on his/her physical attributes and nationality is being predicted using logistic regression, which is machine learning technique. Here are the list of all the objectives of our overall project with clear description:

- Firstly, we have preprocessed the dataset by checking for Null values and finding the unique values for the dataset.
- Additionally, we have formatted the dataset by deleting inconsistent attributes which are not used for our prediction and type casting the attributes which are incompatible.
- We added categorical attributes like Age_level and Height_level which says if a player is young or old and short or tall for drawing better insights in the exploratory analysis.
- Visualised the dataset in such a way that everyone can understand what are the factors that affect the behaviour of a participant to win.
- Followed by visualization, Analysing and exploring the olympic medals of the dataset by using PySpark. Pyspark is used for big data analytics, for handling the big data and for parallel processing of huge amounts of data.
- We have drawn a few key insights from the dataset using the sqldf package in R Studio.
- Furthermore, we performed Principal Component Analysis and Apriori Rules for knowing more about the dataset.
- Then we have Used correlation matrix to know which attributes have strong correlation with attribute and which are affected by it.
- We have Predicted winners by using Machine Learning Techniques like Linear regression technique and Logistic Regression.
- We performed K-Means and Hierarchical Clustering Techniques for classifying and grouping the attributes in the dataset.
- We have Performed Inferential statistics for finding the accuracy of models, and also by performing hypothesis tests we have calculated how accurate our prediction is.

The Dataset

▪ Selection

We have selected the dataset from Github repository.

Source: Shaswat Rungta

Srungta. (n.d.). srungta/120-years-of-olympics.

Retrieved from https://github.com/srungta/120-years-of-olympics

Description (Who, Why, What)

Who and Why:

Sports reference is a group of sites to provide statistics data about sports happening across the world for sport fans. Their aim is to be the easiest-to-use, fastest, most complete sources for sports statistics anywhere.

Our Dataset Description:

The data set is a set of 15 features: 5 numerical and 10 categorical. This dataset has 271,116 entries in the "athletes_events.csv" file. Each entry is based on an individual athlete competing in an individual Olympic event.

Some of the attributes that we are using in this dataset are as follows:

ID - Unique number for each athlete

Name - Name of the Athlete

Sex - Gender of the Athlete

Age - Age of the Athlete

Height - Height of the Athlete

Weight -Weight of the Athlete

Team - Name of the team

NOC - National Olympic Committee 3-letter code

Games - Year and season

Year - Year of the Game

Season - Summer or Winter

City - Name of the Host city

Sport -Name of the Sport

Event - Name of the Event

Medal - Type of Medal (Gold, Silver, Bronze, or NA)

Dataset Schema

Here is a snapshot of Olympic Dataset Schema:

In [1]:
```
#importing the libraries needed for reading the csv files
import pandas as pd
import os
```

In [2]:
```
pwd #checking the default directory
```

Out[2]: 'C:\\Users\\laksh'

In [3]:
```
#reading data from the dataset
data = pd.read_csv(r"C:\users\laksh\downloads\athlete_events.csv")
data.head()
```

Out[3]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

In [5]:
```
#calculation correlation between attributes in the dataset
import seaborn as sns
import matplotlib.pyplot as plt
```

Figure: Showing all the attributes in the dataset after reading file

Pre-processing For Drawing Key Insights and Analysis:

We have 10 numerical and 5 categorical variables in our dataset. We tested our dataset for null, duplicate and missing values. We found that there are no duplicate values but four columns namely Age,Weight,Height and Medal have missing values.

Here's snapshots for missing values before and after Imputing:

```
#Missing Values
data.isnull().sum()
#we can see that the
```

```
ID              0
Name            0
Sex             0
Age          9474
Height      60171
Weight      62875
Team            0
NOC             0
Games           0
Year            0
Season          0
City            0
Sport           0
Event           0
Medal      231333
dtype: int64
```

Figure: Before Imputing Missing values

```
#checking the datase
data.isnull().sum()
```

```
ID              0
Name            0
Sex             0
Age             0
Height          0
Weight          0
Team            0
NOC             0
Games           0
Year            0
Season          0
City            0
Sport           0
Event           0
Medal           0
Age_Level       0
Height_Level    0
dtype: int64
```

Figure: After Imputing Missing Values

We have replaced the missing values of the Age attribute with its mean value for the Age attribute in the data.

Similarly we have replaced the missing values of Height,Weight with the most repeated value for their respective columns. The reason behind using the most repeated value for Height and Weight variables in the dataset is that we are aware that the players in Olympics are supposed to meet a specific Height,Weight criteria.

We have imputed the missing values in the medal column with 'None' which specifies that the player hasn't won any medal.

We have dropped the ID, Event ,NOC column which is a 3 letter code for the national committee as they do not add any value to our analysis.

We have added two categorical variables 'Age_Level' and 'Height_Level' classifying them as old or young and tall or short.

Here's a piece of appendix for new columns addition

Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	Age_Level	Height_Level
A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	Gold	Young	tall
A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	Gold	Young	short
Gunnar Nielsen Aaby	M	24.0	180.0	70.0	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	Gold	Young	tall
Edgar Lindenau Aabye	M	34.0	180.0	70.0	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold	Old	tall
Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	Gold	Young	tall

Figure: After Adding the two new categorical variables Age_level and Height_level

Preprocessing for ML Algorithms:

We selected some attributes from the dataset and made it as a subset for prediction.

Later, we have divided the dataset into test and train sets with proportions of 65 and 35. Reason why we took the test set as 35% is to make our analysis more significant.

We have converted some variables into numeric for getting better prediction results.

We scaled and standardized the data based on all parameters.

Finally, We built the models for prediction and grouping the dataset using clustering.

The Architecture of our Proposed System

Conceptual diagram for proposed System:

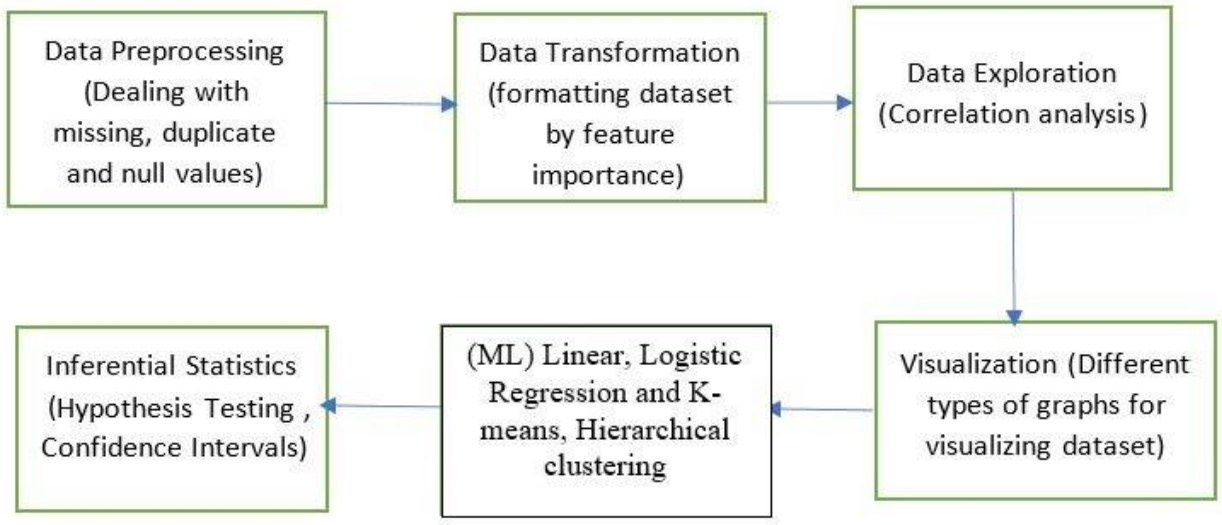


Figure: The overall Architecture of Proposed System

Description of Proposed Architecture:

Firstly, we have preprocessed the data by dealing with the missing and null values.

Then we have performed data transformation by formatting the dataset by feature importance and standardized the data.

Subsequently, we have performed correlation analysis as a part of Data Exploration to find the most correlated variables and have drawn some key insights by using the sqldf package in R studio.

We have performed Data Visualization in Tableau and Python to identify the trends in the data.

We have applied Machine Learning techniques on the data to make some predictions and classifications as well by using clustering algorithms.

In the end we have performed Inferential Statistics by doing hypothesis testing and constructed confidence intervals for numerical attributes.

Data visualization

We have used some plots to visualize the data. This includes Bar graphs, Density plots and other graphs necessary for better visualization of the dataset. We have Visualized the dataset using all the features for finding the correlation between them. Visualization helps us know some facts about the dataset.

Data analysis

We have preprocessed the dataset before analysing it. For this, we have followed some steps like Data preprocessing, Data exploration and Data Transformation. In Data Analysis we mainly have two methods, qualitative and quantitative. Inferential statistics comes under qualitative analysis methods as it performs hypothesis testing and compares the accuracies of different techniques for predicting the grades. Descriptive Analytics comes under the quantitative analytics method which calculates the central tendency of every attribute in the dataset.

Descriptive analytics

This shows how the data has been dispersed with attributes. We have calculated the mean, median and mode of numerical attributes in our dataset which we call as central tendency. It helps to get an in-depth understanding of variables in our data.

We have used NoSQL to find some key insights using the sqldf package in R studio. We also used PySpark in Jupyter Notebook for analysing the dataset and drawing some useful insights which helps for selecting important variables for prediction.

Here are a few insights we derived from the data :

- We have found that the Average age of medal winners is 25.67 in Summer and 25.03 in Winter.
- The count of medals secured in Summer is greater than the count of medals secured in the winter.
- Summer has a greater share of athletes participating with count of athletes 222552 compared to winter with count 48564.
- The most medals secured by the United States were in the sport of Athletics with a count of 1071 followed by Swimming with a count of 1066.
- The most common sport among males is Athletics with a count of 26958 players.
- The most common sport among females is also Athletics with a count of 11666 players.
- The United States bagged the most medals in the entire Olympic games.
- The most common sport played in the United States is Athletics.

Correlation Analysis:

We have performed correlation analysis in Jupyter Notebook using the pearson coefficient of correlation.

From the correlation analysis we can say that the attributes Height and Weight are highly correlated compared with other attributes in the data set.

So, it can be clearly drawn from the figure that Height and Weight of the player affects the winning probability of a medal.

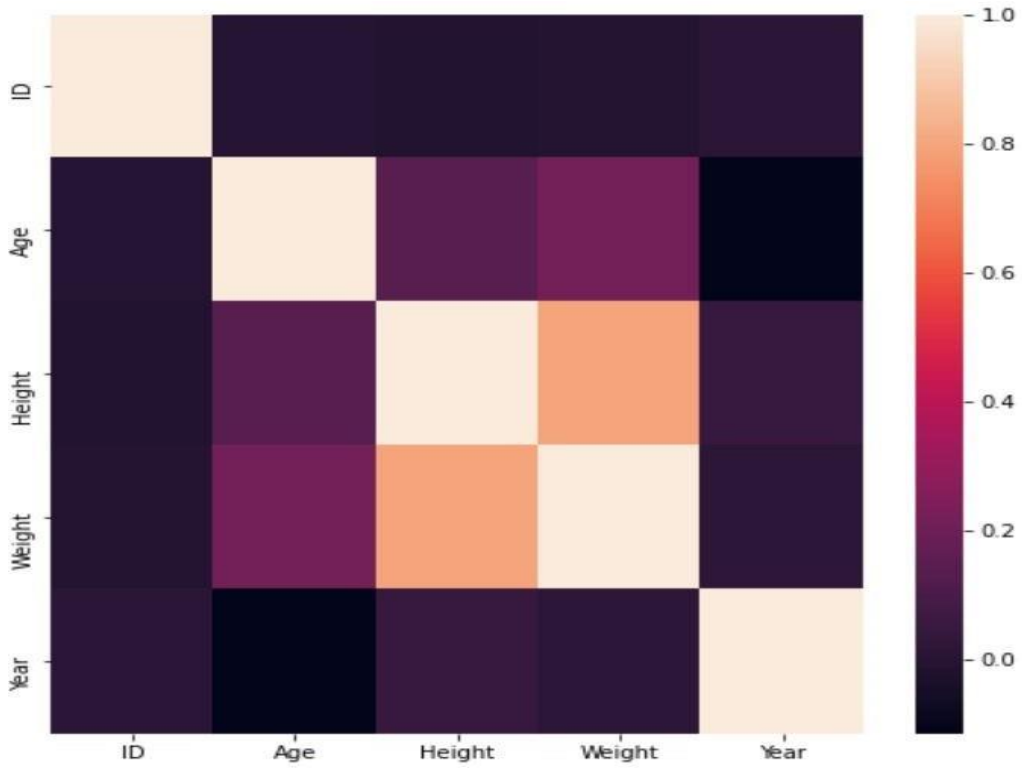


Figure: Correlation Analysis for the variables in Olympic dataset

Inferential Statistics

In Inferential Statistics, we performed hypothesis tests and also we have compared the average performance between the genders in the dataset by considering the t test for differences between groups. The analysis is useful in program outcome evaluation. We have considered the Null and Alternative Hypothesis for one sample t test and initialized confidence interval to 95% by keeping the mean value of sample dataset for every variable.

Hypothesis Testing:

- We Performed one sample t-test for Age variable in the dataset by taking the Null hypothesis as Average Age of Player is 25 and Alternative as Average Age will be less than 25 as shown in the figure below.

```
> #Null Hypothesis: The Average Age of a Player is 25
> #Alternative Hypothesis: The Average Age of Player is <25
> t.test(data$Age, mu = 25.55, conf.level = 0.95) #p-value is greater than significance, so we can accept null hypothesis

        One Sample t-test

data:  data$Age
t = 0.5519, df = 261641, p-value = 0.581
alternative hypothesis: true mean is not equal to 25.55
95 percent confidence interval:
 25.5324 25.5814
sample estimates:
mean of x
  25.5569
```

Figure: Performing Hypothesis Test for Age Variable in the dataset

Explanation: From the above figure, we can conclude that p-value is greater than significance value which is 0.05, so we can accept the null hypothesis by saying that the average age of every player is 25.

- We performed one sample t-test for the Weight variable by considering Null hypothesis as the average weight of all the players will be 70.53 and Alternative as Average Weight of all players will be less than 70.53 as shown in the figure below.

```
> #One Sample t test code on weight Variable
> #Null Hypothesis: The Avverage Weight of all the players will be 70.53
> #Alternative Hypothesis:  The Avverage Weight of all the players will be less than 70.53
> t.test(data$weight.imp.mode, mu=70.53, ci=0.95) # p-value is greater than significant, so we accept null hypothesis

        One Sample t-test

data:  data$weight.imp.mode
t = 0.39325, df = 271115, p-value = 0.6941
alternative hypothesis: true mean is not equal to 70.53
95 percent confidence interval:
 70.49215 70.58685
sample estimates:
mean of x
  70.5395
```

Figure: Performing Hypothesis Test for Weight Variable in the dataset

Explanation: From the above figure, we can conclude that p-value is greater than significance value, so we can accept the null hypothesis by saying that the average Weight of all players is 70.53.

- We performed one sample t-test for the Height variable by considering Null hypothesis as the average Height of all the players will be 177cm and Alternative as Average Height of all players will be less than 177cm as shown in the figure below.

```
> #One Sample t test code on Height Variable
> #Null Hypothesis: The Avverage Height of all the players will be 177cm
> #Alternative Hypothesis:  The Avverage Height of all the players will be less than 177cm
> t.test(data$Height.imp.mode, mu=177, ci=0.95) # p-value is less than significant, so we reject null hypothesis

        One Sample t-test

data:  data$Height.imp.mode
t = -34.421, df = 271115, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 177
95 percent confidence interval:
 176.3378 176.4091
sample estimates:
mean of x
 176.3734
```

Figure: Performing Hypothesis Test for Height Variable in the dataset

Explanation: From the above figure, we can conclude that p-value is less than significance value, so we can reject the null hypothesis and accept alternative hypotheses by concluding that the average Height of all players will be less than 177cm.

Confidence Intervals:

We have constructed confidence intervals for the cleaned data for the variables Age, Weight and Height.

Confidence Interval for Age : Upper limit : 24.91 and Lower limit : 24.87

Confidence Interval for Weight: Upper limit : 69.51 and Lower limit : 69.41

Confidence Interval for Height: Upper limit : 176.15 and Lower limit: 176.07

```
> #Calculating Cofidece Intervals for age, Height and Weight variables in the dataset
> library(Rmisc)
> exp(CI(log(data$Age.imp.mean), ci=0.95))
   upper     mean    lower
24.91503 24.89403 24.87304
> exp(CI(log(data$Weight.imp.mode), ci=0.95))
   upper     mean    lower
69.51017 69.46435 69.41856
> exp(CI(log(data$Height.imp.mode), ci=0.95))
   upper     mean    lower
176.1510 176.1149 176.0789
```

Figure: Confidence intervals for Age, Weight and Height Variables

Advanced analytics (ML, DL, etc.)

In this project, We have used Machine Learning Regression Techniques like logistic and linear in which, logistic regression Model helps to predict the likelihood of participants winning the medal whereas linear regression is used to predict the winning rate of Medal.Furthermore, We also

performed Machine Learning Clustering techniques like K-Means and Hierarchical to know the distribution of height and weight of players who won the medals and who lost the medals. We grouped the Gold, Silver, and Bronze Medal Players having the same age into a single cluster by Hierarchical clustering technique.

Machine Learning Regression Models:

Linear Regression Model:

It is a predictive model which helps in forecasting future outcomes and estimating the metrics which are impractical to measure.

Initially, we formed a subset of the data by removing ID, Year, Event and NOC variables from the original dataset and then we changed the other categorical variables in dataset into numeric including Medal variable which says that (4=None, 3= Silver, 2=Gold and 1= Bronze). We fitted this model to our dataset by splitting the dataset into two parts one is training and other is testing by standardizing the required variables which are used for prediction. We found a summary of the model as shown in the figure below which gives R-squared and Residual Standard Error value for knowing how accurate the model fitted on our dataset.

```
> summary(linear)

Call:
lm(formula = Medal ~ ., data = trainapp)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0084  0.1803  0.2630  0.3358  0.8493

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.615e+00  4.291e-02 107.551   <2e-16 ***
Sex          1.016e-01  4.609e-03  22.047   <2e-16 ***
Age         -2.727e-03  2.876e-04  -9.479   <2e-16 ***
Height      -4.172e-03  2.984e-04 -13.982   <2e-16 ***
Weight      -2.958e-03  2.238e-04 -13.218   <2e-16 ***
NOC         -7.056e-04  2.688e-05 -26.248   <2e-16 ***
Season       6.078e-02  4.726e-03  12.860   <2e-16 ***
City        -3.688e-05  1.495e-04  -0.247    0.805
Sport       -2.068e-03  9.233e-05 -22.401   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7676 on 189365 degrees of freedom
Multiple R-squared:  0.01512,   Adjusted R-squared:  0.01508
F-statistic: 363.5 on 8 and 189365 DF,  p-value: < 2.2e-16
```

Figure: Summary of Linear Regression shows R2 Value and Residual standard Error

Later we found the Prediction rate of every athlete who participated in the Olympic Event by using testset. To know how many values are predicted correctly we found confusion matrix for actual medal variable in test set and predicted medal variable in test set.

Logistic Regression Model:

It is another Technique of Machine Learning which gives a binomial outcome with one or more independent variables. It is an instance of classification technique which can be used to predict qualitative response. It can estimate the probability of happening an event based on some dependent variables.

It fitted well to our dataset and correctly predicted the probability of winning rate of each player in the Olympics. For Performing this Model we used the same test and train sets obtained from the original dataset. We found a summary of the Model as shown in the below figure and Predicted the probability of winning either of the three Medals for every player and built a confusion matrix. We also found multicollinearity between the three independent variables which were used to fit the model.

```
> summary(logistic)

Call:
glm(formula = Medal.imp.none ~ Height.imp.mode + Weight.imp.mode +
    Age.imp.mean, family = "binomial", data = train1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7515   0.2925   0.3166   0.3276   0.5389

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     5.523655   0.262732  21.024  < 2e-16 ***
Height.imp.mode -0.010970   0.001808  -6.068 1.30e-09 ***
Weight.imp.mode -0.007172   0.001277  -5.615 1.97e-08 ***
Age.imp.mean    -0.003832   0.001771  -2.163   0.0305 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 66804  on 171230  degrees of freedom
Residual deviance: 66522  on 171227  degrees of freedom
AIC: 66530

Number of Fisher Scoring iterations: 6
```

Figure: Summary of Logistic Regression showing Residual deviances and p-values

We found the Validation set from the train dataset and predicted the values which gave 95 % accuracy i.e., they have predicted almost equal for how likely the player is winning the medal. We performed ANOVA test which shows us the analysis of the deviance table for both train and

test datasets in the model. We then calculated the coefficients for all the variables with standard error and p-values.

Finally, we interpreted the medal coefficients with predicting the probability of how likely the player is winning the games.

Machine Learning Clustering Techniques:

K-Means Clustering Technique:

K-Means clustering technique is used to get the structure of data we are dealing with. It partitions the given data variables into clusters and assigns them to a cluster such that the sum of squared distance between the data points and cluster's centroid is at minimum.

We Performed K-Means clustering on Weight and Height variables of the dataset who have won the medals. We initialized two centroids for our dataset and kept n=25 which can try 25 different random starting assignments and then select the best results corresponding to one with lowest within the cluster variation. we will get a more stable result by taking nstart=25.

Similarly, we performed K-Means clustering for players who lost the medal by keeping nstart=25 and centers are two which means two centroids. We calculated contingency tables for actual clusters and variables in column to know which observations are lying under which cluster.

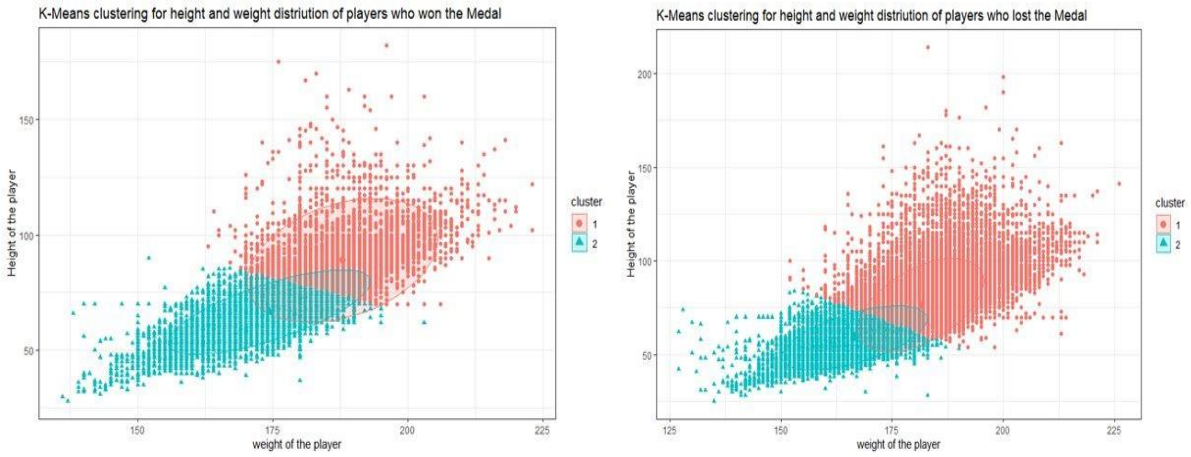


Figure: Height and Weight Distributions of Players who won medals and who lost medals

From the above plot we can clearly say that players who lost medals and who won medals have similar distributions in their height and weight.

Finally, we plotted the clusters for displaying the distributions of height and weight of the players and compared both plots to know how heights and weights of the athletes are distributed.

Hierarchical Clustering Technique:

Hierarchical is an alternative approach to K-Means for identifying groups in variables of the dataset. Before performing Hierarchical clustering we should remove or estimate all the missing values in the variables and data must be standardized to make variables comparable. Once we perform this, the procedure is iterated until all points are combined into a new bigger cluster.

For our dataset we used Hierarchical clustering for Age variable which can group all the players who won the Gold medals with the same age into one cluster/group. Here, we used Agglomerative hierarchical clustering for plotting Dendrograms which can visualize the clusters in a better way by forming rectangle boxes that separates all the clusters. These plots look better not only for visualization but also for understanding the clusters.

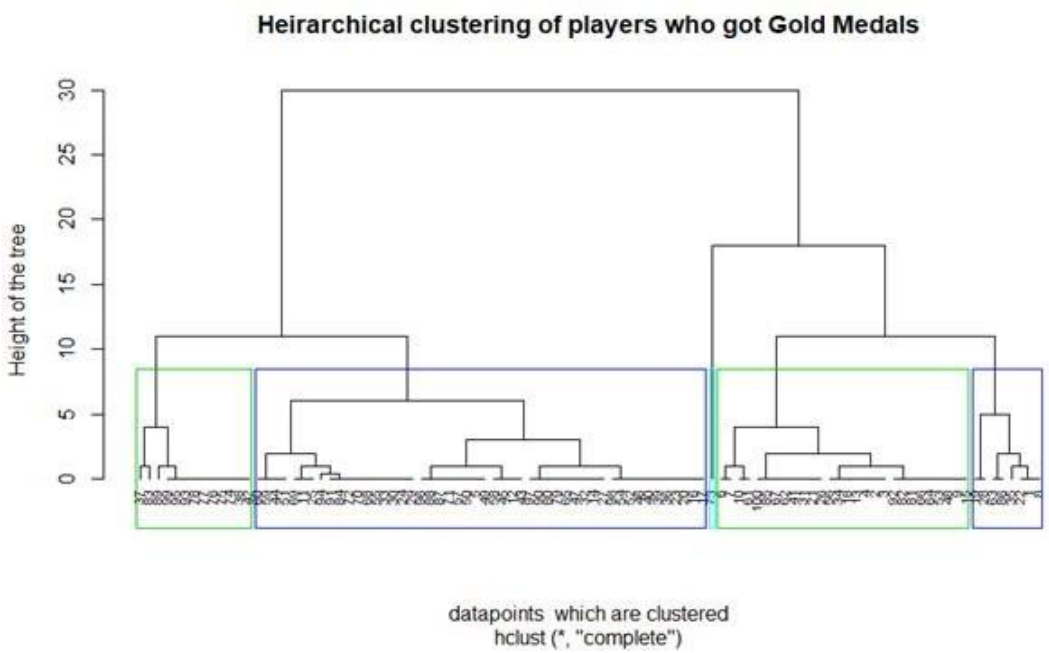


Figure: Hierarchical clustering of players who won gold medals

Similarly, We performed this clustering for players who won Silver and Bronze medals with the same age into one cluster.

Apriori Rules:

Apriori Rules are used to find Association between all the variables in the dataset. It is basically used for Marketing Analysis to know how likely the second product is bought with the first product. Here, we mainly have three metrics namely, Support, Confidence and Lift which are control of rules to be generated.

Support states how frequently the variable is appearing in the dataset. Rule of Confidence implies that whenever a Left hand side item was purchased, the item on right hand side also will be purchased 100% of times. Lift shows how often the rules have been found true, which means how often conditions actually happened compared to the estimated chance for that to happen.

```
> inspect(rules) #It gives the list of all significant association rules.
    lhs                                      rhs              support   confidence lift count
[1] {}                                    => {allmedals=Nan} 1.0000000 1          1    271116
[2] {Height.imp.mode=[180,226]}           => {allmedals=Nan} 0.5040020 1          1    136643
[3] {Sex=M}                               => {allmedals=Nan} 0.7251287 1          1    196594
[4] {Season=Summer}                       => {allmedals=Nan} 0.8208737 1          1    222552
[5] {Medal.imp.none=None}                 => {allmedals=Nan} 0.8532621 1          1    231333
[6] {Sex=M,Season=Summer}                 => {allmedals=Nan} 0.6016207 1          1    163109
[7] {Sex=M,Medal.imp.none=None}           => {allmedals=Nan} 0.6198970 1          1    168064
[8] {Season=Summer,Medal.imp.none=None}   => {allmedals=Nan} 0.6951416 1          1    188464
[9] {Sex=M,Season=Summer,Medal.imp.none=None} => {allmedals=Nan} 0.5107150 1      1    138463
```

Figure: Snapshot of Apriori rules performing on Olympics dataset.

We Used these Rules on our dataset. From the figure above we found that on the right hand side we have a single variable allmedals with respect to all other variables in the dataset on the left hand side. The support metric for season variable is upto 82% with variable allmedals on right hand side which has confidence and lift metrics 1 with more number of count. This implies that players who played olympics in summer won most of the medals.

Finally, we can conclude that the support metric is high for summer season and medal column with respect to allmedals variable, in which we consider only the summer season variable as the other variable medal and allmedal are the same one with categorical values and other with numerical values which makes the same sense.

Principal Component Analysis:

Large Datasets are increasingly widespread in many disciplines, the dataset which we selected also has more than two hundred thousand rows so we performed Principal Component Analysis which is used to reduce the dimensionality of the datasets which are increasing interoperability but at the same time it minimizes the information loss. So we tried to implement it on our dataset to know how attributes in the dataset are varying and how they are being separated by using a scatter plot. Moreover, we did not use this reduced dimensioned variables on any other techniques as we need every observation in our dataset for prediction.

Finally, we performed PCA on Height, Weight and Age variables in the dataset to know how they are separated by both components and also to transform the data into fewer dimensions.

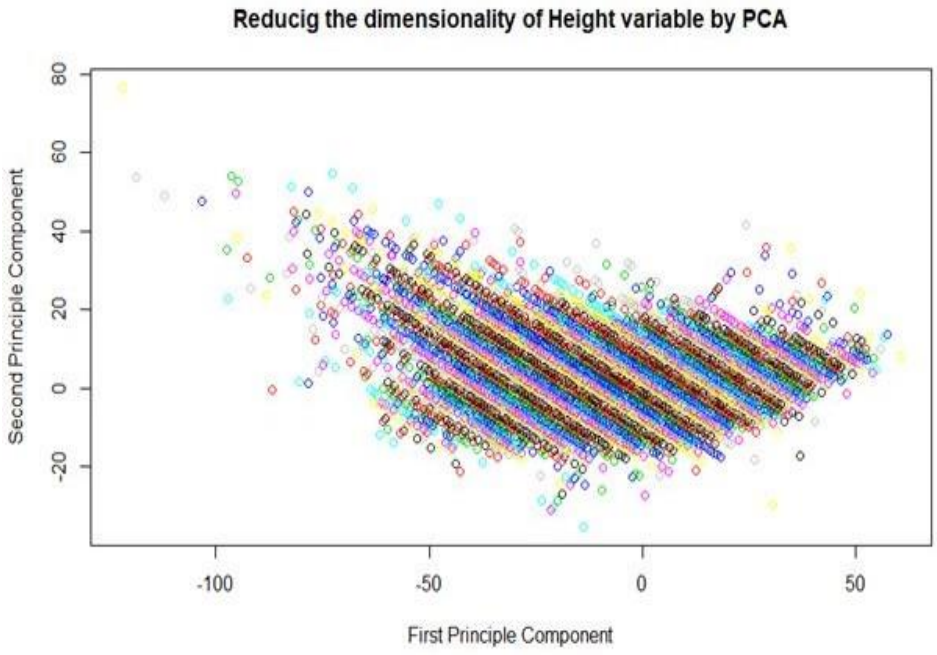


Figure: Snapshot of Principal Component Analysis for Height Variable

We can clearly see from the above scatter plot that the first component is separating the classes well while the second component does not, in which players with specific height are separating with different colours in the plot.

SW Development platforms

Frameworks: Jupyter notebook (python and R kernels), Tableau for visualization, Pyspark and NoSQL,NLTK, Machine Learning regression techniques.

Languages: Python ,R and NoSQL

HW Development platforms

Hardware - Intel Pentium (dual core) compatible processor or any multiprocessor based computer with a minimum 2Ghz or greater processor , a 64-bit system and network interface card.

Disk Space - The space should be a minimum of 1GB.

Memory – The recommended memory is 4GB or more.

Experimental results and analysis

We used Tableau as a Visualization tool to obtain a few Visualizations from our dataset.

Here are some Visualizations relevant to our analysis:

Tableau Visualization for Athlete Count:

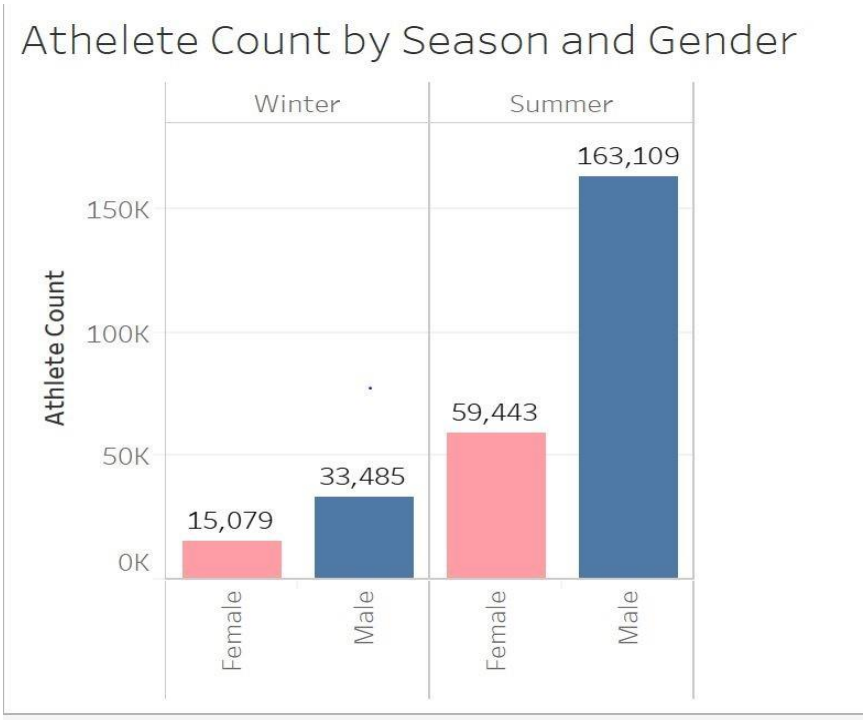


Figure: Visualization for Athlete count by Season and Gender.

Explanation: From the above visualization, We can see that the count of male athletes is greater than the count of female athletes in both seasons.

The total athlete count is greater in Summer compared to Winter.

Visualization for Athlete Count in Summer:

Figure : Visualization for countries participated in Summer

Visualization for countries participated in Winter:

Figure: Visualization for Athlete Count in Winter

Explanation: From the above Visualizations we can state that the countries participating in Summer are almost the same as the countries participating in Winter.

Visualization for Athlete Count in Summer:

Figure : Visualization for Athlete Count in Summer

Explanation:

From the above Visualizations we can say that the Athlete Count in Summer is greater than the athlete count in Winter.

There is a significant rise in the Athlete Count from the beginning to recent years in both the Seasons.

Visualization for top countries to secure medals.

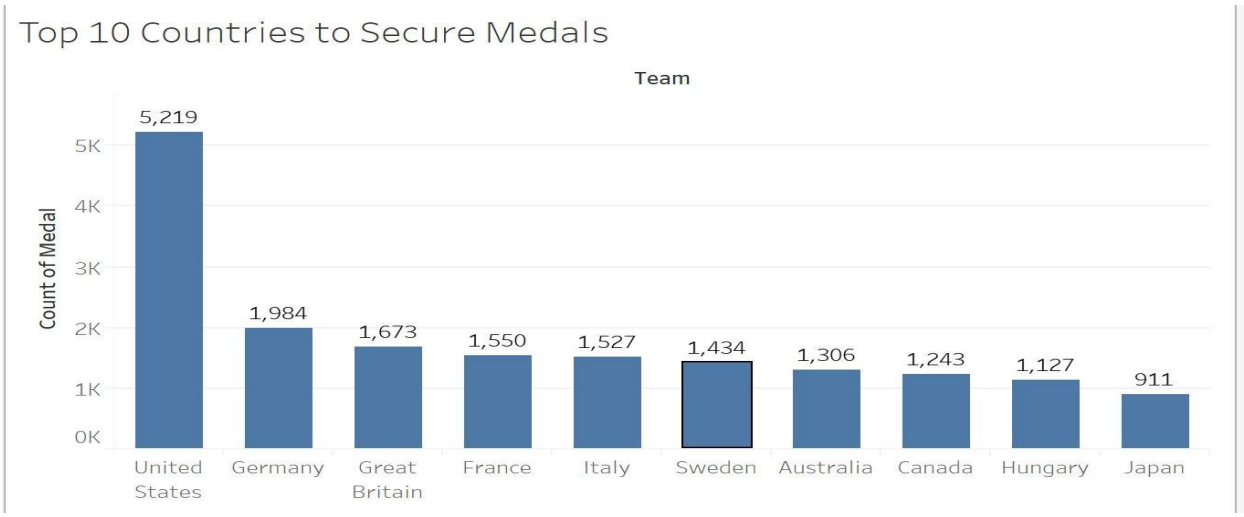


Figure: Visualization for top countries to secure medals.

Explanation:

Above is a visualization for top 10 countries that have secured medals. The United States is the top Country followed by Germany and Great Britain.

Visualization using Word Cloud :

We have used the NLTK package in Jupyter Notebook to construct a word cloud for the popular city and Sports.

Figure: Word Cloud for popular Sports.

Explanation: From the above Word Cloud we can see that Basketball,Ski and Jumping are few of the popular sports in the Olympics.

Figure: Word Cloud for most popular cities.

Explanation: From the above word cloud we can see that London,Barcelona and Sochi are few popular cities for the Olympic games.

Data analytics

We have performed some analysis on the data in NoSQL using the sqldf package in R studio.

Here are some snapshots of few of our insights:

- What is the participation count in both the seasons?

Snapshot:

```
> count_summerathletes <- sqldf("select COUNT(d.name) from data d where d.
Season='Summer'")
> count_winterthletes <- sqldf("select COUNT(d.name) from data d where d.S
eason='Winter'")
> print(count_summerathletes)
  COUNT(d.name)
1        222552
> print(count_winterthletes)
  COUNT(d.name)
1         48564
```

Key Insight: From the above sql analysis we can state that Summer has a larger share of participation compared to Winter.

- What is the Average Age of the Athlete in both the seasons?

Snapshot:

```
> Avgmedalwinner_age <- sqldf("select d.season,avg(d.age) as Avg_age ,COUNT(d.Medal) from
 data d GROUP BY d.season HAVING COUNT(d.Medal)>0")
> print(Avgmedalwinner_age)
  Season  Avg_age COUNT(d.Medal)
1 Summer 25.67405          34088
2 Winter 25.03915           5695
```

Key Insight : From the above sql analysis we can state that Average age in Summer is 25.67 and Average age in Winter is 25.03.

- Which is the common sport among males?

Snapshot:

```
> commonsport_male <- sqldf("select sex,sport,count(sport) as count from data
 where sex= 'M' group by sport order by count desc limit 1")
> print(commonsport_male)
  Sex     Sport count
1   M Athletics 26958
```

Result: Athletics is the most common sport among the males.

- Which is the common sport among females?

Snapshot:

```
> commonsport_female <- sqldf("select sex,sport,count(sport) as count from da
ta where sex= 'F' group by sport order by count desc limit 1")
> print(commonsport_female)
  Sex     Sport count
1   F Athletics 11666
```

Result: Athletics is the most common sport among the females.

- Which is the most common sport in the USA?

Snapshot:

```
> commonsport_USA <- sqldf("select distinct sport, team, count(medal) as coun
t from data where team='United States' group by sport order by count desc lim
it 3")
> print(commonsport_USA)
       Sport          Team count
1  Athletics United States  1071
2   Swimming United States  1066
3 Basketball United States   341
```

Result: Athletics is the most common sport in the USA followed by Swimming and Basketball.

- Compare the male athlete count from the beginning to recent year.

Snapshot:

```
> male_growth <- sqldf("select sex,COUNT(sex) as COUNT,year from data where year in(1900,
2016) group by year,sex HAVING sex='M' order by year asc")
>
> print(male_growth)
  Sex COUNT Year
1   M  1903 1900
2   M  7465 2016
```

Result: Male participation has increased in the recent years compared to the beginning of the games.

- Compare the female count from the beginning to recent year.

Snapshot:

```
female_growth <- sqldf("select sex,COUNT(sex) as COUNT,year from dat
a where year in(1900,2016) group by year,sex HAVING sex='F' order by year asc
")
> print(female_growth)
  Sex      COUNT Year
1   F         33 1900
2   F       6223 2016
```

Result : Females have a higher participation in the recent games compared to the beginning.

Interpret the results

Linear Regression Results:

Here is a snapshot of linear regression analysis where we have converted the categorical variables of Medal type to numerical by assigning the values as the following :

Bronze :1 , Gold : 2 , None :3 , Silver :4

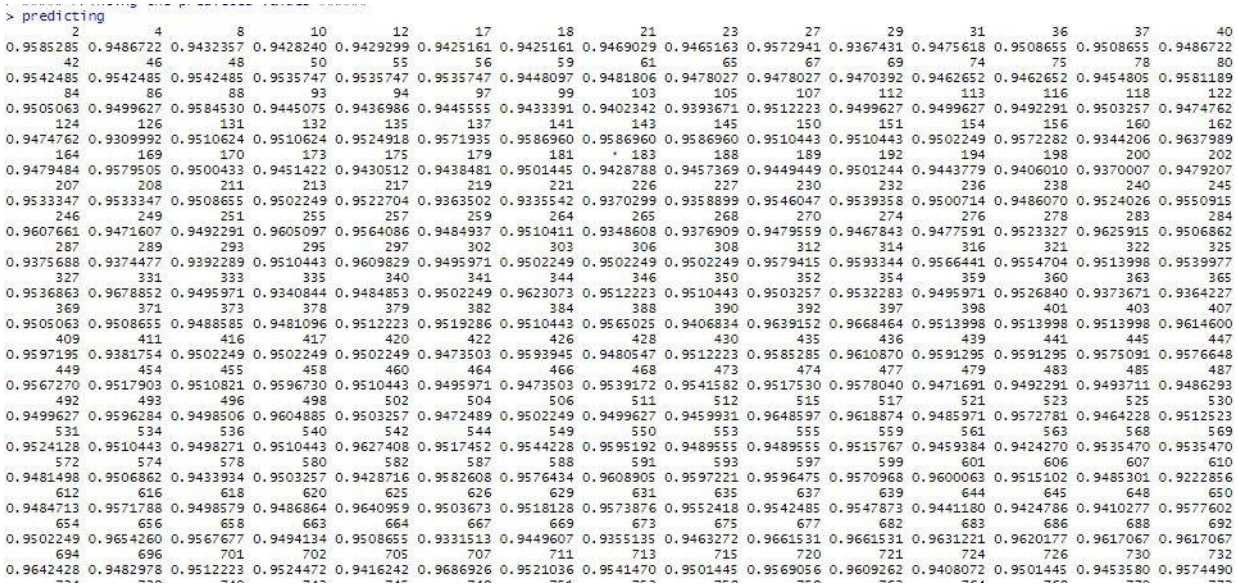
```
> #predicting the winning medal rate
> li=predict(linear,testapp)
> li
       1        2        4       14       15       17       21       24       26       29       30       32       34       35       37       40       41
3.777089 3.830615 3.659129 3.658286 3.652242 3.652242 3.688020 3.681977 3.681977 3.694611 3.686728 3.804828 3.709427 3.770694 3.696749 3.664435 3.729857
      43       47       49       50       52       53       61       62       63       65       67       69       73       74       79       87      100
3.775580 3.775580 3.775580 3.764821 3.764821 3.764821 3.788424 3.788424 3.788424 3.782380 3.782380 3.771179 3.760088 3.760088 3.748776 3.690574 3.657017
     101      109      111      115      117      119      121      122      123      127      134      136      139      144      146      148      149
3.564290 3.675085 3.699452 3.677270 3.685872 3.639956 3.811957 3.800866 3.800866 3.464950 3.720438 3.759958 3.539779 3.825749 3.730852 3.755763 3.847598
     150      152      160      170      179      183      184      185      188      189      194      196      197      198      199      201      209
3.771825 3.830166 3.637098 3.776197 3.720044 3.619392 3.688453 3.606480 3.672291 3.661348 3.532370 3.699334 3.699334 3.691322 3.731216 3.621245 3.695985
     210      211      217      226      233      237      243      251      256      257      259      260      262      264      270      272      275
3.682333 3.682333 3.842400 3.593856 3.863424 3.871289 3.852119 3.773882 3.764316 3.642701 3.720808 3.720421 3.737058 3.594649 3.686354 3.712625 3.724864
     276      285      286      294      298      306      307      317      318      322      325      327      329      336      338      340      341
3.649496 3.494492 3.620138 3.657887 3.691443 3.644553 3.644553 3.834570 3.742374 3.667118 3.800012 3.710949 3.712872 3.744778 3.744778 3.734019 3.570093
     345      348      358      359      364      366      369      373      374      377      378      383      389      390      392      393      406
3.672200 3.687243 3.702722 3.738227 3.701555 3.563052 3.719618 3.739325 3.739325 3.728566 3.728566 3.771233 3.768350 3.692385 3.914816 3.636992 3.672200
     407      409      410      412      416      420      422      426      428      430      433      437      438      440      442      444      447
3.714818 3.681213 3.665760 3.828380 3.750796 3.750796 3.666209 3.793809 3.655901 3.803730 3.791532 3.666434 3.833145 3.833145 3.833145 3.503503 3.904840
     448      458      465      466      469      472      475      477      478      479      485      491      494      500      502      508      509
3.684377 3.889614 3.750600 3.717512 3.705241 3.746097 3.682881 3.682881 3.665760 3.708284 3.730202 3.737917 3.933427 3.738511 3.700534 3.494319 3.483007
     516      518      519      520      531      539      540      547      552      561      567      568      570      587      591      592      596
3.766368 3.643720 3.751629 3.624171 3.788925 3.668983 3.668983 3.779195 3.615374 3.664684 3.617750 3.617750 3.617750 3.744235 3.803744 3.791583 3.695096
     599      601      602      603      605      616      628      630      631      632      635      637      639      640      643      648      655
3.713303 3.693197 3.681885 3.681885 3.783039 3.761295 3.830981 3.820701 3.702115 3.791968 3.804896 3.788241 3.805717 3.805717 3.759065 3.614766 3.946723
     659      668      672      675      677      681      683      684      685      690      695      697      699      700      701      704      706
3.922758 3.745952 3.619404 3.756139 3.802547 3.802547 3.812642 3.801330 3.801330 3.743131 3.657495 3.625476 3.581628 3.581628 3.662182 3.730419 3.818771
     707      712      713      716      726      727      730      732      736      739      745      749      750      752      754      756      757
3.818771 3.623291 3.743942 3.878703 3.698993 3.706027 3.470121 3.658508 3.680931 3.652595 3.811618 3.767723 3.638230 3.607480 3.758520 3.714857 3.714857
     758      764      769      773      780      784      786      787      789      792      794      799      802      803      805      807      810
3.816405 3.750097 3.745855 3.723226 3.716291 3.756695 3.669775 3.647741 3.791770 3.791770 3.791770 3.607388 3.746412 3.528865 3.528865 3.753342 3.834806
     814      817      821      823      831      841      842      844      847      851      852      856      857      861      868      873      879
3.578815 3.728854 3.494176 3.776892 3.562048 3.782062 3.621736 3.765144 3.714932 3.871466 3.776336 3.860898 3.768425 3.683292 3.714049 3.788921 3.690616
     881      885      886      889      902      904      907      908      909      913      914      915      917      923      926      931      937
3.690616 3.663523 3.663523 3.745347 3.554839 3.480662 3.672805 3.810460 3.810460 3.773301 3.773301 3.773301 3.761509 3.725651 3.645405 3.652794 3.660337
     941      943      947      950      953      956      958      961      967      969      971      974      978      983      984      988      993
3.844567 3.833918 3.682985 3.842903 3.832255 3.832255 3.719761 3.813259 3.469301 3.748736 3.549074 3.651492 3.821641 3.677336 3.668824 3.749398 3.643697
     997      999     1000     1010     1015     1017     1018     1019     1022     1023     1027     1030     1039     1044     1048     1049     1052
3.747163 3.643431 3.626186 3.797005 3.635248 3.742020 3.710282 3.664322 3.679660 3.668716 3.800019 3.649930 3.569692 3.627516 3.739274 3.877407 3.719277
```

Figure: Snapshot of linear regression model on Olympic dataset

Explanation: We can state linear regression model is predicting every player who is participating in Olympics will lost the medal and some will gain Silver medal from the experimental results we got as shown in the above figure

Logistic Regression Results:

Below is a snapshot of the Logistic regression model used to predict how likely it is for a non medal winner to win a medal.



```
> predicting
       2        4        8       10       12       17       18       21       23       27       29       31       36       37       40
0.9585285 0.9486722 0.9432357 0.9428240 0.9429299 0.9425161 0.9425161 0.9469029 0.9465163 0.9572941 0.9367431 0.9475618 0.9508655 0.9508655 0.9486722
      42       46       48       50       55       56       59       61       65       67       69       74       75       78       80
0.9542485 0.9542485 0.9542485 0.9535747 0.9535747 0.9535747 0.9448097 0.9481806 0.9478027 0.9478027 0.9470392 0.9462652 0.9462652 0.9454805 0.9581189
      84       86       88       93       94       97       99      103      105      107      112      113      116      118      122
0.9505063 0.9499627 0.9584530 0.9445075 0.9436986 0.9445555 0.9433391 0.9402342 0.9393671 0.9512223 0.9499627 0.9499627 0.9492291 0.9503257 0.9474762
     124      126      131      132      135      137      141      143      145      150      151      154      156      160      162
0.9474762 0.9309992 0.9510624 0.9510624 0.9524918 0.9571935 0.9586960 0.9586960 0.9586960 0.9510443 0.9510443 0.9502249 0.9572282 0.9344206 0.9637989
     164      169      170      173      175      179      181      183      188      189      192      194      198      200      202
0.9479484 0.9579505 0.9500433 0.9451422 0.9430512 0.9438481 0.9501445 0.9428788 0.9457369 0.9449449 0.9501244 0.9443779 0.9406010 0.9370007 0.9479207
     207      208      211      213      217      219      221      226      227      230      232      236      238      240      245
0.9533347 0.9533347 0.9508655 0.9502249 0.9522704 0.9363502 0.9335542 0.9370299 0.9358899 0.9546047 0.9539358 0.9500714 0.9486070 0.9524026 0.9550915
     246      249      251      255      257      259      264      265      268      270      274      276      278      283      284
0.9607661 0.9471607 0.9492291 0.9605097 0.9564086 0.9484937 0.9510411 0.9348608 0.9376909 0.9479559 0.9467843 0.9477591 0.9523327 0.9625915 0.9506862
     287      289      293      295      297      302      303      306      308      312      314      316      321      322      325
0.9375688 0.9374477 0.9392289 0.9510443 0.9609829 0.9495971 0.9502249 0.9502249 0.9502249 0.9579415 0.9593344 0.9566441 0.9554704 0.9513998 0.9539977
     327      331      333      335      340      341      344      346      350      352      354      359      360      363      365
0.9536863 0.9678852 0.9495971 0.9340844 0.9484853 0.9502249 0.9623073 0.9512223 0.9510443 0.9503257 0.9532283 0.9495971 0.9526840 0.9373671 0.9364227
     369      371      373      378      379      382      384      388      390      392      397      398      401      403      407
0.9505063 0.9508655 0.9488585 0.9481096 0.9512223 0.9519286 0.9510443 0.9565025 0.9406834 0.9639152 0.9668464 0.9513998 0.9513998 0.9513998 0.9614600
     409      411      416      417      420      422      426      428      430      435      436      439      441      445      447
0.9597195 0.9381754 0.9502249 0.9502249 0.9502249 0.9473503 0.9593945 0.9480547 0.9512223 0.9585285 0.9610870 0.9591295 0.9591295 0.9575091 0.9576648
     449      454      455      458      460      464      466      468      473      474      477      479      483      485      487
0.9567270 0.9517903 0.9510821 0.9596730 0.9510443 0.9495971 0.9473503 0.9539172 0.9541582 0.9517530 0.9578040 0.9471691 0.9492291 0.9493711 0.9486293
     492      493      496      498      502      504      506      511      512      515      517      521      523      525      530
0.9499627 0.9596284 0.9498506 0.9604885 0.9503257 0.9472489 0.9502249 0.9499627 0.9459931 0.9648597 0.9618874 0.9485971 0.9572781 0.9464228 0.9512523
     531      534      536      540      542      544      549      550      553      555      559      561      563      568      569
0.9524128 0.9510443 0.9498271 0.9510443 0.9627408 0.9517452 0.9544228 0.9595192 0.9489555 0.9489555 0.9515767 0.9459384 0.9424270 0.9535470 0.9535470
     572      574      578      580      582      587      588      591      593      597      599      601      606      607      610
0.9481498 0.9506862 0.9433934 0.9503257 0.9428716 0.9582608 0.9576434 0.9608905 0.9597221 0.9596475 0.9570968 0.9600063 0.9515102 0.9485301 0.9222856
     612      616      618      620      625      626      629      631      635      637      639      644      645      648      650
0.9484713 0.9571788 0.9498579 0.9486864 0.9640959 0.9503673 0.9518128 0.9573876 0.9552418 0.9542485 0.9547873 0.9441180 0.9424786 0.9410277 0.9577602
     654      656      658      663      664      667      669      673      675      677      682      683      686      688      692
0.9502249 0.9654260 0.9567677 0.9494134 0.9508655 0.9331513 0.9449607 0.9355135 0.9463272 0.9661531 0.9661531 0.9631221 0.9620177 0.9617067 0.9617067
     694      696      701      702      705      707      711      713      715      720      721      724      726      730      732
0.9642428 0.9482978 0.9512223 0.9524472 0.9416242 0.9686926 0.9521036 0.9541470 0.9501445 0.9569056 0.9609262 0.9408072 0.9501445 0.9453580 0.9574490
```

Figure : Snapshot of Logistic regression model on Olympic dataset.

Explanation: From the above figure we can conclude that the probability of winning the medal for non medal and medal players is around 95% based on the other attributes mainly height and weights as they have more correlation with medal variables.

K-Means Clustering Results:

Below is a snapshot of the K-Means Clustering technique which clustered all the observations in two clusters.

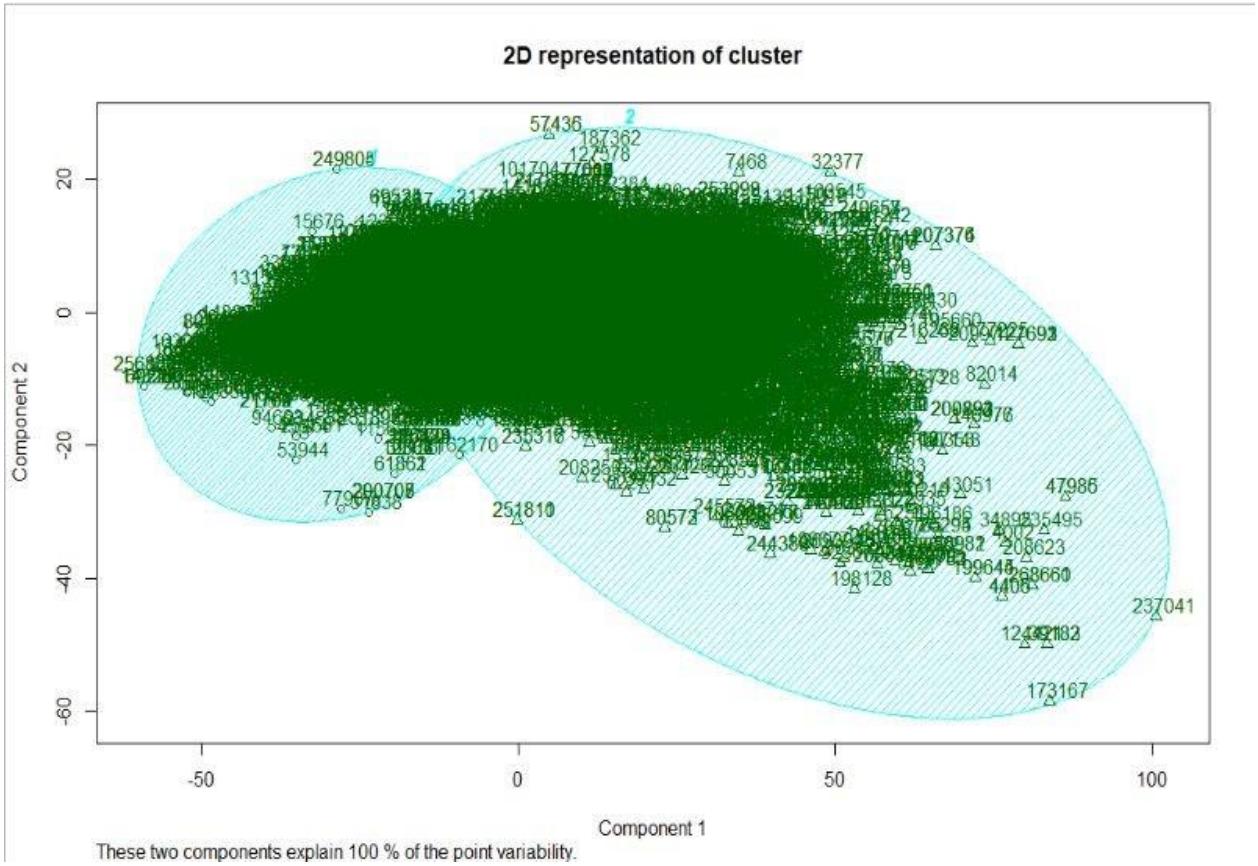
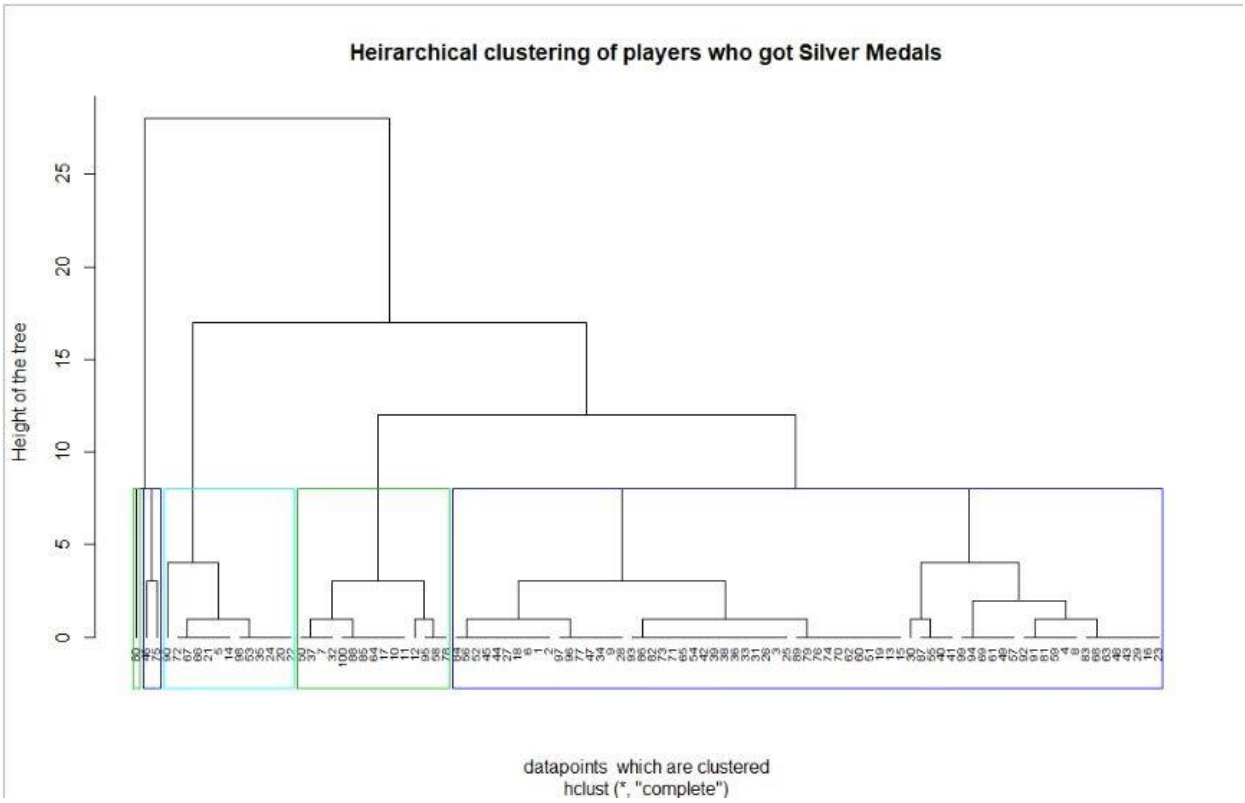


Figure: K-Means Clustering Analysis for height and weight distributions of medal winners

Explanation: From the figure above we can see how data points are clustered in two clusters with respect to centroids and can clearly understand which datapoint is in which cluster by initializing two centroids and nstart=25.

Hierarchical Clustering Results:

Below is the snapshot for Hierarchical clustering technique which groups the Silver medal winners with the same age into a single cluster.



Explanation: From the figure above we can conclude that by performing hierarchical clustering we are able to group players who won silver medals with the same age and separating them with rectangular boxes makes us understand well about which players are grouped together under which cluster.

Conclusions

Conclusions drawn from the overall project are as follows:

- Using K means clustering we found that players who won the medal and who lost the Medal have the same height and weight distributions.
- Using the Logistic regression model we found that the chance of winning a medal for a player with the same height and weight is about 94%.
- Using the linear regression model we can predict every player who is participating in the Olympics will lose the medal and some will gain a Silver medal from the experimental results.
- Height and Weight of the Player plays a very important role in the Olympics as the Correlation for both variables is high.
- Female Participation has shown a significant raise in olympic events from the start to recent years.
- Even though London City participated in more events we analysed that the United states won more number of Medals compared to other countries in the Olympics.
- Athletics is the most common sport played both among males and females.
- The United States is the country which secured the most number of medals.
- The most common sport in the United States is Athletics.

What have you learned from this project and this course? And any suggestions?

We have learnt how to implement and perform analysis using Apache Pyspark in Jupyter Notebook for big data.

We learned Advanced NoSQL for big data analysis and utilised the sqldf package in R studio.

We have learnt applications of clustering algorithm techniques like K-means and Hierarchical Clustering and implemented them in our project.

We also learned the basics about Apriori rules property and applied them to find the association rules for our data.

We Fitted Linear and Logistic Regression models which are machine learning techniques to our dataset to make some predictions.

We have done Principal Component Analysis which simplifies the complexity in high dimensional data.

From lab sessions we learned how to use Virtual machine and also Pig, Hive, Hadoop Mapreduce techniques.

Suggestions:

The overall course work was really awesome, and we enjoyed all classes well by gaining some knowledge about big data tools as well as Advanced SQL and NoSQL languages.

I really suggest doing Project instead of conducting Final Exam for this course. By doing Project we can share ideas and can develop our skills better in this Big Data environment.

Future Work:

- As our dataset contains more categorical variables Random Forest Classifier can be used for better prediction of Winning rate for every non Medal player.
- By using one hot encoding which is a machine learning technique for categorical variables, which are converted to numeric to do a better job in prediction.

References

[1]. Jolliffe, I. T., Cadima, J., Ian T. Jolliffe Ian T. Jolliffe, Jolliffe, I. T., Cadima, J., Jorge Cadima Secção de Matemática, … Ian T. Jolliffe College of Engineering. (2016, April 13). Principal component analysis: a review and recent developments.

Retrieved from https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202

[2]. Prabhakaran, S. (n.d.). eval(ez_write_tag([[728,90],'r_statistics_co-box-3','ezslot_4',109,'0','0']));Association Mining (Market Basket Analysis).

Retrieved from http://r-statistics.co/Association-Mining-With-R.html

[3]. K-means Clustering in R with Example. (n.d.).

Retrieved from https://www.guru99.com/r-k-means-clustering.html

[4]. Hierarchical Cluster Analysis. (n.d.).

Retrieved from https://uc-r.github.io/hc_clustering

[5]. How to perform a Logistic Regression in R. (2015, September 13).

Retrieved from https://www.r-bloggers.com/how-to-perform-a-logistic-regression-in-r/

[6]. Prabhakaran, S. (2019, October 22). Complete Introduction to Linear Regression in R. Retrieved from https://www.machinelearningplus.com/machine-learning/complete-introduction-linear-regression-r/