Assigning Field Positions to Football Players using Machine Learning

Ishita (2022224), Jessica (2022230), Kartikeya (2022242), Medha (2022292)

Group Number = 52

ML Midsem Project Presentation

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Motivation

- Accurately predicting a soccer player's future performance and market value is a significant challenge in football, often leading to underperformance when teams acquire players who don't fit well with their existing squad.
- 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.

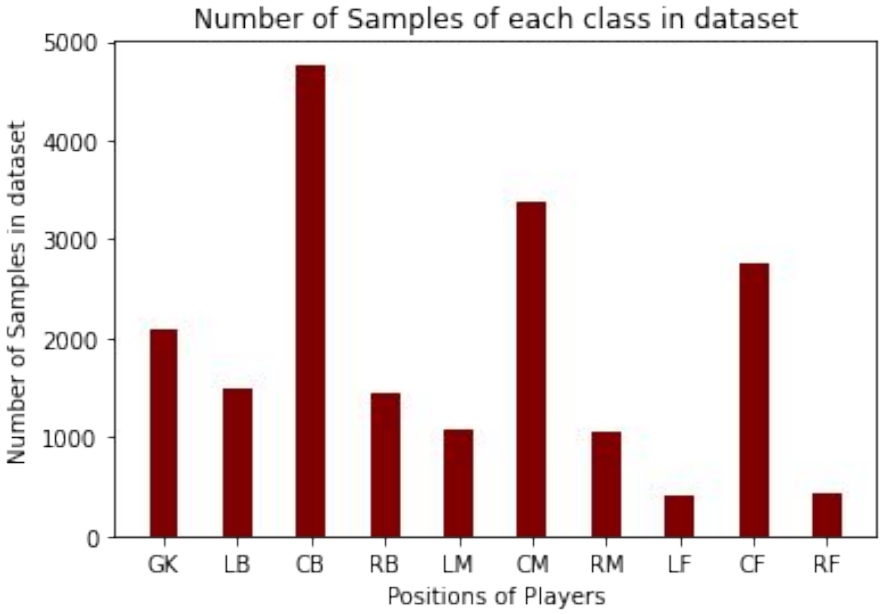
Literature review

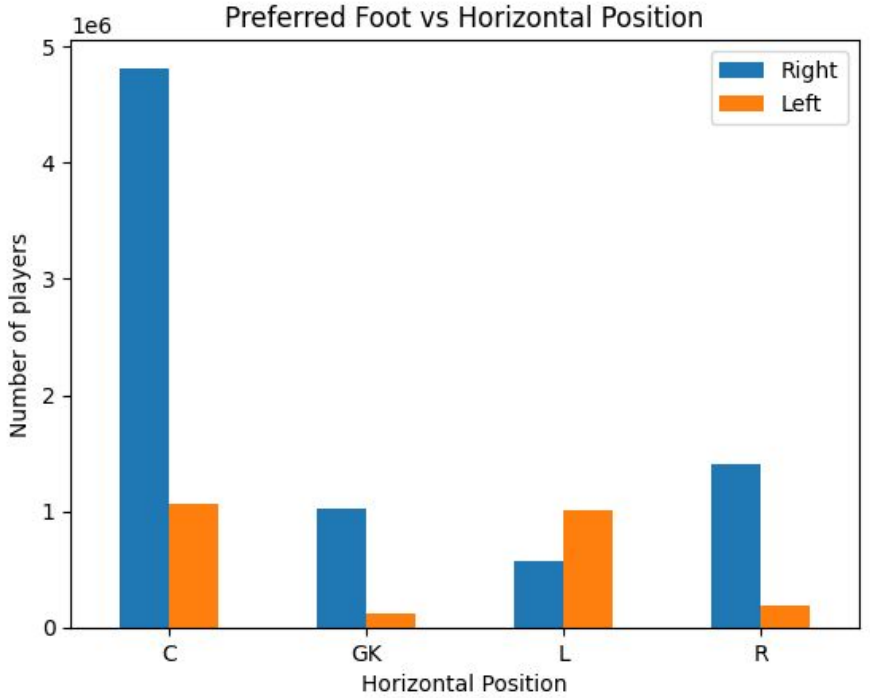
1. Research Work-1 [Link]: The paper proposes PlayeRank, a machine learning based framework for evaluating and ranking soccer players through performance data analysis revealing detailed insights into player abilities and versatility.
2. Research Work-2 [Link]: The paper presents a deep learning method for predicting football player performance developing a multilayer perceptron model with 87.2% accuracy among 119 pricing categories.categories.
3. Research Work-3 [Link]: The paper presents a machine learning model to predict soccer player transfer values using historical performance and market trends, aimed at improving financial decision-making.

Dataset Description

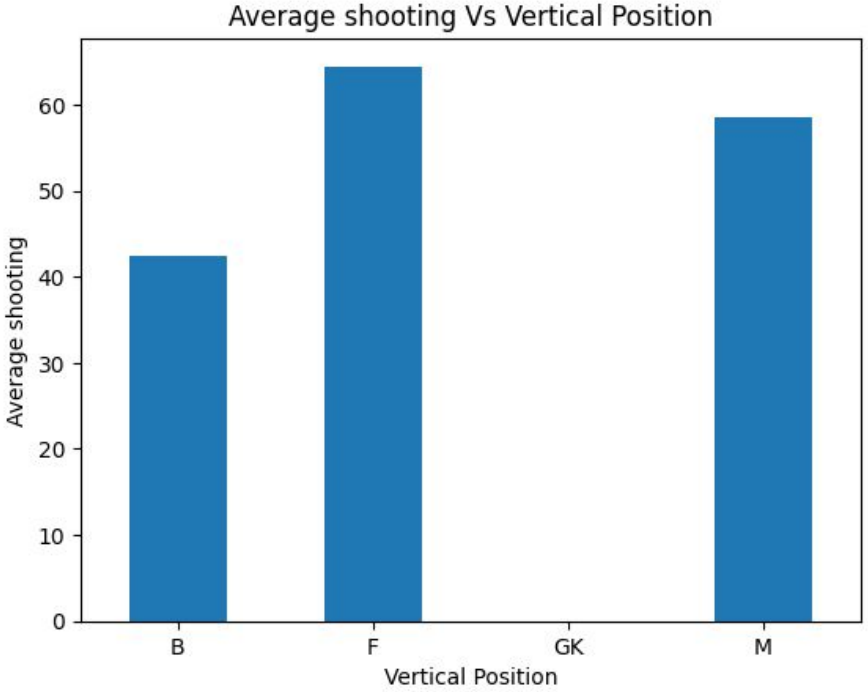
- We are using the **FIFA 23 Complete Player Dataset** (6GB) to analyze player performance and predict potential.
- This dataset includes 110 features, such as player ID, name, height, weight, potential, market value, and club and national positions.
- Key attributes are divided into **technical skills** (pace, shooting, passing, dribbling, defending), **goalkeeping 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.
- Finally, **mentality attributes** evaluate aggression, interceptions, positioning, vision, penalties, and composure, providing a comprehensive view of player capabilities.

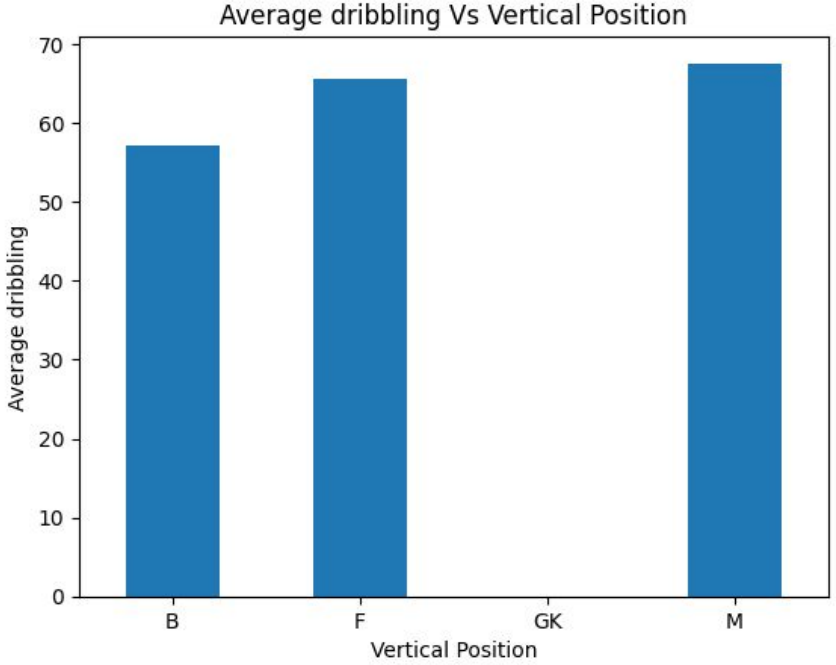
Data Visualization



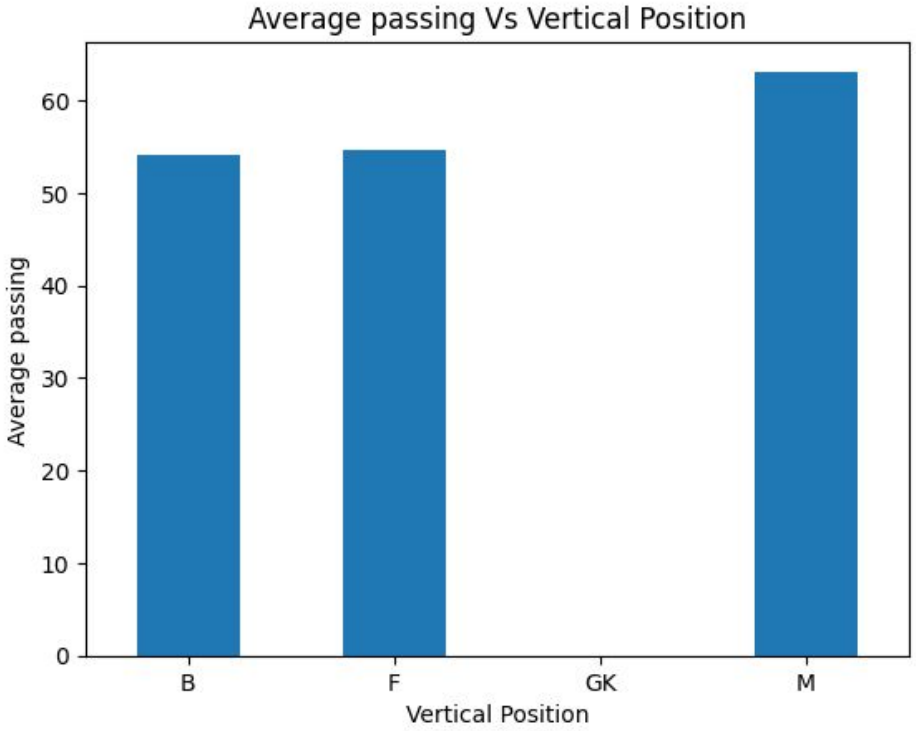


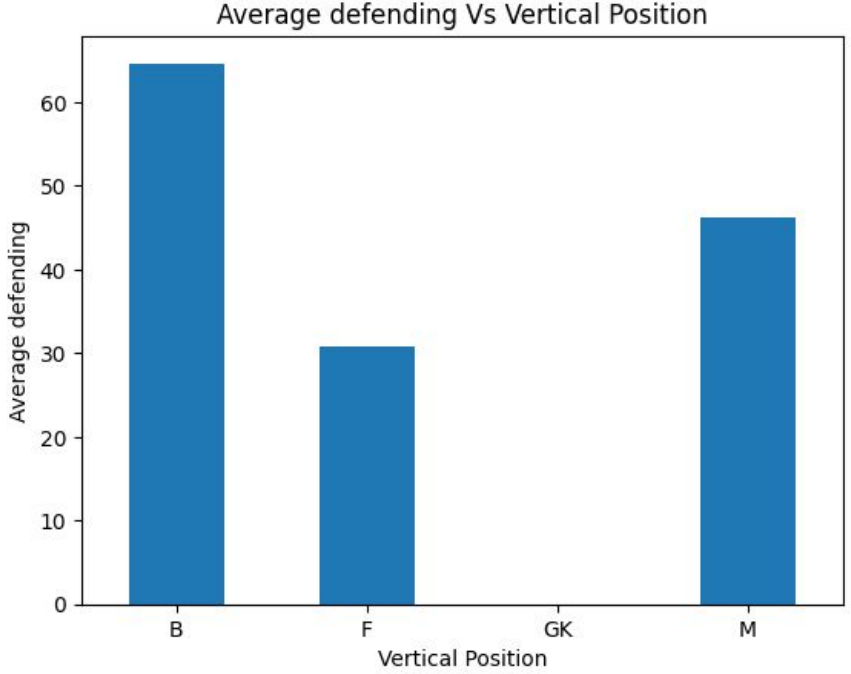
Data Visualization



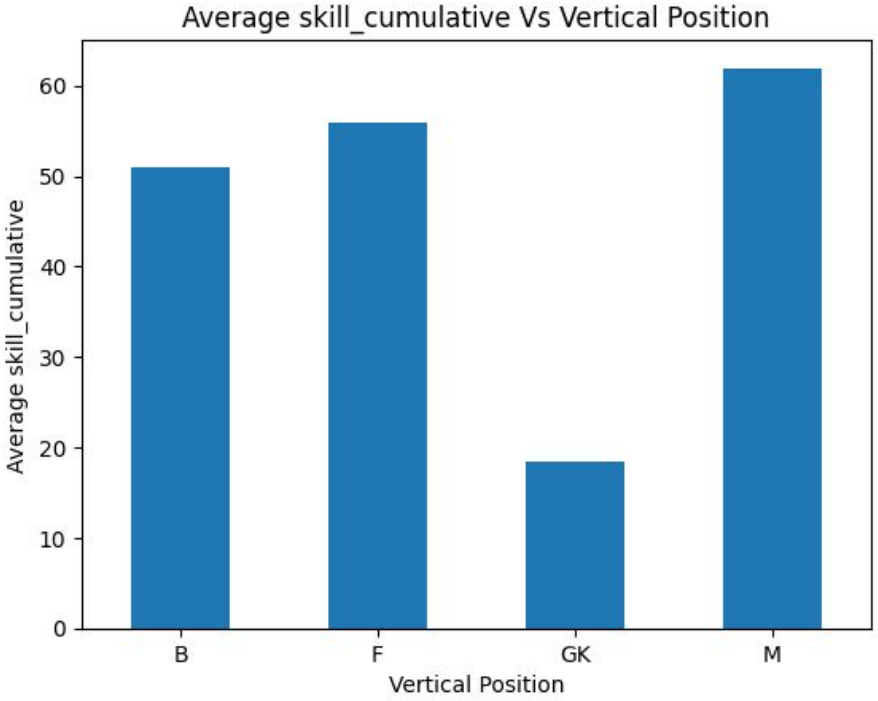


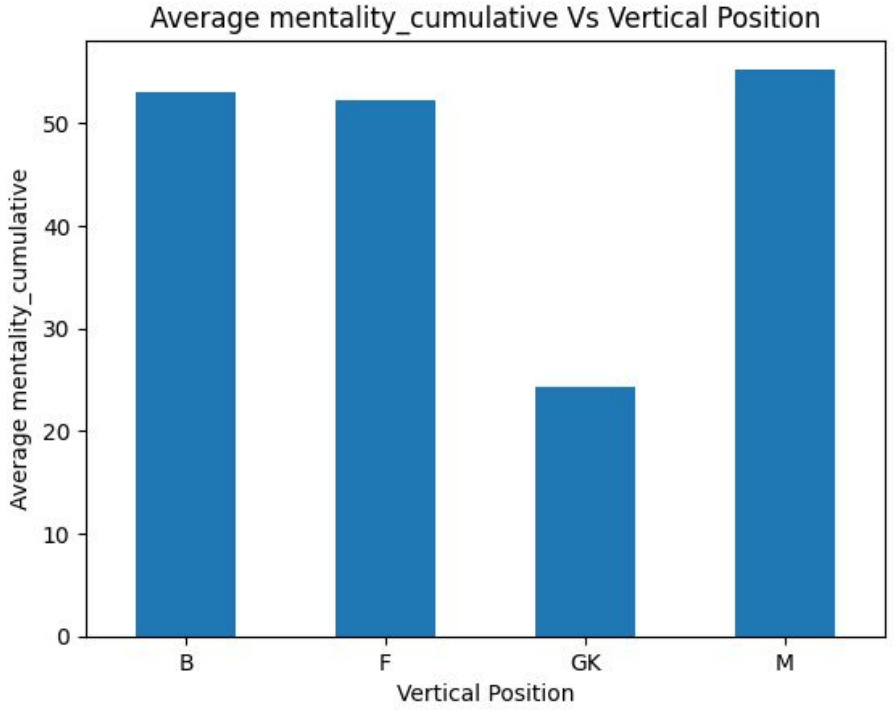
Data Visualization



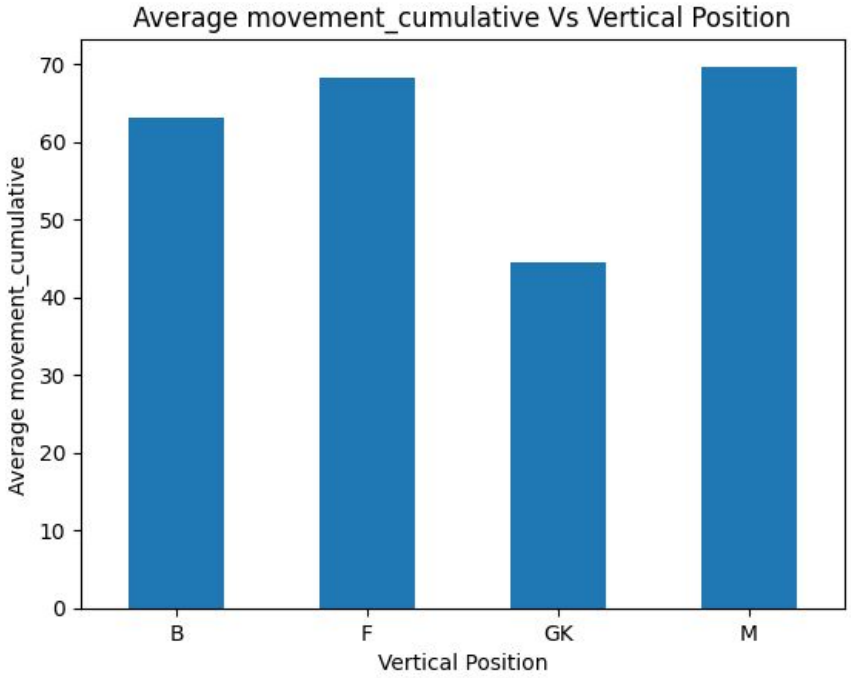


Data Visualization





Data Visualization



Data Preprocessing

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).

Position Mapping

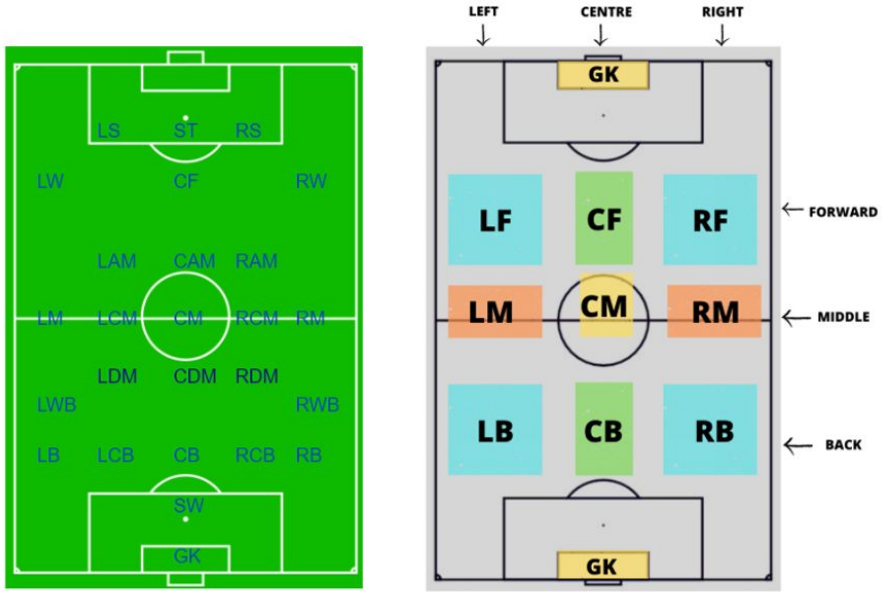


Figure 7: Position Mapping

Data Preprocessing

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.**

Methodology

- **Naive Bayes:** We utilized the NaiveBayes class to classify player positions using the Naive Bayes algorithm. 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:** Used for predicting player positions, incorporating methods for single and multi output classification, model evaluation, and hyperparameter tuning. The *train_model_single* method trained a single-output classifier, while the *train_model_multi* method handled multiple outputs. We optimized parameters with the *gridSearch* method for improved performance.

Methodology

- **Decision Tree:** The *DecisionTree* class employed decision tree algorithm for classification tasks, implementing functionalities for training, hyperparameter optimization, and performance evaluation. We handled attributes related to both training and testing data, along with predictions and critical hyperparameters that impacted model performance. Key methods included model initialization, identification of optimal parameters, training the model, generating predictions, and visualizing the structure and decisions of the decision tree.

Methodology

- **Multiclass Logistic Regression:** The *multiclass_logistic_regression* class allowed us to perform various tasks for training, evaluating, and optimizing a multinomial logistic regression model using scikit-learn. We trained both single and multiple models, evaluated their performance, and conducted hyperparameter optimization.
- **Multilayer Perceptron:** We implemented the *MLP_model* class to create a Multi-Layer Perceptron (MLP) for multi-class classification, incorporating methods for training, evaluating, and optimizing the model. Additionally, we provided hyperparameter optimization and customizable parameters to enhance classification effectiveness.

Methodology

- **Boosting Algorithms:** The code implemented three boosting algorithms: *AdaBoost*, *Gradient Boosting*, and *XGBoost*. Each algorithm utilized 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.

Results and Analysis

- The **Multiclass Logistic Regression** model shows the highest accuracy at 71.68%, effectively classifying data across multiple football positions. The dual variant performs similarly (71.43%), indicating that its added complexity doesn't significantly improve accuracy.
- The **Random Forest** model, with an accuracy of 63.98%, underperforms compared to logistic regression, possibly due to its random nature and overfitting if not properly tuned. The Multiclass Random Forest - Dual Model slightly improves accuracy to 65%, suggesting the dual approach helps reduce overfitting and marginally boosts performance.

Results and Analysis

- The **MLP** gives an accuracy of 62.79%. Although MLPs excel at capturing complex relationships, their performance depends on well-tuned hyperparameters and ample training data. The low accuracy is possibly due to insufficient training.
- Both **Naïve Bayes** models (standard and dual) yield relatively low accuracies of 53.89% and 52.63%, respectively. The poor performance of Naïve Bayes can be attributed to the assumption of feature independence, which may not hold true in this context, leading to incorrect classifications.

Results and Analysis

- The accuracy of a **decision tree** improves with increasing depth as it captures more complex patterns.
-

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

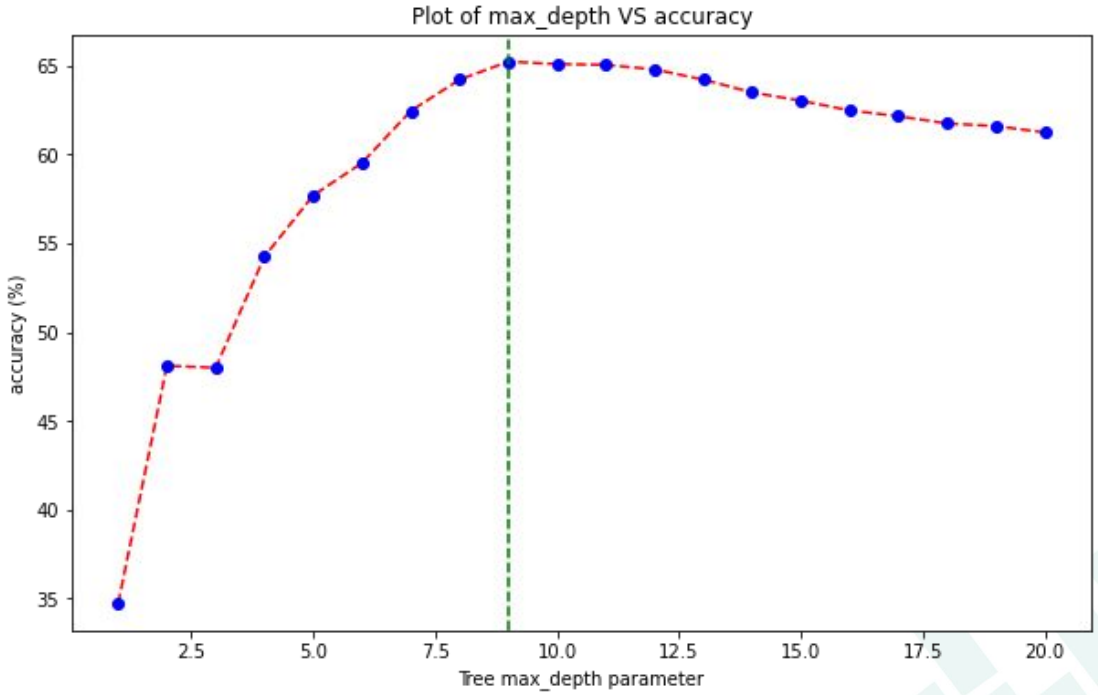
Results and Analysis

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

Result Plots

Decision Tree Plot

- Initially, increasing the depth improves accuracy.
- Accuracy peaks around a depth of 9.
- Beyond which further increases result in diminishing returns and the risk of overfitting.



Result Plots

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

Result Plots

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

Timeline

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

Individual Team Members' Contributions

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

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] Kaggle Dataset

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

Thank You