Machine Learning Assignment Report

Submitted By:

Sai Vara Prasad Kotha (2021B5A71541P)

Ankur Renduchintala (2021B5A71159P)

Akshay Malu (2021B3A71721P)

Rishabh Malik (2021B3A70990P)

Instructors:

Prof. Jagat Sesh Challa

Prof. Tejasvi Alladi

Course:

Machine Learning (BITS F464)



Department of Computer Science and Information Systems

Birla Institute of Technology and Science, Pilani

20 November, 2024

Table of Contents

# Introduction:

## Objective:

The primary objective of this project is to apply machine learning techniques to a labelled dataset containing a mixture of numerical and categorical attributes. The goal is to perform the following tasks:

* **Regression Analysis**: Predict numerical target variables using models like Linear Regression, Random Forests, XGBoost, NGBoost etc.
* **Classification**: Classify categorical target variables using Logistic Regression, Naive Bayes, and Decision Trees.
* **Clustering**: Identify patterns and groups in the dataset using clustering techniques like K-means and Expectation-Maximization (EM).

## Scope:

This project focuses on analysing the performance of machine learning models on a multi-dimensional dataset through rigorous preprocessing, implementation, and evaluation. The results provide insights into:

1. The predictive capability of regression and classification models.
2. The effectiveness of clustering algorithms in uncovering underlying data structures.
3. The importance of preprocessing steps like feature scaling and dimensionality reduction.

## Dataset Overview:

The dataset used in this assignment is the **“FIFA 22 Official Dataset”**, sourced from Kaggle. It provides a comprehensive collection of data related to professional football players, offering a rich blend of numerical and categorical attributes.

Key features of the dataset include:

* **Attributes**: A total of 65 features, capturing diverse aspects such as player statistics, physical attributes, performance metrics, and personal details.
* **Records**: The dataset contains 16,710 records, providing a substantial volume of data for meaningful analysis.

# Methodology:

## Data Preprocessing:

‘FIFA22\_official\_data.csv’ contains 16710 records and 65 columns. The data is not pristine, so it must be cleaned. Categorical data is not very useful for predicting the ‘Overall’ attribute of a player and missing values lead the models astray. So, the logical steps forward would be to cleanse the dataset by removing categorical and redundant variables.

### Categorical and Redundant Features Removal:

There are **20 non-numeric** features in the dataset: 'Name', 'Photo', 'Nationality', 'Flag', 'Club', 'Club Logo', 'Value', 'Wage', 'Preferred Foot', 'Work Rate', 'Body Type', 'Real Face', 'Position', 'Joined', 'Loaned From', 'Contract Valid Until', 'Height', 'Weight', 'Best Position', 'Release Clause' of which Height, Weight, Wage, Value and Release Clause are numeric along with units.

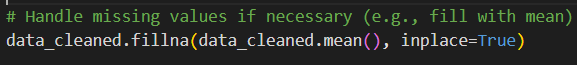
There are **45 numeric attributes** in the dataset: 'ID', 'Age', 'Overall', 'Potential', 'Special', 'International Reputation', 'Weak Foot', 'Skill Moves', 'Jersey Number', 'Crossing', 'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribbling', 'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration', 'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower', 'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression', 'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure', 'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling', 'GKKicking', 'GKPositioning', 'GKReflexes', 'Best Overall Rating', 'DefensiveAwareness' of which, Skill Moves, International Reputation and Weak Foot are categorical in nature.

We remove categorical and redundant features: 'Name', 'Photo', 'Nationality', 'Flag', 'Club', 'Club Logo', 'Value', 'Wage', 'Preferred Foot', 'Work Rate', 'Body Type', 'Real Face', 'Position', 'Joined', 'Loaned From', 'Contract Valid Until', 'Best Position', 'Release Clause', 'International Reputation', 'Weak Foot', 'Skill Moves' to obtain 44 attributes.

There are unnecessary features which add bias to the models. So, we have to handle those cases also by removing the columns which cause this. So, we remove: 'ID', 'Jersey Number', 'Best Overall Rating' to obtain 41 features.

There is a column called ‘Marking’ in our dataset. It has roughly 95% missing data. We cannot handle that by replacing the missing values with mean or median or anything. So, as this column is not adding any value, we drop that column. Now we are left with 40 features.

Our target variable is 'Overall', so we remove it from our data frame and store as target. The remaining 39 attributes make up the data\_cleaned data frame. Missing values are still present in few columns so we fill in the gaps with the means of the respective attributes to obtain our numerical\_data dataset.

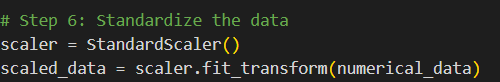


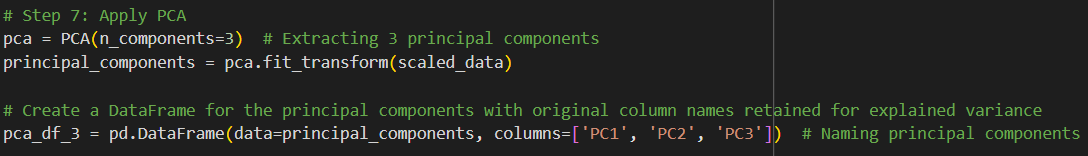
### Principal Component Analysis (PCA):

1. **Standardization**:  
   Before applying PCA, we first standardized the numerical data using **StandardScaler**. This step ensures that every feature in the dataset has a mean of 0 and a standard deviation of 1, which is crucial for PCA to perform optimally. Standardizing the data removes any inherent biases caused by features with different scales, allowing PCA to treat all features equally.
2. **Principal Component Analysis (PCA)**:  
   After standardizing the data, we applied the PCA algorithm to extract the principal components. By retaining the top 3 principal components, we reduced the dataset's dimensionality while preserving the most significant variance in the data. These three principal components, labelled as **'PC1'**, **'PC2'**, and **'PC3'**, now form the core of our transformed dataset.
3. **Renormalization**:  
   Post-PCA, we renormalized the resulting dataset to ensure that the mean of each principal component is 0 and the standard deviation is 1. This ensures that the final dataset is ready for machine learning applications, and each principal component contributes equally to the models.

As a result of the preprocessing steps, we now have three distinct data frames:

* **data\_cleaned**: The cleaned dataset, used for regression, classification, and clustering tasks.
* **scaled\_data (numpy array)**: The scaled version of the data, used specifically for clustering analysis.
* **pca\_df\_3**: The transformed dataset containing the first three principal components, used for classification tasks.





The explained variance ratio for each principal component (PC) indicates how much variance in the original dataset is captured by each component after performing Principal Component Analysis (PCA). It helps you understand how important each principal component is in explaining the overall variability in the data.

Explained Variance Ratios of our three components:

Principal Component 1: 0.4952

Principal Component 2: 0.1412

Principal Component 3: 0.0948

In our case, the first three components together explain about **64.62%** (0.4952 + 0.1412 + 0.0948) of the total variance. This suggests that the first three principal components capture a significant portion of the dataset's variability, making them the most relevant features for further analysis.

## Regression Analysis:

**Regression Models and Evaluation**

In this project, we utilized six different regression models to predict the **Overall** attribute of a football player from the dataset. The models selected for this task are as follows:

* **Linear Regression**
* **Random Forest Regressor**
* **Extra Trees Regressor**
* **AdaBoost Regressor**
* **XGBoost Regressor**
* **NGBoost Regressor**

To ensure a clean and safe approach, we first created a copy of the cleaned dataset, referred to as **X**, and stored the target variable (Overall) in **y**. All subsequent actions related to regression were performed on the **X** dataframe.

**Data Preprocessing and Splitting**

Before training the models, we addressed missing values (NaN) in the dataset. Since **X** contained some NaN values, we dropped the rows with missing values, reducing the size of the data to **15,818** records. Correspondingly, the target variable **y** was also adjusted by removing the entries that correspond to the dropped rows in **X**.

Next, the dataset was split into training and testing subsets:

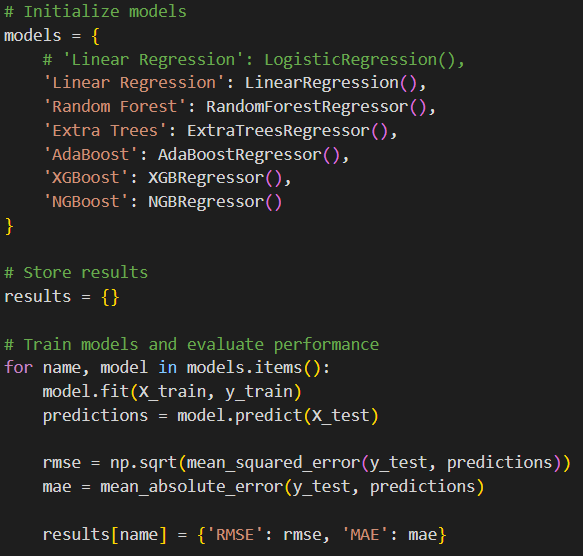
* **X.train** and **y.train**: Training data (80% of the original data)
* **X.test** and **y.test**: Testing data (20% of the original data)

**Model Training and Evaluation**

The models were then trained using **X.train** and **y.train**, with the predictions tested against the unseen data in **X.test**. The performance of each model was evaluated using two key metrics:

* **Root Mean Squared Error (RMSE)**: To assess the model’s overall prediction error.
* **Mean Absolute Error (MAE)**: To evaluate the average magnitude of prediction errors without considering their direction.

By iteratively training and testing each model, we compared their performances based on the calculated RMSE and MAE values, helping us select the most effective regression model for predicting the **Overall** attribute.



## Classification Analysis:

We employed seven machine learning models to classify the Overall attribute of football players:

1. **Logistic Regression**
2. **Naive Bayes**
3. **K-Nearest Neighbours (KNN)**
4. **Linear Support Vector Machine (Linear SVM)**
5. **Kernel Support Vector Machine (RBF Kernel)**
6. **Decision Trees**
7. **Basic Neural Network**

**1. Classification Setup**

To adapt the Overall attribute for classification tasks, we performed the following steps:

* **Binary Classification:** Players were classified into two categories:
  + **Class 1:** Overall rating ≥ 80
  + **Class 0:** Overall rating < 80
* **Target Variable Transformation:** The target variable y was updated to binary values:
  + y = 1 for Overall ≥ 80
  + y = 0 for Overall < 80

**2. Data Splitting**

The dataset was split into training and testing subsets:

* **Training Set (80%)**: X.train and y.train
* **Testing Set (20%)**: X.test and y.test

**3. Model Training and Evaluation**

Each model was:

* **Trained:** Using X.train with y.train as labels.
* **Tested:** Predictions were made on X.test and compared with y.test.
* **Metrics Calculated:** Key performance metrics like **Precision**, **Accuracy**, and **F1 Score** were computed for each model.

**Classification with PCA Data**

We extended the classification task using the **PCA-transformed dataset** (pca\_df\_3) for dimensionality reduction:

* **Target Variable:** Similar to the original data, we transformed the Overall rating into binary classes (y\_classification).
  + y\_classification = 1 for Overall ≥ 80
  + y\_classification = 0 for Overall < 80

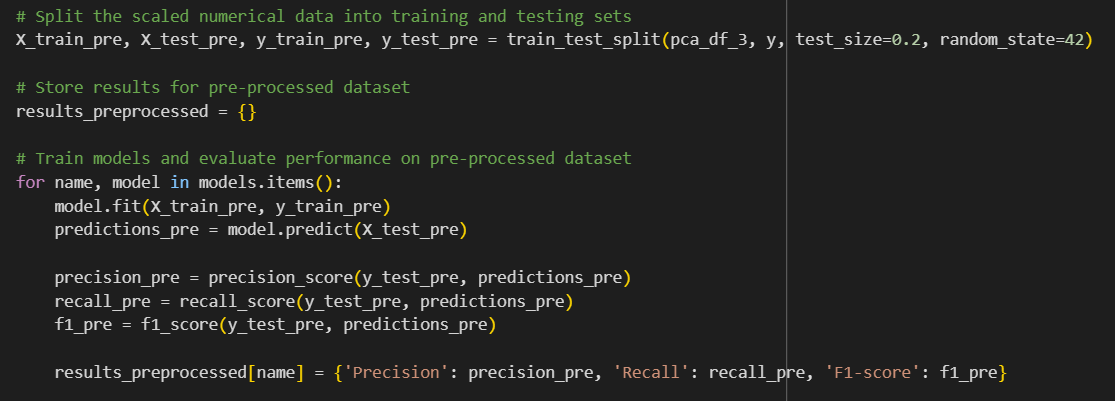
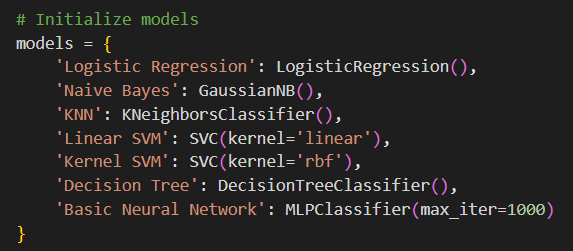
**1. PCA Data Splitting**

* **Training Set (80%)**: X\_train\_pre and y\_train\_pre
* **Testing Set (20%)**: X\_test\_pre and y\_test\_pre

**2. Model Training and Evaluation**

The same models were applied iteratively:

* **Training:** Models were trained on X\_train\_pre with y\_train\_pre as labels.
* **Testing:** Predictions were made on X\_test\_pre and evaluated against y\_test\_pre.
* **Metrics Calculated:** Precision, Accuracy, and F1 Score were computed for performance comparison.



## Clustering Analysis:

**Objective**

The objective was to cluster players based on their physical, technical, and goalkeeper attributes into distinct groups using multiple clustering algorithms. These clusters were analyzed to identify patterns, classify players into positions, and distinguish between player types.

**Attributes and Data Preparation**

1. **Physical and Technical Attributes**:
   * Physical attributes: *Acceleration, SprintSpeed, Strength, Agility, Balance, Jumping, Stamina.*
   * Technical attributes: *Dribbling, BallControl, ShortPassing, Finishing, Crossing, LongShots.*
   * The combined dataset containing these attributes (df\_non\_transformed) was used to cluster players into **three groups** iteratively.
2. **Goalkeeper Attributes**:
   * Attributes: *GKDiving, GKHandling, GKKicking, GKPositioning, GKReflexes.*
   * This dataset (goalkeeper\_columns) was used to cluster goalkeepers into **two groups** iteratively.

**Methodology**

1. **Clustering Algorithms**:
   * **K-Means**: A centroid-based clustering method focusing on minimizing intra-cluster variance.
   * **K-Medoids**: A robust alternative to K-Means, using medoids instead of centroids.
   * **Expectation-Maximization (Gaussian Mixture Model)**: A probabilistic clustering approach modeling data as a mixture of Gaussian distributions.
2. **Dimensionality Reduction and Visualization**:
   * Post-clustering, dimensionality was reduced to **two principal components** using **PCA** for enhanced visualization.
   * Pairwise attribute plots were created to analyze how players are distributed across physical and technical characteristics, helping to classify players into field positions.
3. **Standardization (Transformed Data)**:
   * The datasets were standardized using **Z-scale transformation** to create df\_transformed for comparison with non-transformed clustering results.
   * Transformed data clusters were similarly reduced in dimensionality for visualization and compared with the non-transformed clusters.

**Evaluation Metrics**

Clustering models were rigorously evaluated using the following metrics:

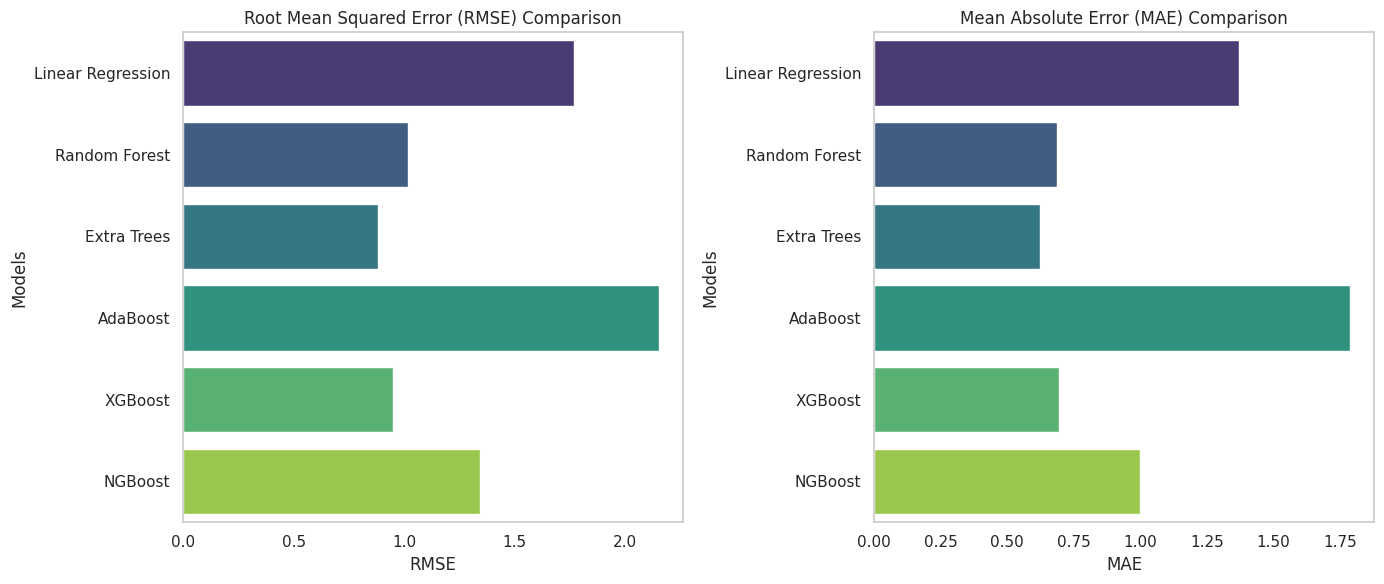
1. **Sum of Squared Errors (SSE)**: Measures the compactness of clusters; lower values indicate better-defined clusters.
2. **Silhouette Score**: Evaluates how similar an object is to its own cluster compared to other clusters.
3. **BetaCV**: Measures the ratio of intra-cluster cohesion to inter-cluster separation.
4. **Dunn's Index**: Assesses the ratio between the smallest distance between clusters and the largest intra-cluster distance.
5. **Hubert’s Statistic**: Analyzes the correlation between proximity matrices from data and clustering assignments, indicating clustering quality.

**Insights**

1. **Field Players**:
   * Clusters revealed distinct groups based on the balance between physical and technical attributes.
   * PCA and pairwise plots helped identify player classifications, correlating attributes to field positions.
2. **Goalkeepers**:
   * Two clusters highlighted distinct types of goalkeepers based on technical attributes.
   * PCA plots further clarified their separation.
3. **Comparison of Standardized vs Non-Standardized Data**:
   * Standardization improved clustering performance for models sensitive to feature scaling, such as K-Means and GMM.
   * Pairwise plots revealed more distinct boundaries in the transformed data, facilitating clearer player classification.

# Results and Interpretations:

# Regression:



The following are the results of the Regression:

                       RMSE       MAE

Linear Regression  1.767013  1.368468

Random Forest      1.015080  0.686874

Extra Trees        0.880751  0.621805

AdaBoost           2.153072  1.787748

XGBoost            0.949505  0.695465

NGBoost            1.343160  0.998671

# Classification:

Results on Original Dataset:

                       Precision    Recall  F1-score

Logistic Regression    0.959184  0.803419  0.874419

Naive Bayes            0.383275  0.940171  0.544554

KNN                    0.918605  0.675214  0.778325

Linear SVM             0.948980  0.794872  0.865116

Kernel SVM             0.961165  0.846154  0.900000

Decision Tree          0.902655  0.871795  0.886957

Basic Neural Network   0.919643  0.880342  0.899563

Pre-Processed Dataset Performance

Results on Pre-Processed Dataset:

                       Precision    Recall  F1-score

Logistic Regression    0.730769  0.147287  0.245161

Naive Bayes            1.000000  0.031008  0.060150

KNN                    0.762887  0.573643  0.654867

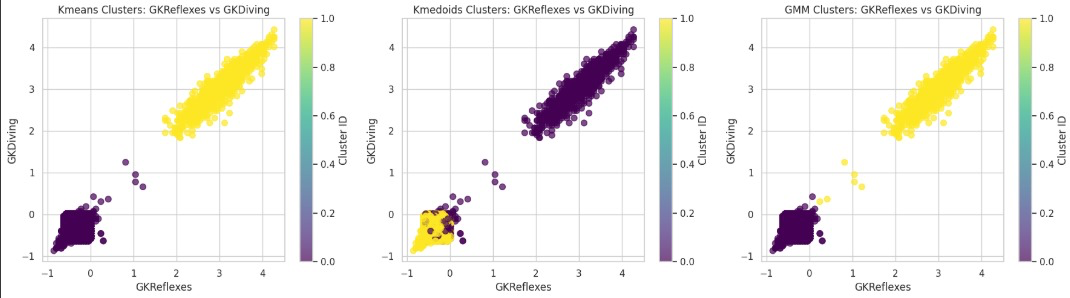
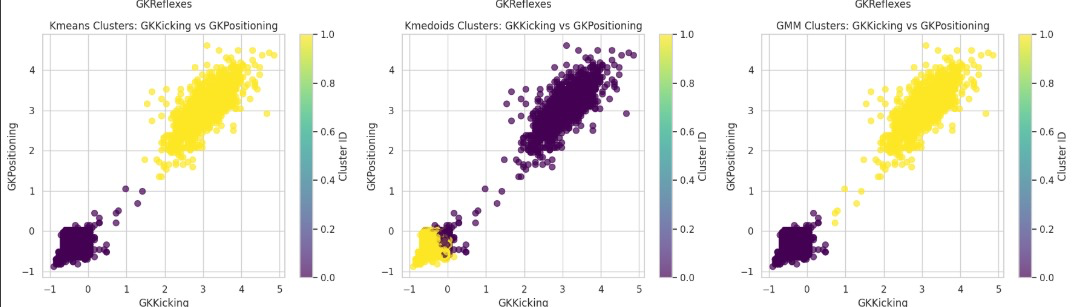
Linear SVM             0.000000  0.000000  0.000000

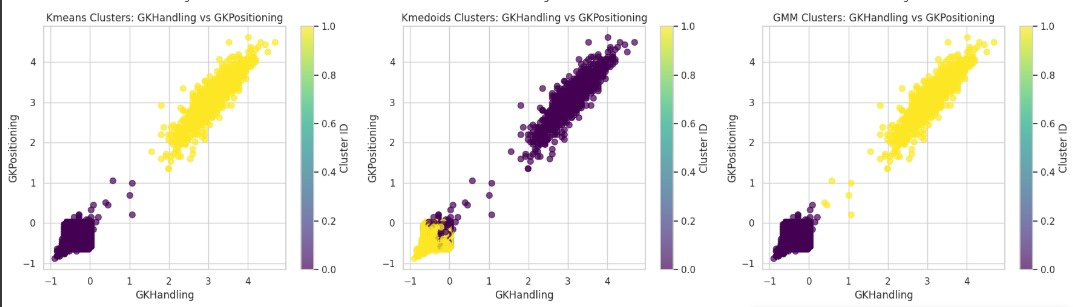
Kernel SVM             0.833333  0.542636  0.657277

Decision Tree          0.674797  0.643411  0.658730

Basic Neural Network   0.829787  0.604651  0.699552

# Clustering:



**Interpretation**

**Shot-Stoppers:**

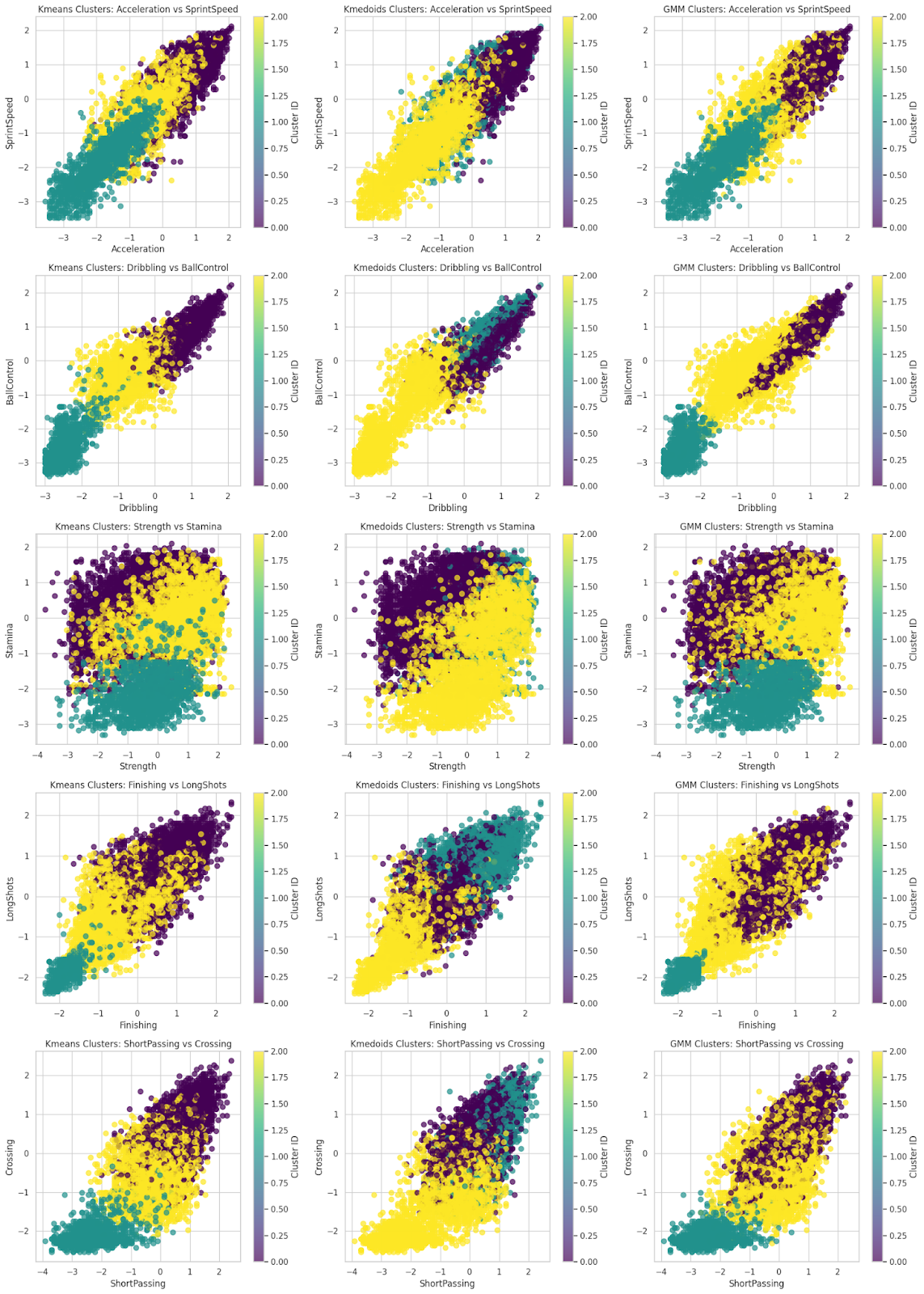
Expect to see one cluster with higher values in GKReflexes and GKDiving. Goalkeepers in this cluster likely excel in saving close-range shots and reacting quickly.

**Ball Distributors:**

One cluster might show higher values in GKKicking and GKPositioning. These goalkeepers are better at distributing the ball and positioning themselves well, potentially acting as playmakers from the back.

**All-Rounders:**

A cluster with moderate scores across GKHandling, GKPositioning, and other attributes indicates balanced goalkeepers. These players don't specialize in one area but are reliable in various scenarios.



For each of these feature pairs, here's what we interpret:

* Acceleration vs Sprint Speed: Cluster Interpretation: You might find a cluster with high acceleration but lower sprint speed, possibly indicating players like wingers who make quick, short runs. Another cluster with both high acceleration and sprint speed could indicate fast strikers or fullbacks.
* Dribbling vs Ball Control: Cluster Interpretation: A cluster with high dribbling and ball control likely represents highly skilled attackers or midfield playmakers, while a cluster with lower values might indicate defenders or goalkeepers.
* Strength vs Stamina: Cluster Interpretation: A cluster with high strength but lower stamina may represent central defenders or physically imposing forwards. Players with high stamina but moderate strength could be midfielders known for covering large distances.
* Finishing vs Long Shots: Cluster Interpretation: Players with high finishing and long shots are typically attacking midfielders or versatile forwards. A cluster with high finishing but low long shots could indicate poachers or penalty box strikers.
* Short Passing vs Crossing: Cluster Interpretation: A high short passing but low crossing cluster may represent central midfielders or playmakers. Conversely, high crossing but lower short passing may indicate wingers or fullbacks.