Report: FIFA21 Player Value Prediction Analysis

Sayan Sinha
2023BCY0006

1. **Objective**
   The primary goal of this analysis was to build a predictive model capable of categorizing a FIFA21 player's market value. Instead of predicting an exact monetary value (which is highly volatile and difficult to model as a specific number), we aimed to create a classification model. This model assigns players to one of five distinct value categories: Very Low, Low, Medium, High, or Very High. This approach provides a more stable and practical assessment of a player's worth.
2. **Data Cleaning and Preparation**
   Before any analysis, the raw dataset (17,125 players, 107 features) required significant cleaning to convert it into a format usable by a machine learning model. Computers require standardized, numerical data, but the dataset contained text, symbols, and mixed units.

What Was Done:

- **Dropped Irrelevant Data:** Columns that do not provide predictive information, such as player ID, Name, and photo links, were removed.
- **Cleaned Monetary Values:** Player Value and Wage were stored as text (e.g., "€1.1M" or "€625K"). These were converted into numerical values (e.g., 1,100,000 or 625,000). Players with a value or wage of €0 (such as free agents) were removed.
- **Standardized Physical Attributes:** Player Height (e.g., "6'0"") and Weight (e.g., "181lbs") were converted into standardized numeric units: centimeters (cm) and kilograms (kg).
- **Cleaned Star Ratings:** Features like W/F (Weak Foot) were stored as "3 ★". The "★" symbol was removed to leave just the number.
- **Handled Missing Data:** Many players had missing data.
  - **Numeric Gaps:** Filled with the **median** (the middle value) of that column. This is better than the **mean** (average) because it isn't skewed by a few superstars with extremely high stats.
  - **Text-based Gaps:** Filled with the **mode** (the most frequent value) of that column. This is a common strategy to fill categorical gaps with the most probable value rather than leaving them blank.

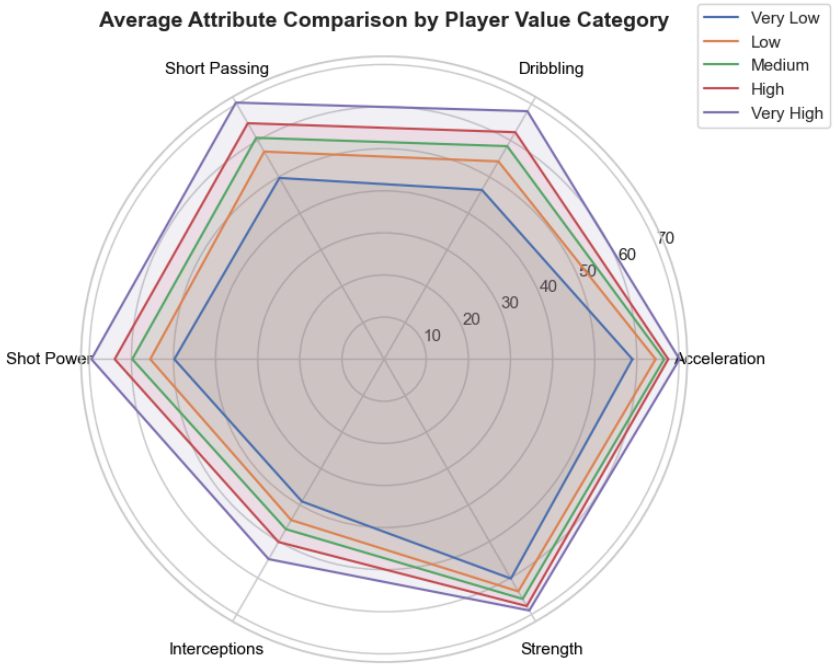
Impact:

This cleaning phase was critical. It transformed the messy, mixed-data spreadsheet into a clean, fully-numeric, and standardized dataset, which is the necessary foundation for any successful predictive model.

3. **Key Findings from Data Visualization**
   After cleaning, we analyzed the data visually to understand the relationships between player attributes and their market value.

- **Player Value vs. Age:** The analysis confirmed that a player's value follows a distinct life cycle, tending to rise sharply between ages 20 and 30 (a player's "peak") and then gradually declining. This confirms that **Age** is a critical factor in player valuation.
- **Key Predictors:** We found that OVA (Overall Rating), POT (Potential), and **Wage_num** have a very strong, clear positive correlation with value. As a player's current ability and future potential increase, their market value rises exponentially.
- **Wage** exhibits a **weaker correlation** — while higher wages generally align with higher player values, the trend is less steep. This may be because wages depend on club budgets, contracts, and league economics, not just market valuation.

- **Most Important Attributes (Radar Plot)**



  - **Finding:** We grouped players by their value category (Very Low to Very High) and compared their average stats for key skills. This radar plot clearly shows what attributes the market values most.
  - **Key Insight:** The biggest gaps between value tiers were in **technical skills**, specifically **Dribbling, Short Passing, and Shot Power**. This means the market heavily rewards players who can control the ball, pass accurately, and convert chances.
  - **Insight 2:** Physical stats like **Acceleration** and **Strength** were very similar for both "High" and "Very High" value players. This suggests that while physical ability is required to reach the top, it's technical skill that separates the "great" players from the "elite" superstars and commands the highest market values.

4. **Building the Prediction Model**

With a clean and understood dataset, we proceeded to build the models.

- **Feature Engineering**
  - Text-based features like Club and Nationality have hundreds of unique values. We converted them into a number representing how "frequent" (or common) that club or nationality is in the dataset.
  - This gives the model a simple numerical way to understand if a player is from a large club or a major footballing nation, which can influence their value.
- **Defining the Target (Classification)**
  - The model's goal is to **classify** players into one of five value categories. We created this target variable, Value_q, by splitting the numerical player values into five equal-sized groups (quintiles).
  - **Why:** Using quintiles ensures that each of the 5 categories (Very Low, Low, Medium, High, Very High) has an equal number of players. This is a crucial step to prevent "class imbalance", a common problem where a model becomes biased simply because most of the data falls into one category.

- **Preparing Data for the Model (Preprocessing)**
  - The final step was to convert all remaining text features (like 'Best Position' and 'Foot') into a numerical format that the model could understand.
  - We used an automated pipeline to fill any remaining numeric gaps and convert text to numbers, producing the final, fully-numeric dataset used for training and testing.

5. **Model Comparison and Final Conclusion**

We trained and tested two powerful, industry-standard classification models to see which could most accurately predict a player's value category.

- **Random Forest (RF):** An ensemble model that builds hundreds of "decision trees" and takes a vote. It is very robust and good at preventing errors.
- **XGBoost (XGB):** A more advanced ensemble model that builds trees sequentially, with each new tree learning from the mistakes of the previous one. It is often a top performer.

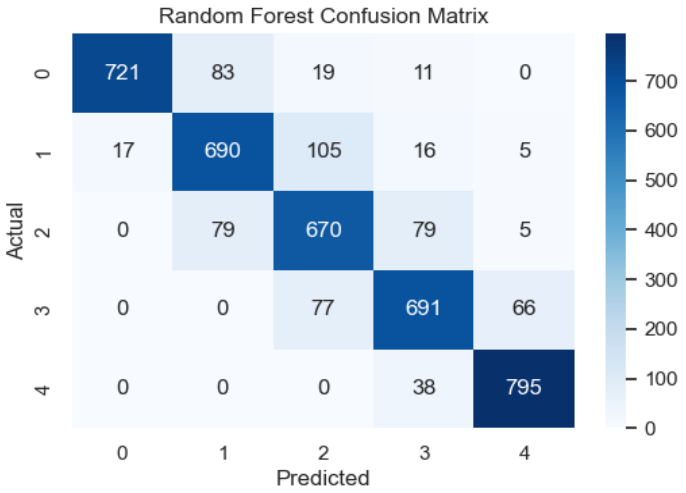
The data was split into a **Training Set (75%)** to teach the models and a **Testing Set (25%)** to evaluate their performance on unseen data.

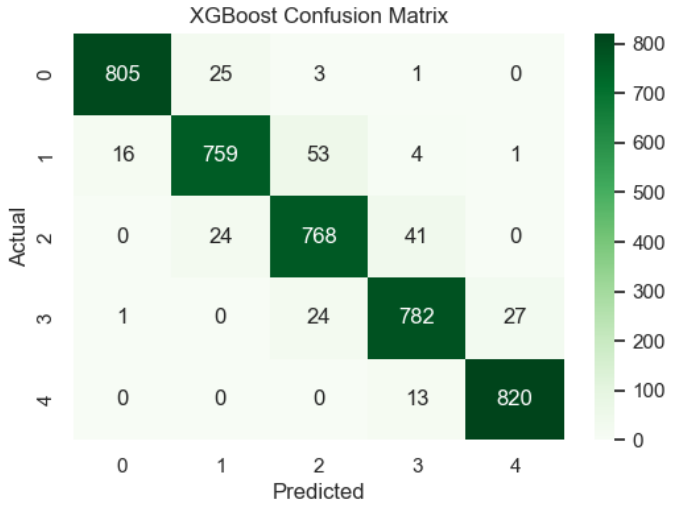
The Results

Model	Training Accuracy	Testing Accuracy	Overfitting Gap
Random Forest	**90.06%**	**85.60%**	**4.46%**
XGBoost	**97.36%**	**94.41%**	**2.95%**

Analysis

- **Testing Accuracy (The "Real" Score):** This is the most important metric. The XGBoost model was **significantly more accurate**, correctly predicting the value category for **94.41%** of players in the test set, compared to **85.60%** for the Random Forest.
- **Overfitting Gap:** This gap measures how much the model "memorized" the training data versus "learning" the general pattern. A smaller gap is much better, as it means the model will be more reliable on new data. The **XGBoost model had a tiny gap of 2.95%** (compared to RF's 4.46%), indicating it is very stable and trustworthy.
- **Precision/Recall:** Additionally, the macro average precision, recall, and f1-score was **0.86** for Random Forest and **0.94** for XGBoost, which is much more preferable.
- **Confusion Matrix:** A visual analysis (confusion matrix) of both models' errors confirmed this. The XGBoost model's predictions (the diagonal line) were much clearer and more accurate, while the Random Forest model made more mistakes in classifying players between adjacent categories.





Conclusion

Looking at all this information, we can conclude that using an **XGBoost model** with a testing accuracy of **0.9441** is the best choice for this dataset.