DSAI Mini-Project
120 years of Olympic games

Sannabhadti Shipra Deepak
Ng Ziqi Natasha
Asok Kumar Gaurav

Aim:

Predict Medal using
Height, Weight, Age and Team

Overview

- Data Preparation
- Exploratory Analysis and Visualisation
- Prediction:
    1. Random Forest
    2. Classification Tree
- Conclusion
- Division of work

Data Cleaning and Preparation

Olympics + NOC Data

- Merge NOC data with Olympic Data
- Get rid of rows with missing data

	ID	Name	Sex	Age	Height	Weight	NOC	Games	Year	Season	City	Sport	Event	Medal	Team
40	16	Juhamatti Tapio Aaltonen	M	28.0	184.0	85.0	FIN	2014 Winter	2014	Winter	Sochi	Ice Hockey	Ice Hockey Men's Ice Hockey	Bronze	Finland
41	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Individual All-Around	Bronze	Finland
42	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Team All-Around	Gold	Finland
44	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Horse Vault	Gold	Finland
48	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Pommelled Horse	Gold	Finland

Summer/Winter Data

- Extract data based on season
- Edit medal column

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
201803	101352	George Stuart Robertson	M	23.0	NaN	NaN	Great Britain	GBR	1896 Summer	1896	Summer	Athina	Tennis	Tennis Men's Singles	0.0
123038	62185	Georgios Koletis	M	NaN	NaN	NaN	Greece	GRE	1896 Summer	1896	Summer	Athina	Cycling	Cycling Men's 100 kilometres	1.0
123037	62185	Georgios Koletis	M	NaN	NaN	NaN	Greece	GRE	1896 Summer	1896	Summer	Athina	Cycling	Cycling Men's 10,000 metres	0.0
116776	59090	Alexandros Khalkokondylis	M	NaN	171.0	64.0	Greece	GRE	1896 Summer	1896	Summer	Athina	Athletics	Athletics Men's 100 metres	0.0
116777	59090	Alexandros Khalkokondylis	M	NaN	171.0	64.0	Greece	GRE	1896 Summer	1896	Summer	Athina	Athletics	Athletics Men's Long Jump	0.0

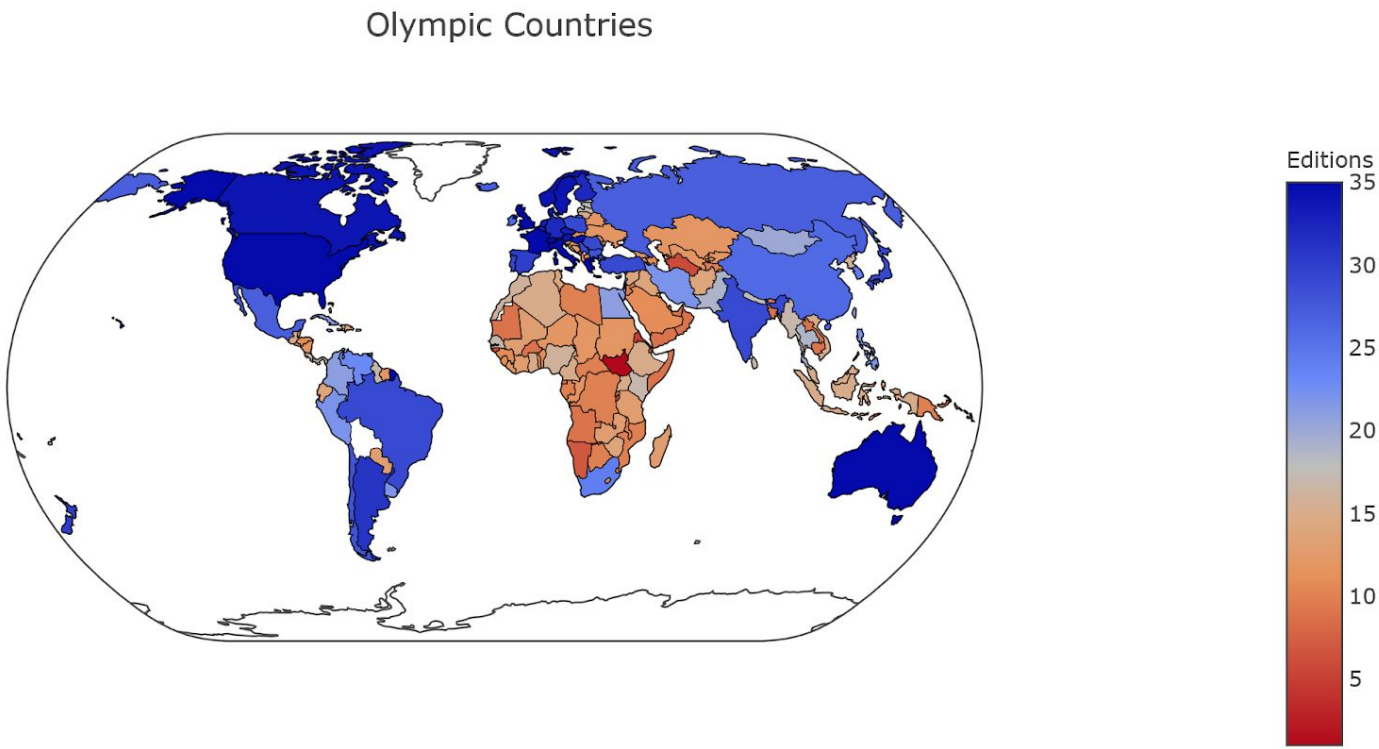
Medal Data

```
test = CountryPerformance(complete, Country = "Brazil", Medal = True, gender = "M")
test = WhichMedal(test, medal = "Gold")
test = YearSort(test, ascending = False)
test = test.reset_index(level=0, drop=True)
test = test.drop(['index', "ID"] , axis='columns')
test.head()
```

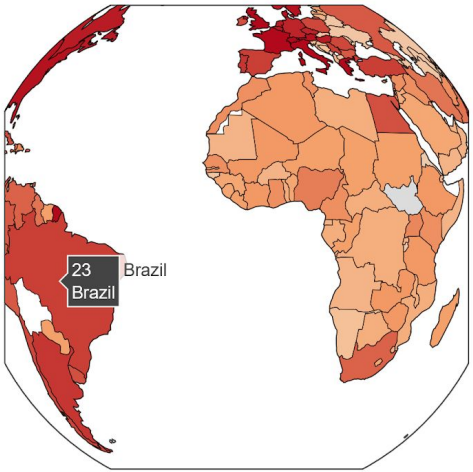
	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	Ricardo Samuel Lucarelli Santos de Souza	M	24.0	195.0	79.0	Brazil	BRA	2016 Summer	2016	Summer	Rio de Janeiro	Volleyball	Volleyball Men's Volleyball	Gold
1	Rodrigo Caio Coquette Russo	M	22.0	182.0	70.0	Brazil	BRA	2016 Summer	2016	Summer	Rio de Janeiro	Football	Football Men's Football	Gold
2	Evandro Motta Guerra	M	34.0	207.0	106.0	Brazil	BRA	2016 Summer	2016	Summer	Rio de Janeiro	Volleyball	Volleyball Men's Volleyball	Gold
3	Luan Guilherme de Jesus Vieira	M	23.0	180.0	71.0	Brazil	BRA	2016 Summer	2016	Summer	Rio de Janeiro	Football	Football Men's Football	Gold
4	Luan Garcia Teixeira	M	23.0	183.0	79.0	Brazil	BRA	2016 Summer	2016	Summer	Rio de Janeiro	Football	Football Men's Football	Gold

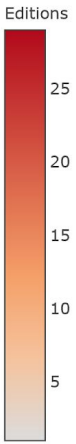
Introductory Exploratory Analysis and Visualisation

Countries' Participation in the Olympics

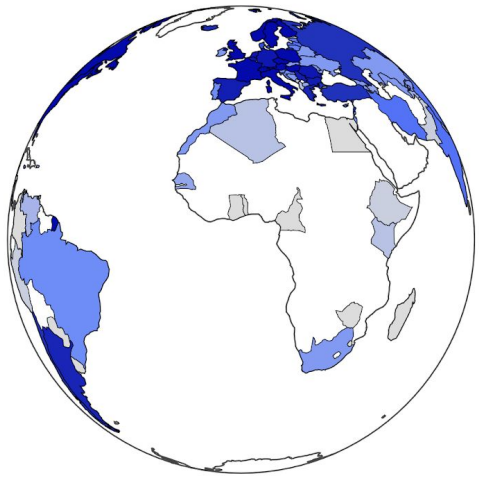


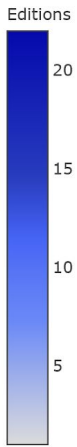
Olympic countries (Summer games)





Olympic countries (Winter games)



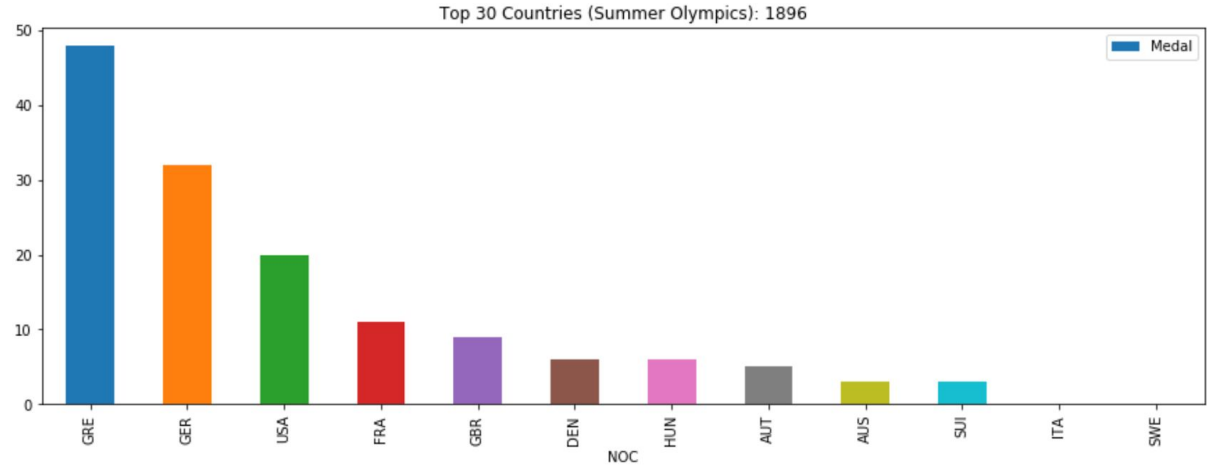


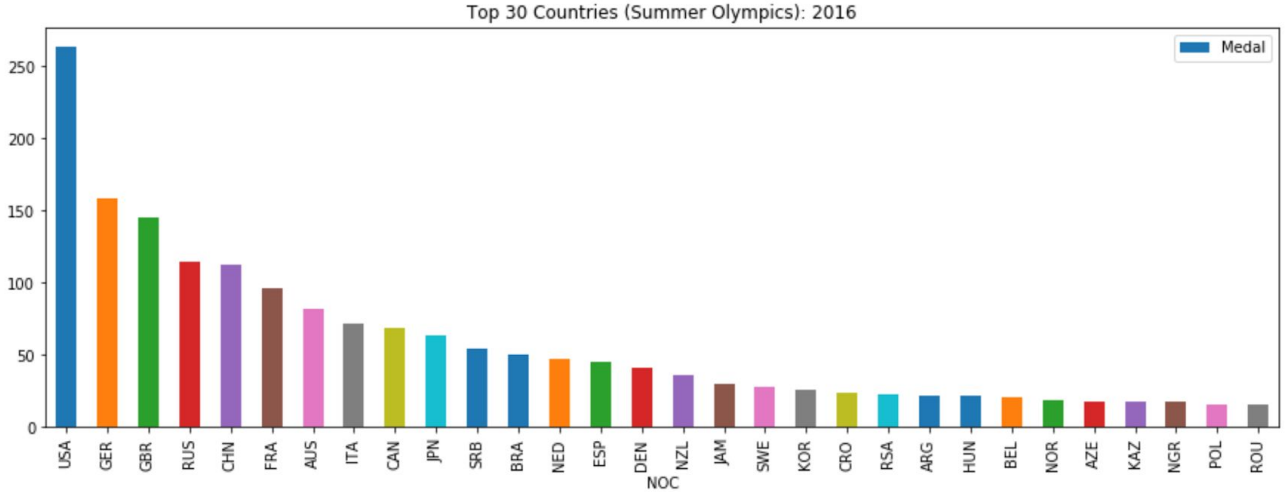
The winter olympics are much less popular and we find that warmer countries tend to participate less - which makes sense intuitively. The lack of participation can also be linked with the history of colonialism and political turmoil - which can be inferred by the overall 'Olympic Countries' map.

Country-Wise Wins:

Summer

Olympics

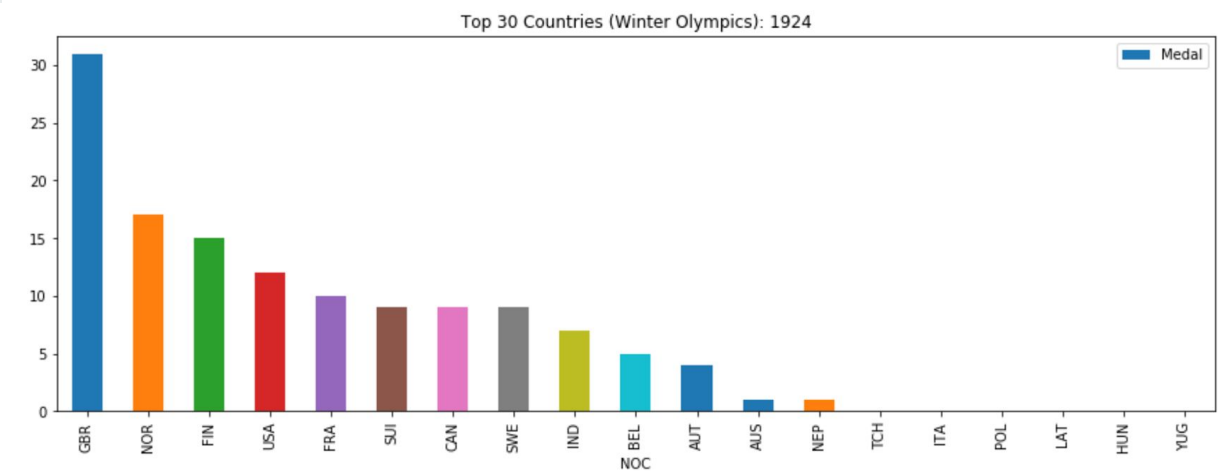


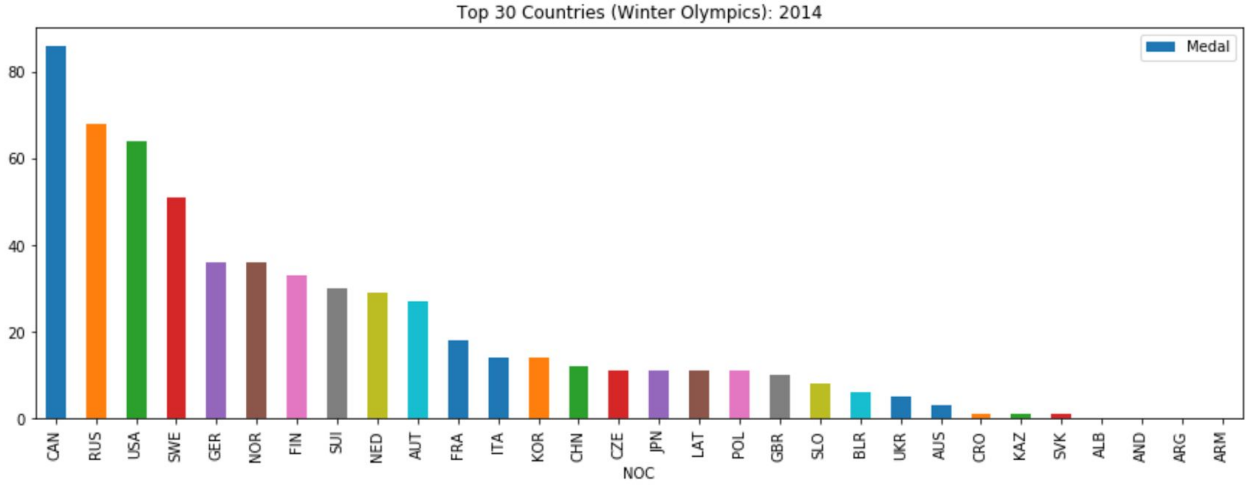


Country-Wise Wins:

Winter

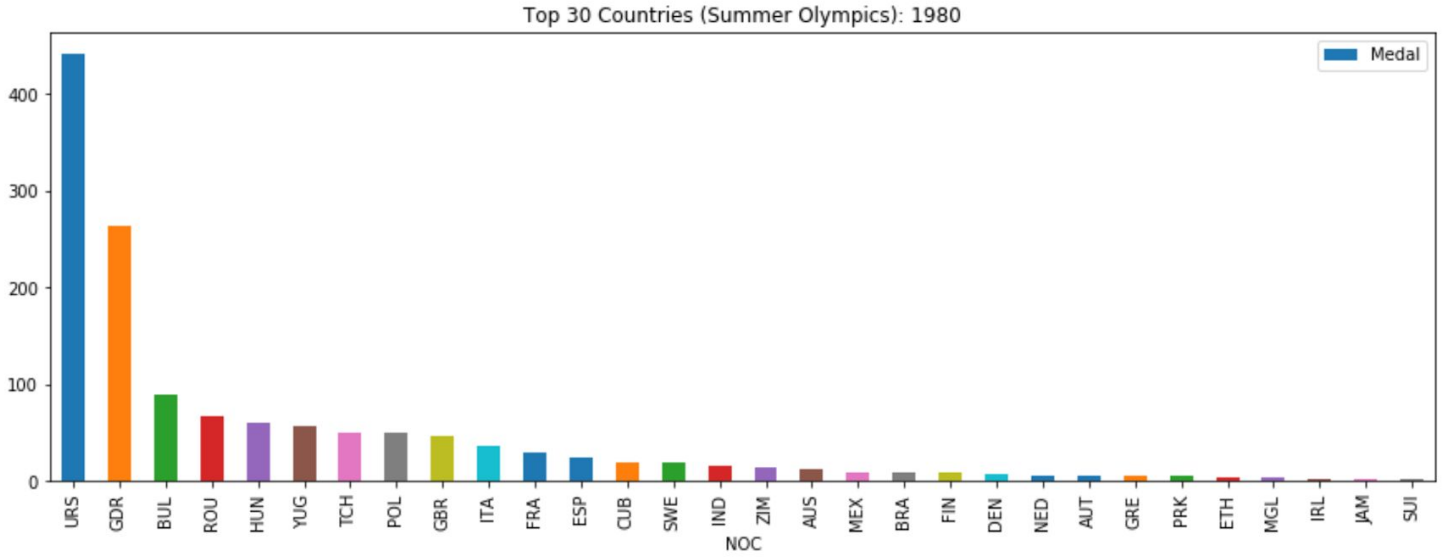
Olympics



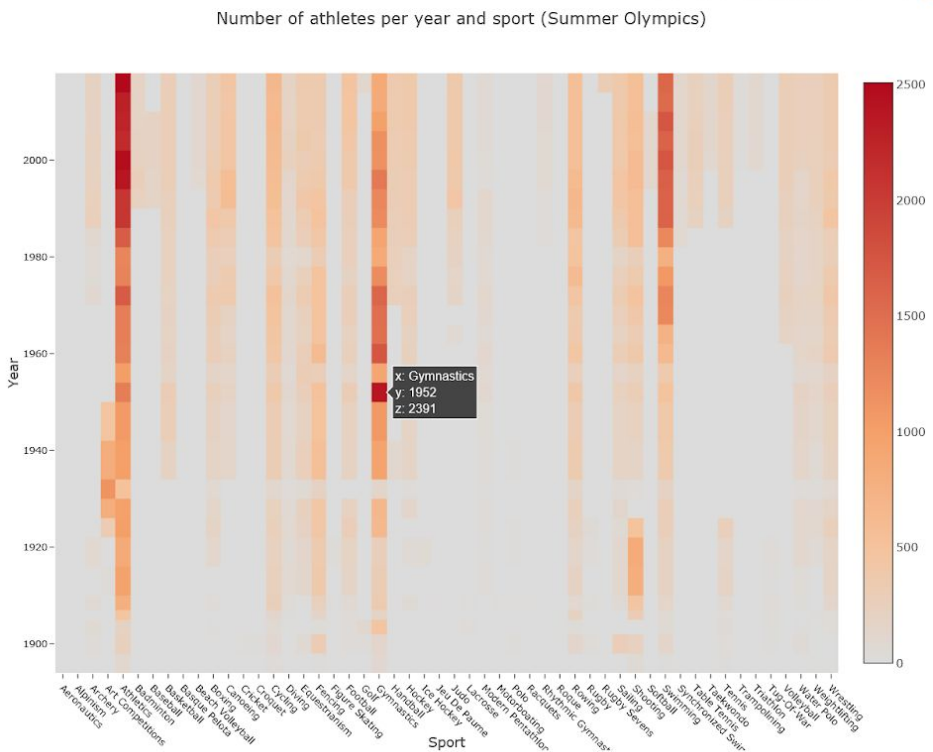


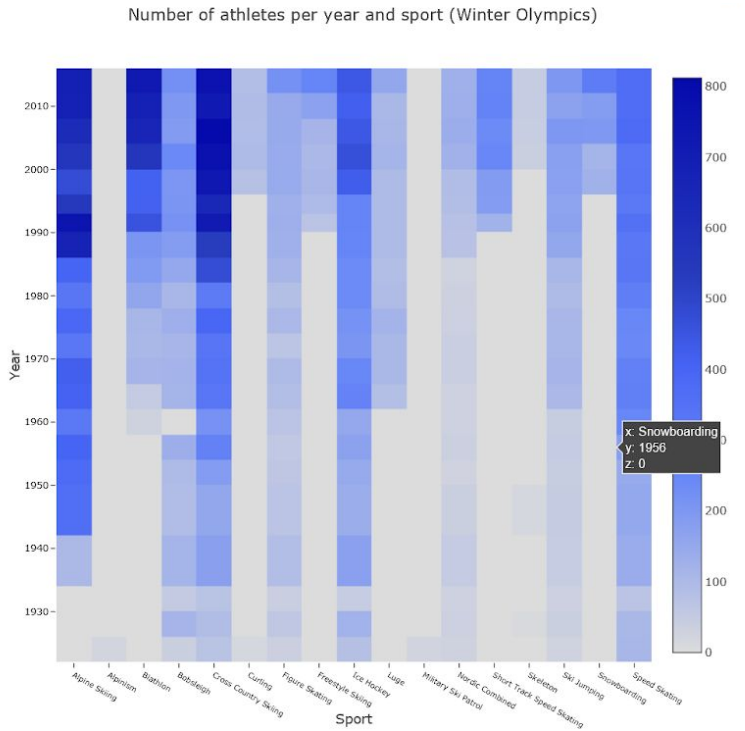
Anomaly

Years



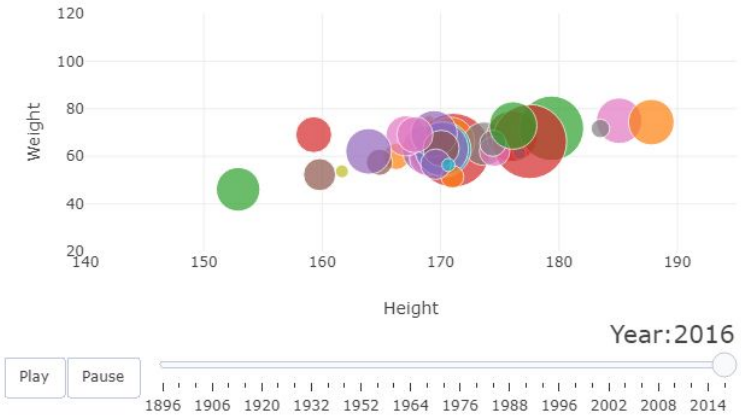
Athlete Participation



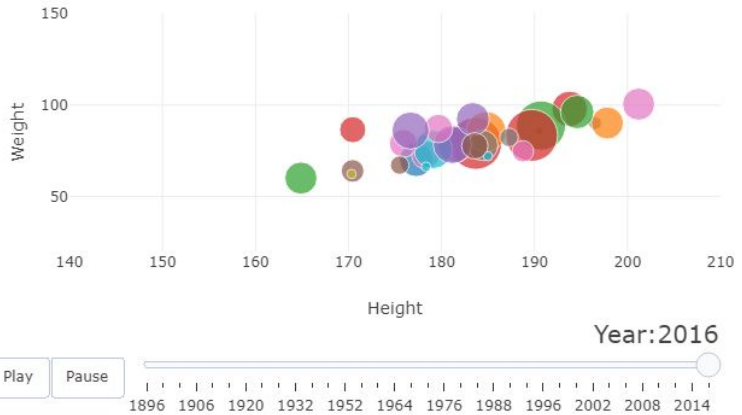


Athletes body measurements, grouped by Sex and Sport

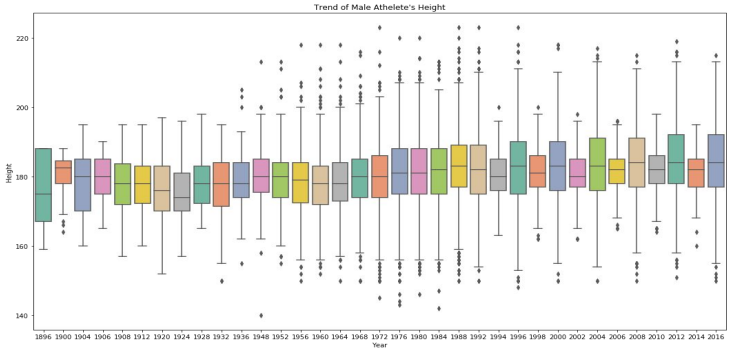
FEMALE

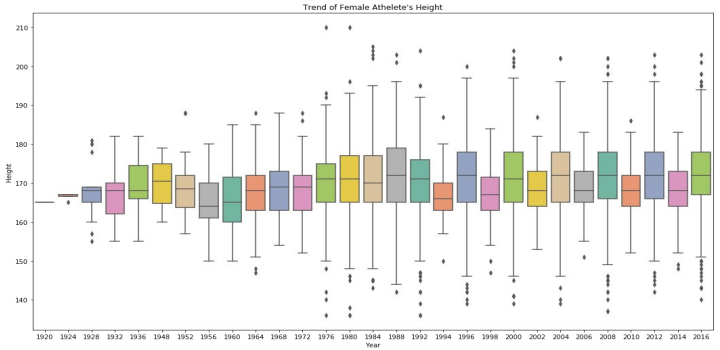


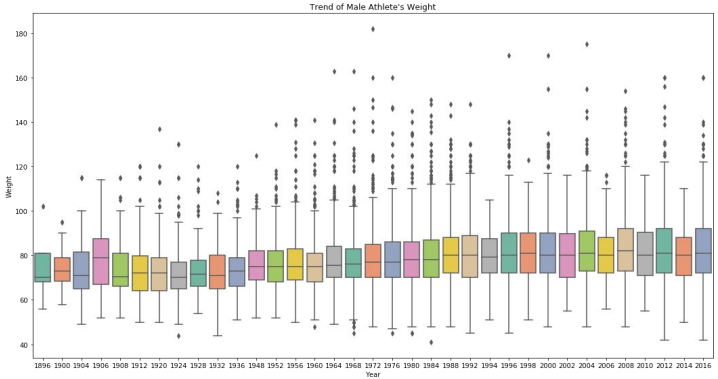
MALE

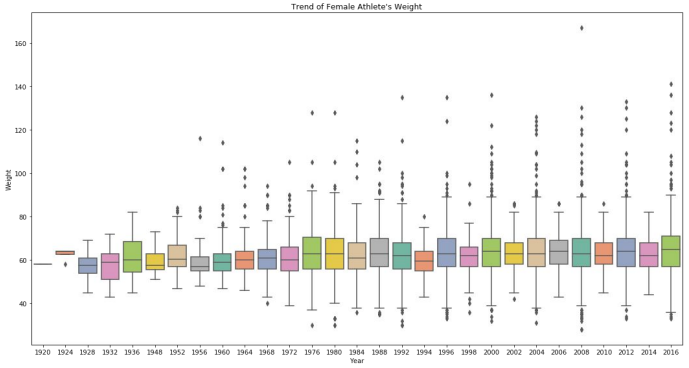


Boxplots for Height and Weight over the Years









Prediction

Random Forest Classifier

Predictors : Age 

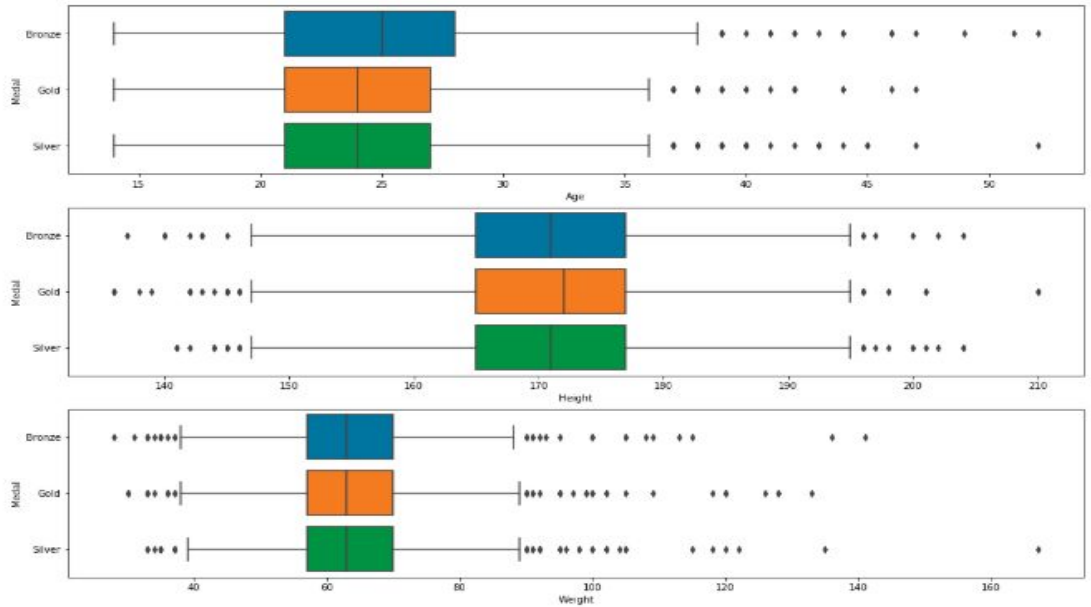
Height 

Weight 

Team 

Response : Medals

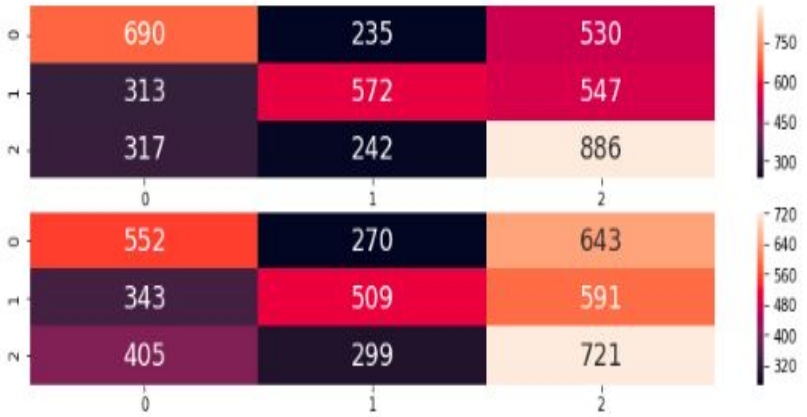
Relationship between Response and Predictors



Training and Goodness of Fit of Model

```
# Random Forest using Train Data
rforest = RandomForestClassifier(n_estimators = 500, max_depth = 5)  # create the object
rforest.fit(X_train, y_train.values.ravel())                         # Fit Random Forest on Train Data
```

```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 0.49584487534626037

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.41126240480036924
```



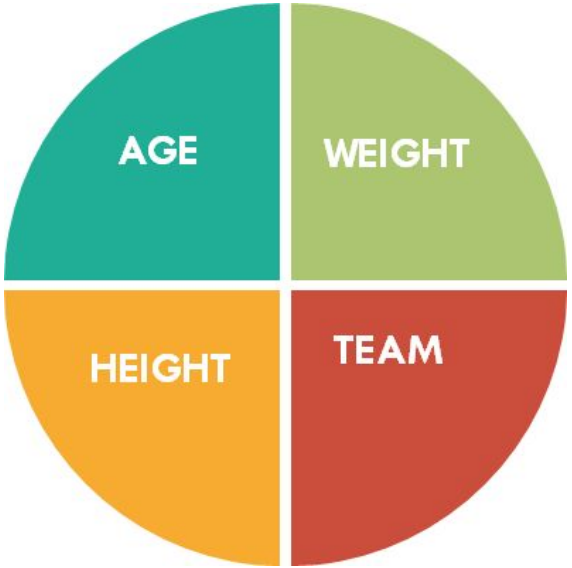
Try Predict Medal corresponding to Predictors

	Name	Age	Height	Weight	Medal	Predicted Medal	Result
0	Ann Kristin Aarnes	23.0	182.0	64.0	Bronze	Bronze	1.0
1	Patimat Abakarova	21.0	165.0	49.0	Bronze	Silver	0.0
2	Mariya Vasilyevna Abakumova (-Tarabina)	22.0	179.0	80.0	Silver	Gold	0.0
3	Tamila Rashidovna Abasova	21.0	163.0	60.0	Silver	Silver	1.0
4	Monica Cecilia Abbott	23.0	191.0	88.0	Silver	Gold	0.0
5	Nia Nicole Abdallah	20.0	175.0	56.0	Silver	Gold	0.0
6	Reema Abdo	21.0	173.0	59.0	Bronze	Bronze	1.0
7	Irene Abel	19.0	160.0	48.0	Silver	Bronze	0.0
8	Jennifer Abel	20.0	160.0	62.0	Bronze	Bronze	1.0
9	Elvan Abeylegesse	25.0	159.0	40.0	Silver	Silver	1.0
10	Nelli Mikhaylovna Abramova	24.0	171.0	60.0	Silver	Silver	1.0

```
Accuracy          : 45.27409117137911
```

Classification Tree

	Age	Height	Weight	Team	Medal	Bronze	Gold	Silver
10	41.0	175.0	75.0	64	Silver	0	0	1
11	21.0	172.0	78.0	19	Bronze	1	0	0
12	24.0	169.0	69.0	6	Silver	0	0	1
13	29.0	183.0	93.0	21	Gold	0	1	0
14	37.0	170.0	97.0	30	Silver	0	0	1
15	31.0	172.0	84.0	22	Bronze	1	0	0
16	37.0	163.0	82.0	62	Bronze	1	0	0



Predict

Assisting Functions

```
def GenerateSportsDF(sport, DF):
    '''Returns a sports DataFrame'''
```

```
def country_sport_recent(df, country, sport, gender, All_Medal_bool, Medal = None, ascending = True)
    '''Returns DataFrame of Country Performance SPORT in reverse chronological order'''
```

```
def GenerateGenderDF(sport, gender, DF = complete):
    ''' Returns DataFrame of  sport in a specific gender'''
```

```
def MedalDF_Sport(sport, gender = None, DF = complete):
    '''Returns DataFrame of sports of a specific gender'''
```

```
def MedalDF_Event(DF, Event, gender = None):
    '''Returns Data Frame of Specific gender in an event'''
```

```
def YearSort(df, ascending = True):
    '''Reorders DataFrame in Chronological order or Reverse chronological order'''
```

```
def WhichMedal(DF, medal = None):
    '''Returns DataFrame of specific Medal winners'''
```

```
def CountryPerformance(DF, Country, Medal = False, gender = None):
    '''Returns DataFrame of Country Performance'''
```

```
def Number_of_medals(df, country, event, gender, medal):
    '''Returns number of medals won by specific country in event'''
```

```
def annual_medal_sportlist(df, medal, country, season, year):
    '''Returns list of Sports played a country in specific year'''
```

```
def Number_of_medals_country_sport_year(df, country, sport, medal, year):
    '''Returns the integer number of medals won in a sport in an year'''
```

```
def Total_annual_medal(df, country, medal, season, year):
    '''Returns the total number of medals won by a country in an year'''
```

```
def TreeAccuracy(df, sport, thismedal, depth):
    '''Return the accuracy of Decision Tree'''
```

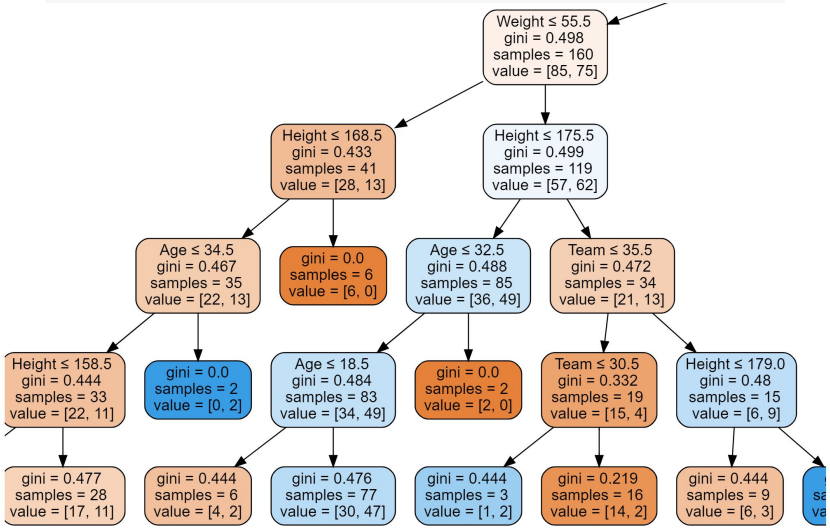
```
def TreeHeat(df, sport, thismedal, depth):
    '''Return the accuracy of Decision Tree'''
```

```
def GenerateAccuracyDF():
    '''Returns a DataFrame of sports with respective medal accuracies'''
```

```
def TreeDP(df, sport, thismedal, depth):
    '''Prints the Decision Tree'''
```

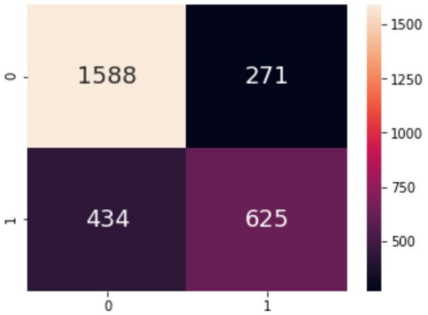
Decision Tree

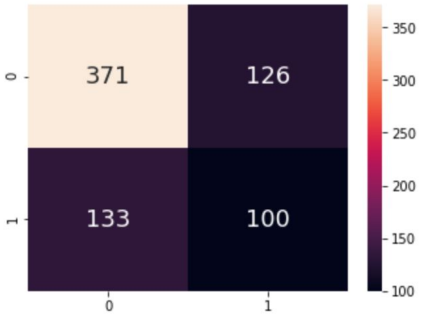
```
def TreeDP(df, sport, thismedal, depth):
    '''Prints the Decision Tree'''
test  = TreeDP(complete, "Athletics", "Gold", 10)
test
```



Heat Map

```
def TreeHeat(df, sport, thismedal, depth):
    '''Return the accuracy of Decision Tree'''
TreeHeat(df = complete, sport = "Athletics", thismedal = "Gold", depth = 10)
```





Accuracy

```
def TreeAccuracy(df, sport, thismedal, depth):
    '''Return the accuracy of Decision Tree'''
```

```
def GenerateAccuracyDF():
    '''Returns a DataFrame of sports with respective medal accuracies'''
```

```
test  = TreeDP(complete, "Athletics", "Gold", 10)
test
```

```
test = GenerateAccuracyDF()
test
```

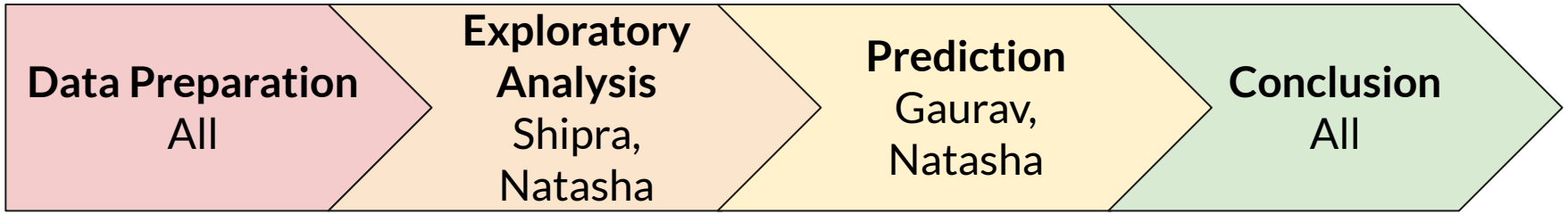
```
Classification Accuracy         : 0.7323509252912954

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.6520547945205479
```

	Sport	Gold Accuracy	Silver Accuracy	Bronze Accuracy
0	Basketball	83.50	79.50	79.00
1	Judo	71.70	67.92	52.83
2	Football	72.35	78.80	76.96
3	Tug-Of-War	100.00	100.00	100.00
4	Speed Skating	60.20	53.06	68.37
5	Cross Country Skiing	65.22	63.04	64.49
6	Athletics	64.11	64.93	65.75
7	Ice Hockey	77.78	65.52	73.56
8	Swimming	65.26	61.24	66.67
9	Badminton	51.61	58.06	64.52
10	Sailing	64.71	60.29	64.71

Conclusion

Accuracy for Classification Tree > Random Forest.
Therefore, classification tree is a better predictor for the medal won

Division of Work



Thank You!