Assigning Football Players' Field Positions

Ishita (20222224) Jessica (2022230) Kartikeya (2022242) Medha (2022292)

Abstract

In football, predicting a player's future performance and market value is crucial for effective team management. This project aims to address this issue by developing a tool that can predict player positions and guide decision-making. Our approach involves reviewing existing work, analyzing a large dataset, implementing predictive models, and developing a tool that empowers football stakeholders to make informed decisions, ultimately increasing the chances of the team's success on the field.

1. Introduction

Football (soccer) is the most-watched sport globally, with player performance heavily influenced by the position they occupy on the pitch. When players are deployed in unfamiliar roles, their effectiveness can decline. A well-known example is Eden Hazard, who struggled with inconsistent performances when played as a striker (false 9) at Chelsea, compared to his natural position on the left wing.

Building a winning team goes beyond having talented players; it requires knowing their optimal positions and formations. Our project develops a tool that predicts ideal player positions based on skills, aiding soccer clubs in smarter recruitment, effective squad rotation, and personalized training. By analyzing traits such as speed, technique, movement, power, and defensive ability, the tool assists scouts in discovering talent and gives fantasy soccer fans a competitive advantage.

2. Literature Survey

- In [1]**PlayeRank**, Pappalardo et al.(2020) introduce a machine-learning framework that evaluates soccer player performance using a dataset of over 31 million event logs. This method moves beyond traditional metrics like goals and market value, considering players' roles on the field for a more nuanced analysis. The framework outperformed established performance metrics and professional scout assessments in accuracy, highlighting key traits of exceptional players and showing promise for improving player recruitment and versatility assessments in soccer.
- In the study [2]**Pricing Football Players using Neural Networks**, Sourya Dey develops a multilayer perceptron neural network to estimate football player prices based on data from over 15,000 players in FIFA 2017. By fine-tuning various parameters like activation functions, neuron setups, and learning rates, the model achieves an impressive top-5 accuracy of **87.2%** across 119 pricing categories, with an average deviation of only **6.32%** from actual prices.
- In [3]**FIFANet: Deep Learning to Predict Player Value,** the authors use deep learning to predict professional soccer players' future value and performance from FIFA video game data. They develop models based on attributes like age and position, with a top 5-layer neural network achieving 57% accuracy. Notably, **logistic regression** outperformed complex models with an accuracy of **62%**. This research demonstrates the potential of deep learning in sports analytics for better player development and transfer decisions.

3. Dataset Details & Preprocessing Techniques

3.1. Dataset Description

We are using the [4]**FIFA 23 Complete Player Dataset**, a 6GB resource to analyze player performance and predict future potential. This dataset contains 110 feature columns.

The dataset contains detailed information about football players, including **identifiers** like player ID, short name, height, and weight, along with **performance indicators** such as potential, market value, club position, and national position.

Key attributes include **technical skills** (pace, shooting, passing, dribbling, defending), **goalkeeper skills** (diving, handling, kicking, reflexes, speed, positioning), and **attacking skills**(crossing, finishing, heading accuracy, short passing, volleys). **Movement attributes** cover acceleration, sprint speed, agility, reactions, and balance, while **power attributes** assess shot power, jumping, stamina, strength, and long shots. **Mentality attributes** evaluate aggression, interceptions, positioning, vision, penalties, and composure, providing a comprehensive view of player capabilities.

3.2. Data Preprocessing

3.2.1 Position Mapping

Players with multiple positions were assigned their primary position. A mapping of positions was created:

- **Grid Mapping:** Various positions were grouped into 10 major categories such as `GK`, `LB`, `CB`, `RB`, etc.
- **Vertical Mapping:** Positions were grouped into three vertical categories: Goalkeepers (`GK`), Defenders (`B`), Midfielders (`M`), and Forwards (`F`).
- **Horizontal Mapping:** Positions were categorized based on their position on the field as `L` (Left), `C` (Center), or `R` (Right).

3.2.2 Feature Selection and Engineering

We dropped non-essential columns and created cumulative metrics by averaging relevant attributes to capture key performance areas. To reduce dimensionality, we removed the original features after calculating these metrics. This process resulted in the creation of six cumulative scores: **Goalkeeping, Attacking, Skill, Movement, Power, and Mentality.**

4. Exploratory Data Analysis

4.1. Dataset EDA

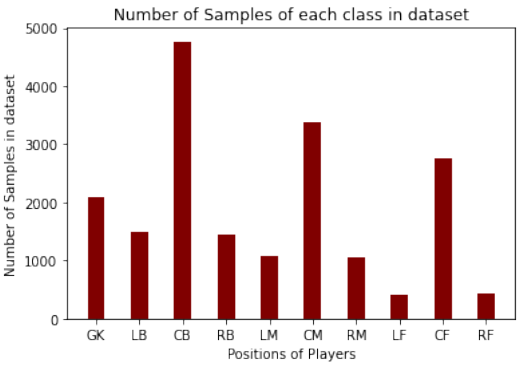


Figure 1. Number of samples per player position.

Figure-1 visualizes the number of players across various positions as a bar chart. By comparing the number of samples of each class, it is evident that 'LF' and 'RF' are the minority classes and 'CB' and 'CM' are the majority classes.

Figure-2 shows that more left-footed players play on the left side and more right-footed players play on the right side. Also, we infer that most players playing in the center are right-footed.

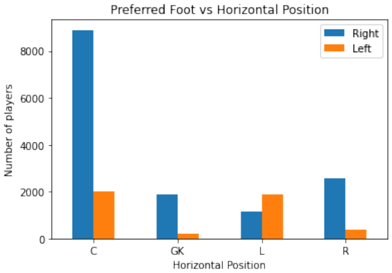
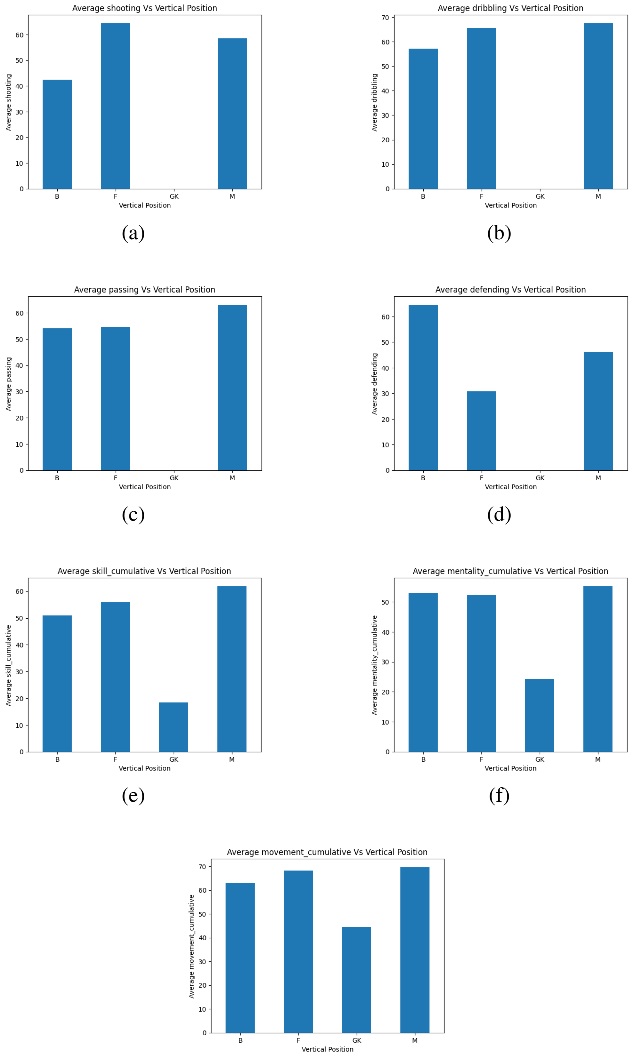


Figure 2. Number of samples per player position.



(a) (b) (c) (d) (e) (f) (g)

Figure 3. Analysis of player performance metrics across positions

Figure-3 implies that goalkeepers consistently show the lowest statistics, reflecting their focus on shot-stopping rather than offensive contributions. **(a)** shows that forwards lead in shooting averages, followed by midfielders. **(b)**, **(c)**

and **(g)** show that midfielders excel in dribbling and passing with the highest movement averages, showcasing their ability to control the game. They are followed by forwards. **(d)** shows that defenders rank highest in defending metrics. **(f)** shows that most players demonstrate strong mental attributes, though goalkeepers appear to face unique pressures affecting their resilience.

4.2. Visualisation using PCA

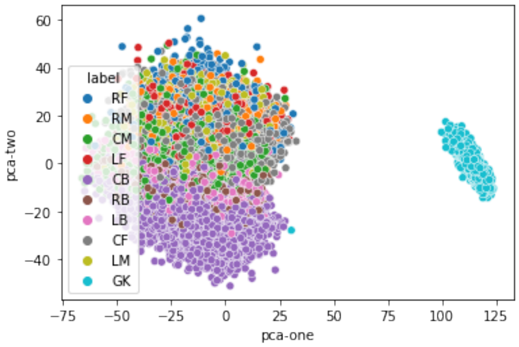


Figure 4. PCA 2D

Figure-4 PCA 2D plot shows distinct clustering of data points, with a larger cluster in the bottom left and a smaller one in the top right. Within the larger cluster, sub-clusters indicate further groupings among football player positions, while a few outliers are present. The distribution of labels suggests that certain positions share similar characteristics or playing styles based on the principal components.

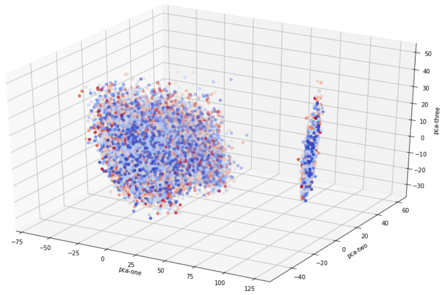


Figure 5. PCA 3D

Figure-53 3D PCA plot shows distinct clustering of data points, with a dense cluster in the bottom left and a more scattered cluster extending upwards along the PCA-three axis. Color variations within the clusters suggest correlations with the categorical variable represented by the colors. The data exhibits a general trend of increasing values along the PCA-three axis, alongside fluctuations and outliers. Two main clusters indicate the potential division of data into two groups based on the principal components.

5. Methodology and Model Details

- **Naive Bayes** We utilized the Naive Bayes algorithm to classify player positions. The ***gnb_single*** method performed single-output classification by fitting a Gaussian model to the training data, while the ***gnb_multi*** method handled multi-output classification by combining predictions from multiple outputs.

- **Random Forest:** We implemented the Random Forest algorithm for predicting player positions. It included the ***train_model_single*** method, which trains a single-output classifier, and the ***train_model_multi*** method, which managed multiple outputs. We optimized parameters with the ***gridSearch*** method for improved performance.

- **Decision Tree:** We implemented Decision Tree algorithm for classification tasks, implementing functionalities for training, hyperparameter optimization, and performance evaluation.

- **Multiclass Logistic Regression:** We implemented multiclass logistic regression model for training, evaluating, and optimizing. We trained both single and multiple models, evaluated their performance, and conducted hyperparameter optimization.

- **Multilayer Perceptron:** We implemented the MLP model incorporating methods for training, evaluating, and optimizing the model. Additionally, we provided hyperparameter optimization and customizable parameters to enhance classification effectiveness.

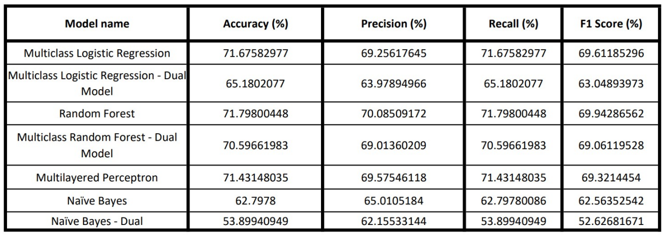
- **Boosting Algorithms:** The code implements three boosting algorithms: ***AdaBoost***, ***Gradient Boosting***, and ***XGBoost***. Each algorithm utilizes a base estimator, typically a decision tree, with varying hyperparameters to optimize performance. The models were trained on training data and evaluated on test data to determine accuracy. For each algorithm, the best-performing configurations were identified based on accuracy, and classification reports were generated.

6. Results and Analysis

6.1. Without Application of Boosting Algorithms

- The **Multiclass Logistic Regression** model demonstrates the highest accuracy. Its ability to model probabilities across multiple classes enables it to effectively distinguish between different football positions.

- The **Random Forest** model underperforms compared to logistic regression, likely due to its randomness and risk of overfitting without proper tuning. However, the **Multiclass Random Forest - Dual Model** shows



Model name	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Multiclass Logistic Regression	71.67582977	69.25617645	71.67582977	69.61185296
Multiclass Logistic Regression - Dual Model	65.1802077	63.97894966	65.1802077	63.04893973
Random Forest	71.79800448	70.08509172	71.79800448	69.94286562
Multiclass Random Forest - Dual Model	70.59661983	69.01360209	70.59661983	69.06119528
Multilayered Perceptron	71.43148035	69.57546118	71.43148035	69.3214454
Naïve Bayes	62.7978	65.0105184	62.79780086	62.56352542
Naïve Bayes - Dual	53.89940949	62.15533144	53.89940949	52.62681671

Figure 6. Results without Boosting Algorithm

slight improvement, suggesting that the dual approach may help reduce overfitting and enhance performance.

- The **Multilayer Perceptron (MLP)** captures complex relationships well but requires careful hyperparameter tuning and sufficient training data to achieve optimal performance.
- On the other hand, **Naïve Bayes** models (standard and dual) exhibit relatively low accuracy. This underperformance stems from the feature independence assumption, which likely does not hold in this context, leading to incorrect classifications.
- The **Decision Tree** model reveals a clear relationship between its maximum depth and accuracy. Initially, increasing the depth improves accuracy however, accuracy peaks around a depth of **9**, beyond which further increases result in diminishing returns and the risk of overfitting.

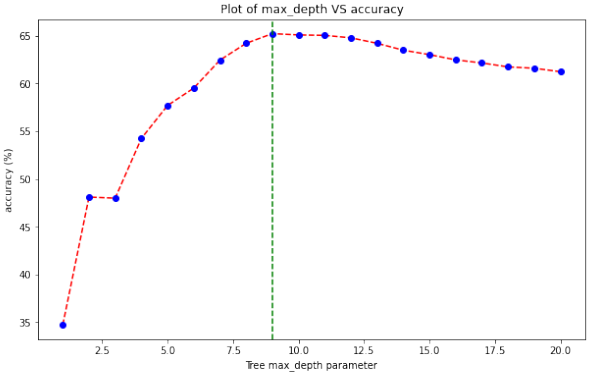
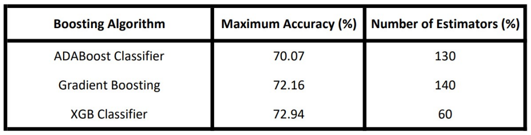


Figure 7. Decision Tree Max Depth vs Accuracy Curve

6.2. Application of Boosting Algorithms

With decision trees as the base classifier, the **XGB Classifier** with **60 estimators** achieves the highest accuracy. In contrast, increasing the estimators to **140** lowers the accuracy slightly, reflecting diminishing returns. The **AdaBoost Classifier** achieves an accuracy of **70.07%** with **130 estimators**, but it lags in both accuracy and efficiency. These results show that more estimators do not always guarantee better performance. Overall, the **XGB Classifier with 60 estimators** emerges as the most effective and efficient model.



Boosting Algorithm	Maximum Accuracy (%)	Number of Estimators (%)
ADABoost Classifier	70.07	130
Gradient Boosting	72.16	140
XGB Classifier	72.94	60

Figure 8. Results with Boosting Algorithm

7. Conclusion

This study finds that the Multiclass Logistic Regression model is the most accurate for classification tasks. The Random Forest model underperforms and Multilayer Perceptron requires careful tuning. Naïve Bayes models struggle due to independence assumptions, and Decision Trees peak at a depth of 9 before risking overfitting. The XGB Classifier with 60 estimators is the most effective among boosting algorithms. Further, we wish to implement more complex models to get better results.

8. Timeline

Yes we were able to follow the proposed timeline as we completed literature review by 27 August, 2023.

BEFORE MIDSEM:

- Sept 10 - Sept 15: Feature engineering
- Sept 22 - Sept 27: Data pre-processing
- Sept 30 - Oct 5: Exploratory Data Analysis (EDA)
- Oct 7 - Oct 13: Implementation of models
- Oct 17 - Oct 23: Applying boosting algorithms

AFTER MIDSEM:

- Oct 27 - Nov 15: Trying more complex models
- Nov 18 - Oct 23: Analyzing Results
- Nov 24 - Nov 26: Report & Presentation

9. Member Contributions

- Ishita (2022224): Data Preprocessing, Multi-Class Logistic Regression, Lit. Review[1], Report
- Jessica (2022230): Lit. Review[2], Data Visualization, Random Forests, Various Boosting Algorithms, Presentation
- Kartikeya (2022242): Naive Bayes, MLP, Presentation
- Medha (2022292): Lit.Review[3], Decision Trees, Results Analysis, Report

10. References

[1] https://arxiv.org/pdf/1802.04987

[2] https://arxiv.org/abs/1711.05865

[3] https://cs230.stanford.edu/projects_spring_2019/reports/18681023.pdf

[4] https://www.kaggle.com/datasets/stefanoleone992/fifa-23-complete-player-dataset?select=male_players+%28legacy%29.csv