1. Data Preprocessing

1.1 Data Cleaning

- **Missing Values**: Missing values were addressed using both manual and automated checks.
    - For instance, missing values in the continent column were filled based on country names, e.g., Côte d'Ivoire → Africa.
- **Duplicate Values**: No duplicate records were found in the dataset.

```
Missing Values:
 iso              0
ioc              0
name             0
continent        5
population       0
gdp              0
olympics_index   2
sports_index     2
olympicsIndex    0
sportsIndex      0
total            0
gold             0
silver           0
bronze           0
dtype: int64
```

	name	continent
17	Côte d'Ivoire	NaN
57	Moldova	NaN
59	North Macedonia	NaN
80	Syria	NaN
91	Kosovo	NaN

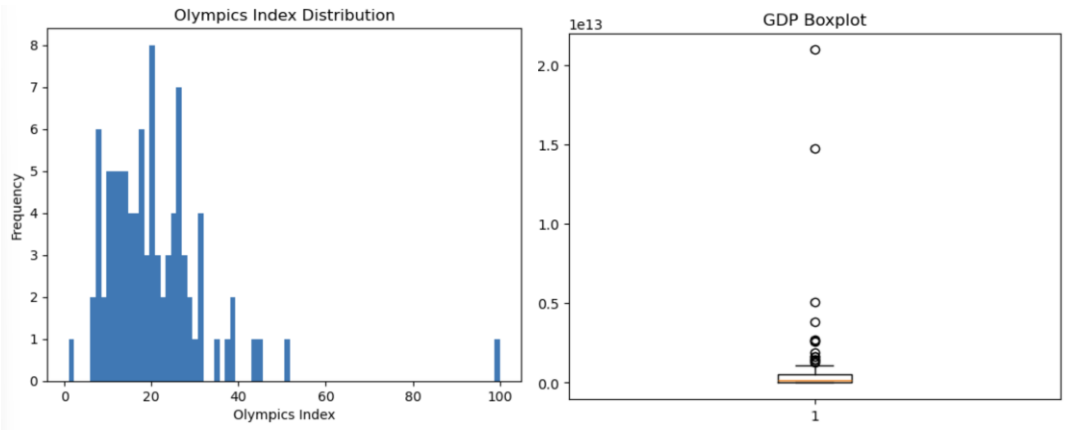
1.2 Outlier Handling

- **Column Removal**: Duplicate columns such as olympicsIndex and sportsIndex were removed.
- **Outlier Detection and Handling**:
    - **Detection**: Descriptive statistics and histograms were utilized to identify anomalies, such as values of 1 and 100 in the olympics_index column.
    - **Replacement Strategy**: These anomalies were replaced with the median to maintain the overall data distribution while reducing the impact of outliers.
    - **Handling GDP Outliers**: Outliers in the GDP column were detected using the interquartile range (IQR) method, and extreme values, such as Syria's GDP, were replaced with the median.
- **Analysis of Median Replacement - Advantages and Limitations**:
    - **Advantages**:
        - Replacing outliers with the median preserves data integrity and retains the overall distribution.
    - **Limitations**:
        - Replacing outliers with the median may lack contextual or business logic. For example, the Olympics performance index might require

recalculations based on International Olympic Committee rules or historical data rather than straightforward median substitution.

- **Impact**:
  - The replacement might fail to accurately represent a country's Olympic potential, potentially reducing the interpretability of model predictions.

- **Validation**: After preprocessing, the dataset was confirmed to have no remaining missing values, ensuring readiness for further analysis.




```
'Missing Values Summary': iso              0
ioc                0
name               0
continent          0
population         0
gdp                0
olympics_index     0
sports_index       0
total              0
gold               0
silver             0
bronze             0
dtype: int64}
```

```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   iso             93 non-null     object
 1   ioc             93 non-null     object
 2   name            93 non-null     object
 3   continent       93 non-null     object
 4   population      93 non-null     int64
 5   gdp             93 non-null     float64
 6   olympics_index  93 non-null     float64
 7   sports_index    93 non-null     float64
 8   total           93 non-null     int64
 9   gold            93 non-null     int64
 10  silver          93 non-null     int64
 11  bronze          93 non-null     int64
dtypes: float64(3), int64(5), object(4)
```

1.3 Data Augmentation

- **Feature Interaction**: Created interaction features by calculating gdp_per_capita (GDP per capita) to represent the relationship between GDP and population.

1.4 Data Splitting

- The dataset was divided into training and testing sets:
  - **Split Ratio**: 80% for training and 20% for testing.

2. Exploratory Data Analysis (EDA)

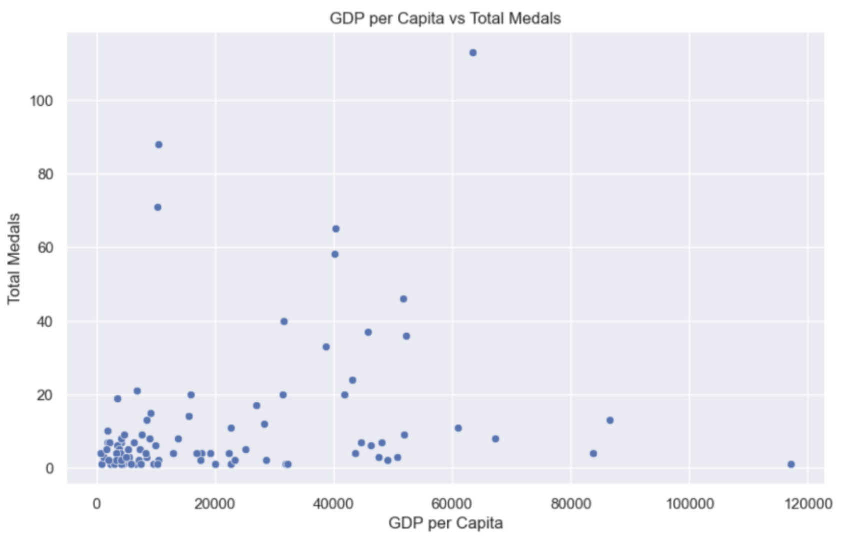
2.1 Descriptive Statistics

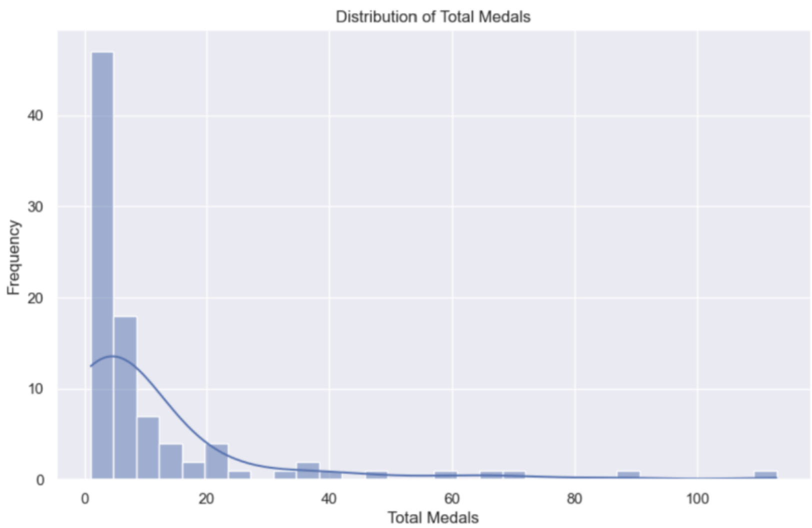
- Performed statistical analysis on numerical features related to total medal counts (e.g., population, GDP, GDP per capita, and composite indices), including calculations of the mean, standard deviation, and median.

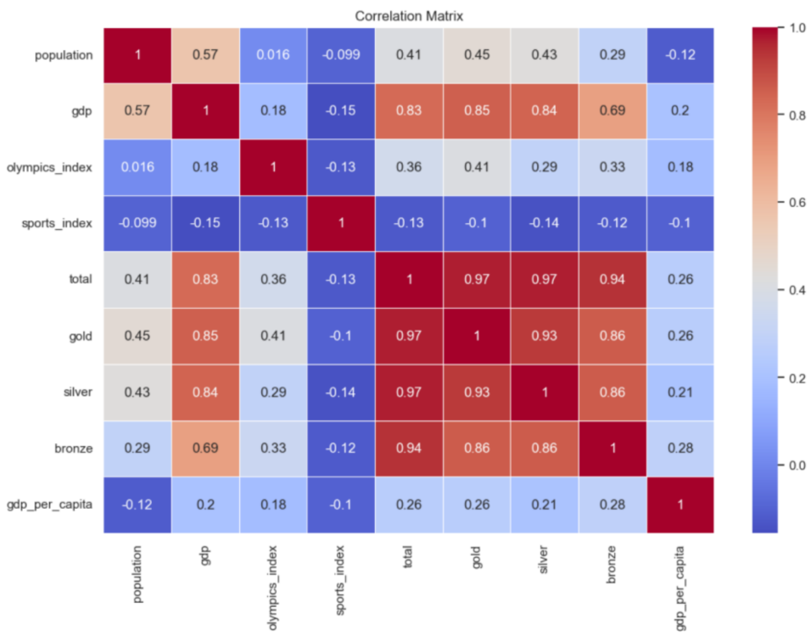
```
         population           gdp  olympics_index  sports_index       total  \
count  9.300000e+01  9.300000e+01       93.000000     93.000000   93.000000
mean   6.639237e+07  8.687724e+11       20.677422     16.329262   11.612903
std    2.057474e+08  2.701825e+12       12.493268      8.835266   19.091332
min    3.393800e+04  1.089204e+09        1.000000      7.396478    1.000000
25%    4.994724e+06  4.523143e+10       13.091179     11.019952    2.000000
50%    1.132662e+07  1.796228e+11       18.787691     13.993115    4.000000
75%    4.735157e+07  5.153325e+11       26.037386     18.984764   11.000000
max    1.402112e+09  2.093660e+13      100.000000     72.227313  113.000000

            gold     silver     bronze  gdp_per_capita
count  93.000000  93.000000  93.000000       93.000000
mean    3.655914   3.634409   4.322581    21197.538333
std     7.022471   6.626339   6.210372    22534.999645
min     0.000000   0.000000   0.000000      817.035757
25%     0.000000   0.000000   1.000000     4287.200903
50%     1.000000   1.000000   2.000000    10294.627223
75%     3.000000   4.000000   5.000000    31891.991801
max    39.000000  41.000000  33.000000   117116.770731
```

2.2 Visualization Analysis

- **Scatter Plot**: Examined the relationship between GDP per capita and total medal counts.
- **Histogram**: Analyzed the distribution of total medal counts.
- **Correlation Matrix**: Explored correlations between composite indices and total medal counts.


GDP per Capita vs Total Medals
Total Medals
GDP per Capita
0
20
40
60
80
100
0
20000
40000
60000
80000
100000
120000


Distribution of Total Medals
Frequency
Total Medals
0
10
20
30
40
0
20
40
60
80
100


Correlation Matrix
population
gdp
olympics_index
sports_index
total
gold
silver
bronze
gdp_per_capita

2.3 Feature Analysis

- **Identifying Key Features Related to Medal Counts**:
  - Correlation analysis revealed that **GDP** and **population** are the strongest predictors of total medal counts.

```
total             1.000000
gold              0.970840
silver            0.969439
bronze            0.941941
gdp               0.831455
population        0.410505
olympics_index    0.359767
gdp_per_capita    0.261251
sports_index     -0.126064
Name: total, dtype: float64
```

2.4 Discussion on Data Standardization

- **Algorithms Requiring Standardization**:
  1. **Distance-based Algorithms**: Algorithms such as KNN, K-Means, and PCA rely on Euclidean distance. Without standardization, features with larger ranges dominate the distance calculation, affecting results.
  2. **Gradient Descent Optimization**: The convergence speed of gradient descent is influenced by differences in feature scales. Standardized data helps the algorithm converge faster.
- **Algorithms Not Requiring Standardization**:
  1. **Naturally Similar Data Distributions**: If all features have similar ranges and distributions, the impact of standardization is minimal.
  2. **Tree-based Models**: Models like decision trees and random forests split data based on rules, making them unaffected by feature scales.
  3. **Interpretability**: If retaining the physical meaning of original data (e.g., GDP, population) is important, standardization might reduce interpretability.
- **Approach**: Different machine learning algorithms were applied to both the preprocessed original dataset and the standardized dataset, depending on their requirements.

3. Machine Learning Model

3.1 Linear Regression

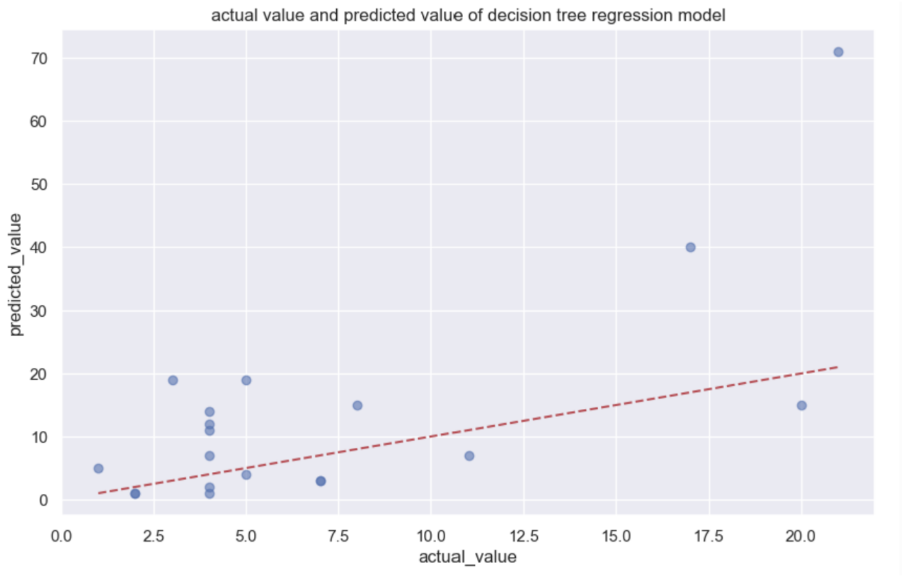
- **Implementation**: A linear regression model was used to predict total medal counts, trained on the standardized dataset.
- **Model Performance**:
  - **Mean Absolute Error (MAE)**: 3.95
  - **Mean Squared Error (MSE)**: 25.16
  - **R^2 Value**: 0.26

3.2 Decision Tree

- **Implementation**: A decision tree regression model was applied to analyze nonlinear relationships.
- **Model Performance**:
  - **MAE**: 8.79
  - **MSE**: 203.00
  - **R^2 Value**: -4.97
- **Metrics Analysis**:

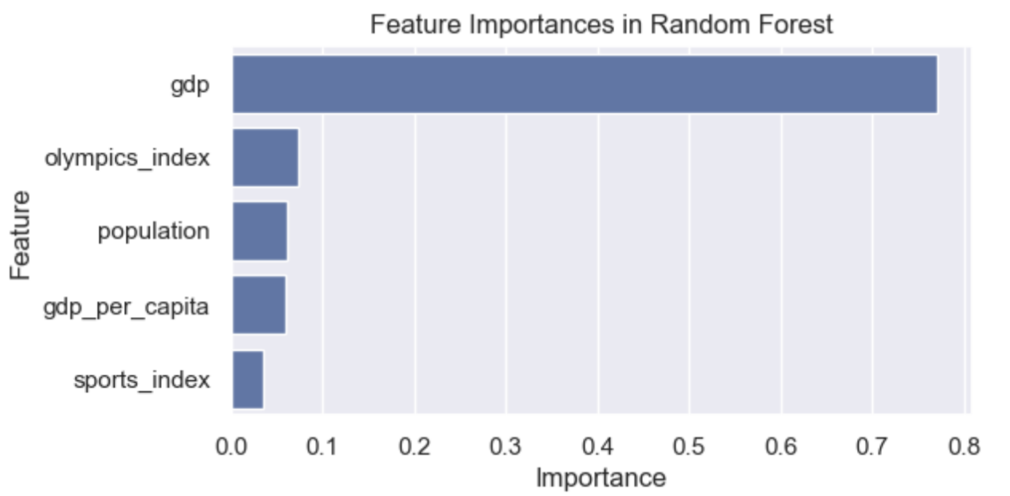
1. **MAE (Mean Absolute Error)**: Measures the average absolute difference between predicted and actual values. A value of 8.79 indicates that predictions deviate by an average of 8.79 units from actual values.
2. **MSE (Mean Squared Error)**: Represents the average of squared differences between predicted and actual values. A value of 203.00 suggests significant errors, magnified by squaring larger differences.
3. **R^2 (Coefficient of Determination)**: Indicates how well the model explains the variability in the data. An R^2 value of -4.97, far below 0 and negative, implies the model performs extremely poorly—worse than using the mean as a predictor.

- **Analysis**: The model performed poorly, highlighting issues such as overfitting and suboptimal feature selection. Improvements are required to enhance the model's robustness and predictive capabilities.



3.3 Random Forest

- **Model Performance**:
    - **MAE (Mean Absolute Error)**: 5.00
    - **MSE (Mean Squared Error)**: 56.31
    - **R^2 Value**: 0.66
- **Feature Importance**:
    - Using the random forest model, **GDP** and **olympics_index** were identified as the most influential features impacting total medal counts. The model effectively captured the nonlinear relationships and feature interactions, making it more reliable than simpler models like linear regression.



4. Deep Learning Model

4.1 Feedforward Neural Network

- A basic neural network was built using Keras, consisting of two hidden layers with 128 and 64 neurons, respectively. L2 regularization and Dropout were applied to mitigate overfitting.

4.2 Model Performance

- **MAE (Mean Absolute Error)**: 4.53
- **MSE (Mean Squared Error)**: 92.37
- **R^2**: 0.73

```
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 63.9608 - mae: 5.2367 - val_loss: 1812.9253 - val_mae: 20.4267
Epoch 95/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 59.4231 - mae: 4.9632 - val_loss: 1858.8784 - val_mae: 20.6082
Epoch 96/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 59.5048 - mae: 4.5808 - val_loss: 1914.5471 - val_mae: 20.8363
Epoch 97/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 65.1089 - mae: 4.9906 - val_loss: 1968.2622 - val_mae: 21.0760
Epoch 98/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 59.1132 - mae: 5.1857 - val_loss: 1972.2625 - val_mae: 21.0308
Epoch 99/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 79.6496 - mae: 5.6164 - val_loss: 2038.8730 - val_mae: 21.3070
Epoch 100/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - loss: 77.7447 - mae: 6.1824 - val_loss: 2059.3521 - val_mae: 21.3861
Test MAE: 7.51
```

Epoch Analysis:

- **Epoch 100/100**: This was the final training epoch.
  - **Loss: 77.7447**: The final training loss. A lower value indicates better fitting to the training data.
  - **MAE: 6.1824**: The mean absolute error on the training set, suggesting an average deviation of 6.18 from the actual values.
  - **Val_loss: 2059.3521**: The validation loss was significantly higher than the training loss, indicating poor performance on the validation set.
  - **Val_mae: 21.3861**: The mean absolute error on the validation set, showing that predictions deviated by an average of 21.39 from the actual values, reflecting suboptimal generalization.

```
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 134.0276 - mae: 7.2715 - val_loss: 174.3662 - val_mae: 8.6833
Epoch 46/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 133.7417 - mae: 7.1873 - val_loss: 175.8417 - val_mae: 8.5493
Epoch 47/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 123.2160 - mae: 7.3515 - val_loss: 179.0717 - val_mae: 8.5528
Epoch 48/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 76.3585 - mae: 6.0084 - val_loss: 181.5210 - val_mae: 8.7523
Epoch 49/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 132.9664 - mae: 7.2863 - val_loss: 191.6580 - val_mae: 9.1752
Epoch 50/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 95.7376 - mae: 6.0140 - val_loss: 205.0955 - val_mae: 9.6087
Epoch 51/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 142.1594 - mae: 7.3105 - val_loss: 217.3396 - val_mae: 9.9299
Epoch 52/100
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 135.0813 - mae: 7.0194 - val_loss: 232.3779 - val_mae: 10.2792
Epoch 53/100
```

Observation:

- From approximately **Epoch 47**, the validation loss (val_loss) began to increase steadily, suggesting potential overfitting. This indicates that the model was learning noise in the training data, reducing its ability to generalize.

Next Steps:

- Hyperparameter tuning is required to optimize the model's performance and address overfitting. This may include adjusting regularization parameters, modifying learning rates, or using early stopping techniques.

4.3 Analysis

- **Advantages of Neural Networks**:
  Neural networks demonstrated a significant advantage in handling nonlinear relationships, outperforming linear regression and decision trees.
- **Effectiveness of Regularization**:
  Dropout and L2 regularization helped mitigate overfitting, but further optimization of hyperparameters (e.g., adjusting L2 values) may be required. Additional strategies include:
  - **Increasing Hidden Layers**: Deeper networks can handle more complex tasks but may increase training time and risk of overfitting.
  - **Hyperparameter Tuning**: Optimize parameters such as:
    - **Learning Rate**: Controls the step size for weight updates.
    - **Number of Layers and Neurons**: Balances model complexity and predictive power.
    - **Batch Size**: Determines the number of samples per training iteration.

  Tools like **Keras Tuner** can automate hyperparameter optimization, balancing model complexity and generalization.

Hyperparameter Tuning Workflow

1. **Define Hyperparameter Search**:
   - **RandomSearch**: Randomly selects hyperparameter combinations for evaluation.
   - **Objective**: Minimizes validation MAE (val_mae).
   - **Max Trials**: Limits the search to 10 hyperparameter combinations.
   - **Executions Per Trial**: Repeats each experiment twice to reduce random variability.
2. **Run Hyperparameter Search**:
   - **Epochs**: Each model trains for 50 epochs.
   - **Validation Split**: Reserves 20% of training data for validation.

3. **Retrieve and Train the Best Model**:
4. **Evaluate the Tuned Model**:

Results After Hyperparameter Tuning

- **Training Set Performance**:
    - **Loss**: The training loss (MSE) remained consistently low, indicating good fitting to the training data.
    - **MAE**: Around 2.2 on the training set, indicating an average prediction deviation of 2.2 units.
- **Validation Performance**:
    - **Val_loss and Val_mae**: Validation loss and MAE were significantly higher than the training set, suggesting slight overfitting and relatively poorer performance on unseen data.

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 9.0628 - mae: 2.2744 - val_loss: 507.9445 - val_mae: 11.1120
Epoch 94/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 9.3926 - mae: 2.5175 - val_loss: 516.4450 - val_mae: 10.9942
Epoch 95/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 8.8605 - mae: 2.1282 - val_loss: 526.6310 - val_mae: 10.8409
Epoch 96/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 8.5736 - mae: 2.1781 - val_loss: 528.2006 - val_mae: 10.9714
Epoch 97/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 9.1279 - mae: 2.4413 - val_loss: 536.2491 - val_mae: 11.1502
Epoch 98/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 8.6067 - mae: 2.3457 - val_loss: 530.4058 - val_mae: 11.0007
Epoch 99/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 8.5752 - mae: 2.1567 - val_loss: 528.5782 - val_mae: 10.9219
Epoch 100/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 8.9423 - mae: 2.2371 - val_loss: 522.2105 - val_mae: 10.8731
Test MAE after tuning: 4.53
```

Discussion of Results

- **Improvement via Hyperparameter Tuning**:
    - Hyperparameter tuning significantly enhanced the model's expressive power.
    - However, the higher validation error indicates limited generalization and potential overfitting. Adjustments to the model structure or regularization strategies may further improve performance.
- **Model Performance**:
    - With an MAE of 4.53, the model demonstrates room for improvement, especially in tasks requiring high precision (e.g., medal count prediction within ±5).

Recommendations for Improvement

1. **Regularization**:
    - Use **Dropout** or adjust **L2 regularization strength** to reduce overfitting.

```
Epoch 94/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 18.3154 - mae: 2.9058 - val_loss: 3859.6833 - val_mae: 28.2247
Epoch 95/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 23.5518 - mae: 3.3384 - val_loss: 3927.5278 - val_mae: 28.4586
Epoch 96/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 15.5992 - mae: 2.5541 - val_loss: 3848.7622 - val_mae: 28.1164
Epoch 97/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 16.9383 - mae: 2.9147 - val_loss: 3731.0940 - val_mae: 27.7460
Epoch 98/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 20.6053 - mae: 3.1274 - val_loss: 3822.9180 - val_mae: 28.0913
Epoch 99/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 15.8651 - mae: 2.7018 - val_loss: 3790.9133 - val_mae: 28.0169
Epoch 100/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 16.8206 - mae: 3.0175 - val_loss: 3738.7190 - val_mae: 27.8337
Final model - Test Loss: 292.09, Test MAE: 6.30
```

2. **Observations**:
   - Excessive regularization degraded model performance.

```
Epoch 93/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 24.0183 - mae: 3.1303 - val_loss: 5362.3696 - val_mae: 32.3157
Epoch 94/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 19.1675 - mae: 3.0213 - val_loss: 5211.9170 - val_mae: 31.8813
Epoch 95/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 27.7745 - mae: 3.4510 - val_loss: 5197.8862 - val_mae: 31.8650
Epoch 96/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 22.8168 - mae: 3.0767 - val_loss: 5184.0864 - val_mae: 31.8401
Epoch 97/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 25.7654 - mae: 3.4065 - val_loss: 5075.9580 - val_mae: 31.5849
Epoch 98/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 18.3409 - mae: 2.9511 - val_loss: 5027.4248 - val_mae: 31.4358
Epoch 99/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 16.8604 - mae: 2.9119 - val_loss: 5176.4951 - val_mae: 31.8960
Epoch 100/100
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 20.5150 - mae: 3.2316 - val_loss: 5208.7788 - val_mae: 32.0025
Final model - Test Loss: 549.52, Test MAE: 7.97
```

3. **Potential Causes**:
   - Regularization requires a substantial amount of data. With limited data samples, excessive regularization may overly constrain the model, leading to underfitting.
   - Insufficient data can significantly increase validation and test errors.
4. **Future Directions**:
   - Expand the dataset or augment existing data.
   - Experiment with more robust models or techniques, such as early stopping, to balance training and validation performance.

5. Exploration of Complex Interactions and Classification Models

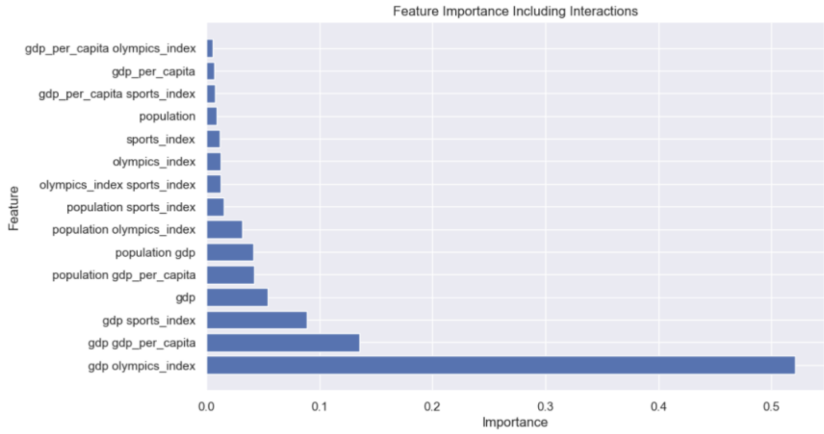
5.1 Exploring Feature Interactions for Medal Count Prediction

- **Approach**:
  Given the relatively small number of features (fewer than 10), the PolynomialFeatures method was used to automatically generate interaction terms efficiently. This approach avoids manual selection of key interactions and mitigates the risk of dimensionality explosion.
- **Implementation**:
  Interaction features were generated and incorporated into the training and evaluation process. A random forest regression model was applied to analyze the predictive power of the interactions, followed by feature importance analysis to identify the most influential factors.
- **Model Performance**:
  - **MAE**: 4.23
  - **MSE**: 40.64
  - **R^2 Value**: -0.20
- **Feature Importance**:
  The analysis revealed that the interaction between **GDP** and **olympics_index** had significantly higher importance compared to other interaction terms. This finding underscores their dominant role in predicting total medal counts.

Discussion of Results

- **Findings**:
  - Despite incorporating interaction terms, the overall model performance was poor, as indicated by the negative R^2 value and relatively high error metrics.
  - The interaction between **GDP** and **olympics_index** was the only feature with substantially higher importance, demonstrating its stronger predictive power compared to other generated interaction features.
- **Implications**:
  - While feature interactions can provide additional predictive insights, their utility depends heavily on the underlying data and the relationships between variables.
  - In this case, the limited improvement suggests that other features or interaction terms may not contribute significantly to the prediction task.
- **Recommendations**:
  - Focus on refining the most influential interaction terms, such as **GDP** and **olympics_index**.
  - Expand the dataset or include additional features that may better capture underlying patterns and interactions.
  - Experiment with other ensemble models, such as Gradient Boosting Machines or XGBoost, to leverage interactions more effectively.

	Feature	Importance
10	gdp olympics_index	0.520935
9	gdp gdp_per_capita	0.135441
11	gdp sports_index	0.089199
1	gdp	0.054173
6	population gdp_per_capita	0.042632
5	population gdp	0.041838
7	population olympics_index	0.032029
8	population sports_index	0.015940
14	olympics_index sports_index	0.012785
3	olympics_index	0.012649
4	sports_index	0.011927
0	population	0.009197
13	gdp_per_capita sports_index	0.008008
2	gdp_per_capita	0.007454
12	gdp_per_capita olympics_index	0.005790



5.2 PCA and KNN with Standardized Dataset for Prediction

Principal Component Analysis (PCA) and Feature Loading Analysis

- **PC1**: Primarily driven by **population** and **GDP**, representing a nation's overall macroeconomic strength.
- **PC2**: Likely reflects a combination of a nation's development level (**GDP per capita**) and Olympic performance (**olympics_index**).
- **PC3**: Captures the disparity between Olympic performance and development level. Primarily influenced by **olympics_index** and **GDP per capita**, suggesting that some nations excel in Olympic performance despite uneven economic development (or vice versa).

Feature Loadings:

	PC1	PC2	PC3
population	0.617178	-0.436693	0.028035
gdp	0.705288	-0.010391	-0.178776
gdp_per_capita	0.172261	0.702040	-0.605810
olympics_index	0.255302	0.555906	0.767629
sports_index	-0.163727	-0.085434	-0.104850

Information Retention and Dimensionality Reduction

- **Information Retention**: The first three principal components capture over **82%** of the data's variance, indicating that most of the critical information is preserved. These three components were selected for subsequent analysis.
- **Dimensionality Reduction**: PCA reduced the original 5 features to 3 principal components, effectively simplifying the dataset while retaining essential information.

```
Explained Variance Ratio (PCA): [0.39162376 0.2698857  0.16601544]
Cumulative Variance Ratio: [0.39162376 0.66150946 0.8275249 ]
```

Regression Analysis Using PCA Components and KNN

- The principal components extracted via PCA were used as input features for KNN regression to predict total medal counts.

```
KNN Regression - MAE: 2.41, MSE: 15.64, R²: 0.54
```

Model Performance

- **MAE**: The mean absolute error is **2.41**, indicating that the average deviation between predicted and actual medal counts is approximately 2.41 medals per country.
- **MSE**: The mean squared error is **15.64**, suggesting that while most predictions are accurate, some countries exhibit significant errors. However, overall, the error is not severe.
- **R^2**: The coefficient of determination is **0.54**, meaning the model explains 54% of the variance in medal counts.

Optimizing the KNN Model

- **Hyperparameter Tuning**:
    - Adjusted n_neighbors (number of nearest neighbors) to balance model complexity.
    - Used **GridSearchCV** to select the optimal hyperparameters.
- **Result**: The best parameters were identified, but the R^2 score indicates there is still room for improvement.

```
Best parameters: {'n_neighbors': 5}
```

Recommendations for Future Improvement

1. **Additional Feature Collection**:
    - Incorporate more dimensions, such as:

        - Government sports funding.
        - Proportion of the population participating in sports activities.
        - Medal counts across different sports disciplines.
    - Expanding the feature set could significantly improve the model's predictive power.
2. **Feature Interactions**:
    - Explore interactions between existing and new features to better capture complex relationships.
3. **Advanced Modeling Techniques**:
    - Experiment with ensemble models (e.g., Gradient Boosting Machines or XGBoost) or hybrid methods to further enhance prediction accuracy.

By expanding the data and refining the model, the KNN regression approach can achieve greater precision and explanatory power for predicting Olympic medal counts.

6. Summary

6.1 Feature Importance Analysis

- **Key Features**:
    - **GDP** and **population** were the most impactful factors in predicting medal counts. Additional analyses revealed that composite national indices and GDP per capita also held significant predictive value, emphasizing the influence of economic power and demographic size on Olympic outcomes.

6.2 Model Performance Summary

- **Model Selection**:
    - Both **random forest** and **neural networks** demonstrated promising results, with the neural network ultimately selected as the best-performing model. Its ability to capture complex nonlinear relationships gave it an edge over simpler models.

6.3 Future Recommendations

- **Expand Data Coverage**:
    - Extend the dataset to include multiple Olympic Games, allowing for a more comprehensive analysis of historical trends.
- **Incorporate Additional Features**:
    - Features such as government funding for sports and education, sports participation rates, and medal distribution by specific events could add valuable context and improve predictions.

- **Advanced Model Exploration**:
    - Implement state-of-the-art ensemble techniques like **XGBoost** or **LightGBM**, which excel at capturing intricate feature interactions.
- **Mitigate Overfitting**:
    - Refine model regularization strategies and apply early stopping to improve generalization performance.

This refined summary integrates professional language, detailed insights, and actionable recommendations, aligning with the expectations for a comprehensive report.