**IST 718 Project Report**

**Group 10**

# **FIFA 20 complete player dataset**

**Data: FIFA 2020 Complete Player**

**Group Member Names:**

**Yanning Fan, Jinlong Zhou, Zimeng Zhang, Kun Yang**

Contents

[Abstract 4](#_Toc57590213)

[2. Data Collection/ Cleaning / Exploration 5](#_Toc57590214)

[2.1 Dataset description 5](#_Toc57590215)

[2.1.1 Overview 5](#_Toc57590216)

[2.1.2 Size of dataset 5](#_Toc57590217)

[2.1.3 Sample columns description 5](#_Toc57590218)

[2.1.4 URL link to the dataset 5](#_Toc57590219)

[2.2 Data exploration 6](#_Toc57590220)

[2.2.1 Statistical summary of key numerical features 6](#_Toc57590221)

[2.2.2 Correlation heatmap of key numerical attributes 6](#_Toc57590222)

[2.2.3 Plot of value vs overall rating and age 7](#_Toc57590223)

[2.2.4 Plot of player count by position 7](#_Toc57590224)

[2.2.5 Interesting findings of dataset 7](#_Toc57590225)

[2.3 Data cleaning 8](#_Toc57590226)

[2.3.1 Drop useless and duplicated columns 8](#_Toc57590227)

[2.3.2 Features splitting 8](#_Toc57590228)

[2.3.3 Missing values 8](#_Toc57590229)

[2.3.4 Duplicate values 8](#_Toc57590230)

[2.3.5 Feature engineering for modeling 8](#_Toc57590231)

[**3. Regression Model** 9](#_Toc57590232)

[3.1 Regression 9](#_Toc57590233)

[3.1.1 Model type 9](#_Toc57590234)

[3.1.2 Data transformation 9](#_Toc57590235)

[3.1.3 Test result 9](#_Toc57590236)

[3.2 Random Forest Regression 9](#_Toc57590237)

[3.2.1 Model type 9](#_Toc57590238)

[3.2.2 Data transformation 9](#_Toc57590239)

[3.2.3 Test result 10](#_Toc57590240)

[3.3 Gradient Boosting Machine Regression 10](#_Toc57590241)

[3.3.1 Model type 10](#_Toc57590242)

[3.3.2 Data transformation 10](#_Toc57590243)

[3.3.3 Test result 10](#_Toc57590244)

[3.4 Inference 11](#_Toc57590245)

[**4. Classification Model** 11](#_Toc57590246)

[4.1 Methodology 11](#_Toc57590247)

[4.2 Model 11](#_Toc57590248)

[4.2.1 Multi-class Logistic Regression 11](#_Toc57590249)

[4.2.2 Random Forest Classification 12](#_Toc57590250)

[**5. Clustering model and Recommender System** 14](#_Toc57590251)

[5.1 Methodology 14](#_Toc57590252)

[5.2 Model 14](#_Toc57590253)

[5.2.1 Recommender system 14](#_Toc57590254)

[5.2.2 Clustering 16](#_Toc57590255)

[**6. Conclusion** 17](#_Toc57590256)

[**7. Appendix** 17](#_Toc57590257)

# Abstract

FIFA 20 is a football simulation video game published by Electronic Arts as part of the FIFA series. It is the 27th installment in the FIFA series, and was released on 27 September 2019 for Microsoft Windows. Throw this analysis we can understand how we select our FUT team for online play against rivals. For the project we are trying to provide a reference for player reference and salary evaluation. The results also can help the real word football club to decide like finding the best cost performance player.

From this project we will achieve three goals:1. How to improve value is a crucial question for both game players and real-life soccer players. What we do here is to figure out key factors to help players to increase the value most efficiently. 2. For the new players, in most cases they don’t know which is the best position for them. So, we are going to build the model to solve this question. 3. Sometimes players or managers want to pick the substitute for the team, but the question is there are thousands of players for them to pick. What we do here is to build a recommender system for helping them.

To achieve those goals, we made three specific predictions. 1. We use some regression model to predict the value of a player including linear regression, random forest, Gradient Boosting Machine Regression. 2. We use classification models to predict the best fitting position of a player. 3. We use k-means clustering to create a recommender system for alternate players.

For the inference, we use features like attacking\_volleys’, skill\_dribbling’, skill\_move’, ‘international\_reputation’ to predict value, position and alternative of a specific player

The model we build successfully defines the value of each player, can accurately find the position, and alternative of a specific player.

# 2. Data Collection/ Cleaning / Exploration

## 2.1 Dataset description

### 2.1.1 Overview

The dataset we are using for the project contains different statuses of a player in the video game. The performance of all players on the court is digitized into features in this data set. A recording of the data set includes information of a player, such as unique identity, physical information, ability status. A feature of the data set that it contains mixed types: numerical data like skill abilities, categorical data like position and text data like description of the players.

### 2.1.2 Size of dataset

Dataset contains 18,278 rows with 104 columns.

### 2.1.3 Sample columns description

|  |  |  |
| --- | --- | --- |
| **Feature** | **Type** | **Description** |
| height\_cm | num | Height in cm of a player |
| weight\_kg | num | Weight in kg of a player |
| shooting | num | Score of a player's shooting ability |
| players\_position | str | Position of a player |
| players\_traits | text | Description of the play style |

### 2.1.4 URL link to the dataset

<https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset?select=players_20.csv>

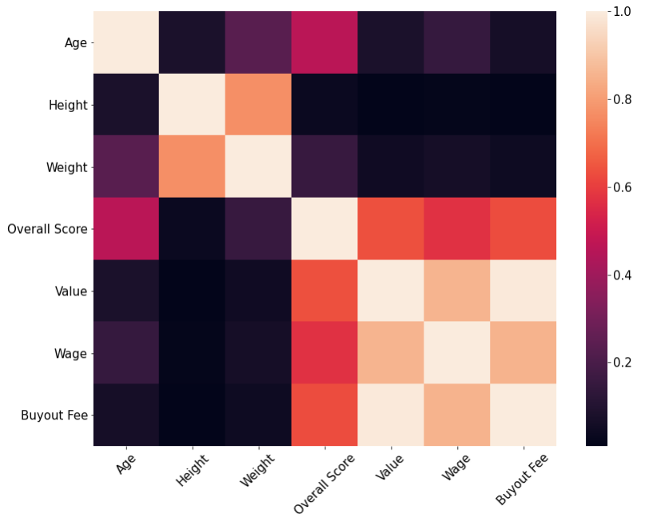
## 2.2 Data exploration

### 2.2.1 Statistical summary of key numerical features



From the table, we can find out that there are no obvious outliers existing in our dataset.

### 2.2.2 Correlation heatmap of key numerical attributes



This is the heatmap of correlation for age, height, weight, overall score, value, wage and buyout fee. We can find out that the three-money related features which are value, wage and buyout fee have high correlation which conform to common sense.

### 2.2.3 Plot of value vs overall rating and age

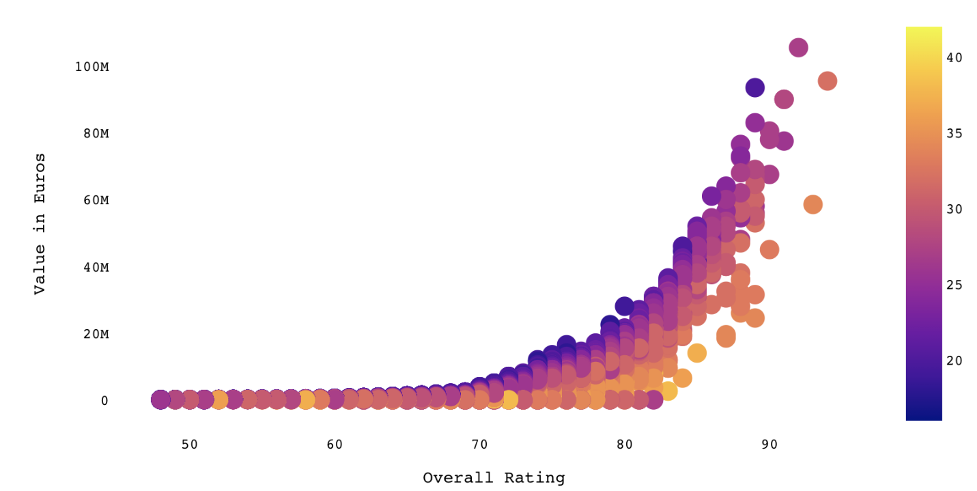
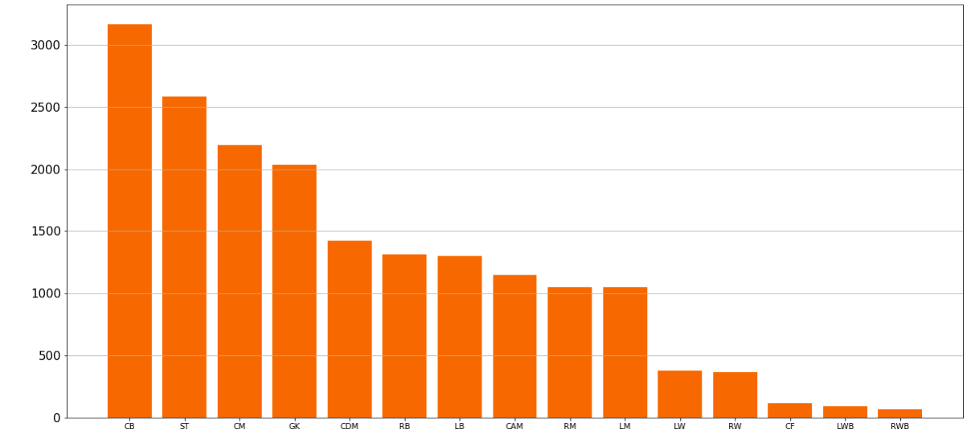


Figure is the chart of one of our targets, value, corresponding to overall score and age. The lighter the color, the older the player. And based on this chart, we can easily find there is a positive linear pattern between value and overall score.

### 2.2.4 Plot of player count by position



This is another target position and this plot just shows the number of players per specific position. From this chart, we know that the CB, ST, and CM are the most positions.

### 2.2.5 Interesting findings of dataset

One interesting point here is that if we check the country with the most players, England, is 400 more than the second highest, Germany, although the population of Germany is 83.02 million and England has only 55.98 million.

Another finding is that in common sense, there is only one goalkeeper on a team, but based on the bar chart, the goalkeepers’ total players are the fourth highest in all positions.

## 2.3 Data cleaning

### 2.3.1 Drop useless and duplicated columns

Firstly, we drop 11 columns which are ‘sofifa\_id’, ‘player\_url’, ‘long\_name’, ‘player\_tags’, ‘player\_traits’, ‘real\_face’, ‘nation\_jersey\_number’, ‘team\_jersey\_number’, ‘loaned\_from’, ‘joined’, ‘contract\_valid\_until’ because these are irrelevant with our targets and some of them has no realistic meaning.

Secondly, from figure 1.1 we can find that the value, wage and buyout fee have high correlation and we decide to choose value only and drop others.

Thirdly, some ability features are the average of others, so we decide to drop these columns.

### 2.3.2 Features splitting

For column player\_positions and work\_Rate, the sample records are [RW, CF, ST] and [Medium/Low] and for player\_positions, we use the first position as primary position and for work\_Rate, we divide it to two columns, Attack\_workrate and Defend\_workrate.

### 2.3.3 Missing values

No missing value after we checked the data.

### 2.3.4 Duplicate values

No duplicated value after we checked the data.

### 2.3.5 Feature engineering for modeling

After preprocessing the data as above stated, there are only five categorical features. For Attack\_workrate and Defend\_workrate, we use StringIndexerModel.from\_labels to numerlize because we want convert it by common sense which is High is the biggest and low is the smallest, and for these two features, we are not going to use one hot encoder because the number do has a meaning.

For preferred\_foot, body\_type and Positon\_General, we use StringIndexer and OneHotEncoder to numerlize.Besides, we standardize our features based on the model we will use, such as K-Means clustering.

# **3. Regression Model**

## 3.1 Regression

In the regression model, we use all the numerical variables to predict the player’s value and build three regression models to predict the player’s value.

### 3.1.1 Model type

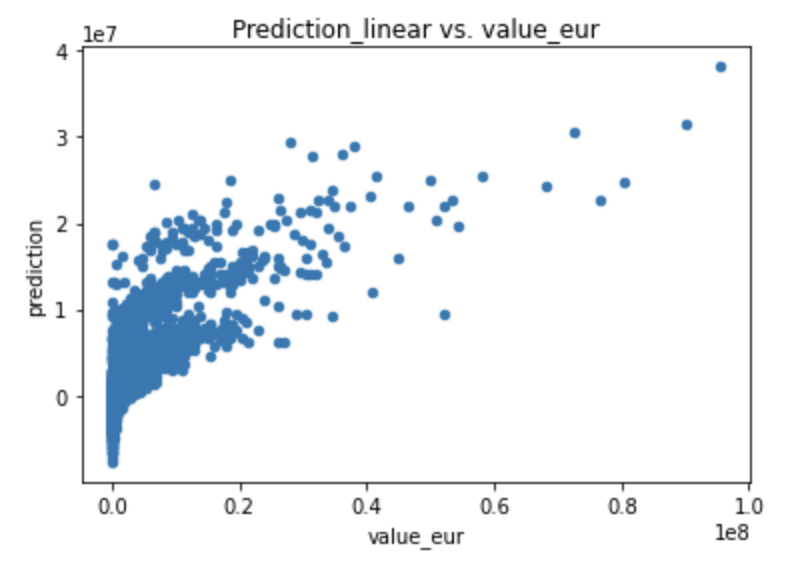
Linear Regression

### 3.1.2 Data transformation

There is no data needed for linear regression model transformation.

### 3.1.3 Test result

We added regularization to linear regression and did grid search to find the best model. After doing grid search, we find that the best model is when RegParam = 0.75, ElasticNetParam = 0, MaxIter = 10. The method we scored the model on is using R^2 and MSE score. The R^2 of the linear regression model is 0.546 and the mse score is 12653485714529.701. The plot shows the relationship prediction\_linear and value\_eur.



## 3.2 Random Forest Regression

### 3.2.1 Model type

Random Forest Regression

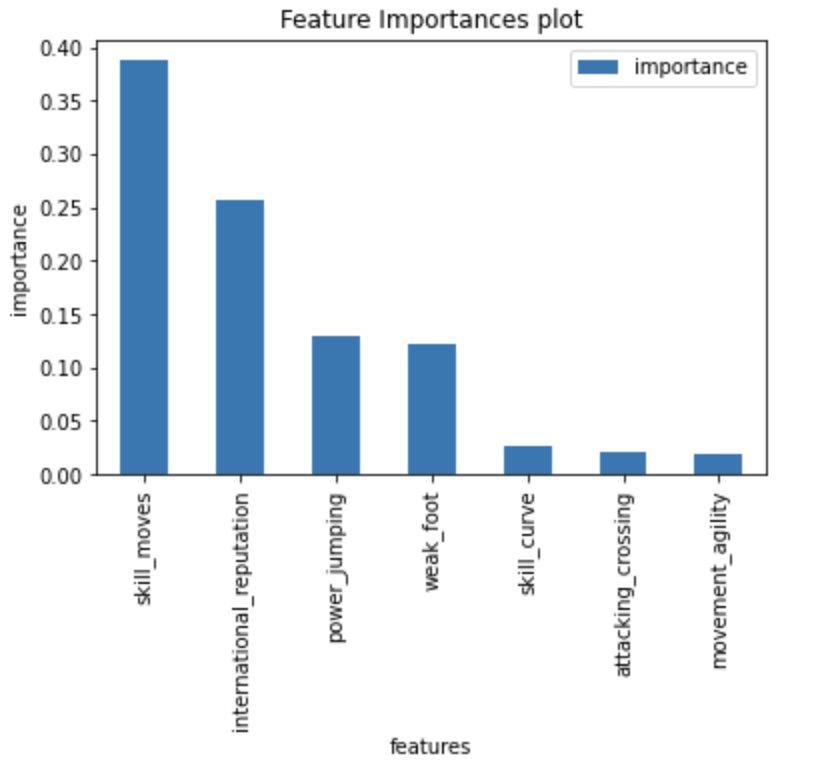
### 3.2.2 Data transformation

There is no data needed for random forest regression model transformation.

### 3.2.3 Test result

Add regularization to linear regression and do grid search to find the best model. And do grid search we find that the best model is when NumTrees = 100, MinInstancesPerNode = 10, MaxDepth = 20, SubsamplingRate = 0.2.

The method we scored the model on is using R^2 and MSE score. The R^2 of the random forest model is 0.9318 and the mse score is 1899928037170.8784. Here the distribution shows the important features that >0.1.



## 3.3 Gradient Boosting Machine Regression

### 3.3.1 Model type

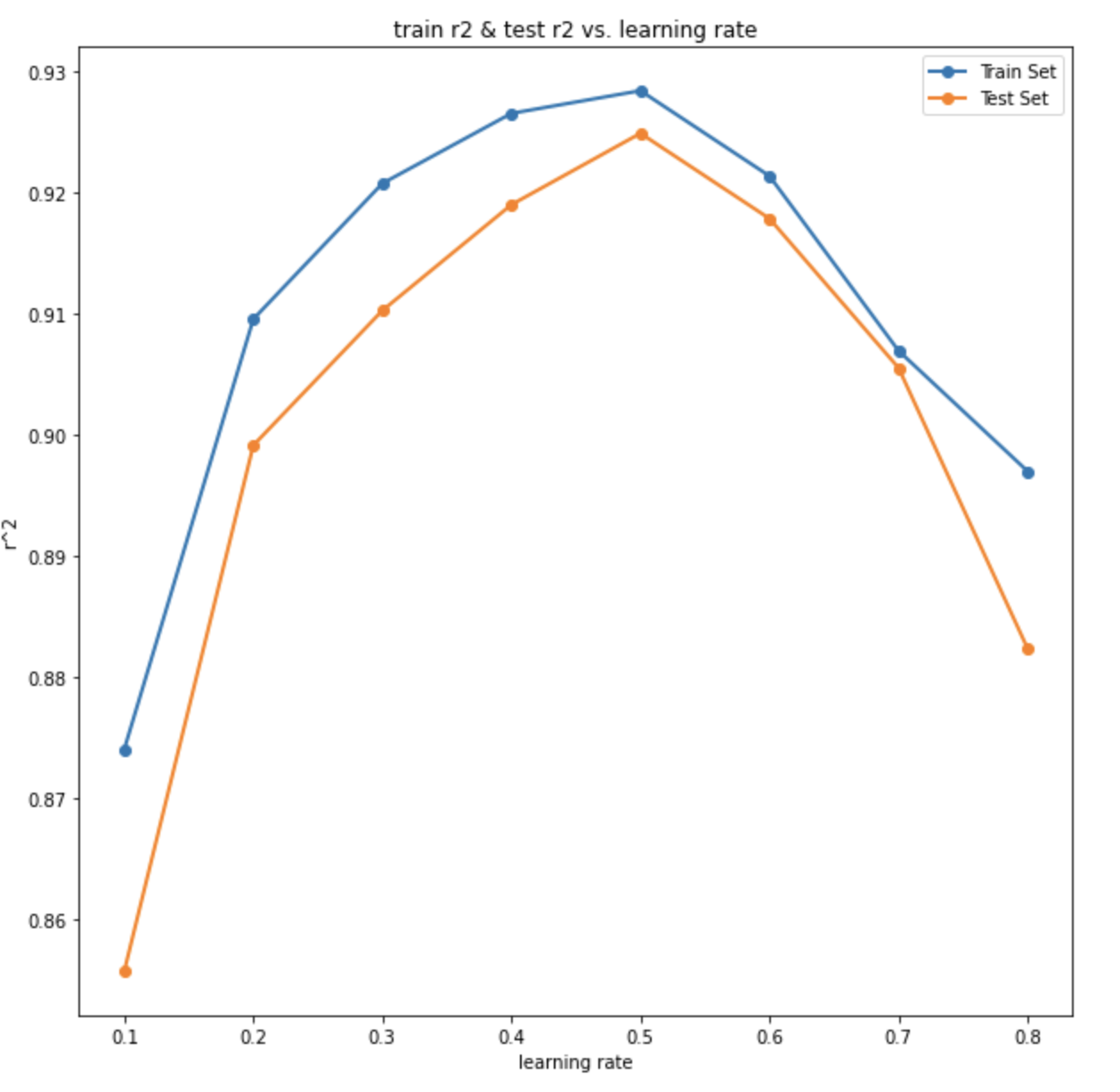
Gradient Boosting Machine Regression

### 3.3.2 Data transformation

There is no data needed for GBT model transformation.

### 3.3.3 Test result

Build the GBT model with maxdepth = 10, stepSize=0.5, maxIter=4. Test the model with MSE and R^2. The MSE is 7925517095656.42 and R^2 is 0.72. The plot shows the relation learning rate with R^2.



## 3.4 Inference

We use all the numeric variables to predict the player's value.

# **4. Classification Model**

## 4.1 Methodology

In this part, we conducted multi-class logistic regression and random forest classification. Firstly, there is no need for preprocessing for classification, so we didn’t do pre-procession here. Although standardization can accelerate the process, we didn’t conduct that procedure. Afterwards, we used vectorassembler to create feature columns and apply it into the model with default parameters. We used f1 score and accuracy to evaluate the performance of the model. Then we used grid search to find the best parameters to make f1 score and accuracy largest and recall by label to evaluate precision. Finally, the model could produce the coefficient or importance to obtain inference and apply the data we can get the prediction we want.

## 4.2 Model

### 4.2.1 Multi-class Logistic Regression

#### 4.2.1.1 Model type

Multi-class Logistic Regression

#### 4.2.1.2 Data transformation

There is no need for Multi-class Logistic Regression to do the data transformation.

#### 4.2.1.3 Choose Best Parameters

When we used the default parameters in the model, the result is as following:

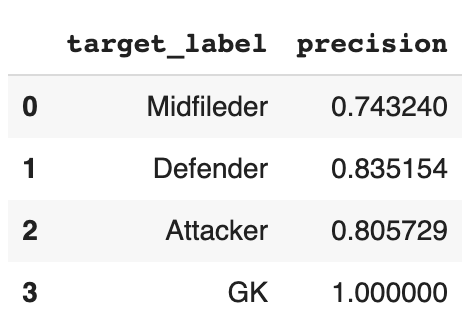
|  |  |
| --- | --- |
| **Basic model f1** | 0.7960171869417222 |
| **Basic model accuracy** | 0.796078431372549 |

After we used elastic regularization and grid search by cross-validator to find the best parameters to make f1 score and accuracy largest, we obtained the result is as following:

|  |  |
| --- | --- |
| **Lambda** | **0.0** |
| **Alpha** | **0.0** |
| **F1 after tuning** | **0.7960171869417222** |
| **Accuracy after tuning** | **0.796078431372549** |

#### 4.2.1.4 Test Results

The Output Precision by Label



As we can see from the table, the precision of the goalkeeper player has a large difference from that of other positions.

### 4.2.2 Random Forest Classification

#### 4.2.2.1 Model type

Random Forest Classification

#### 4.2.2.2 Data transformation

There is no need for Random Forest Classification to do the data transformation.

#### 4.2.2.3 Choose Best Parameters

When we used the default parameters in the model, the result is as following:

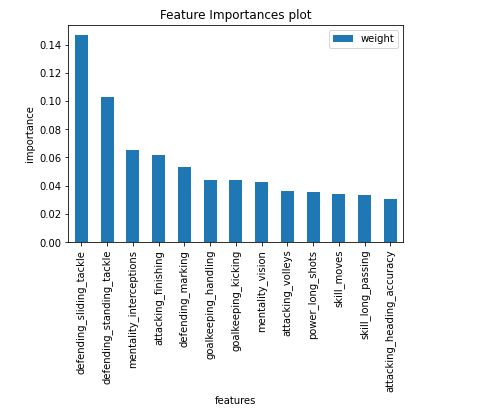
|  |  |
| --- | --- |
| **Basic model f1** | 0.838273254114523 |
| **Basic model accuracy** | 0.796078431372549 |

After hyperparameter tuning, we obtained the result is as following:

|  |  |
| --- | --- |
| **NumTrees** | 100 |
| **MaxDepth** | 20 |
| **Impurity** | entropy |
| **F1 after tuning** | 0.8844617010192888 |
| **Accuracy after tuning** | 0.8848484848484849 |

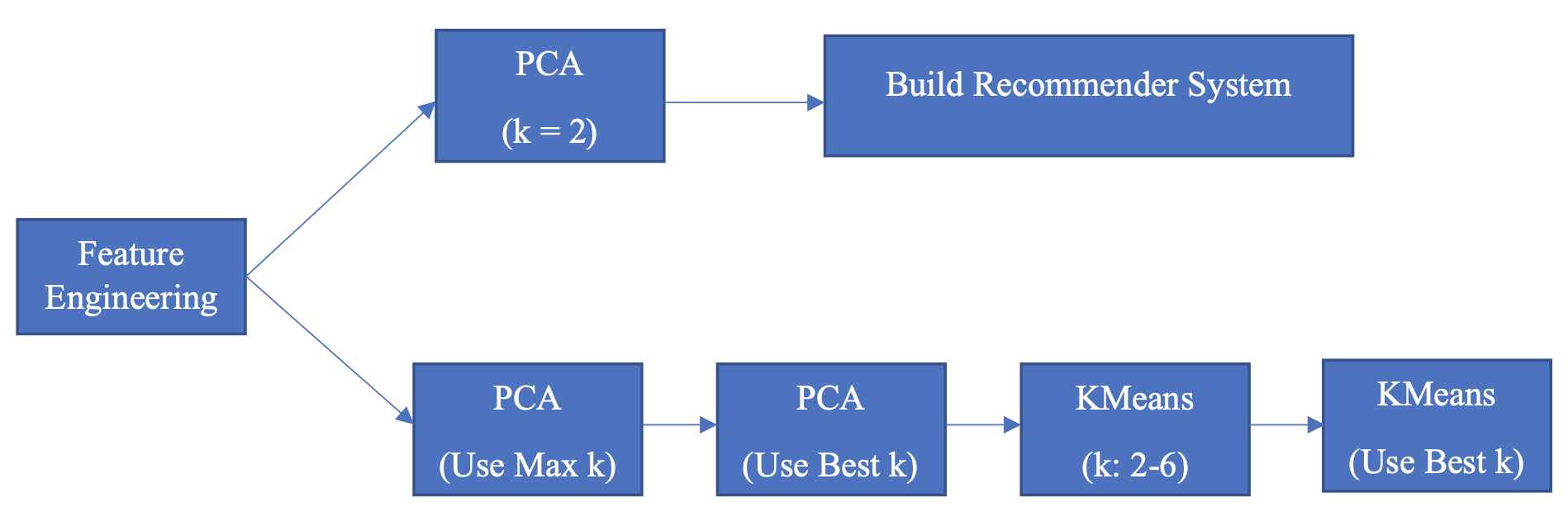
#### 4.2.2.4 Inference analysis

As we can see from the bar chart, the 3 most important features to determine the position of a player to play are “sliding tackle”, “standing tackle” and “interceptions”.



# **5. Clustering model and Recommender System**

## 5.1 Methodology



First of all, for clustering and recommender systems, the standardization is the most important because all the models will use in future steps are based on distances.

For building recommender systems, we firstly build two principal components analysis and then build the system.

For clustering, we firstly use maximum k for PCA and choose a best k for future programming. Secondly, we build a k-means model using k equal 2 to 6 and based on Silhouette Score choosing the best k for the final clustering.

## 5.2 Model

### 5.2.1 Recommender system

#### 5.2.1.1 Model type

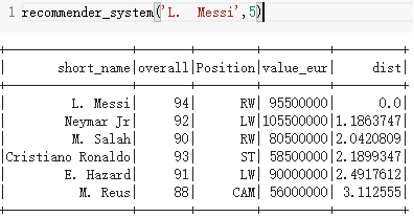
L2 distance

#### 5.2.1.2 Data Transformation

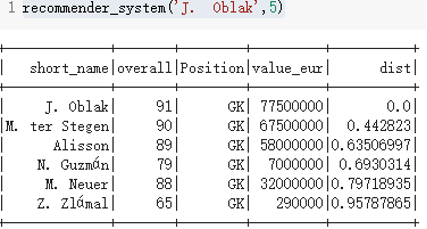
Because this system is based on L2 distance, we are going to use two principal components PCA to reduce the dimension of our dataset. Before PCA, we also standardized the features we will use.

#### 5.2.1.3 Test Results

* recommender\_system('L. Messi',5)



* recommender\_system('J. Oblak',5)



#### 5.2.1.4 Conclusion

From the three test results stated above, the system works well. For the player in a different position, the system recommends the real alternative players.

#### 5.2.1.5 Inference analysis

|  |  |
| --- | --- |
| **Features** | **Loading** |
| attacking\_volleys | 0.210889 |
| skill\_dribbling | 0.206001 |
| mentality\_vision | 0.202503 |
| movement\_agility | 0.193173 |
| attacking\_crossing | 0.19311 |

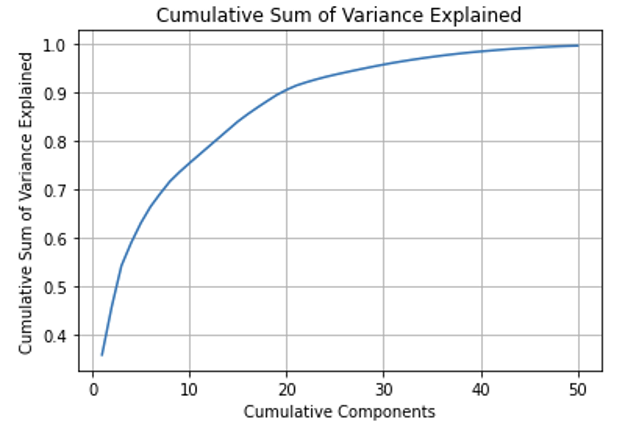
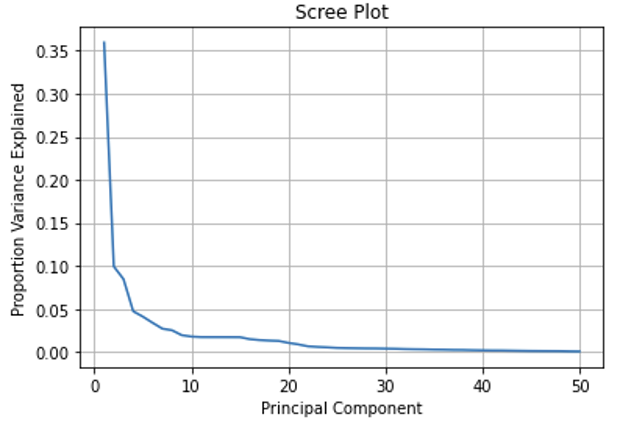
From the table above, the most feature is attacking\_volleys which is a very important ability in the real soccer match and looking through the whole table, all features are reasonable.

### 5.2.2 Clustering

#### 5.2.2.1 Model type

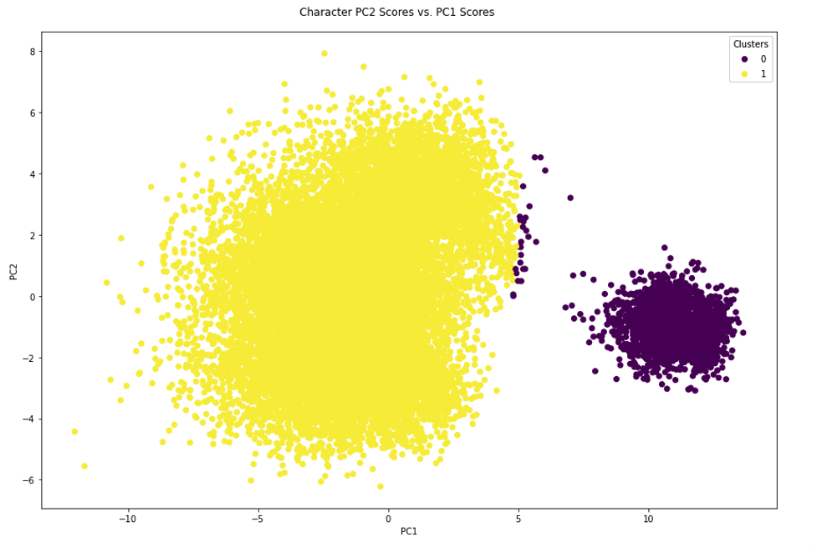
K-Means Clustering

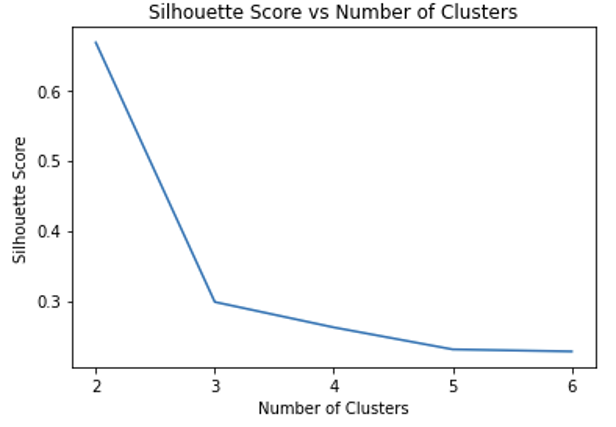
#### 5.2.2.2 Data Transformation

Firstly, we tried the maximum k for PCA which is the minimum of number of rows -1 and number of features. Here is a minimum of 18278 and 50, so we choose 50.

Based on figures above, we choose the best k that is 30. It represents about 95% variance of the original dataset and the competent after 30 has almost 0 variance.

#### 5.2.2.3 Choose best number of clusters and result

We use the Silhouette Score to find out the best number of clusters when we train our k-means model using k equals 2 to 6. After checking the Silhouette Score from figure 2.6 stated below, we choose the best number of clusters is 2.



# **6. Conclusion**

In this project, we utilize players’ physics and skills status like, ‘attacking\_volleys’, ‘skill\_dribbling’ completed three goals.

To predict the value of a specific player, we do three regression models. The best one is Random Forest regression which mainly takes use of features like ‘skill\_move’, ‘international\_reputation’ to generate models. Random forest outperformed other algorithms with a r square 0.93

To predict the potential position of a player, we construct 2 classification models. The best one is Random Forest classifier which takes use of features like “sliding tackle”, “standing tackle” and “interceptions” and reaches an accuracy and f1-score of around 0.88.

To recommend an alternative of a specific player, we construct a K-means clustering model and visualize it with PCA. From tuning, we know when K equals to 2, the model performs the best. And features like attacking\_volleys, skill\_dribbling, mentality\_vision influenced principal components most.

In summary, the random forest regression model is a very informative model covering a high level of knowledge provided, which can be used to predict the value/wages of a player. Among the classification models, the performance of random forest classification, whose accuracy is over 0.88, is better than the performance of multi-class logistic regression. By this classification, we can figure out the question we raised before, how to predict a position for a player regarding all his other features. At the same time, the recommender system works really well, who can resolve our questions we mentioned before. This system can recommend alternative players for managers if one player is absent, which is useful and economically meaningful. This analysis successfully answers our questions we mentioned before and provides valuable insight to the FIFA 2020 player data.

# **7. Appendix**

1. Tibshirani, R. (2014). An introduction to statistical learning: With applications in R. Springer.
2. Overview - Spark 3.0.1 documentation. (n.d.). Apache Spark - Unified Analytics Engine for Big Data. https://spark.apache.org/docs/latest/
3. Vanderplas, J. T., & VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media.