Projet : analyse des joueurs de Fifa 2022

Auteur : Simon Decomble

On écrit un module Python fifa avec un sous-module fifa_utils contenant des fonctions utiles au nettoyage des données ou traitement particuliers. Cela permet d'alléger le notebook, de séparer le code métier et potentiellement d'effectuer des tests.

Le dataset brut contient 19 239 lignes et 110 colonnes.

1. Analyse descriptive des données

1.2 Préparation des données

Suppression initiale de colonnes

On supprime les colonnes qui ne nous serviront nécessairement pas à l'analyse, ex : images, liens, redondances (noms inutiles puisqu'on a un id correspondant), positions en équipe nationale (puisqu'on ne garde que la première position).

Position des joueurs

On ne garde que la première position du joueur. On réalise donc ce traitement sur l'ensemble des lignes.

On change la granularité des positions de joueurs, pour ne garder que 4 valeurs "GK" pour goal, "DEF" pour les défenseurs, "MID" pour les milieux, "FWD" pour les attaquants.

Age des joueurs

On créé des groupes d'âge : -20, 20-25, 25-30, 30-35, 35+, ce qui est plus pertinent pour traiter un attribut d'âge.

Stats des joueurs

On détecte les colonnes susceptibles de contenir des "+/- <nombre>" à enlever, c'est à dire les colonnes du type object contenant des + ou -.

On retire la colonne "age" de cet ensemble car c'est notre traitement précédent qui a mis un "-" dans les noms de classes de cet attribut.

On supprime de ces colonnes détectées les valeurs après le +/-, puis on convertit les valeurs en entier.

IMC des joueurs

Les masses des joueurs sont toutes formatées en kg, les tailles en cm, on convertit les tailles en m pour le calcul de l'IMC (indice de masse corporelle).

On créé un attribut BMI, body mass index, qui correspond à l'IMC des joueurs : $$\frac{poids}{taille^2}$$

poids en kg
taille en m^2

On supprime les colonnes de masse et taille puisque l'IMC porte ces informations et on cherche à éviter les features corrélées.

Valeur des joueurs

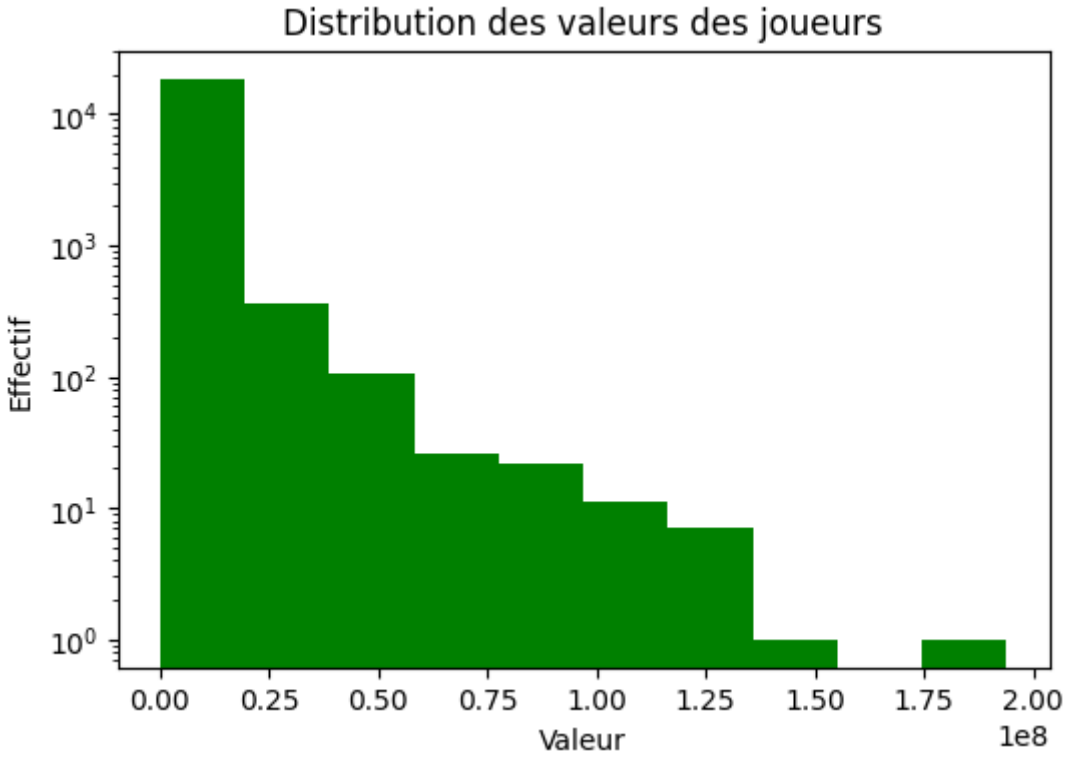
On va discrétiser les valeurs en classes pertinentes, pour déterminer ces classes, on regarde les statistiques de ces colonnes et la distribution des données sous forme d'histogramme.

```
Nombre de valeurs manquantes : 74
```

On a quelques joueurs qui n'ont pas de valeur renseignée, pour éviter de les supprimer de la base de donnée sur un seul critère, on les placera dans une catégorie à part "unknown". On réalise des stats sur les autres.

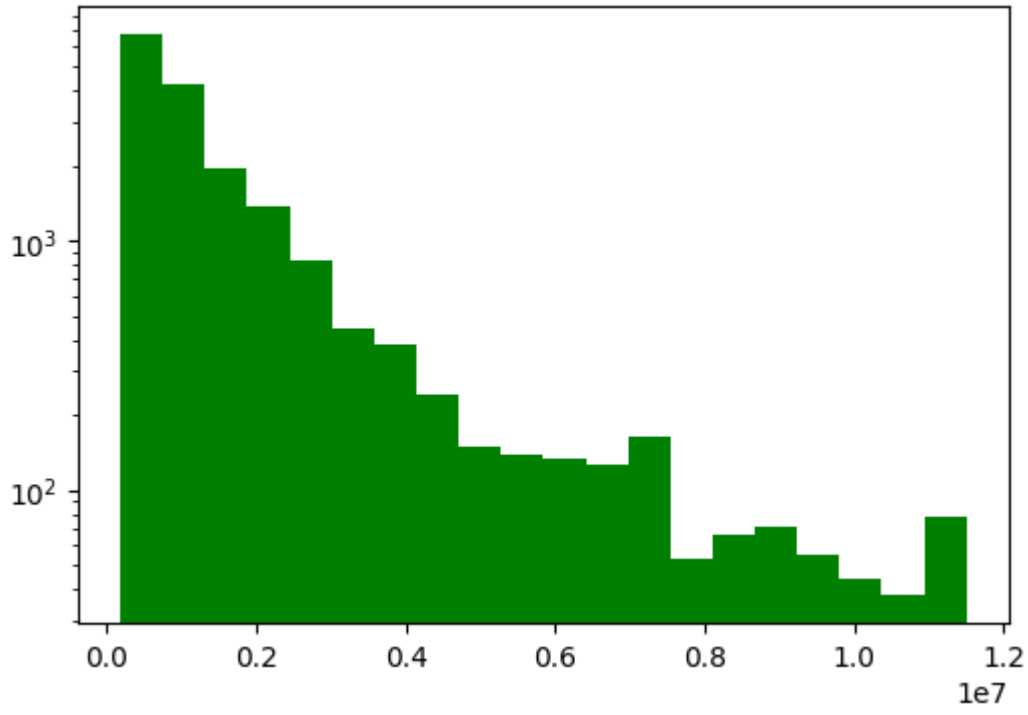
```
count    1.916500e+04
mean     2.850452e+06
std      7.613700e+06
min      9.000000e+03
25%      4.750000e+05
50%      9.750000e+05
75%      2.000000e+06
max      1.940000e+08
Name: value_eur, dtype: float64
```

On remarque que les valeurs sont comprises entre 9 000 et 194M, avec une médiane à 97 500€. Le 1er quartile à 47 500 et le 3e quartile à 2M laisse donc penser qu'il y a des outliers aux deux extrémités du spectres, on ne peut donc pas séparer linéairement en classes. On regarde la distribution des données. Il est nécessaire d'appliquer une échelle logarithmique à cause des valeurs extrêmes supérieures.



Sur l'histogramme, les outliers polluent la visualisation, on va garder uniquement les 90% au centre du dataset selon les valeurs (entre 5e centile et 95 centile).

```
5e centile : 180000.0
95e centile : 11500000.0
```



On calcule l'écart-type empirique et on forme 3 catégories dans l'histogramme : valeurs en dessous d'une fois σ (l'écart-type), valeurs entre σ et $2.\sigma$ et valeurs supérieure à $2.\sigma$.

```
low         13829
moderate     2759
high         2577
unknown        74
Name: Dvalue, dtype: int64
```

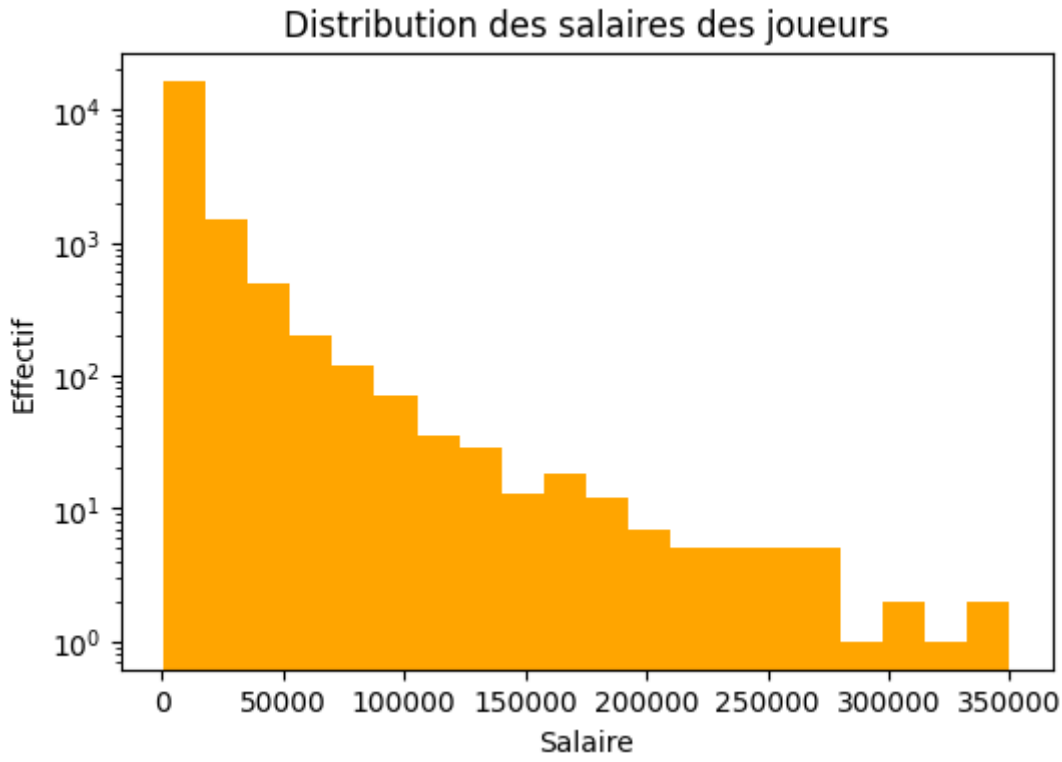
Salaire des joueurs

On applique la même démarche que pour les valeurs.

```
Nombre de valeurs manquantes : 61
```

Statistiques sur les salaires :

```
count     19178.000000
mean       9017.989363
std       19470.176724
min         500.000000
25%        1000.000000
50%        3000.000000
75%        8000.000000
max      350000.000000
Name: wage_eur, dtype: float64
```



On calcule l'écart-type empirique et on forme 3 catégories dans l'histogramme : valeurs en dessous d'une fois σ (l'écart-type), valeurs entre σ et $2.\sigma$ et valeurs supérieure à $2.\sigma$.

```
low         16922
moderate     1330
high          926
unknown        61
Name: Dwage, dtype: int64
```

On supprime les colonnes avec les valeurs et salaires, dont on a plus besoin (l'information -ou son absence- a été discrétisée sous forme de 4 catégories).

```
Le dataset préparé est désormais de taille (19239, 84).
```

1.2 Analyse

Attributs et statistiques descriptives

Description des attributs

Tous les attributs gardés jusqu'ici (pertinence et information compactée) son listés ci-dessous.

```
sofifa_id  overall  potential  age  club_team_id  league_level
nationality_id  nation_team_id  preferred_foot  weak_foot  skill_moves
international_reputation  pace  shooting  passing  dribbling  defending
physic  attacking_crossing  attacking_finishing  attacking_heading_accuracy
attacking_short_passing  attacking_volleys  skill_dribbling  skill_curve
skill_fk_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  mentality_aggression
mentality_interceptions  mentality_positioning  mentality_vision
mentality_penalties  mentality_composure  defending_marking_awareness
defending_standing_tackle  defending_sliding_tackle  goalkeeping_diving
goalkeeping_handling  goalkeeping_kicking  goalkeeping_positioning
goalkeeping_reflexes  goalkeeping_speed  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  gk  player_position  BMI  Dvalue  Dwage
```

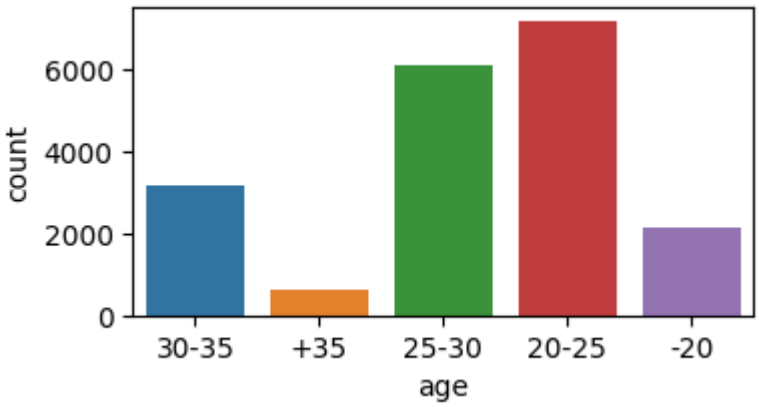
Sofifia_id est la clé primaire d'un joueur, c'est le numéro qui identifie de manière unique chaque joueur.
Age est la catégorie d'âge du joueur.
Club_team_id, nation_team_id et nationality_id sont les identifiants de son équipe de championnat, de son équipe nationale et de sa nationalité.
League level est le niveau de championnat dans lequel son club évolue.
Player_position indique son poste (goal, défenseur, milieu, attaquant).

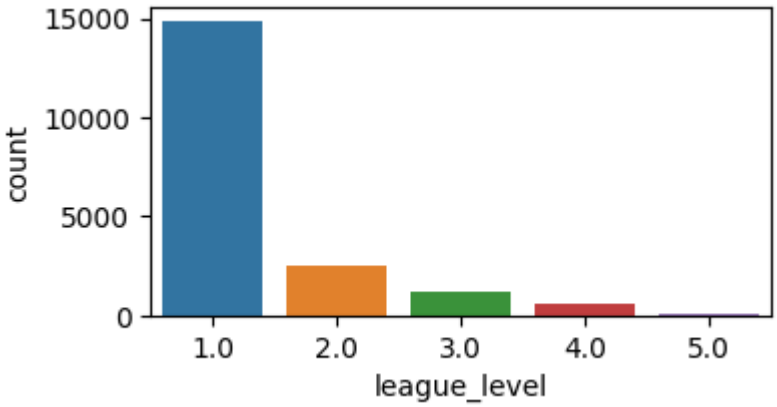
BMI est son indice de masse corporelle.
Dvalue indique la catégorie de valeur financière à laquelle il appartient (inconnue, basse, moyenne, haute).
Dwage indique la catégorie de salaire qu'il reçoit (inconnu, bas, moyen, haut).
Preferred_foot indique s'il est droitier ou gaucher du pied.
Tous les autres attributs sont des statistiques numériques de jeu, liées aux déplacements, tirs, dribles etc.
Overall et potential sont des statistiques globales d'évaluation du joueur.

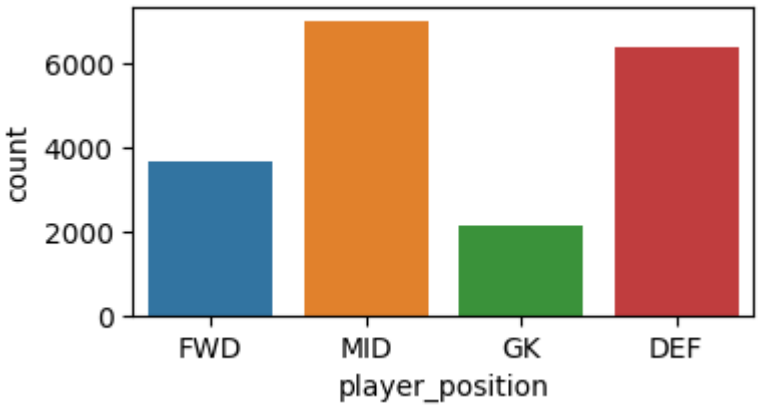
Statistiques des attributs catégoriels

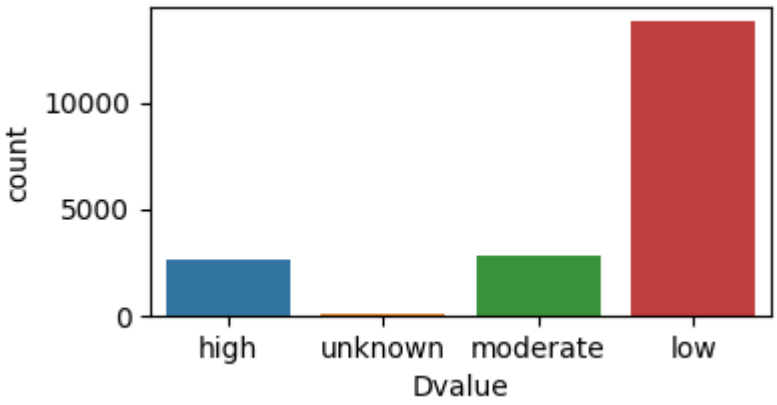
On extrait les features catégoriels, qui n'ont pas trop de valeurs (ce qui exclut les id d'équipe ou de nationalité).

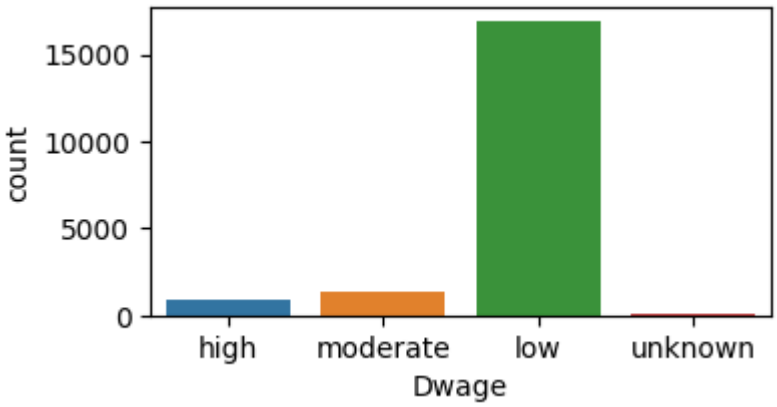
On peut afficher la répartition des différentes catégories pour ces attributs catégoriels. On remarque ainsi quels sont les modes, classes les plus représentées. On se rend aussi compte du déséquilibre de certaines classes.

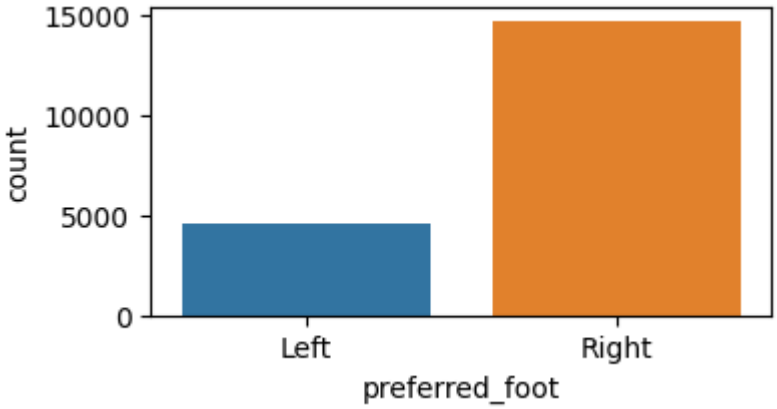


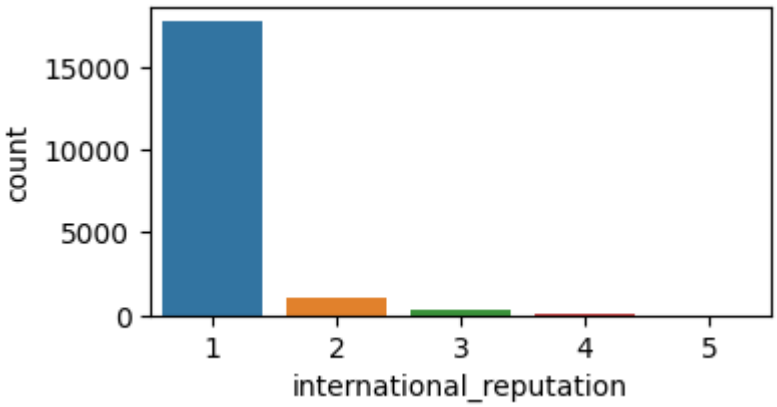












Ces visualisations nous permettent de savoir rapidement qu'il y a par exemple une grande majorité de droitiers, de joueurs de 20 à 30 ans. Que les quantités reflétant un avantage (valeur financière, salaire,

réputation, niveau de ligue) sont élevées pour une minorité de joueurs et élevées pour très peu.
Ainsi que la répartition entre les postes.

Statistiques des attributs numériques

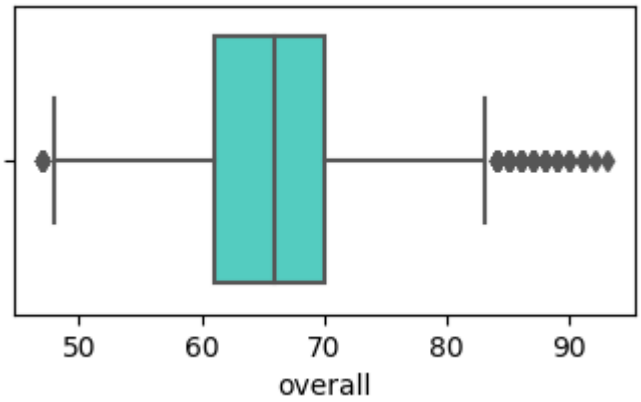
On observe les statistiques (plus détaillées) sur les attributs numériques d'intéret. On retire en effet les identifiants de joueur ou d'équipes, qui ne sont pas propices à la réalisation de statistiques (leurs valeurs étant des clés et non des quantités).

```
Nous avons, en tout, 73 attributs numériques / statistiques de joueurs



Nombre de goals : 2132
Nombre de valeurs de goalkeeping_speed renseignées dans le dataset : 2132
```

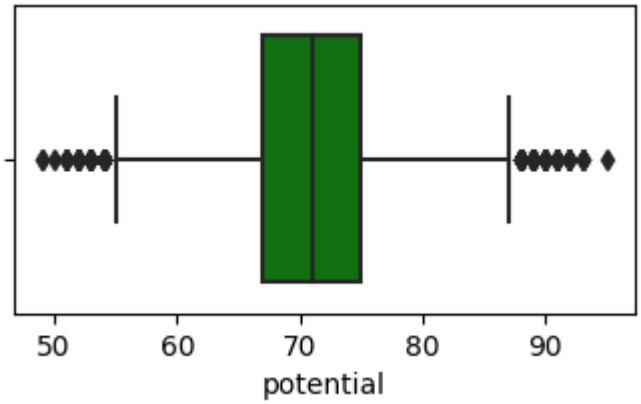
Ces statistiques sont nombreuses, nous allons creuser davantage :

- les statistiques générales : overall et potential
- une autre statistique propre à tous les joueurs : power_stamina
- une statistique propre aux joueurs non goal : physic
- une statistique propre aux goals : goalkeeping_speed
- une statistique commune à tous les joueurs, calculée lors de la phase de préparation à partir de la masse et de la taille : BMI

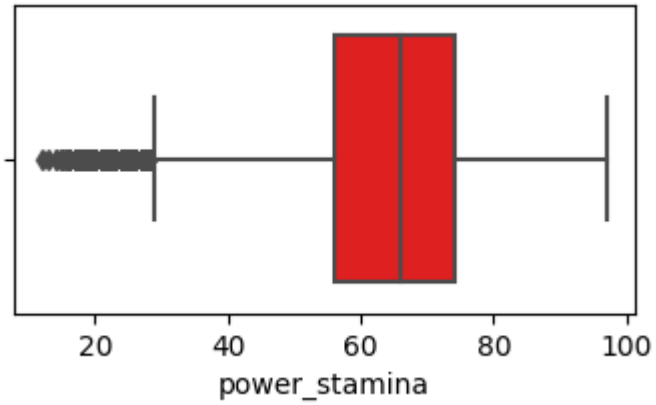
Pour visualiser et interpréter plus facilement ces statistiques, on peut tracer les boîtes à moustache.



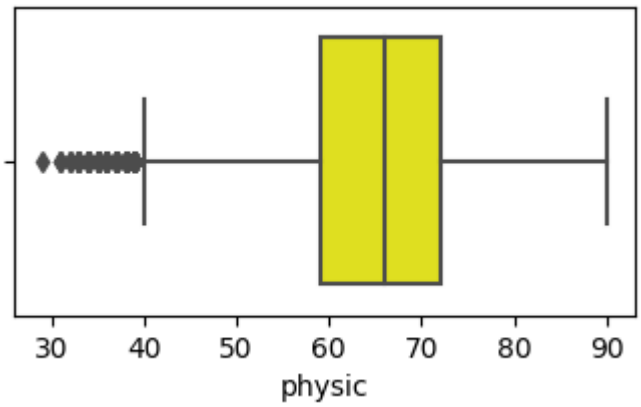
Les valeurs de l'attribut overall sont assez étalées, les 3/4 des données sont situées entre 60 et 70, bien que les autres valeurs s'étalent jusqu'à 50 voir 80. On constate un nombre non négligeable d'outliers (les points gris, déterminées par l'algorithme natif de seaborn) notamment autour de 90 dans les extrêmes supérieurs.



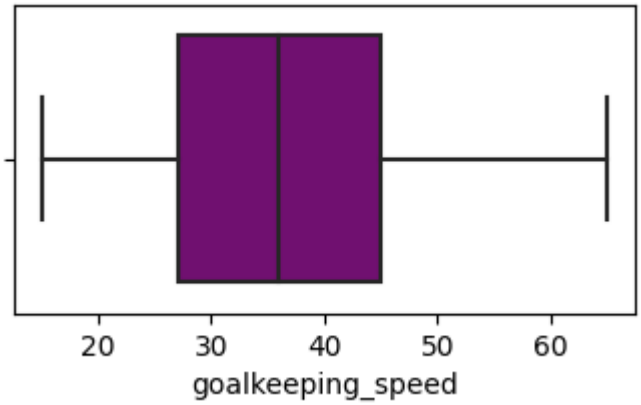
De même, les potentiels sont peu concentrées autour de la valeur 70, les 3/4 centraux des données sont situés entre 65 et 75 et le quart restant de données est assez étalé, et on note encore plus d'outliers pour cette feature.



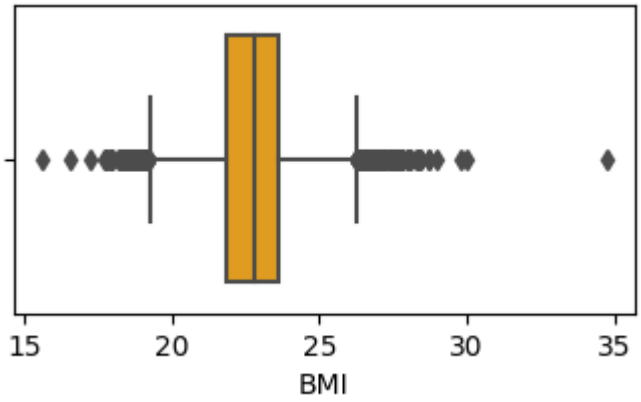
La power_stamina médiane est à 66, avec un étalement des valeurs un peu plus important, et beaucoup d'outliers dans les extrêmes inférieurs, entre les valeurs 10 et 30.



La répartition pour "physic" est analogue à la précédente.



La vitesse des gardiens de but est concentrée autour de 35, la majorité des données s'étalent entre 25 et 45, le quart restant s'étalent relativement peu comparé aux features précédemment observées (la boîte est plus large, les moustaches moins longues).



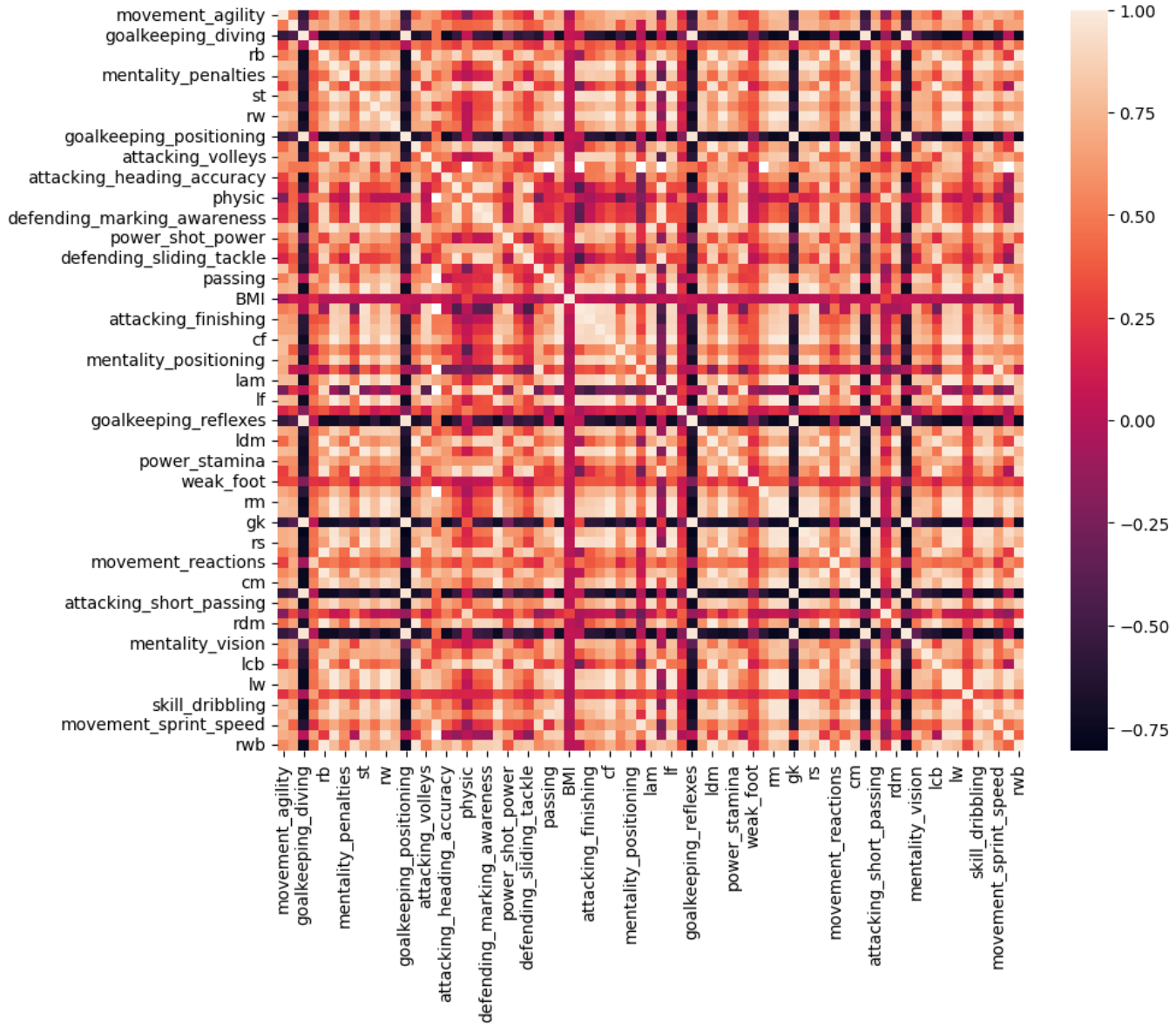
Pour cet attribut d'indice de masse corporelle, on remarque que les données sont très concentrées, la grande majorité autour de la valeur de 23 (boîte très étroite). Le quart restant de données s'étale vers 18 pour la partir inférieure et jusque 27 pour la partie supérieure.

On note un assez grand nombre d'outliers. Notamment un point au BMI de 35, très isolé du reste des données : il y a 5 unités d'écart avec le point le plus proche, alors que 3/4 des données sont comprises dans un intervalle inférieur à 2 unités.
On peut essayer d'identifier le joueur correspondant à cet outlier :

```
'Saheed Adebayo Akinfenwa'
```

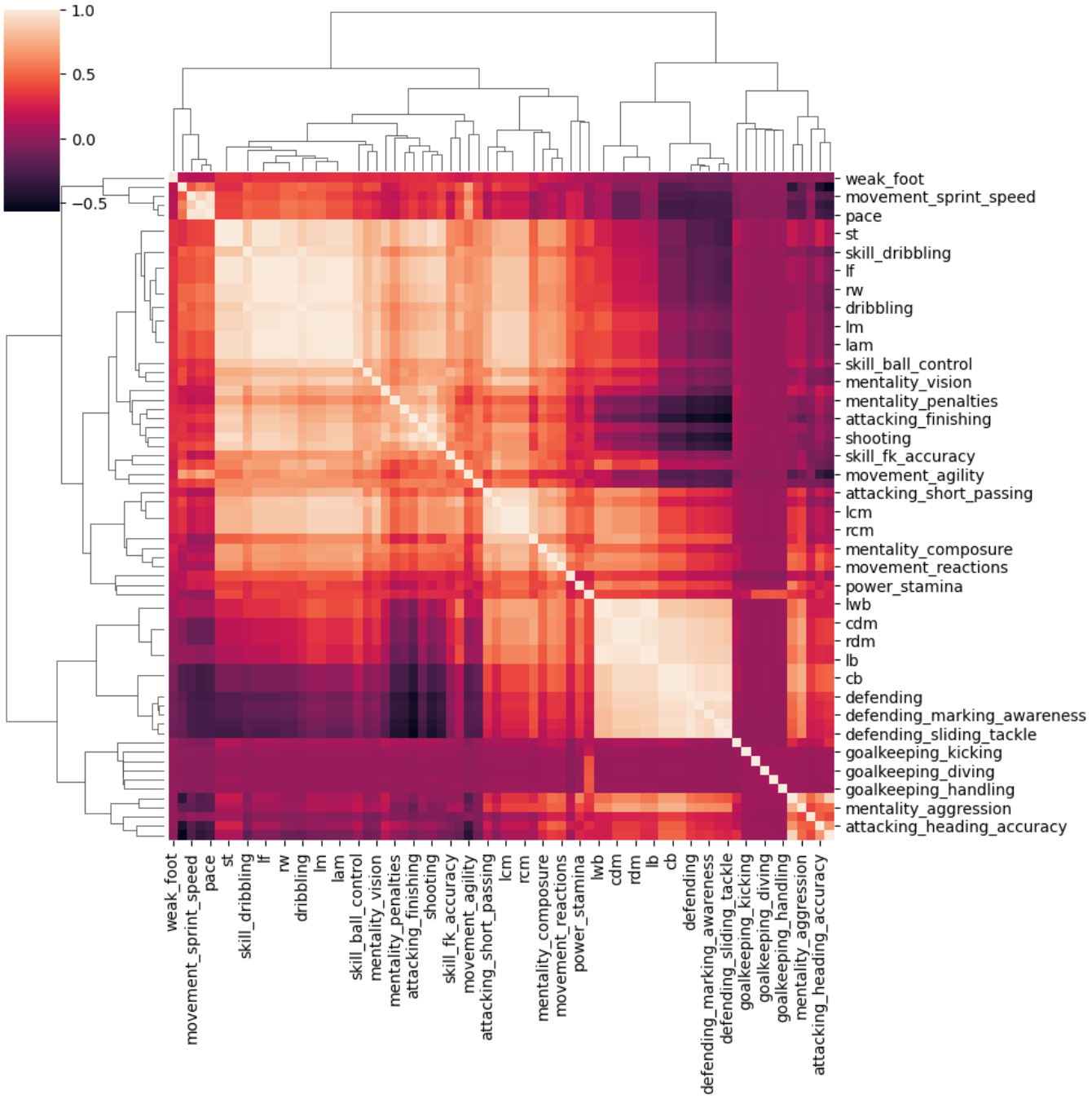
Analyse de corrélation

Nous avons effectué la préparation initiale des données, à partir de critères logiques de formattage et de compression de certaines informations. Pour pousser plus loin le nettoyage de nos données afin de les fournir à un modèle d'apprentissage, on va analyser les corrélations entre features et supprimer celles qui sont trop corrélées à d'autres, car cela réduit les performances de nombreux modèles. On cherche à ne pas dupliquer l'information portée par ces attributs.
On ne garde que les features numériques.



On observe pas mal de corrélations positives comme négatives, il est difficile d'y voir clair.
Pour faciliter cette tâche nous allons afficher cette heatmap après un traitement de clustering.
Ce traitement n'est pas possible avec des valeurs nulles (NaN), on va donc supprimer les goals et la colonne relative à ces derniers (goalkeeping_speed).

On retire les lignes avec des valeurs à NaN, on peut tracer la clustermap.



Cette visualisation est beaucoup plus informative sur les groupes de features fortement corrélées entre-elles.
On remarque rapidement 3 groupes de features correspondant aux 3 grands carrés blancs présents sur la diagonale.

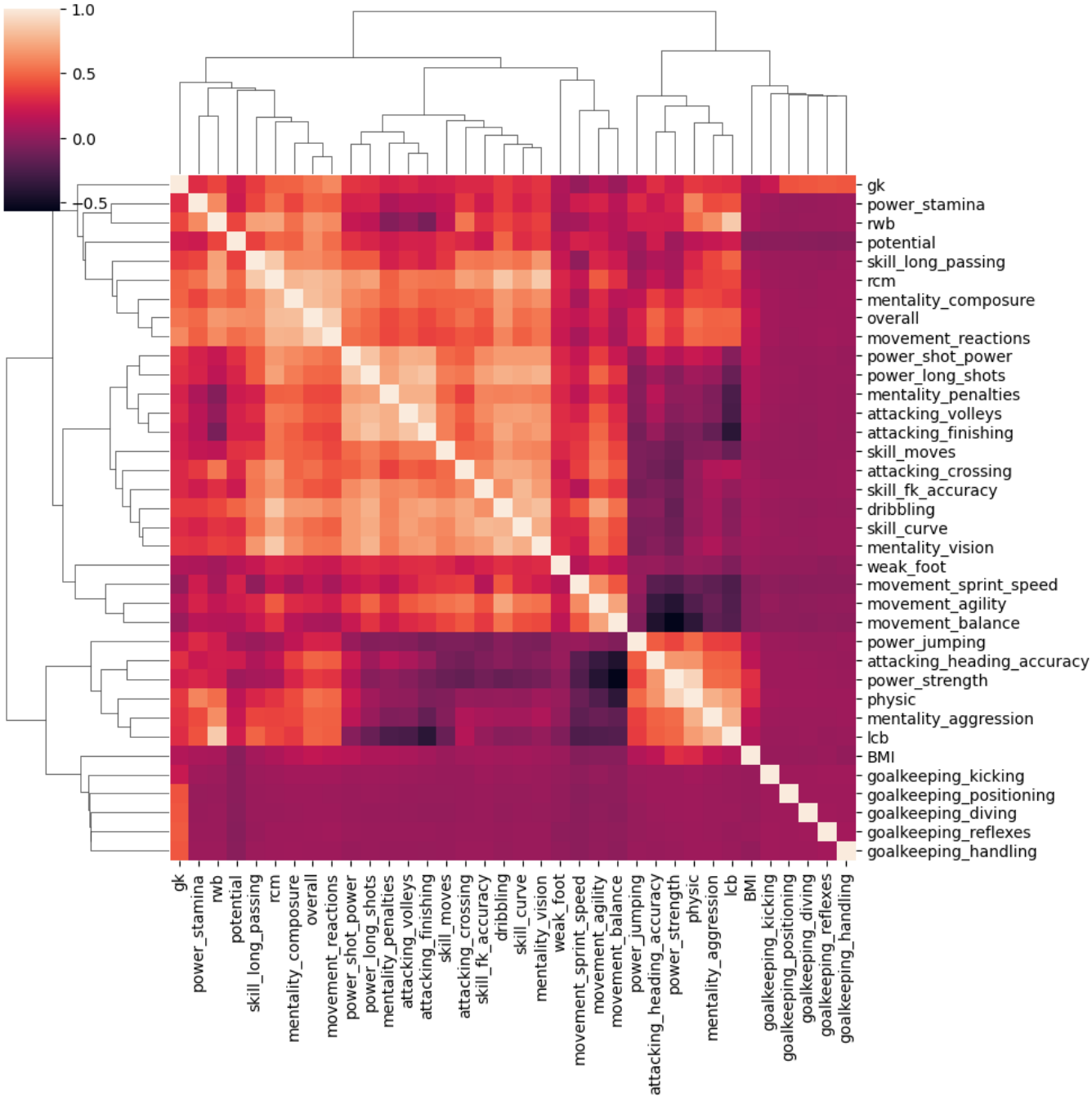
On supprime les features qui sont corrélées à plus de 90% avec une autre (positivement ou négativement). On applique un traitement itératif qui trouve les features qui ont une telle corrélation, deux à deux, puis on en supprime une, et on recommence.

Avec ce traitement, il ne reste que 36 colonnes qui sont "décorellées" (ou à moins de 90%). On a donc réalisé un plongement intéressant qui pourra nous servir plus tard lors de l'application de modèles d'apprentissage. Les features de ce dataset sont les suivantes :

```
'movement_agility  mentality_composure  goalkeeping_diving  overall
mentality_penalties  skill_fk_accuracy  attacking_crossing
```

```
goalkeeping_positioning  attacking_volleys  attacking_heading_accuracy
physic  power_shot_power  mentality_aggression  BMI  attacking_finishing
power_long_shots  movement_balance  power_jumping  goalkeeping_reflexes
skill_curve  power_stamina  weak_foot  skill_moves  gk  movement_reactions
goalkeeping_kicking  power_strength  goalkeeping_handling  mentality_vision
skill_long_passing  lcb  potential  rcm  movement_sprint_speed  dribbling
rwb'
```

On affiche la clustermap de ces attributs numériques décorrelés.



Prise en main des données

Equipe la plus chère

On créé l'équipe la plus chère, on se base sur leur valeur financière, et donc la feature "value_eur".

On extrait les goals et on trouve le plus cher.

```
'Gianluigi Donnarumma'
```

On extrait les 10 joueurs non goal les plus chers.

```
Composition de l'équipe la plus chère :
 Gianluigi Donnarumma   Kylian Mbappé Lottin   Erling Braut Haaland   Harry
Kane   Neymar da Silva Santos Júnior   Kevin De Bruyne   Robert Lewandowski
Frenkie de Jong   Jadon Sancho   Trent Alexander-Arnold   Joshua Walter
Kimmich


Coût total de l'équipe la plus chère : 1412500000.0 euros
```

Equipe la plus forte

On se base sur la statistique "overall".

```
Composition de l'équipe la plus forte :
 Jan Oblak   Lionel Andrés Messi Cuccittini   Robert Lewandowski
Cristiano Ronaldo dos Santos Aveiro   Neymar da Silva Santos Júnior   Kevin
De Bruyne   Kylian Mbappé Lottin   Harry Kane   N'Golo Kanté   Virgil van
Dijk   Joshua Walter Kimmich



Joueurs en commun dans les deux équipes :
 Kylian Mbappé Lottin  Neymar da Silva Santos Júnior  Robert Lewandowski
Harry Kane  Kevin De Bruyne  Joshua Walter Kimmich
```

Il y a 6 joueurs / 11 qui sont communs aux deux équipes, il y a donc probablement une corrélation importante entre la valeur d'un joueur et sa force (pour le top du classement en tout cas).

2. Segmentation

Préparation des données pour l'étude

Comme spécifié, on écarte les goals de cette étude, ainsi que les features "overall", "Dwage", "Dvalue".

```
MID    7033
DEF    6394
FWD    3680
GK     2132
```

```
MID    7033
DEF    6394
FWD    3680
```

On remarque que toutes les valeurs de la feature goalkeeping_speed sont nulles pour le dataset de cette étude, on supprime donc cette colonne pour la segmentation.

On observe les types des colonnes du dataset :

```
['int64', 'object', 'float64']
```

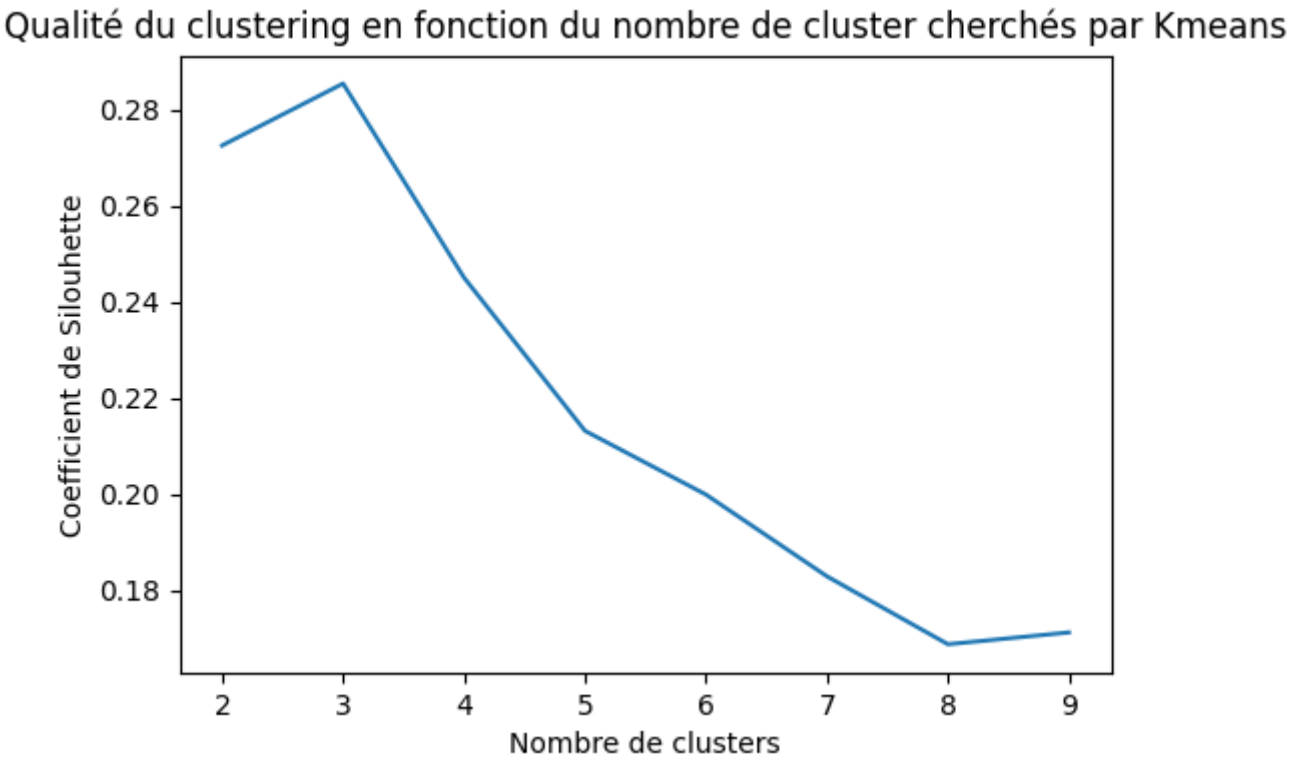
On ne garde que les features numériques, à savoir les int64 et float64.

On stocke l'indice de Mbappé pour plus tard.

On supprime l'id des joueurs, attribut qui va sinon polluer le clustering.

1ere tentative de clusterisation des joueurs : avec Kmeans

On évalue la qualité du clustering avec la métrique de Silhouette, qui est d'autant plus élevé que les clusters sont bien séparés et denses (minimisation de la distance intra-cluster et maximisation de la distance inter-clusters).
On essaye différentes valeurs de nombre de clusters et on cherche le nombre qui maximise cette métrique.



On choisit donc un nombre de clusters de 3, valeur qui maximise le coefficient de silouhette.

Interprétation des groupes

On réalise un clustering avec 3 clusters et on regarde si les positions des joueurs sont liées à ce clustering.

On compte le nombre de chaque poste dans chaque groupe.

```
classe 0 : {'FWD': 3283, 'MID': 2345, 'DEF': 16}
classe 1 : {'FWD': 386, 'MID': 3629, 'DEF': 2014}
classe 2 : {'DEF': 4364, 'MID': 1059, 'FWD': 11}
```

On remarque que chaque cluster comporte au moins 1.5 fois plus de joueurs d'une certaine position que des 2 autres positions.
C'est particulièrement marqué pour la classe 2 qui ne comporte que très peu de défenseurs.

En résumé :
La classe 0 correspond aux joueurs milieux avec des capacités défensives.
La classe 1 correspond aux joueurs très défensifs, avec des propriétés de milieu de terrain.
La classe 2 correspond aux joueurs offensifs et milieux offensifs.

Recherche de remplaçant pour MBappé

On réalise une transformation de l'espace basée sur l'apprenant Kmeans, puis on regarde le point (correspondant au joueur) le plus proche du point de Mbappé (le plus proche au sens de la distance euclidienne). Nous obtenons :

```
'Mohamed Salah Ghaly'
```

En prenant le joueur le plus proche de Mbappé dans l'espace transformé par le clustering à 3 groupes, on trouve **Mohamed Salah Ghaly**. On pourrait donc estimer d'après notre étude qu'il s'agit d'un remplaçant judicieux pour Kylian Mbappé.