Olympic Insights

Deep Learning

MSCS 612N-816

DeepLearners

Marist College

School of Computer Science and Mathematics

Submitted To:

Dr. Reza Sadeghi

April 19, 2025

Project Report of Olympic Insights

Team Name

Deep Learners

Team Members

1. Elizabeth Herrera Elizabeth.Herrera1@marist.edu (Team Head)
2. Easton Eberwein Easton.Eberwein1@marist.edu (Team Member)
3. Austin Frank Austin.Frank1@marist.edu (Team Member)
4. Luca Gristina Luca.Gristina1@marist.edu (Team Member)

Description of Team Members

1. Elizabeth graduated from Marist in 2023 with a B.S. in cybersecurity. She is now back at Marist to complete her MSCS with a concentration in AI. Elizabeth also works as a function tester for a z/OS security product at IBM.

…….

2. Easton Eberwein graduated from Marist in 2024 with a B.S. in Information Systems and Technology. He is in his final semester of Marist's 5-year M.S.I.S. graduate program to receive a degree in Information Systems Management and Business Analytics. He has also been a member of the Marist men's track and field team for 4 years, being a captain for 3.

…….

3. Austin Frank graduated from Marist in 2024 with a B.S. in Computer Science - Software Development. He is in his final semester of Marist's 5-year graduate program to receive his M.S. in Software Development. Additionally, he is a developer for the z/TPF database and business events.

…….

4. Luca graduated from Marist in 2024 with a B.S. in Computer Science - Software Development. He is currently in his final semester as a part of Marist's 5-year graduate program to receive his M.S. in Software Development.

Table of Contents

Introduction

General Description

The Olympic Games are one of the world's most popular sporting events. Held every four years, competitors worldwide come to the games to achieve personal excellence and win medals for their countries. It's always exciting to see which athletes will come away with medals, but what if we had a way to predict the outcome of Olympic events? In this project, we aim to see how accurately an AI model can predict an athlete's Olympic performance based on age, height, weight, sport, and country.

Research Question

How accurately can an AI model predict an athlete's Olympic performance (e.g. medal likelihood or placement) based on age, height, weight, sport, and country?

GitHub Repository Address

https://github.com/ElizabethHerrera12/DeepLearners

URL of your dataset

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

This is a historical dataset on the modern Olympic Games, including all the Games from Athens 1896 to Rio 2016. This dataset includes basic bio data on the athletes as well as medal results. The dataset was collected from www.sports-reference.com in May 2018 using R code.

Name, definition, and characteristics of features

1. **ID** - Unique number for each athlete
2. **Name** - Athlete's name
3. **Sex** - M or F
4. **Age** - Integer
5. **Height** - In centimeters
6. **Weight** - In kilograms
7. **Team** - Team name
8. **NOC** - National Olympic Committee 3-letter code
9. **Games** - Year and season
10. **Year** - Integer
11. **Season** - Summer or Winter
12. **City** - Host city
13. **Sport** - Sport
14. **Event** - Event
15. **Medal** - Gold, Silver, Bronze, or NA

Review of Related Work

Machine learning and statistical models have been increasingly used to predict Olympic performance, particularly an athlete's likelihood of winning a medal. These studies take different approaches, with some focusing on country-level trends and others analyzing individual athlete characteristics. By comparing these studies, we can better understand their methodologies, results, strengths, and limitations in predicting Olympic success.

Country-level models primarily rely on economic and historical data to forecast Olympic medal counts. Studies using machine learning techniques, such as decision trees, random forests, and multiple linear regression, have identified GDP, population size, and past performance as the most significant predictors of a nation's success. These models achieve high accuracy, with some exceeding 90% in predicting medal counts for upcoming Olympics. A major advantage of these approaches is their ability to analyze long-term trends using extensive datasets spanning multiple Olympic Games. However, they fail to account for real-time adaptability, such as an athlete's current fitness level, injuries, or training improvements, limiting their applicability for predicting individual performance.

In contrast, athlete-centered models attempt to predict an individual's probability of winning a medal by incorporating personal attributes like age, height, weight, and previous results. These models, often built using classification techniques such as random forests and boosting algorithms, offer a more granular perspective on performance prediction. The Player Winning Probability Model (PWPM), for example, achieved an accuracy of 70.22%, demonstrating the potential of athlete-level predictions. However, compared to country-level models, these approaches generally have lower accuracy, possibly due to the challenge of quantifying non-physical factors such as mental resilience, injury history, or in-game decision-making.

Each approach has its strengths and weaknesses. Country-level models provide high-level insights into Olympic success, making them useful for policymakers and sports analysts, but they lack precision in predicting individual outcomes. Athlete-focused models are more tailored to personal performance but require further refinement to improve accuracy. A hybrid approach that combines macroeconomic indicators with individual athlete characteristics could enhance predictive capabilities, offering a more comprehensive model for forecasting Olympic success. Integrating real-time data, such as an athlete's recent training performance and injury reports, may further refine these predictions and provide a more dynamic and adaptable framework for future research.

Future Plan for Olympic Insights

Step 1: Data Collection

Our perspective sources for datasets:

1. Kaggle
2. Official Olympics Website

Step 2: Data Preprocessing

Categorization

Our features could be put into the following:

1. Identification Features (Not Useful for Modeling, But Needed for Tracking)
    a. ID – Unique number for each athlete
    b. Name – Athlete's name
2. Categorical Features (Require Encoding: One-Hot Encoding or Label Encoding)
    a. Sex – M or F (Binary Encoding: 0 for Male, 1 for Female)
    b. Team – Team name (One-Hot Encoding)
    c. NOC – National Olympic Committee 3-letter code (One-Hot Encoding)
    d. Season – Summer or Winter (Binary Encoding)
    e. City – Host city (One-Hot Encoding)
    f. Sport – Sport type (One-Hot Encoding)
    g. Event – Specific event (One-Hot Encoding)
    h. Medal – Gold, Silver, Bronze, or NA (Label Encoding: NA → 0, Bronze → 1, Silver → 2, Gold → 3)
3. Numerical Features (Require Normalization or Standardization)
    a. Age – Integer (Standardization)
    b. Height – In centimeters (Standardization)
    c. Weight – In kilograms (Standardization)
    d. Year – Integer (Could be used as a feature or transformed into Olympic cycles)
4. Composite Feature (Can Be Used to Derive Trends)
    a. Games – Combination of year and season (Extract year for trends or use season separately)

Standardization of Numerical Data

We will apply a standardization, most likely Z-score normalization, to our numerical features

1. Age
2. Height
3. Weight

Continuous Data

Ensure continuous data throughout our dataset and use appropriate statistical measures or predictive modeling for gaps.

Splitting Data

To test the effectiveness of our data preprocessing and model performance, we will split the dataset into training, validation, and test sets. This ensures that our model generalizes well to unseen data.

Initial Split: Training (70%) / Validation (15%) / Test (15%)

We will experiment with values to find the best balance between training size and generalization performance. Our goal is to maximize accuracy while minimizing overfitting.

Step 4: Choosing a Deep Learning Model

1. Artificial Neural Networks (ANNs)
    a. Best for tabular data like medal counts per country.
    b. Fully connected layers will capture relationships between country factors and Olympic success.
2. Recurrent Neural Networks (RNNs) / LSTMs
    a. Used if temporal dependencies are strong (e.g., predicting medals based on past Olympics).
    b. Can analyze sequential patterns in Olympic history.
3. Convolutional Neural Networks (CNNs)
    a. If image data (athlete photos, competition visuals) is included, CNNs will be used for feature extraction.

Step 5: Model Training and Evaluation

Once our dataset is preprocessed and split appropriately, we will proceed with training and evaluating our deep learning model. Our approach includes implementing optimizations such as early stopping and hyperparameter tuning to enhance model performance.

Training the Model

1. **Deep Learning Architecture:**
    a. We will likely use a Neural Network (MLP or CNN), depending on the dataset's structure.
    b. The model will consist of multiple layers with activation functions like ReLU and softmax (for classification).
    c. We will experiment with batch normalization and dropout layers to prevent overfitting.
2. **Training Process:**
    a. We will use backpropagation and gradient descent (Adam optimizer) to optimize model weights.

b. We will monitor validation loss during training to detect overfitting.
c. Batch Size & Epochs will be fine-tuned based on performance.

Hyperparameter Tuning

1. To optimize model performance, we will tune parameters such as:
    a. Learning rate (e.g., 0.001 vs. 0.0001)
    b. Batch size (e.g., 32 vs. 64)
    c. Number of layers & neurons per layer
    d. Dropout rate (e.g., 0.2 vs. 0.5)
2. Techniques used:
    a. Grid Search or Random Search to find the best hyperparameters.
    b. Bayesian Optimization for more advanced tuning.
3. Early Stopping:
    a. We will monitor the validation loss and stop training if it starts increasing, preventing overfitting.

Model Evaluation Metrics:

- Root Mean Squared Error (RMSE) – Measures prediction errors; lower RMSE indicates better accuracy.
- Mean Absolute Error (MAE) – Shows the average error in predictions.
- Confusion Matrix & F1 Score (if predicting medal categories) – Evaluates classification performance.

Comparison & Optimization:

- After initial evaluation, we will compare different models (e.g., CNNs vs. LSTMs vs. Fully Connected Networks) to determine the best-performing architecture.
- Model performance will be validated using the test set and real-world Olympic data trends.

Step 6: Insights and Visualization

Generate visualizations of medal predictions vs. actual outcomes.

Provide insights into which countries are over performing or underperforming based on socio-economic factors.

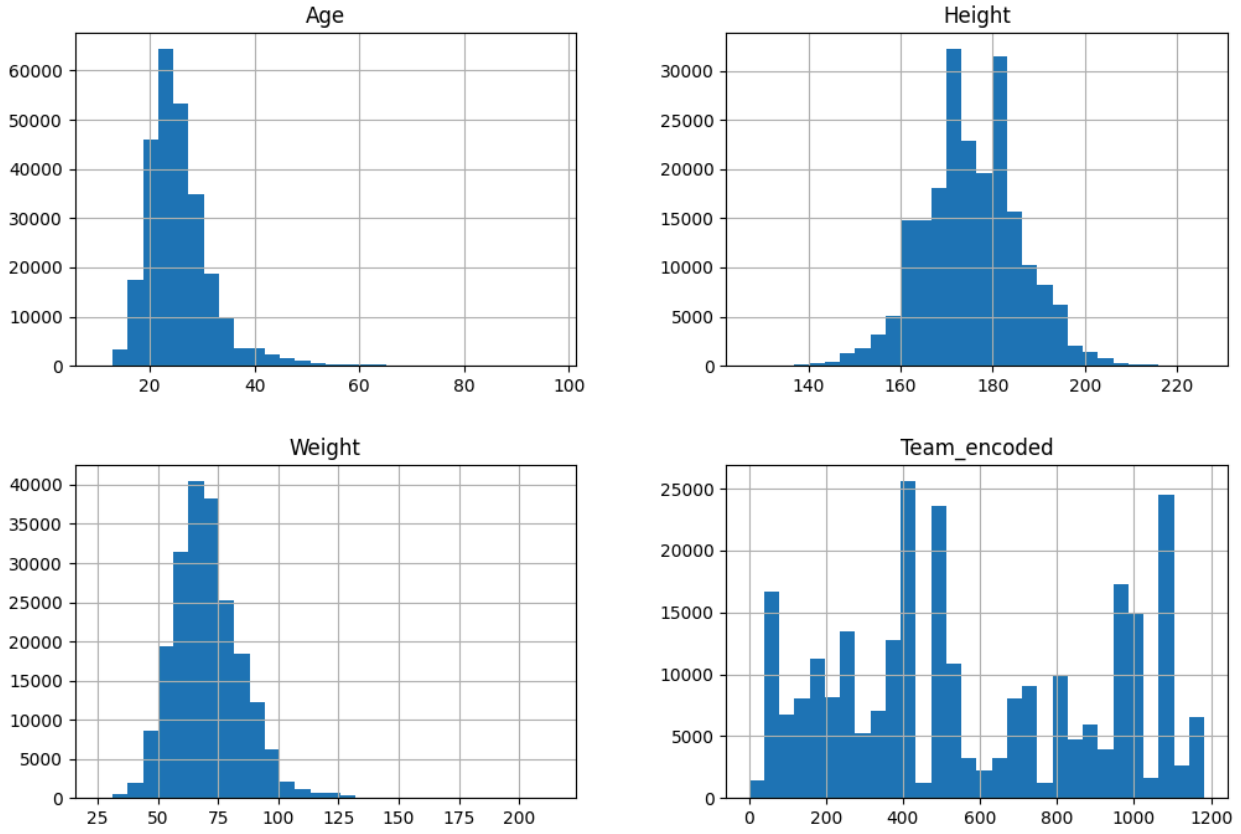
Data Exploration

Univariate Analysis

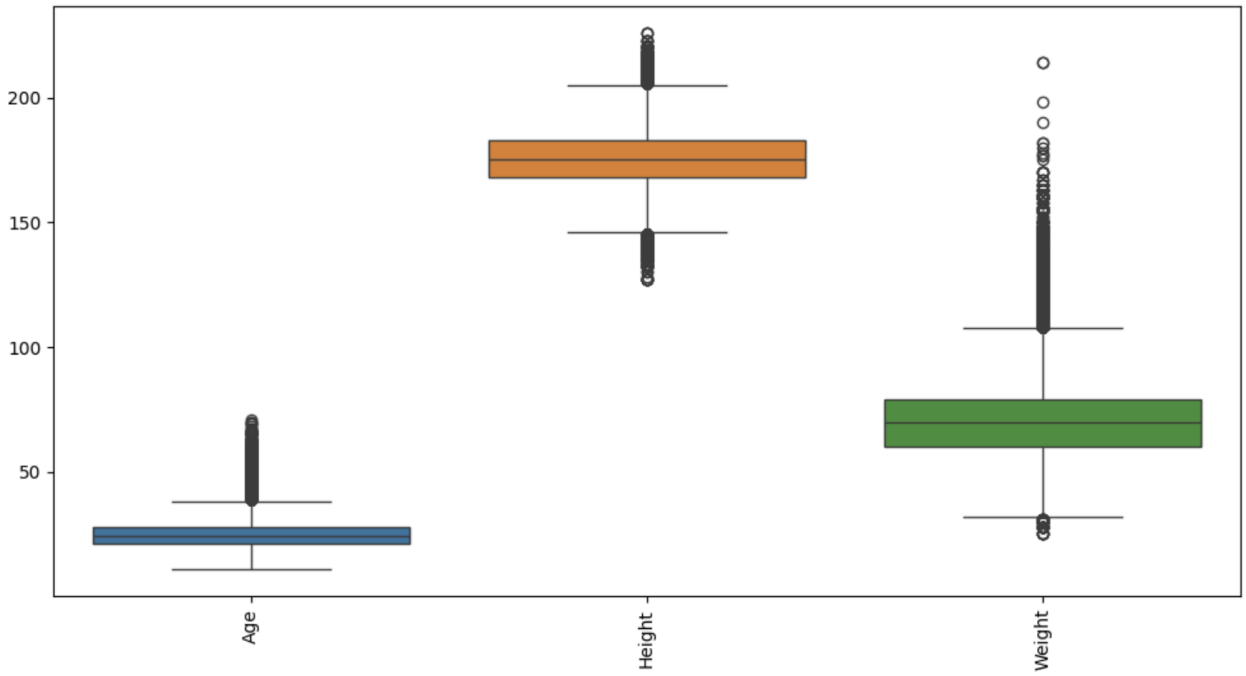
1. Descriptive Analysis

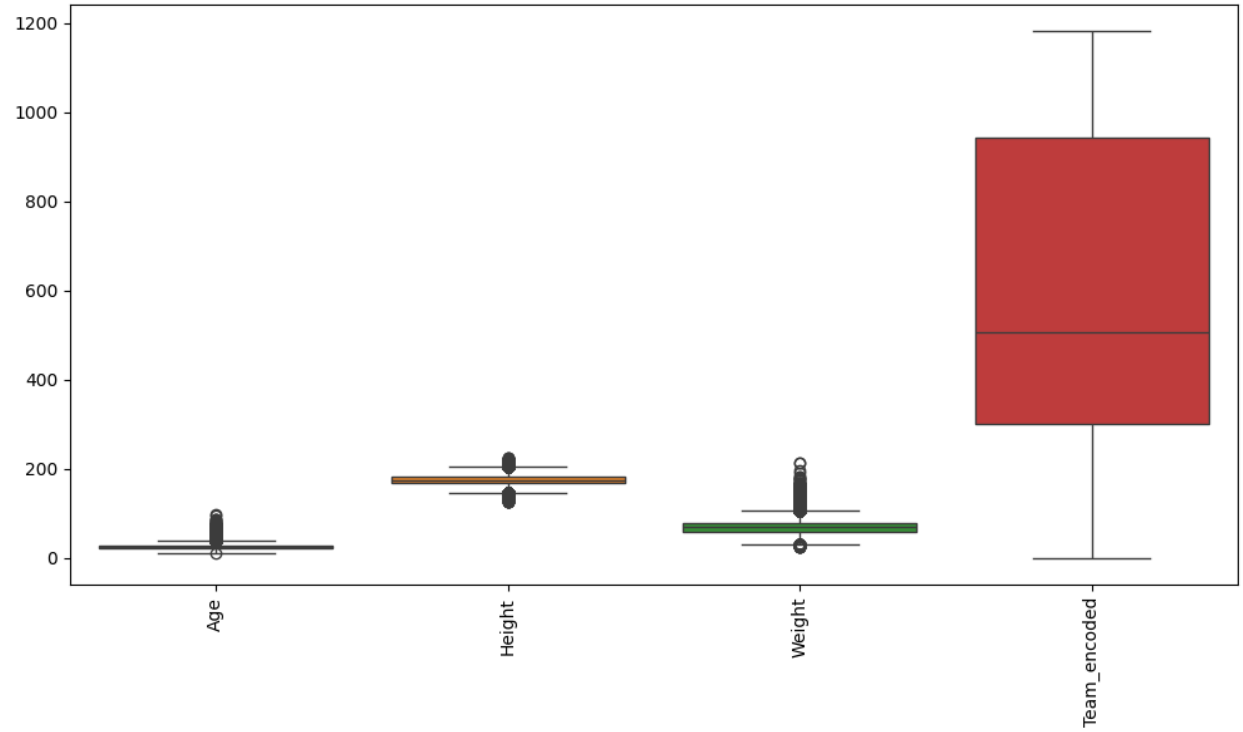
	ID	Age	Height	Weight	Year	Team_encoded	Medal_Encoded
count	30181.000000	30181.000000	30181.000000	30181.000000	30181.000000	30181.000000	30181.000000
mean	70225.949604	25.429012	177.642358	73.753554	1988.005964	136.980551	2.000630
std	38839.720551	5.049684	10.924188	15.004992	22.718451	80.363868	0.820443
min	16.000000	13.000000	136.000000	28.000000	1896.000000	0.000000	1.000000
25%	37494.000000	22.000000	170.000000	63.000000	1976.000000	75.000000	1.000000
50%	69771.000000	25.000000	178.000000	73.000000	1992.000000	119.000000	2.000000
75%	104111.000000	28.000000	185.000000	83.000000	2006.000000	206.000000	3.000000
max	135563.000000	66.000000	223.000000	182.000000	2016.000000	262.000000	3.000000

2. Distribution Analysis



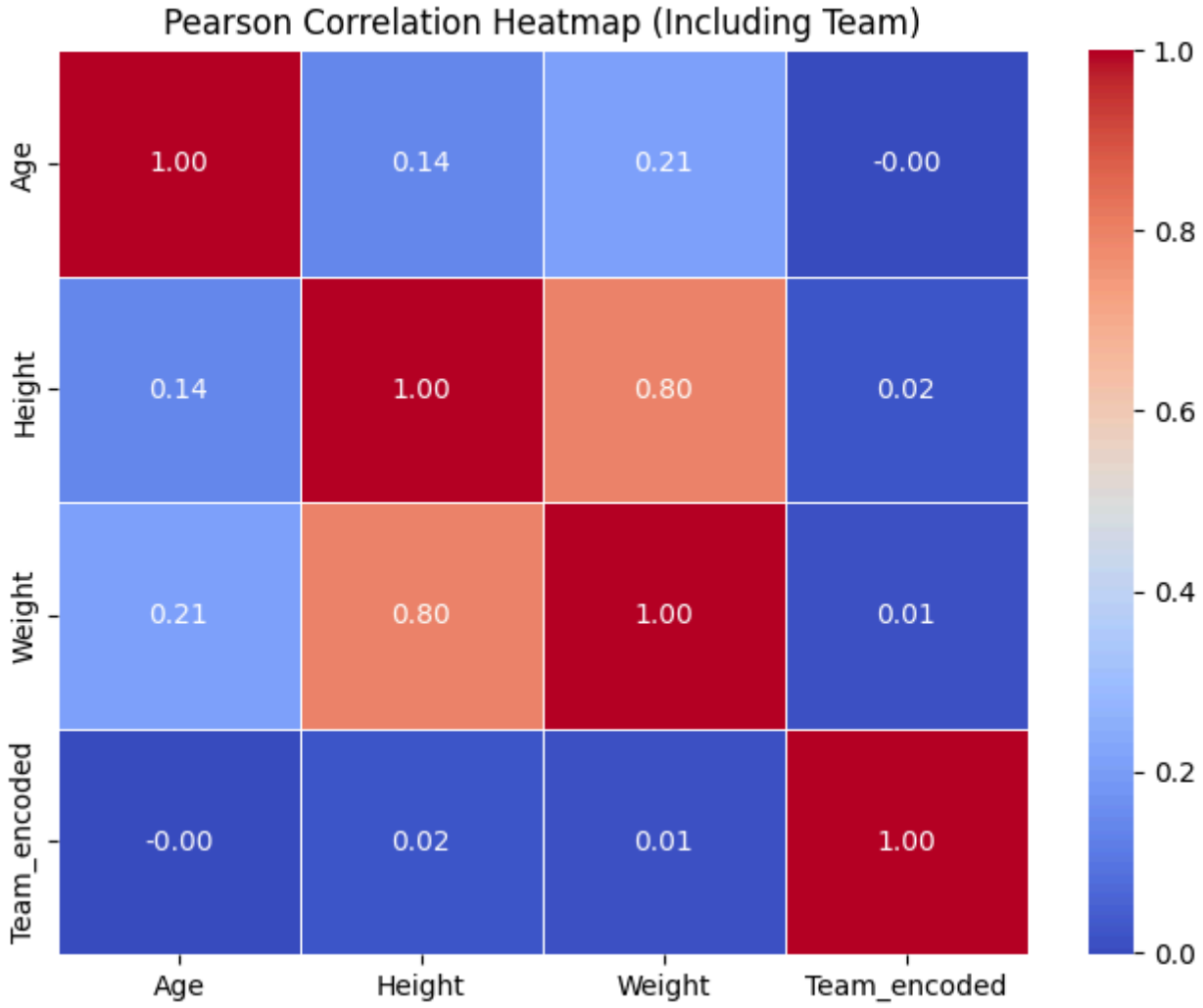
3. Outlier Detection


200
150
100
50
Age
Height
Weight

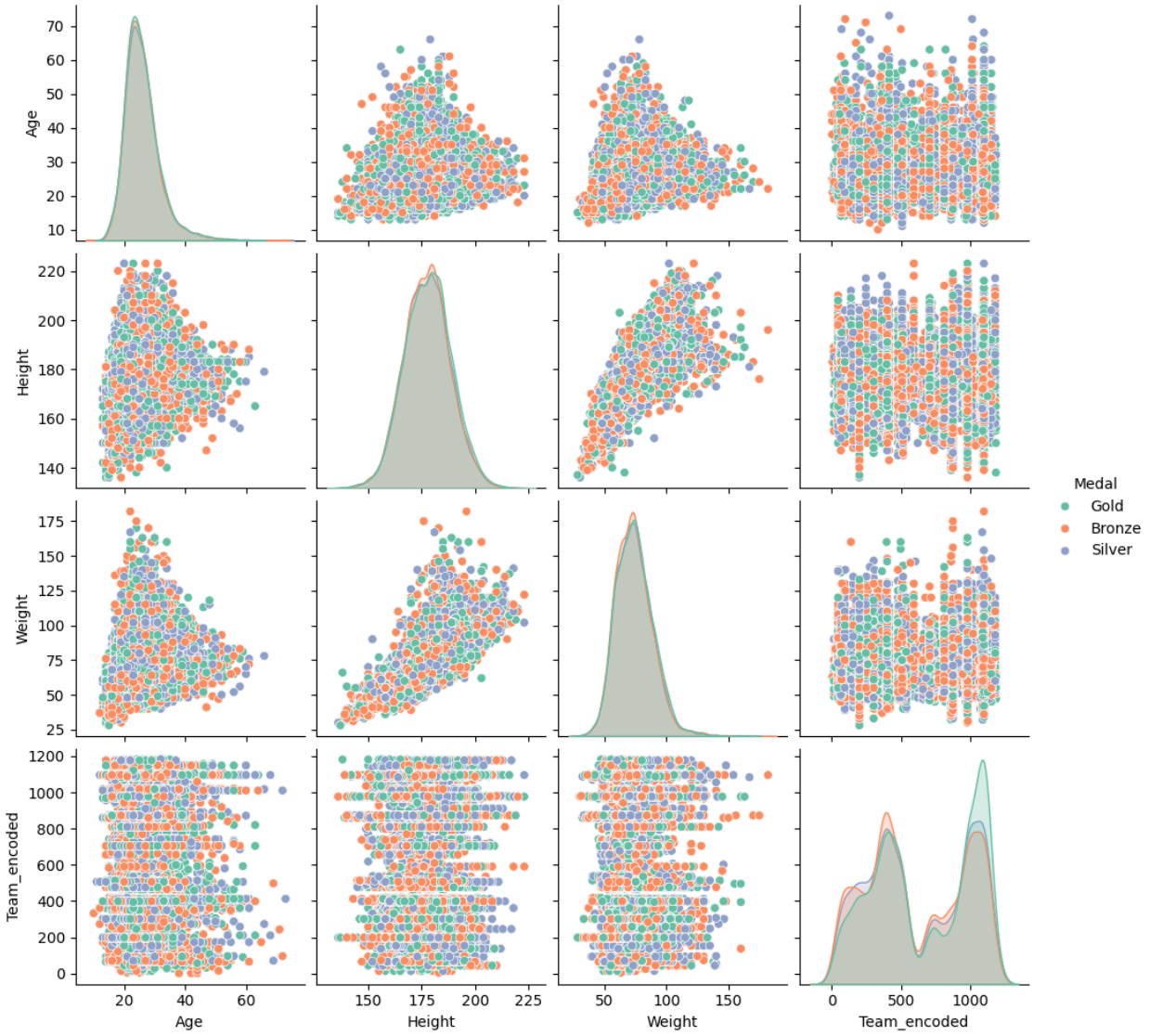

1200
1000
800
600
400
200
0
Age
Height
Weight
Team_encoded

Bivariate Analysis

1. Pearson correlation



2. Pair plot



Data Exploration Analysis

When looking at the information in our data frame we noticed multiple columns with NaN values. This posed a problem when cleaning our data to remove unfilled values and duplicates. Ultimately, this led to 85% of our data being removed. This would cause the accuracy of our model to decline as the dataset shrunk immensely severely. To combat this we encoded the medals column as integer values with the following mapping: 0: none, 1: bronze, 2: silver, 3: gold. When further looking into our data we realized that two other columns had a significant amount of null values, these columns being height and weight. We noticed that most of the null values for these fields came from earlier Olympics, 1960 and earlier. To solve this we are considering only looking at data from the 1960 Olympics and beyond. This wouldn't entirely alleviate the issue but would remove far fewer rows from the dataset relative to those who won medals.

Data Modeling

Preprocessing

We preprocessed our data by first encoding all the categorical columns, including sex, season, sport, event, medal, and team, to numerical values. Due to numerous null values we also removed all data prior to 1960. Additionally, we standardized the features height, weight, and age using StandardScaler to improve model performance and comparability.

Data Splitting

We split our data into three subsets of train, test, and validation. We put 70% of the data into the test set and 15% in both the test and validation sets.

Fitting the model

The model is a fully connected neural network designed for multi-class classification, predicting an athlete's likelihood of winning a medal. It consists of four dense layers: an input layer with 128 neurons, followed by hidden layers with 64 and 32 neurons, all using ReLU activation to introduce non-linearity. Batch normalization is applied after each dense layer to stabilize training, and dropout (30%) is used to prevent overfitting. The output layer has four neurons with a softmax activation function, corresponding to the four medal categories. The model is compiled using the Adam optimizer for efficient gradient-based learning and sparse categorical cross-entropy as the loss function, suitable for integer-labeled multi-class classification. Training is conducted over 50 epochs with a batch size of 32, using 70% of the data for training and 15% for validation, ensuring generalization before final evaluation on the test set.

Measuring Performance

Accuracy:

We decided to use accuracy rather than MAE as it is better for classification problems such as this. Our network had an accuracy of 0.8591 or almost 86% after training.

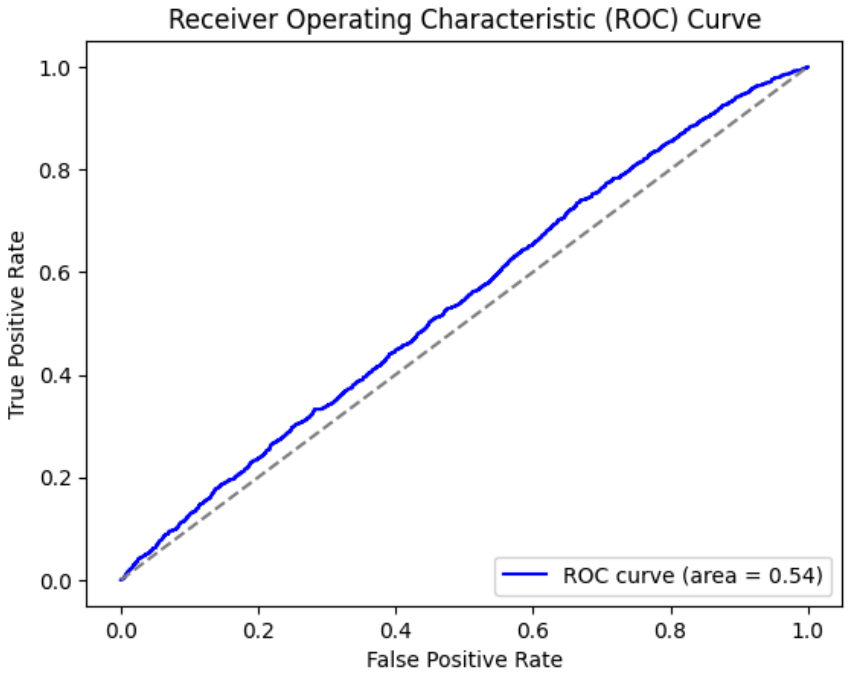
Confusion Matrix & Classification Report:

```
Confusion Matrix:
 [[25267     2     0     0]
 [ 1419     0     0     0]
 [ 1370     0     1     0]
 [ 1354     0     0     0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.86      1.00      0.92     25269
           1       0.00      0.00      0.00      1419
           2       1.00      0.00      0.00      1371
           3       0.00      0.00      0.00      1354

    accuracy                           0.86     29413
   macro avg       0.46      0.25      0.23     29413
weighted avg       0.78      0.86      0.79     29413
```

ROC Curve:



Optimization

Hyperparameter Tuning:

In this project phase, we focused on optimizing the performance of our model by performing hyperparameter tuning using Keras Tuner. We employed the Random Search strategy to find the best combination of hyperparameters, including the number of units in hidden layers, activation functions, and learning rate. After conducting five trials, the tuner identified the optimal configuration was 128 units in the first dense layer, the tanh activation function, and a learning rate of 0.0001. We then trained the model using these optimal hyperparameters for 20 epochs. The best validation accuracy achieved was 0.8594.

F1 Score:

In our model, calculating the weighted F1 score resulted in an output of 0.7940, indicating a strong balance between precision and recall across all classes. This means that our model is correctly identifying instances with nearly 79.4% effectiveness, considering both false positives and false negatives. Since the F1 score is the harmonic mean of precision and recall, this value suggests that our model is performing well, but there is still room for improvement. The use of `average=weighted'` ensures that each class contributes proportionally based on its

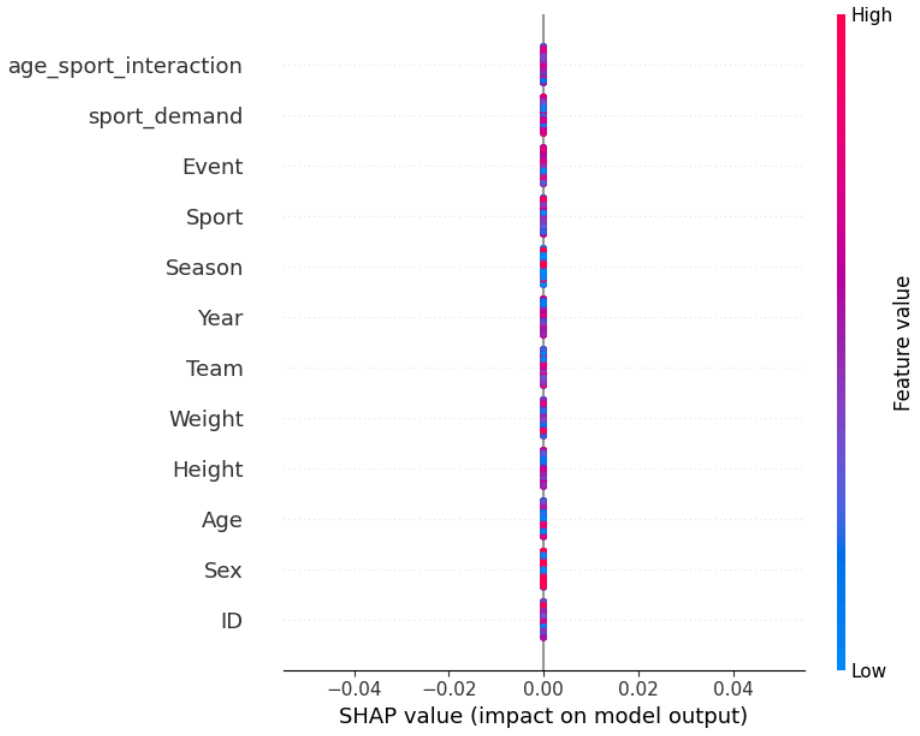
frequency in the dataset, preventing the majority class from dominating the evaluation metric. By optimizing for the F1 score, we can refine our model to enhance its classification performance, particularly for underrepresented classes, leading to a more reliable and balanced model overall.

Interaction Feature:

In an effort to improve our model, we created a new feature which captures the interaction between age and sport. We achieve this by giving each sport that is featured in the Olympics a difficulty rating based upon the "physical demand" of the sport, then multiplying that value by the age of the participant. This calculation gives us a feature that reflects the importance of age within each sport. After incorporating this new feature in the models training, the accuracy improved to .8591 or 85.91%.

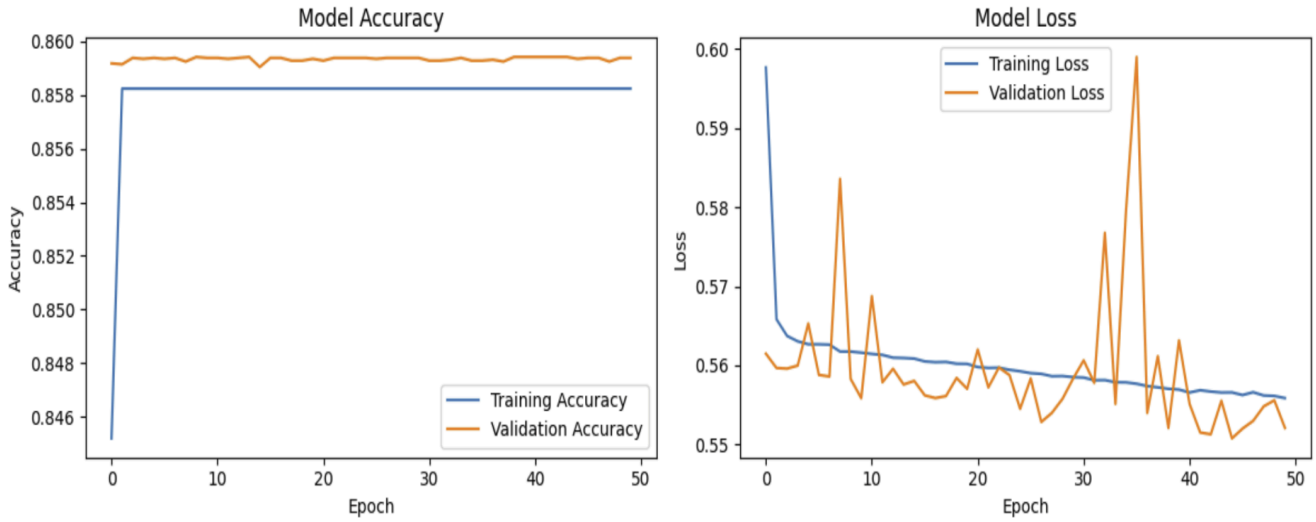
Model Evaluation

Feature Importance:



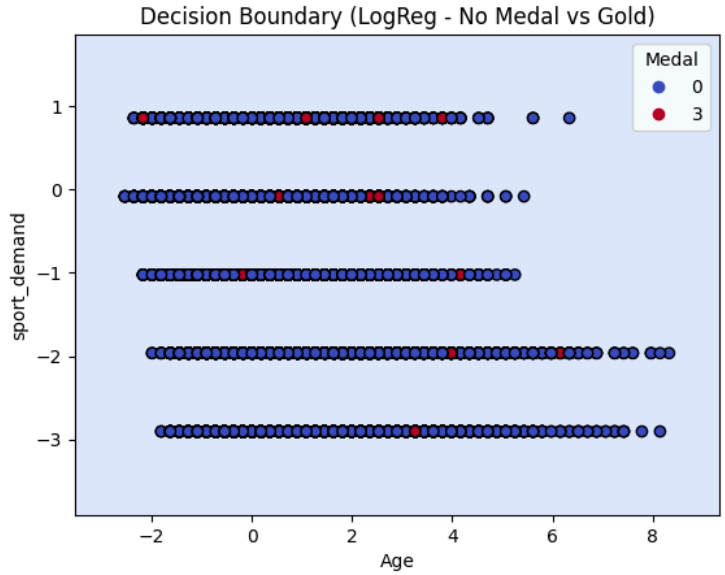
We sampled 1000 examples to use for feature importance evaluation. The output of this graph indicates that the features of our dataset do not heavily influence our model. We believe this is because roughly 85 percent of the athletes in our data set did not qualify for a medal. Due to this, it makes it very difficult for the model to identify feature impacts on medal qualification. For future steps, we will look into downsampling non qualifying data or oversampling medal classes.

Performance Visualization in Different Epochs:



These graphs demonstrate the model's performance across 50 epochs in terms of accuracy and loss for both training and validation sets. In the left graph, we observe that training accuracy stabilizes around 85.8%, while validation accuracy remains slightly higher and consistent around 85.9%. This indicates that the model generalizes well and is not overfitting. On the right graph, training loss steadily decreases, showing that the model is learning effectively. However, the validation loss fluctuates, which suggests that while the model encounters instability on some batches of validation data. Overall, the model appears to be well-trained with stable accuracy, but the noisy validation loss warrants further tuning to improve consistency.

Decision Boundary Visualization



The plot above illustrates the decision boundary produced by a logistic regression classifier trained to distinguish between athletes who won no medal (label 0) and those who won gold medals (label 3) using two features: Age and sport_demand. Each point represents an athlete, with blue indicating no medal and red indicating gold. The visualization shows how the model attempts to separate the two classes based on these features, though the overlap suggests that Age and sport_demand alone do not offer strong linear separability. The sparse distribution of gold medalists also highlights class imbalance, emphasizing the importance of using additional features and more complex models for improved predictive performance.

References

1. https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results
2. https://www.smithsonianmag.com/science-nature/can-statistical-model-accurately-predict-olympic-medal-counts-180949627/
3. https://nycdatascience.com/blog/student-works/olympics-medals-prediction/
4. https://www.ijfmr.com/papers/2024/2/18036.pdf
5. https://ijariie.com/AdminUploadPdf/OLYMPIC_GAME_ANALYSIS_AND_PREDICTION_USING_MACHINE_LEARNING_ijariie24076.pdf?srsltid=AfmBOopdu1a8PNutXlzheNrDUJwh_yTL6DKKoO77nzqo9gDdVGLkLyji
6. https://www.researchgate.net/profile/Nongmeikapam-Thoiba-Singh/publication/378535086_Predicting_Medal_Counts_in_Olympics_Using_Machine_Learning_Algorithms_A_Comparative_Analysis/links/65f5549d1f0aec67e29d3db3/Predicting-Medal-Counts-in-Olympics-Using-Machine-Learning-Algorithms-A-Comparative-Analysis.pdf