

# Recommend FIFA Players

## **Abstract:**

The goal of this project is to use the K-nearest neighbor model to recommend similar players in order to help clubs and managers to determine the best players to buy to increase the club value and start the road to glory. I worked with this dataset that is provided by Kaggle, I did some analysis and data manipulation to make the dataset better to read and understand as well as organizing the dataset. In addition, feature engineering was done along with knn model to get accurate recommendations.

## **Design:**

I got this idea from FIFA video game where I can be the manager of the team and start building my team with a budget limitation. Of course, this can also be used by real life clubs to help top managers take right decisions when buying new players. Recommending accurate players by a machine learning algorithm can support real clubs and FIFA game fans to take the right decisions to buy the most suitable players for their teams.

## **Data:**

The dataset contains 17,305 rows with a total of 93 features in which 9 categorical features. Many features have no proper use and around 47 features were dropped due to their useless. And to be clear only 2 features were used in the knn algorithm for similarity, in other words, these 2 features are the base of which knn algorithm will look for the nearest neighbor.

## **Algorithm:**

### **Feature Engineering:**

- 1- Dropping unneeded features and renaming other features
- 2- Converting a categorical feature (Player\_position) to binary dummy variables.
- 3- Classifying the Player positions into 4 classes (attacker – midfielders – defenders – keepers).

- 4- EDA was done, displaying unique values of categorical features and getting top 10 players of each attribute.

### Model:

The data was split to train and test by 80% to 20%. K-nearest neighbor model was the chosen algorithm to get the nearest neighbor of the recommender player. It was used as an unsupervised learning. In addition, K-nearest classifier was used to classify players based on 4 classes of positions. Moreover, naïve bayes classifier was also implemented to classify players into 4 classes of positions. After all, accuracy, recall and f1 scores were calculated for both classifiers to compare which is better.

#### Naive Bayes

Test Accuracy Score: 74.13905133203379

Test Recall Score: 74.13905133203379

Test F1 Score: 74.13905133203377

#### Naive Bayes

Train Accuracy Score: 75.38986354775828

Train Recall Score: 75.38986354775828

Train F1 Score: 75.38986354775827

#### K-Nearest Neighbor

Test Accuracy Score: 68.58349577647823

Test Recall Score: 68.58349577647823

Test F1 Score: 68.58349577647823

#### K-Nearest Neighbor

Train Accuracy Score: 85.7699805068226

Train Recall Score: 85.7699805068226

Train F1 Score: 85.76998050682259

### Tools:

Numpy and Pandas for data manipulation and EDA

Matplotlib and Seaborn for visualization

Scikit Learn for modeling

### Communication:

Power Point Slides will be presented. Also, you can see below recommendation result

```
recommendPlayers('Mbappé')
```

The Recommender Player: Mbappé | 91 | Rare | Gold - Rare | ST

The Recommended Players:

Owen | 91 | Icon | ST

Ronaldo | 91 | OTW | ST

Ronaldo | 91 | Rare | ST

Van Basten | 91 | Icon | ST