h e g

Haute école de gestion
Genève

Analyse et optimisation de l'expected goal: application au machine learning

Travail de Bachelor HES réalisé en vue de l'obtention du Bachelor par :

David Paulino

Conseillers au travail de Bachelor :

Pr Alexandros KALOUSIS

Dr Nils SCHÄTTI

Genève, le 22 juin 2023

Haute école de Gestion de Genève (HEG-GE)

Filière Informatique de gestion

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale

1 Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre de Bachelor of Science HES-SO en Informatique de gestion.

L'étudiant a envoyé ce document par email à l'adresse remise par son conseiller au travail de Bachelor pour analyse par le logiciel de détection de plagiat URKUND, selon la procédure détaillée à l'URL suivante : `https://www.urkund.com`

L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommendations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

"J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie."

Fait à Genève, le 22 juin 2023

David Paulino

2 Remerciements

J'aimerais tout d'abord remercier mon conseiller au travail de Bachelor, le docteur Nils Schätti pour son aide et ses conseils tout au long de ce travail. Ses remarques et son investissement ont été très utiles pour améliorer la qualité de cette thèse.

Je remercie également mon deuxième conseiller au travail de Bachelor, le professeur Alexandros Kalousis pour l'apport de ses idées et ses explications sur les concepts de machine learning.

Je remercie également mon ami Rayane Ammad ainsi que ma belle-soeur, Alessia Marques, pour leur relecture et leurs retours sur ce travail.

Finalement, je remercie ma famille pour leur soutien et leur patience tout au long de ma formation, spécialement mon père et ma mère.

3 Résumé

Ce travail a pour but d'analyser et d'optimiser l'expected goal (xG) dans le football. On cherche à comprendre comment est calculé l'expected goal, quelles sont ses variables les plus influentes et quels sont les modèles les plus performants pour établir un xG.

Ce travail passe par plusieurs phases :

— Une phase de recherche pour comprendre le fonctionnement du xG et les modèles existants
— Une phase de préparation des données pour les modèles
— Une phase de modélisation pour trouver le modèle le plus performant
— Une phase d'analyse des résultats

Liste des tableaux

Table des figures

Table des matières

4 Introduction

4.1 Introduction à la problématique

Lorsque les premiers sports sont apparus, l'information la plus importante était le score et le vainqueur de la confrontation. Au fur et à mesure, plus d'informations sur les matchs sont venues s'ajoutées. Le nombre de tirs dans un matchs par équipes, le nombre de passes, le nombre de tirs cadrés, la possession du ballon, le pourcentage de passes réussies, le nombre de passes décisives et d'autres sont venus s'ajouter aux statistiques dans le football. Le pourcentage de réussite aux lancers francs, le nombre de rebonds, le nombre de passes décisives, le pourcentage de réussite aux tirs, le nombre de fautes, le nombre de minutes jouées, le pourcentage de réussite à 3 points et d'autres sont venus s'ajouter aux statistiques dans le basketball. Ces statistiques sont également devenus personnelles à chacun des joueurs. On peut également compter pour le baseball le nombre de fois qu'un joueur était au bâton, son nombre de double, de triples, son nombre de buts et bien d'autres.

C'est d'ailleurs dans le baseball que l'on peut retrouver la première utilisation de statistiques avancées pour établir des stratégies. En effet, au début des années 1970, un joueur des Baltimore Orioles a développé une analyse statistiques pour choisir le meilleur alignement possible pour son équipe de départ. Il s'agit de Davey Johnson. Cependant, il n'a pas pu l'utiliser à ce moment-là puisque le président de sa franchise n'avait pas confiance. C'est qu'à partir de 1984 où il fut le coach des New York Mets qu'il a pu mettre en place son analyse statistiques avancées pour établir le meilleur choix pour son équipe de départ. [11] Deux saisons plus tard, il remporte la Série mondiale 1986 [1]. Les Mets étaient situés à la dernière place de leur conférence avant l'arrivée de Davey Johnson et son management orienté sur les statistiques.
Après cette réussite, les autres franchises de la MLB [2] ont également commencé à adopter l'analyse de statistiques dans le sport et cela a également été populaire dans les autres sports avec par exemple Daryl Morey qui a été le premier coach analyste statistiques recruté chez les Rockets de Houston en NBA en 2007. [1] Les franchises de la NBA [3] ont par la suite également

1. En MLB, la Série mondiale est la série finale qui permet de déterminer qui est l'équipe championne de la ligue.
2. Ligue majeure de baseball
3. Ligue nationale de basketball

adopté une approche managériale statistique. On constate alors que cette culture de la statistique dans le sport provient des États-Unis.
Il est désormais important d'amener l'arrivée des expected goals. L'une des premières études sur un modèle d'expected goals vient d'Alan Ryder qui a publié une étude sur la qualité des tirs effectuées dans des matchs de hockey. [15] Ce dernier a pu analyser les différentes circonstances lors d'un tir et développer un modèle qui prédit la probabilité d'un tir selon les circonstances de ce tir. Dans le football, l'une des premieres études sur l'expected goal vient de Richard Pollard, Jake Ensum et Samuel Taylor qui ont analysé les facteurs qui influent la chance de marquer un but. [14] La problématique de ce travail est donc de pouvoir analyser et optimiser l'expected goal.

Il semble maintenant important de savoir ce qu'est l'expected goal. L'expected goal [4] est une métrique qui permet de déterminer la probabilité qu'un tir soit transformé en but selon les données de ce tir [7]. Un tir qui a un xG de 0.4 a une probabilité de 40% d'être transformé en but. Un tir avec un xG à 1 est la plus grande valeur possible et aurait donc 100% de chance d'être transformé en but. [5] [13]

Pour observer ce qu'est réellement un xG, nous allons l'observer avec l'emplacement des tirs sur le terrains. Sur la figure 1, on peut observer un example d'emplacement de tirs et de leur xG.

Par ailleurs, les xG se sont tellement développés que des métriques dérivées ont été créées. On peut par exemple citer le xA qui est l'expected assist. C'est une métrique qui permet de déterminer la probabilité qu'une passe soit transformée en passe décisive selon les données de cette passe. [7] Également, les xGA qui est les expected goals against. Cette métrique permet de déterminer la probabilité qu'un tir soit transformé en but selon les données de ce tir mais pour l'équipe adverse. [13]
Il y en a également d'autres comme les expected points qui sont les points qu'une équipe devrait avoir gagnés basé sur les données relatives aux xG. D'autres dérivées sont indiquées sur l'article de Pinnacle écrit par Luke Petty. [13]

4. Très souvent réduit par xG
5. Il est important d'indiquer qu'il est très rare qu'un xG d'un tir soit égal à 1 mais il va généralement s'en rapprocher fortement selon ses paramètres.

Maintenant que nous avons vu ce qu'est un xG et ces dérivées actuelles, il semble pertinent de décrire l'utilisation de cette métrique dans le football actuel.

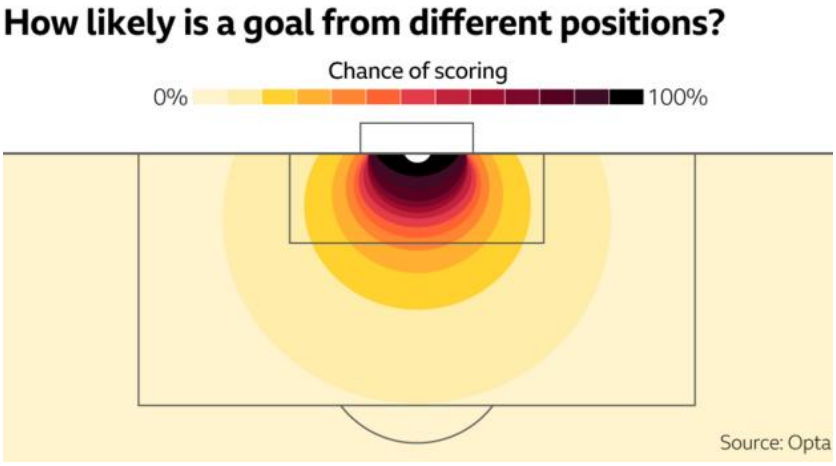

FIGURE 1 – Schéma qui montre les chances de buts selon les positions des tirs - Source : Opta

4.2 Intérêt de la problématique

Cette problématique est très intéressante puisque c'est une donnée qui est dernièrement très populaire dans le monde du football. Elle donne plus d'informations sur le match que les autres statistiques d'un matchs (possession, nombre de passes, nombre de tirs cadrés, nombre de buts, etc.). Par exemple en ce qui concerne la possession, une équipe peut avoir la possession du ballon pendant 70% du match mais ne pas marquer de buts, ni même être dangereuse avec le ballon. Le nombre de tirs cadrés, s'ils sont effectuées tous à l'extérieur de la surface de réparation, ne sont pas forcément dangereux. L'expected goal permet de donner une valeur à chaque tir et de déterminer si un tir a une chance d'être transformé en but.

Certaines personnes utilisent cette métrique pour leurs paris sportifs. Ils observent la métrique lors des derniers matchs et la compare avec le score réel du match. Si la différence est grande, cela peut permettre de constater un

manque de réalisme ou un sur-régime d'une des deux équipes. [17]

Les analystes de données des équipes utilisent cette donnée pour analyser les performances de leurs joueurs et de leurs équipes. Par exemple, la comparaison des xG et du nombre de buts marqués d'un joueur sur une période donnée peut permettre de déterminer la dangerosité et la capacité d'un buteur à terminer des actions. [13] Les xG peuvent aussi être utilisés pour analyser les performances d'une équipe dans des situations bien précises. Une équipe qui possède un haut xG sur des contre-attaques montre qu'elle est très dangereuse sur ce type de situation. [7] L'une des choses les plus intéressante de cette métrique est qu'elle peut être utilisée pour analyser les forces et faiblesses des équipes. Par exemple, une équipe peut observer que ses xG sont très faibles lorsqu'elle tente des centres. Cela peut lui permettre de trouver son identité de jeu et de décider d'abandonner cette stratégie. Cette métrique est tout autant valable pour analyser les forces et faiblesses d'une équipe adverse. Dans la même situation, si une équipe vient à affronter l'équipe qui a dû mal à produire des xG sur des centres, elle peut décider de la forcer à jouer sur des centres en libérant de l'espace sur les côtés par exemple.

Cette métrique est également utilisée pour aider les recruteurs à juger les performances de finition d'un joueur [10]. Nous avons vu précédemment qu'il existait des dérivées des xG comme les xA [6]. Les différentes dérivées peuvent permettre de juger les performances et qualités d'un joueur qui se trouve dans un poste bien précis. Par exemple, un milieu de terrain ou un défenseur peut être jugé sur ses xA et un attaquant sur ses xG.

4.3 Questions que l'on souhaite répondre dans ce travail

Il y a deux grandes questions à répondre dans ce travail.

- Quels sont les paramètres qui influencent le plus l'expected goal ?
- Quel est le meilleur modèle pour prédire l'expected goal ?

Le but de ce travail est de comprendre quelles sont les variables qui influences le plus les xG. Cela permettra de comprendre quelles sont les variables les plus importantes pour prédire ces estimations de buts. Grâce à cela, il est possible pour un analyste de données d'une équipe de football de savoir quelles sont les facteurs qui influencent le plus la qualité d'un tir. Cela peut permettre de

6. Passes décisives attendues

déterminer les forces et faiblesses d'une équipe et de savoir sur quels aspects travailler pour améliorer les performances de l'équipe.

Le deuxième objectif de ce travail est de trouver le meilleur modèle pour prédire les xG. En effet, le but sera d'avoir le modèle le plus performant et de comparer les différents modèles qui peuvent être utilisée pour établir cette métrique. Il faut également veiller à ce que le modèle ne fasse pas de sur-apprentissage. Il est donc important de faire attention à la complexité du modèle et de trouver le meilleur compromis entre la complexité et la performance du modèle.

5 Plan du document

Concernant le déroulement de ce travail, il y a plusieurs étapes. Tout d'abord, il y a la recherche des travaux existants sur le sujet. Cette étape est effectuée dans la section 6. Parmi les travaux existants, je vais chercher à savoir quels datasets ont été utilisées ainsi que les résultats obtenus. Cela me permettra de comparer mes résultats avec les résultats obtenus dans les autres travaux et ainsi savoir si les résultats sont cohérents par rapport à l'état de l'art actuel des recherches scientifiques.

La suite sera de trouver un dataset avec les informations nécessaires pour implémenter le modèle. Cette étape de la thèse est très importante puisque c'est la base de la thèse. Une fois ce dataset trouvé, il va falloir le documenter. En effet, il est important de comprendre ce que chaque attribut représente. L'étape suivante est également importante puisque le but sera de visualiser les données. Cela permettra de voir si les données sont exploitables et si elles sont cohérentes. Cela pourra également nous indiquer si des biais seront présents dans le modèle. Cette section d'analyse du dataset est trouvable dans la section 7.

Une fois que les données sont documentées et visualisées, il va falloir les préparer. En effet, il pourrait y avoir des données manquantes mais qui sont disponibles après un traitement. Il pourrait également y avoir des données qui ne sont pas exploitables et qui doivent être supprimées. Par exemple, l'identifiant de la base de données d'un tir pourrait être supprimée car il n'appporte rien pour la prédiction de l'expected goal. Cette section de transformation du dataset est également trouvable dans la section 7 qui fait suite à la section d'analyse du dataset initial.

Ensuite, on pourra commencer à implémenter le modèle et observer les facteurs qui influencent le plus sa prédiction du xG. Cette partie est trouvable dans la section 8. C'est également à ce moment-là qu'il faudra comparer les différents modèles pour voir lequel est le plus performant pour prédire les xG. La comparaison est trouvable dans la section 9 qui visent à montrer la fiabilité et la cohérence du modèle par rapport à la problématique initial.

6 Synthèse des travaux existants

Le premier travail repertorié sur les xG et la qualité d'un tir est celui de Richard Pollard, Jake Ensum et Samuel Taylor [14]. Dans ce travail datant de 2004, la seule information indiquée concernant le dataset est que les données proviennent de la Coupe du monde 1986 et de celle de 2002. Le modèle a été implémenté en utilisant une régression logistique. Le nombre de tirs répertoriées dans ce travail est de 1096. La conclusion de ce travail est que les 3 facteurs les plus influents pour la prédiction des xG sont :

— La distance entre le tireur et le but
— L'angle du but en fonction de la position du tir
— L'espace entre le tireur et le défenseur le plus proche

Le résultat final de l'analyse de la régression logistique de ce travail ressemble à cela.

Predictor	**Coefficient**	**z**	**p**	**ratio**	**Lower**	**Upper**
Constant	0.3771	1.20	0.229			
Distance	-0.1586	-9.51	0.000	0.85	0.83	0.88
Angle	-0.0222	-3.81	0.000	0.98	0.97	0.99
Space	0.7991	3.22	0.001	2.22	1.37	3.62

TABLE 1 – Résultat de la régression logistique du travail de Pollard, Ensum et Taylor

Le deuxième travail est celui de Izzatul Umami, Deden Hardan Gutama et Heliza Rahmania Hatta [18]. Ce travail utilise les données de Wyscout des 5 championnats majeurs en Europe de la saison 2019-20. Ces derniers ont décidés de prendre comme données :

— La distance
— L'angle
— Si le tir est un tir de la tête ou pas

Dans le dataset, 32000 tirs ont été utilisées pour la création du modèle. Comme pour le travail précédent, la régression logistique a été utilisée. Il est également indiqué qu'une séparation du dataset a été faite pour avoir des données d'entraînement et de tests. Dans leurs tests du modèle, il est indiqué que le but est de faire de la classification pour de futures instances, il faut donc utiliser un seuil. Suite à l'utilisation d'un seuil pour la classification, une matrice de confusion a été faite pour ensuite calculer la spécificité et

la sensibilité du modèle. Ce principe de sensibilité et spécificité permet de choisir la meilleure performance selon le contexte d'utilisation du modèle.

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative} \tag{1}$$

$$Specificity = \frac{TrueNegative}{TrueNegative + FalsePositive} \tag{2}$$

La sensibilité permet de voir la capacité du modèle à prédire correctement les tirs qui sont des buts. De l'autre côté, la spécificité permet de voir la capacité du modèle à prédire correctement les tirs qui ne sont pas des buts. Comme indiqué dans le travail de Umami, Gutama et Hatta, la spécificité est plus importante que la sensibilité selon le contexte. Dans le cas où le modèle est utilisé pour prédire un cancer, nous allons chercher à avoir une meilleur sensibilité pour éviter de passer à côté d'un cancer. [18] Cela leur permet de savoir comment choisir le seuil pour la classification. En conclusion de leur travail, ils ont obtenu une sensibilité de 0.9671945701357466 et une spécificité de 0.19034406215316316 pour un seuil 0.02. Ils indiquent finalement que le modèle de xG est plus performant si l'on prend la distance et l'angle en compte plutôt que de prendre uniquement la distance.

Un travail qui fournit également du code est celui de David Sumpter [16]. Le travail de ce dernier est de créer un modèle de xG en utilisant une régression logistique. Il utilise les données de Wyscout du championnat anglais de la saison 2017-18. Ce dernier explique étape par étape ce qui est effectué pour créer et améliorer son modèle. Il y a également des pistes pour convertir les positions X et Y en distance et en angle. Il commence par créer un modèle de xG en utilisant uniquement la distance. Par la suite, il le fait uniquement avec l'angle. Ensuite, il utilise de multiples facteurs, comme la distance au carré ou encore l'angle multiplié par la position X du tir, pour créer son modèle et il produit un résumé de la régression logistique.
Il n'y a pas moyen de connaître les coefficients uniquement pour la distance et l'angle car aucun résumé n'est fait pour ces deux facteurs uniquement.

Le dernier travail est fait par H.P.H Eggels. Le but est d'expliquer le résultat d'un match en utilisant les xG. Dans ce travail, il utilise un modèle de xG pour prédire le résultat d'un match. Les données du travail proviennent de ORTEC, de Immotio et FIFA. En effet, son travail utilise trois datasets

Predictor	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-0.5103	0.887	-0.576	0.565	-2.248	1.228
Angle	-0.6338	0.319	-1.989	0.047	-1.258	-0.009
Distance	0.2798	0.118	2.381	0.017	0.049	0.510
X	-0.1243	0.124	-1.001	0.317	-0.368	0.119
C	0.0300	0.040	0.750	0.454	-0.048	0.109
X2	-0.0014	0.001	-1.422	0.155	-0.003	0.001
C2	-0.0041	0.003	-1.398	0.162	-0.010	0.002
AX	0.1251	0.118	1.063	0.288	-0.105	0.356

TABLE 2 – Résumé de la régression logistique du modèle de xG de David Sumpter

pour créer son modèle de xG. [8]. Il y a eu un travail de "merging" des données pour avoir un dataset qui contient toutes les informations nécessaires pour créer le modèle. Cependant, les datasets peuvent avoir des problèmes entre eux lors du "merging". Par exemple, les noms des joueurs qui sont différents (majuscule, accent, encodage, surnom, etc.) a été un problème mais une solution a été trouvée.
Le dataset utilisé contient donc trois sources de données différentes.

ORTEC	**FIFA**	**Immotio**
Context	Player quality	Number of attackers in line
Part of body	Goal keeper quality	Number of defenders in line
Dist to goal		Distance nearest defender in line
Angle to goal		Distance goal keeper
Originates from		
Current score		
High		

TABLE 3 – Sources de données utilisées par H.P.H Eggels

Parmi les modèles testés, ce travail utilise :

- Un modèle de régression logistique
- Random Forest
- Un arbre de décision
- Ada-boost

Pour chacun de ces modèles, il y a une liste des différents paramètres qui vont être utilisés pour trouver le meilleur modèle. Chacun des modèles a suivi une procédure avec un set d'entraînement, un set de validation et un set de test. Le set de validation permet de trouver les meilleurs paramètres pour le modèle. Le set de test permet de tester le modèle avec les paramètres trouvés et le comparer avec les autres modèles Le modèle le plus performant parmi les quatres cités est le Random Forest avec une précision de 0.771. Cependant, il n'est pas possible de connaître les meilleurs hyper paramètres pour chacun des modèles. Ce travail donne tout de même de bonnes pistes pour l'ajout de nouvelle données pour améliorer le modèle de xG.

6.1 Récapitulatif des travaux existants

Auteur(s)	**Source de données**	**Conclusion**
Richard Pollard Jake Ensum Samuel Taylor	Coupe du monde 1986 et 2002	La distance, l'angle et l'espace avec le joueur le plus proche sont les variables qui ont le plus d'influences sur l'estimation de but
Izzatul Umami Deden Hardan Gautama Heliza Rahmania Hatta	Wyscout, 5 championnats majeuren Europe. Saison 2019-20	La distance et l'angle apporte plus d'informations qu'uniquement la distance
David Sumpter	Wyscout, Premier League. Saison 2017-18	La distance et l'angle sont les facteurs avec le plus d'influence
H. P. H Eggels	ORTEC, Immotio, FIFA	Le but du travail est de prédire les résultats des matchs. Rien n'indique les facteurs les plus influents

Table 4 – Récapitulatif des travaux existants

On observe donc que la majorité des travaux existants utilisent la distance et l'angle pour estimer les xG. C'est essentiellement ces deux facteurs qui sont utilisés pour établir la prédiction. Concernant les autres facteurs, ils sont utilisés pour améliorer la prédiction mais rien n'indique s'ils ont autant d'influence que la distance et l'angle.

7 Dataset

7.1 Présentation du dataset

Le dataset qui a été trouvé pour cette thèse est un dataset provenant de Wyscout. Wyscout est un site web qui fournit des données sur le football. On peut par exemple trouver des données sur les joueurs, les équipes, les matchs, les événements d'un match comme les passes, les tirs, etc. Ces données sont collectées grâce à une équipe d'experts nommés "opérateur" qui utilisent un logiciel propriétaire nommé "marqueur" pour décrire les événements. Pour garantir la précision des données, la description des événements d'un match est faite par une équipe de trois opérateurs. Deux d'entre eux sont attribués aux deux équipes et le troisième a le rôle de superviseur et de responsable des données sortantes du match. [12] Il peut y avoir parfois un quatrième opérateur qui a le rôle d'accélérer le processus de description d'événements d'un match.

"La description des événements d'un match possède trois étapes. La première étape consiste à décrire la formation de départ des équipes. Un opérateur décrit la formation de départ, la position des joueurs et le numéro de maillot de chaque joueur. Les joueurs remplaçants sont aussi indiqués. La deuxième étape consiste à décrire les événements d'un match. Pour chaque touche de balle dans le match, un opérateur décrit l'événement. Il sélectionne un joueur et crée un événement sur la timeline du match. Il indique ensuite de quel type d'événement il s'agit (tir, passe, duel) et quel est le sous-type d'événement (duel aérien ou au sol). Il entre ensuite la position de l'événement sur le terrain et les informations supplémentaires. Finalement, la troisième étape est un contrôle qualité. Ce contrôle possède deux phases. La première est automatique et faite par le marqueur [7] qui se charge d'éviter la majorité des erreurs des opérateurs. Par exemple, il vérifie que la position des duels spécifiés par les deux opérateurs soit corrects et qu'ils possèdent les mêmes informations. Le marqueur se charge également de proposer des événements manquants et vérifie si des combinaisons de données d'événements impossibles ont été inscrites. Par exemple, un événement de type tir qui aurait un sous-événement duel aérien serait une combinaison impossible. La deuxième phase de contrôle qualité est manuelle est gérée par des contrôleurs qualités

7. Le logiciel propriétaire

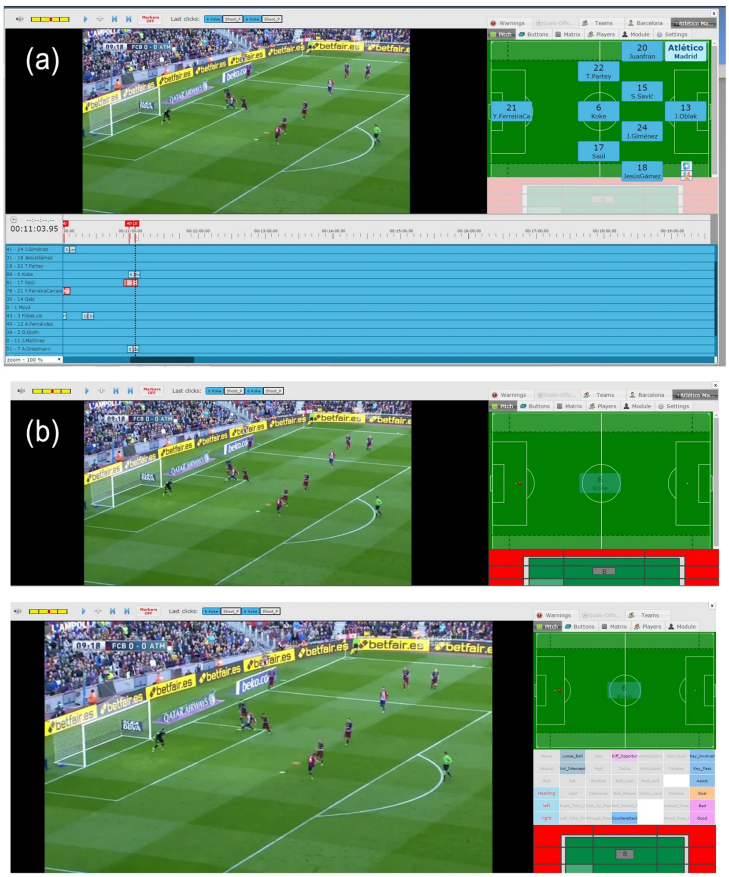

FIGURE 2 – Aperçu du logiciel propriétaire de Wyscout

qui vont vérifier de manière approfondie les événements de certains matchs et les corriger si besoin." [12]
Il se peut que certains matchs n'aient pas subi de contrôle qualité car ils utilisent un algorithme de sélection de matchs pour le contrôle qualité. Sur la figure 2, on peut voir un aperçu du logiciel propriétaire de Wyscout. La partie (a) montre la timeline du match avec les événements qui ont été ajoutés. La partie (b) montre les informations d'un événement ainsi que l'indication de la position de l'événement sur le terrain.

Le dataset utilisé est un sample du dataset de Wyscout. Il a été mis à disposition par Luca Pappalardo et Emanuele Massuco dans leur article "A public data set of spatio-temporal match events in soccer competitions" [12].

Ce dataset est composé des 5 championnats européens majeurs (Premier League, La Liga, Bundesliga, Serie A et Ligue 1) de la saison 2017-2018. Le dataset contient également les informations de la Coupe du Monde 2018 et de l'Euro 2016. Les données sont au format JSON qui est un format très récurrent dans le monde du web. Il est également utilisé lors de communication avec des API REST.

7.2 Events

Le dataset fournit par Luca Pappalardo et Emanuele Massuco contient des données sur les événements d'un match. Ces événements sont décrits par des attributs qui sont détaillés dans le tableau 5.
Parmi les informations fournies par le dataset, il y a "eventName". Cet attribut permet de savoir quel type d'événement a eu lieu. Parmi les types d'événements [8], il y a :

- Passe
- Faute
- Tir
- Duel
- Coup franc
- Hors-jeu
- Touche de balle

L'attribut "tags" est également important. Cette attribut est une liste de tags qui permettent d'apporter des informations supplémentaires sur l'événement. La documentation des tags peut être trouvée sur le glossaire de Wyscout. [6] Par exemple, le tag 101 indique que l'événement est un but. Le tag 401 indique que le tir a été fait du pied gauche, le tag 402 indique que le tir a été fait du pied droit. Finalement, le tag 403 indique que le tir a été fait de la tête.

Le dernier point important est la localisation de l'événement. En effet, comme indiqué dans la section "Pitch coordinates" du glossaire de Wyscout, [6] les coordonnées X et Y sont relatives à la taille du terrain. Il y a un aperçu de la taille du terrain sur la figure avec ses coordonnées sur la figure 3. Cela donne une idée de la taille du terrain. Le choix de la taille du terrain en 100 par 100 a été fait car les terrains n'ont pas tous la même taille malgré les règles de l'IFAB.

8. Le nom des événements est en anglais mais a été traduit pour les lister

Nom de l'attribut	**Description**
eventId	L'identifiant du type d'événement. Chaque identifiant d'événement est lié à un nom d'événement
eventName	Le nom du type d'événement.
subEventId	L'identifiant du sous-type d'événement
subEventName	Le nom du sous-événement
tags	Une liste d'événement de tag. Chaque tag permet d'apporter une information
eventSec	Le temps (en secondes) écoulé depuis la période actuelle du match
id	L'identifiant unique de l'événement
matchId	L'identifiant du match auquel l'événement est lié.
matchPeriod	La période du match à laquelle l'événement a lieu.
playerId	L'identifiant du joueur qui a généré l'événement. Est lié à l'attribut "wyId" du dataset "Players"
positions	La localisation en X et Y de l'événement
teamId	L'identifiant de l'équipe du joueur qui a généré l'événement

Table 5 – Description des attributs du dataset "Events"

L'IFAB est l'International Football Association Board qui est l'organisme qui définit les règles du football. C'est cet organisme qui définit donc la taille des terrains de football. En effet, les terrains de football ont une taille qui peut varier entre 90 et 120 mètres de longueur et entre 45 et 90 mètres de largeur. [4]

Une autre information importante qui est indiqué dans la documentation de l'API de Wyscout [5] est que les positions des événements ont été normalisées par rapport à l'emplacement du but de l'équipe qui a généré l'événement. Cela veut dire que dans les données fournies par le dataset, les équipes jouent toujours dans le même sens du terrain. On peut d'ailleurs constater cela lorsqu'on regarde les positions des événements de tirs 4, en effet sur la figure 4, on peut voir que les tirs sont toujours effectués du même côté du terrain [9]. Il est également possible de visualiser la distribution des positions X des tirs sur la figure 5. On voit bien que les tirs sont toujours effectués dans la même partie du terrain. Le fait que les tirs sont relatives à la position du but de l'équipe qui a généré l'événement règle déjà un problème qui aurait pu se poser. En effet, il aurait fallu déterminer quelle équipe joue dans quel sens du terrain et donc savoir dans quelle direction les tirs sont effectuées. Ce travail aurait été nécessaire et extrêmement important car cela aurait pu fausser les données sur la distance et l'angle de tir.

On sait donc que, dans le dataset, les équipes jouent toutes dans le même sens du terrain, Cela nous simplifie la tâche pour la suite de la thèse.

9. Certains sont effectuées dans l'autre partie du terrain mais jamais dans la surface de réparation opposée.

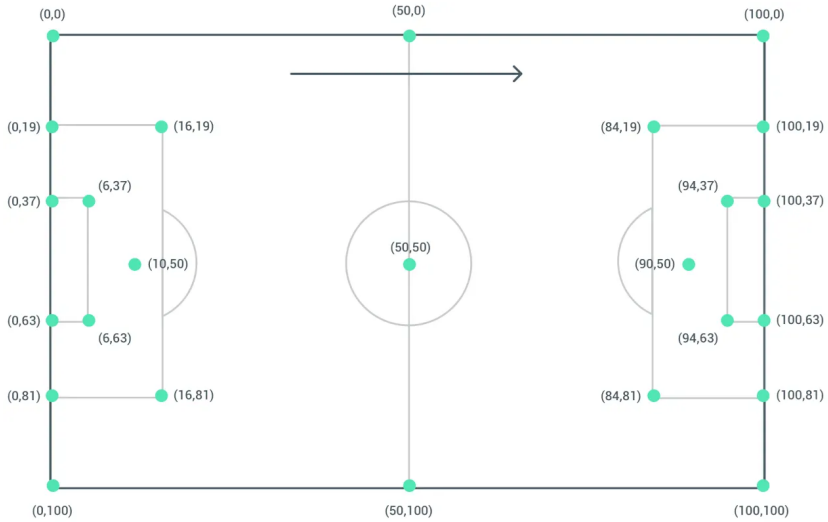

FIGURE 3 – Représentation du terrain de football avec les coordonnées X et Y

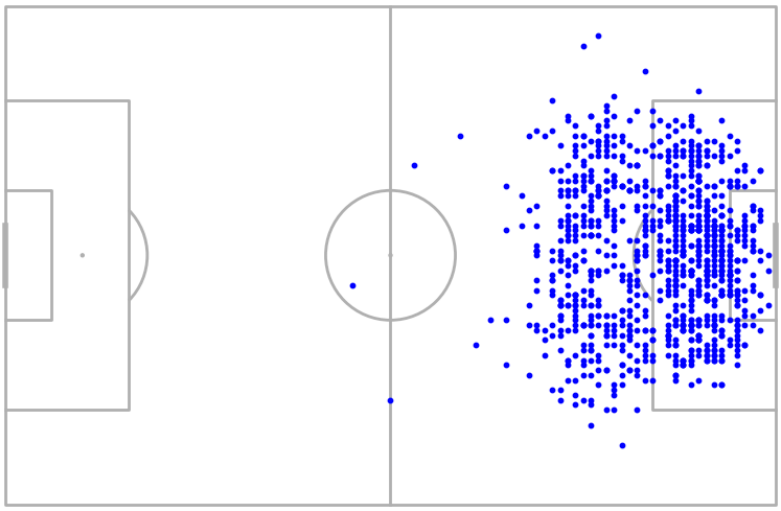

FIGURE 4 – Représentation des positions des 1000 premiers tirs

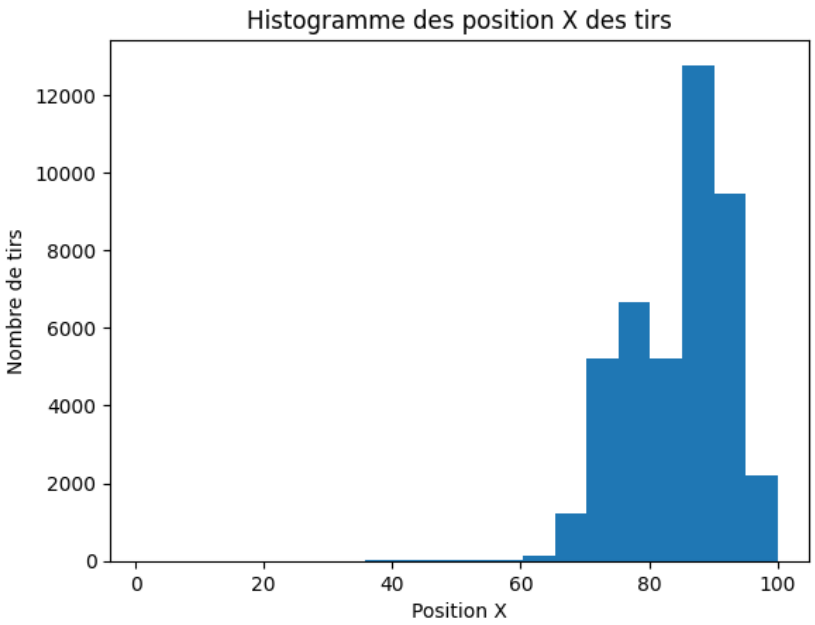

FIGURE 5 – Histogramme des positions X des tirs

7.3 Players

Le dataset contient également des informations sur les joueurs. Le tableau 6 décrit les attributs du dataset "Players".

On peut voir que le dataset contient des informations sur les joueurs comme leur taille, leur poids, leur date de naissance, leur nationalité, leur poste, etc. L'une des informations qui est intéressante dans la suite de la thèse et le pied fort du joueur. En effet, le pied fort du joueur peut être un facteur qui influence le résultat de la prédiction des xG. C'est pour cela qu'il est important de garder cet attribut dans le dataset final qui va être utilisé pour entraîner le modèle.

Nom de l'attribut	Description
birthArea	Pays de naissance du joueur
birthDate	Date de naissancxwe du joueur
currentNationalTeamId	Equipe nationale du joueur
currentTeamId	Equipe actuelle du joueur. Est lié à l'attribut "wyId" de l'équipe
firstName	Prénom du joueur
lastName	Nom du joueur
foot	Pied fort du joueur
height	Taille du joueur en centimètres
middleName	Deuxième nom du joueur. S'il en possède un
passportArea	Nationalité du joueur
role	Poste du joueur
shortName	Nom raccourci du joueur
weight	Poids du joueur en kilogrammes
wyId	Identifiant du joueur. Permet de faire le lien avec le dataset "Events"

TABLE 6 – Description des attributs du dataset "Players"

7.4 Transformation des données

Comme discuté dans la section 7.2, les coordonnées X et Y sont relatives à la taille du terrain. Un présupposé qui est fait est que la taille du terrain est de 105 par 68. Cette taille de terrain est la plus commune dans le football professionnel. On peut notamment le voir sur le site officiel de la Premier League [3]. Je démarre donc par un présupposé que tous les terrains qui ont accueilli les matchs dans le dataset ont comme dimension 105 mètres par 68. Ce présupposé nous permettra ainsi de pouvoir calculer la distance en mètre par rapport au but adverse ainsi que l'angle de tir.

Nous elaborerons notre travail sur la transformation des données en quatre parties distinctes. Pour commencer, nous transformons les coordonnées X et Y qui sont à l'échelle 100 par 100 en coordonnées X et Y à l'échelle 105 par 68. Pour cela, il suffit de multiplier les coordonnées X par 1.05 et les coordonnées Y par 0.68.

$$\begin{aligned} x_{105} &= x_{100} * 1.05 \\ y_{68} &= y_{100} * 0.68 \end{aligned} \tag{3}$$

On passe également les coordonnées X de l'autre côté du terrain. Cela permettra de calculer la distance par rapport au centre du terrain. Il suffit de soustraire les coordonnées X à 105 [10].

Par la suite, nous allons créer un attribut qui contient la distance par rapport au centre du terrain sur l'axe Y. Cet attribut permettra de calculer la distance du tir ainsi que son angle. Pour cela, il suffit de calculer la distance entre le centre du terrain et la position du tir, mais uniquement sur l'axe Y.

$$y_{Centre} = |y_{68} - 34| \tag{4}$$

34 étant la moitié de la largeur du terrain. Cela nous permet d'avoir la distance par rapport au centre du terrain sur l'axe Y [11].

Ensuite, nous calculons la distance du tir par rapport au but adverse.
Sur la figure 6, [C] représente le centre du terrain, [E] représente le centre du but adverse. [DG] représente l'attribut que l'on a calculer précédemment qui contient la distance par rapport au centre du terrain sur l'axe Y. [DF] représente la position en X par rapport au but adverse. Ce qui est effectué pour calculer la distance du tir par rapport au but adverse est d'utiliser le théorème de Pythagore. On peut voir sur la figure 6 que l'on a un triangle rectangle [GDF]. Avec le théorème de Pythagore, on peut savoir que :

$$DE^2 = GF^2 = DG^2 + DF^2 \tag{5}$$

Il suffit de faire une racine carré de la somme de DG^2 et DF^2 pour avoir la distance du tir par rapport au centre du but adverse.

10. Dans mon cas, j'ai soustrait les coordonnées avant de passer à l'échelle 105x68, donc j'ai soustrait les coordonnées de X à 100.

11. Pareil que pour la coordonnée X, la soustraction a été faite sur l'échelle 100x100 avant de passer à l'échelle 105x68

Pour finir, nous allons calculer l'angle du tir par rapport au but adverse. Il est d'abord important d'amener une autre information qui, cette fois-ci, n'est pas un présupposé. En effet, la largeur des buts de football est de 7.32 mètres [4]. Pour cette partie, le choix a été d'utiliser l'inverse de la tangente et la règle qui dit que la somme des angles d'un triangle est égale à 180 degrés. Grâce à cela, on peut calculer l'angle du tir par rapport au but adverse. On peut voir sur la figure 7 que l'on a un triangle JGF. L'objectif est de calculer l'angle situé au sommet G. On sait que GI représente y_{Centre} et que HF représente la valeur de x. FJ représente la largeur du but qui est, comme indiqué précédemment, de 7.32 mètres. On peut donc avoir la valeur de GH et GK en additionnant ou soustrayant la moitié de la largeur du but à y_{Centre}.

$$\begin{aligned} GH &= y_{Centre} - \frac{7.32}{2} \\ GK &= y_{Centre} + \frac{7.32}{2} \end{aligned} \tag{6}$$

En utilisant la tangente inverse, on peut calculer l'angle GJK ainsi que l'angle GFH. Avec ces deux angles, on peut calculer l'angle à l'intérieur du triangle JGF pour finalement récupérer l'angle du tir par rapport au but adverse en faisant [12] :

$$\alpha = 180 - (Angle_{GJK} + Angle_{GFH}) \tag{7}$$

Finalement l'équation totale ressemble à cela, une fois que l'on a toutes les valeurs :

$$\alpha = 180 - (tan^{-1}(\frac{GH}{HF}) + 90) - (90 - tan^{-1}(\frac{GK}{HF})) \tag{8}$$

Il est d'ailleurs possible d'accéder au schémas géométrique sur Geogebra. Cela permet de manipuler les différents points et de voir les valeurs se mettre à jour automatiquement. Ils sont disponibles en bas de page [13] [14].

7.5 Fusion des datasets

Cette section vise à expliquer comment les datasets ont été fusionnés. On a pu voir que le dataset events contient l'identifiant du joueur qui a effec-

12. En degrés
13. Angle : `https://www.geogebra.org/geometry/d7tqw4fe`
14. Distance : `https://www.geogebra.org/geometry/fnx3swex`

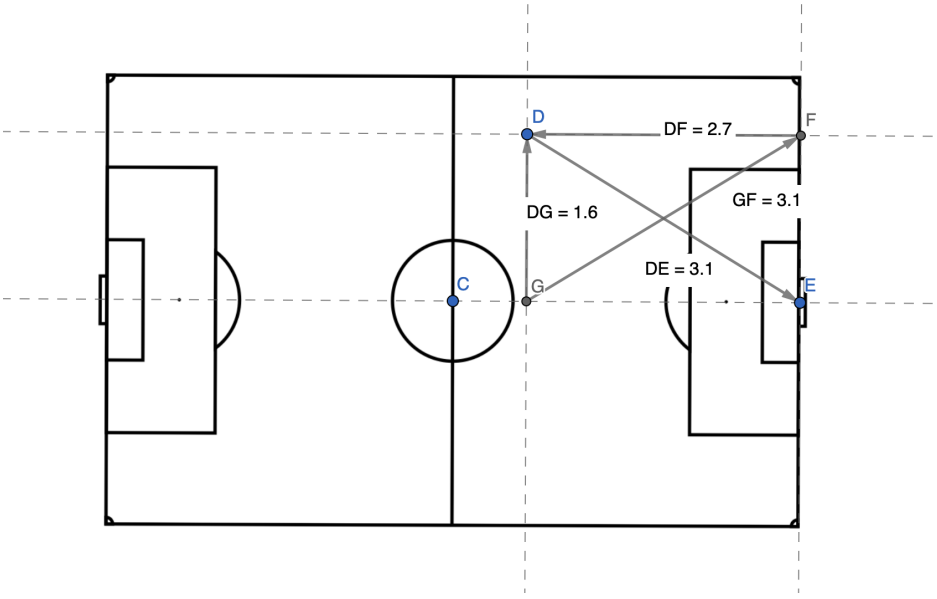

FIGURE 6 – Distance du tir par rapport au but adverse

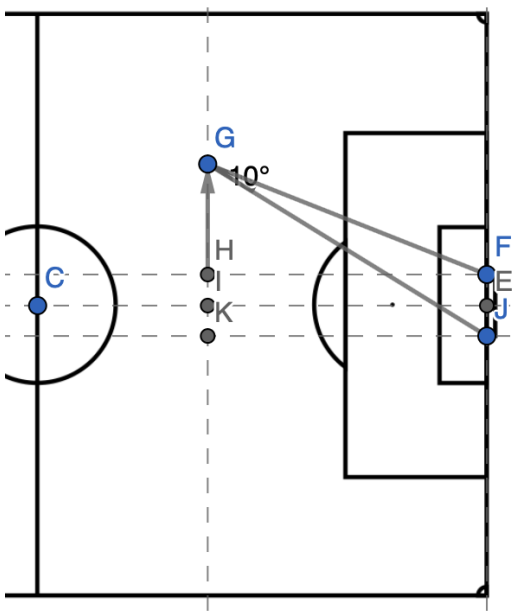

FIGURE 7 – Angle du tir par rapport au but adverse

tué l'action sous le nom d'attribut "playerId" (voir table 5). Cet identifiant correspond à l'identifiant nommé "wyId" dans le dataset players (voir table 6). L'objectif est donc de fusionner les deux datasets en utilisant ce deux attributs pour permettre d'avoir les informations des joueurs dans le dataset events. Dans notre cas, nous allons faire une jointure pour permettre de récupérer quel est le pied fort du joueur ayant généré l'événement. Grâce à cela, il nous est possible de savoir si le joueur a tiré avec son pied fort ou non.

Nom de l'attribut	**Description**
X	Position X sur le terrain. Permet de visualiser les tirs via un pitch chart
Y	Position Y sur le terrain. Permet de visualiser les tirs via un pitch chart
distance	Distance du tir par rapport au milieu du but
angle	Angle des deux poteaux par rapport au tir. En radians
angle_abs	L'angle du tir par rapport au centre du terrain.
goal	Indique si le tir est un but. **La variable cible.**
header	Indique si le tir est fait de la tête. Si c'est le cas, "good_foot_used" est False
good_foot_used	Indique si le tir est fait avec le pied fort du joueur. Si c'est le cas, "header" est False.

TABLE 7 – Attributs du dataset events

7.6 Visualisation des données

Cette section vise à visualiser les données du dataset final. Tout d'abord, nous allons effectuer une description simple du dataset. Le dataset contient 43075 tirs sur une saison de football (2017/2018) dans 7 compétitions différentes :

— Ligue 1
— Premier League
— Serie A
— Bundesliga

— La Liga
— Coupe du monde 2018
— Euro 2016

La description du dataset est disponible sur le tableau 8. On peut voir que la distance maximale d'un tir a été de 103.95 mètres et la distance minimale de 0.68 mètre. On constate donc que certains tirs n'ont pas été normalisés par rapport au sens du jeu de l'équipe comme indiqué dans la section 7.2. C'est pour cela que notre but va être de retirer certains tirs considérés comme des "outliers".

	X	**Y**	**distance**	**angle**	**angle_abs**
count	43075	43075	43075	43075	43075
mean	15.99	33.47	18.59	0.4141	2.6576
std	8.53	9.37	8.42	0.2532	0.3131
min	0.00	0.00	0.68	0.0000	1.5708
25%	9.45	26.52	12.25	0.25019	2.4418
50%	13.65	33.32	17.15	0.3278	2.6894
75%	23.10	40.80	24.94	0.5060	2.9143
max	103.95	68	103.95	3.1416	3.1416

Table 8 – Description du dataset final

Tout d'abord, nous allons visualiser la fréquence des tirs en fonction de leur position à l'aide d'une heatmap sur un pitch chart. Un pitch chart est un graphique qui affiche un terrain de football. Grâce à cela on peut visualiser plus simplement d'où ont été effectuées les tirs. Le but de notre pitch chart est de visualiser les tirs qui ont été effectués à leurs positions respectives. Dans mon cas, j'ai décidé d'utiliser une heatmap pour visualiser la fréquence des tirs à chacune des positions sur le terrain. Le pitch chart peut être vu sur la figure 8. On peut voir qu'une grande majorité des tirs ont été effectués dans la surface de réparation. On peut également remarquer qu'il y a très peu de tirs à l'entrée de la surface de réparation. Il y a plusieurs hypothèses qui peuvent être faites pour expliquer cela. La première est que les joueurs préfèrent tirer dans la surface de réparation car ils ont plus de chances de marquer. Cependant, on remarque qu'il y a également des tirs plus lointain. La deuxième hypothèse est que ce phénomène est dû à comment les données sont récoltées. Il est effectivement possible que les opérateurs ou que le logiciel propriétaire de Wyscout détermine qu'un tir ait été fait dans la surface de

réparation ou en dehors de celle-ci, plutôt que sur la ligne de la surface de réparation. Cela nous montre déjà une des limitations des données que nous utilisons.

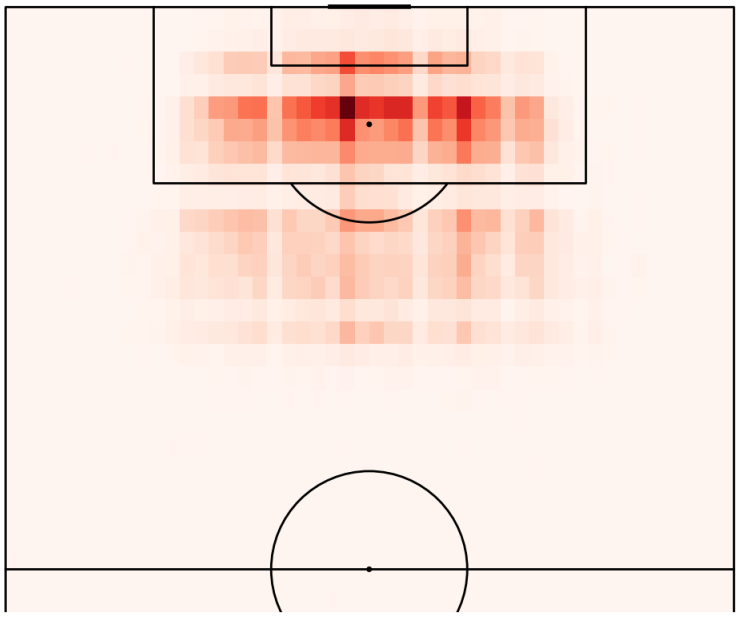

FIGURE 8 – Pitch chart des tirs effectués

Maintenant, nous allons nous focaliser sur la détection d'éventuelles "outliers". En effet, nous avons pu voir que la distance maximale d'un tir a été de 103.95 mètres. Il faut donc vérifier si ce tir est un "outlier" ou non. Nous allons également nous focaliser sur d'autres tirs considérés comme "outliers". Tout d'abord, visualisons les graphiques de la distance, l'angle de tir et l'angle de tir absolu (voir fig. 9). Lorsque l'on observe les histogrammes, on remarque que pour la distance et l'angle de tir, il y a une majorité de tirs qui ont été effectués à une distance inférieure à 50 mètres et un angle de tir inférieur à 1.5 radian. On peut également voir qu'il n'y a pas de séparation distincte entre les distributions des tirs selon leur valeur cible via les graphiques de la deuxième colonne. Cela aurait été très pertinent puisque l'on aurait pu voir quel attribut aurait été le plus influent directement en observant les données.

Cependant, on peut remarquer que des outliers ont été trouvés pour la distribution des différentes variables via les boxplots. Les outliers sont déterminées en se basant sur la règle suivante [9] :

$$\begin{aligned} x &< Q_1 - 1.5 * IQR \\ x &> Q_3 + 1.5 * IQR \end{aligned} \tag{9}$$

Comme indiqué sur la documentation de Matplotlib concernant les boxplots [2], la valeur de 1.5 est une valeur basée sur la définition initiale des boxplots de John Tukey. Ils permettent de définir les "maximum non-outlier" et "minimum non-outlier".
Via cette formule, on peut récupérer tous les outliers pour chaque variable, on obtient donc un tableau comme celui-ci (voir tab. 9) : L'objectif est maintenant de visualiser la probabilité de chacun des sets d'outliers. Si l'on observe des anomalies ou des données aberrantes, il faudra les supprimer.

	Distance	**Angle**	**Angle absolu**
Nombre d'outliers	197	2163	70

TABLE 9 – Nombre d'outliers pour chaque variable quantitatives

Grâce à la figure 10, plusieurs observations peuvent être faites. Tout d'abord, il est remarquable que les outliers pour les angles ne sont pas aberrants. En effet, théoriquement, plus l'angle est grand, plus le but est facile à réaliser. Cependant, en ce qui concerne les distances, il existe un nombre très limité d'outliers. En examinant les valeurs situées entre 90 et 105, on constate que trois buts ont été marqués à cette distance. En revanche, le nombre de tirs manqués dans la même plage est plus élevé. Théoriquement, marquer un but à 90 mètres du but est considéré comme irréalisable. Par conséquent, ces données peuvent être considérées comme aberrantes et il serait envisageable de les supprimer. Une hypothèse plausible est que ces données n'ont pas été correctement traitées par Wyscout lors de la normalisation des positions des tirs.

Il est également intéressant d'examiner la relation entre la distance et l'angle. La figure 11 met en évidence le fait que plus la distance est petite, plus l'angle est grand. Cette tendance est observée dans le premier graphique de la figure 11, bien que ce ne soit pas toujours le cas. En jouant avec le graphique Geogebra fourni précédemment, il est possible de constater que les tirs effectués

depuis le coin de l'équipe attaquante auront un angle plus élevé que les tirs effectués depuis le coin de l'équipe adverse. Un tir depuis le coin de l'équipe adverse aura systématiquement un angle de 0 radian puisqu'il est effectué le long de la ligne de but.

Ensuite, nous allons analyser les attributs qualitatifs en utilisant des diagrammes en barres (bar charts). Ces graphiques permettent de visualiser la distribution des données pour chaque attribut qualitatif. Ils sont présentés dans la figure 11. À travers ces bar charts, il est possible d'observer que 10% des tirs sont des buts. On remarque également qu'il y a une majorité de tirs effectués avec le pied fort du tireur. Cependant, les tirs qui ne sont pas effectués avec le pied fort incluent également les tirs de la tête. Environ 15% des tirs sont réalisés de cette manière, ce qui indique que la majorité des tirs sont effectués avec le pied (faible ou fort).

Enfin, nous allons analyser la corrélation entre les variables. L'objectif de cette analyse est d'évaluer si les variables sont corrélées les unes avec les autres. Pour rappel, deux variables sont corrélées si elles ont tendance à varier ensemble. Une corrélation négative indique que lorsque la valeur d'une variable augmente, la valeur de l'autre variable diminue. Une corrélation positive indique que lorsque la valeur d'une variable augmente, la valeur de l'autre variable augmente également. Une corrélation élevée entre une variable et la variable cible peut améliorer la prédiction et faciliter l'interprétation du modèle. Une corrélation élevée suggère une relation claire entre les deux variables, ce qui peut faciliter la compréhension de l'influence de la variable sur la variable cible.

Dans notre cas, en observant la figure 12, les variables présentant la corrélation la plus élevée par rapport à l'attribut "goal" sont les attributs "distance" et "angle". Nous remarquons également une forte corrélation entre les attributs "distance" et "X". Cette corrélation s'explique par le fait que l'attribut "distance" est une variable dérivée de l'attribut "X" et de l'attribut "C" qui est elle-même dérivée de l'attribut "Y".

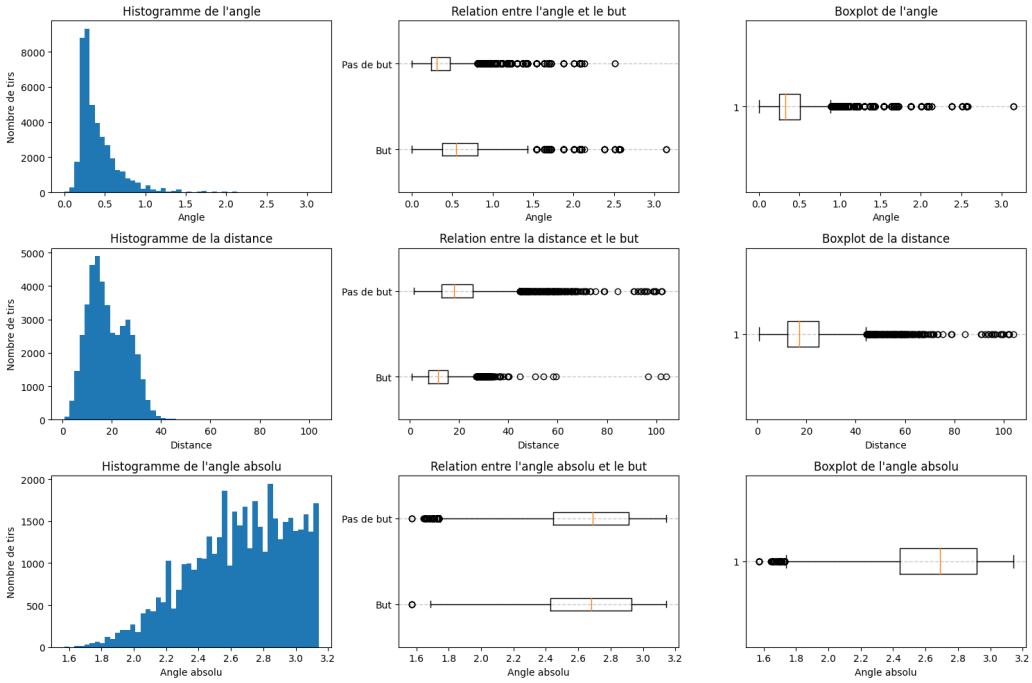

FIGURE 9 – Analyse des attributs quantitatifs

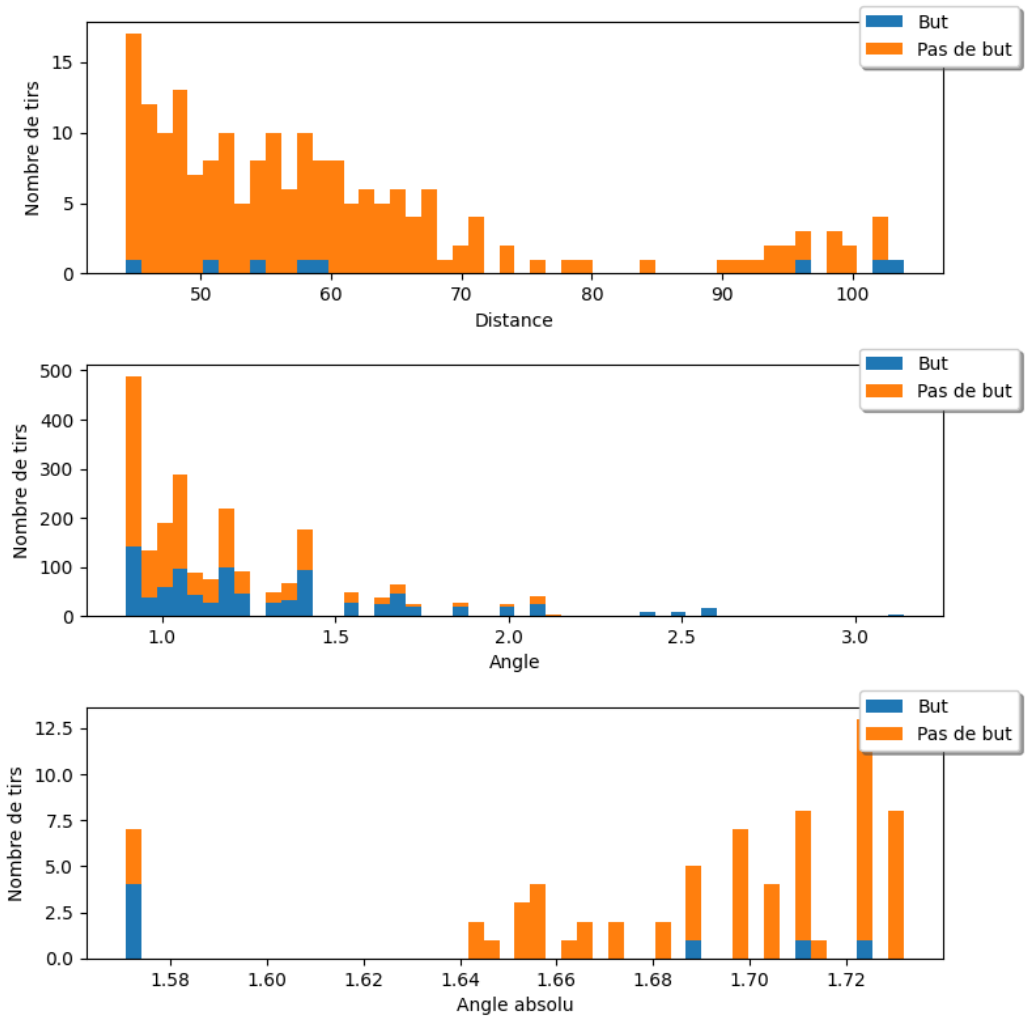

FIGURE 10 – Analyse des outliers

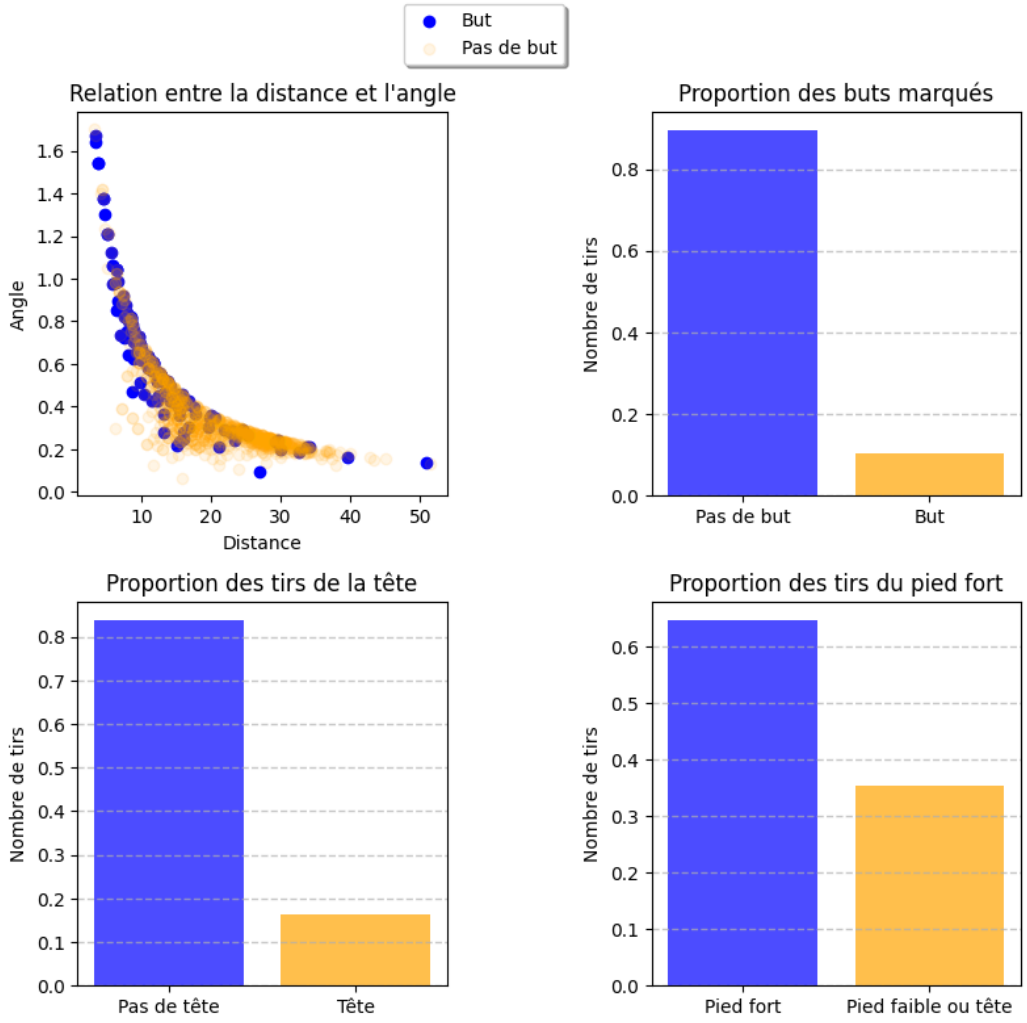

FIGURE 11 – Relation distance-angle et analyse des attributs qualitatifs

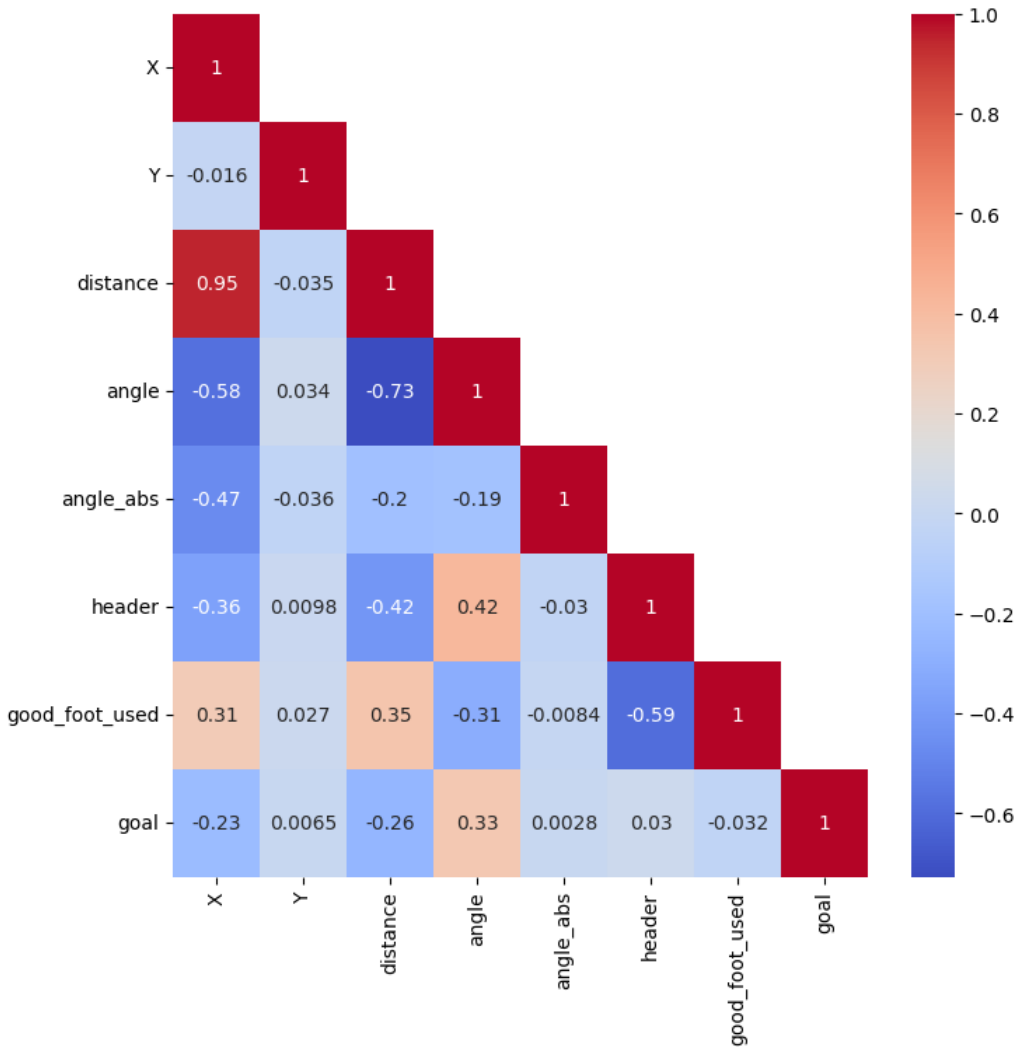

FIGURE 12 – Matrice de corrélation des attributs

Grâce à cette visualisation approfondie, nous avons pu acquérir une compréhension approfondie des données et des relations entre les différentes variables. Nous avons également identifié certaines données incohérentes et des aberrations dans le jeu de données. Ces observations soulignent l'importance de l'examen attentif et critique des données afin d'éliminer les valeurs aberrantes et d'obtenir des résultats plus fiables.

En analysant les corrélations entre les variables, nous avons constaté que certaines d'entre elles présentaient une forte corrélation avec la variable cible. Cela suggère que ces variables sont des indicateurs significatifs pour prédire la réussite d'un tir au but. Par exemple, les variables "distance" et "angle" ont montré une corrélation élevée avec la variable "goal". Cela signifie que plus la distance et l'angle sont favorables, plus la probabilité de marquer un but est élevée.

Il est également important de souligner la forte corrélation entre les variables "distance" et "X". Cette corrélation s'explique par le fait que l'attribut "distance" est calculé en utilisant les coordonnées horizontales "X" et "C". Ainsi, lorsque la valeur de l'attribut "X" change, cela influence directement la valeur de l'attribut "distance".

En résumé, cette analyse approfondie des données nous a permis de mieux comprendre les relations entre les variables et d'identifier les facteurs les plus pertinents pour prédire la réussite d'un tir au but. En éliminant les données aberrantes et en tenant compte des corrélations significatives, nous pouvons améliorer la précision des prédictions. Cependant, il convient de noter que des analyses supplémentaires et des modélisations plus avancées peuvent être nécessaires pour tirer des conclusions plus précises et éclairantes.

8 Méthodologie

Dans cette section, nous allons discuter de la méthodologie utilisée pour résoudre le problème de prédiction de la réussite d'un tir au but. Tout d'abord, nous allons faire une séparation sur le dataset pour avoir un jeu de données d'entraînement et un jeu de données de test. Le jeu de tests sera utilisé pour évaluer la performance des modèles finaux. Tandis que le jeu d'entraînements, nous permettra d'entraîner notre modèle et de choisir le meilleur set d'hyper paramètres pour chaque modèle. Le partitionnement choisi pour les jeux de données est le suivant : 90% des données pour le jeu d'entraînement et 10% des données pour le jeu de test.

8.1 Choix des modèles

Plusieurs modèles vont être utlisés pour résoudre ce problème. Nous allons utiliser des modèles de classification binaire, car nous cherchons à prédire si un tir au but est réussi ou non. Je rappelle également que le but de la thèse est de répondre à la question suivante : "Quels sont les paramètres qui influencent le plus l'expected goal ?". Certains de ces modèles ne seront pas les plus performants et les plus cohérents pour avoir la meilleure prédiction possible mais ils permettront de comprendre les paramètres qui influencent le plus l'expected goal. Nous allons tester les modèles suivants :

- Régression logistique
- Arbre de décision
- Random Forest
- KNN
- Réseau de neurones (MLP)

8.2 Cross Validation

Pour chaque modèle, nous allons utiliser la cross validation pour choisir le meilleur set d'hyper paramètres. La cross validation est une méthode qui permet de tester la performance d'un modèle sur un jeu de données. Elle consiste à séparer le jeu de données en plusieurs sous-ensembles. Comme on peut le voir sur la figure 13, le jeu de données est séparé en 5 sous-ensembles. Chaque itération consiste à entraîner le modèle sur 4 sous-ensembles et à tester le modèle sur le dernier sous-ensemble. Cette opération est répétée 5 fois pour que chaque sous-ensemble soit utilisé comme jeu de test. La performance

du modèle est calculée en faisant la moyenne des performances obtenues sur chaque sous-ensemble. Dans notre cas, cette performance est effectuée en utilisant la "log loss" comme métrique. Cette métrique est utilisée pour évaluer la performance d'un modèle de classification probabiliste [15]. C'est ainsi que nous pouvons choisir le meilleur set d'hyper paramètres pour chaque modèle.

Après avoir choisi nos modèles et leurs hyper paramètres, nous allons les entraîner sur le jeu de données d'entraînement entier et les tester sur le jeu de données de test. Cela nous permettra ensuite de comparer les résultats entre les différents modèles. Le but est d'avoir la meilleure version d'un modèle et de la comparer avec la meilleure version des autres modèles. C'est ainsi que l'on pourra déterminer quel modèle est le plus performant pour résoudre notre problème.

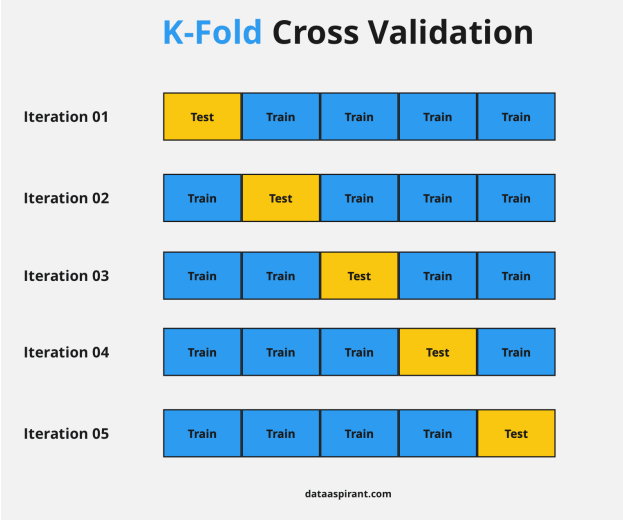

FIGURE 13 – Cross Validation - Source : Dataspirant

8.3 Sélection des meilleurs attributs

Pour la sélection des meilleurs attributs, nous allons utiliser la méthode de forward elimination. Cette méthode consiste à tester tous les attributs un par

15. Pour rappel, le xG est une métrique probabiliste puisqu'elle indique la chance qu'un tir soit transformé en but sous forme de pourcentage.

un en les ajoutant au fur et à mesure au modèle et observer la performance du modèle. La p-value est utilisée pour déterminer si un attribut est significatif ou non et s'il apporte une amélioration à la performance du modèle. Plus la p-value est faible, plus l'attribut est significatif.

Par exemple, dans le tableau 10, on peut voir que l'attribut "X" a une p-value de 0.000. Cela indique que cet attribut est significatif et qu'il apporte une amélioration à la performance du modèle en plus de la constante.

Sur le tableau 11, on peut voir que l'attribut "Y" a une p-value de 0.960. Cela indique que cet attribut n'est pas du tout significatif et qu'il n'apporte aucune amélioration à la performance du modèle. On le remarque d'ailleurs lorsque l'on observe la valeur du coefficient qui est très proche de 0. Il faut donc supprimer cet attribut du modèle.

Pour la troisième itération sur le tableau 12, on ajoute l'attribut "distance" au modèle. Cependant, on a supprimé l'attribut "Y" qui n'était pas significatif. On remarque donc que les p-values des attributs "X" et "distance" sont très faibles. Cela indique que ces deux attributs sont significatifs et qu'ils apportent une amélioration à la performance du modèle. La valeur du coefficient de l'attribut "distance" est négative. Cela veut dire que plus la distance est grande, plus la probabilité d'avoir un but est faible.

Pour la quatrième itération sur le tableau 13, on ajoute l'attribut "angle" au modèle. Cependant, dans ce cas-ci, on observe que la p-value de l'attribut "X" est très élevée. Cela est dû à la corrélation entre les attributs "X" et "angle" qui est répétitive dû à la corrélation avec l'attribut "distance". C'est pour cela que l'attribut "X" n'est plus significatif et qu'il n'apporte plus d'amélioration à la performance du modèle. C'est justement dû au fait que l'attribut "distance" et l'attribut "angle" sont des attributs dérivés de l'attribut "X" et de l'attribut "Y".

Lors de la cinquième itération, on ajoute l'attribut "angle_abs" au modèle. Ici, on observe que la p-value de l'attribut "angle_abs" est très élevée. Cela est dû à la corrélation entre les attributs "angle" et "angle_abs" qui est répétitive. C'est pour cela que l'attribut "angle_abs" n'est pas significatif et qu'il doit être retiré.

Pour finir, on peut voir sur le tableau 15 que l'ajout de l'attribut "header" et "good_foot_used" apporte une amélioration significative au modèle. Cela se voit grâce aux p-values très faibles de ces deux attributs lorsqu'on les ajoute aux autres attributs du modèle.

Logistic Regression Result

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.4809	0.037	-13.120	0.000	-0.553	-0.409
X	-0.1280	0.003	-42.743	0.000	-0.134	-0.122

TABLE 10 – Résultat de la régression logistique - 1ère itération

Logistic Regression Result

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.4843	0.076	-6.334	0.000	-0.634	-0.334
X	-0.1280	0.003	-42.729	0.000	-0.134	-0.122
Y	9.939e-05	0.002	0.050	0.960	-0.004	0.004

TABLE 11 – Résultat de la régression logistique - 2ème itération

Logistic Regression Result

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.1926	0.043	4.488	0.000	0.108	0.277
X	0.0692	0.008	8.579	0.000	0.053	0.085
distance	-0.2118	0.008	-26.825	0.000	-0.227	-0.196

TABLE 12 – Résultat de la régression logistique - 3ème itération

Logistic Regression Result

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.3884	0.123	-11.294	0.000	-1.629	-1.147
X	0.0060	0.009	0.660	0.509	-0.012	0.024
distance	-0.0984	0.011	-8.799	0.000	-0.120	-0.076
angle	1.3701	0.102	13.395	0.000	1.170	1.571

TABLE 13 – Résultat de la régression logistique - 4ème itération

Logistic Regression Result

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.3245	0.154	-8.573	0.000	-1.627	-1.022
distance	-0.0897	0.005	-18.200	0.000	-0.099	-0.080
angle	1.4499	0.102	14.243	0.000	1.250	1.649
angle_abs	-0.0592	0.064	-0.932	0.352	-0.184	0.065

TABLE 14 – Résultat de la régression logistique - 5ème itération

Logistic Regression Result

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.0751	0.114	-9.413	0.000	-1.299	-0.851
distance	-0.1126	0.005	-23.827	0.000	-0.122	-0.103
angle	1.5497	0.093	16.609	0.000	1.367	1.733
header	-0.9067	0.059	-15.468	0.000	-1.022	-0.792
good_foot_used	0.1224	0.046	2.634	0.008	0.031	0.214

TABLE 15 – Résultat de la régression logistique - Dernière itération

Pour conclure cette partie de sélection d'attributs, on voit donc que les attributs les plus significatifs parmi ceux que l'on a sélectionné pour le dataset final sont :

- La distance entre la position du tir et le but
- L'angle entre les deux poteaux et la position du tir
- Le tir a été effectué avec le pied fort du joueur
- Le tir a été fait de la tête

On peut donc en conclure que les attributs que l'on a sélectionné sont pertinents pour la prédiction de la réussite d'un tir. On peut également voir que les attributs que l'on a sélectionné sont cohérents avec les résultats de la littérature. En effet, on retrouve les attributs suivants :

- La distance entre la position du tir et le but
- L'angle entre les deux poteaux et la position du tir

Avec cette conclusion, on a pu répondre à la première question de la thèse : **Quels sont les paramètres qui influencent le plus l'expected goal ?**

8.4 Jeu d'hyper paramètres et séléction

Pour le choix des hyper paramètres pour chacun des nos modèles, le but est d'utiliser une méthode de validation croisée. Chacun des modèles sera créé avec un jeu d'hyper paramètres différent et passera par une validation croisée. Chaque résultat de validation croisée sera ensuite comparé pour trouver le meilleur jeu d'hyper paramètres pour chaque modèle. Comme indiqué dans la partie 8.2, on utilisera une validation croisée à 5 folds pour chacun des modèles en utilisant la métrique du log loss. On utilisera la méthode en grille pour trouver les meilleurs hyper paramètres pour chacun des modèles. Il est possible de trouver les hyper paramètres les plus optimaux pour chacun des modèles relatifs à notre cas dans les tableaux 16, 17, 18, 19 et 20 .

Régression logistique		
Hyper paramètre	**Valeur**	**Description**
penalty	l2	Indique la méthode de régularisation du modèle
C	100	Indique la force de la régularisation du modèle
max_iter	100.0	Nombre d'itérations maximale pour arriver à une convergence
solver	liblinear	L'algorithme utilisé pour résoudre le modèle

TABLE 16 – Jeu d'hyper paramètres choisi pour la régression logistique

Arbre de décision		
Hyper paramètre	**Valeur**	**Description**
criterion	gini	Critère qui détermine la séparation de l'arbre
max_depth	5	Profondeur maximale acceptée.
min_samples_leaf	100	Nombre d'instances minimales acceptée en tant que feuille.
min_samples_split	5	Nombre d'instances minimales pour créer une séparation sur un noeud

TABLE 17 – Jeu d'hyper paramètres choisi pour l'arbre de décision

Random Forest		
Hyper paramètre	**Valeur**	**Description**
criterion	entropy	Critère qui détermine la séparation de chaque arbre dans la forêt
max_depth	5	Profondeur maximale acceptée.
min_samples_leaf	10	Nombre d'instances minimales acceptée en tant que feuille.
min_samples_split	100	Nombre d'instances minimales pour créer une séparation sur un noeud
n_estimators	25	Nombre d'arbres dans la forêt

TABLE 18 – Jeu d'hyper paramètres choisi pour la forêt aléatoire

KNN		
Hyper paramètre	**Valeur**	**Description**
n_neighbors	101	Nombre de voisins à observer pour déterminer la classification. La classe majoritaire parmi les voisins est prédite.

TABLE 19 – Jeu d'hyper paramètres choisi pour KNN

Perceptron multi-couches		
Hyper paramètre	**Valeur**	**Description**
activation	logistic	Type de fonction d'activation
alpha	0.001	La force de la régularisation (pénalité)
hidden_layer_sizes	(50, 50, 50)	Indique le nombre de couches et le nombre de neurones par couche
learning_rate	constant	Indique le taux d'apprentissage pour mettre à jour les poids de chacun des neurones
solver	adam	L'algorithme utilisé pour apprendre les poids pour chacun des neurones

TABLE 20 – Jeu d'hyper paramètres choisi pour le perceptron multi-couches

9 Résultats

9.1 Comparaison des performances sur la log loss

Cette section vise à discuter des résultats. En effet, après avoir effectuée la validation croisée sur tous les modèles pour trouver leurs meilleurs hyper paramètres, il a fallu tester les meilleurs versions des modèles sur les données de test qui étaient, jusque là, jamais utilisées. Les résultats sont présentés dans le tableau 21. Chacun des modèles a été évalué en utilisant la métrique de perte logarithmique (log loss), qui mesure la performance des prédictions probabilistes. Une valeur plus faible de log loss indique de meilleures performances du modèle.

Modèle	**Valeur de log loss**
Régression logistique	0.2882034140214798
Random Forest	0.2884725624041776
Perceptron multi-couches	0.2888731795934776
Arbre de décision	0.29005568551254723
KNN	0.43044530384474966
Classificateur naïf	3.8152056512157433

TABLE 21 – Résultats des modèles sur les données de test

Le classificateur naïf part du principe que toutes les futures instances sont de la classe majoritaire du jeu de données d'entraînement. Il est important de noter que ce classifieur ne sera jamais utilisé dans un contexte réel. Il est utilisé ici pour avoir une référence de base et donc, pouvoir comparer les autres modèles sur cette base. Cela explique pourquoi il obtient une log loss aussi élevée. En effet, le décalage entre la prédiction de ce classifieur et la valeur réelle est très grand et cela a une influence direct sur la valeur finale du log loss.

En examinant les résultats, nous constatons que la régression logistique est le modèle le plus performant. Cependant, le modèle de Random Forest suit de près. La différence entre ces deux modèles est très faible, ce qui indique qu'ils ont des performances similaires.

Le perceptron multi-couches et l'arbre de décision affichent également des

performances comparables. Encore une fois, la différence entre ces deux modèles est minime.

En revanche, le modèle KNN se classe comme le modèle le moins performant parmi ceux testés (hormis le classificateur naïf). De ce fait, on peut déjà savoir que ce modèle n'est pas le plus approprié pour ce problème.

9.2 Visualisation de la distribution des prédictions

Par la suite, il a été intéressant d'observer la distribution des résultats prédits sur les données de tests. Cela nous permet de constater les différences parmi les modèles qui sont similaires en termes de log loss. Par exemple, les modèles de régression logistique et de Random Forest ont des performances similaires, mais leurs distributions de résultats prédits diffèrent légèrement. Cependant, cela n'indique toujours pas si le modèle est significativement plus performant qu'un autre puisque c'est basée sur les données de tests seulement. Autrement, on peut observer que la distribution des résultats prédits par le modèle KNN est très différente des autres modèles. Il en va de même pour l'arbre de décision qui était très similaire au perceptron multi-couches en termes de log loss mais qui a une distribution de résultats prédits très différente.

Finalement, on observe que le classificateur naïf prédit systématiquement la classe majoritaire (qui est un tir manqué), ce qui était prévisible puisque c'est le principe de ce modèle.

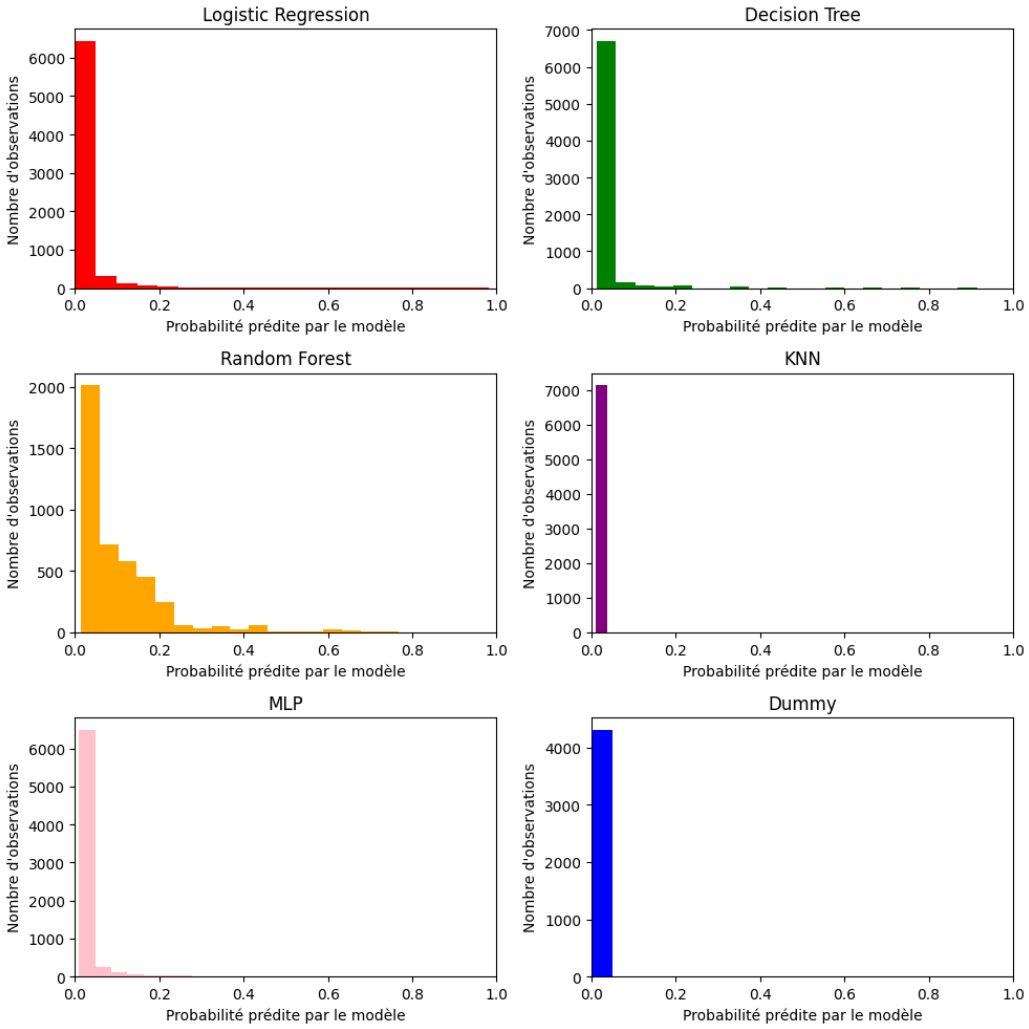

FIGURE 14 – Distribution des résultats prédits sur les données de test

9.3 Visualisation des résultats sur un terrain

Pour ajouter une nouvelle manière d'analyser et visualiser les résultats, j'ai décidé de créer une heatmap pour chacune des positions du terrains de football. À partir de cela, on calculera la distance et l'angle. J'ai également indiqué que tous les tirs effectués à chacune des positions n'ont pas été effectuées de la tête et qu'ils ont été effectuées avec le pied fort du joueur. On peut voir justement que la majorité des tirs ont été effectués avec le pied fort du joueur et qu'ils n'ont pas été effectuées de la tête (voir figure 11). Il est donc plus pertinent, pour cette visualisation, de se concentrer sur la position du tir pour éviter d'ajouter un éventuel biais à notre visualisation en prenant uniquement en compte les tirs du pied faible ou de la tête.

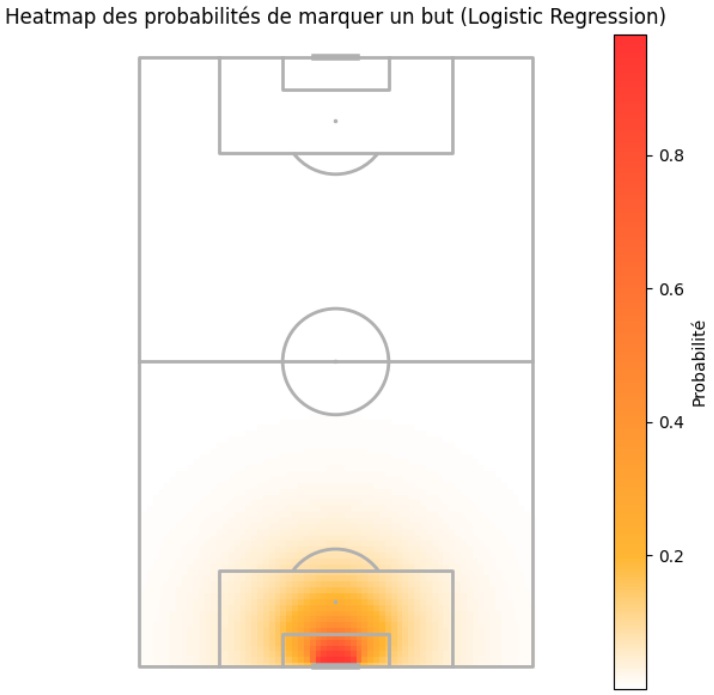

FIGURE 15 – Heatmap des résultats prédits par la régression logistique

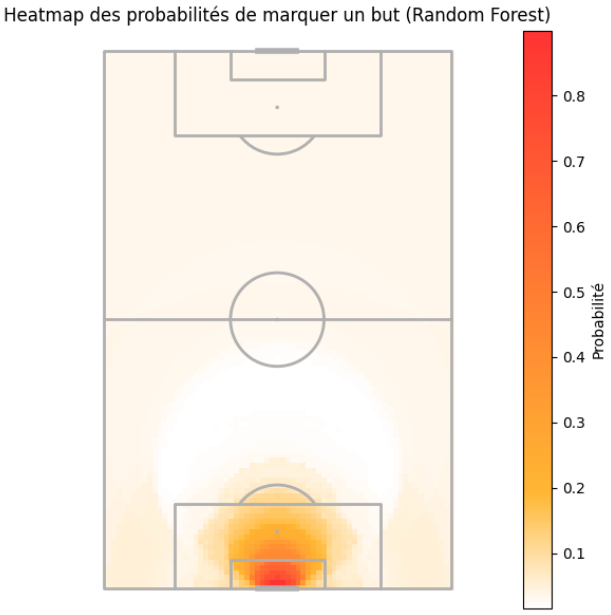

FIGURE 16 – Heatmap des résultats prédits par le modèle Random Forest

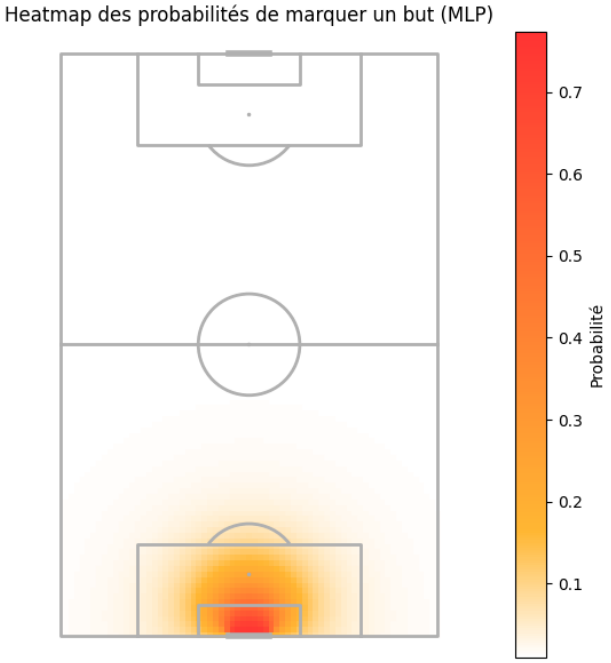

FIGURE 17 – Heatmap des résultats prédits par le perceptron multi-couches

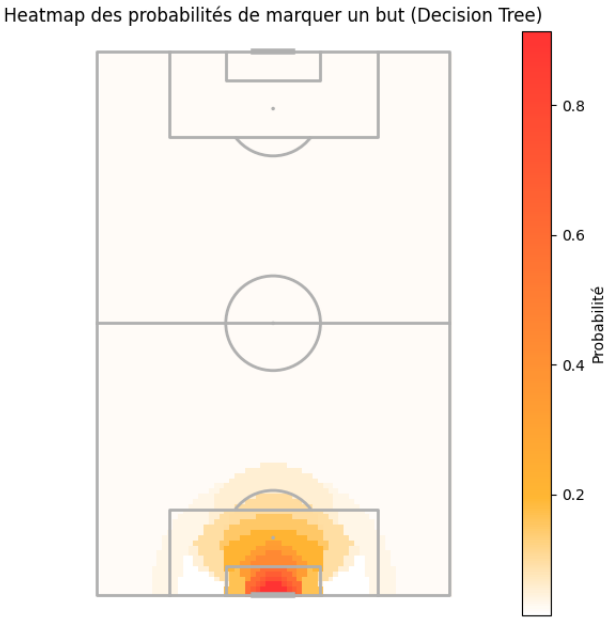

FIGURE 18 – Heatmap des résultats prédits par l'arbre de décision

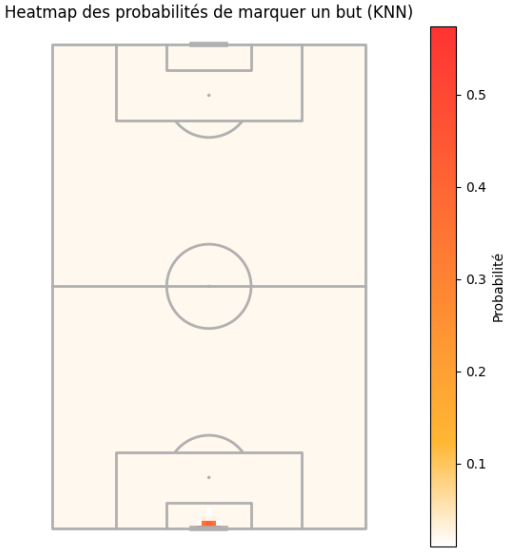

FIGURE 19 – Heatmap des résultats prédits par le modèle KNN

Lorsque l'on observe les figures 15, 16, 17, 18, on constate que la tendance générale des modèles est la même. C'est-à-dire qu'ils prédisent que les tirs effectués dans la surface de réparation ont une plus grande chance d'être transformée en but que ceux effectués à l'extérieur de la surface. Cependant, on peut constater plusieurs choses. Tout d'abord, on peut voir que le modèle Random Forest prédit que les tirs effectués à l'extérieur de notre côté du terrain ont une plus grande chance d'être marqué que ceux qui sont effectuées à l'extérieur de la surface mais dans le camp adverse. C'est également visible pour l'arbre de décision qui prédit une plus forte chance de marquer pour les tirs effectués dans le propre camp que ceux effectués dans la surface adverse dans un angle relativement faible.

Ensuite, on peut voir que le modèle KNN prédit de manière incorrecte les probabilités. On a pu le voir notamment via sa log loss qui était relativement élevée par rapport aux autres modèles. On peut également le voir sur la figure 19 où l'on constate que les probabilités prédites sont très similaires pour toutes les positions du terrain, sauf pour les tirs effectués à 1 mètre du but adverse.

C'est alors via ces visualisations que l'on peut constater et comparer les différences entre les modèles. On constate donc que les deux modèles les plus performants sont la régression logistique et le perceptron multi-couches. La régression logistique a une meilleure log loss que le perceptron multi-couches. On voit notamment que malgré le fait que la log loss du perceptron multi-couches soit plus élevée que le modèle Random Forest, sa visualisation en heatmap est plus proche de la réalité que le modèle Random Forest qui lui fournit quelques résultats incohérents.

Pour conclure cette section sur l'analyse des résultats, on peut dire que la régression logistique est le modèle le plus performant pour résoudre ce problème. Il est important de rappeler qu'uniquement 5 modèles de classifications ont été testés. Dû à un manque de puissance de calcul, il n'a pas été possible de tester plus de modèles ni de fournir davantage d'hyper paramètres. Il est donc possible qu'un autre modèle soit plus performant que la régression logistique pour la résolution de ce cas.

10 Conclusion

10.1 Conclusion générale

10.2 Améliorations possibles

Références

[1] Daryl Morey's 13-year run with the Rockets summed up in 5 incredible stats.

[2] Matplotlib.pyplot.boxplot — Matplotlib 3.7.1 documentation.

[3] Premier League Clubs – Fixtures, Results, Stats & Profiles.

[4] Terrain | IFAB.

[5] Wyscout API.

[6] Wyscout Glossary.

[7] xG Explained | FBref.com.

[8] Hph Harm Eggels. Expected goals in soccer :explaining match results using predictive analytics.

[9] Michael Galarnyk. Understanding Boxplots : How to Read and Interpret a Boxplot | Built In.

[10] Fred Garratt-Stanley. What is Expected Goals (xG) ?

[11] Ziff Davis Inc. *PC Mag*. Ziff Davis, Inc.

[12] Luca Pappalardo, Paolo Cintia, Alessio Rossi, Emanuele Massucco, Paolo Ferragina, Dino Pedreschi, and Fosca Giannotti. A public data set of spatio-temporal match events in soccer competitions. 6(1) :236.

[13] Luke Petty. What is expected goals ? Expected goals explained.

[14] Richard Pollard, Jake Ensum, and Samuel Taylor. Estimating the probability of a shot resulting in a goal : The effects of distance, angle and space. 2.

[15] Alan Ryder. Isolating Shot Quality - Hockey Analytics.

[16] David Sumpter. Fitting the xG model — Soccermatics documentation.

[17] David Tennerel. Bien utiliser les expected goals (xg) pour vos paris sportifs.

[18] Izzatul Umami, Deden Hardan Gautama, and Heliza Rahmania Hatta. Implementing the Expected Goal (xG) model to predict scores in soccer matches. 4(1) :38–54.