**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**
**SCHOL OF INFORMATION TECHNOLOGY AND COMMUNICATION**

-----


# Introduction to Data Science
# CAPSTONE PROJECT REPORT

## Project name: FIFA 21 Clustering

Assistant Lecturer: PhD. Muriel Visani
Group number: 14
Student names:
Nguyen Nho Trung - 20204894
Tong Thi Thu Anh – 20194728
Nguyen Thi Thu Giang - 20194750

***Ha Noi, 01/2022***

# 1. Introduction

Our subject for the Capstone is applying Machine Learning to cluster football players. In detail, our project aims to look at the class of young players and cluster them into groups based on their attributes. We will filter and consider the players who are promised based on their potential attributes.

The FIFA dataset is not just totally the characters in a video game but it's based on the football players in the real world, so we can apply it to some extent practical work. For example, the football specialists like managers, scouts, or coaches can refer to this analysis as the first step to find and target new young players appropriate to their teams or match their expectations. Also, they can discover the similarities and differences between the clusters to build different lines up and static. Of course, this analysis will be helpful for the FIFA gamers too.

# 2. Dataset

The dataset used in this project is the FIFA data taken from Kaggle:
https://www.kaggle.com/stefanoleone992/fifa-21-complete-player-dataset?select=players_21.csv

The dataset contains 18,944 players with 106 different attributes but we only take into account the players under 20 years, their potential more than 70 (out of 100), we not consider the footballer play at goalkeeper position, some attributes related to goalkeeper skills and some attributes related to player's personal data

Therefore, the dataset include:

- Name: Name of the player.
- Age: Age of the player.
- Height: Height of the player in inches.
- Overall: The score of general performance quality and value of a player, rated between 1-99
- Potential: Maximum expected overall score in the top of his career, rated between 1-99
- Work Rate: Degree of the effort the player puts in terms of attack and defense rated as low, medium and high.
- Team Position: main position/role of a player in the team

- Positional Skills: Player's general ability when playing in a specified position rated between 1-99.
  - List of positional skill attributes: LS, ST, RS, LW, LF, CF, RF, RW, LAM, CAM, RAM, LM, LCM, CM, RCM, RM, LWB, LDM, CDM, RDM, RWB, LB, LCB, CB, RCB, RB.
  - Explanation:
    - S: Striker
    - F: Forward
    - M: Midfielder
    - B: Back
    - W: Wing
    - A: Attacking
    - D: Defensive
    - L: Left
    - R: Right
    - C: Center

- Playing Skills:
  - List of playing skills: Attacking Crossing, Attacking Finishing, Attacking Heading Accuracy, Attacking Short Passing, Attacking Volleys, Skill Dribbling, Skill Curve, Skill Free Kick Accuracy, Skill Long Passing, Skill Ball Control, Movement Acceleration, Movement Sprint Speed, Movement Agility, Movement Reactions, Movement Balance, Power Shot Power, Power Jumping, Power Stamina, Power Strength, Power Long Shots, Defending Standing Tackle, Defending Sliding Tackle.
  - Value range: Rated between 1-99.

# 3. Data preprocessing and EDA

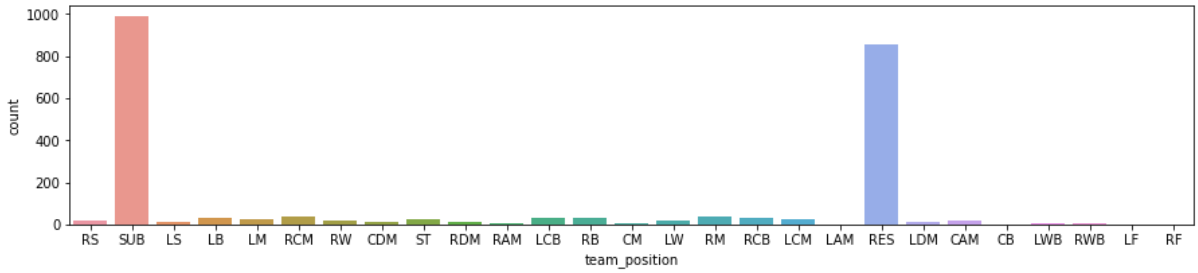
Data preprocessing and EDA help us to prepare the structure of the data for modeling, determine the relationships between the features and choose which features are useful for clustering the players.

## 3.1. Data preprocessing

- Step 1: Check Nan values
- Step 2: Drop unrelated and nan-value columns
- Step 3: Filter players
  - Remove Goalkeepers
  - Keep players whose age is less than or equal to 20 and potential rating is larger than or equal to 70
- Step 4: Transform data, change categorical to numerical data
  - Extract position ratings
  - Extract work_rate into attack_work_rate and defense_work_rate, change rating label to numeric (High - 1, Medium - 0.5, Low - 0)

## 3.2. Exploring Data Analysis

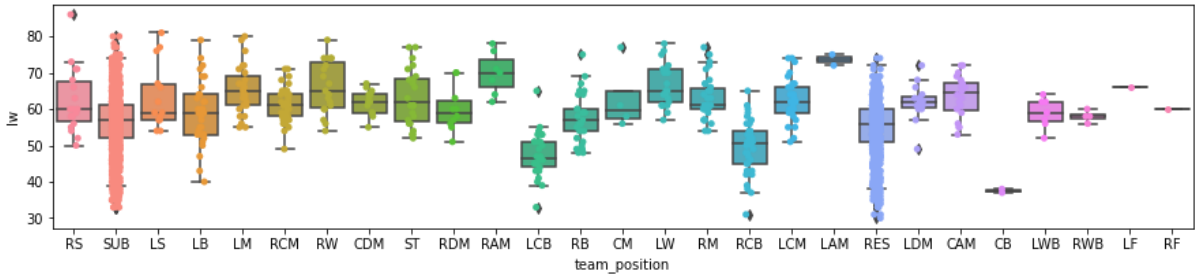
### 3.2.1. Plot the number of players in each position




Comment:

- The plot shows that there are many players whose team positions are SUB or RES, which means they are currently substitute players (not included in the starting lineup) in their team. The major reason is because of their young age and poor experience.
- This is good news for the football specialists who are seeking new and naive players for training.

### 3.2.2. Plot the relationship between team positions and positional skills.

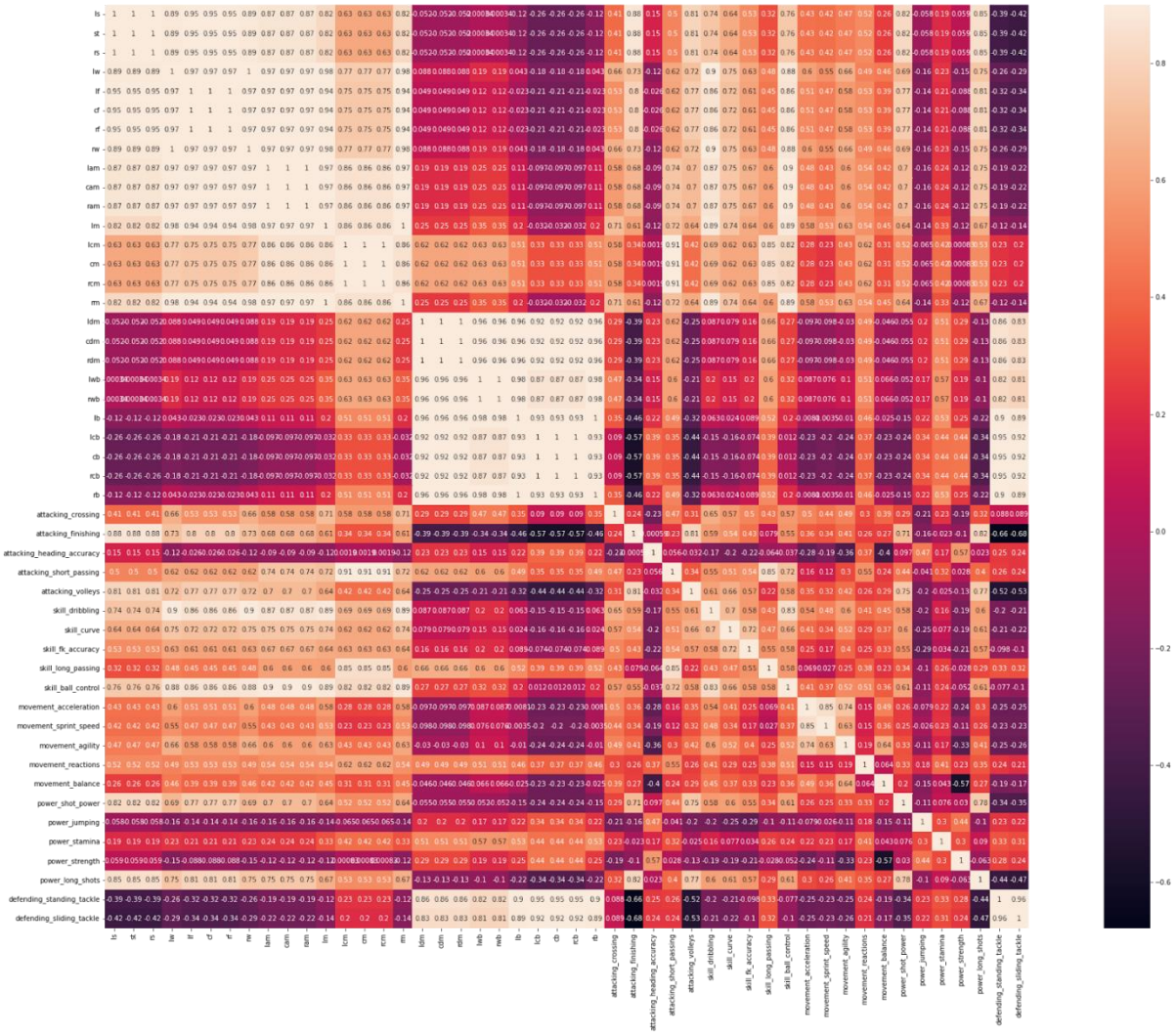



Comment:

- The median LW skill of players who play in RAM position is higher than the median of LW position players. The median LW skill of RW players is also as good as LW players.
- In addition, many substitute players (SUB and RES team_position players) have high ratings for many positions.
- The conclusion here is that the playing position does not matter too much, but the skill ratings are more important. A player in position CAM may be good at position LW also, so we do not use positions for clustering because it would lack information.

### 3.2.3. Correlation between the position and playing skills features

We plot a heat map to see the spearman correlation between the position and playing skills features



Comment:

- The heat map allows us to determine groups of position skills and playing skills that are highly correlated to each other, for example, group of striker skills (ST, RS, LS), group of defensive skills.
- This information is helpful for our dimension reduction later.

## 3.2.3. Discover the relationship between work_rate and other skills

We take the comparison of attack work rate attributes with other attacking skills and the comparison of defense work rate attributes with other defending skills

| attack_work_rate | attacking_crossing | attacking_finishing | attacking_short_passing | attacking_volleys |
|---|---|---|---|---|
| **0.0** | 37 | 26 | 56 | 29 |
| **0.5** | 49 | 48 | 59 | 42 |
| **1.0** | 55 | 56 | 60 | 47 |

| defense_work_rate | defending_standing_tackle | defending_sliding_tackle |
|---|---|---|
| 0.0 | 28 | 26 |
| 0.5 | 52 | 49 |
| 1.0 | 62 | 59 |

Comment:

- The player who put more effort in attacking has higher attacking skills rating
- Similar to defense work rate and defending skills

# 4. Modeling Data

- Step 1: Drop unused attributes for clustering
  - The result obtained from Exploratory Data Analysis is used to determine which features will be inputs to the model. The distinctive features which can be useful in clustering are selected as all features explained above except Name, Overall, and Position.
- Step 2: Reduce Dimension
  - We have to deal with a large number of data dimensions, so the number of dimensions can be reduced by applying Principal Component Analysis (PCA) especially to highly correlated features. PCA method is obtaining components that explain the highest variance or most of the information in the data with a smaller number of attributes.
  - We group the attributes which have a high correlation with each other based on the correlation heat map plotted in the EDA process and then use PCA for each group.
- Step 3: Standardized data.
  - Data standardization helps us scale data of different ranges to the same range value, which is better for our clustering.
  - Since this dataset contains outliers in some of its features and standardization can be influenced by outliers, so we robust scaler is a good option in this case. This method will remove the median and scale the data in the range between 1st quartile and 3rd quartile

We use Euclidean distance for all algorithms that we have chosen for clustering

### Principal Component Analysis (PCA)

Base on the heat map, we put position skills into following group:

- Striker group: LS, ST, RS
- Forward group: LF, CF, RF
- Attack Winger group: RW, LW
- Midfielder group: LAM, CAM, RAM, LM, LCM, CM, RCM, RM, LDM, CDM, RDM
- Defensive Winger group: LWB, RWB
- Back group: LB, LCB, CB, RCB, RB

```
Explained variance ratio for Striker PCA application:  [1.]
Explained variance ratio for Forward PCA application:  [1.]
Explained variance ratio for Attack Winger PCA application:  [1.]
Explained variance ratio for Midfielder PCA application:  [0.70388172]
Explained variance ratio for Defensive Winger PCA application:  [1.]
Explained variance ratio for Back PCA application:  [1.]
```

Base on the heat map, we put playing skill into following group:

- Defensive skills: Defending Standing Tackle, Defending Sliding Tackle
- Movement: Movement Acceleration, Movement Sprint Speed, Movement Agility, Movement Reactions
- Control skills: Skill Dribbling, Skill Ball Control
- Passing skills Attacking Short Passing, Skill Long Passing
- Finishing skills: Attacking Finishing, Attacking Volleys

```
Explained variance ratio for defensive skills PCA application:  [0.98299723]
Explained variance ratio for movement PCA application:  [0.82825812]
Explained variance ratio for control skills PCA application:  [0.91608148]
Explained variance ratio for passing skills PCA application:  [0.92329027]
Explained variance ratio for defending skills PCA application:  [0.91549397]
```

## 4.1. Kmeans:

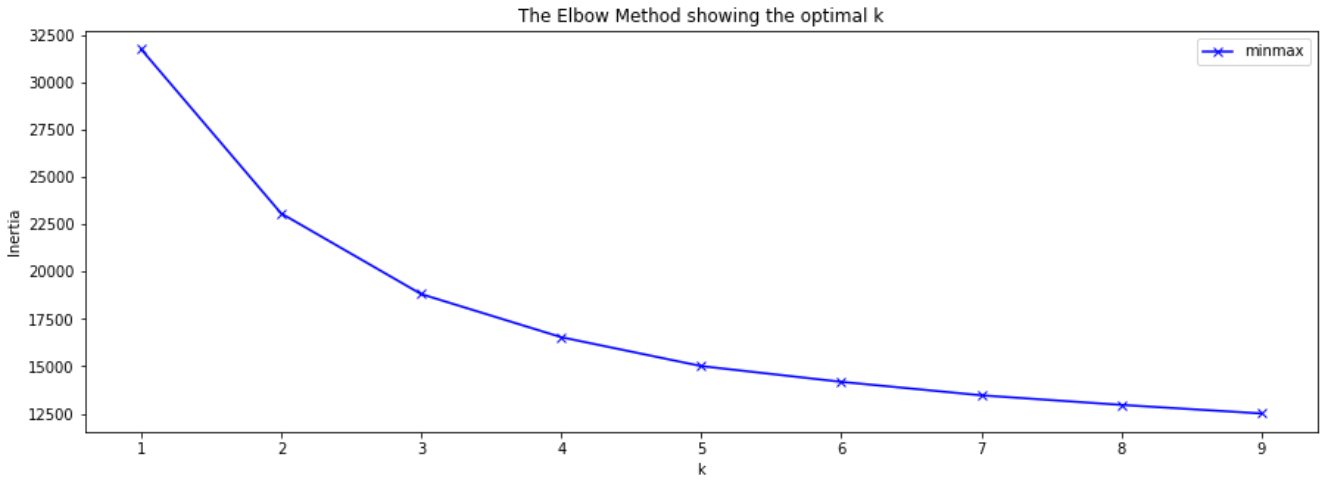
K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

K-means is still the most popular algorithm used to solve clustering problems due to its simplicity and efficiency.

### 4.1.1. Elbow method

We used K-Means clustering to cluster the dataset. To determine the number of K, we use the Elbow method. The method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use.

Result of applying Elbow method to the dataset:




The optimal K when using Elbow method is 2, 3 or 4

### 4.1.2. Silhouette method

We use another method to evaluate choosing k clusters which is the Silhouette method.

Intuition of Silhouette method:

- Silhouette refers to a method of interpretation and validation of consistency within clusters of data.
- The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The value of the silhouette ranges between [-1, 1], where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

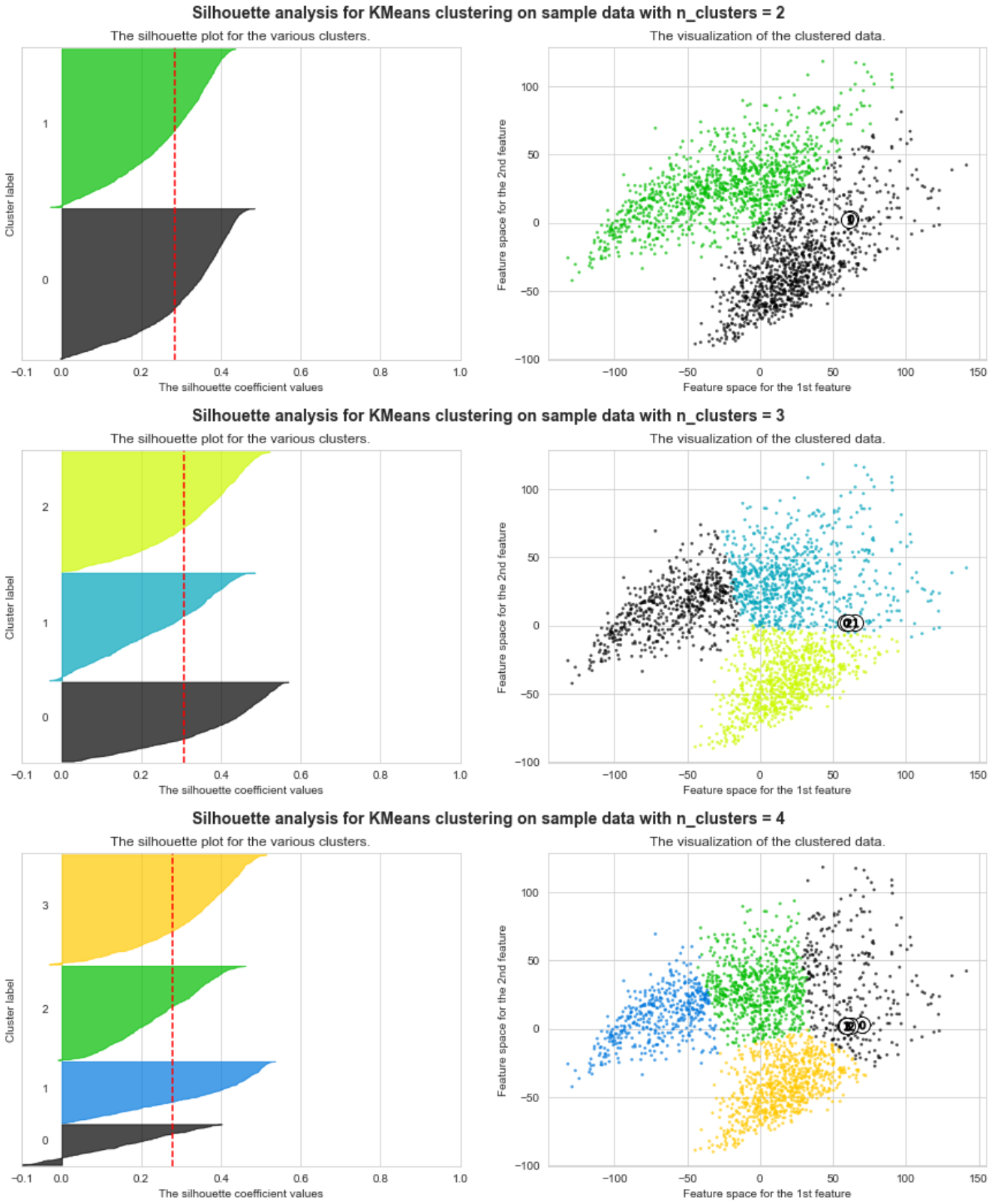
in Worst case s(i) = -1

$$s(i) = \frac{b(i) - a(i)}{\text{larger of } b(i) \text{ and } a(i)}$$

a(i) = average distance inside cluster
b(i) = average distance nearest other cluster

Results of silhouette scores with different k:

```
For n_clusters = 2 The average silhouette_score is : 0.28272602345704606
For n_clusters = 3 The average silhouette_score is : 0.3084938723509853
For n_clusters = 4 The average silhouette_score is : 0.2795438505249029
For n_clusters = 5 The average silhouette_score is : 0.22999870816740864
For n_clusters = 6 The average silhouette_score is : 0.21964100225379676
For n_clusters = 7 The average silhouette_score is : 0.20744387774940753
For n_clusters = 8 The average silhouette_score is : 0.19222045902464063
For n_clusters = 9 The average silhouette_score is : 0.1860086216190746
```




K = 4 has the maximum Silhouette score, combined with the Elbow method, we decide K=4 is the optimal number of clusters for K-means clustering.

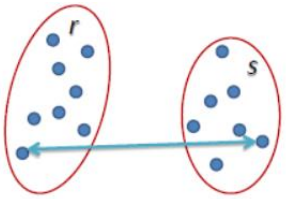
## 4.2. Hierarchical Clustering

We use Hierarchical agglomerative clustering.
Hierarchical Agglomerative Clustering (HAC) will build the dendrogram from the bottom up.
The algorithms:

- At the beginning, each instance forms a cluster (also called a node in dendrogram)
- Merge the most similar (nearest) pair of clusters :i.e., The pair of clusters that have the least distance among all the possible pairs
- Continue the merging process
- Stop when all the instances are merged into a single cluster (i.e., the root cluster in dendrogram)

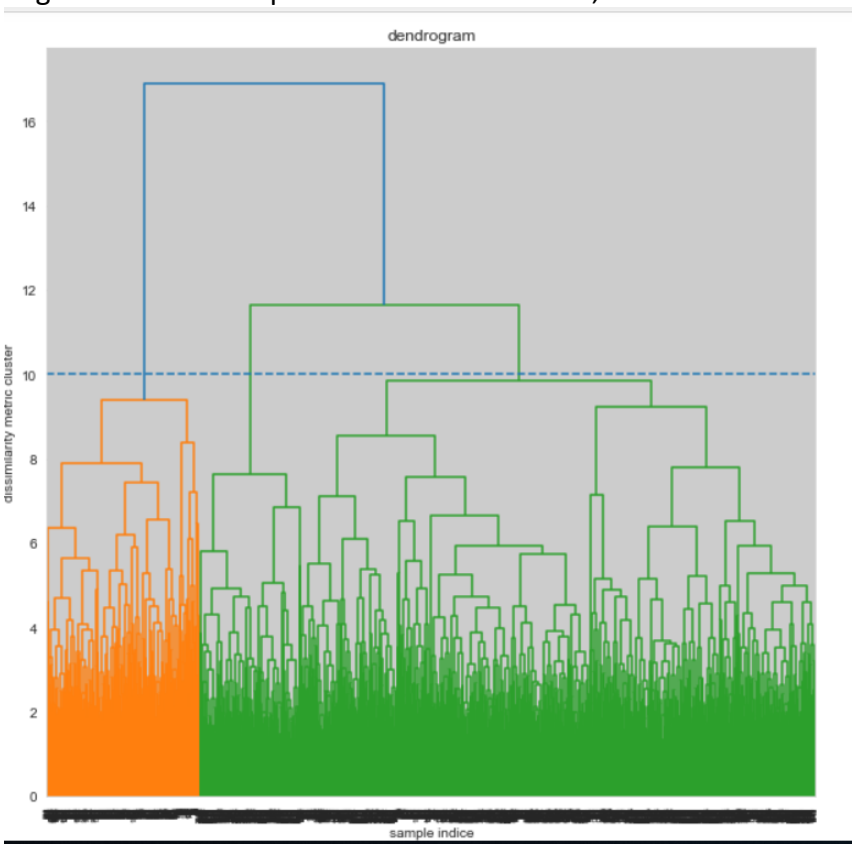
We use **Complete Linkage:**

- In complete linkage hierarchical clustering, the distance between two clusters is defined as the longest distance between two points in each cluster. For example, the distance between clusters “r” and “s” to the left is equal to the length of the arrow between their two furthest points.




$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

We using the dendrogram to find the optimal number of cluster, we obtain:



According to the rules, according to territory, I must choose the longest vertical line and make intersections and another vertical line , so we have 3 clusters . So, we obtain K=3 in this algorithm

## 4.3. Birch algorithm

BIRCH (balanced iterative reducing and clustering using hierarchies) is an unsupervised data mining algorithm used to perform hierarchical clustering over particularly large data-sets.

An advantage of BIRCH is its ability to incrementally and dynamically cluster incoming, multi-dimensional metric data points in an attempt to produce the best quality clustering for a given set of resources (memory and time constraints). In most cases, BIRCH only requires a single scan of the database.

Birch is a Hierarchical clustering algorithm using top-down

- Idea: Each successful iteration, a cluster is split into smaller clusters according to the value of some similarity measure until each object is a cluster or until the stopping condition is satisfied.
- This approach into using divide and conquer strategy in the clustering process

## 4.4. Comparision 3 algorithms:

- **Optimal**
  - We have table about average silhouette_score:

| | K cluster | kmean_silhouette_scores | birch_silhouette_scores | agg_silhouette_scores |
|---|---|---|---|---|
| **0** | 2 | 0.282726 | 0.238964 | 0.238964 |
| **1** | 3 | 0.308363 | 0.236685 | 0.236685 |
| **2** | 4 | 0.279544 | 0.212797 | 0.212797 |
| **3** | 5 | 0.229999 | 0.193872 | 0.193872 |
| **4** | 6 | 0.219155 | 0.185722 | 0.185722 |
| **5** | 7 | 0.206945 | 0.154607 | 0.154607 |
| **6** | 8 | 0.191965 | 0.155710 | 0.155710 |
| **7** | 9 | 0.185837 | 0.155961 | 0.155961 |

  - We see that, for K=3, then K-means has the highest silhouette_scores, so K-Means is the best in terms of efficiency and the most optimal
- **Running time**
  - We see that the data after processing is not large, so K_means and Birch run very fast in practice, we have:

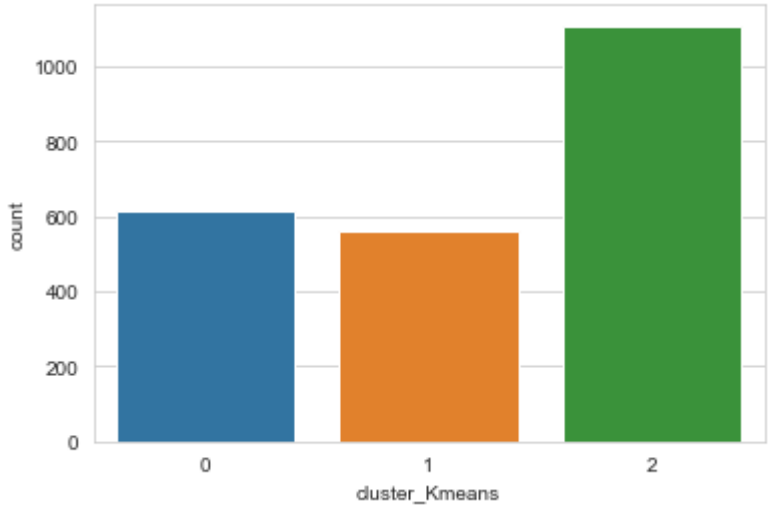
    - K-Means: 12.636946439743042 s
    - Birch: 1.4609653949737549 s
- With hierarchical clustering, in worst case, the running time is O(n^2*log(n) with n : size of input.

Therefore, in this problem, the data size is not small, so the running time is quite long in practice, the running time is about 2 minutes.

**With all the above reasons, we decided to choose K-means to analyze the results with K=3**

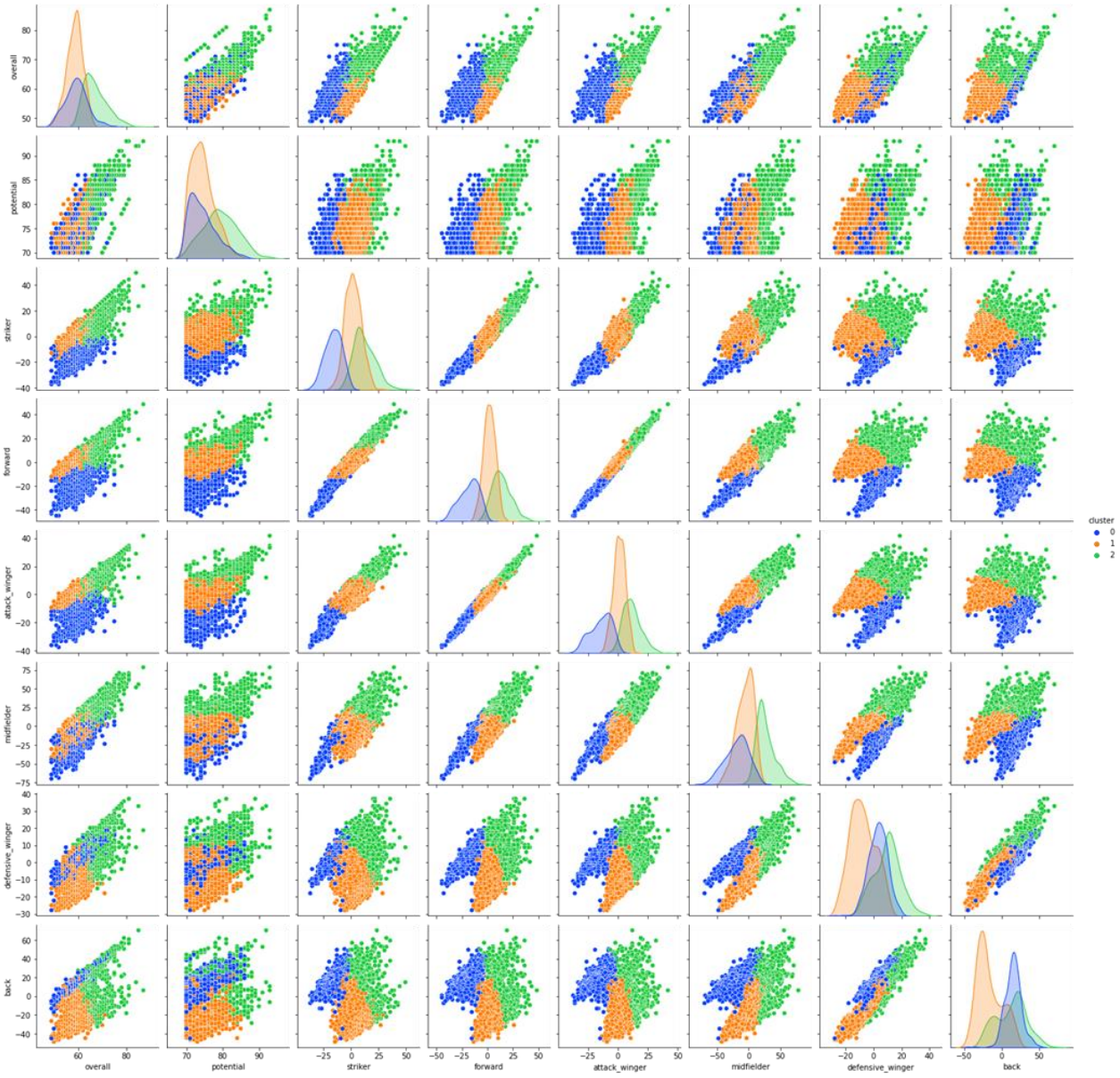
# 5. Result analysis: K-means with K=3

After clustering, numbers of players in each cluster are shown as following:




```
2    1109
0     615
1     562
Name: cluster_Kmeans, dtype: int64
running time of Kmeans  70.96637034416199
```

Pair plots of overall, potential and position skills for each cluster:




Number of players in cluster 2 is the largest, and Cluster 2 also has potential and overall rate higher than the others. Cluster 0 and Cluster 1 show similar potential and overall.

| cluster_Kmeans | potential | overall |
|---|---|---|
| **2** | 79.738211 | 67.549593 |
| **1** | 74.443060 | 59.176157 |
| **0** | 74.352570 | 58.725879 |

While Cluster 2 shows the good rate at any position skills, Cluster 0 has the highest rate when they play at back position. Position skills for Strike, Forward, Attack Winger and Midfielder of cluster 1 are higher than Cluster 0 and still lower than Cluster 2, but for playing at Defensive Winger and Back position Cluster 2 is the worst.

| cluster_Kmeans | striker | forward | attack_winger | midfielder | defensive_winger | back |
|---|---|---|---|---|---|---|
| 0 | -16.899717 | -19.061039 | -15.009631 | -20.386868 | 2.683868 | 15.038853 |
| 1 | 1.252315 | 1.377387 | 0.804718 | -4.563811 | -6.374995 | -13.515913 |
| 2 | 13.185078 | 14.934604 | 12.265010 | 26.859651 | 9.043148 | 10.629777 |

For playing skills, Cluster 0 shows the best ability at Defensive, Movement and Control while Cluster 2 is excellent at Passing and Finishing. Cluster 1 has average rate at Movement, Control, Passing and Finish but shows not very good skill for Defensive.

| cluster_Kmeans | defensive | movement | control | passing | finishing |
|---|---|---|---|---|---|
| 0 | 21.025102 | 12.107377 | 16.329092 | -9.784023 | -23.755704 |
| 1 | -13.352598 | -0.435819 | -0.791915 | -2.035538 | 5.300393 |
| 2 | 4.864916 | -10.278085 | -13.493846 | 12.611436 | 12.150520 |

Attack Work Rate and Defense Work Rate are high for Cluster 2, especially at Attack Work Rate. Cluster 0 shows higher Defense Work Rate than its Attack Work Rate. Cluster 1 is better at Attack Work Rate while its Defense Work Rate is the worst in all

| cluster_Kmeans | attack_work_rate | defense_work_rate | defensive |
|---|---|---|---|
| 0 | 0.494662 | 0.588968 | 21.025102 |
| 1 | 0.638864 | 0.458070 | -13.352598 |
| 2 | 0.704878 | 0.518699 | 4.864916 |

**General conclusions**

Therefore, Cluster 1 is a collection of players who play well in the attacking positions, they have a very good offensive rate and also a pretty good defensive rate. Their good Passing, Attacking skills and Power shot are the most crucial skills for playing at Attack positions. They have the highest potential overall compared to the rest. They should be included in the team as Strike, Forward, or Midfielder.

Not as high potential and overall as Cluster 1 but Cluster 2 shows good skill when they're playing at Defensive positions, they have the highest rate for defensive, movement and control which are the most important skills for playing as Defensive. They are the good choice for the team lacking in Back positions.

The players in cluster 1 are better at Attacking than Defensive although their skill rates related to Attacking are not as high as cluster 2, they have pretty bad defensive rate, their potential and overall are not really good. In general, they have shown the worst of all.

## Detail contribution of each member

Programming:

- Preprocessing Data and EDA:
    - Trung 20%
    - Thu Anh 40%
    - Giang 40%

- Data Modeling and Algorithms:
    - Trung 35%
    - Thu Anh 30%
    - Giang 35%
- Result presentation :
    - Thu Anh: 50%
    - Giang: 50%
- Modify code Trung: 100%

Analytic:

- Preprocessing Data and EDA: Giang 100%
- Data Modeling and Algorithms: Trung 100%
- Result presentation : Thu Anh 100%

## References:


1. Scikit learn. Principal component analysis (PCA) https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

2. Scikit learn. Clustering https://scikit-learn.org/stable/modules/clustering.html

3. Thuật toán Brich , Khai phá dữ liệu https://www.slideshare.net/hop23typhu/thut-ton-brich

4. Pham Dinh Khanh. Hierarchical Clustering https://phamdinhkhanh.github.io/deepai-book/ch_ml/index_HierarchicalClustering.html