Football Player Rating Prediction using Linear Regression

1. Introduction

This project's goal is to predict the **overall rating** of football players using their physical, technical, and mental attributes. The dataset contains various player details, from age and nationality to specific performance metrics like passing and dribbling. We chose a **Linear Regression** model for its simplicity and effectiveness in modeling the linear relationship between these attributes and the player's overall rating.

---

2. Dataset Description

The dataset is rich with player profiles, featuring a mix of personal, physical, and performance-based attributes.

- **Personal Information:** Includes name, full name, birth date, age, and nationality.
- **Physical Attributes:** Height (in cm) and weight (in kg).
- **Career Information:** Player positions and potential rating.
- **Technical and Mental Attributes (Features):** This section contains the core predictive features, such as passing, dribbling, shooting, crossing, finishing, vision, composure, positioning, aggression, interceptions, and tackling.
- **Target Variable:** The **Overall Rating**, a numerical score representing the player's overall ability.

The dataset's numerical focus makes it a great fit for a Linear Regression model.

---

3. Feature Engineering and Preprocessing

To prepare the data for the model, we followed these steps:

- **Data Cleaning:** We removed irrelevant fields, like player names and IDs, since they don't contribute to the prediction. We also handled any missing values.
- **Encoding Categorical Features:** We used **One-Hot Encoding** to convert categorical data, such as positions and nationality, into numerical formats that the regression model can understand.
- **Feature Scaling:** We applied **Standardization** to all continuous features (height, weight, skill attributes). This process scales the data to a similar range, preventing features with larger values from dominating the model.

---

4. Model Used: Linear Regression

Why we chose it: Linear Regression is easy to implement and interpret. It provides a solid baseline for understanding how well a player's attributes can explain their overall rating. It works best when the relationship between features and the target variable is roughly linear.

Mathematical Formulation: The model is represented by the following equation:

$Y = \beta 0 + \beta 1x1 + \beta 2x2 + \cdots + \beta nxn + \epsilon$

Where:

- y = The **Overall Rating**
- x1,x2,…, xn = The player's **features**
- β = The **weights** the model learns
- ϵ = The error term

The model works by minimizing the **Sum of Squared Errors (SSE)** to find the best-fitting line.

---

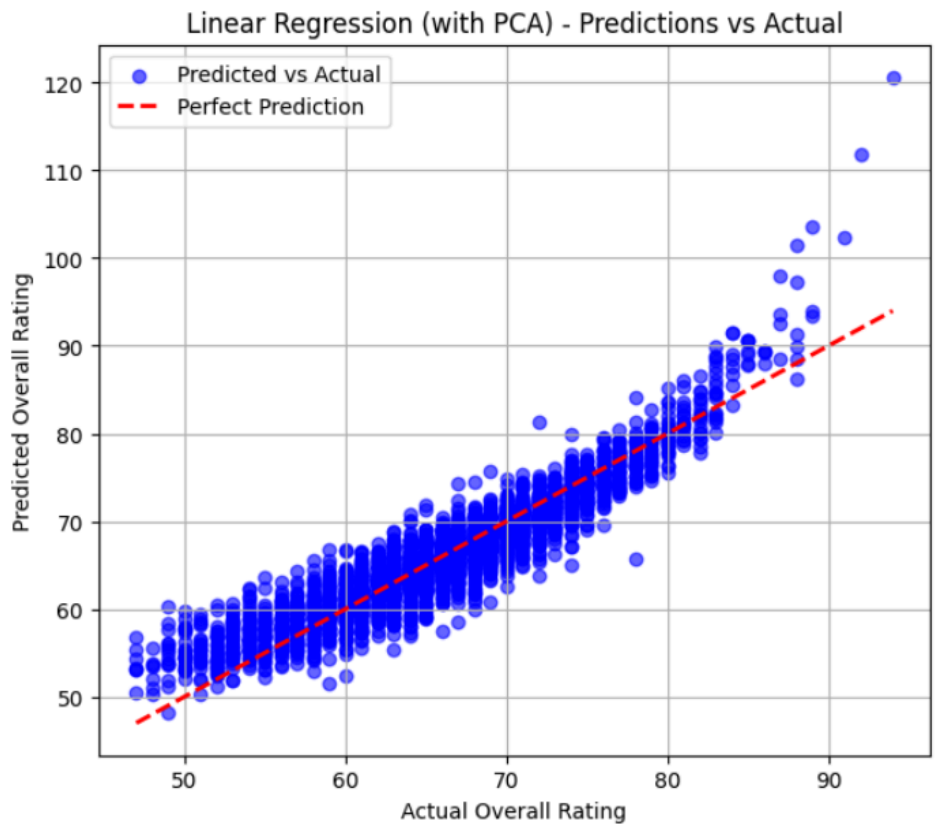
5. Evaluation Metrics

Since this is a regression problem, we used the following metrics to evaluate the model's performance:

- **R^2 Score:** This metric shows how much of the variance in the overall rating is captured by the features. A score of **0.8590** is excellent, meaning our features explain 85.9% of the variability in the ratings.
- **Root Mean Squared Error (RMSE):** This penalizes larger errors. An **RMSE of 2.65** means the average prediction error is roughly 2.6 rating points.
- **Mean Absolute Error (MAE):** This shows the average prediction error. An **MAE of 2.01** indicates that, on average, our predictions are within 2 points of the player's true rating.

---

6. Predicted vs Actual Graph



---

7. Results and Insights

The Linear Regression model performed very well, achieving a high R^2 score and low error values.

Our analysis showed that **Vision, Passing, Dribbling, Positioning,** and **Composure** are the key attributes that most strongly influence a player's overall rating. Physical attributes like height and weight had a much smaller effect compared to these technical and mental skills. This suggests that a player's quality is determined more by their skill and decision-making than by their physical build.

---

8. Conclusion

This project successfully demonstrated that Linear Regression can predict football players' overall ratings with high accuracy. The results confirm that skill-based attributes like passing and vision are the most crucial factors in determining a player's ability.

Future Improvements could include:

- Exploring **polynomial regression** to capture non-linear effects.
- Comparing performance with more advanced models, such as **Random Forest** or **XGBoost**.
- Adding more dynamic data, like match-level statistics (goals, assists, tackles), for richer predictions.

---

9. Code and Outputs

Making Imports

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

[3]:
```
file_path = '/kaggle/input/football-players-data/fifa_players.csv'
```

```
[4]: df = pd.read_csv(file_path)
     df.head()
```

[4]:

	name	full_name	birth_date	age	height_cm	weight_kgs	positions	nationality	overall_rating	potential	...	long_shots	aggression	interceptions	pc
0	L. Messi	Lionel Andrés Messi Cuccittini	6/24/1987	31	170.18	72.1	CF,RW,ST	Argentina	94	94	...	94	48	22	
1	C. Eriksen	Christian Dannemann Eriksen	2/14/1992	27	154.94	76.2	CAM,RM,CM	Denmark	88	89	...	89	46	56	
2	P. Pogba	Paul Pogba	3/15/1993	25	190.50	83.9	CM,CAM	France	88	91	...	82	78	64	
3	L. Insigne	Lorenzo Insigne	6/4/1991	27	162.56	59.0	LW,ST	Italy	88	88	...	84	34	26	
4	K. Koulibaly	Kalidou Koulibaly	6/20/1991	27	187.96	88.9	CB	Senegal	88	91	...	15	87	88	

5 rows × 51 columns

Data Cleaning

+ Code + Markdown

```
[5]: drop_cols = ['name', 'full_name', 'birth_date', 'positions', 'nationality']
     df = df.drop(columns=drop_cols, errors='ignore')

     df = df.fillna(df.mean(numeric_only=True))  # for missing data we fill with column mean

     df = pd.get_dummies(df, drop_first=True)   # One-Hot Encode categorical columns automatically
```

Defining Input and Target Features

```
[6]: X = df.drop(columns=['overall_rating'])
     y = df['overall_rating']
```

```
[7]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

     scaler = StandardScaler()
     X_train_scaled = scaler.fit_transform(X_train)
     X_test_scaled = scaler.transform(X_test)
```

Applying Sklearns PCA and Linear Regression and Training the Model

```
[8]: pca = PCA(n_components=0.95, random_state=42)
     X_train_pca = pca.fit_transform(X_train_scaled)
     X_test_pca = pca.transform(X_test_scaled)

     print(f"Original features: {X_train.shape[1]}")
     print(f"Reduced features after PCA: {X_train_pca.shape[1]}")
```

```
Original features: 181
Reduced features after PCA: 133
```

```
[9]: model = LinearRegression()
     model.fit(X_train_pca, y_train)
```

```
[9]: ▾ LinearRegression
     LinearRegression()
```

Model Evaluation

[10]:

```
y_pred = model.predict(X_test_pca)

r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
mae = mean_absolute_error(y_test, y_pred)

print(f"R² Score: {r2:.4f}")
print(f"RMSE: {rmse:.4f}")
print(f"MAE: {mae:.4f}")
```

```
R² Score: 0.8590
RMSE: 2.6548
MAE: 2.0101
```

[12]:

```
import pickle

with open("linear_regression_model.pkl", "wb") as f:
    pickle.dump(model, f)
```
