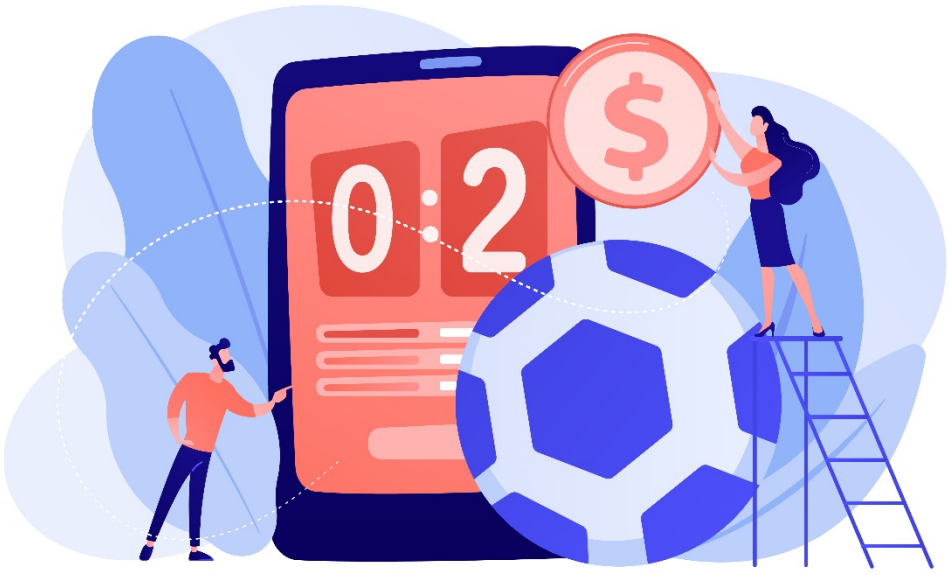
DataScientest • com

SPORTSBETPY

Projet paris sportifs



MATHIEU CROSNIER – AROUL SELVARADJOU

PARCOURS DATA SCIENTIST – PROMOTION JUILLET 2021

Table des matières

1. Introduction

De nos jours, les quantités de données ainsi que la puissance de calcul disponibles sont devenues telles qu'elles ont permis un essor considérable de l'Intelligence Artificielle, si bien que de très nombreux domaines y trouvent des applications. C'est notamment le cas du sport, et plus particulièrement les paris sportifs.

Tout parieur rêverait d'avoir à disposition un algorithme qui lui permettrait de déterminer, à sa place, le résultat d'un match à venir. C'est ce que nous allons tenter de faire au cours de ce projet.

L'objectif de notre projet consiste donc à développer un algorithme de Machine Learning qui serait capable de battre les bookmakers et ainsi permettre d'être gagnant financièrement sur le long terme.

2. Concepts théoriques

Avant de s'intéresser à la façon dont nous avons mené notre projet, il est nécessaire d'expliciter quelques notions liées au domaine des paris sportifs.

Le pari sportif est un jeu d'argent dans lequel l'objectif est de prédire le résultat d'une rencontre sportive. Dans le cas du football, que nous étudions dans notre projet, il y a 3 issues possibles :

- Victoire de l'équipe jouant à domicile,
- Victoire de l'équipe jouant à l'extérieur,
- Match nul.

Chaque issue possède une cote, et c'est elle qui permet de connaître à l'avance, en fonction de la somme misée, la somme remportée si l'événement se produit. Les cotes sont déterminées par des bookmakers, qui sont des sociétés permettant de parier de l'argent sur des évènements, le plus souvent sportifs.

Dans le monde, il existe plusieurs manières de représenter les cotes. Pour notre projet, nous utiliserons la cote européenne, qui est définie par la formule suivante :

$$Cote = \frac{Gain\ absolu}{Mise}$$

Le gain absolu auquel le parieur peut prétendre correspond donc à la cote multipliée par la mise. Pour obtenir le gain réel, il suffit de retrancher la mise de départ. Pour une mise donnée, plus la cote est importante et plus le gain potentiel est important.

Ce qui va intéresser le parieur, c'est d'être gagnant en moyenne sur le long terme. Ainsi, nous devons nous intéresser à l'espérance du gain, qui correspond au gain moyen que l'on peut espérer remporter par pari. L'espérance du gain, pour un match, est définie par la formule suivante :

$$E(Gain) = \sum_i x_i \times p_i$$

où x_i et p_i correspondent respectivement au gain et à la probabilité liés à l'événement i.

Ici, nous n'avons que 2 événements possibles :

- Gagner le pari,
- Perdre le pari.

L'espérance du gain s'écrit alors :

$$E(Gain) = \big[(\text{probabilité de gagner le pari}) \times (\text{gain absolu d'un pari gagnant}) + (\text{probabilité de perdre le pari}) \times (\text{gain absolu d'un pari perdant})\big] - Mise$$

Le gain absolu d'un pari perdant étant égal à 0, si nous appelons p la probabilité de gagner le pari, nous obtenons :

$$E(Gain) = p \times Mise \times Cote - Mise$$

soit

$$E(Gain) = Mise \times (p \times Cote - 1)$$

L'objectif étant d'obtenir une espérance positive, cela se traduit par :

$$E(Gain) \geq 0 \leftrightarrow p \times Cote \geq 1$$

soit

$$E(Gain) \geq 0 \ \leftrightarrow Cote \geq \frac{1}{p}$$

Tout l'enjeu des paris sportifs consiste à déterminer le plus précisément possible la valeur de p.

La cote est en fait analogue à l'inverse d'une probabilité, $\frac{1}{p}$ correspond donc à la cote estimée par le parieur.

Il est à noter que c'est ce que fait le bookmaker pour déterminer ses cotes, il estime la probabilité de chaque issue. Cependant, il prend une marge, qui lui assure d'être gagnant à long terme. Ainsi, les cotes sont réduites par rapport à ce qu'elles devraient être si elles suivaient les probabilités (les probabilités sont surestimées et leur somme est supérieure à 1). Cette marge varie selon les bookmakers, c'est pourquoi il est nécessaire de comparer les cotes qu'ils proposent.

Nous pouvons donc finalement écrire que :

$$E(Gain) \geq 0 \ \leftrightarrow Cote_{bookmaker} \geq Cote_{parieur}$$

ou encore

$$E(Gain) \geq 0 \ \leftrightarrow p_{parieur} \geq p_{bookmaker}$$

Pour chacun des matchs, l'objectif du parieur est donc de déterminer si, pour une des 3 issues possibles, la probabilité qu'il estime est supérieure à celle qu'aura estimée le bookmaker (ou de manière analogue que la cote proposée par le bookmaker est supérieure à la cote qu'il aura estimée). C'est ce qu'on appelle la recherche des ***Value Bets***, c'est-à-dire les paris ayant de la valeur.

C'est ce que nous allons tenter de faire dans ce projet.

3. Constitution du jeu de données

Pour réaliser notre projet, nous avons décidé de travailler sur le football. Le football étant le sport le plus populaire en Europe, le volume de données disponibles est conséquent.

Nous voulions que notre jeu de données soit à la fois constitué de données sur les matchs joués, mais aussi sur les équipes qui participaient à ces matchs.

Pour ce faire, nous avons récupéré les données à 2 endroits :

- https://www.football-data.co.uk/data.php pour les données relatives aux matchs.
  Ce site contient l'historique des matchs des principaux championnats à travers le monde sur de nombreuses saisons. Les données récupérées comportent les résultats, les statistiques ainsi que les cotes des différents matchs.
- https://www.kaggle.com/stefanoleone992/fifa-21-complete-player-dataset pour les données relatives au jeu vidéo FIFA. Ce site contient les données des jeux vidéo FIFA 15 jusqu'à FIFA 21 correspondants aux saisons 2014-2015 à 2020-2021.
  Le jeu vidéo FIFA est la référence dans le domaine des simulations de football. Au fil des années, le jeu s'est étoffé aussi bien en termes de nombre de joueurs et d'équipes présents dans le jeu, mais aussi de caractéristiques considérées, ce qui en fait aujourd'hui une base de données très solide.

Afin que notre jeu de données soit tel que nous le souhaitions, il fallait respecter plusieurs conditions :

- Que les structures des données relatives aux matchs soient semblables, quels que soient la saison et le championnat considérés.
- Que les équipes des championnats et saisons sélectionnés soient présentes dans le jeu vidéo FIFA correspondant.

Compte-tenu de ces critères, voici un tableau récapitulatif des différents championnats et saisons considérés dans notre jeu de données :

CHAMPIONNAT	SAISON						
	2014 2015	2015 2016	2016 2017	2017 2018	2018 2019	2019 2020	2020 2021
Belgian Jupiler Pro League				X	X	X	X
English Premier League	X	X	X	X	X	X	X
English League Championship	X	X	X	X	X	X	X
English League One	X	X	X	X	X	X	X
English League Two	X	X	X	X	X	X	X
French Ligue 1	X	X	X	X	X	X	X
French Ligue 2				X	X	X	X
German 1. Bundesliga	X	X	X	X	X	X	X
German 2. Bundesliga				X	X	X	X
Holland Eredivisie				X	X	X	X

Italian Serie A	X	X	X	X	X	X	X
Italian Serie B				X	X	X	
Portuguese Liga ZON SAGRES				X	X	X	X
Scottish Premiership	X	X	X	X	X	X	X
Spain Primera Division	X	X	X	X	X	X	X
Spanish Segunda División				X	X	X	X
Turkish Süper Lig				X	X	X	X

3.1. Jeu de données des résultats de matchs

La 1ère étape de la construction de notre jeu de données consiste à concaténer l'ensemble des données relatives aux matchs pour les différents championnats et saisons mentionnés dans le tableau ci-dessus.

Nous avons effectué plusieurs traitements afin d'effectuer un 1er nettoyage ce jeu de données :

- Suppression des matchs ayant des statistiques incomplètes.
- Suppression des colonnes inutiles ou inexploitables, notamment les colonnes descriptives et tout ce qui concerne des cotes complexes fournies par les bookmakers.
- Conversion des formats des données.
- Renommage de colonnes.

Ces différents traitements ne suffisent pas puisqu'il reste toujours des valeurs manquantes. En effet, le jeu de données contient 24 variables explicatives donnant des lots de 3 cotes pour 8 différents bookmakers. Sachant que chaque bookmaker ne propose pas forcément des cotes pour tous les matchs de notre jeu de données, les variables liées aux cotes contiennent de nombreuses valeurs manquantes. Cependant, chaque match possède des cotes chez au moins un bookmaker. Nous avons donc décidé de récupérer, pour chaque match, les cotes maximales parmi tous les bookmakers (*Max H*, *Max D*, *Max A*). Cela nous permet à la fois de traiter les valeurs manquantes mais aussi de maximiser nos gains lorsque nous les calculerons par la suite, à l'étape du Machine Learning. En effet, nous avons vu au §2 que plus la cote était élevée, plus le gain potentiel était élevé. Nous avons aussi vu que compte tenu des marges que prennent les bookmakers, qui sont variables de l'un à l'autre, il est important de comparer les cotes proposées et de sélectionner les plus intéressantes.

Une fois les maximums de cotes calculés, nous n'avons plus besoin des 24 variables explicatives donnant les cotes des différents bookmakers, ce qui nous permet de ne plus avoir de valeurs manquantes dans notre jeu de données.

Tous ces différents traitements nous permettent d'obtenir un jeu de données constitué de 35918 matchs, pour lesquelles nous avons 23 variables explicatives qui sont les suivantes :

- Environnement du match :
  - Saison en cours (*Season*),
  - Championnat concerné (*Division*),
  - Date (*Date*),
  - Noms des 2 équipes (*Home team*, *Away team*).
- Statistiques du match pour chacune des 2 équipes :
  - Nombre de buts marqués (*FTHG*, *FTAG*),

- Nombre de tirs (*HS*, *AS*),
- Nombre de tirs cadrés (*HST*, *AST*),
- Nombre de fautes (*HF*, *AF*),
- Nombre de corners (*HC*, *AC*),
- Nombre de cartons jaunes (*HY*, *AY*),
- Nombre de cartons rouges (*HR*, *AR*).

- Cotes des bookmakers :
  - Cote maximale pour la victoire de l'équipe à domicile (*Max H*),
  - Cote maximale pour le match nul (*Max D*),
  - Cote maximale pour la victoire de l'équipe à l'extérieur (*Max A*).

- Résultat du match (*FTR*).
  Cette variable constitue notre variable cible, c'est une variable catégorielle qui peut prendre 3 valeurs possibles :
  - H : victoire de l'équipe à domicile (Home en anglais),
  - D : match nul (Draw en anglais),
  - A : victoire de l'équipe à l'extérieur (Away en anglais).

Voici un aperçu des 5 premières lignes de ce jeu de données :

	Season	Division	Date	Home team	Away team
0	2018-2019	German 2. Bundesliga	2018-08-03	Hamburg	Holstein Kiel
1	2018-2019	German 2. Bundesliga	2018-08-04	Bochum	FC Koln
2	2018-2019	German 2. Bundesliga	2018-08-04	Greuther Furth	Sandhausen
3	2018-2019	German 2. Bundesliga	2018-08-04	Regensburg	Ingolstadt
4	2018-2019	German 2. Bundesliga	2018-08-05	Darmstadt	Paderborn

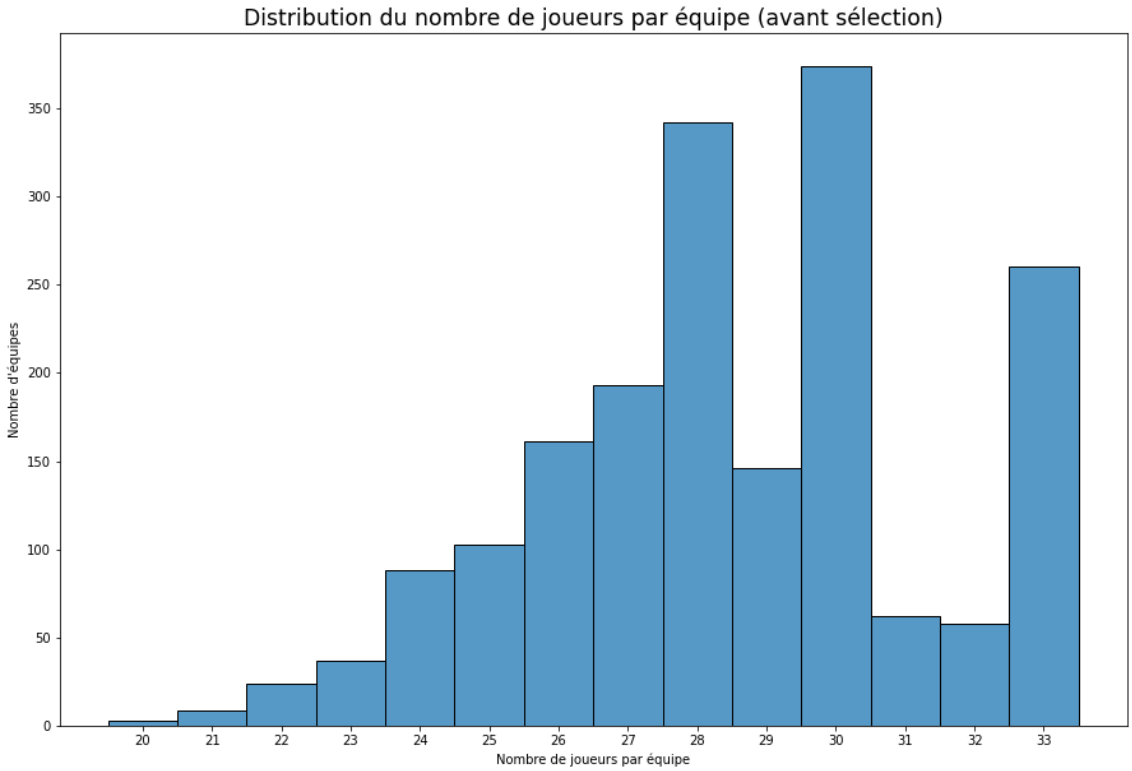
FTHG	FTAG	FTR	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR	Max H	Max D	Max A
0.0	3.0	A	14.0	16.0	3.0	8.0	11.0	11.0	4.0	10.0	1.0	3.0	0.0	0.0	1.78	3.75	5.25
0.0	2.0	A	14.0	15.0	2.0	5.0	17.0	19.0	6.0	1.0	3.0	2.0	0.0	1.0	2.91	3.30	2.62
3.0	1.0	H	16.0	10.0	5.0	5.0	19.0	11.0	6.0	8.0	1.0	1.0	0.0	0.0	2.40	3.30	3.43
2.0	1.0	H	13.0	13.0	9.0	3.0	23.0	20.0	5.0	9.0	1.0	3.0	0.0	0.0	3.20	3.40	2.40
1.0	0.0	H	7.0	7.0	2.0	1.0	14.0	21.0	8.0	3.0	3.0	4.0	0.0	0.0	2.65	3.30	2.95

3.2. Jeu de données du jeu vidéo FIFA

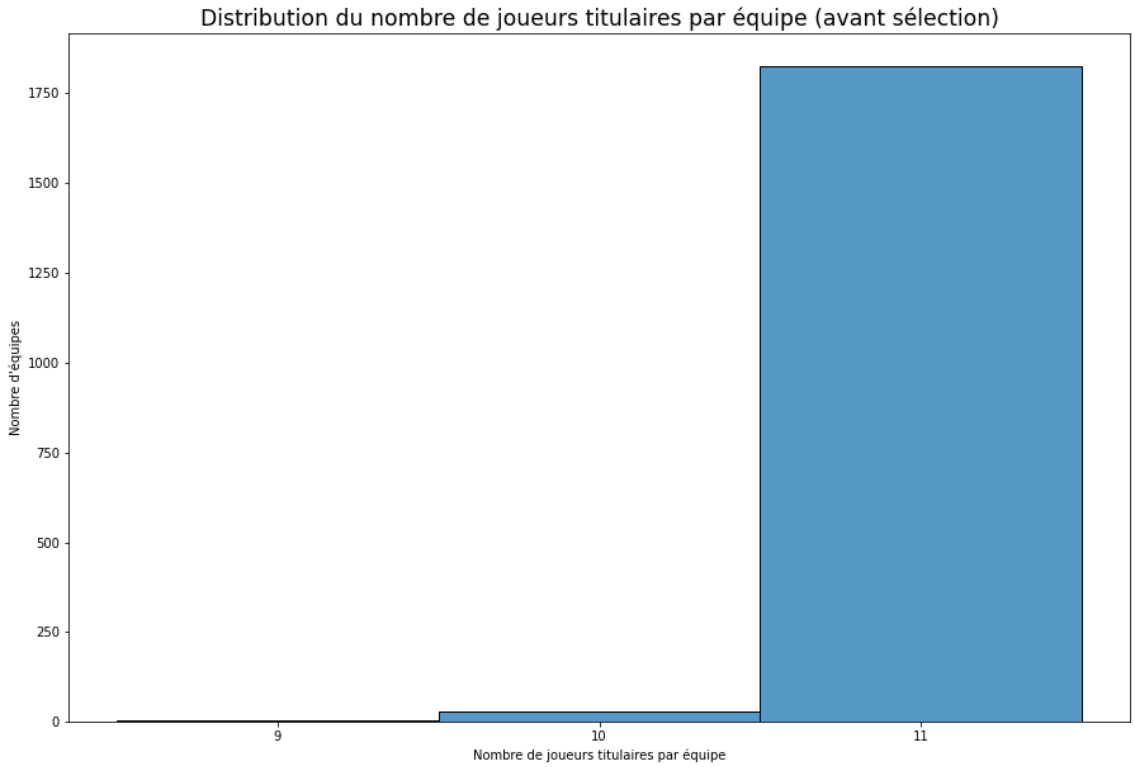
La 2ème étape de la constitution de notre jeu de données consiste à concaténer les données issues des jeux vidéo FIFA. Nous obtenons ainsi un jeu de données composé des 122841 joueurs présents dans ces jeux, pour lesquels nous avons de très nombreuses informations sur les caractéristiques du joueur, individuellement et au sein de son équipe, pour lesquelles nous ne rentrerons pas dans les détails pour le moment. Ce jeu de données n'est pas directement exploitable puisque, d'une part, il contient de nombreux joueurs qui ne font pas partie des championnats et des équipes qui nous intéressent, et d'autre part, nous souhaitons avoir des données par équipe et non par joueur.

Nous avons donc effectué divers traitements afin d'obtenir ce que nous souhaitions :

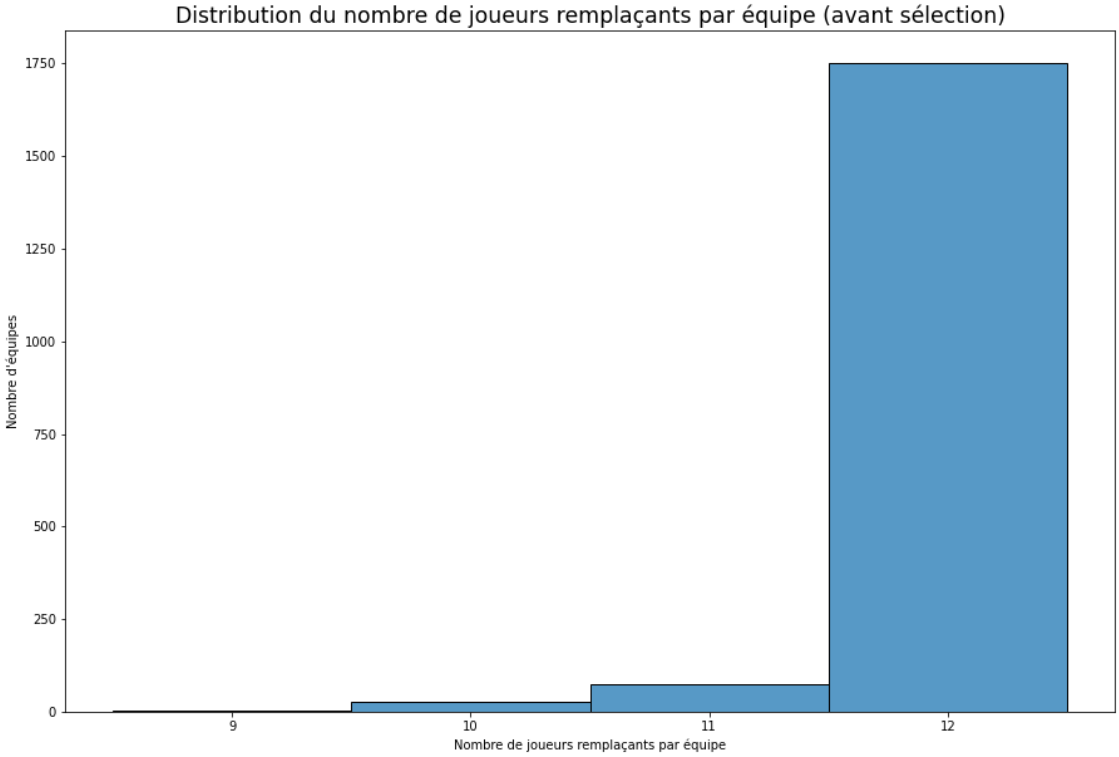
- Filtre afin de conserver uniquement les joueurs jouant dans les équipes des championnats et saisons que nous avons retenus. Nous avons désormais 53100 joueurs.
- Filtre afin de conserver uniquement les joueurs titulaires et remplaçants dans chaque équipe. Si nous regardons la distribution du nombre de joueurs par équipe, nous remarquons qu'il y a une assez grande disparité. Sur le graphe ci-dessous, nous observons que le nombre de joueurs par équipe varie entre 20 et 33.



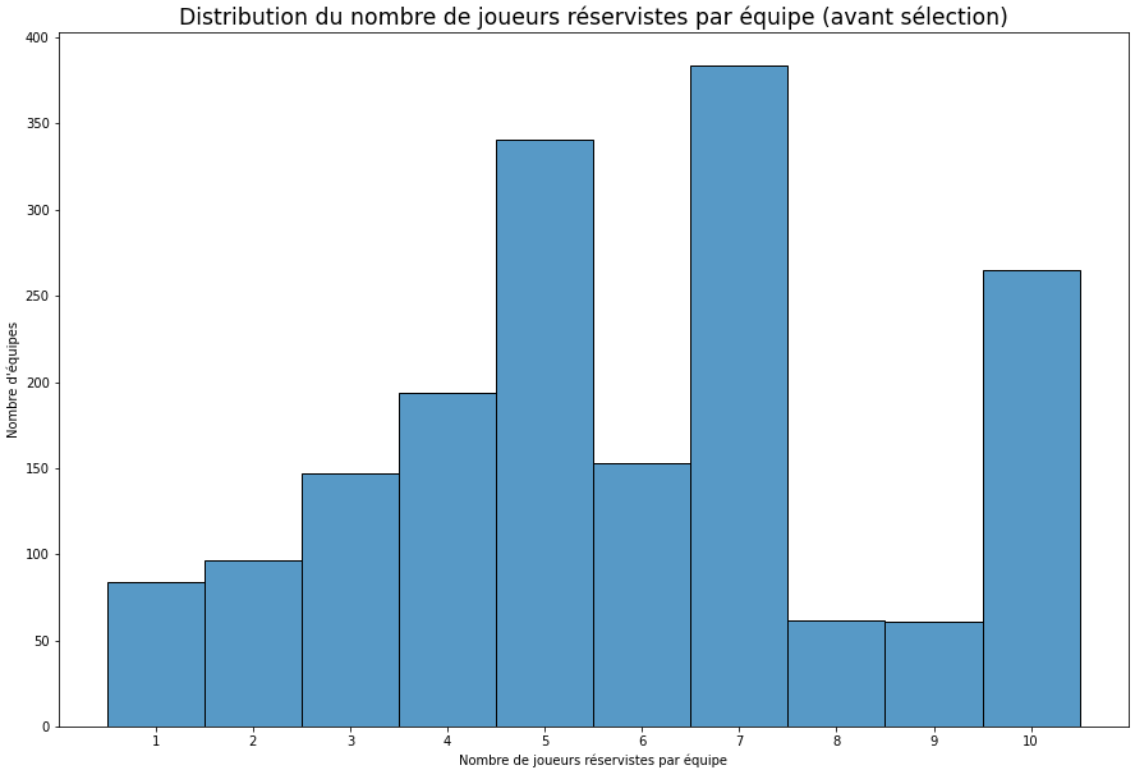
Chaque équipe possède des joueurs titulaires, remplaçants et réservistes. Regardons le détail du nombre de joueurs par équipe selon ces postes.



Le nombre de titulaires au football est de 11, il est donc logique de retrouver une très grande majorité d'équipes avec 11 joueurs titulaires (dans notre jeu de données, certaines équipes en ont moins, 9 ou 10, de manière inexpliquée).

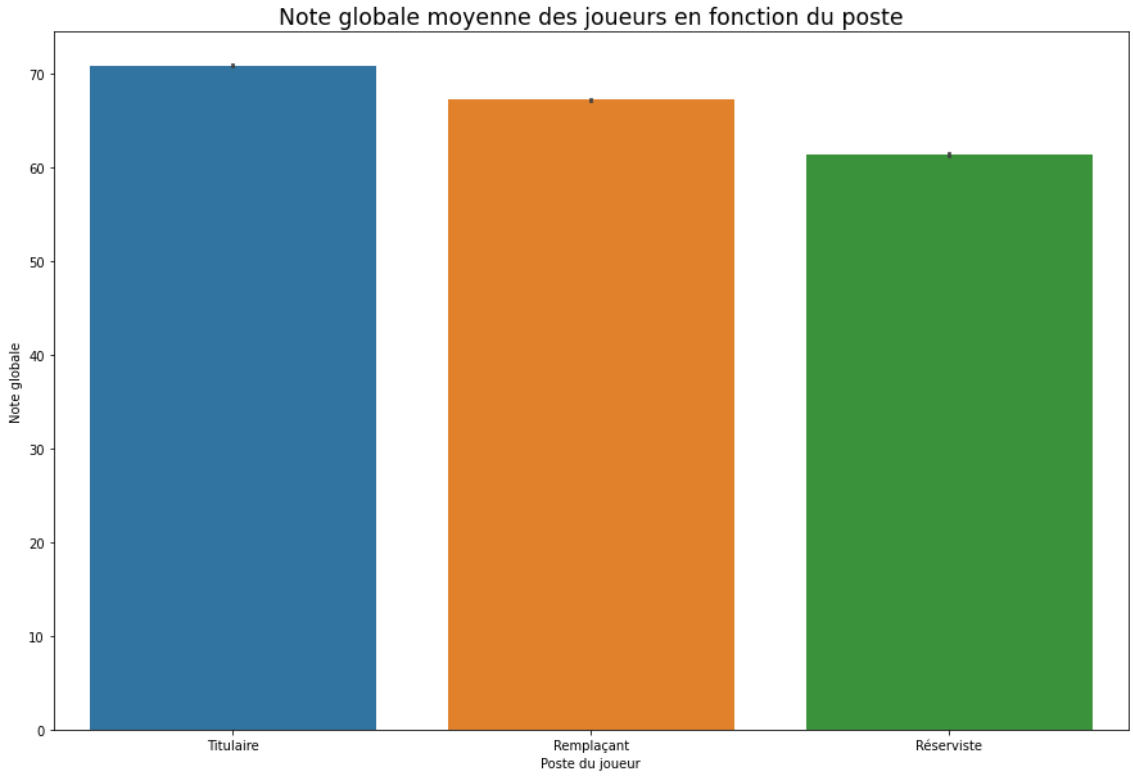


Le nombre de remplaçants étant en général réglementé lui aussi, son nombre de varie pas beaucoup (entre 9 et 12 par équipe).



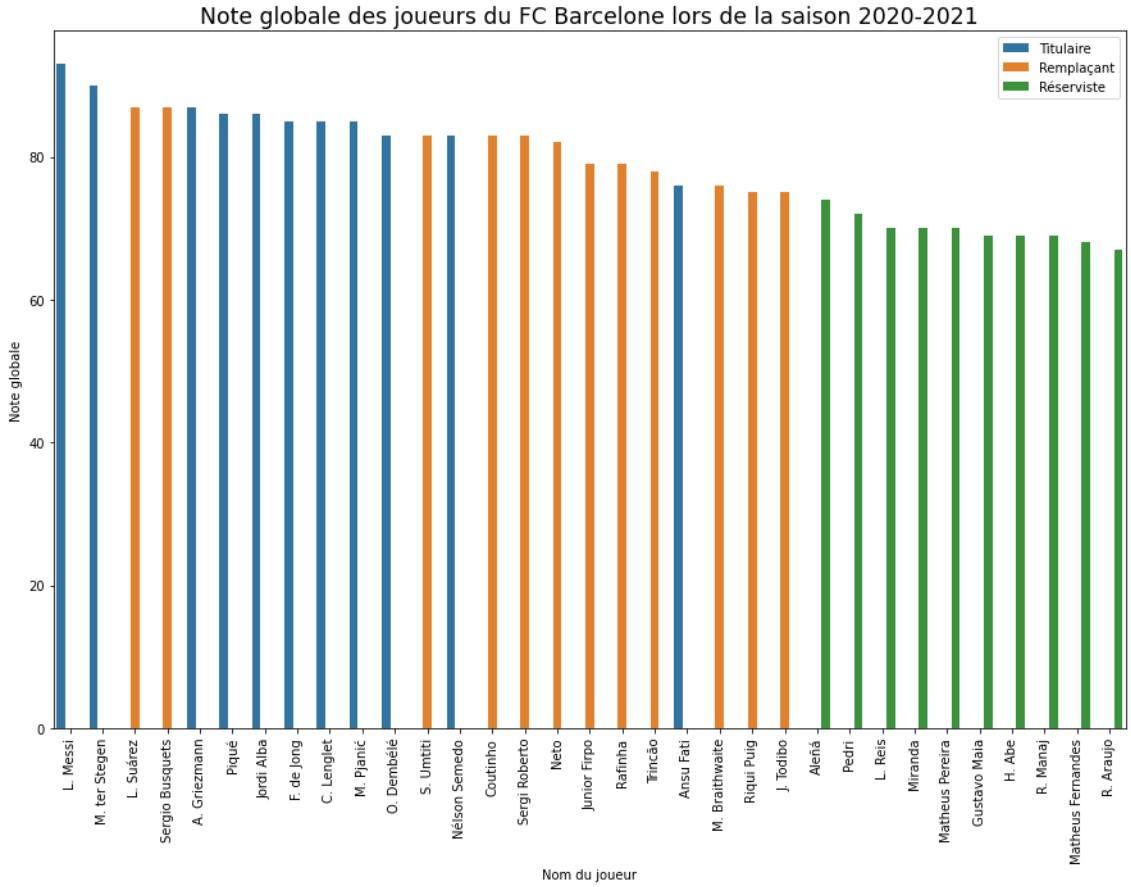
Nous observons que c'est principalement le nombre de joueurs réservistes qui varie beaucoup et qui provoque cette disparité dans le nombre total de joueurs dans une équipe. Nous comptons entre 1 et 10 joueurs réservistes par équipe.

Ce qui fait la différence entre un joueur titulaire, remplaçant ou réserviste, c'est généralement son niveau de jeu. Regardons ce qu'il en est de la note globale moyenne des joueurs en fonction de leur poste.



Nous pouvons observer sur le graphe ci-dessus que la note globale moyenne des joueurs réservistes est nettement inférieure aux joueurs titulaires et remplaçants.

Regardons désormais ce qu'il en est au sein d'une même équipe.

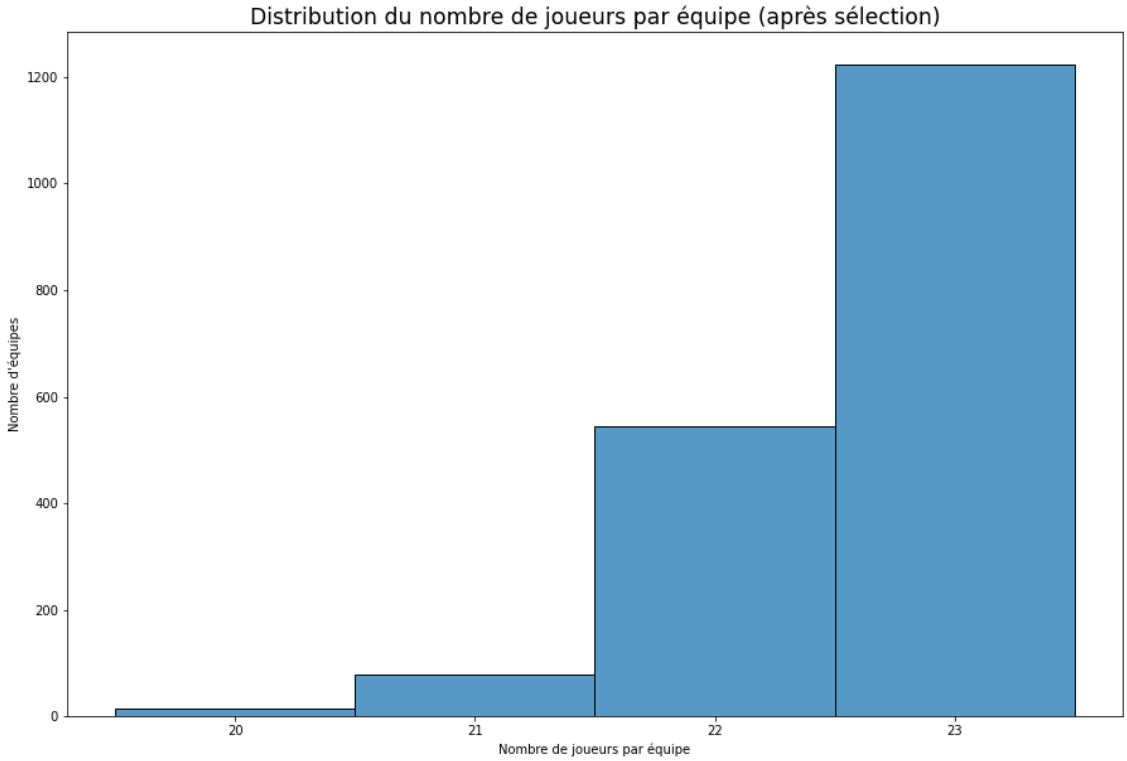


Au sein d'une même équipe, nous observons au travers de l'exemple du FC Barcelone de la saison 2020-2021 que tous les réservistes sont les joueurs ayant la note globale la moins élevée.

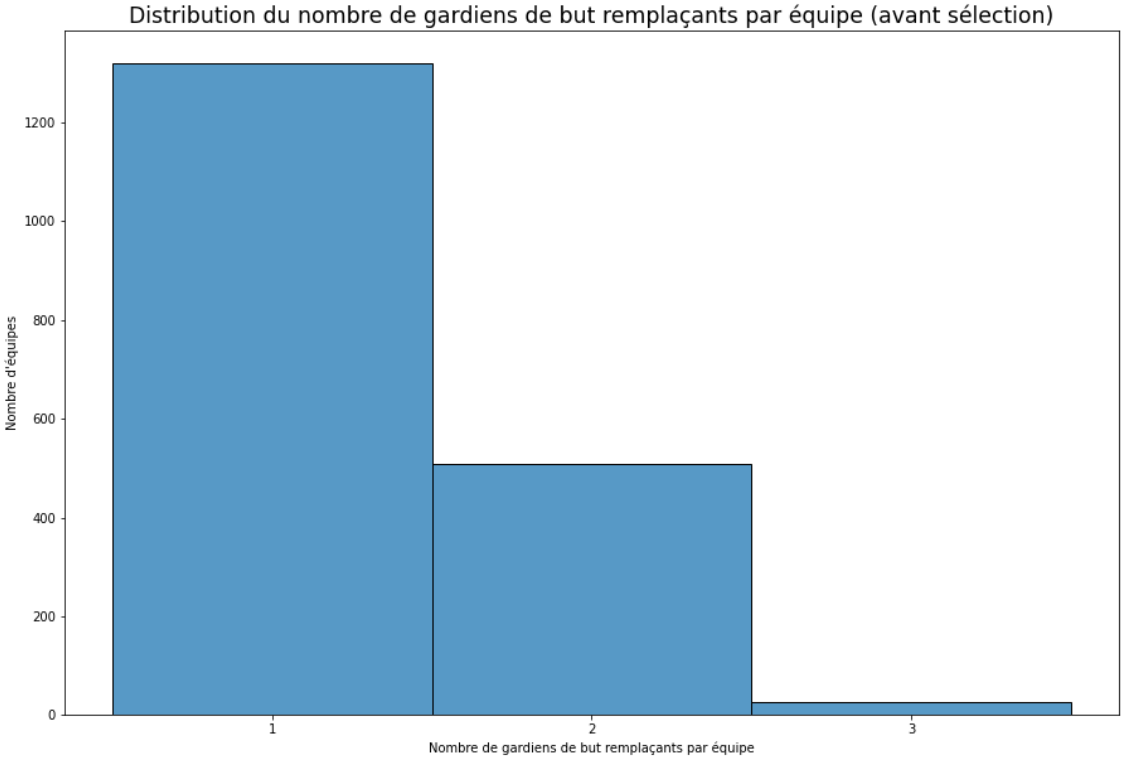
Les joueurs réservistes sont donc généralement des joueurs moins forts et qui de surcroit ne jouent pas ou très peu.

Afin de ne pas pénaliser les équipes en possédant un plus grand nombre, en faisant baisser la moyenne de leurs caractéristiques, nous avons décidé de les exclure de notre jeu de données.

Le graphe ci-dessous nous permet d'observer que cela a ainsi permis d'uniformiser le nombre de joueurs pris en compte par équipe (entre 20 et 23 désormais).



- Filtre afin d'avoir seulement 1 gardien de but remplaçant par équipe. En effet, le nombre de gardiens de but parmi les remplaçants varie d'une équipe à l'autre (entre 1 et 3), comme nous pouvons le voir sur le graphe ci-dessous.



Pour les mêmes raisons que pour les joueurs réservistes, nous avons exclu les gardiens au-delà du 1er gardien remplaçant.

La suppression des réservistes et des gardiens au-delà du 1er gardien remplaçant nous réduit le nombre de joueurs de notre jeu de données à 42038.

- Suppression des variables inutiles ou inexploitables.
- Transformation de variables pour les rendre exploitables.
- Groupement des données selon la saison, le championnat et l'équipe, afin d'obtenir les valeurs moyennes de chaque variable non plus pour chaque joueur mais pour chaque équipe.

Nous obtenons finalement un jeu de données contenant 54 variables pour les 1860 équipes ayant participé aux différents championnats et saisons que nous avons sélectionnés. Ces variables sont les suivantes :

- Environnement de l'équipe :
  - Saison en cours (*Season*),
  - Championnat concerné (*Division*),
  - Nom de l'équipe (*Team*).
- Caractéristiques physiques :
  - Age (*Age*),
  - Taille en cm (*Height (cm)*),
  - Poids en kg (*Weight (kg)*).
- Niveau général :
  - Niveau général de l'équipe (*Overall*),
  - Niveau général de l'équipe en considérant le potentiel des jeunes joueurs (*Potential*).
- Valeur financière :
  - Valeur sur le marché des transferts (*Value (€)*),
  - Salaire (*Wage (€)*).
- Contrats :
  - Durée de validité du contrat (*Contract valid until*),
  - Durée de présence dans l'équipe (*Joined*).
- Statistiques générales :
  - Vitesse (*Pace*),
  - Tir (*Shooting*),
  - Passe (*Passing*),
  - Dribble (*Dribbling*),
  - Défense (*Defending*),
  - Physique (*Physic*),
  - Réputation internationale (*International reputation*),
  - Capacité à utiliser le pied faible (*Weak foot*),
  - Capacité à réaliser des gestes techniques (*Skill moves*).
- Statistiques détaillées :
  - Gardiens de but :
    - Détente horizontale (*Gk diving*),
    - Jeu à la main (*Gk handling*),
    - Jeu au pied (*Gk kicking*),
    - Réflexes (*Gk reflexes*),
    - Vitesse (*Gk speed*),
    - Positionnement (*Gk positioning*).

- Attaque :
  - Centres (*Attacking crossing*),
  - Finition (*Attacking finishing*),
  - Jeu de tête (*Attacking heading accuracy*),
  - Passes courtes (*Attacking short passing*),
  - Reprises de vollée (*Attacking volleys*).
- Technique :
  - Dribble (*Skill dribbling*),
  - Effets (*Skill curve*),
  - Coups francs (*Skill fk accuracy*),
  - Passes longues (*Skill long passing*),
  - Contrôle de balle (*Skill ball control*).
- Mouvement :
  - Accélération (*Movement acceleration*),
  - Vitesse de sprint (*Movement sprint speed*),
  - Agilité (*Movement agility*),
  - Temps de réaction (*Movement reactions*),
  - Équilibre (*Movement balance*).
- Puissance :
  - Puissance de tir (*Power shot power*),
  - Détente verticale (*Power jumping*),
  - Endurance (*Power stamina*),
  - Force (*Power strength*),
  - Tirs de loin (*Power long shots*).
- Mental :
  - Agressivité (*Mentality aggression*),
  - Interceptions (*Mentality interceptions*),
  - Positionnement (*Mentality positioning*),
  - Vision du jeu (*Mentality vision*),
  - Pénalties (*Mentality penalties*).
- Défense :
  - Tacles debouts (*Defending standing tackle*),
  - Tacles glissés (*Defending sliding tackle*).

Voici un aperçu des 5 premières lignes de ce jeu de données :

	Season	Division	Team	Age	Height (cm)	Weight (kg)	Overall	Potential	Value (€)	Wage (€)	International reputation	Weak foot	Skill moves	Contract valid until	Pace	Shooting	Passing	Dribbling
0	2014-2015	English League Championship	Birmingham City	25.304348	182.217391	76.782609	65.217391	69.478261	5.615217e+05	6608.695652	1.086957	2.956522	2.260870	1.565217	72.523810	55.476190	57.238095	60.761905
1	2014-2015	English League Championship	Blackburn Rovers	25.869565	183.608696	76.000000	66.869565	70.521739	7.823913e+05	8695.652174	1.086957	3.000000	2.260870	2.000000	70.142857	54.285714	57.523810	61.619048
2	2014-2015	English League Championship	Blackpool	25.739130	183.173913	78.086957	63.347826	66.826087	4.434783e+05	5739.130435	1.000000	2.956522	2.434783	1.043478	68.000000	52.619048	54.047619	59.000000
3	2014-2015	English League Championship	Bolton Wanderers	27.181818	183.500000	75.590909	67.727273	69.727273	1.009318e+06	12181.818182	1.227273	3.045455	2.409091	1.909091	68.250000	56.200000	59.900000	63.250000
4	2014-2015	English League Championship	Bournemouth	26.347826	180.391304	76.391304	66.304348	69.652174	7.093478e+05	7695.652174	1.043478	3.304348	2.521739	2.130435	69.190476	54.666667	60.476190	63.047619

Defending	Physic	Gk diving	Gk handling	Gk kicking	Gk reflexes	Gk speed	Gk positioning	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
49.571429	69.857143	72.0	66.5	67.5	72.5	37.5	67.0	54.173913	51.521739	51.913043	58.043478	43.130435	54.826087	49.391304	46.565217	53.130435	58.913043
51.619048	67.714286	70.0	68.0	75.0	69.0	54.5	70.5	52.391304	49.739130	56.521739	59.521739	47.521739	55.565217	48.173913	44.086957	54.391304	60.695652
46.952381	65.666667	68.5	57.5	59.0	67.5	49.5	59.0	47.782609	46.739130	54.478261	54.391304	47.130435	54.086957	47.086957	47.739130	50.043478	56.826087
55.200000	69.350000	71.5	67.0	64.5	74.5	50.5	68.5	54.636364	51.863636	59.681818	60.818182	51.227273	57.636364	51.409091	46.181818	56.500000	61.318182
52.809524	67.095238	65.0	66.0	65.5	66.5	42.5	64.5	56.000000	49.043478	52.173913	60.478261	42.304348	57.217391	50.782609	51.130435	56.217391	61.130435

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	Defending standing tackle	Defending sliding tackle	Joined
68.173913	70.565217	66.608696	61.608696	65.739130	58.695652	69.304348	70.434783	68.434783	54.043478	61.347826	47.478261	52.695652	53.565217	51.260870	48.043478	47.260870	1
68.521739	68.956522	64.913043	65.000000	63.608696	57.521739	69.782609	67.478261	68.000000	51.260870	60.173913	48.086957	53.217391	54.130435	50.434783	50.826087	47.043478	2
64.565217	67.652174	62.391304	59.304348	60.695652	60.000000	63.913043	63.260870	69.086957	49.304348	52.086957	46.260870	51.478261	53.000000	51.217391	44.086957	43.260870	0
66.045455	66.954545	64.409091	64.954545	63.363636	60.500000	66.454545	69.590909	69.590909	52.454545	60.909091	52.000000	52.727273	55.681818	48.272727	53.000000	50.818182	2
66.956522	66.608696	69.478261	62.478261	67.782609	60.217391	64.434783	68.782609	66.913043	53.304348	58.565217	50.043478	53.478261	56.869565	53.608696	50.260870	48.086957	2

3.3. Fusion des jeux de données des résultats de matchs et du jeu vidéo FIFA

Dans les sections précédentes, nous avons construit nos 2 jeux de données principaux. L'objectif est désormais de les fusionner afin d'obtenir un jeu de données contenant à la fois les résultats et statistiques des matchs mais aussi les caractéristiques des équipes participant à chacun de ces matchs.

Pour ce faire, nous devons effectuer notre fusion selon la saison, le championnat et l'équipe. Or, chaque match faisant intervenir 2 équipes, il faudra effectuer 2 fusions : une première pour l'équipe jouant à domicile puis une seconde pour l'équipe jouant à l'extérieur.

Afin de pouvoir réaliser ces fusions, il faut au préalable s'assurer que les valeurs prises par les 3 variables sur lesquelles nous voulons effectuer notre fusion correspondent bien entre les 2 jeux de données. Il est facile de le vérifier et de le corriger facilement pour la saison et le championnat, qui prennent respectivement 7 et 17 valeurs différentes, mais c'est beaucoup plus compliqué pour les équipes, qui sont au nombre de 1860.

Nous avons donc dû créer une fonction capable de trouver la correspondance entre les noms d'équipes des 2 jeux de données. Elle fonctionne de la façon suivante :

Pour chaque saison et chaque championnat :

- Comparaison du nombre d'équipes présentes dans chacun des jeux de données. S'il est différent, la fonction renvoie un message d'erreur. Cela peut signifier :
    - Qu'il manque les matchs de certaines équipes, auquel cas nous pouvons tout de même continuer puisque nous avons les informations nécessaires pour les matchs que nous avons,
    - Que certaines équipes n'existent pas dans le jeu vidéo FIFA, dans ce cas nous devons supprimer les matchs correspondants aux équipes manquantes.
- Chaque nom d'équipe du jeu de données des résultats de matchs est comparé à chaque nom d'équipe du jeu de données du jeu vidéo FIFA, mot pat mot. Les 2 noms d'équipes ayant le

plus de lettres en commun (et à nombre de lettres en commun égal, le moins de lettres différentes), seront déclarés comme étant correspondant,

- Une vérification est faite pour s'assurer que chacun des noms d'équipes du jeu vidéo FIFA (que nous prenons comme référence) n'est attribué qu'une seule fois.

Nous obtenons finalement un dictionnaire, qui pour chaque saison et chaque championnat, effectue la correspondance des noms des équipes entre le jeu de données des matchs et le jeu de données du jeu vidéo FIFA. En voici un aperçu pour le championnat « English League Championship » de la saison 2014-2015 :

```
{'2014-2015': {'English League Championship': {'Birmingham': 'Birmingham City',
   'Blackburn': 'Blackburn Rovers',
   'Blackpool': 'Blackpool',
   'Bolton': 'Bolton Wanderers',
   'Bournemouth': 'Bournemouth',
   'Brentford': 'Brentford',
   'Brighton': 'Brighton & Hove Albion',
   'Cardiff': 'Cardiff City',
   'Charlton': 'Charlton Athletic',
   'Derby': 'Derby County',
   'Fulham': 'Fulham',
   'Huddersfield': 'Huddersfield Town',
   'Ipswich': 'Ipswich Town',
   'Leeds': 'Leeds United',
   'Middlesbrough': 'Middlesbrough',
   'Millwall': 'Millwall',
   'Norwich': 'Norwich City',
   "Nott'm Forest": 'Nottingham Forest',
   'Reading': 'Reading',
   'Rotherham': 'Rotherham United',
   'Sheffield Weds': 'Sheffield Wednesday',
   'Watford': 'Watford',
   'Wigan': 'Wigan Athletic',
   'Wolves': 'Wolverhampton Wanderers'},
```

Nous pouvons désormais remplacer les noms des équipes dans notre jeu de données des matchs et le fusionner avec le jeu de données du jeu vidéo FIFA.

3.4. Création du jeu de données final

Une fois la fusion des 2 jeux de données effectuée, nous obtenons un jeu de données composé des 35918 matchs pour lesquels nous avons désormais 125 variables explicatives. En effet, nous avons toujours les 23 variables explicatives du jeu de données des matchs auxquelles nous ajoutons 2 fois (une fois pour chacune des 2 équipes participant à chaque match) les 51 variables explicatives du jeu de données du jeu vidéo FIFA.

Jusqu'alors, nous ne nous sommes pas réellement posé la question de la pertinence de nos variables explicatives pour le Machine Learning. C'est ce que nous allons faire à présent.

3.4.1. Différence domicile / extérieur pour les variables explicatives

La grande majorité de nos variables explicatives existent à la fois pour l'équipe à domicile et pour celle à l'extérieur. En ne prenant plus la valeur pour chaque équipe mais la différence de ces valeurs (valeur pour l'équipe à domicile – valeur pour l'équipe à l'extérieur) nous réduisons de manière conséquente

le nombre de ces variables explicatives. Ce format sera adopté aussi pour les variables que nous créerons par la suite.

3.4.2. Moyennes mobiles

Les statistiques du match présentées au §3.1 ne peuvent pas être exploitées telles quelles pour le Machine Learning. En effet, la prédiction du résultat d'un match se faisant avant le début de celui-ci, nous n'avons pas accès à ces informations au moment de la prédiction. Il est donc nécessaire de regarder à la place l'historique de ces statistiques sur les précédents matchs que chaque équipe aura joué, afin d'obtenir en quelque sorte un état de forme des équipes sur une période donnée. C'est pourquoi nous allons créer de nouvelles variables explicatives en effectuant des moyennes mobiles sur ces statistiques de match.
Voici les choix que nous avons faits concernant le calcul de ces moyennes mobiles :

- Suppression du nombre de fautes, de corners et de cartons (jaunes ou rouges). Nous avons jugé que l'historique de ces statistiques sur les précédents matchs n'était pas pertinent pour juger de l'état de forme d'une équipe et de sa capacité à gagner. Nous conserverons donc uniquement le nombre de buts marqués ainsi que le nombre de tirs total et le nombre de tirs cadrés de chaque équipe,
- Calcul du nombre de points marqués, selon la méthode appliquée dans la totalité des championnats de football considérés :
    - 3 points pour une victoire,
    - 1 point pour un match nul,
    - 0 point pour une défaire.
- Calculs uniquement sur la saison en cours. Les équipes variant d'une année sur l'autre, nous ne calculons pas de moyennes mobiles à cheval sur plusieurs saisons,
- Calculs avec et sans différenciation de la notion de matchs à domicile ou à l'extérieur. Pour l'équipe jouant à domicile, nous regardons à la fois l'historique de tous ses matchs joués, mais aussi celui concernant uniquement ses matchs joués à domicile. Pour l'équipe jouant à l'extérieur, nous regardons à la fois l'historique de tous ses matchs joués, mais aussi celui concernant uniquement ses matchs joués à l'extérieur,
- Calculs à court terme (1, 3 et 5 derniers matchs) mais aussi depuis le début de la saison (20 derniers matchs pour la différenciation domicile / extérieur, 40 derniers matchs sans différenciation). Afin de ne pas perdre de données, si le nombre de matchs disponibles est inférieur à la fréquence définie pour effectuer nos moyennes mobiles celles-ci s'effectuent tout de même sur le nombre de matchs disponibles au lieu de retourner un NaN.

Pour récapituler ce qui a été dit, voici les moyennes mobiles que nous calculons :

- Nombre de buts marqués en moyenne sur les x derniers matchs, tous matchs confondus (*Full time goals scored (x games)*),
- Nombre de buts concédés en moyenne sur les x derniers matchs, tous matchs confondus (*Full time goals conceded (x games)*),
- Nombre de tirs en moyenne sur les x derniers matchs, tous matchs confondus (*Shots (x games)*),
- Nombre de tirs cadrés en moyenne sur les x derniers matchs, tous matchs confondus (*Shots on target (x games)*),

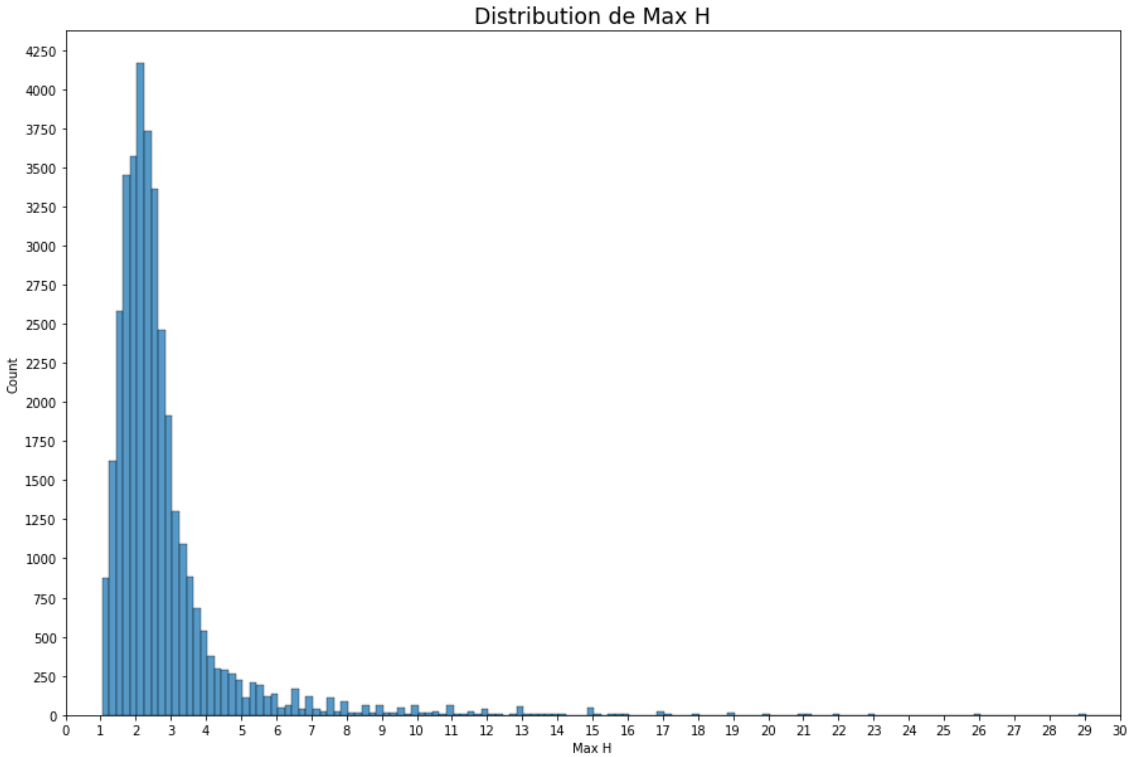
- Nombre de points marqués en moyenne sur les x derniers matchs, tous matchs confondus (*Points (x games)*),
- Nombre de buts marqués en moyenne sur les x derniers matchs à domicile ou à l'extérieur (*Full time goals scored (home or away) (x games)*),
- Nombre de buts concédés en moyenne sur les x derniers matchs à domicile ou à l'extérieur (*Full time goals conceded (home or away) (x games)*),
- Nombre de tirs en moyenne sur les x derniers matchs à domicile ou à l'extérieur (*Shots (home or away) (x games)*),
- Nombre de tirs cadrés en moyenne sur les x derniers matchs à domicile ou à l'extérieur (*Shots on target (home or away) (x games)*).
- Nombre de points marqués en moyenne sur les x derniers matchs à domicile ou à l'extérieur (*Points (home or away) (x games)*).

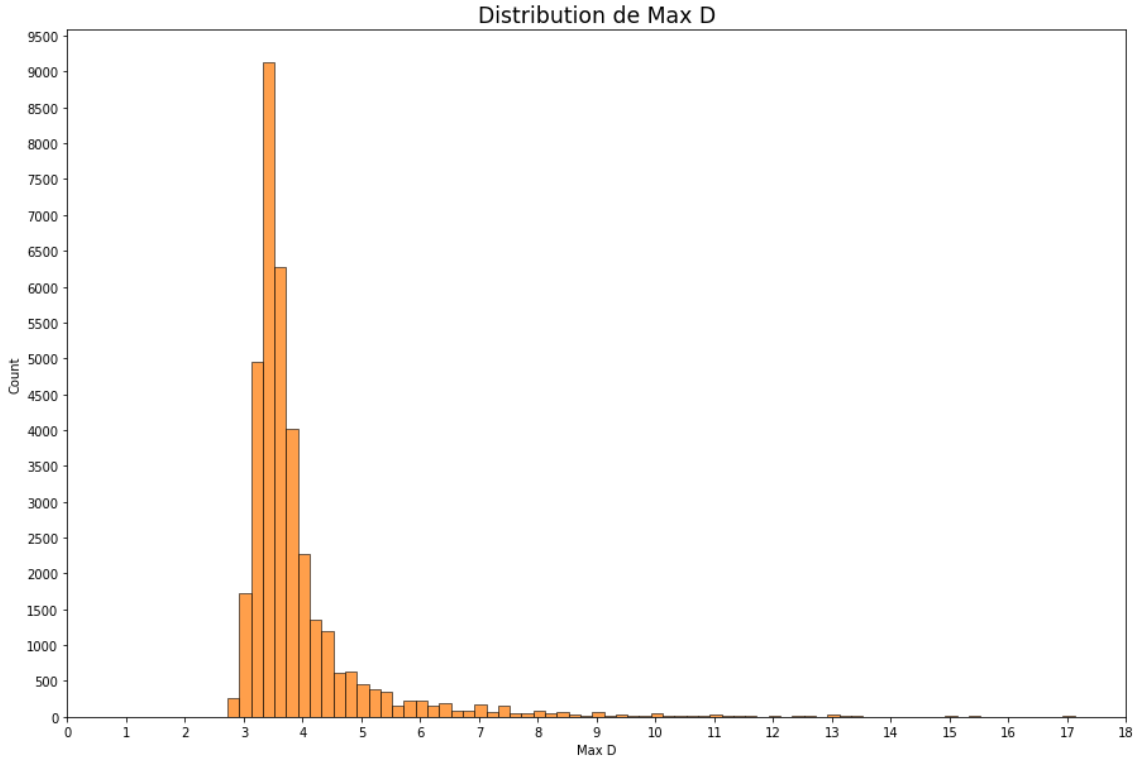
Avec x pouvant prendre les valeurs 1, 3, 5, 20 ou 40.

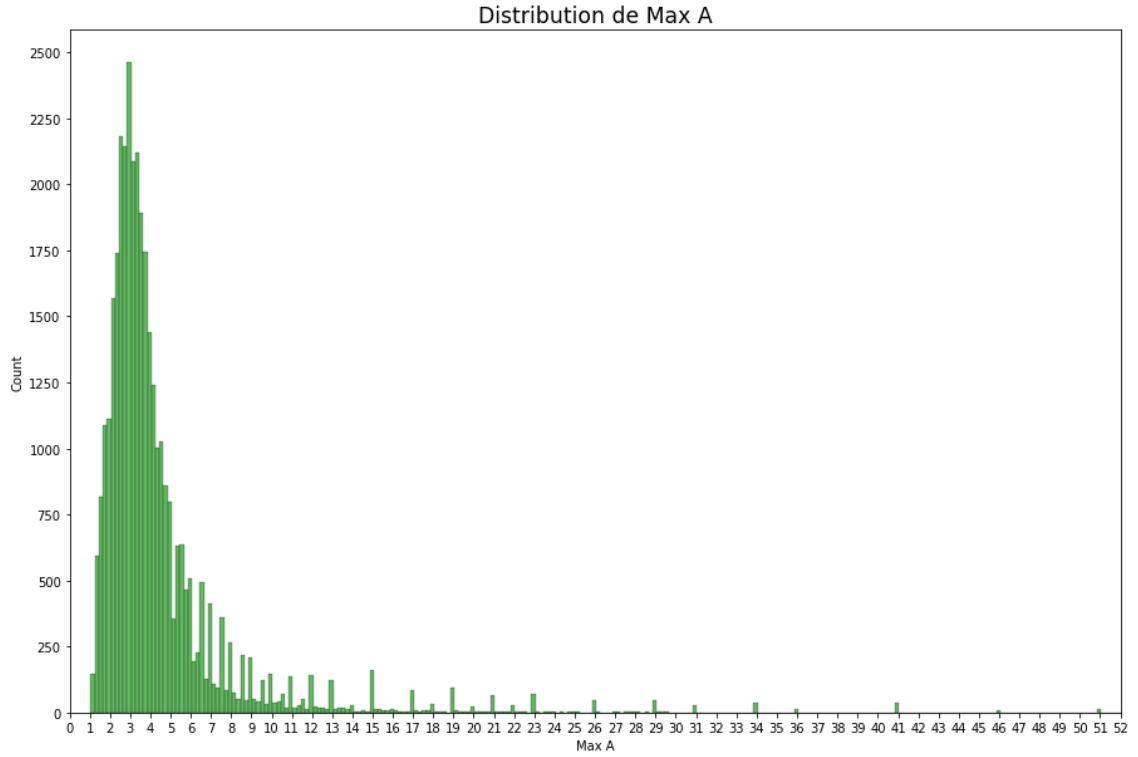
3.4.3. Cotes des bookmakers

Nous avons expliqué au §3.1 que nous ne travaillerions pas directement avec les cotes des bookmakers, mais avec la valeur maximale de chacune d'entre elles. Ces cotes maximales ne nous serviront pas directement comme variable explicative de notre modèle, mais nous servirons à calculer les gains attendus à la suite de nos prédictions. Cependant, nous avons jugé utile de conserver tout de même l'information contenue dans les cotes au travers de la variable explicative *Cote*, qui est égale à *Max A* – *Max H* (§3.1).

Si nous regardons la distribution des variables *Max H*, *Max D* et *Max A* ci-dessous, nous pouvons voir qu'elles possèdent des valeurs très extrêmes :

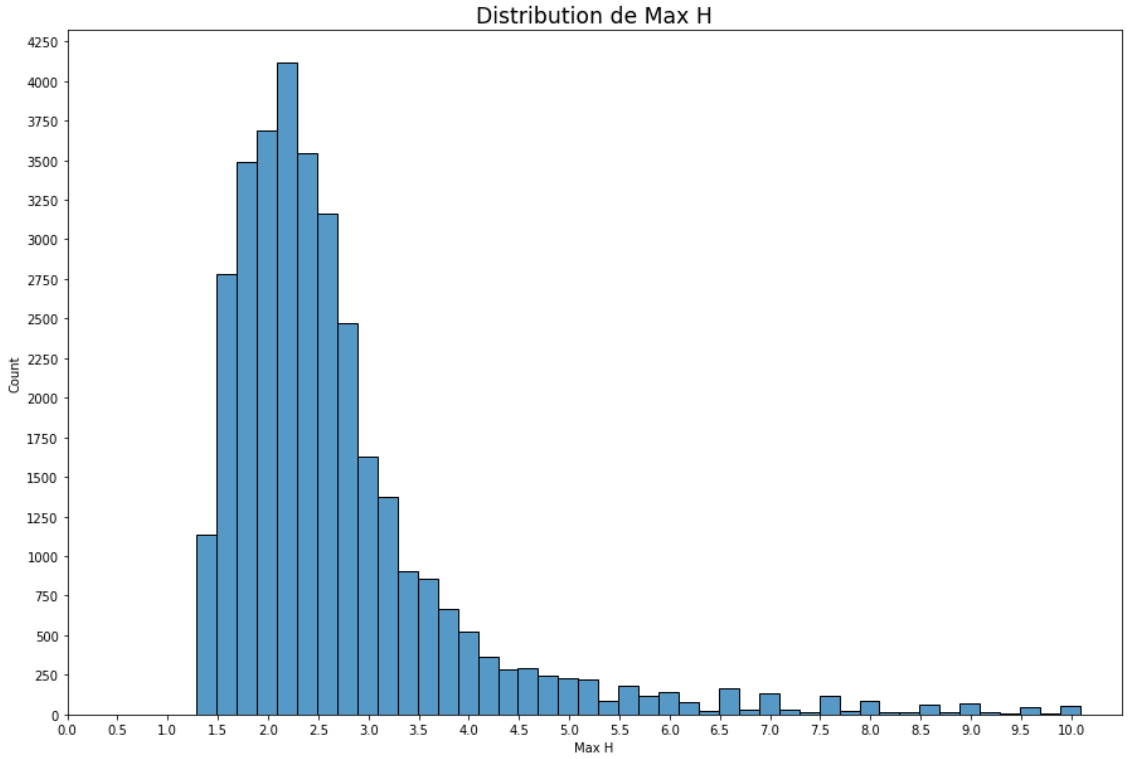


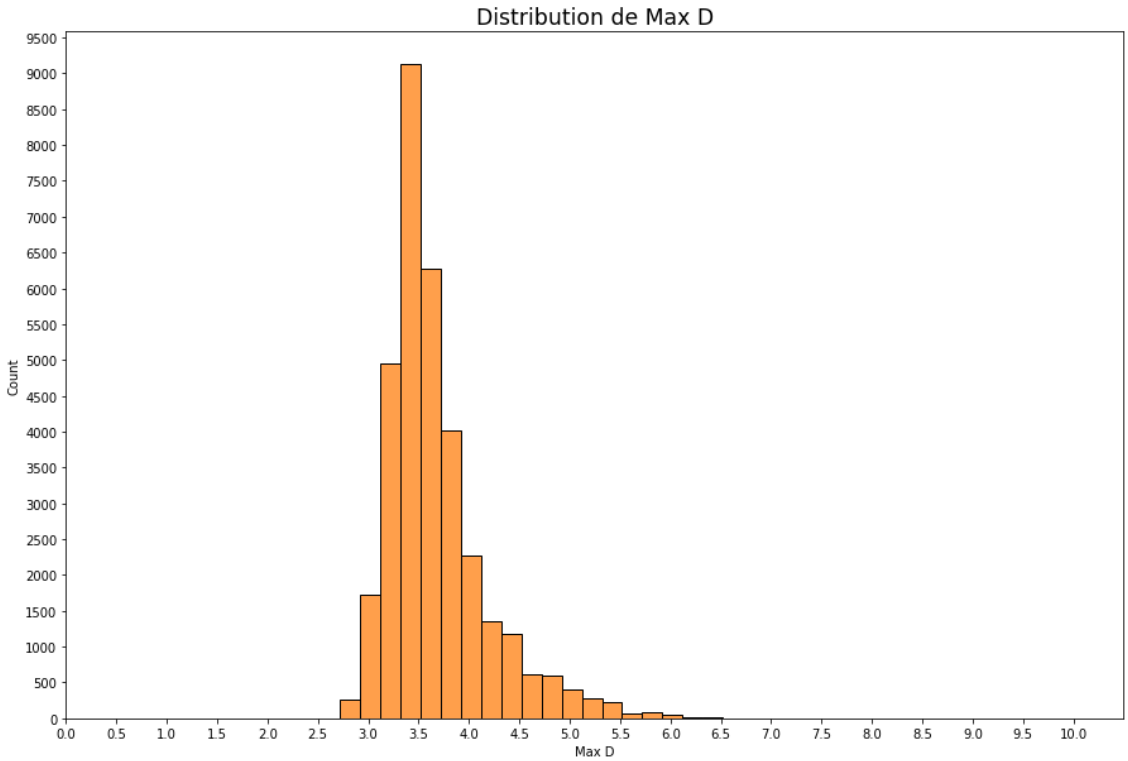


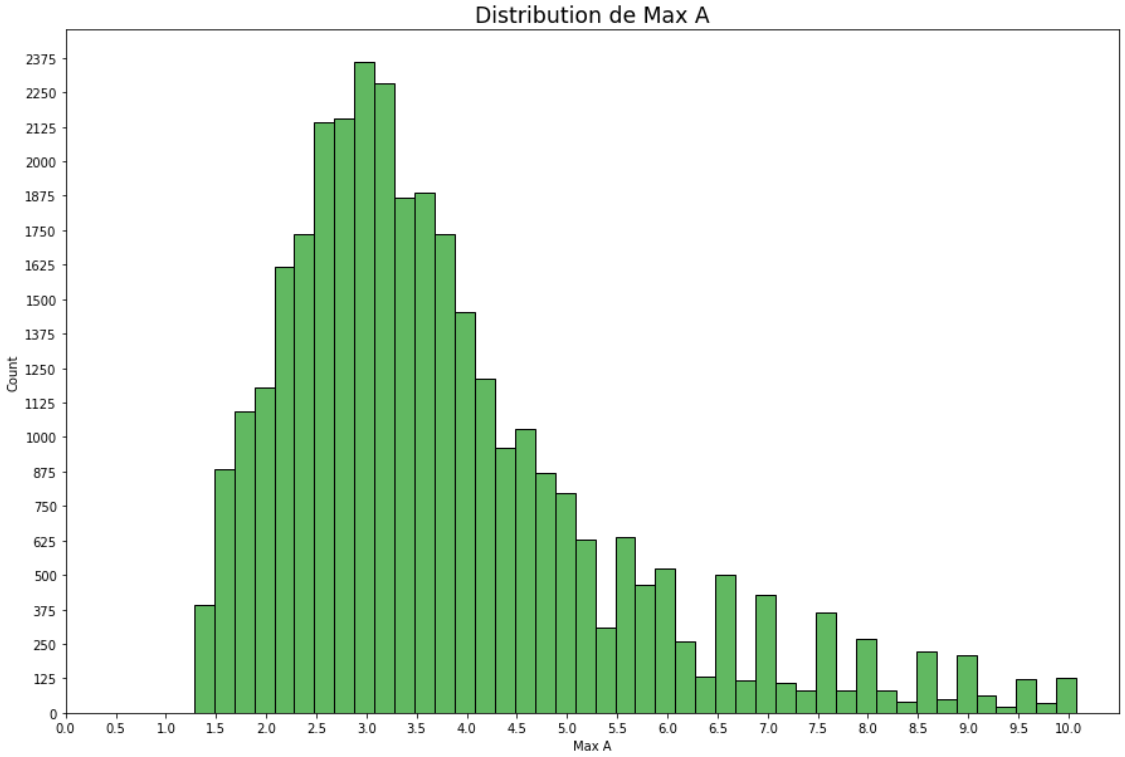


L'objectif de nos modèles de Machine Learning sera de calculer des cotes afin de les comparer à celles des bookmakers et ainsi identifier les ***Value Bets***. Si nous conservons des valeurs de cotes de bookmakers trop extrêmes (qui correspondent à un nombre de matchs très réduit) nos modèles de Machine Learning risquent de considérer systématiquement l'issue correspondant à cette cote extrême comme un ***Value Bet***. Pour ne pas perturber les performances de nos modèles de Machine Learning, nous devons réduire ces distributions en éliminant les valeurs extrêmes.

Nous avons choisi de ne conserver que les matchs pour lesquels les 3 cotes sont inférieures ou égales à 10. Voici les nouvelles distributions des variables *Max H*, *Max D* et *Max A* :







3.4.4. Statistiques du jeu vidéo FIFA

Afin de continuer à réduire le volume de notre jeu de données dans le but de pouvoir effectuer le Machine Learning dans des temps raisonnables, nous avons décidé de ne pas retenir l'ensemble des variables explicatives provenant du jeu vidéo FIFA (§3.2). En effet, certaines d'entre elles nous paraissaient trop spécifiques ou redondantes, et ainsi nous avons décidé de retenir uniquement celles qui nous avons jugé essentielles. Nous avons essayé de faire cette sélection à l'aide de techniques de réduction de dimensions, mais celles-ci se sont révélées infructueuses, c'est pourquoi nous avons opté pour une sélection manuelle des variables explicatives. Les variables que nous avons retenues sont les suivantes :

- Age (*Age*),
- Niveau général de l'équipe (*Overall*),
- Niveau général de l'équipe en considérant le potentiel des jeunes joueurs (*Potential*).
- Valeur sur le marché des transferts (*Value (€)*),
- Vitesse (*Pace*),
- Tir (*Shooting*),
- Passe (*Passing*),
- Dribble (*Dribbling*),
- Défense (*Defending*),
- Physique (*Physic*).

3.4.5. Jeu de données final

La dernière étape consiste à supprimer les variables descriptives que sont *Season*, *Division*, *Date*, *Home team* et *Away team* (§3.1) et nous obtenons notre jeu de données final, prêt pour le Machine Learning, constitué de 33524 lignes et 52 colonnes dont voici un aperçu des 5 dernières lignes (les 5 premières lignes contiennent beaucoup de 0 à cause des moyennes mobiles) :

	Full time goals scored (1 game)	Full time goals conceded (1 game)	Shots (1 game)	Shots on target (1 game)	Full time goals scored (home or away) (1 game)	Full time goals conceded (home or away) (1 game)	Shots (home or away) (1 game)	Shots on target (home or away) (1 game)	Full time goals scored (3 games)	Full time goals conceded (3 games)	Shots (3 games)	Shots on target (3 games)	Full time goals scored (home or away) (3 games)
33519	0.0	-1.0	4.0	1.0	0.0	-2.0	-1.0	1.0	0.000000	-1.333333	-5.333333	-0.666667	0.000000
33520	-1.0	1.0	3.0	0.0	0.0	-2.0	2.0	-1.0	0.000000	-0.333333	4.000000	0.666667	0.000000
33521	-2.0	1.0	-2.0	1.0	0.0	4.0	3.0	-2.0	-0.666667	1.666667	1.666667	0.666667	0.333333
33522	-2.0	-1.0	-4.0	1.0	-1.0	-3.0	1.0	-2.0	-1.333333	-1.333333	-4.666667	-1.333333	-0.666667
33523	0.0	-1.0	1.0	-3.0	3.0	0.0	0.0	0.0	1.000000	-0.333333	2.000000	0.000000	2.000000

Full time goals conceded (home or away) (3 games)	Shots (home or away) (3 games)	Shots on target (home or away) (3 games)	Full time goals scored (5 games)	Full time goals conceded (5 games)	Shots (5 games)	Shots on target (5 games)	Full time goals scored (home or away) (5 games)	Full time goals conceded (home or away) (5 games)	Shots (home or away) (5 games)	Shots on target (home or away) (5 games)	Full time goals scored (home or away) (20 games)	Full time goals conceded (home or away) (20 games)
-0.666667	2.000000	0.000000	0.2	-0.2	-1.6	0.0	0.0	-0.2	3.0	-0.2	0.15	-0.45
-1.333333	2.333333	0.000000	0.0	0.0	3.8	0.6	-0.4	-1.0	-0.4	0.0	0.05	-0.55
1.000000	2.000000	-0.666667	-1.0	1.2	1.4	-1.0	0.4	0.6	-0.6	-1.4	0.30	0.40
-1.000000	-2.333333	-2.666667	-1.2	-1.2	-3.4	-1.8	-1.0	-0.6	-1.8	-2.0	-0.15	-0.60
0.000000	1.666667	1.666667	0.4	-0.2	4.6	1.0	1.4	-1.2	2.6	2.0	0.70	-0.90

Shots (home or away) (20 games)	Shots on target (home or away) (20 games)	Full time goals scored (40 games)	Full time goals conceded (40 games)	Shots (40 games)	Shots on target (40 games)	Points (1 game)	Points (3 games)	Points (5 games)	Points (40 games)	Points (home or away) (1 game)	Points (home or away) (3 games)	Points (home or away) (5 games)
2.20	0.20	0.025	-0.125	0.775	0.300	0.0	0.000000	0.0	0.300	0.0	0.000000	0.0
1.00	0.55	0.025	-0.175	2.025	0.700	-2.0	-0.333333	-0.2	0.250	1.0	1.333333	0.4
0.15	0.25	-0.250	0.525	-1.225	-0.375	-3.0	-2.000000	-2.0	-0.625	-3.0	-0.333333	0.0
0.45	0.05	-0.525	-0.350	-1.625	-0.800	-2.0	0.000000	0.0	-0.125	3.0	0.666667	0.0
2.55	0.55	0.350	-0.300	3.100	0.775	0.0	0.666667	0.4	0.500	2.0	1.333333	1.4

Points (home or away) (20 games)	Cote	Age	Overall	Potential	Value (€)	Pace	Shooting	Passing	Dribbling	Defending	Physic	FTR
0.45	1.61	-1.086957	2.000000	3.260870	4.702174e+05	1.095238	1.000000	2.238095	3.666667	3.761905	0.333333	H
0.45	-2.94	-4.695652	-1.043478	2.869565	2.228261e+05	2.285714	-4.095238	-0.380952	0.857143	0.000000	-1.238095	A
-0.15	-3.94	-1.521739	-4.826087	-2.956522	-2.088043e+06	-2.714286	-5.857143	-2.380952	-3.619048	1.523810	-5.380952	D
0.40	2.13	1.260870	-1.043478	-2.000000	-9.150000e+05	-1.666667	3.142857	0.904762	-0.904762	-2.619048	3.190476	A
1.25	4.17	-0.841897	3.104743	3.454545	1.063162e+06	5.188095	3.178571	1.647619	0.914286	1.900000	1.738095	A

4. Machine Learning

Maintenant que nous avons choisi les variables explicatives de notre jeu de données, nous pouvons commencer la phase de Machine Learning. Nous allons traiter un problème de classification dans lequel notre variable cible est *FTR*. *FTR* est une variable catégorielle qui possède 3 valeurs possibles, H, D ou A (voir §3.1). Cependant, comme nous l'avons vu dans la partie des concepts théoriques (§2), notre objectif ne va pas être de prédire une classe mais plutôt la probabilité de chacune des classes afin de calculer nos propres cotes pour ensuite les comparer aux cotes des bookmakers et identifier les ***Value Bets***.

4.1. Prétraitement

La 1ère étape consiste à effectuer le prétraitement préalable nécessaire à tout projet de Machine Learning :

- Isolation de la variable cible *FTR*,
- Séparation de notre jeu de données en un jeu de données d'entrainement et jeu de données de test. Pour effectuer cette séparation, nous n'avons pas choisi d'utiliser une séparation aléatoire avec train_test_split par exemple, mais nous avons choisi d'effectuer cette séparation chronologiquement selon les saisons. En effet, notre jeu de données d'entrainement est constitué des matchs des saisons 2014-2015 à 2019-2020, et notre jeu de données de test des matchs de la saison 2020-2021. Notre jeu de données de test représente ainsi environ 18% du jeu de données initial. Nous avons fait ce choix car nous souhaitions que notre test s'apparente à si nous avions parié sur les matchs de la saison 2020-2021.
- Standardisation du jeu de données.

4.2. Choix de la métrique

Dans la plupart des problèmes de classification, la métrique que nous souhaitons maximiser est l'accuracy, c'est-à-dire le taux de bonnes prédictions. Dans le cas des paris sportifs, notre objectif est de maximiser les gains auxquels on peut prétendre en pariant. Cependant, même si cela parait contre-intuitif, il n'y a pas de corrélation directe entre le pourcentage de paris gagnants (accuracy) et le gain final. Si nous reprenons ce que nous avons détaillé dans les concepts théoriques (§2), nous avions démontré que l'espérance du gain s'écrivait :

$$E(Gain) = Mise \times (p \times Cote - 1)$$

Pour un ensemble de matchs pour lequel nous misons toujours la même somme, p devient l'accuracy et la cote devient la cote moyenne des paris gagnants. Ainsi :

$$E(Gain) = Mise \times (accuracy \times Cote_{moyenne} - 1)$$

Soit :

$$E(Gain) \geq 0 \leftrightarrow accuracy \times Cote_{moyenne} \geq 1$$

Il s'agit donc d'un problème à 2 inconnues, l'accuracy seule ne suffit pas à garantir des gains élevés.

Afin d'illustrer cela, nous avons préparé plusieurs cas simples sur notre jeu de données de test pour lesquels nous avons calculé les gains associés :

- Prédiction de H systématique : nous parions systématiquement pour la victoire de l'équipe à domicile,
- Prédiction de D systématique : nous parions systématiquement pour le match nul,
- Prédiction de A systématique : nous parions systématiquement pour la victoire de l'équipe à l'extérieur,
- Prédiction du favori systématique : nous parions systématiquement pour le résultat présentant la cote la plus faible,
- Prédiction de l'outsider systématique : nous parions systématiquement pour le résultat présentant la cote la plus élevée.

Voici un tableau récapitulatif des résultats que nous avons obtenus pour les 5898 matchs de notre jeu de données de test :

CAS	ACCURACY	GAIN MOYEN PAR MATCH (ROI)
H systématique	39,8%	-4,9%
D systématique	26,8%	-3,6%
A systématique	33,4%	-0,061%
Favori systématique	48,5%	-0,75%
Outsider systématique	23,8%	-3,3%

Ce tableau nous donne pour chaque cas simple l'accuracy obtenue, ainsi que le gain moyen par match, qui est analogue au retour sur investissement (ROI). Le ROI s'exprime en % de la mise.

Ces résultats illustrent bien l'absence de relation de proportionnalité et donc de corrélation directe entre accuracy et gain. Ce ne sont pas nécessairement les cas qui possèdent l'accuracy la plus élevée qui ont le ROI le plus élevé (et inversement). Tout dépend de la cote moyenne des paris gagnants.

Lorsqu'un algorithme de classification effectue sa prédiction, il choisit la classe pour laquelle la probabilité calculée est la plus élevée. Dans le domaine des paris sportifs, cela revient à choisir l'issue possédant la cote la plus faible. Nous n'aurions alors pas besoin de développer un algorithme pour faire cela, puisqu'il suffirait seulement de regarder les cotes. L'algorithme ferait en réalité le choix de parier pour le favori systématiquement, comme dans le cas que nous avons illustré plus haut. Si nous voulions maximiser l'accuracy, nous nous retrouverions probablement coincé autour d'une valeur maximale qui se situerait aux alentours de 48,5%, pourcentage des matchs remporté par le favori.

4.3. Fonction de gain

Comme expliqué précédemment, notre métrique ne sera donc pas l'accuracy mais le gain. Ne s'agissant pas d'une métrique que l'on peut retrouver dans les bibliothèques de Machine Learning, nous allons devoir la créer. La 1ère étape consiste donc à créer une fonction qui va calculer le gain de nos prédictions, dont voici les principales étapes :

- Récupération des cotes des bookmakers (cotes réelles) et des résultats des matchs pour les index correspondant à notre jeu de données de test,
- Calcul de nos propres cotes. Les cotes calculées correspondent à l'inverse des probabilités estimées par notre modèle de Machine Learning pour chacune des classes (§2),

- Recherche des ***Value Bets***. Nous avons vu au §2 qu'un ***Value Bet*** correspondait à une cote du bookmaker supérieure à la cote calculée par l'algorithme. Ainsi, pour chacune des 3 issues possibles d'un match (H, D ou A), nous calculons l'écart relatif entre la cote réelle et la cote calculée. Si tous les écarts sont négatifs, aucun pari sur ce match n'est un ***Value Bet***. Si au moins l'un des 3 écarts est positif, nous parions sur l'issue présentant le plus gros écart.

  Exemple :

  Pour un match, le bookmaker propose les cotes suivantes :
  - Cote H = 1.5
  - Cote D = 4
  - Cote A = 5

  Grâce à notre modèle de Machine Learning, nous calculons les cotes suivantes :
  - Cote H = 2
  - Cote D = 4.5
  - Cote A = 5.5

  Dans ce cas, les cotes des bookmakers sont toutes inférieures aux cotes que nous avons calculées, nous n'identifions donc pas de ***Value Bet***.

  En revanche, si nous calculons les cotes suivantes :
  - Cote H = 2
  - Cote D = 3.8
  - Cote A = 4.5

  Nous avons 2 issues pour lesquelles les cotes des bookmakers sont supérieures aux cotes que nous avons calculées, ce sont les issues D et A. Ce sont donc 2 ***Value Bets***. Cependant, l'écart relatif entre les cotes est plus important pour l'issue A que pour l'issue D (10% contre 5%), l'issue A est donc un ***Value Bet*** plus intéressant que l'issue D, nous parierons donc pour la victoire de l'équipe à l'extérieur.

  - Comparaison entre le résultat prédit et le résultat réel,
  - Calcul des gains pour chaque match,
  - Tri des ***Value Bets*** par ordre d'importance, du plus intéressant au moins intéressant,
  - Somme cumulative des gains.

La fonction retourne alors le maximum de cette somme cumulative.

La raison pour laquelle nous effectuons une somme cumulative et non une somme simple est que nous voulons observer l'évolution des gains selon la proportion des ***Value Bets*** les plus intéressants que l'on considère (les x % des ***Value Bets*** les plus intéressants, x variant entre 0 et 100), et pas seulement sur l'ensemble des ***Value Bets*** (x = 100). L'objectif par la suite sera d'essayer de trouver s'il n'y a pas une optimisation possible dans la sélection de nos ***Value Bets***.

Une fois la fonction créée, il suffit de définir notre nouvelle métrique à l'aide de la fonction ***make_scorer***, de la bibliothèque ***sklearn.metrics***.

4.4. Modélisations

Une fois le prétraitement réalisé et la métrique définie, nous sommes prêts pour développer nos modèles de Machine Learning.

Nous avons voulu tester différents modèles afin de comparer leurs performances. Les modèles que nous avons testés sont les suivants :

- Random Forest,
- SVC,
- KNN,
- XGBoost,
- Voting Classifier (en utilisant tous les modèles ci-dessus).

L'objectif était à la fois de tester des modèles plutôt simples tels que le SVC ou le KNN, mais aussi des modèles qui nécessitent des optimisations d'hyperparamètres plus complexes, comme le Random Forest Classifier ou le XGBoost. Enfin, nous voulions tester un Voting Classifier pour essayer d'obtenir un modèle qui serait un bon compromis entre tous les modèles testés.

Pour chaque modèle, nous avons suivi la même démarche :

- Optimisation des hyperparamètres selon la métrique de gain. Nous avons utilisé principalement ***GridSearchCV*** pour cela, mais nous avons aussi testé la méthode ***hyperopt***,
- Entrainement du modèle sur le jeu de données d'entrainement, avec le meilleur lot d'hyperparamètres,
- Tracé de courbes pour analyser les résultats du modèle sur le jeu de données de test.

4.5. Résultats

Voici un tableau récapitulatif des résultats que nous obtenons pour les différents algorithmes testés :

ALGORITHME	GAIN MAX	NOMBRE DE MATCHS	GAIN MOYEN PAR MATCH (ROI)
Random Forest (GridSearchCV)	19,25	24	80,21%
SVC	1	9	11,11%
KNN	56,72	1746	3,25%
XGBoost	19	18	105,56%
Random Forest (Hyperopt)	23,25	20	116,25%
Voting Classifier	15,7	39	40,26%

Le gain max correspond au maximum de la courbe de gain en fonction du % des meilleurs Value Bets lorsque nous considérons toutes les issues prédites (courbe rouge ci-après). C'est la valeur que retourne notre métrique de gain et que nous souhaitons maximiser. Elle n'a pas d'unité puisqu'il s'agit d'un facteur qu'il faut multiplier par la mise.

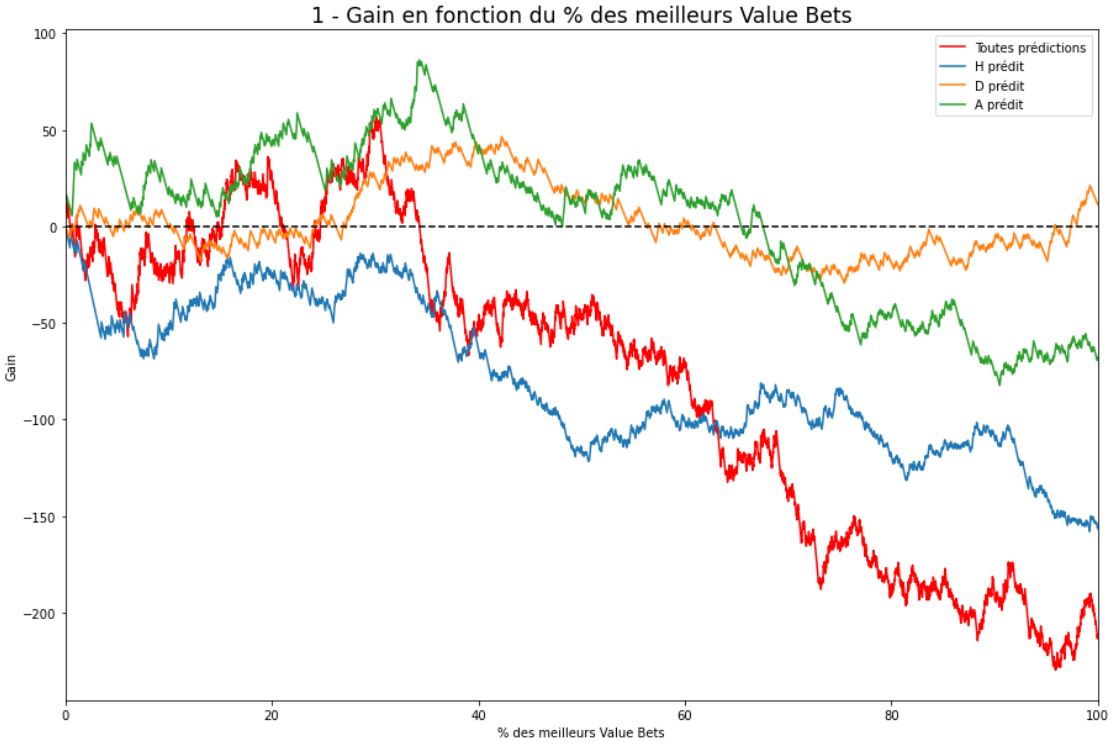
Nous pouvons observer dans nos résultats que hormis le KNN, les autres algorithmes atteignent le maximum très vite, pour des nombres de matchs très faibles (notre jeu de données de test contient 5898 matchs). Cela signifie que seule une très faible partie des meilleurs ***Value Bets*** nous donne de bons résultats. Cette situation n'est pas viable pour une stratégie de pari puisque nous ne parierions

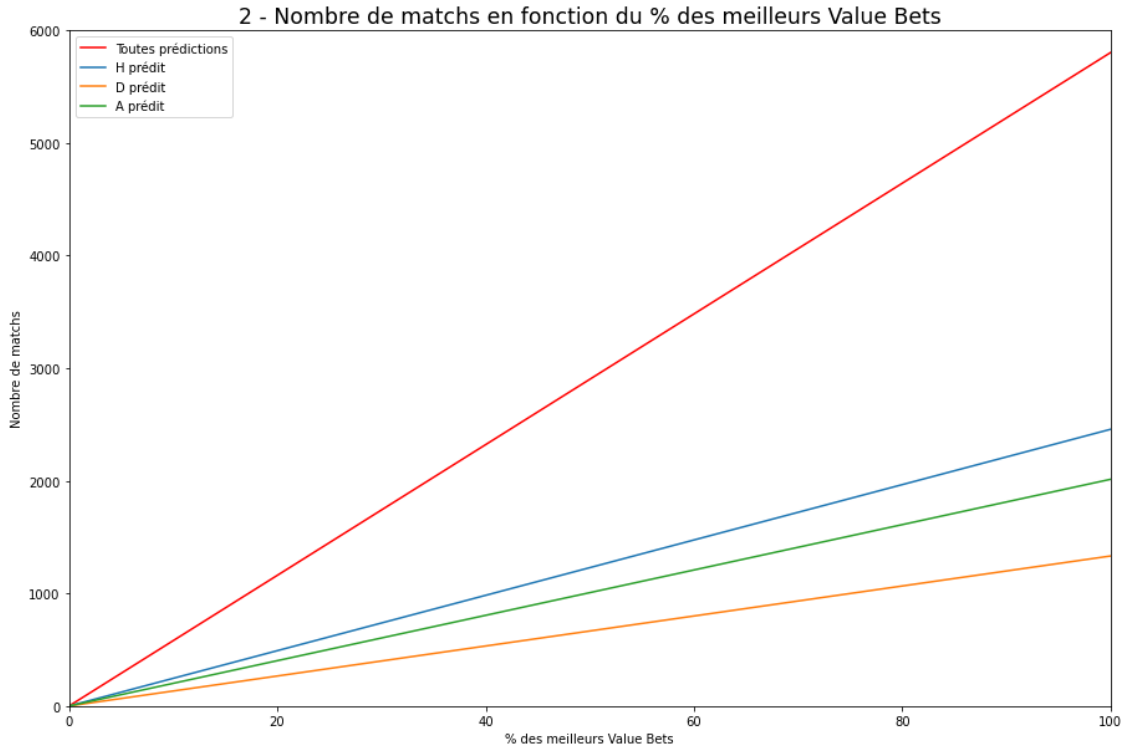
quasiment jamais. Ce que nous souhaiterions, c'est d'avoir un ROI le plus élevé possible, mais pour un nombre de matchs le plus élevé possible.

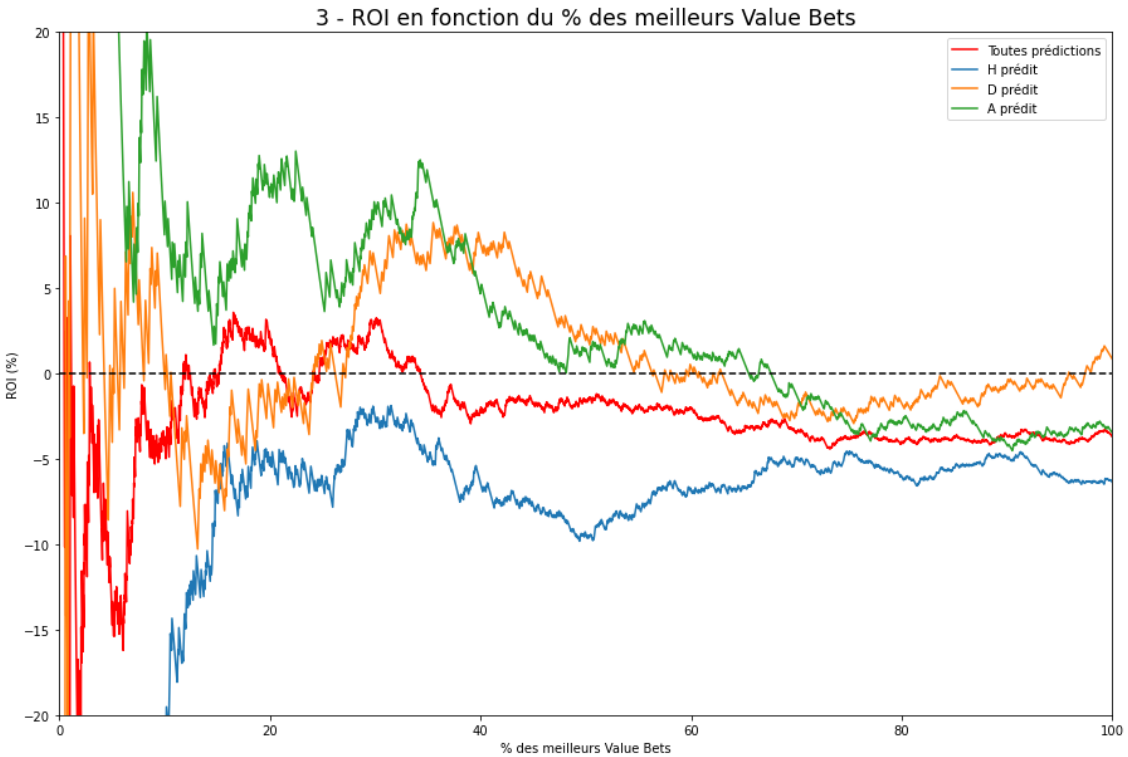
L'algorithme KNN est celui qui nous donne les meilleurs résultats, c'est lui qui possède le gain max le plus élevé. Son ROI est beaucoup plus faible que pour les autres algorithmes, mais il concerne 1746 matchs, soit près de 30% de notre jeu de données de test.

Par soucis de synthèse, nous ne présenterons pas ici l'intégralité des résultats obtenus, mais seulement ceux pour le modèle le plus performant, c'est-à-dire le KNN.

Pour visualiser nos résultats, nous traçons des graphes, que voici ci-dessous. Leur objectif est à la fois de présenter les résultats, mais aussi d'essayer de mettre en évidence si une sélection plus fine des ***Value Bets*** permettrait d'optimiser les gains.





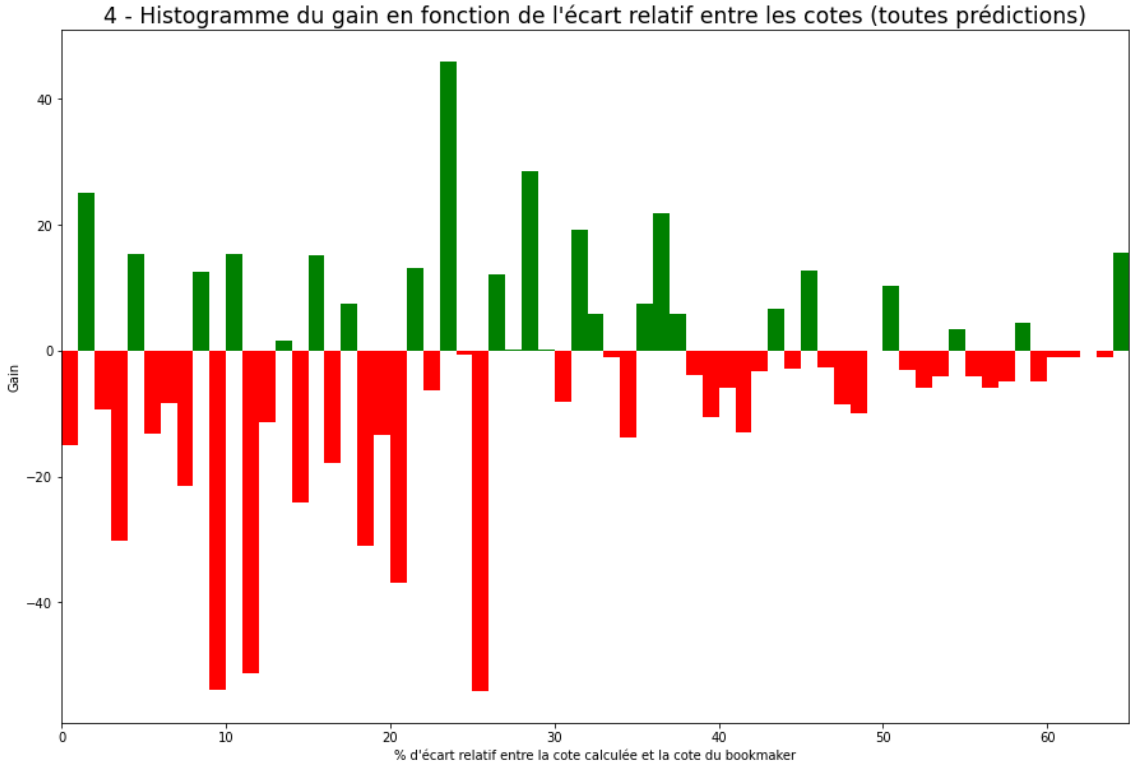


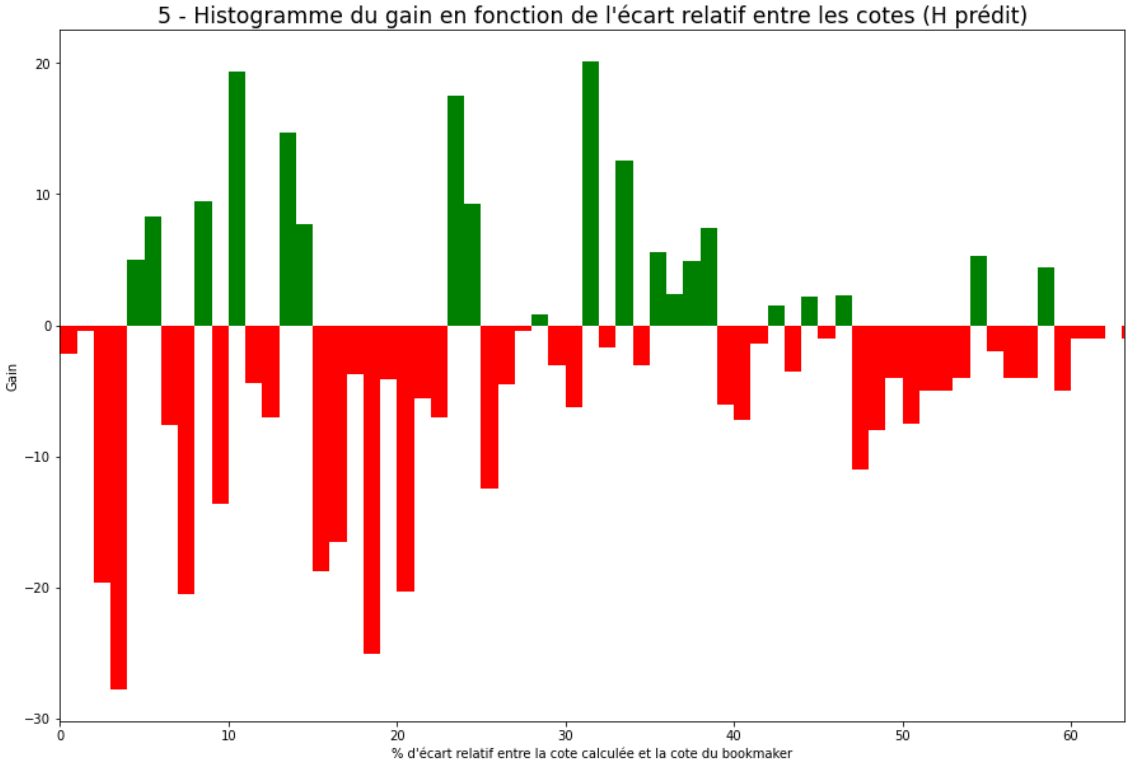
Le 1er graphe représente le gain en fonction du pourcentage des meilleurs ***Value Bets***. Après avoir trié les ***Value Bets*** du plus intéressant au moins intéressant, nous calculons, pour x entre 0 et 100, le gain pour les x % des ***Value Bets*** les plus intéressants. Ainsi, nous ne regardons pas uniquement les gains pour l'intégralité des ***Value Bets*** (x = 100), mais aussi pour une fraction d'entre eux. De plus, nous regardons aussi les gains en fonction de l'issue prédite.

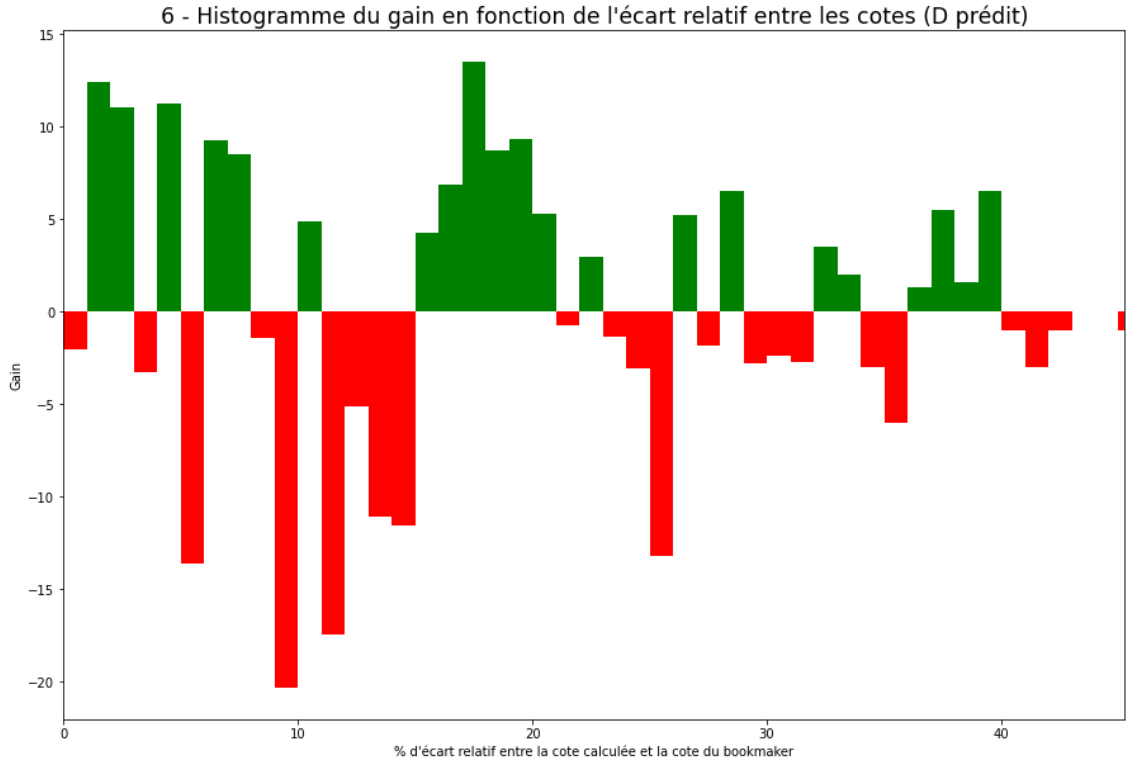
Le 2ème graphe permet de connaitre le nombre de matchs qui sont représentés par chaque x % des meilleurs ***Value Bets***.

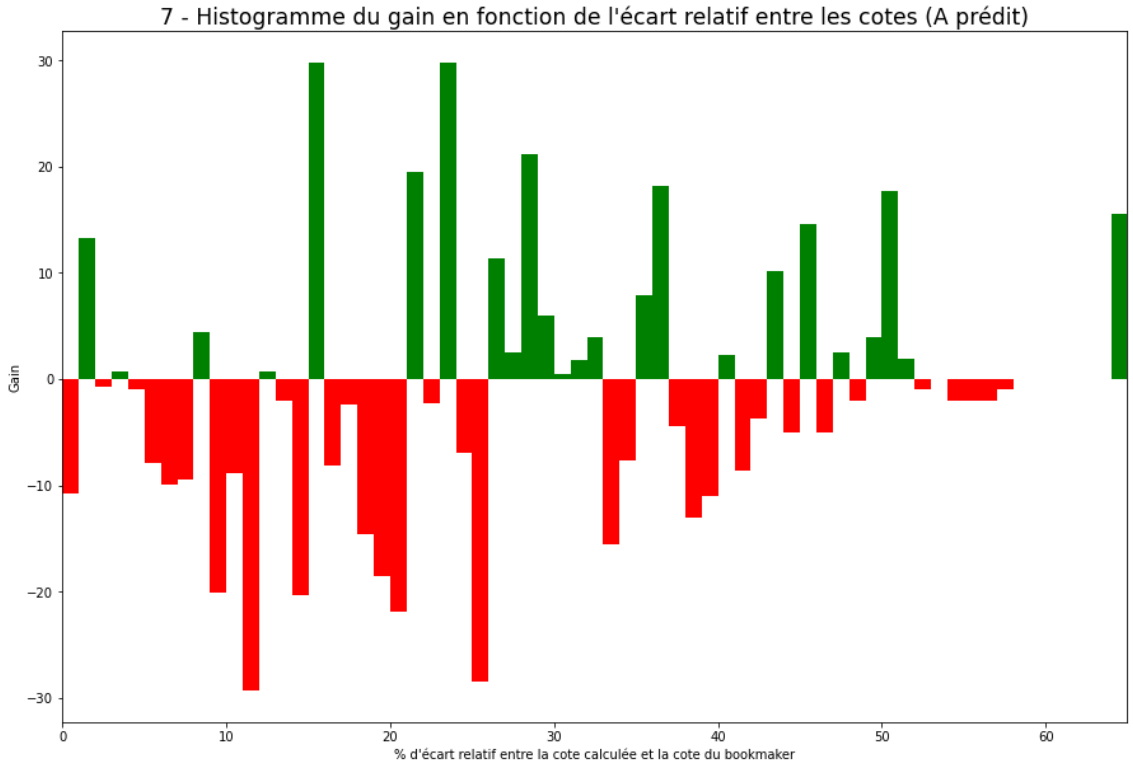
Enfin, le 3ème graphe représente le rapport entre le 1er et le 2ème graphe, c'est-à-dire le gain moyen par match (ROI).

Ce que nous pouvons observer au travers de ces graphes, et notamment le 1er, c'est que lorsque nous considérons l'ensemble des ***Value Bets*** (x = 100), les résultats sont assez mauvais puisque les gains sont largement négatifs. Cependant, si nous réduisons notre sélection de ***Value Bets*** (x < 100), il semble exister une zone pour laquelle on peut espérer obtenir des gains positifs. De plus, certaines issues semblent mieux s'en sortir que d'autres. Cela nous conforte dans l'idée qu'en effectuant une sélection intelligente de nos ***Value Bets***, il pourrait être possible d'obtenir de bons résultats. L'enjeu de la sélection des ***Value Bets*** est de pouvoir obtenir un ROI le plus élevé possible pour un nombre de matchs le plus élevé possible.




5 - Histogramme du gain en fonction de l'écart relatif entre les cotes (H prédit)
Gain
% d'écart relatif entre la cote calculée et la cote du bookmaker


6 - Histogramme du gain en fonction de l'écart relatif entre les cotes (D prédit)
Gain
% d'écart relatif entre la cote calculée et la cote du bookmaker



Les 4ème, 5ème, 6ème et 7ème graphes représentent l'histogramme du gain en fonction de l'écart relatif entre la cote du bookmaker et la cote que nous avons calculée. Précédemment, nous ordonnions les ***Value Bets*** par ordre d'importance et prenions les x premiers %, ce qui nous obligeait à considérer systématiquement ceux que l'algorithme a jugé comme étant les meilleurs d'entre eux. Or, cela peut poser problème si jamais notre algorithme n'est pas très précis pour certaines prédictions. En effet, un très grand écart de cotes peut signifier un ***Value Bet*** très intéressant mais peut aussi signifier une prédiction complètement ratée. Il est très difficile de faire la différence entre ces 2 situations.

Ici, il n'est donc plus question d'ordonner les ***Values Bets*** par ordre d'importance mais de regarder directement s'il pourrait y avoir des zones d'écarts de cotes pour lesquels on pourrait obtenir des gains intéressants.

Au vu des histogrammes, il est délicat d'apporter la moindre conclusion à ce sujet, sachant que nous avons beaucoup d'alternance de positif et de négatif. Nous ne pouvons pas vraiment identifier de zones avec des gains positifs qui représenteraient suffisamment de matchs pour être intéressantes.

Nous voyons donc toute la difficulté de la recherche des meilleurs ***Value Bets*** et de l'optimisation des gains. Pour que cela puisse fonctionner, il faut avoir obtenu au préalable un algorithme qui prédise de manière très précise les cotes, ce qui n'est pas forcément notre cas. En effet, nous pouvons observer sur les histogrammes que nous obtenons des écarts relatifs de cotes entre la cote calculée et la cote du bookmaker pouvant aller parfois jusqu'à près de 70%. Ces écarts très importants sont en réalité plutôt représentatifs d'une mauvaise évaluation de notre modèle que d'un ***Value Bet*** très intéressant. En effet, il est peu probable qu'un bookmaker, dont le métier est d'évaluer des cotes et qui possède des moyens bien supérieurs aux nôtres, ait pu se tromper à ce point dans l'évaluation de certaines cotes. Ainsi, il faut désormais se questionner sur les limites de notre projet et ce qu'il faudrait faire pour l'améliorer.

5. Perspectives d'amélioration du projet

Nous venons de voir que nos algorithmes étaient encore très perfectibles. Dans une perspective d'amélioration future du projet, de nombreux axes d'améliorations sont possibles.

Le 1er axe, le plus important, se situe au niveau des données. Nous avons récupéré les données de 17 championnats et 7 saisons différentes, ce qui n'est pas suffisant. Si nous voulons prédire les cotes aussi bien que le ferait un bookmaker, nous aurions besoin d'un historique de matchs beaucoup plus conséquent.
De plus, les statistiques que nous avions sur les matchs étaient assez basiques, et nous n'en avons exploité que peu finalement. Il aurait pu être pertinent d'avoir des statistiques plus complètes et variées.
Enfin, nous avons déterminé les statistiques sur les équipes grâce aux données du jeu vidéo FIFA, en faisant la moyenne sur l'ensemble des joueurs présents dans l'effectif de l'équipe durant la saison en question. Cependant, ces statistiques ne sont jamais mises à jour selon la réalité du match à venir. En effet, nous n'actualisons pas les statistiques selon la composition réelle de l'équipe ce jour-là, les éventuels blessés, bref, les joueurs qui jouent effectivement le match. Ce dernier point pourrait expliquer pourquoi nous obtenons parfois de tels écarts de cotes entre ce que nous prédisons et ce que donne le bookmaker.
Tout ce travail au niveau des données n'est pas forcément évident à mettre en œuvre. En effet, les données sont parfois difficilement disponibles. Aussi, plus nous en ajoutons et plus nous avons besoin de capacités de calcul importantes si nous voulons pouvoir exécuter nos calculs dans des temps raisonnables. Enfin, plus nous voulons être précis dans nos données et plus la complexité du problème s'en trouve impactée.

Le 2ème axe se situe au niveau du Machine Learning. Nous avons testé différents algorithmes, dans la mesure de nos connaissances, du temps imparti pour réaliser le projet et de nos moyens en termes d'exécution de calculs. Tous nos modèles sont évidemment perfectibles et nécessiteraient de continuer à se pencher sur leur optimisation.

Enfin, le football est peut-être le sport le plus difficile à prédire, puisque c'est un sport qui autorise les matchs nuls, et qui génère ainsi 3 issues possibles dans le résultat d'un match. De plus, nous avons vu toute la problématique que pose un sport collectif, en ce qui concerne le traitement des données lié à une équipe entière et non seulement un joueur unique. Ainsi, il pourrait être judicieux de tester notre démarche sur un sport individuel, qui ne laisse pas la possibilité d'avoir des matchs nuls.

6. Conclusion

Tout au long de notre parcours de formation chez DataScientest, nous avons pu développer nos compétences dans le domaine de la Data Science. Ce projet aura constitué notre 1er cas pratique dans la peau d'un Data Scientist, et nous aura ainsi permis de mener un problème de Machine Learning de A à Z, en mettant en pratique toutes les compétences apprises.

Au travers de ce rapport, nous avons présenté l'ensemble du travail que nous avons accompli. Nos recherches sur le sujet des paris sportifs, que nous avons présentées ici, nous ont amenés à développer notre propre démarche de résolution du sujet, avec la définition de notre propre métrique, qui est certes singulière de la manière dont se résout habituellement un problème de classification, mais qui nous a semblé plus pertinente pour répondre à notre problématique. L'objectif initial était de développer un algorithme capable de battre les bookmakers, et même si cet objectif n'est pas réellement atteint, nous avons soulevé de nombreuses perspectives d'amélioration qui pourraient permettre de s'en approcher un peu plus à l'avenir.