**IST 707 Project Report**

# **FIFA 20 complete player dataset**

**Data: FIFA 2020 Complete Player**

**Group Member Names:**

**Tiehao Chen, Zimeng Zhang, Kun Yang**

# 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>

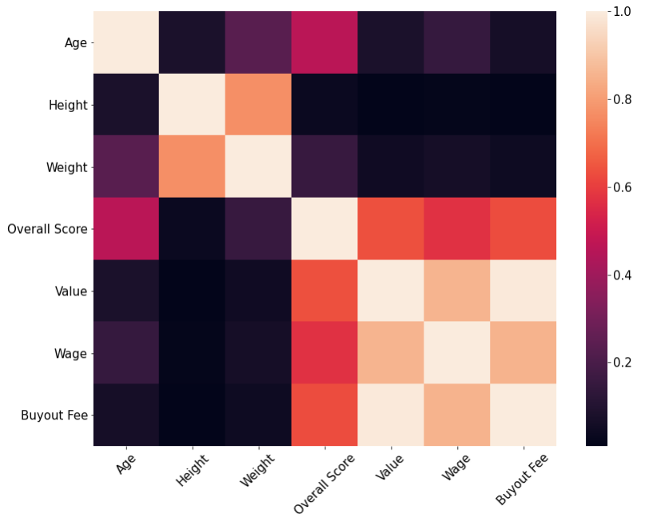
## 2.2 Data exploration

### 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

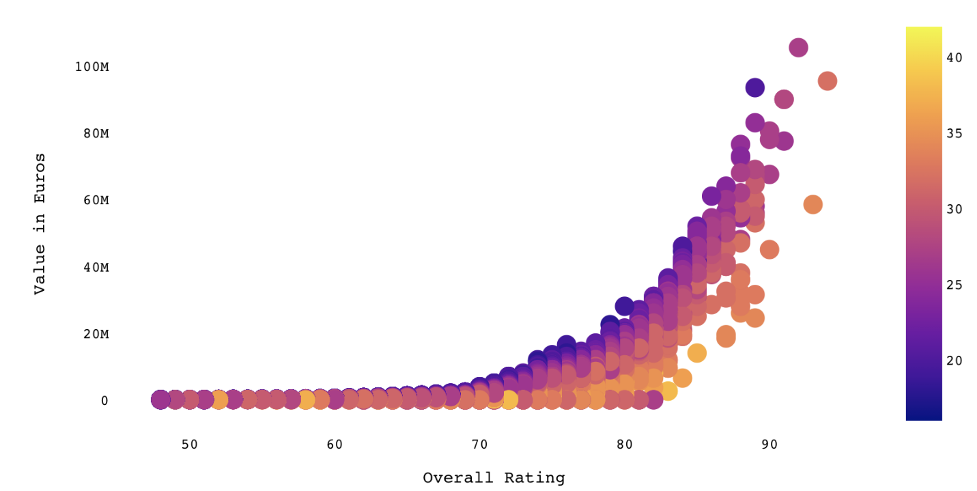
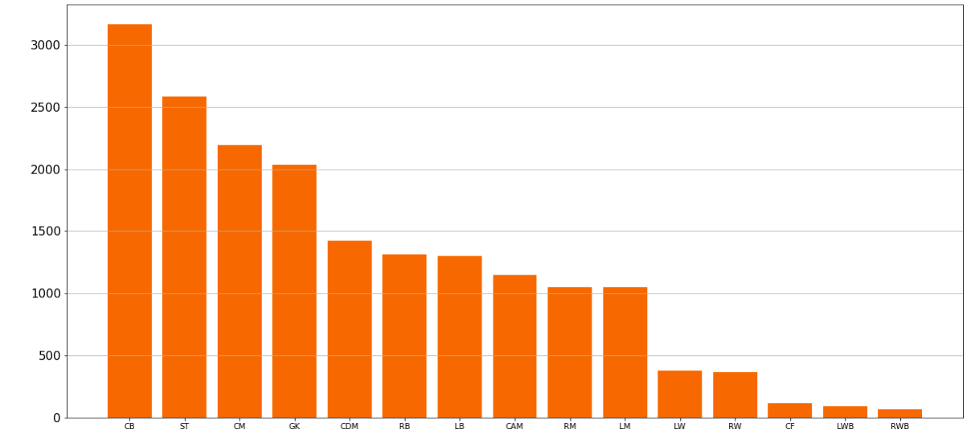


Figure is the chart of one of our targets, value, corresponding to overall score and age. The lighter the color, the older the player. And based on this chart, we can easily find there is a positive linear pattern between value and overall score.

### 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 chart above 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 Methodology

First of all, for linear regression with l1 penalty, the standardization is necessary, because the scale of the variables affect the how much regularization will be applies to specific variable. Then we use grid search to set different alpha from 0.01 to 0.1 and the step is 0.01. If the alpha is low, it means the penalty is low.

Secondly, for random forest regression, it is not necessary to do standardization. Because Random forest regression model is based on tree partition algorithms. Because there are many hyperparameter can tune in random forest regression model. In order to improve efficiency to find best hyperparameter, random grid search is a better choice. The main hyperparameter we tune is like number of estimators, max\_features, max\_depth, min\_samples\_split, min\_samples\_leaf, bootstrap.

Finally, we can compare the performance of two models and know what is the key feature of player value. Because lasso regression can make umimpotant feature’s coffefficient be zero, and random forest can get feature impotance from model.

3.2 Model

3.2.1 lasso regression

3.2.1.1 Model hyperparameter

param\_grid = {'alpha':np.arange(0.01,0.1,0.01)}

The best parameter of lasso regression: {'alpha': 0.01}

3.2.1.2 Model evaluation

The R^2 of lasso regression: 0.6014637257738532

3.2.1.3 Important feature

Table

Description automatically generated

There are more then 40 features’ coefficients become zero. We find some feature like player’s overall score, international\_reputation, movement\_raction,power\_stamina have positive relationship with player’s value, and features like age, power\_jumping have negative relationship with player’s value

3.2.1.4 Test dataset prediction

Chart, scatter chart

Description automatically generated

According to this plot, lasso regression is not a good model to predict player’s value in this case. The player with higer real value, the prediction is going to be low.

3.2.2 random forest regression

3.2.2.1 Model hyperparameter

{'n\_estimators': 100,'min\_samples\_split': 2, 'min\_samples\_leaf': 1,

'max\_features': 'auto','max\_depth': 30,'bootstrap': True}

3.2.2.2 Model evaluation

The R^2 of random forest regression: 0.9699827042483831

3.2.1.3 Important feature

Table

Description automatically generated

According to feature importance we get from random forest regression model, we find overall, age , movement\_Reaction are important features which is same as lasso regression. And what’s more, player’s pontential score, some ability like attacking finishing score, skill dribbling plays an important role in predicting palyer’s value.

3.2.2.4 Test dataset prediction

Chart, scatter chart

Description automatically generated

According to this plot, random forest regression is a good model to predict player’s value in this case. The player real value is very close to the prediction 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 decision tree, so we did not do pre-procession here. And standardization is required for distanced-based algorithm like SVM that we are using in this report. For decision tree and random forest model, we passed the data directly after train, test splitting and for SVM, a pipeline model containing standardization process is utilized to. 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 and evaluate model via one vs rest roc curve for each label. 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 Decision Tree

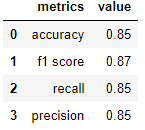
#### 4.2.1.1 Choose Best Parameters

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

|  |  |  |  |
| --- | --- | --- | --- |
| Decision Tree | criterion = 'gini', max\_depth = 10, max\_leaf\_nodes = 20,  min\_samples\_leaf = 1, min\_samples\_split = 2 | accuracy | 0.85 |
| f1-score | 0.87 |

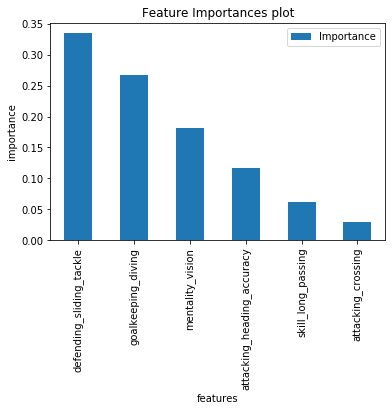
#### 4.2.1.2 Test Results

Here we calculated the macro metrics, the result of decision tree after tuning is as followed:



#### 4.2.1.4 Inference analysis

We selected features with importance greater than 0, we can see that in this model, only 6 features is used to constructed the model.



### 4.2.2 Random Forest Classification

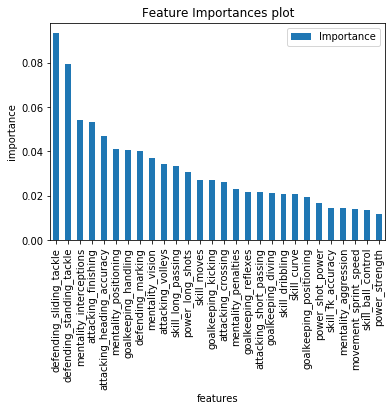
#### 4.2.2.1 Choose Best Parameters

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

|  |  |  |  |
| --- | --- | --- | --- |
| Random Forest | criterion = 'entropy',max\_depth = 25  max\_features = 5,n\_estimators = 100 | accuracy | 0.9 |
| f1-score | 0.91 |

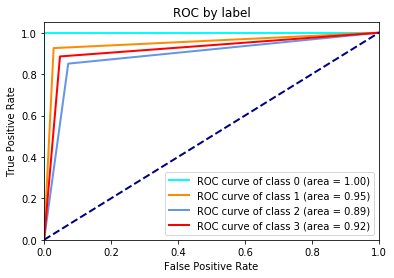
#### 4.2.2.2 Inference analysis

Comparing to decision tree model, more features are usd in Random forest. 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”.



#### 4.2.2.3 Model evaluation

Among all the three classification algorithms, random forest has the best performance. To prevent overfitting, here we evaluate the model with roc curve and auc score. For multi-class classficication we used one vs rest method to print roc curve. Classes 0 to 3 represents 'GK','Defender','Midfileder','Attacker' respectively. Auc score shows a good performance of this model.

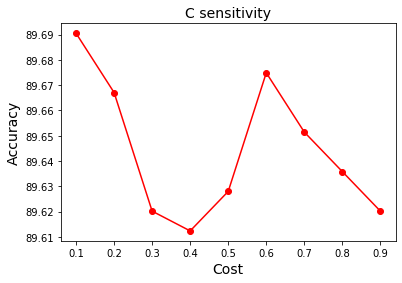


### 4.2.3 Support Vector Machine

#### 4.2.3.1 Choose Best Parameters

To choose best hyperparameters, we search for ‘C’ with best accuracy of the model mannually and plot cost sensitivity, the result is as followed:

We can see the best hyperparameter is C = 0.1.

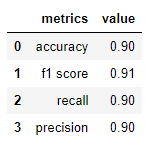


#### 4.2.3.2 Model evaluation

We use hyperparameter as listed:

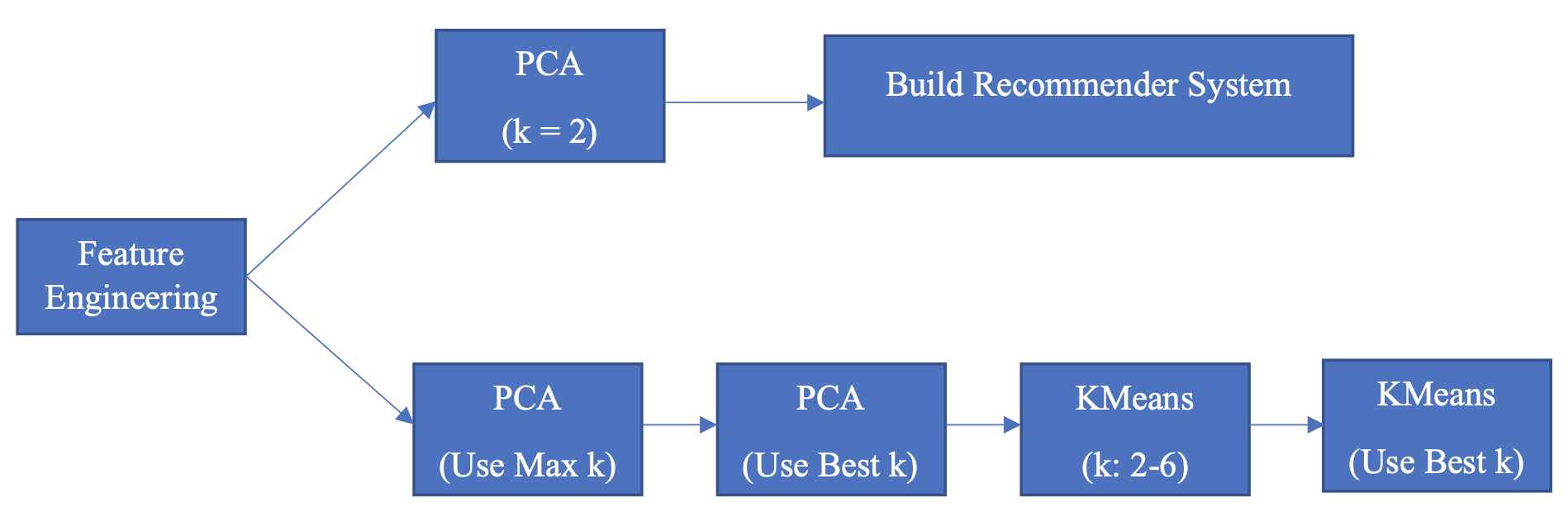
|  |  |  |  |
| --- | --- | --- | --- |
| Supporting Vector Machine | max\_iteration = 10000, cost = 0.1 | accuracy | 0.9 |
| f1-score | 0.91 |

And then evaluate the model with accuracy, f1-score, recall, precision using micro method



# **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)

A picture containing text, monitor, black, screen

Description automatically generated

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

A picture containing text, monitor, black, screenshot

Description automatically generated

#### 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 |

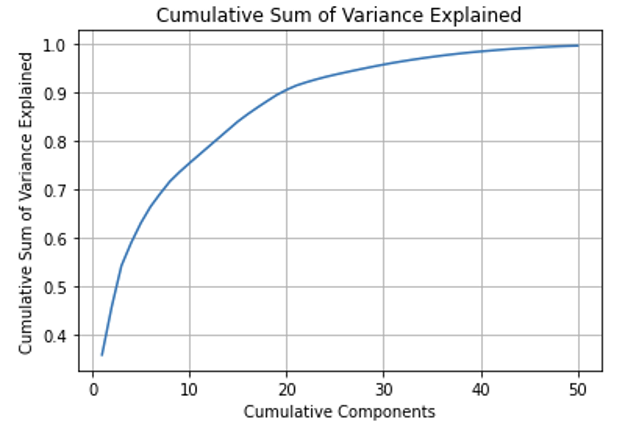
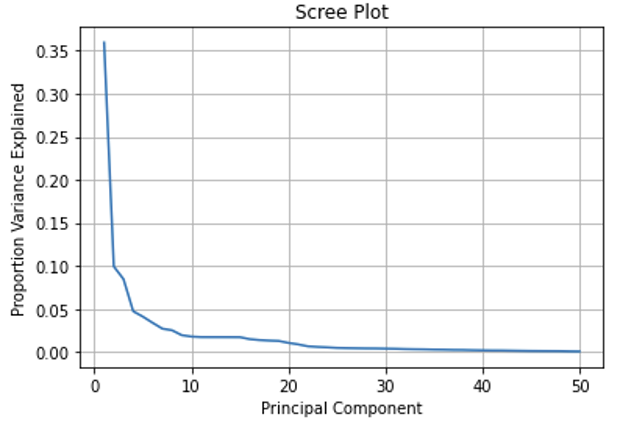
From the table above, the most feature is attacking\_volleys which is a very important ability in the real soccer match and looking through the whole table, all features are reasonable.

### 5.2.2 Clustering

#### 5.2.2.1 Model type

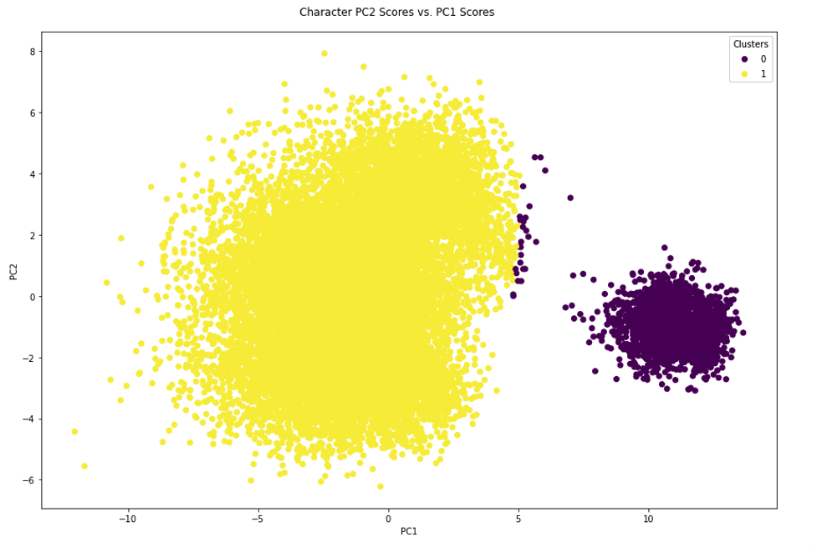
K-Means Clustering

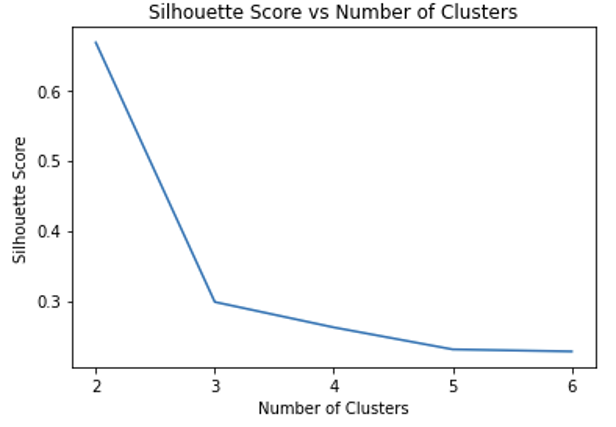
#### 5.2.2.2 Data Transformation

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 two regression models. The best one is Random Forest regression which mainly takes use of features like ‘overall, ‘potential to generate models. Random forest outperformed other algorithms with a r square 0.97

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.