B“H

Deep learning project

Submitted by: Hadar Froimowich 213118458, Ron Yakobovich 212767214, Lidor Mondel 207478256

1. Goal and Dataset Description

link to the data set:
https://www.kaggle.com/datasets/sanjeetsinghnaik/fifa-23-players-dataset

the link to the code: https://github.com/ron12120/deep-learning-project-.git

The goal of this project is to predict the value of FIFA 23 players based on their statistics attributes and Wikipedia introductions. The dataset includes both numerical features and text descriptions (that we added manually to the data set), making this a regression problem.

The dataset is shuffled and split into training (80%) and testing (20%) (and validation when needed) sets to ensure proper evaluation. Numerical features are normalized for better model performance, and text data is tokenized using the DistilBERT tokenizer.
also the data contains information about 1,700 FIFA 23 players and includes attributes essential for predicting their market value. It integrates **numerical features**, such as performance metrics, and **textual data**, such as Wikipedia introductions, making it ideal for hybrid machine learning models. Below is a breakdown of relevant aspects:

Numerical Features:

These features directly influence a player's performance and market value:

- **Performance Metrics**: Ratings for shooting, passing, dribbling, defending, and physicality.
- **Skill Attributes**: Weak foot rating, skill moves, finishing, sprint speed, and agility.
- **Physical Stats**: Age, height, weight, and stamina.
- **Contract and Club Data**: Wages and release clauses.

Textual Feature:

Wikipedia_Intro: Short biographies or descriptions of players from Wikipedia, offering qualitative context about their popularity, achievements, or unique traits.

Target Variable:

Value (in Euro): Represents the player's market value in euros, which is the focus of prediction.

Train-Test Split:

- Training set: 80% of the data.
- Testing set: 20% of the data.

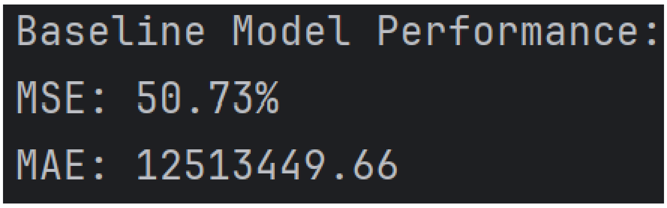
2. Baseline Model Description and Results

The baseline model always predicts the mean value of the target variable Value(in Euro).

Metrics for Regression:

- Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values.
- Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual values.

Results:



```
Baseline Model Performance:
MSE: 50.73%
MAE: 12513449.66
```

Baseline Model Performance:

MSE: The value 50.73% indicates that the baseline predictions (which are simply the mean of the target values) deviate significantly from the actual values when scaled relative to the mean squared value of the target variable (y). A higher MSE percentage implies less accuracy, as the predictions are far from the actual values on average.

MAE: The 12,513,449.66 euro MAE suggests that, on average, the baseline model's predictions differ from the actual player values by about **12.5 million euros**. This provides a measure of how much error there is in absolute terms, regardless of direction (overestimation or underestimation).

3. Linear Regression Model

A linear regression model was implemented to predict player value using numerical features. The model serves as a benchmark for comparison with more advanced models.

Data splitting:

- The data is split into training (80%) and testing (20%) sets.
- The train and test are being shuffled in order to avoid overfitting.

Scaling:

Numerical features and the target variable, player value, is scaled with StandardScaler to normalize the data.

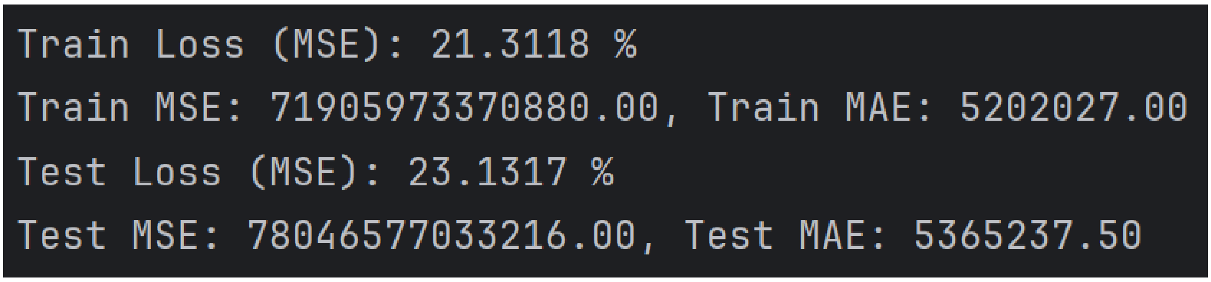
Model evaluation:

- Performance is evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE).
- Loss percentages are calculated for both the training and test sets.

Initial Attempts:

- The initial implementation did not scale the numerical features, resulting in poor performance due to the varying magnitudes of features.
- Correction: Features were normalized using StandardScaler.

Results after Correction:



```
Train Loss (MSE): 21.3118 %
Train MSE: 71905973370880.00, Train MAE: 5202027.00
Test Loss (MSE): 23.1317 %
Test MSE: 78046577033216.00, Test MAE: 5365237.50
```

Linear Regression Outperforms Baseline:

- Both MSE and MAE values for the linear regression model are significantly lower, reflecting better performance.
- The percentage loss shows that the linear regression model reduces error substantially compared to the baseline.
- This shows that the linear regression model effectively captures relationships in the data, providing much better predictions compared to the simplistic baseline approach.

Improvement Across Metrics:

- The baseline model simply predicts the mean, leading to large errors, especially in MSE. Linear regression leverages features to provide much more accurate predictions.
- **Training Set:**

MSE Improvement: **29.42%**

MAE Improvement: **58.42%**

- **Test Set:**

MSE Improvement: **27.60%**

MAE Improvement: **57.12%**

Room for Improvement:

While linear regression performs better, test loss still indicates scope for enhancing the model further (e.g., using more complex models or feature engineering).

4. Fully Connected Neural Network

A basic fully connected neural network was implemented using numerical features.

Load and preprocess the dataset:

- Features and the target variable are extracted.
- The data is split into training (80%), validation (10%), and testing (10%) sets using randomized shuffling.

Scaling features and target:

- The features (X) and target (y) are scaled separately using `StandardScaler` to normalize the data.
- The target values are reshaped and scaled with a separate scaler.

Convert data to PyTorch tensors:

The scaled features and target values are converted into PyTorch tensors, which are used for training and testing.

Define the neural network model:

A simple neural network is defined with:

- An input layer, one hidden layer with 64 neurons, and another hidden layer with 32 neurons.
- ReLU activations are used between layers.
- An output layer with a single neuron for predicting the player value.

Loss function and optimizer:

- The Mean Squared Error (MSE) loss function is used for regression.
- The Adam optimizer is chosen with a learning rate of 0.001.
- Dropout was added in order to avoid overfitting.

Early stopping parameters:

The training process incorporates early stopping, where training will stop if validation loss doesn’t improve for 20 consecutive epochs.

Training the model:

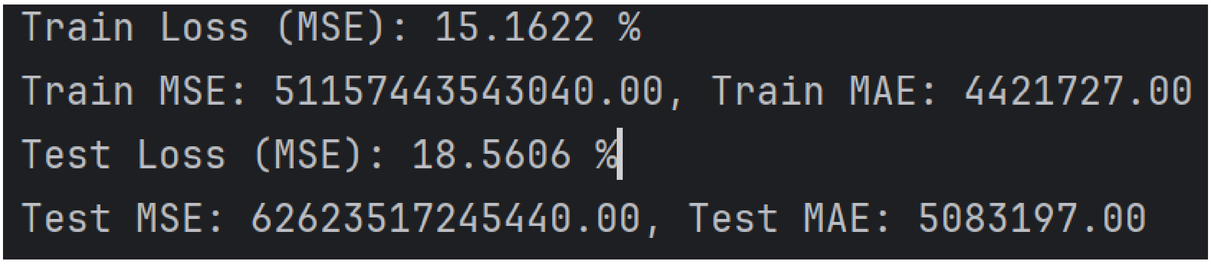
- The model is trained for up to 200 epochs.
- For each epoch, forward and backward passes are performed, and the optimizer updates the model parameters.
- Validation loss is calculated after each epoch to monitor performance and check for early stopping.

Test the model:

- After training, the model is tested on the test set, and predictions are made.
- The predictions are inverse-transformed to get the actual player values.
- MSE and MAE are calculated for both training and test data to assess model performance.

Initial Attempts:
 - The model overfit the training data due to a lack of regularization.
- Correction: Dropout layers were added, and early stopping was introduced to prevent overfitting.

Results after Correction:



```
Train Loss (MSE): 15.1622 %
Train MSE: 51157443543040.00, Train MAE: 4421727.00
Test Loss (MSE): 18.5606 %
Test MSE: 62623517245440.00, Test MAE: 5083197.00
```

Comparison:

1. Mean Squared Error (MSE):

- **Train MSE**:
  The neural network has a **lower MSE (15.16%)** compared to the linear

regression model (21.31%), indicating it performs better on the training set.
Improvement: **6.15%.**

- **Test MSE**:
  The neural network also has a **lower MSE (18.56%)** compared to the linear regression model (23.13%), indicating it generalizes better to unseen data.
  Improvement: **4.57%.**

2. Mean Absolute Error (MAE):

- **Train MAE**:
  The neural network has a **lower MAE (4,421,727 euros)** compared to the linear regression model (5,202,027 euros).
  Improvement: **780,300 euros.**
  As a percentage: **15.00%**.
- **Test MAE**:
  The neural network has a **lower MAE (5,083,197 euros)** compared to the linear regression model (5,365,237.50 euros).
  Improvement: **282,040.50 euros.**
  As a percentage: **5.26%**.

5. Advanced Neural Network

The Advanced Neural Network model used for predicting player values is **RNN**, combining numerical features and text data through separate processing branches. The numerical features, such as player attributes and statistics, are processed by a multi-layer fully connected network with ReLU activations. For the text data (Wikipedia introductions), a pre-trained DistilBERT tokenizer is used to tokenize the text, which is then processed through an LSTM (Long Short-Term Memory) layer to capture sequential information. The output of both branches is concatenated and passed through a series of fully connected layers for the final prediction.

The hybrid design allows for the integration of both structured and unstructured data, aiming to improve the model's performance in estimating player values.

Load and preprocess the dataset:

- The data is shuffled randomly to ensure randomness, and then split into training and testing sets.
- The numerical features are scaled using `StandardScaler`.
- The target values (`Value(in Euro)`) are scaled similarly.

Tokenize text data:

- The Wikipedia_Intro text column is tokenized using the DistilBERT tokenizer.
- The tokenizer processes the text into token IDs and handles padding and truncation.

Define the Hybrid RNN model:

- The model consists of two branches:
  - **Text branch:** An embedding layer followed by an LSTM layer to process the tokenized text data.
  - **Numerical branch:** A fully connected network that processes the scaled numerical features.
- The output of both branches is combined, and a final fully connected layer produces the output prediction.

Initialize the model:

The model is initialized with parameters such as the vocabulary size, embedding dimension, and the hidden size of the RNN.

Loss function and optimizer:

- The Mean Squared Error (MSE) loss function is used to measure the difference between predicted and actual values.
- The Adam optimizer is used to update the model's parameters.

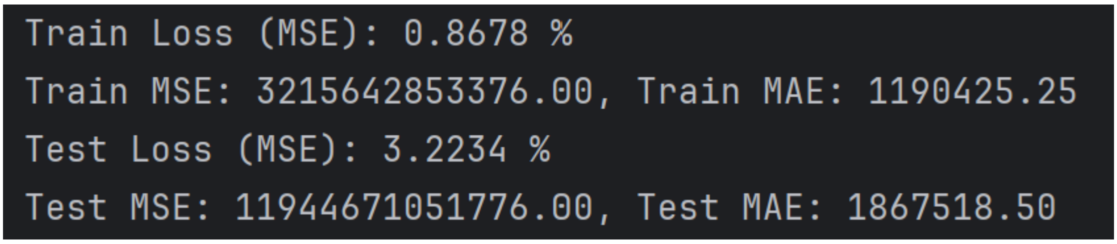
Training the model:

- The model is trained for 150 epochs.
- For each epoch, the forward pass computes the predictions, and the backward pass updates the model’s parameters based on the loss.

Initial Attempts:

- The text embeddings were not pre-trained, leading to poor performance.
- Correction: DistilBERT pre-trained embeddings were utilized to improve textual feature representation. Batch normalization was also added to stabilize training.

Results:



```
Train Loss (MSE): 0.8678 %
Train MSE: 3215642853376.00, Train MAE: 1190425.25
Test Loss (MSE): 3.2234 %
Test MSE: 11944671051776.00, Test MAE: 1867518.50
```

Comparison:

Mean Squared Error (MSE):

- Linear Regression Model:
  The MSE for the training set was **71.9 trillion**, while the test set had a slightly higher value of **78 trillion**. This reflects the model's baseline performance.
- Initial Neural Network:
  The MSE improved significantly on the training set, decreasing to **51.2 trillion** (~**28.8%** improvement compared to the Linear Regression Model). The test set MSE also decreased to 62.6 trillion (**~19.7%** improvement).
- Advanced Neural Network:
  The training MSE was further reduced to **32.2 trillion** (**~55.2%** improvement compared to the Linear Regression Model). The test set MSE also decreased to **11.9 trillion**, the best result across all models (**~84.7%** improvement).

Mean Absolute Error (MAE):

- Linear Regression Model:
  The MAE for the training set was **5.20 million**, and the test set was **5.37 million**, reflecting the model's baseline performance.
- Initial Neural Network:
  The MAE on the training set decreased to **4.42 million** (**~15%** improvement), and the test set MAE dropped to **5.08 million** (**~5.4%** improvement).
- Advanced Neural Network:
  The training set MAE improved further to **1.19 million** (**~77.1%** improvement compared to the Linear Regression Model), while the test set MAE was reduced to **1.87 million**, the best result across all models (**~65.2%** improvement).

Training Loss (MSE Percentage):

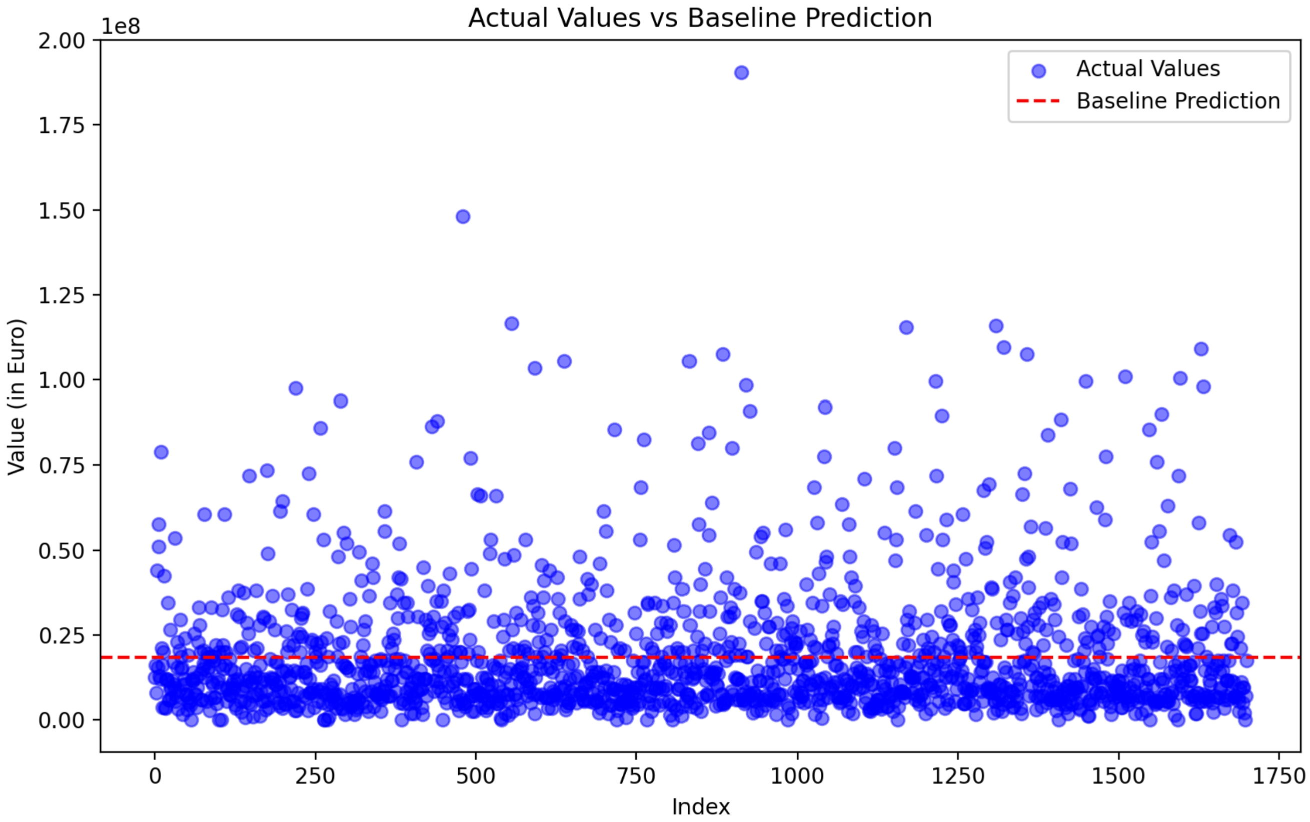
- Linear Regression Model:
  The training loss stood at **21.31%**, reflecting baseline inefficiencies.
- Initial Neural Network:
  The training loss decreased significantly to **15.16%**, an improvement of **~28.8%**.
- Advanced Neural Network:
  The training loss was further reduced to **0.87%**, representing an improvement of **~95.9%** compared to the Linear Regression Model.

Test Loss (MSE Percentage):

- Linear Regression Model:
  The test loss was **23.13%**, reflecting relatively poor generalization.
- Initial Neural Network:
  The test loss dropped to **18.56%**, an improvement of **~19.8%**.
- Advanced Neural Network:
  The test loss decreased further to **3.22%**, the best performance among all models (**~86.1%** improvement compared to the Linear Regression Model).

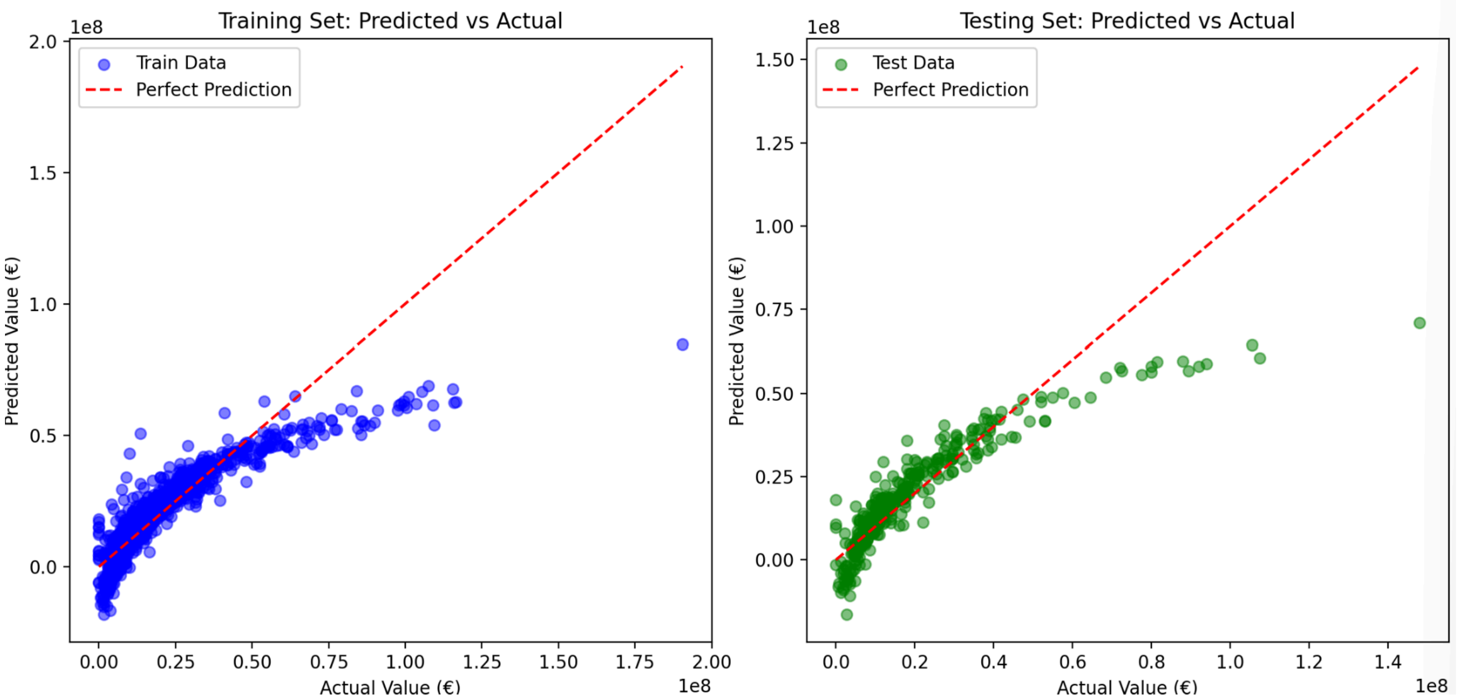
6. Visualizations

Baseline Model:



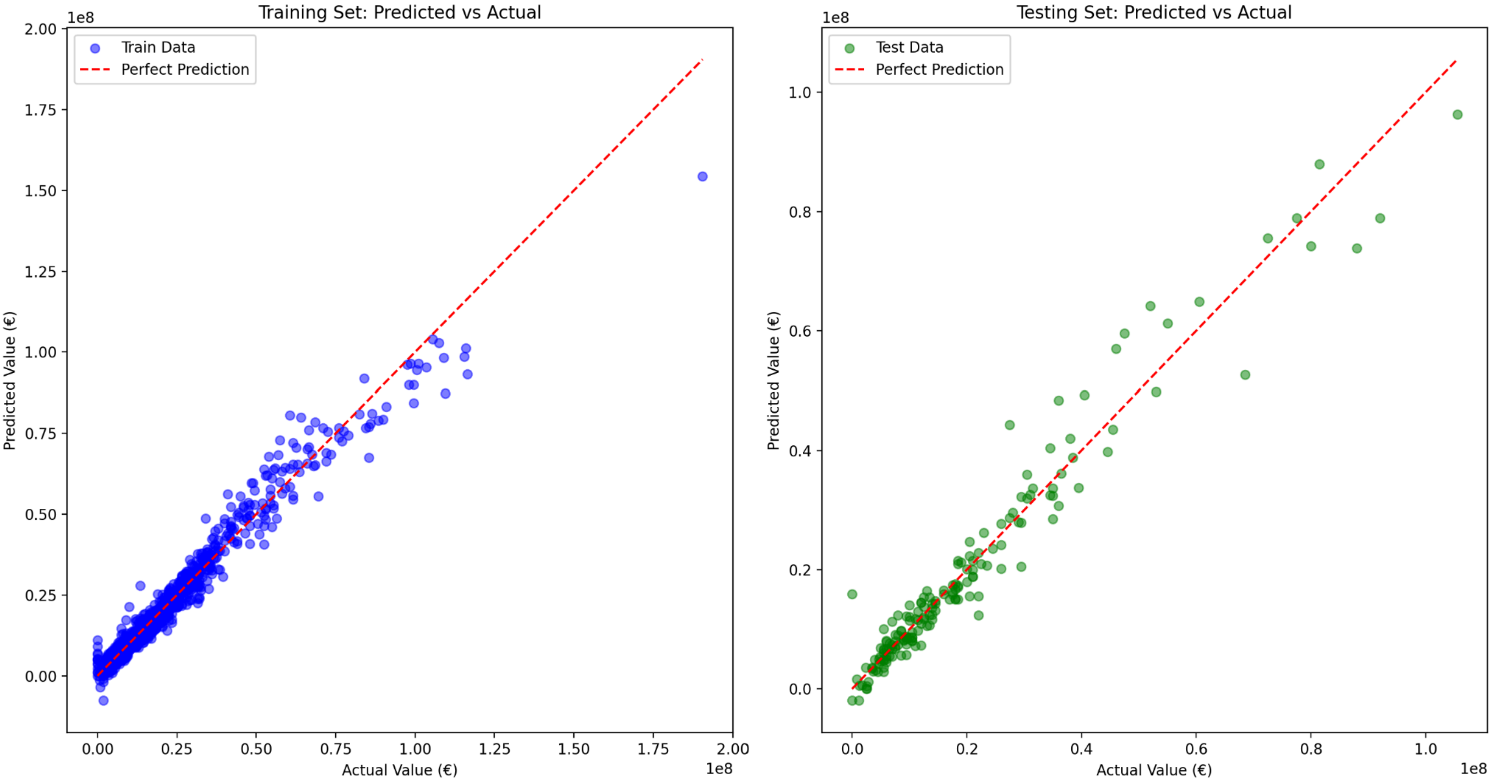
This graph shows how a simple baseline model performs in predicting FIFA 23 player values. The red dashed line represents the model's constant prediction (around 20 million euros), while the blue dots show actual player values. The wide spread of blue dots above the red line indicates the model consistently underestimates high-value players, showing its limitations as a predictive tool.

Linear Regression Model:



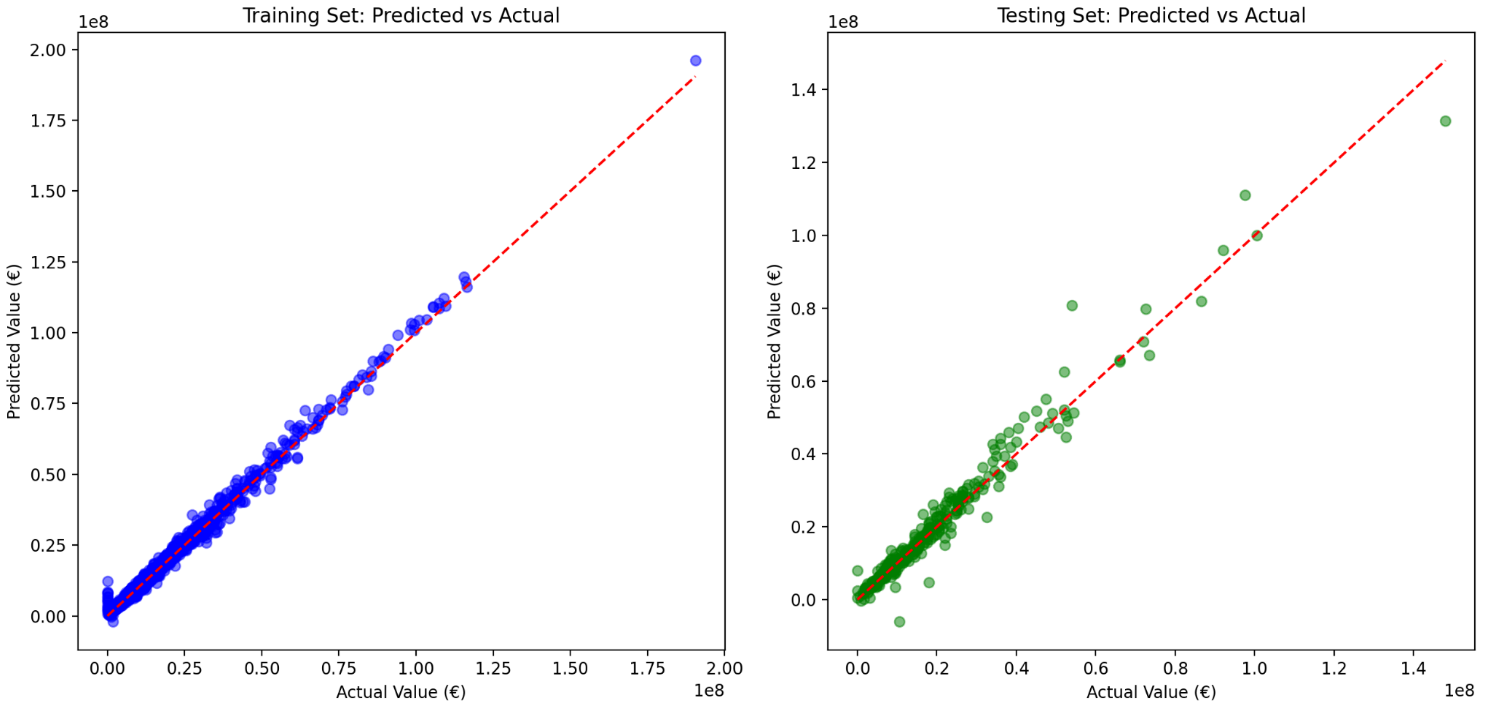
These graphs show how well the linear regression model predicts FIFA player values. In both training (blue) and test (green) sets, points close to the red line indicate accurate predictions.
The model generally performs well for lower-valued players but tends to underestimate some high-value players, as shown by points deviating from the ideal prediction line. The similar pattern in both sets suggests the model is consistent but has some limitations in accurately predicting extreme values.

Fully Connected Neural Network:



These graphs shows improved performance compared to the previous models. The data points (blue for training, green for testing) cluster more tightly around the red perfect prediction line, indicating more accurate predictions across different value ranges. While there's still some deviation for higher-valued players, the overall predictions are more consistent and accurate than both the baseline and linear regression models.

Advanced Neural Network:



The RNN graphs demonstrates even better performance than the fully connected neural network. In the training set (blue), points align almost perfectly with the ideal prediction line, showing excellent learning. The test set (green) also shows strong performance, though with slightly more scatter at higher values. This indicates the RNN has captured the underlying patterns in player values more effectively than previous models, as text data often contains rich contextual information about a player's skills, achievements, and potential that directly influence their market value.