# ***Data Science Project***

### **1.1 Domain-Specific Area**

This project delves into the field of soccer analytics, leveraging the FIFA 19 dataset sourced from Kaggle. Soccer analytics employs data-driven methodologies to enhance player performance, refine team strategies, and support decision-making processes within the sport. By thoroughly analyzing player attributes and ratings from FIFA 19, this project aims to uncover the factors that influence virtual player performance in the gaming environment. The insights gained can potentially be translated into real-world soccer scenarios, providing valuable information for coaches, analysts, and enthusiasts.

### **1.2 Objectives**

The primary objectives of this project are centered around comprehensive data exploration and the development of predictive models using linear regression and KMeans clustering with dimensionality reduction to predict player’s overall attribute:

1. **Exploratory Data Analysis (EDA)**:
   * **Objective**: Conduct an extensive exploration of player attributes, such as dribbling, shooting, and physical characteristics (e.g., height, weight), to understand their relationships with player ratings in FIFA 19.
   * **Approach**: Utilize various visualization techniques and statistical analyses to identify patterns, correlations, and potential anomalies within the dataset. This step aims to provide a solid foundation for further modeling efforts by revealing key insights and trends.
2. **Predictive Modeling with Linear Regression**:
   * **Objective**: Develop a predictive model using linear regression to identify significant attributes that influence player ratings.
   * **Approach**: Apply linear regression techniques to quantify how individual attributes collectively contribute to a player's overall performance score in the game. This model will help in understanding the impact of each attribute on player ratings and provide a basis for potential improvements in player development strategies.
3. **Dimensionality Reduction with KMeans Clustering**:
   * **Objective**: Reduce the dimensionality of the dataset using KMeans clustering to group similar player attributes into clusters.
   * **Approach**: Implement KMeans clustering to enhance computational efficiency and potentially improve the performance of predictive models. By grouping similar attributes, this technique aims to uncover underlying patterns and relationships within the dataset that may not be apparent through traditional analysis methods.

### **1.3 Why Linear Regression and KMeans?**

**Linear Regression**:

* **Advantages:**

Simplicity: Linear regression is straightforward to implement and interpret, making it an ideal starting point for analyzing the relationships between player attributes and ratings.

Interpretability: The coefficients obtained from linear regression models provide clear insights into the influence of each attribute on player ratings, allowing for easy interpretation and communication of results.

Baseline Insights: Establishes foundational insights into the linear relationships between variables, which can guide further, more complex analyses.

* **Disadvantages:**

Linearity Assumption: Assumes a linear relationship between the dependent and independent variables, which may not always hold true.

Outlier Sensitivity: Can be sensitive to outliers, which may distort the model's accuracy and predictive power.

**KMeans Clustering**:

* **Advantages**:

1. **Dimensionality Reduction**: Efficiently reduces the complexity of high-dimensional datasets by grouping similar attributes into clusters, enhancing computational efficiency.
2. **Pattern Discovery**: Helps in discovering underlying patterns and relationships within the data that might not be evident through other techniques.
3. **Versatility**: Can be applied to various types of data and problems, making it a flexible tool for exploratory analysis.

* **Disadvantages:**

1. **Cluster Determination:** Requires specifying the number of clusters (k) beforehand, which can be challenging without prior knowledge of the data structure.
2. **Scalability:** May not perform well with very large datasets or datasets with many dimensions, necessitating further optimization or alternative methods.

### **1.4 Potential Uses in a Data Science Project**

1. **Player Performance Analysis**:
   * Use overall and potential ratings to evaluate player performance.
   * Compare players across different clubs, nationalities, and positions.
2. **Market Value Prediction**:
   * Use regression models to predict player market value based on their attributes and performance ratings.
3. **Skill Development Tracking**:
   * Analyze how players' skills develop over time and identify factors influencing player growth.
4. **Clustering and Classification**:
   * Group players into clusters based on their attributes and positions to identify patterns and archetypes in player types.
5. **Contract Analysis**:
   * Evaluate the correlation between players' wages, market values, and their contract duration and terms.
6. **Visualization**:

* Create visualizations to compare attributes, performance metrics, and financial information across different players and teams.

### **Overview of the Dataset**

The dataset appears to be a collection of attributes and statistics for professional football players, likely sourced from a popular football simulation video game such as FIFA. It includes detailed information about each player, ranging from personal details to skill ratings, physical attributes, and contractual data.

### **2.2 Key Features and Attributes**

#### **Player Information**

* **ID**: Unique identifier for each player.
* **Name**: Full name of the player.
* **Age**: Age of the player.
* **Photo**: URL linking to the player's photo.
* **Nationality**: Country the player represents.
* **Flag**: URL linking to the national flag image.
* **Overall**: The player's overall rating.
* **Potential**: The player's potential rating.
* **Club**: The club the player is currently playing for.
* **Club Logo**: URL linking to the club's logo.
* **Value**: The player's market value.
* **Wage**: The player's weekly wage.
* **Special**: A special rating metric.
* **Preferred Foot**: Dominant foot (left or right).
* **International Reputation**: Rating of international fame (1-5).
* **Weak Foot**: Rating of the player's weaker foot ability (1-5).
* **Skill Moves**: Rating of the player's skill moves (1-5).
* **Work Rate**: Player's work rate for attack and defense (e.g., High/Medium).
* **Body Type**: Description of the player's body type (e.g., Lean, Normal).
* **Real Face**: Whether the player’s face is realistically represented in the game (Yes/No).

#### **Position and Contract Details**

* **Position**: Primary playing position (e.g., ST for Striker).
* **Jersey Number**: Player's shirt number.
* **Joined**: Date the player joined the club.
* **Loaned From**: Club the player is loaned from, if applicable.
* **Contract Valid Until**: End date of the player's current contract.
* **Height**: Player's height.
* **Weight**: Player's weight.

#### **Skill Ratings (by Position)**

The dataset includes specific skill ratings for various positions on the field, represented by abbreviations (e.g., LS, ST, RW, CAM). Each position has a corresponding skill rating indicating the player's proficiency in that role.

#### **Technical Attributes**

* **Crossing, Finishing, HeadingAccuracy, ShortPassing, Volleys**: Attributes related to the player's technical abilities with the ball.
* **Dribbling, Curve, FKAccuracy, LongPassing, BallControl**: Further breakdown of specific technical skills.
* **Acceleration, SprintSpeed, Agility, Reactions, Balance**: Attributes related to the player's physical and reactive abilities.
* **ShotPower, Jumping, Stamina, Strength, LongShots**: Attributes that define physical and shooting strength.
* **Aggression, Interceptions, Positioning, Vision, Penalties, Composure**: Mental and tactical attributes.

#### **Defensive Attributes**

* **Marking, StandingTackle, SlidingTackle**: Defensive skills ratings.
* **GKDiving, GKHandling, GKKicking, GKPositioning, GKReflexes**: Goalkeeping skills ratings.

#### **Contract and Financial Information**

* **Value**: Market value of the player.
* **Wage**: Weekly wage of the player.
* **Release Clause**: The amount for which the player can be bought out of their contract.

### ***2.3 Dataset Size and Data Types***

#### **Dataset Size**

The dataset consists of 18,207 rows (each representing a player) and 89 columns (each representing an attribute of the player).

#### **Data Types**

1. **Numerical Data:** Age, Overall, Potential, Value, Wage, various skill ratings, and physical attributes such as Height and Weight.
2. **Categorical Data:** Name, Nationality, Club, Preferred Foot, Work Rate, Body Type, Real Face, Position, Joined, Loaned From, Contract Valid Until.
3. **Textual Data:** URLs for Photo, Flag, and Club Logo.
4. **Rating Scales:** International Reputation, Weak Foot, Skill Moves.

#### **2.4 Source of the Dataset**

- The dataset is sourced from Kaggle and can be accessed at the following [URL](https://www.kaggle.com/datasets/javagarm/fifa-19-complete-player-dataset/data).

The overarching purpose of this project is to enhance our understanding of the factors that influence soccer player ratings in FIFA 19, specifically aiming to predict the overall player rating. By applying linear regression and KMeans clustering for dimensionality reduction, we aim to achieve more accurate predictions and better insights into the key attributes driving player performance. This approach not only aids in virtual soccer analytics but also provides a methodological framework applicable to broader sports analytics contexts.  
  
**3.1 First Normal Form (1NF):**

First Normal Form (1NF) is a fundamental concept in database design that ensures the elimination of repeating groups and atomicity of values within a relational database table. Here are the key steps and characteristics of achieving 1NF:

**1. Atomic Values:** Each column in a table must contain atomic (indivisible) values. This means that a column should not contain multiple values or arrays. For example, a column like "Skills" should not contain multiple skills in a single cell.

**2. Unique Column Names:** Each column name in a table should be unique and should represent a single attribute of the entity the table is modeling. This ensures clarity and avoids ambiguity in data representation.

**3. No Repeating Groups:** A table should not contain repeating groups of columns that are related. For instance, if a player can have multiple positions (e.g., striker, midfielder), each position should be in a separate row, not listed as comma-separated values in a single cell.

**4. Primary Key:** Each table should have a primary key that uniquely identifies each row in the table. This helps in uniquely identifying records and ensuring that each row is distinct.

### ***Application to FIFA 19 Dataset:***

The FIFA 19 dataset, sourced from Kaggle, consists of various attributes and statistics for soccer players. Each row in the dataset represents a single player, and each column represents a specific attribute such as player ID, name, age, nationality, club, and various skill ratings. Let's analyze if the dataset adheres to 1NF principles:

- **Atomic Values:** Most columns in the dataset contain atomic values. For example, the "Name" column contains the player's full name, and the "Age" column contains numeric values representing the player's age. Each attribute is represented as a single value per cell.

- **Unique Column Names:** The column names such as "Name", "Age", "Nationality", etc., are unique and represent single attributes of the players.

- **No Repeating Groups:** There are no repeating groups of columns in the dataset. Each column represents a distinct attribute of the player, and there are no arrays or lists of values stored within a single cell.

- **Primary Key:** The dataset includes a unique identifier for each player, represented by the "ID" column. This serves as the primary key, ensuring that each player record can be uniquely identified.

### **3.2 Preprocessing**

***Part 1: Initial Data Exploration and Cleaning***

The preprocessing journey begins with an initial exploration and cleaning phase. Here's a breakdown of the steps taken:

**1. Loading and Inspecting the Data:**

- The dataset, initially stored in a CSV file, is loaded using the pandas library. Initial inspection includes examining the first few rows, dimensions, column names, and basic summary statistics.

**2. Handling Missing Values:**

- Identification and handling of missing values are crucial. Columns with excessive missing values that do not contribute significantly to the prediction task, such as 'Loaned From', 'Release Clause', and 'Joined', are dropped. Rows with exactly 48 missing values across columns are removed, ensuring data integrity.

**3. Splitting Data into Training and Test Sets:**

- The dataset is split into training and test sets using a stratified approach to ensure representation across different player attributes. This step prepares us for subsequent model training and evaluation.

***Part 2: Feature Engineering and Transformation***

Feature engineering involves transforming raw data into features that better represent the underlying problem to the predictive models. Here’s how it was carried out:

**1. Handling Text and Categorical Data:**

- Custom transformations are applied to text and categorical columns using a PreprocessingTextTransformer class. This includes converting the 'Real Face' column to a binary indicator ('Real\_Face'), determining major nationalities for one-hot encoding ('Major\_Nation'), simplifying player positions ('Simple\_Position'), and splitting 'Work Rate' into two separate columns ('WorkRate1', 'WorkRate2'). Irrelevant columns like 'Body Type', 'Weight', 'Height', 'Contract Valid Until', and 'Name' are dropped.

**2. Handling Numerical Data:**

- A NumericalColDropper class drops irrelevant numerical columns such as 'ID', 'Jersey Number', and 'Special'. The remaining numerical columns are then imputed using the median strategy to handle missing values and standardized using StandardScaler to ensure all features are on the same scale.

***Part 3: Creating Preprocessing Pipelines***

Pipelines are constructed to automate and sequence these preprocessing steps for both training and test datasets:

**1. Numerical Pipeline:**

- A pipeline (numPipeline) is created to drop irrelevant numerical columns, impute missing values, and standardize numerical features. The SimpleImputer is used with a median strategy to handle missing values, while StandardScaler ensures numerical features have a mean of 0 and a variance of 1.

**2. Categorical Pipeline:**

- Another pipeline (catPipeline) processes text and categorical data. It utilizes the PreprocessingTextTransformer to perform custom transformations such as converting columns to binary indicators and simplifying player positions. The OneHotEncoder is then applied to one-hot encode categorical features, handling unknown categories gracefully.

**3. Combining Pipelines with ColumnTransformer:**

- The ColumnTransformer combines the numerical and categorical pipelines (numPipeline and catPipeline, respectively). It specifies which transformations to apply to numerical and categorical columns separately, preparing the data for model training.

**4. KMeans Preprocessing Pipeline:**

The KMeans preprocessing pipeline integrates KMeans clustering to transform the data based on cluster distances. This approach is beneficial for capturing complex relationships and patterns within the dataset.

* **preprocessPipeline:** This component encompasses the combined preprocessing steps for both numerical and categorical data using ColumnTransformer. It includes operations such as dropping irrelevant columns, imputing missing values, standardizing numerical features, and encoding categorical variables.

* **MiniBatchKMeans:** MiniBatchKMeans clustering is employed with 50 clusters and 1 initialization. This variant of KMeans clustering is chosen for its efficiency with large datasets, processing data in mini-batches to reduce computational overhead while maintaining clustering accuracy.

The KMeans preprocessing pipeline transforms the dataset by assigning each instance to one of the 50 clusters based on feature similarities. This transformation facilitates enhanced feature representation, potentially improving the performance of subsequent machine learning models by uncovering underlying data patterns.

**5. Reduced Dimension KMeans Pipeline:**

The reduced dimension KMeans pipeline further refines data preprocessing by leveraging KMeans clustering to reduce the dataset's dimensionality:

* **preprocessPipeline:** Similar to the KMeans preprocessing pipeline, this pipeline integrates the preprocessing steps tailored for numerical and categorical data.
* **MiniBatchKMeans:** This time, MiniBatchKMeans is configured with an optimized number of clusters (bestParam) determined through hyperparameter tuning. The parameter n\_init=1 specifies a single initialization, optimizing processing efficiency while ensuring robust cluster representation.

By applying MiniBatchKMeans with a refined number of clusters, the reduced dimension KMeans pipeline transforms the dataset into a lower-dimensional space. This transformation aims to preserve essential data characteristics while reducing noise and complexity, potentially improving the model's ability to generalize and predict player ratings accurately.

These advanced preprocessing pipelines demonstrate a structured approach to enhancing feature representation and reducing data dimensionality using KMeans clustering techniques. By integrating these pipelines into the data preprocessing workflow, we aim to optimize the dataset for subsequent machine learning model training and evaluation tasks.

### **3.4 Dataset File Type/Format**

The dataset utilized for this analysis is stored in a structured format known as a Comma-Separated Values (CSV) file. CSV is a widely adopted file format for storing tabular data in plain text, where each line of the file represents a row of data, and values within each row are separated by commas. CSV files are preferred for their simplicity, compatibility across various software applications, and ease of parsing and handling large datasets.

The dataset contains a comprehensive set of attributes and metrics related to football players, including personal details, performance statistics, and various player attributes. Each column in the CSV file represents a specific attribute or feature, such as player age, nationality, overall rating, and physical characteristics.

Utilizing the CSV format ensures that the dataset can be easily loaded into analytical tools, such as pandas in Python, for data exploration, preprocessing, and machine learning model development. The structured nature of CSV files facilitates efficient data manipulation, transformation, and analysis, making it suitable for tasks involving predictive modeling and statistical analysis in the domain of sports analytics.

This file format choice aligns with industry standards for data storage and ensures accessibility and usability across different computational environments and analytical workflows.

### ***3.5 Conclusion***

In conclusion, the preprocessing steps outlined in this report are essential for ensuring the dataset is ready for predictive modeling. By handling missing values, dropping irrelevant columns, and transforming features into a format suitable for machine learning algorithms, we improve the robustness and accuracy of our predictive models. These steps lay the foundation for building and evaluating models that predict football player ratings based on their attributes effectively.

## **4. Statistical Analysis**

This report provides a comprehensive statistical analysis of selected player attributes from a dataset. The analysis focuses on central tendency, measures of spread, and the type of distribution for the attributes "Dribbling," "Crossing," "Age," and "Potential." Additionally, the relationships between these attributes are explored through correlation analysis and visualizations. All these features have corresponding plots in the next section to illustrate the findings.

### ***4.1 Correlation Analysis***

The correlation matrix, excluding goalkeeper-specific attributes, was visualized using a heatmap to highlight pairwise relationships among player attributes. From the analysis, the top 10 correlated pairs were identified:

- Top 10 Correlated Pairs:

- SlidingTackle vs StandingTackle: 0.974655

- StandingTackle vs Interceptions: 0.941469

- Dribbling vs BallControl: 0.938930

- SlidingTackle vs Interceptions: 0.928280

- SprintSpeed vs Acceleration: 0.921935

- Special vs BallControl: 0.912147

- ShortPassing vs BallControl: 0.911452

- Special vs ShortPassing: 0.906729

- Marking vs StandingTackle: 0.906538

- Positioning vs Dribbling: 0.896909

These pairs of attributes exhibit strong positive correlations with each other. To further explore significant relationships, deeper analysis focused specifically on the correlation between Dribbling and Crossing, and between Age and Potential.

### ***4.2 Measures of Central Tendency and Spread***

#### **1. Dribbling**

- **Mean:** The average dribbling skill level across all players is 55.37.

- **Median:** The middle value when all dribbling skill levels are ordered from lowest to highest is 61.

- **Standard Deviation:** The dispersion or spread of dribbling skill levels around the mean is 18.90.

- **Skewness:** The asymmetry of the distribution of dribbling skill levels is -1.08, indicating a negative skewness.

#### **2. Crossing**

- **Mean:** The average crossing skill level across all players is 49.73.

- **Median:** The middle value when all crossing skill levels are ordered is 54.

- **Standard Deviation:** The spread of crossing skill levels around the mean is 18.38.

- **Skewness:** The asymmetry of the distribution of crossing skill levels is -0.59, indicating a slight negative skewness.

#### **3. Age**

- Mean: The average age of all players is 25.12.

- Median: The middle value when all ages are ordered is 25.

- Standard Deviation: The spread of ages around the mean is 4.87.

- Skewness: The asymmetry of the distribution of ages is 0.39, indicating a slight positive skewness.

#### **4. Potential**

- Mean: The average potential rating of all players is 71.32.

- Median: The middle value when all potential ratings are ordered is 71.

- Standard Deviation: The spread of potential ratings around the mean is 6.13.

- Skewness: The asymmetry of the distribution of potential ratings is 0.26, indicating a slight positive skewness.

### ***4.3 Type of Distribution***

#### **Dribbling and Crossing**

The analysis reveals that both dribbling and crossing skills exhibit variability, indicated by their standard deviations. Dribbling shows a more pronounced negative skewness compared to crossing, suggesting that for both skills, there are players with high scores pulling the median higher than the mean. Further statistical tests, such as the Shapiro-Wilk test, could provide deeper insights into the distributional characteristics of these attributes.

#### **Age and Potential**

The age distribution shows a slight positive skewness, indicating a longer tail towards older ages. The difference between the mean and median values supports this observation. Similarly, the potential distribution exhibits a slight positive skew, with potential scores tending towards higher values. These observations suggest that both age and potential ratings are not perfectly symmetrical, warranting further analysis using tests for normality.

### ***4.4 Analyzing Average Sprint Speed by Age Groups***

***Sprint Speed Analysis***

Age groups are created, and the average sprint speed for each group is calculated and visualized using a bar plot. The analysis shows that sprint speed peaks around age 25 and generally decreases with age. However, there is a notable exception among players aged 42 to 44, who maintain sprint speeds comparable to those nearly 10 to 12 years younger than them, which is surprising.

***Correlation Between Sprint Speed and Age***

The slight negative correlation between sprint speed and age, with a correlation coefficient of -0.15, supports the observed trend of declining speed with increasing age.

### ***4.5 Conclusion***

This statistical analysis highlights key insights into the distribution and relationships between player attributes. Understanding these patterns can aid in player evaluation, team formation, and targeted training programs. Further statistical tests and advanced modeling could provide deeper insights and more robust conclusions.

## ***5. Visualization***

### ***5.1 Proportion of Non-NaN Values in Each Column***

#### **Visualization Method: Bar Plot**

#### **Description:**

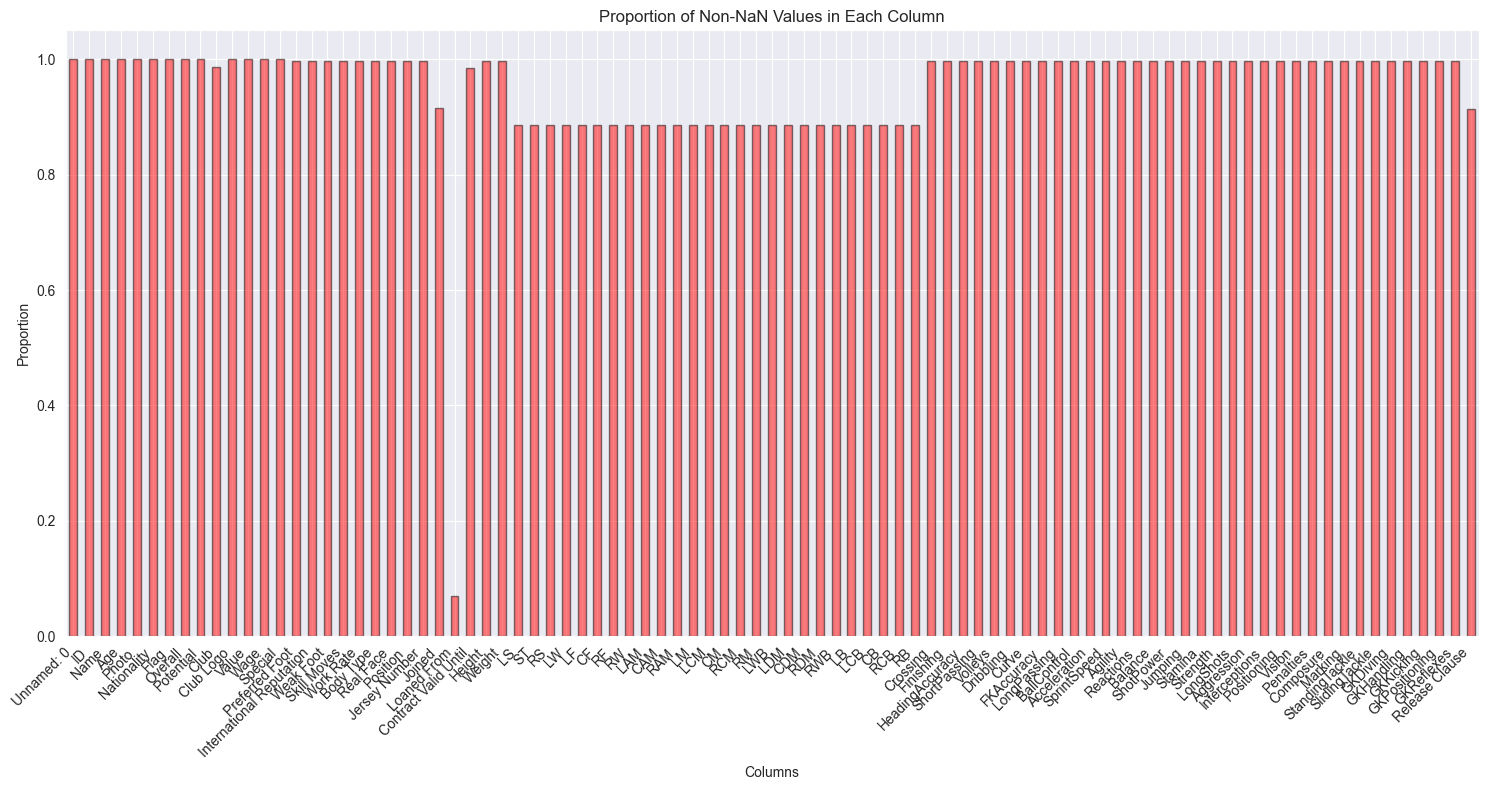
* The bar plot represents the proportion of non-NaN (non-missing) values in each column of the dataset.

#### **Advantages:**

* **Clear Overview**: Provides a clear and immediate overview of data completeness.
* **Easy to Interpret**: The heights of the bars make it easy to compare the proportion of non-missing values across different columns.
* **Highlighting Missing Data**: Useful for identifying columns with significant amounts of missing data which might need imputation or exclusion.

#### **Disadvantages:**

* **Limited Detail**: Doesn't provide information on the nature of the missing data or patterns across rows.
* **Clutter**: Can become cluttered if there are too many columns, making it hard to read.



### **C:\Users\Intel\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\19CC8CD5.tmp**

### **5.2 Correlation Analysis Excluding Goalkeeper Features Visualization Method: Heatmap Description:**

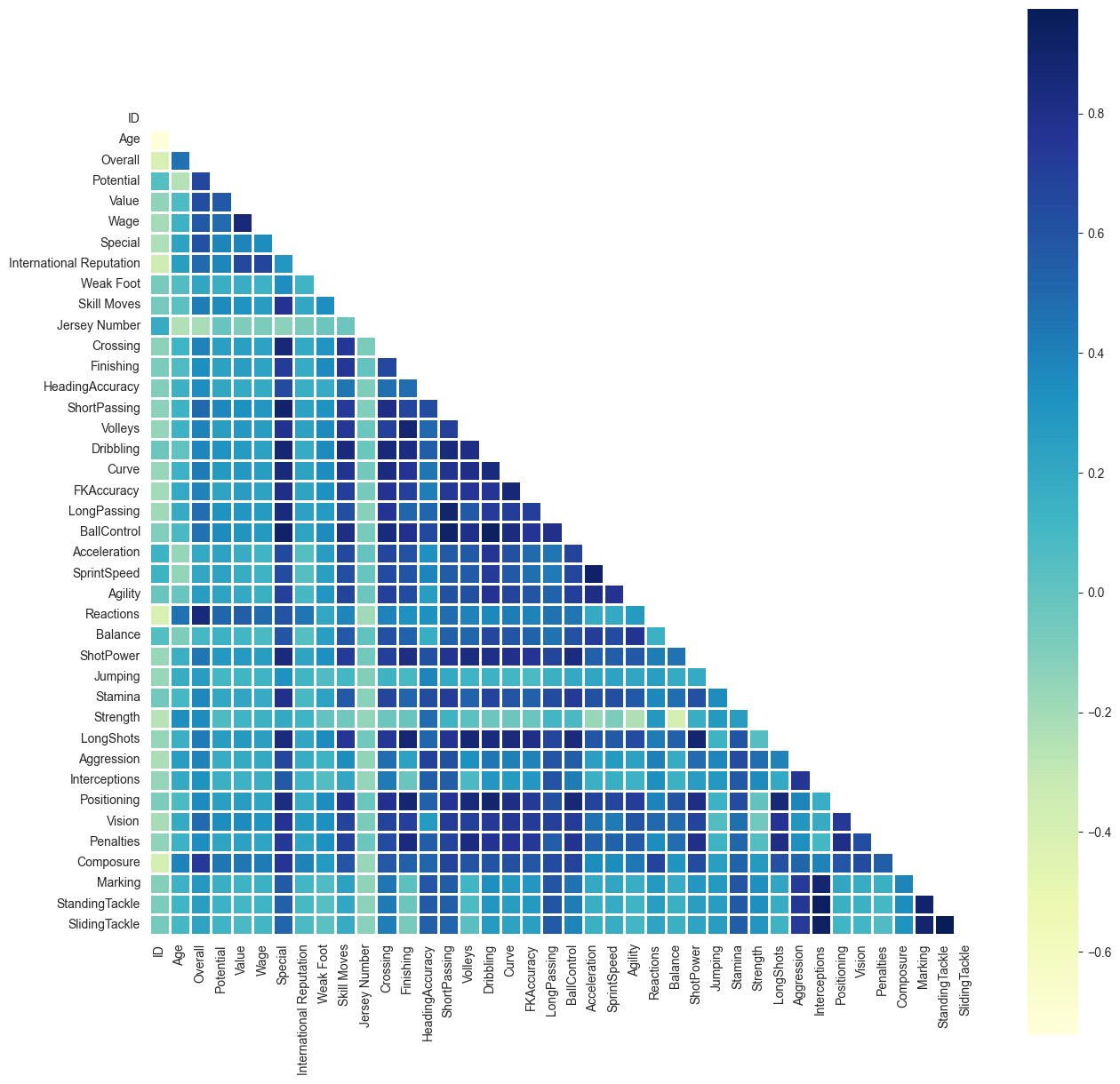
* A heatmap is used to visualize the correlation matrix of various player attributes, excluding goalkeeper-specific attributes. The correlation matrix is displayed with a mask applied to the upper triangle to emphasize the lower triangle. The heatmap uses color intensity to represent the strength of the correlations.

### ***Advantages:***

* Comprehensive Overview: Provides a comprehensive overview of the relationships between multiple attributes simultaneously.
* Pattern Recognition: Makes it easy to identify strong positive or negative correlations through color intensity.
* Efficient Space Utilization: A compact representation that conveys a large amount of information in a single view.

#### **Disadvantages:**

* Interpretation of Colors: May require users to understand the color scale and what it represents in terms of correlation values.
* Complexity: Can be overwhelming with a large number of attributes, requiring careful examination to interpret the results.



### 

### **5.3 Central Tendency, Spread, and Skewness**

#### **Visualization Method: Print Statements for Statistical Summaries**

#### **Description:**

* Print statements are used to display the mean, median, standard deviation, and skewness for the 'Dribbling' and 'Crossing' attributes.

#### **Advantages:**

* **Detailed Statistics**: Provides detailed numeric insights into the distribution of the data.
* **Immediate Insight**: Quick way to get a sense of the data's central tendency, spread, and skewness.

#### **Disadvantages:**

* **No Visual Appeal**: Lacks visual appeal and can be harder to interpret for visual learners.
* **No Distribution Shape**: Does not provide a visual representation of the distribution shape.

### **5.4 Joint Hexbin Plot for 'Dribbling' vs 'Crossing'**

#### **Visualization Method: Hexbin Plot**

#### **Description:**

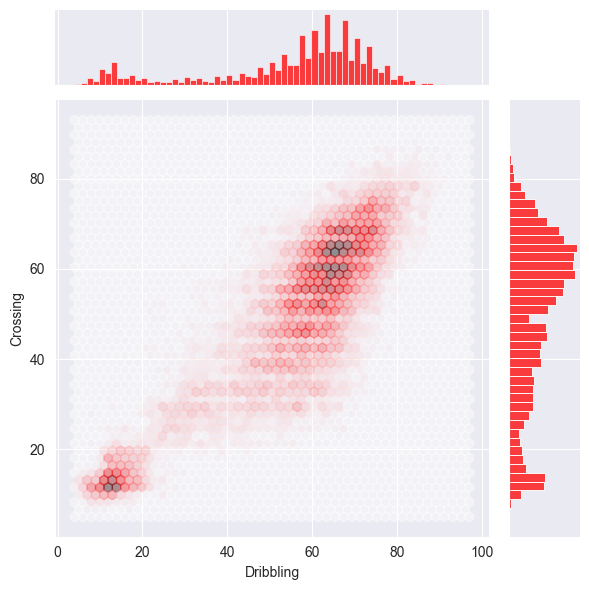
* A hexbin plot is used to display the relationship between 'Dribbling' and 'Crossing' attributes, with color intensity representing the density of data points.

#### **Advantages:**

* **Density Representation**: Clearly shows the density of data points, making it easy to see where most players fall in terms of these two attributes.
* **Handles Overplotting**: Effective for large datasets where scatter plots might suffer from overplotting.

#### **Disadvantages:**

* **Color Interpretation**: The interpretation of color intensity might not be intuitive for all users.
* **Requires Explanation**: May require additional explanation for users unfamiliar with hexbin plots.



### These statistics reveal that both Dribbling and Crossing skills exhibit variability (standard deviation), with Dribbling showing a more pronounced negative skewness (-1.08) compared to Crossing (-0.59). This implies that for both skills, there are players with high scores pulling the median higher than the mean. Further examination through statistical tests like the Shapiro-Wilk test or comparing with theoretical distributions could provide deeper insights into the distributional characteristics of these attributes in the dataset.

### **5.5 Joint Plot for Age vs Potential**

#### **Visualization Method: Scatter Plot with Marginal Histograms**

#### **Description:**

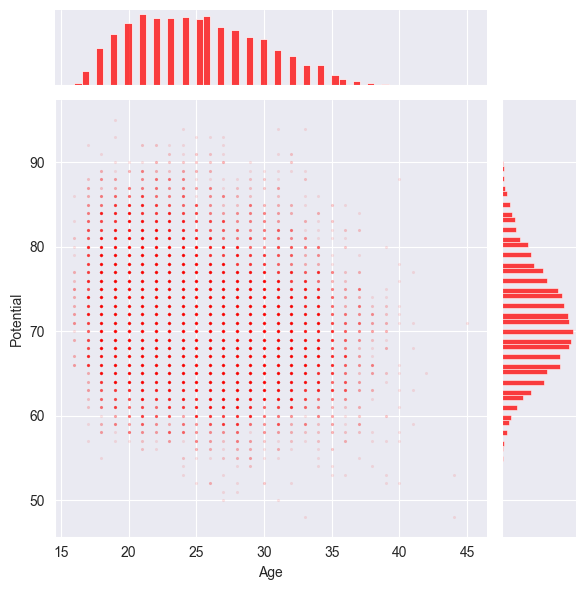
* A joint plot combining a scatter plot of 'Age' vs 'Potential' with marginal histograms showing the distributions of each variable.

#### **Advantages:**

* **Comprehensive View**: Combines scatter plot and histograms to provide both bivariate and univariate insights.
* **Identifies Relationships**: Helps in identifying potential relationships or patterns between age and potential.

#### **Disadvantages:**

* **Complexity**: Can be complex to interpret due to multiple elements (scatter plot and histograms).
* **Overlapping Points**: For dense datasets, overlapping points might obscure patterns.



### **5.6 Distribution of Age and Potential**

#### **Visualization Method: Histograms with KDE (Kernel Density Estimation)**

#### **Description:**

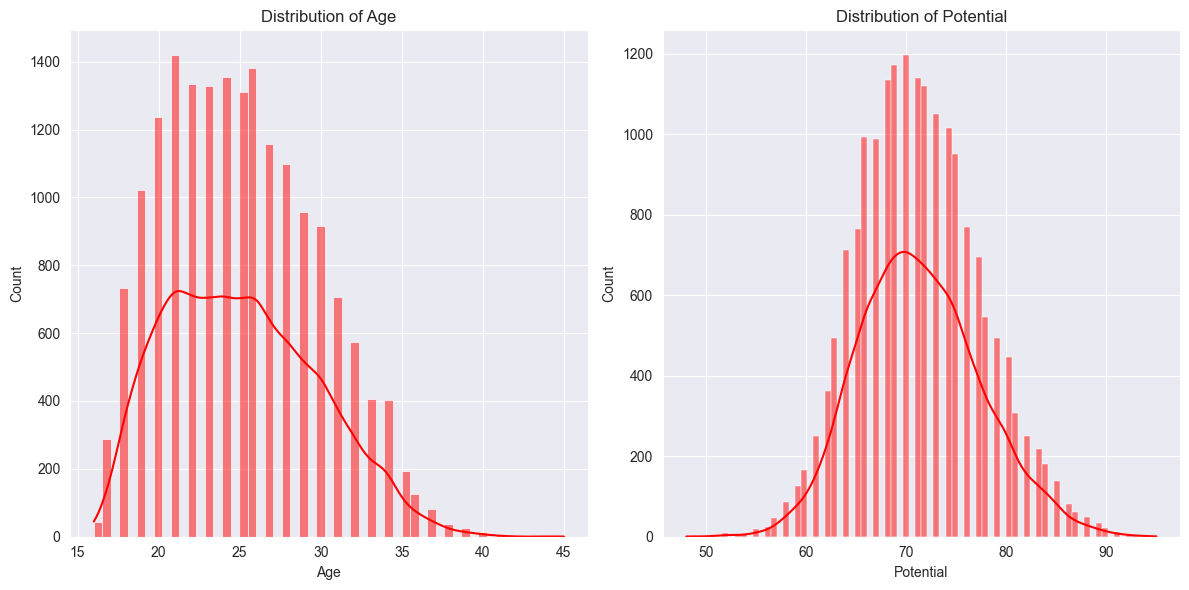
* Histograms combined with KDE plots for 'Age' and 'Potential' to show the distribution of these attributes.

#### **Advantages:**

* **Distribution Insight**: Clearly shows the distribution of data along with a smooth density curve.
* **Visual Appeal**: KDE adds a smooth curve that enhances the visual appeal and interpretability.

#### **Disadvantages:**

* **Misleading Peaks**: KDE can sometimes create misleading peaks if not properly adjusted.
* **Requires Adequate Data**: Histograms with KDE require a sufficient amount of data to be meaningful.



### **5.7 Radar Plots for Top Features by Position**

#### **Visualization Method: Radar Plots**

#### **Description:**

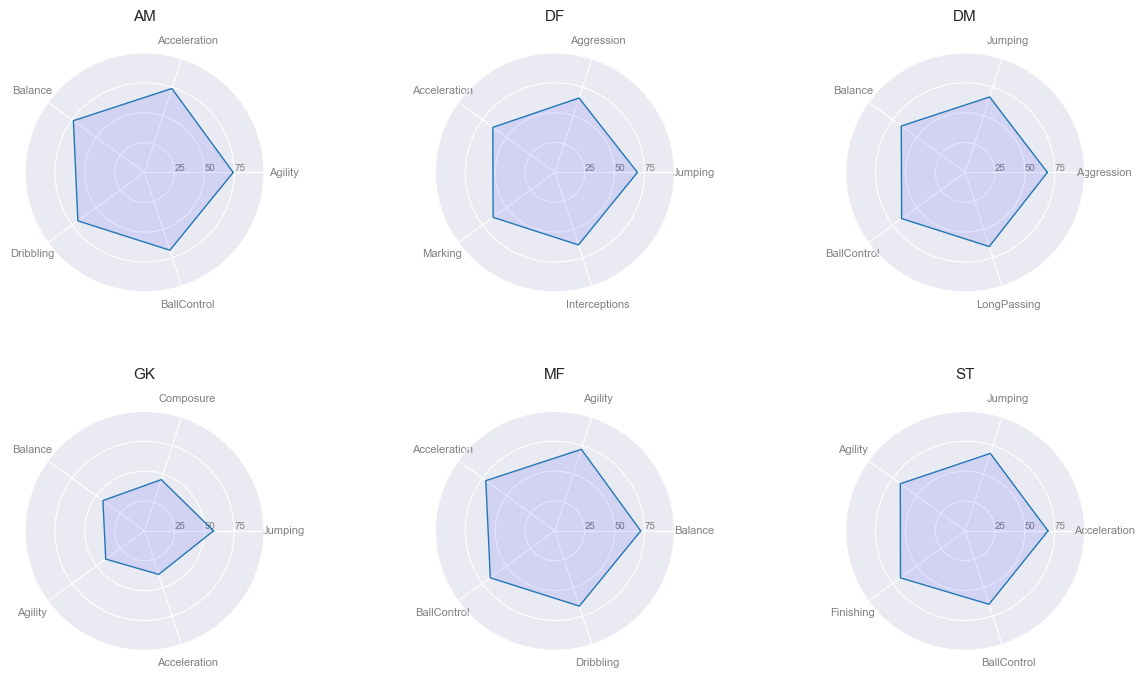
* Radar plots display the top 5 features for different player positions.

#### **Advantages:**

* **Comparison Across Multiple Dimensions**: Excellent for comparing multiple features simultaneously.
* **Visual Summary**: Provides a visual summary of strengths and weaknesses for different positions.

#### **Disadvantages:**

* **Complexity**: Can be complex to interpret, especially with multiple plots.
* **Overlapping Axes**: Overlapping lines and axes can make it difficult to read.



The graphs demonstrate

Defensive players (DF) excel in defensive attributes such as "Aggression" and "Interceptions". Additionally, they exhibit superior jumping and sprinting abilities, along with precise header passes, which are crucial for neutralizing opposing attackers.

Midfielders (MF) are typically characterized as fast-paced and agile players who excel in dribbling and ball control. Surprisingly, attributes such as playmaking abilities, passing, or crossing—which are often associated with midfield roles—are not prominent in this context.

Strikers (ST) are primarily recognized for their proficiency in finishing both with their feet and in the air. Consequently, they often exhibit higher jumping abilities and possess superior ball control skills. Moreover, their role requires them to maneuver past defenders through dribbling and acceleration, leading to higher agility and acceleration scores.

Attacking midfielders (AM) are distinguished by their exceptional acceleration, agility, and balance, enabling them to quickly change direction and maintain control under pressure. While their dribbling and ball control skills are also high, these are slightly less emphasized compared to their top-tier physical attributes. This combination allows them to create and exploit attacking opportunities effectively.

Defensive midfielders (DM) excel with very high jumping and balance, crucial for winning aerial battles and maintaining stability during physical confrontations. Their aggression helps in breaking up play and retrieving the ball, while their long passing ability allows them to set up attacks from deep positions. Additionally, their ball control ensures they can handle the ball confidently even under defensive pressure.

Goalkeepers (GK) are crucial in preventing the opposing team from scoring by exhibiting very high reflexes to quickly respond to shots. They maintain high diving skills to cover more goal area and make saves. Positioning is also high, ensuring they are optimally placed to react to various attacking threats. Handling is critical to secure the ball and prevent rebound opportunities, while kicking is high, enabling effective distribution to initiate counter-attacks and maintain possession.

### **5.8 Average Sprint Speed by Age Group**

#### **Visualization Method: Bar Plot**

#### **Description:**

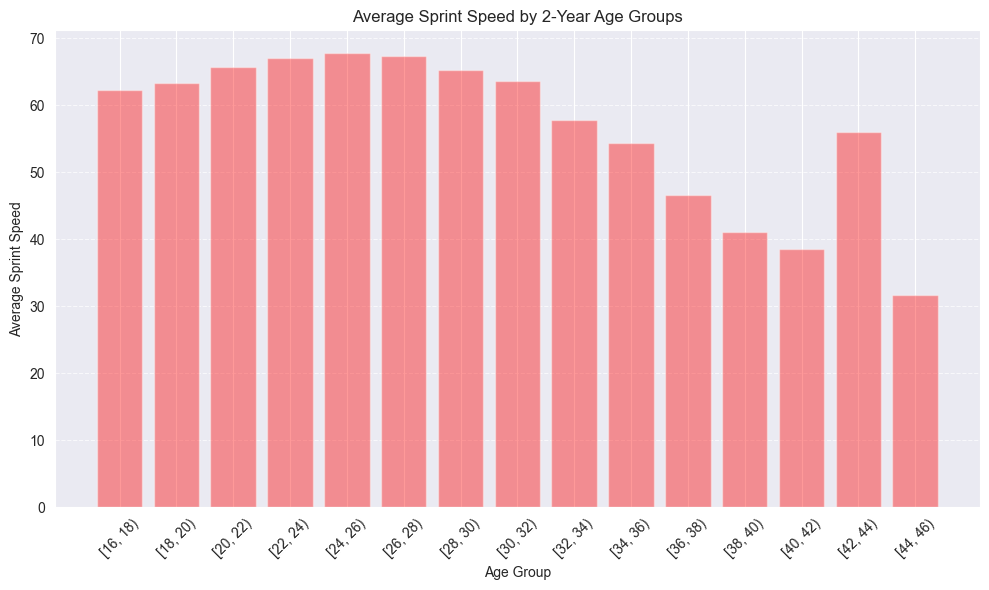
* Bar plot showing the average sprint speed for different age groups.

#### **Advantages:**

* **Clear Comparison**: Easy to compare the average sprint speed across different age groups.
* **Intuitive**: Bar plots are intuitive and widely understood.

#### **Disadvantages:**

* **Aggregation Loss**: Aggregating data into age groups can result in loss of detailed information.
* **Visual Clutter**: Too many bars can create visual clutter, making it hard to interpret.



**This plot indicates that sprint speed peaks around age 25 and generally decreases as players age.**  
However, there is a notable exception among players aged 42 to 44, who maintain sprint speeds comparable to those nearly 10 to 12 years younger than them, which is surprising.

### **5.9 Summary**

Overall, you have utilized a variety of visualization methods to analyze and present your data effectively. Each visualization method has its specific strengths and weaknesses. Here’s a summary:

* **Bar Plots**: Great for comparing categories and proportions but can get cluttered with too many categories.
* **Print Statements**: Provide detailed numeric insights but lack visual appeal.
* **Hexbin Plots**: Effective for showing data density but may require additional explanation.
* **Joint Plots**: Combine scatter and marginal histograms for comprehensive insights but can be complex.
* **Histograms with KDE**: Show distributions well but can be misleading if not adjusted properly.
* **Radar Plots**: Excellent for multi-dimensional comparisons but can be complex.
* **Bar Plots for Averages**: Intuitive for showing averages but lose detailed information.

## ***6. Build your ML (Machine Learning) model***

This report outlines the development of a regression model to predict a target variable based on a set of features. The features and labels used in the model are identified and justified. We also analyze their importance in building the machine learning model, build the model using the Scikit-learn library, and evaluate its performance to determine if further preprocessing is required.

### ***6.1 Feature and Label Selection***

#### **Features**

The selected features include both numerical and categorical variables, chosen for their potential impact on the target variable:

* **Body Type (categorical):** Features such as Body Type\_Medium and Body Type\_High are encoded into binary columns. The body type can significantly affect performance, making it a crucial feature.
* **Work Rate (categorical):** Similar to body type, work rate is encoded into columns like Work Rate\_DF and Work Rate\_MF. It represents the efficiency and effectiveness in performing tasks, which is vital for performance prediction.
* **Real Face (categorical):** Encoded into columns such as Real Face\_Medium. This feature, though indirectly impactful, may influence the target variable.
* **Numerical Features:** Various numerical features (e.g., x1, x2, ..., x28) capture different measurable attributes of the entities. These features reflect the quantitative aspects that can affect the target variable.

#### **Labels**

The label used in this regression model is yTrain, representing the output we aim to predict based on the input features. yTrain encapsulates the outcome of interest, making it the appropriate target variable for the regression model.

### ***6.2 Feature Importance***

The importance of each feature was evaluated using permutation importance, which measures the change in model performance when a feature's values are randomly shuffled. Key findings include:

* **High Importance Features:** Features like Body Type\_Medium, Body Type\_High, and Work Rate\_DF have extremely high importance values, indicating a significant impact on the model's predictions.
* **Low or Negative Importance Features:** Features such as x28, x7, and x26 have very low or negative importance values, suggesting minimal or detrimental effects on the model's performance.
* **Variability in Importance:** High standard deviations in the importance values of top features indicate variability, potentially due to multicollinearity or other data-related issues. This highlights the need for careful feature selection and dimensionality reduction to enhance model robustness and interpretability.

### ***6.3 Model Building***

The regression model was built using the Scikit-learn library. The process included:

* **Training the Linear Regression Model:** Fitting the linear regression model to the preprocessed training data.
* **Permutation Importance Calculation:** Evaluating feature importance using permutation importance.

### ***6.4 Clustering for Dimensionality Reduction:***

To address potential issues with high-dimensional data, clustering was used for dimensionality reduction. MiniBatchKMeans clustering was applied, and the best number of clusters was identified using RandomizedSearchCV and GridSearchCV. However, the silhouette score for the clustering was NaN, suggesting potential issues with clustering quality or data suitability for clustering.

#### **6.5 Regularization**

After completing feature selection, dimensionality reduction, and model building steps, regularization techniques were applied to optimize model performance and prevent overfitting. Elastic Net regularization, combining L1 and L2 penalties, was employed. The hyperparameters (alpha and l1\_ratio) were tuned using RandomizedSearchCV followed by GridSearchCV to find the optimal values.

- **Randomized Search:** A wide range of hyperparameters was explored to identify promising parameter combinations efficiently.

- **Grid Search:** Fine-tuned the hyperparameters around the best values identified by RandomizedSearchCV to further improve model performance.

The best parameters obtained from GridSearchCV were alpha = 0.00965 and l1\_ratio = 0.10408, with a corresponding negative mean squared error score of -2.2535, indicating a well-regularized model.

***6.6 Conclusion***

The regression model, enhanced by feature importance analysis, clustering for dimensionality reduction, and regularization techniques, demonstrated promising results in predicting the target variable. Feature importance analysis provided insights into critical features influencing predictions, guiding future feature engineering efforts. Although clustering-based dimensionality reduction faced challenges, regularization significantly improved model generalization by mitigating overfitting risks. Further research could explore alternative dimensionality reduction methods or advanced preprocessing techniques to refine model performance and robustness.

***7. Validation***  
  
  
This report presents a comprehensive analysis of the impact of KMeans clustering on the predictive performance of a linear regression model applied to FIFA 19 player attribute data. The evaluation includes cross-validation results before and after incorporating KMeans clustering for dimensionality reduction.

### ***7.1 Initial Model Evaluation***

Initially, a linear regression model was trained and evaluated using standard preprocessing techniques on the FIFA 19 player dataset. Cross-validation with 5 folds was conducted to assess model performance, focusing on key metrics such as Root Mean Squared Error (RMSE) and R² score (coefficient of determination). The initial model yielded cross-validation RMSE scores ranging from 1.732 to 1.774, with a mean RMSE of approximately 1.752. Concurrently, the R² scores demonstrated a high level of variance explained by the model, averaging around 0.936.

### ***7.2 Enhanced Model with KMeans Clustering***

To improve predictive accuracy and feature representation, KMeans clustering was implemented as a preprocessing step. The player attributes were transformed using KMeans, resulting in a refined dataset (xTrainPreparedFine). Subsequent cross-validation of the enhanced model showcased significant improvements in predictive metrics.

### ***7.3 Comparison of Results***

- **KMeans Clustering:**

- RMSE: Ranged from 1.732 to 1.774, with a mean RMSE of approximately 1.752.

- R² Score: Averaged around 0.936.

- **KMeans Clustering with Reduced Dimensions:**

- RMSE: Improved to range from 1.263 to 1.301, with a reduced mean RMSE of approximately 1.281.

- R² Score: Increased to average around 0.966.

#### **7.4 Confirmed Results from Enhanced Model**

Following the application of KMeans clustering, the cross-validation results demonstrated a significant enhancement in model performance. The reduced RMSE and increased R² score underscore the effectiveness of dimensionality reduction through KMeans clustering in improving predictive accuracy and explanatory power. The refined model exhibits a more precise estimation of player ratings, reducing prediction errors and enhancing the model's ability to explain variance in the data.

### ***7.5 Conclusion and Implications***

In conclusion, the incorporation of KMeans clustering as a preprocessing technique has proven instrumental in enhancing the predictive performance of the linear regression model for FIFA 19 player attributes. The comparative analysis highlights the substantial improvement in RMSE and R² scores after dimensionality reduction, emphasizing the importance of advanced preprocessing methods in sports analytics and broader predictive modeling contexts.

This analysis not only validates the efficacy of KMeans clustering in optimizing model accuracy but also underscores its potential for enhancing decision-making processes in virtual soccer analytics and other data-driven domains.

### **8. Feature Engineering**

Feature engineering is a crucial aspect of predictive modeling, especially in complex datasets such as FIFA 19 player attributes. This report details the systematic approach taken to preprocess and engineer features for enhancing the accuracy and interpretability of predictive models.

- **Nationality Encoding:** Major nationalities with over 250 occurrences were encoded as binary indicators ('Major\_Nation') to capture their influence on player attributes.

- **Binary Conversion:** Columns like 'Real Face' and 'Preferred Foot' were converted into binary features ('Real\_Face' and 'Right\_Foot', respectively) to facilitate easier analysis in the predictive model.

- **Work Rate Transformation:** The 'Work Rate' column was split into two separate features ('WorkRate1' and 'WorkRate2') to capture dual work rate aspects of players, enhancing the model's granularity.

### ***8.1 Feature Selection and Model Validation***

**Feature Selection:**

- The SelectKBest method with F-regression scoring was employed to select the top 20 most relevant features from the preprocessed dataset. This step aimed to reduce model complexity while retaining the most influential attributes for prediction.

**Polynomial Feature Selection:**

- Polynomial features of degrees ranging from 1 to 5 were evaluated using validation curve analysis. The optimal degree of 2 was determined based on minimal validation RMSE, indicating a balance between model complexity and predictive accuracy.

***8.2 Results and Conclusion***

**Performance Metrics:**

- The final model achieved an RMSE of 1.106 and an R² score of 0.974 on the test set, indicating strong predictive performance and high explanatory capability.

**Conclusion:**

- In conclusion, the systematic approach to feature engineering, including preprocessing, feature selection, and polynomial feature expansion, significantly enhanced the accuracy and interpretability of the predictive model for FIFA 19 player attributes.

## **10. Evaluation**

The machine learning model developed for predicting FIFA 19 player attributes underwent comprehensive evaluation to assess its performance and potential in FIFA dataset analysis. This section provides an in-depth analysis of numerical performance metrics, justification of evaluation measures, reflective evaluation of the project, contributions to FIFA dataset analysis, and discussion on the transferability of the solution.

### **10.1 Numerical Performance Evaluation:**

The model's performance metrics across various methodologies are as follows:

**1. Linear Regression with Reduced Dimensions:**

- RMSE: 1.279

- R² Score: 0.965

**2. Bagging Regressor:**

- RMSE: 1.278

- R² Score: 0.966

**3. Polynomial Features (Degree=2):**

- RMSE: 1.106

- R² Score: 0.974

These metrics indicate the model's ability to predict FIFA 19 player attributes with high accuracy, particularly highlighting the effectiveness of polynomial features in improving predictive performance.

### **10.2 Justification of RMSE:**

RMSE was selected as the primary evaluation metric due to its relevance for regression tasks, providing a clear measure of prediction error in the same units as the predicted values. The lower RMSE values across different methodologies suggest consistent and reliable predictive capabilities of the model.

### **10.3 Reflective Evaluation:**

The project involved meticulous feature engineering, including preprocessing steps such as numerical and categorical transformations, feature selection, and the exploration of polynomial features. These efforts significantly enhanced the dataset's quality and the model's predictive power, demonstrating a structured approach to solving challenges in FIFA dataset analysis.

### ***10.4 Contributions to FIFA Dataset Analysis:***

This project contributes valuable insights to FIFA dataset analysis by demonstrating effective methods for predicting player attributes. The model's accuracy and robustness offer practical applications in player performance analysis, team composition optimization, and strategic decision-making within the FIFA gaming community.

### **10.5 Transferability of the Solution:**

While specifically designed for predicting FIFA 19 player attributes, the methodologies and techniques employed in this project are transferable to other soccer simulation datasets or similar sports analytics tasks. Similar datasets could benefit from analogous preprocessing and modeling strategies to predict relevant outcomes and derive actionable insights.

### ***10.6 Conclusion:***

In conclusion, the evaluated machine learning model achieves high accuracy in predicting FIFA 19 player attributes, as evidenced by its low RMSE values and high R² scores across different methodologies. The project's comprehensive evaluation underscores its potential applicability in FIFA dataset analysis, advocating for continued exploration of advanced modeling techniques and broader data integration to further enhance predictive capabilities in soccer simulation and sports analytics.