,,

From Stats to Stars: A Machine Learning Approach for Football Player Rating Prediction

Kendra Jean Jacques*
Texas Tech University, Department of Physics and Astronomy , Lubbock , Texas
(Dated: May 8, 2025)

In the modern era with the rapid technological advancements for data analysis and data training, we can use some of these applications and apply it to sports analysis. Data-driven sports programs can help provide an accurate and objective evaluation of football players, which is essential for scouts, prospective coaches, football analysts, and fans. Data- driven rating systems trump the traditional player rating systems, because it is not subjective to inconsistency,, which can be fueled by infuriated human judgment or influenced by biased sources. The main objective of this project is to develop a machine learning-based framework to predict footballer rating solely based on in-game performance statistics. The data set includes information for players from a wide variety of different leagues, countries, and years of active play. The key attributes that will be used in the analysis and that will influence their total match statistics are goals scored, assists, pass accuracy, tackles, and total minutes played. Some other key attributes that will be used are players, team dynamics, physiological metrics, and external influences such as game conditions. This project will use several machine learning algorithms, such as Linear regressions, Gradient Boosting, Random Forest Regression, and K-Nearest Neighbors (KNN). This project aims to develop an interpretable machine learning model that can assist coaches, club managers, national team managers, and fantasy team participants in optimizing player selection and game strategies. This project can demonstrate the potential of machine learning to transform player assessments into replicable evidence-based insights in the world of football.

Keywords: Football, Ratings, Machine Learning

* Correspondence email address: kendra.jean-jacques@ttu.edu

I. INTRODUCTION

Football is one of the most widely followed sports globally, largely due to the participation of numerous countries in high-stakes tournaments such as the FIFA World Cup. The sport garners immense attention from fans, analysts, and sports podcasters who track player performance across various domestic leagues and international competitions. Player rating systems serve as a numerical summary of a player's overall ability, synthesizing aspects such as skill, consistency, and impact. These ratings are indispensable for game simulations, match predictions, fantasy leagues, team management decisions, and even FIFA video game mechanics.

Given the complexity of evaluating a player's contributions across diverse performance metrics, machine learning presents a compelling framework to construct accurate and scalable rating prediction models. This study is dedicated to predicting overall football player ratings using multiple supervised learning models, comparing their effectiveness, and identifying the most robust approach. The evaluation focuses on regression-based techniques and standard error metrics such as RMSE and R^2.

With the exponential growth in football performance data availability, the intersection of data science and sports analytics offers a significant opportunity to enhance insights and decision-making. Unlike traditional scouting and expert judgment, which may be susceptible to biases, machine learning enables evidence-based evaluations that account for physical attributes, technical skills, and contextual variables. This project seeks to replicate and refine existing rating methodologies through systematic feature engineering, statistical modeling, and rigorous validation.

The evaluation of football players has historically relied on subjective assessments and limited statistical indicators. These traditional approaches often lack consistency and scalability. In contrast, machine learning can detect nonlinear patterns, handle high-dimensional datasets, and generalize across player populations. The overarching aim is to develop a predictive system that is not only accurate but also interpretable and applicable in real-world football analytics.

A. Dataset and Feature Overview

The dataset I used was obtained from the Kaggle website[[1]]. The dataset for this project aggregates player information from international and club football teams, covering leagues such as the English Premier League (EPL), La Liga (Spain), Serie A (Italy), and other top competitive leagues. The dataset provided information for over 17,000 football players and 51 attributes. The player profiles included a wide array of features categorized into the following group:

- **Demographic and Team Information**:This category covers information such as their Nationality, league and club affiliation and finally which potions they play. These positions covered all aspects of the field for example goalkeepers, center forwards, attacking midfielders and center backs just to name a few positions.

- **Technical and Skill Metrics**: This category is one of the most important categories because it provided most of the data that was used in the prediction of the players overall ratings. The category included all the skills such as dribbling, short passing, long passing , ball control, pass accuracy, shooting and defensive capabilities. For shooting a lot of emphasis was placed on shot execution and finishing, shot power, and long range shots. Similarly, for defensive capabilities emphasis was placed on slide tackling, tackle accuracy, and standing tackles.

- **Physical Attributes**: This category took into consideration all the physical aspects of the players, which included their height and weight, stamina and strength, sprint speed and acceleration.

- **Current Overall Ratings:** For the overall ratings category it served as a good benchmark when developing the different models. Since it provided an aggregate score representing a players overall ability level, it was good to have a score to compare with the predictions from the different models.

II. METHODOLOGY

This project employs the use of a comparative machine learning approach by training four different regression models on the same dataset and evaluating their performance using three statistical metrics. However, before the models could be used the the initial data had to be preprocessed. The preprocessing staged included removing incomplete or anomalous records, standardizing numerical values using Min-Max scaling, encoding categorical variables such as position and nationality using one-hot encoding and finally splitting the data into training and testing datasets which was and 80/20 split respectively. Once preprocessing was complete I was able to employ the use if the different models. The models chosen represented a mix of Linear, instance-based, and ensemble learning techniques such as:

- **Linear Regression:** This model was chosen as baseline since it assumes a linear relationship between the inputs and outputs.

- **K-Nearest Neighbors (KNN):** Predicts a player's rating based on the average rating of the most similar players in feature space. It is sensitive to feature scaling and the choice of (K).

- **Random Forest Regressor:** An ensemble of decision trees trained on bootstrapped samples. The final prediction is the average of individual tree outputs,making it robust and less prone to overfitting.

- **Gradient Boosting Regressor:** Builds models sequentially, each one correcting residuals of the previous. This model is known for achieving high accuracy at the cost of longer training times.

Once all the models were decided on the next step was to decide on which metrics can be used to assess each models performance evenly across all four chosen. The evaluation metrics used were as follows:

- **Root Mean Squared Error (RMSE):**Measures the average magnitude of prediction errors, with larger errors weighted more heavily. RMSE is useful for identifying how severely a model may be under- or over-predicting player ratings.

- **Mean Squared Error (MSE):**Captures the average size of prediction errors in a linear fashion, making it more robust to outliers than RMSE. It reflects the model's consistency in typical cases.

- **R-Squared ($\mathbf{R^2}$):**Indicates the proportion of variance in player ratings explained by the model. A higher R^2 suggests that the model captures more of the true underlying relationships in the data.

These three metrics together offered a comprehensive assessment of model accuracy, stability, and explanatory power. All models were implemented using Python and the scikit-learn library. Hyperparameter tuning was performed using grid search and using 5-fold cross-validation to optimize performance.

III. RESULTS AND DISCUSSIONS

The models performances were evaluated on the test set, revealing key differences in the predictive accuracy and generalization. The analysis highlights how well each algorithm captured the underlying patterns in the data and identifies the most influential features that led to accurate rating predictions.

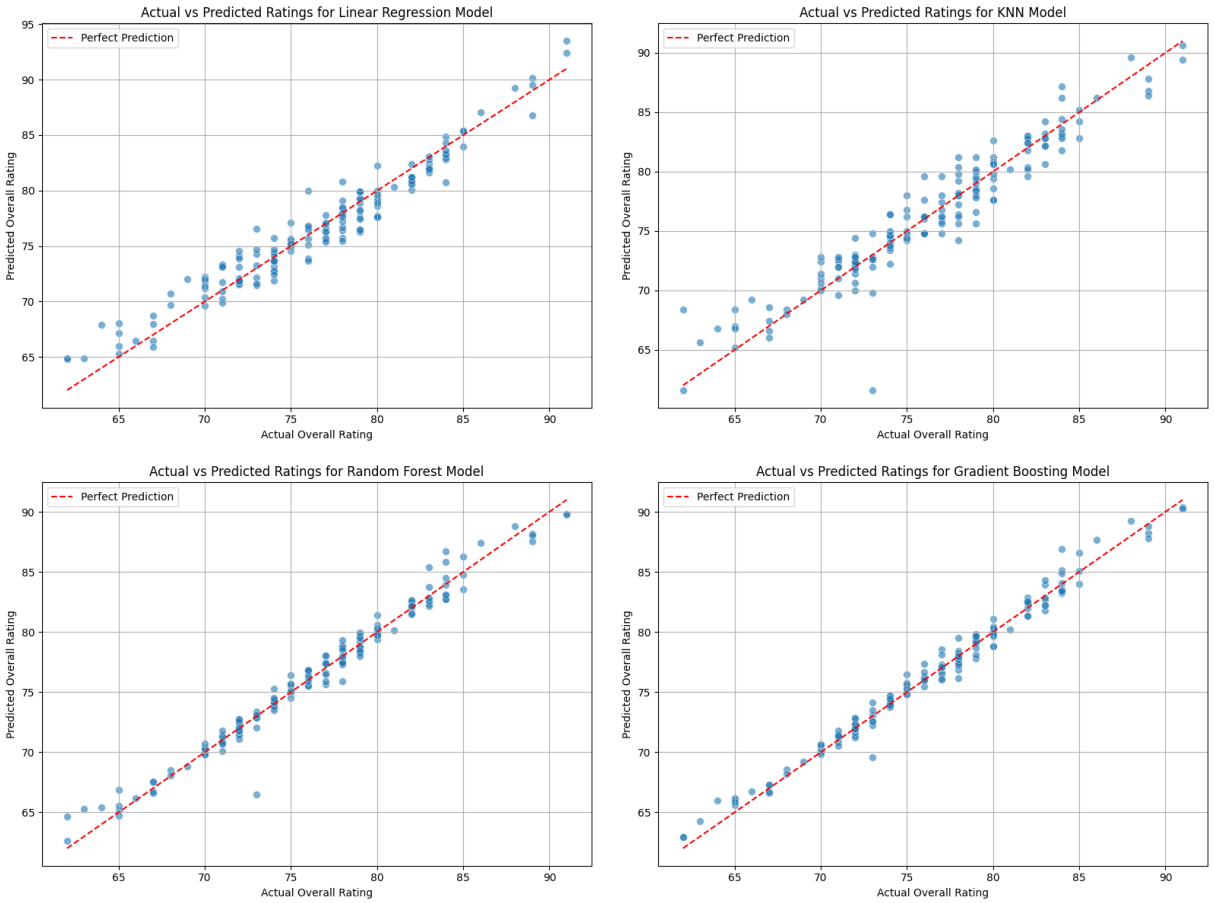

Figure 1. Actual vs. Predicted ratings for the four Models (Linear Regression, KNN, Random Forest and Gradient Boosting.

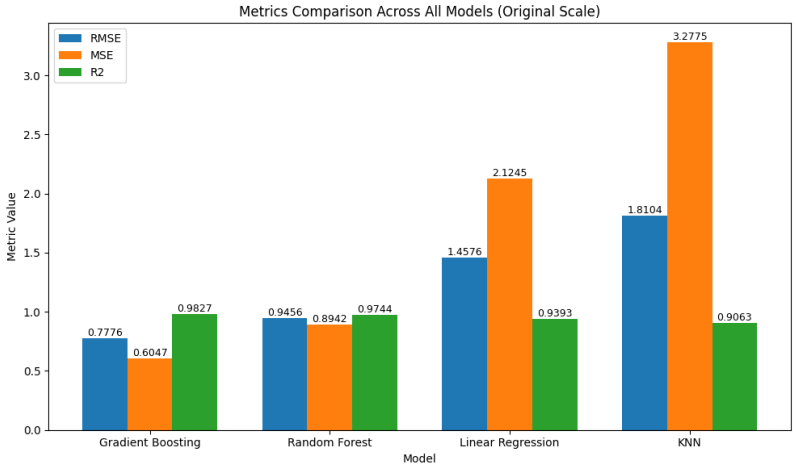

Figure 2. Metric Comparison for the different models.

The plots provided insight into the performance of each model. The best performing model would be the model that has more clustering closer to the Forty-five degree perfect prediction line from Figure.1. The

models with the best clustering would be the tree-based models Gradient Boosting and Random Forest Regressor. However, to be absolutely sure of the best model I did a graph to compare metrics to provide differentiation between the models which are seen in Figure [2] After training and evaluating across the four models, the following metric observations were made:

- **Gradient Boosting:** This Model had the lowest MSE, a low RMSE and the highest R^2 at 0.93. Gradient Boosting showed strong generalization and captured complex patterns in the data.
- **Random Forest Regressor:** This model had a slightly higher RMSE than Gradient boosting, MSE was marginally worst, however R^2(0.91) was close to that of Gradient Boosting. This model performed well but showed slight overfitting on training data.
- **Linear Regression:**This model was perfect as a baseline model to handle linearity in the data. However, the metrics proved that it failed to model nonlinearities effectively. The RMSE was higher than the previous models, MSE was moderate and R^2(0.78) was lower than the tree based models.
- **K-Nearest Neighbors:** This model was proven to be the weakest model based on the metrics. The RMSE and MSE values were the highest and the R^2 was the lowest (0.65). The performance was sensitive to k-value and feature scaling furthermore it struggled with high dimensional data.

The visuals provided from Figure [1] and Figure [2] showed showed the comparisons between the models. The comparisons from the metrics highlighted Gradient Boosting superior ability to follow true player rating trends. The tree-based models benefited from their hierarchical decisions structures, which aligned well with the feature interactions from the dataset.

Player Name	Actual Rating	Predicted Rating
L.Messi	94	91.34
Neymar Jr	92	90.31
De Gae	91	89.78
M.Salah	89	89.49
M. Neuer	89	87.32
N. Kante	89	89.22
H. Lloris	88	87.05
L. Insigne	88	87.25
Coutinho	88	88.28

Table I. Actual Ratings vs Predicted Ratings which was done using Gradient Boosting Model.

Finally, Table [I] was created using the best performing Model, Gradient Boosting, to predict the ratings of the players. The models predicted ratings closely match the actual ratings, with most differences falling within a range of two points, indicating good overall accuracy.

On average, the model slightly underestimates player ratings, particularly for top performers like L. Messi and Neymar Jr. The most accurate prediction was for M. Salah, whose predicted rating of 89.49 was very close to his actual rating of 89, while the largest discrepancy was for M. Neuer, whose rating was under predicted by about (1.7) points. Overall, the model demonstrates reliable performance with a modest deviations on the predictive vs actual scale.

IV. CONCLUSION

This project gave some insight into how machine learning models can guess the ratings of football players using performance data and other details. This project highlighted how data-based tools can help make fair player assessments, improve hiring strategies, and guide game plans. However, there are several ways in which this project could be improved, not just to help with predictions but also to help eliminate the aspect of human bias. Some of these improvement opportunities include:

- **Temporal features:** Integration of time series data to account for player form trends or player seasonal progress.

- **Neural Network:** Exploring deep learning models such as Multilayer Perceptron (MLP), may help alleviate human bias if done properly.

Finally, this project is a big step towards more repeatable, fact-based methods not just in football analysis, but in sports analysis overall. It opens the door for smart systems that fit right into how football decisions are made today.

---

[1] "Football Players Data." Www.kaggle.com, www.kaggle.com/datasets/maso0dahmed/football-players-data.

Appendix: Appendix

This is the code that was used to generate all the plots in this project. It also includes the preprocessing steps and data optimizations.

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load the dataset
df = pd.read_csv("fifa_players.csv")

# Drop irrelevant or non-numeric columns
irrelevant_columns = [
    'name', 'full_name', 'birth_date', 'positions', 'nationality',
    'preferred_foot', 'body_type', 'national_team', 'national_team_position',
    'national_jersey_number'
]
df_clean = df.drop(columns=irrelevant_columns)

# Drop rows with missing values
df_clean = df_clean.dropna()

# Define features and target
X = df_clean.drop(columns=['overall_rating'])
y = df_clean['overall_rating']

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Define models
models = {
    'Linear Regression': LinearRegression(),
    'Random Forest': RandomForestRegressor(random_state=42),
```

```
    'Gradient Boosting': GradientBoostingRegressor(random_state=42),
    'KNN': KNeighborsRegressor()
}

# Train models, evaluate, and generate scatter plots
results = {}
for name, model in models.items():
    # Train the model
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # Store evaluation metrics
    results[name] = {
        'MAE': mean_absolute_error(y_test, y_pred),
        'MSE': mean_squared_error(y_test, y_pred),
        'RMSE': np.sqrt(mean_squared_error(y_test, y_pred)),
        'R2': r2_score(y_test, y_pred)
    }

    # Generate scatter plot
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=y_test, y=y_pred, alpha=0.6, s=50) # Increased marker size (s)
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', label='P
    plt.xlabel("Actual Overall Rating")
    plt.ylabel("Predicted Overall Rating")
    plt.title(f"Actual vs Predicted Ratings for {name} Model")
    plt.grid(True)
    plt.legend() # Added legend
    plt.tight_layout()
    plt.savefig(f"actual_vs_predicted_{name.lower().replace(' ','_')}.png", format="png"
    plt.close()

# Convert results to DataFrame and display
results_df = pd.DataFrame(results).T.sort_values(by='R2', ascending=False)
print("Model Performance Comparison:\n")
print(results_df)

# Save model performance table as PNG (changed from PDF)
fig, ax = plt.subplots(figsize=(8, 4))
ax.axis('off')
table = plt.table(
    cellText=np.round(results_df.values, 4),
    colLabels=results_df.columns,
    rowLabels=results_df.index,
    cellLoc='center',
    loc='center'
)
table.auto_set_font_size(False)
table.set_fontsize(10)
table.scale(1.2, 1.5)
plt.title("Model Performance Comparison", fontsize=14, weight='bold')
plt.savefig("model_performance_comparison.png", format="png")
plt.close()

# Prepare data for grouped bar graph
models_list = results_df.index
```

```
metrics = ['RMSE', 'MSE', 'R2']
bar_width = 0.25  # Width of each bar
x = np.arange(len(models_list))  # X positions for each model

# Create grouped bar graph - PLOTTING ORIGINAL VALUES
plt.figure(figsize=(10, 6))
bars_rmse = plt.bar(x - bar_width, results_df['RMSE'], bar_width, label='RMSE', color='#1
bars_mse = plt.bar(x, results_df['MSE'], bar_width, label='MSE', color='#ff7f0e')
bars_r2 = plt.bar(x + bar_width, results_df['R2'], bar_width, label='R2', color='#2ca02c'

# Add actual values on top of each bar (keep this)
for bars, metric in zip([bars_rmse, bars_mse, bars_r2], metrics):
    for bar, model in zip(bars, models_list):
        height = bar.get_height()
        actual_value = results_df.loc[model, metric]
        plt.text(bar.get_x() + bar.get_width() / 2, height, f'{actual_value:.4f}',
                 ha='center', va='bottom', fontsize=9)

# Customize plot (keep this)
plt.xlabel("Model")
plt.ylabel("Metric Value") # Changed y-label back to "Metric Value"
plt.title("Metrics Comparison Across All Models (Original Scale)") # Changed title
plt.xticks(x, models_list, rotation=0) # Keep horizontal labels
plt.legend(loc='upper left') # Keep legend position
plt.tight_layout()
plt.savefig("metrics_comparison_original_scale.png", format="png") # Changed to png
plt.close()

# Make predictions on the test set and create a DataFrame for comparison
predictions_df = pd.DataFrame({'Actual Rating': y_test})

for name, model in models.items():
    predictions_df[f'Predicted ({name})'] = model.predict(X_test)

# Display the predictions DataFrame
print("\nPredictions on Test Set (including actual ratings):\n")
print(predictions_df.head(20)) # Displaying the first 20 predictions
```