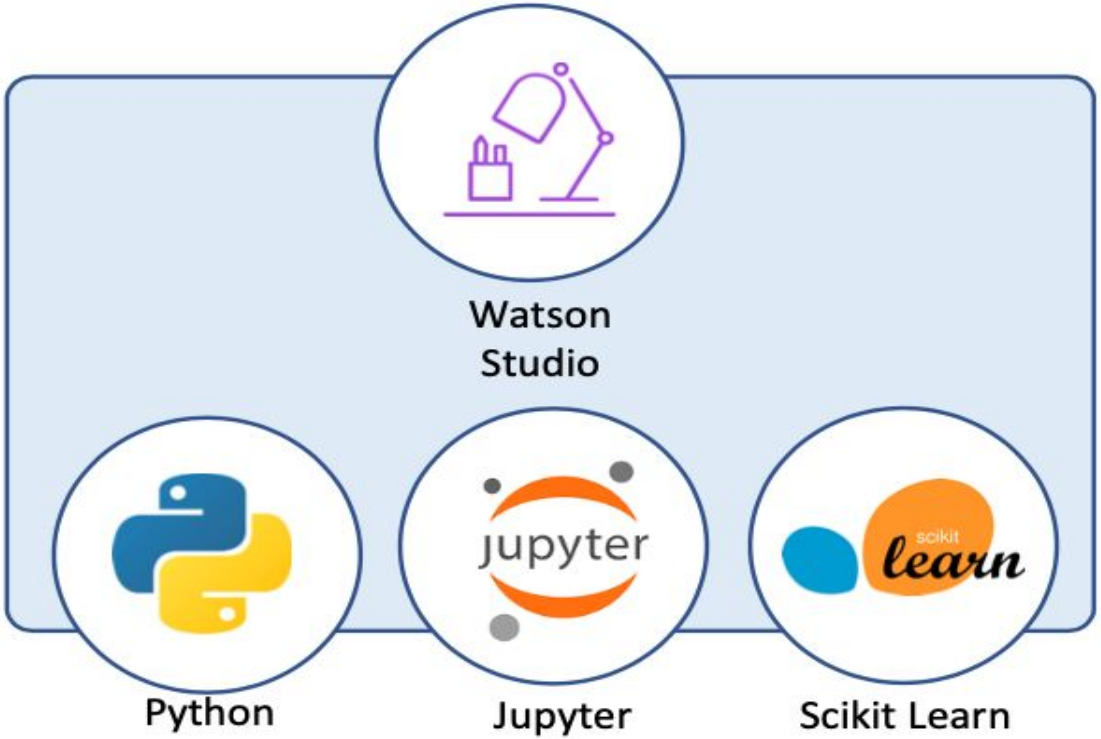
Predicting Athletes' Success based on Physical Built

by Samson Lo

for IBM Advanced Data Science Capstone

Architecture



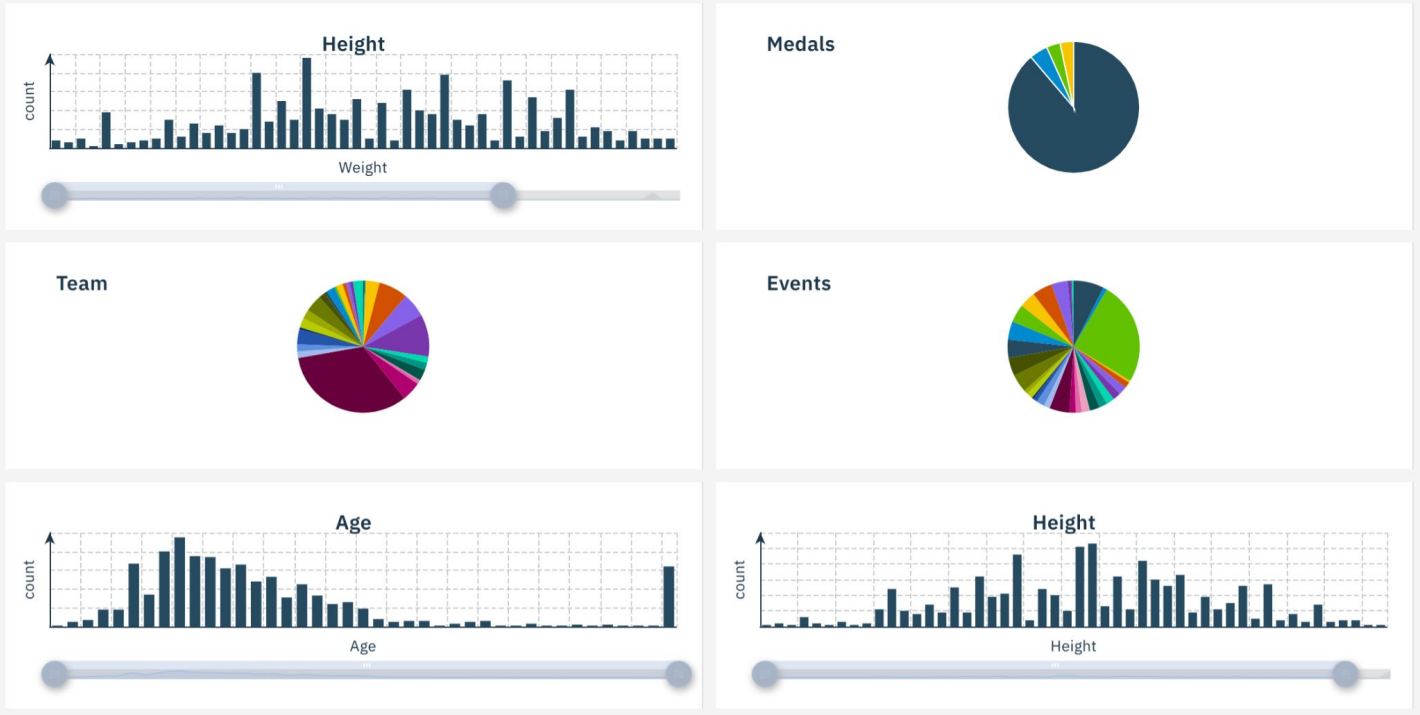
Technologies

- Python
- Jupyter
- Pandas
- Sklearn
- Matplotlib, Seaborn
- Keras
- IBM Watson

Dataset

Basic bio data on athletes and medal results from Athens 1896 to Rio 2016

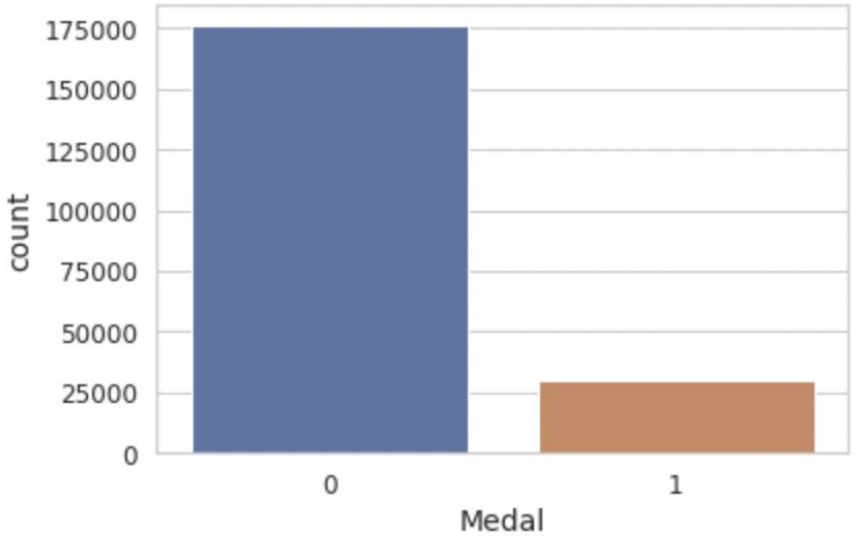
https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results



Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   ID      271116 non-null  int64
 1   Name    271116 non-null  object
 2   Sex     271116 non-null  object
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  object
 7   NOC     271116 non-null  object
 8   Games   271116 non-null  object
 9   Year    271116 non-null  int64
 10  Season  271116 non-null  object
 11  City    271116 non-null  object
 12  Sport   271116 non-null  object
 13  Event   271116 non-null  object
 14  Medal   39783 non-null   object
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

- Missing values in Height and Weight
- Mixed events & sex data
- Numerical features are skewed
- Prediction classed are imbalanced



Preprocessing

Data cleaning

Processing missing values

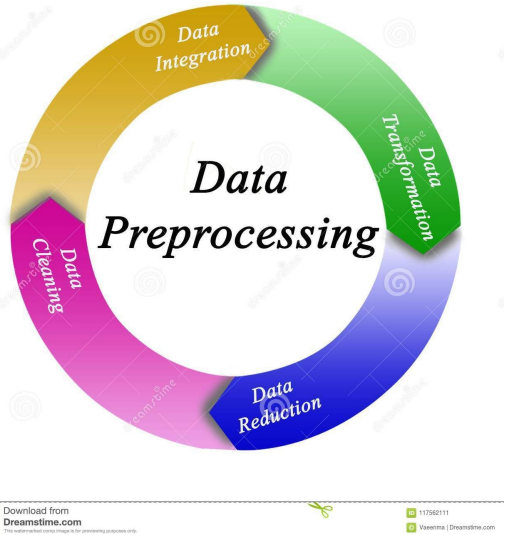
Transformation

Splitting

One-Hot encoding

Scaling

Uniting



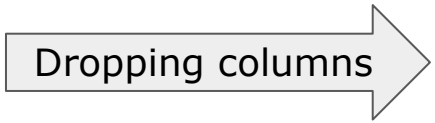
Data Cleaning

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   ID      271116 non-null  int64
 1   Name    271116 non-null  object
 2   Sex     271116 non-null  object
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  object
 7   NOC     271116 non-null  object
 8   Games   271116 non-null  object
 9   Year    271116 non-null  int64
 10  Season  271116 non-null  object
 11  City    271116 non-null  object
 12  Sport   271116 non-null  object
 13  Event   271116 non-null  object
 14  Medal   39783 non-null   object
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

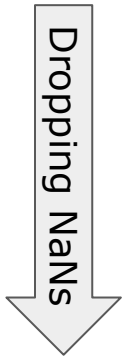
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 206165 entries, 0 to 206164
Data columns (total 8 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   ID      206165 non-null  int64
 1   Sex     206165 non-null  object
 2   Age     206165 non-null  float64
 3   Height  206165 non-null  float64
 4   Weight  206165 non-null  float64
 5   Team    206165 non-null  object
 6   Event   206165 non-null  object
 7   Medal   206165 non-null  int64
dtypes: float64(3), int64(2), object(3)
memory usage: 12.6+ MB
```

Dropping missing values

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   ID      271116 non-null  int64  
 1   Name    271116 non-null  object 
 2   Sex     271116 non-null  object 
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  object 
 7   NOC     271116 non-null  object 
 8   Games   271116 non-null  object 
 9   Year    271116 non-null  int64  
 10  Season  271116 non-null  object 
 11  City    271116 non-null  object 
 12  Sport   271116 non-null  object 
 13  Event   271116 non-null  object 
 14  Medal   39783 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

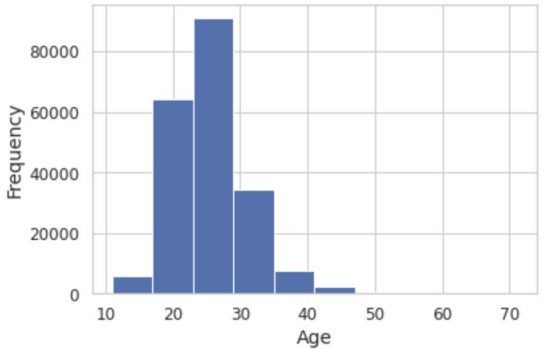


271116 rows

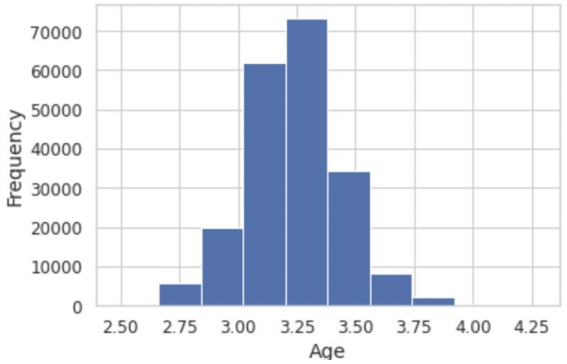


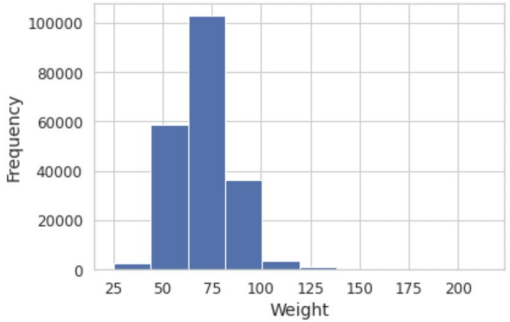
206165 rows

Transformation

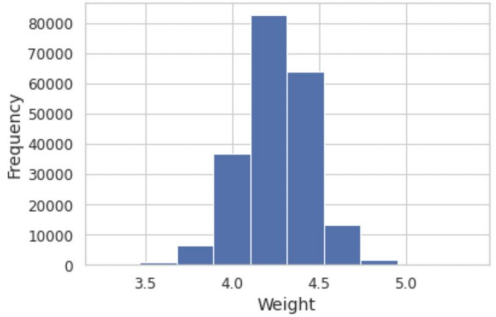


log(x+1)





log(x+1)



One-Hot Encoding

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 206165 entries, 0 to 206164
Data columns (total 6 columns):
 #   Column  Non-Null Count   Dtype   
---  ------  --------------   -----   
 0   Sex     206165 non-null  category
 1   Age     206165 non-null  float64 
 2   Height  206165 non-null  float64 
 3   Weight  206165 non-null  float64 
 4   Team    206165 non-null  category
 5   Event   206165 non-null  category
dtypes: category(3), float64(3)
memory usage: 5.8 MB
```

In [17]:
```
numeric_cols = ['Age', 'Height', 'Weight']
categorical_cols = list(set(df.columns.values.tolist()) - set(numeric_cols))
```

In [18]:
```
data_cat = df[categorical_cols]
data_num = df[numeric_cols]
enc = OneHotEncoder(sparse=False)
data_cat_oh = enc.fit_transform(data_cat)
data_cat_oh.shape
```

Out[18]:
```
(206165, 1252)
```

Train / Test Sets Splitting

```
(X_train_num, X_test_num, y_train, y_test) = train_test_split(data_num, y,
                                                              test_size=0.3,
                                                              random_state=42,
                                                              stratify=y)

(X_train_cat_oh, X_test_cat_oh) = train_test_split(data_cat_oh,
                                                   test_size=0.3,
                                                   random_state=42,
                                                   stratify=y)
```

Scaling

```
scaler = StandardScaler()

X_train_num_scaled = scaler.fit_transform(X_train_num, y_train)
X_test_num_scaled = scaler.transform(X_test_num)
```

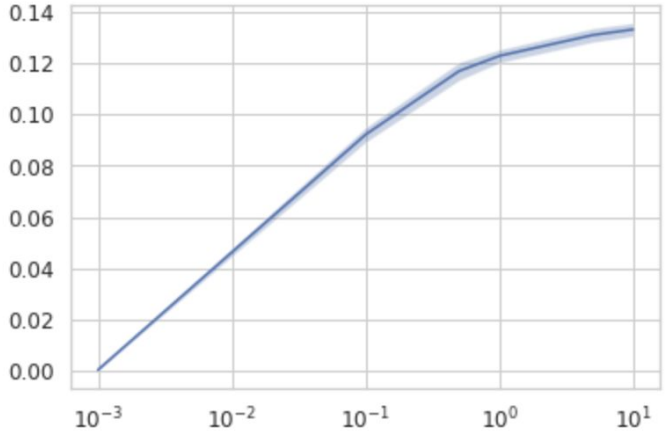
Uniting

```
train_data = np.hstack((X_train_num_scaled, X_train_cat_oh))
test_data = np.hstack((X_test_num_scaled, X_test_cat_oh))
```

Model Evaluation

Algorithm	Accuracy
Logistic Regression	86.03%
Sequential NN	85.36%

```
In [10]: plot_scores(optimizer_zeros)
```



Iterations

Iterations	Train	Test
Default	0.8543	0.8536
Normalization	0.8542	0.8536
Dropout	0.8545	0.8536

Difficulty Encountered

Dead Kernel 

- Potential cause:
  - Huge amount of data (esp. after transformation)
- Solution:
  - Opened another notebook

Used up free tier resources

- Potential cause:
  - Same as above
- Solution:
  - Upgraded to pay-as-you-go