AI Football Performance Analyzer

Members:

- **Adrien Schuttig**, ECE Paris, adrienschuttig@gmail.com
- **Maxime Laurent**, ESILV Nantes, maxime.laurent@edu.devinci.fr
- **Louis Le Forestier**, ESILV Nantes, louis.le_forestier@edu.devinci.fr

---

Table of Contents

I. Introduction

Motivation: Why are we doing this?

Football is a data-rich sport where analyzing player performance can significantly impact scouting, training, and match strategy. We aim to leverage machine learning to move beyond subjective observation and provide data-driven insights into player capabilities and potential.

What do we want to see at the end?

We aim to build a system that can:

1. **Analyze** current player statistics to evaluate their overall performance.
2. **Predict** a player's future development trajectory (e.g., will they improve, stay stable, or decline?).
3. **Visualize** these insights in an accessible way for coaches and analysts.

II. Datasets

We are using the **Football Players Data** dataset from Kaggle for this study.

Source and Description

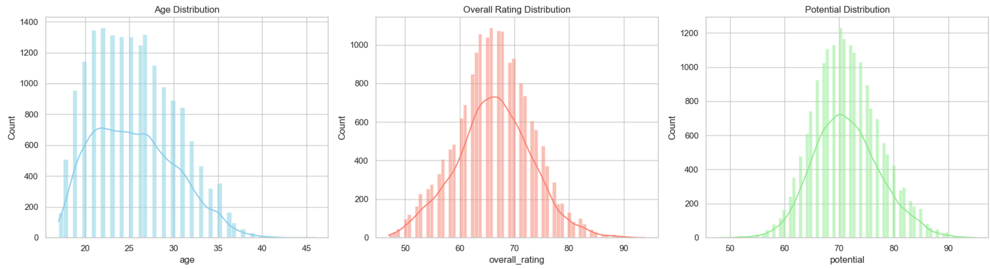
Attribute	Details
Source	Kaggle Link: Football Players Data
Size	17,954 rows (players)
Columns	51 attributes (physical, technical, mental, ratings)

Exploratory Data Analysis (EDA)

The EDA helped in understanding the data structure and identifying key relationships for modeling.

1. Skill Distribution

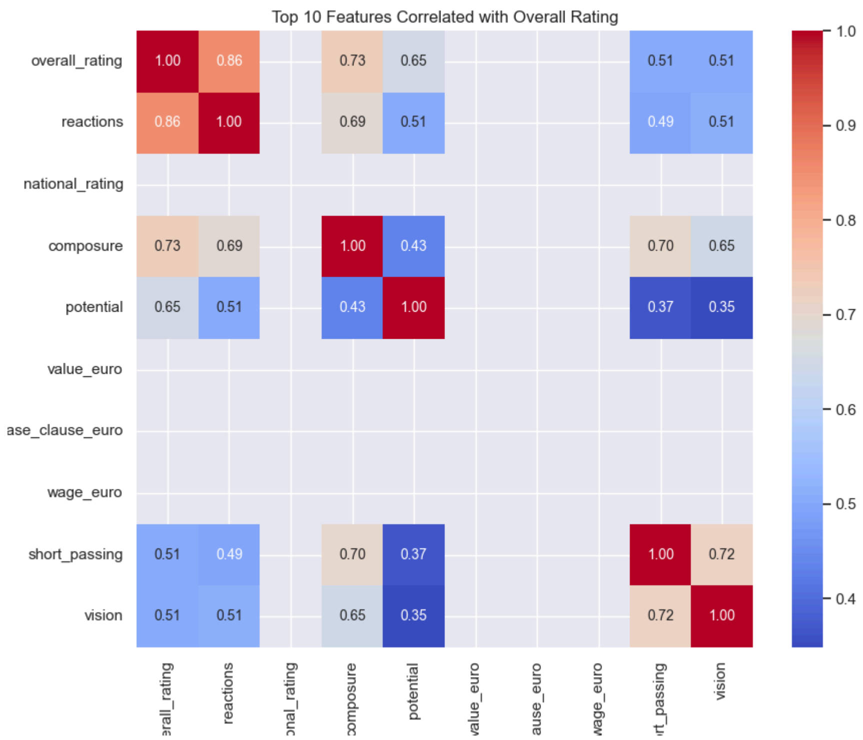
The distribution of ratings (`overall_rating` and `potential`) is **strongly skewed to the right**, indicating that the majority of players fall into the low to medium rating categories.



2. Correlation with Performance (`overall_rating`)

The correlation matrix highlights the attributes that most influence the overall rating.

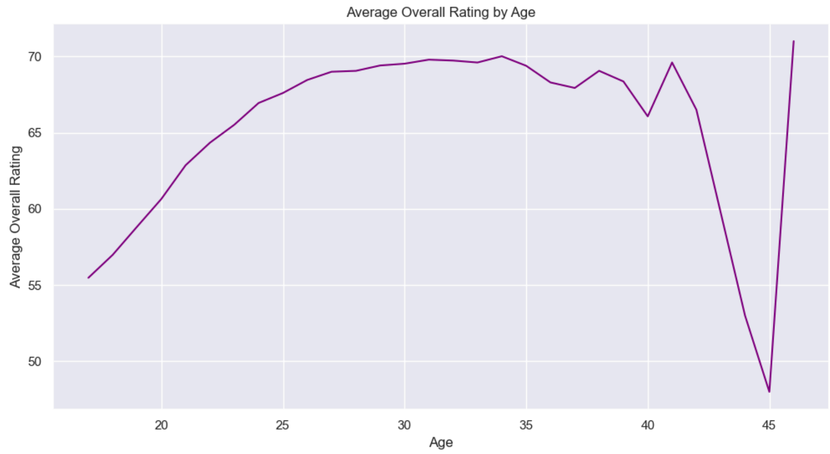
Attribute	Correlation	Observation
`reactions`	**0.84**	The most predictive attribute, emphasizing the importance of rapid decision-making.
`potential`	0.69	Indicates a strong influence of future prospects on the current assessment.
`composure`	0.68	A key mental factor for performance.
`short_passing`	0.59	Core technical skills are crucial.



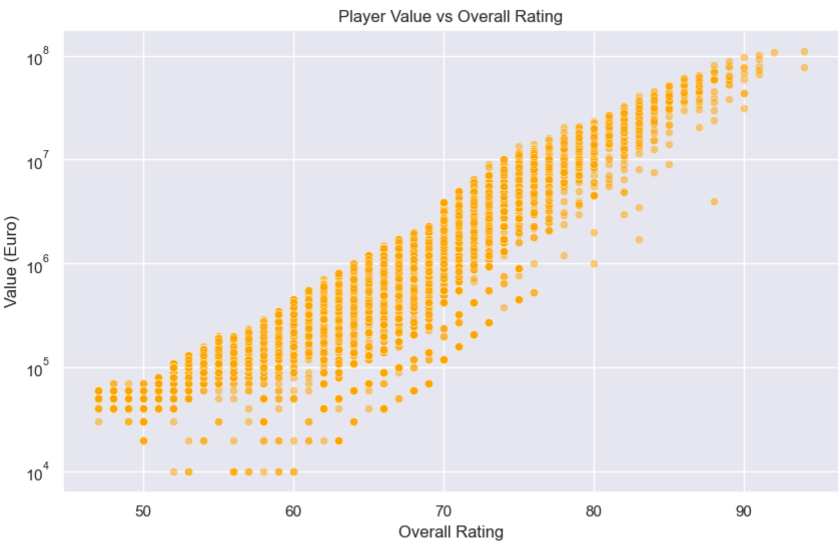
3. Age, Value, and Nationality Relationship

Age analysis confirms that the average player level (`overall_rating`) **peaks between 27 and 31 years** before declining.

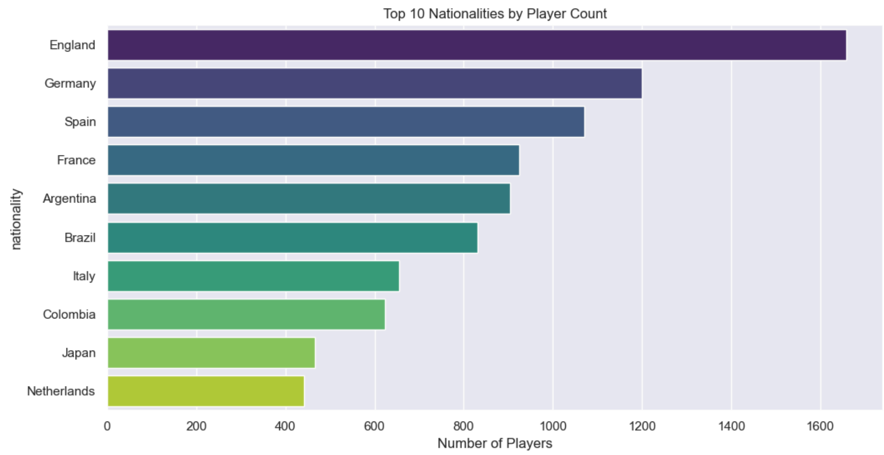
- **Age vs Rating:**



- **Value vs Rating:** Market value (`value_euro`) is highly correlated with the overall rating, increasing exponentially, as shown on the logarithmic scale.



- **Top Nationalities: Spain** (1,341 players), **Argentina**, and **France** are the most represented countries.



Critical Data Assessment

Category	Positive Points	Negative Points
Feature Richness	✅ **51 varied attributes** (physical, technical, mental) for comprehensive analysis.	❌ High **Multicollinearity** (several attributes are highly correlated with each other), which will necessitate rigorous feature selection.
Modeling Objective	✅ `potential` is an excellent indicator for future growth prediction.	❌ The `overall_rating` relies heavily on the `reactions` variable, which could bias the model if overused, at the expense of other skills.

III. Methodology

We implemented a dual-model approach to analyze both current ability and future potential.

1. Feature Engineering: Defining Future Growth

To predict a player's trajectory, we first needed to define what "growth" means. We created a custom target variable `future_class` based on the gap between a player's `potential` and their current `overall_rating`, while also considering their `age`.

Code Implementation:

```
def build_future_label(row):
    """
    Categorizes a player's future growth potential.
    """
    gap = row["potential"] - row["overall_rating"]
    age = row["age"]

    # Young players with huge potential gap
    if gap >= 10 and age <= 23:
        return "high_growth"
    # Players with significant room for improvement
```

```python
    elif gap >= 4:
        return "likely_improve"
    # Players near their peak
    elif gap >= -2:
        return "stable"
    # Players in decline
    else:
        return "decline"
```

2. Algorithm Selection & Implementation

We utilized `scikit-learn` to implement two distinct models, chosen for their interpretability and effectiveness on tabular data.

A. Linear Regression (Current Performance)

Goal: Predict the continuous `overall_rating`. **Why:** Linear Regression allows us to quantify exactly how much each specific skill (e.g., +1 in Dribbling) contributes to the overall rating.

```python
# 1. Select Features
X = df_clean[feature_cols] # Age, Physical & Technical stats
y = df_clean["overall_rating"]

# 2. Split Data (80% Train, 20% Test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# 3. Train Model
reg_model = LinearRegression()
reg_model.fit(X_train, y_train)
```

B. Logistic Regression (Future Classification)

Goal: Classify players into one of the 4 growth categories. **Why:** Logistic Regression provides not just a classification, but the *probability* of a player belonging to each class, which is crucial for risk assessment in scouting.

```python
# 1. Prepare Target
y_cls = df_clean["future_class"]

# 2. Train Model (Multinomial for multi-class classification)
clf = LogisticRegression(max_iter=1000, multi_class="multinomial")
clf.fit(X_train, y_cls)

# 3. Predict Probabilities
future_proba = clf.predict_proba(example_player_stats)
```

3. Key Features Used

We focused on attributes with the highest correlation to performance, avoiding overfitting by selecting a balanced subset:

- **Physical:** `age`, `height_cm`, `weight_kgs`, `acceleration`, `sprint_speed`, `stamina`, `strength`

- **Technical:** `finishing` , `dribbling` , `short_passing`

IV. Evaluation & Analysis

Regression Results (Predicting Overall Rating)

- **Model:** Linear Regression
- **Metrics:**
    - **MSE (Mean Squared Error):** Measures the average squared difference between estimated values and the actual value.
    - **R^2 Score:** Indicates how well the data fit the regression model.
- **Visualization:** We generated a scatter plot (`reg_true_vs_pred.png`) comparing true ratings vs. predicted ratings. A tight clustering around the diagonal indicates high accuracy.

Classification Results (Predicting Future Growth)

- **Model:** Logistic Regression (Multinomial)
- **Classes:**
    - `high_growth` : Young players with a large potential gap.
    - `likely_improve` : Players with significant room for improvement.
    - `stable` : Players near their peak.
    - `decline` : Players whose potential is lower than their current rating.
- **Metrics:** We use Precision, Recall, and F1-Score to evaluate the classifier's performance across all classes.

V. Related Work

- **Libraries Used:**
    - `pandas` : For data manipulation and cleaning.
    - `scikit-learn` : For implementing Linear and Logistic Regression models.
    - `matplotlib` / `seaborn` : For data visualization.
- **References:**
    - Scikit-learn Documentation: https://scikit-learn.org/
    - Kaggle Dataset: Football Players Data

VI. Conclusion: Discussion

This project demonstrates that standard player attributes can effectively predict both current ability and future potential.

- **Findings:** Physical stats combined with technical skills like passing and dribbling are strong predictors of a player's overall rating.
- **Future Work:** We could enhance the model by:
    - Incorporating match performance data (goals, assists per game).
    - Using more complex models like Random Forests or Neural Networks for non-linear relationships.
    - Building a web interface (Streamlit) to allow users to input player stats and get real-time predictions.