

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 30 JUIN 2023

Haute école de Gestion de Genève (HEG-GE)

Filière Informatique de gestion

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.orkund.com>

L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations 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 30 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 Rayane Ammad, Raphaël Lopes ainsi que ma belle-soeur, Alessia Marques, pour leur relecture et leur retour 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 se concentre sur l'analyse des statistiques sportives, en mettant particulièrement l'accent sur les expected goals (xG) dans le football. La première partie du travail présente la problématique générale des statistiques dans le sport, puis introduit les expected goals comme une métrique pertinente pour évaluer les performances d'une équipe ou d'un joueur.

Ensuite, une revue de la littérature est réalisée pour explorer les travaux existants sur les expected goals. Cette étape permet d'acquérir une compréhension approfondie des avancées antérieures dans ce domaine et d'identifier les lacunes ou les opportunités de recherche supplémentaires.

Pour poursuivre, l'obtention d'un ensemble de données approprié est cruciale. Ainsi, des recherches sont menées pour trouver un dataset pertinent et fiable contenant les informations nécessaires pour calculer les expected goals. Une fois le dataset obtenu, une analyse approfondie des méthodes de collecte des données est effectuée afin de bien comprendre leur origine et leur fiabilité.

Après avoir collecté les données, des étapes de prétraitement, d'analyse et de transformation sont réalisées. Cela peut inclure des tâches telles que le nettoyage des données, l'extraction de caractéristiques pertinentes et la normalisation des variables, afin de préparer les données pour l'apprentissage automatique.

Finalement, plusieurs modèles de classification sont mis en œuvre pour résoudre la problématique spécifique de prédire les expected goals dans le football. Ces modèles sont entraînés sur les données préalablement traitées et évalués quant à leurs performances. Les mesures de performance comparatives entre les modèles permettent de déterminer le meilleur modèle répondant à la problématique étudiée.

Liste des tableaux

1	Résultats de la régression logistique du travail de Pollard, Ensum et Taylor	14
2	Résumé de la régression logistique du modèle de xG de David Sumpter	16
3	Sources de données utilisées par H.P.H Eggels	17
4	Récapitulatif des travaux existants	18
5	Description des attributs du dataset "Events"	22
6	Description des attributs du dataset "Players"	26
7	Attributs du dataset events	30
8	Description du dataset final	31
9	Outliers pour chaque variable quantitatives	33
10	Résultats de la régression logistique - 1ère itération	44
11	Résultats de la régression logistique - 2ème itération	44
12	Résultats de la régression logistique - 3ème itération	44
13	Résultats de la régression logistique - 4ème itération	45
14	Résultats de la régression logistique - 5ème itération	45
15	Résultats de la régression logistique - Dernière itération	45
16	Jeu d'hypers paramètres choisi pour la régression logistique	47
17	Jeu d'hypers paramètres choisi pour l'arbre de décision	47
18	Jeu d'hypers paramètres choisi pour la forêt aléatoire	47
19	Jeu d'hypers paramètres choisi pour KNN	48
20	Jeu d'hypers paramètres choisi pour le perceptron multi-couche	48
21	Résultats des modèles sur les données de test	49

Table des figures

1	Schéma montrant les chances de buts selon les positions des tirs - Source : Opta	10
2	Aperçu du logiciel propriétaire de Wyscout	20
3	Représentation du terrain de football avec les coordonnées X et Y	24
4	Représentation des positions des 1000 premiers tirs	24
5	Histogramme des positions X des tirs	25
6	Distance du tir par rapport au but adverse	29
7	Angle du tir par rapport au but adverse	29
8	Pitch chart des tirs effectués	32
9	Analyse des attributs quantitatifs	35
10	Analyse des outliers	36
11	Relation distance-angle et analyse des attributs qualitatifs	37
12	Matrice de corrélation des attributs	38
13	Cross Validation avec 5 sous-ensembles	42
14	Distribution des résultats prédits sur les données de test	51
15	Heatmap des résultats prédits par la régression logistique	52
16	Heatmap des résultats prédits par le perceptron multi-couche	53
17	Heatmap des résultats prédits par le modèle Random Forest	53
18	Heatmap des résultats prédits par l'arbre de décision	54
19	Heatmap des résultats prédits par le modèle KNN	54

Table des matières

1 Déclaration	1
2 Remerciements	2
3 Résumé	3
Liste des tableaux	4
Table des figures	5
4 Introduction	8
4.1 Introduction à la problématique	8
4.2 Intérêt de la problématique	10
4.3 Questions auxquelles nous souhaitons répondre dans ce travail	11
4.4 Plan du document	13
5 Synthèse des travaux existants	14
5.1 Récapitulatif des travaux existants	18
6 Dataset	19
6.1 Présentation du dataset	19
6.2 Events	21
6.3 Players	25
6.4 Transformation des données	26
6.5 Fusion des datasets	30
6.6 Visualisation des données	30
7 Méthodologie	40
7.1 Choix des modèles	40
7.2 Cross Validation	41
7.3 Sélection des meilleurs attributs	42
7.4 Jeu d'hypers paramètres et sélection	46
8 Résultats	49
8.1 Comparaison des performances sur la log loss	49
8.2 Visualisation de la distribution des prédictions	50
8.3 Visualisation des résultats sur un terrain	52

9 Conclusion	57
9.1 Conclusion générale	57
9.2 Améliorations possibles	59
Références	60

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'ajouter. Le nombre de tirs dans un match par équipe, 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 devenues personnelles à chacun des joueurs. Nous pouvons également compter, pour le baseball, le nombre de fois qu'un joueur était au bâton, son nombre de doubles, de triples, son nombre de buts et bien d'autres.

C'est d'ailleurs dans le baseball que nous retrouvons 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 statistique 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 statistique avancée pour établir le meilleur choix pour son équipe de départ. [1] Deux saisons plus tard, il remporte la Série mondiale 1986 [2]. 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 statistique recruté chez les Rockets de Houston en NBA en 2007. [3] 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. Nous constatons 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 effectués 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 celui-ci. Dans le football, l'une des premières études sur l'expected goal vient de Richard Pollard, Jake Ensum et Samuel Taylor ayant 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, qui est très souvent réduit par le terme "xG". Il est une métrique qui permet de déterminer la probabilité qu'un tir soit transformé en but selon les données de ce tir [7]. Ce dernier ayant un xG de 0.4 a une probabilité de 40% d'être transformé en but. Un autre avec un xG à 1, qui est la plus grande valeur possible, aurait donc 100% de chance d'être transformé en but.⁴ [13]

Pour comprendre ce qu'est réellement un xG, nous allons l'observer avec l'emplacement des tirs sur le terrain. Sur la figure [1], nous pouvons voir un exemple d'emplacement de tirs et de leur xG.

Par ailleurs, les xG se sont tellement développés que des métriques dérivées ont été créées. Nous pouvons, 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] Il existe également les xGA, qui sont les expected goals against. Ceux-ci permettent 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, ceci basé sur les données relatives aux xG. D'autres dérivés sont indiquées sur l'article de Pinnacle écrit par Luke Petty. [13]

4. 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 dérivés actuels, il semble pertinent de décrire l'utilisation de cette métrique dans le football actuel.

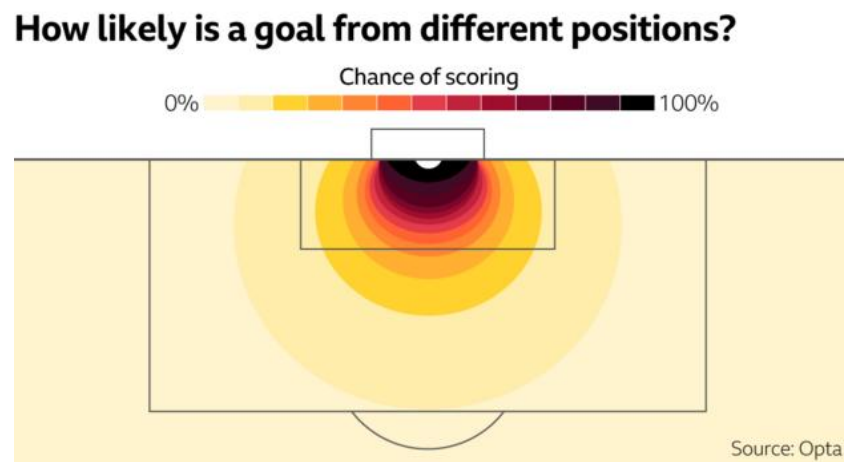


FIGURE 1 – Schéma montrant les chances de buts selon les positions des tirs
- Source : Opta

4.2 Intérêt de la problématique

Cette problématique est très intéressante puisque cette métrique représente une donnée dernièrement très populaire dans le monde du football. Elle donne plus d'informations sur le match que les autres statistiques d'un match (possession, nombre de passes, nombre de tirs cadrés, nombre de buts, etc.). Prenons comme exemple la possession de balle, une équipe peut en avoir 70% pendant le match, mais ne pas marquer de buts, ni même être dangereuse avec le ballon. Les tirs cadrés, s'ils sont tous effectués 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 celui-ci 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 comparent 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 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 une équipe qui a du 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 dérivés des xG comme les xA⁵. Les différents dérivés peuvent permettre de juger les performances et qualités d'un joueur qui se trouve à 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 auxquelles nous souhaitons 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 influencent 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.

5. Passes décisives attendues

Cela peut permettre de 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 utilisés 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.

4.4 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 [5](#). Parmi ceux-ci, nous chercherons à savoir quels datasets ont été utilisés ainsi que les résultats obtenus. Cela nous permettra de comparer nos résultats avec ceux 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 notre recherche. 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 importante pour visualiser les données. Cela permettra de voir si les données sont exploitables et 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 [6](#).

Une fois les données 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 supprimé, car il n'apporte rien pour la prédiction de l'expected goal. Cette section de transformation du dataset est également trouvable dans la section [6](#) qui fait suite à la section d'analyse du dataset initial.

Ensuite, nous pourrions commencer à implémenter le modèle et à observer les facteurs qui influencent la prédiction du xG. Cette partie est trouvable dans la section [7](#). 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 [8](#) qui visent à montrer la fiabilité et la cohérence du modèle par rapport à la problématique initial.

5 Synthèse des travaux existants

Le premier travail répertorié sur les xG et la qualité d'un tir est celui de Richard Pollard, Jake Ensum et Samuel Taylor [14]. Dans ce dernier 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 répertoriés 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 – Résultats de la régression logistique du travail de Pollard, Ensum et Taylor

Le deuxième travail est celui d'Izzatul Umami, Deden Hardan Gutama et Heliza Rahmania Hatta [18]. Il utilise les données de Wyscout des 5 championnats majeurs en Europe de la saison 2019-20. Les auteurs ont décidé de prendre comme données :

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

Dans le dataset, 32'000 tirs ont été utilisés 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 test. 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 de ce dernier 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.

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative} \quad (1)$$

$$Specificity = \frac{TrueNegative}{TrueNegative + FalsePositive} \quad (2)$$

Pour calculer la sensibilité et la spécificité, nous pouvons voir que nous avons besoin des "True Positive", "True Negative", "False Positive" et "False Negative". Les "True Positive" sont les tirs qui sont des buts ayant été correctement prédits et les "True Negative" sont les tirs qui sont manqués ayant été également correctement prédits. À l'inverse, les "False Positive" sont les tirs qui ont été manqués ayant été prédits comme des buts et de même pour les "False Negative" qui sont des tirs qui sont des buts ayant été prédits comme des tirs manqués. La valeur de chacune de ces variables est donc le nombre d'instances correspondant à la description.

La sensibilité permet de voir la capacité du modèle à prédire correctement les tirs qui seront des buts. De l'autre côté, la spécificité permet de voir la capacité du modèle à prédire correctement les tirs qui ne seront pas des buts. Comme indiqué dans le travail d'Umami, Gutama et Hatta, la spécificité est plus importante que la sensibilité selon le contexte. Dans le cas où le modèle serait utilisé pour prédire un cancer, nous allons chercher à avoir une meilleure 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 0.9672 et une spécificité de 0.1903 pour un seuil 0.02. Ils indiquent finalement que le modèle de xG est plus performant si nous prenons 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]. Son objectif 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. David Sumpter 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. Finalement, 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 produit un résumé de la régression logistique.

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

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 d'ORTEC, d'Immotio et FIFA. En effet, son article scientifique utilise trois datasets 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 (voir tab. 3).

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 d'entre eux. 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 quatre cités est le Random Forest avec une précision de 0.771. Cependant, il n'est pas possible de connaître les meilleurs hypers 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.

5.1 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 majeur en 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. Ce sont essentiellement ces deux facteurs qui sont utilisés pour établir la prédiction. Concernant les autres facteurs, ils servent à améliorer la prédiction, mais rien n'indique s'ils ont autant d'importance que la distance et l'angle.

6 Dataset

6.1 Présentation du dataset

Le dataset qui a été trouvé pour cette thèse est un dataset provenant de Wyscout. Ce dernier est un site web qui fournit des données sur le football. Nous pouvons, 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 utilise un logiciel propriétaire nommé "marqueur" afin de 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 marqueur⁶ 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 est correct et que ces entrées 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

6. Le logiciel propriétaire

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, nous pouvons apercevoir le 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.



FIGURE 2 – Aperçu du logiciel propriétaire de Wyscout

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.

6.2 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 [7](#), il y a :

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

L'attribut "tags" est également important. Il représente 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 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.

7. 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'événements 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. C'est un organisme qui définit les règles du football comme la taille des terrains. Celle-ci pouvant varier entre 90 et 120 mètres de longueur et entre 45 et 90 mètres de largeur. [4]

Une autre information importante indiquée 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. Nous pouvons d'ailleurs constater cela lorsque nous regardons les positions des événements de tirs [4], en effet sur la figure 4, nous pouvons voir que les tirs sont toujours effectués du même côté du terrain⁸. Il est également possible de visualiser la distribution des positions X des tirs sur la figure 5. Nous pouvons apercevoir que les tirs sont toujours effectués dans la même partie du terrain. Les tirs sont relatifs à la position du but de l'équipe qui a généré l'événement. Cela 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 effectués. Ce travail aurait été nécessaire et extrêmement important, car cela aurait pu fausser les données sur la distance et l'angle de tir.

Nous savons 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.

8. Certains sont effectués dans l'autre partie du terrain, mais jamais dans la surface de réparation opposée.

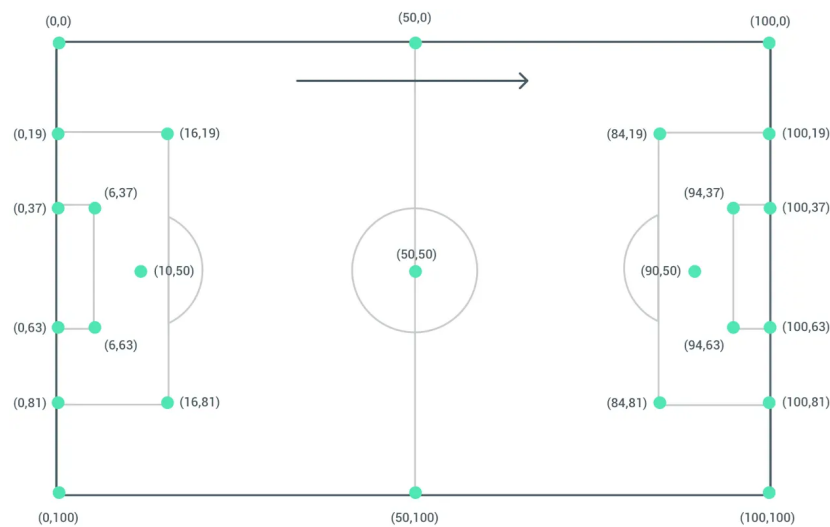


FIGURE 3 – Représentation du terrain de football avec les coordonnées X et Y

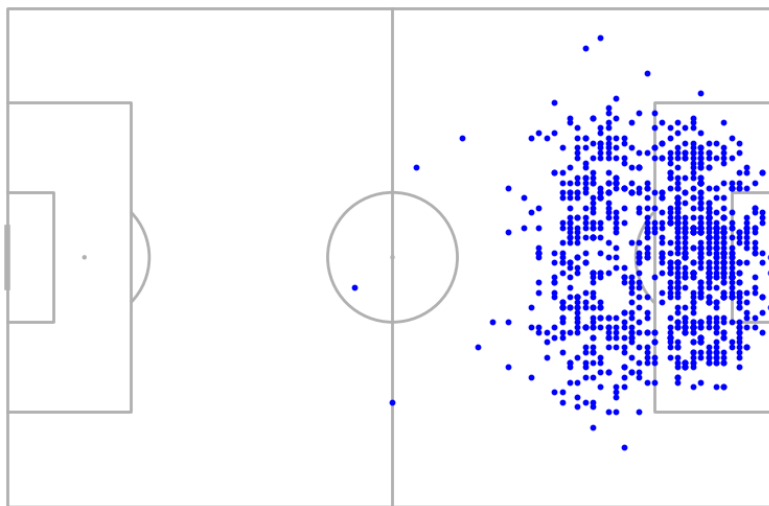


FIGURE 4 – Représentation des positions des 1000 premiers tirs

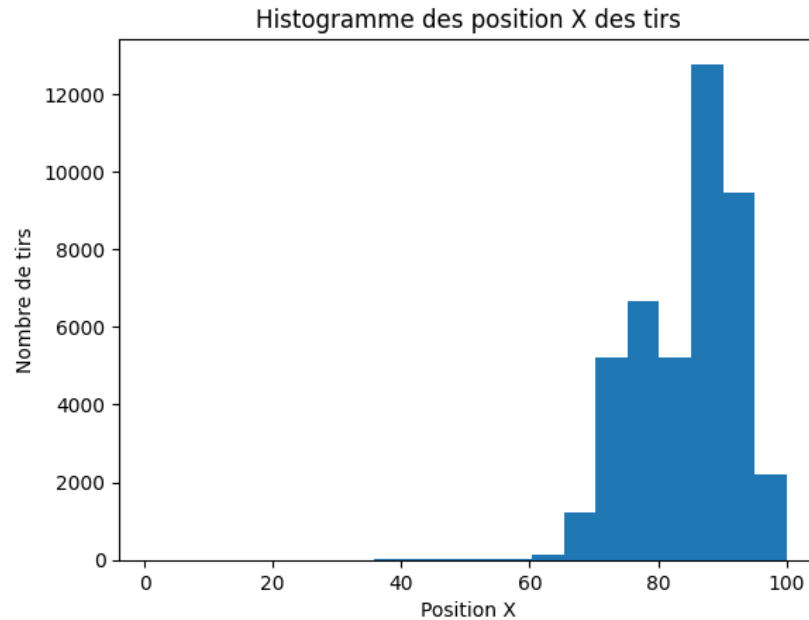


FIGURE 5 – Histogramme des positions X des tirs

6.3 Players

Le dataset contient également des informations sur les joueurs. Le tableau [6](#) décrit les attributs du dataset "Players".

Nous pouvons 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 intéressante dans la suite de la thèse est le pied fort du joueur. En effet, celui-ci 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 sera utilisé pour entraîner le modèle.

Nom de l'attribut	Description
birthArea	Pays de naissance du joueur
birthDate	Date de naissance 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"

6.4 Transformation des données

Comme discuté dans la section [6.2](#), 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. Celle-ci est la plus commune dans le football professionnel. Nous pouvons notamment le voir sur le site officiel de la Premier League [\[3\]](#). Nous démarrons 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 élaborerons 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.

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

Nous passons 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 105⁹.

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.

$$y_{Centre} = |y_{68} - 34| \quad (4)$$

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 Y¹⁰.

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 calculé 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. Nous pouvons voir sur la figure ⁶ que nous avons un triangle rectangle [GDF]. Avec le théorème de Pythagore, nous pouvons savoir que :

$$DE^2 = GF^2 = DG^2 + DF^2 \quad (5)$$

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

9. Dans notre cas, nous soustrayons les coordonnées avant de passer à l'échelle 105x68, donc nous avons soustrait les coordonnées de X à 100.

10. Tout comme 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, nous pouvons calculer l'angle du tir par rapport au but adverse. Nous pouvons 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. Nous pouvons donc avoir la valeur de GH et GK en additionnant ou soustrayant la moitié de la largeur du but à y_{Centre} .

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

En utilisant la tangente inverse, il est possible de calculer l'angle GJK ainsi que l'angle GFH. Avec ces deux angles, nous pouvons calculer l'angle à l'intérieur du triangle JGF pour finalement récupérer l'angle du tir par rapport au but adverse en faisant [11] :

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

Finalement l'équation totale ressemble à cela, une fois toutes les valeurs réunies :

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

Il est d'ailleurs possible d'accéder aux schémas géométriques 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 [12][13].

11. En degrés

12. Angle : <https://www.geogebra.org/geometry/d7tqw4fe>

13. Distance : <https://www.geogebra.org/geometry/fnx3swex>

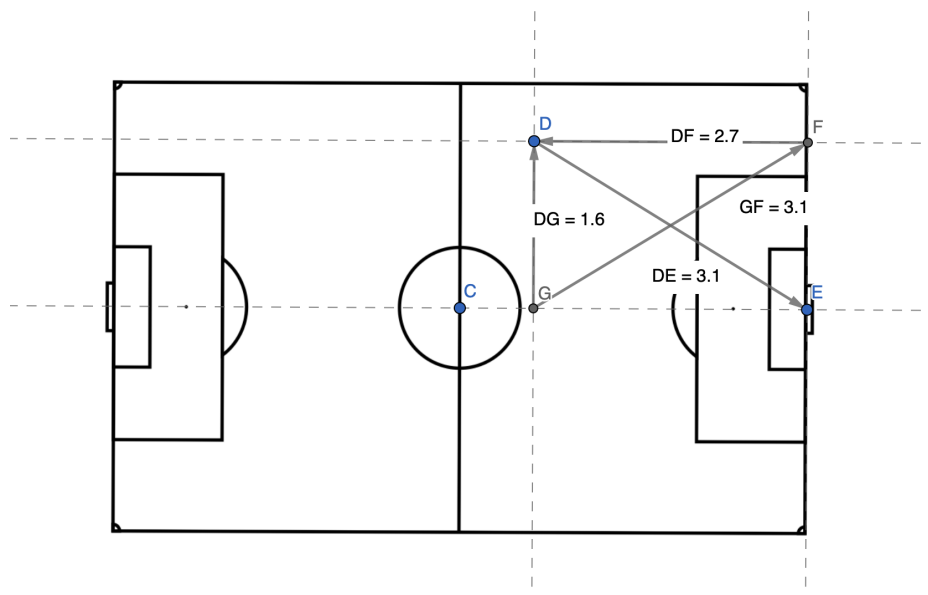


FIGURE 6 – Distance du tir par rapport au but adverse

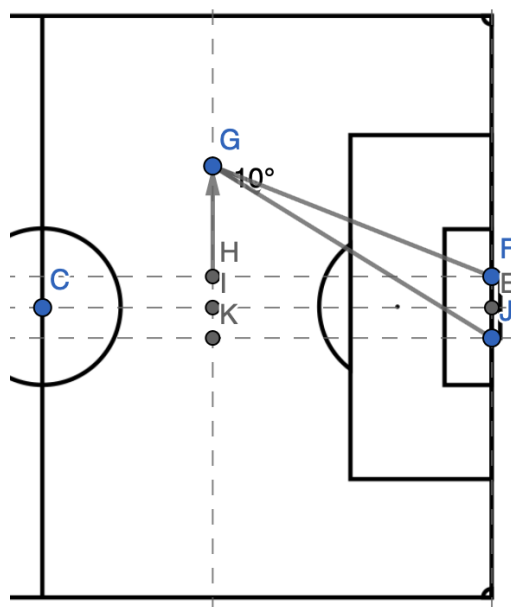


FIGURE 7 – Angle du tir par rapport au but adverse

6.5 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 effectué 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 ces deux attributs pour permettre d'avoir les informations des joueurs dans le dataset events. Dans notre cas, nous allons faire une jointure pour permettre d'identifier 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

6.6 Visualisation des données

Cette section sert à visualiser les données du dataset final. Tout d'abord, nous allons effectuer une description simple du dataset. Le dataset contient 43'075 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. Nous pouvons voir que la distance maximale d'un tir a été de 103.95 mètres et la distance minimale de 0.68 mètre. Nous constatons que certains tirs n'ont pas été normalisés par rapport au sens du jeu de l'équipe comme indiqué dans la section 6.2. 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. Ce dernier est un graphique qui affiche un terrain de football. Grâce à cela, nous pouvons apercevoir plus simplement d'où ont été effectués les tirs. Le but de notre pitch chart est de visualiser les tirs qui ont été effectués à leurs positions respectives. Dans notre cas, nous avons 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. Nous pouvons voir qu'une grande majorité des tirs ont été effectués dans la surface de réparation. Nous sommes amenés à remarquer qu'il y a très peu de tirs à l'entrée de la surface de réparation. Plusieurs hypothèses 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, nous remarquons qu'il y a également des tirs plus lointains. La deuxième hypothèse est que l'origine de ce phénomène

réside dans la méthode de collecte des données. 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.

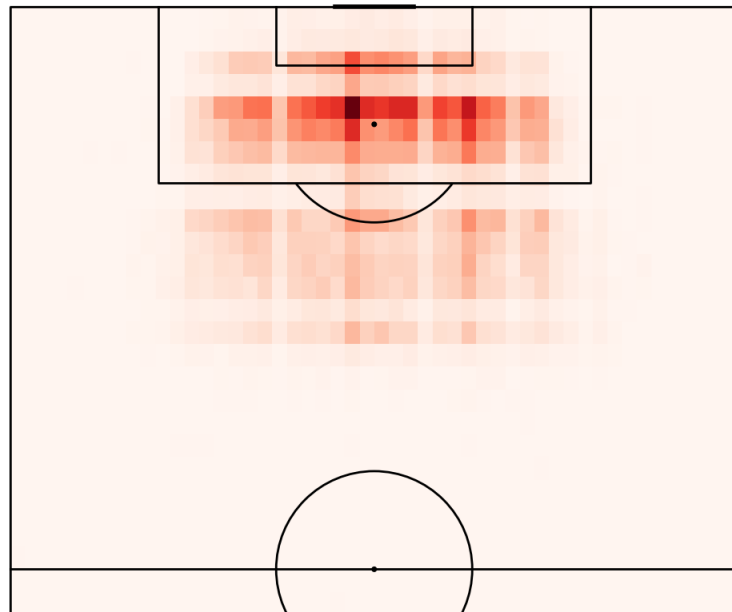


FIGURE 8 – Pitch chart des tirs effectués

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 nous observons les histogrammes, nous nous apercevons 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. Nous pouvons 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 nous aurions pu voir quel attribut aurait été le plus influent directement en observant les données. Cependant, nous pouvons remarquer que des outliers ont été trouvés pour la distribution des différentes variables via les boxplots. Les outliers sont déterminés en se basant sur la règle suivante [9] :

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

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, nous sommes capables de récupérer tous les outliers pour chaque variable, nous obtenons un tableau comme celui-ci (voir tab. 9) : L'objectif est maintenant de visualiser la probabilité de chacun des sets d'outliers. Si nous observons des anomalies ou des données aberrantes, il sera nécessaire de les supprimer.

	Distance	Angle	Angle absolu
Nombre d'outliers	197	2163	70

TABLE 9 – 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, nous pouvons constater 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 corner 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. Nous constatons 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 la valeur d'une variable augmente lorsque la variable corrélée augmente également. Une corrélation élevée suggère une relation claire entre une variable et la variable cible, ce qui peut faciliter la compréhension de l'influence entre les deux variables. Une corrélation élevée peut améliorer la prédiction et faciliter l'interprétation du modèle.

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".

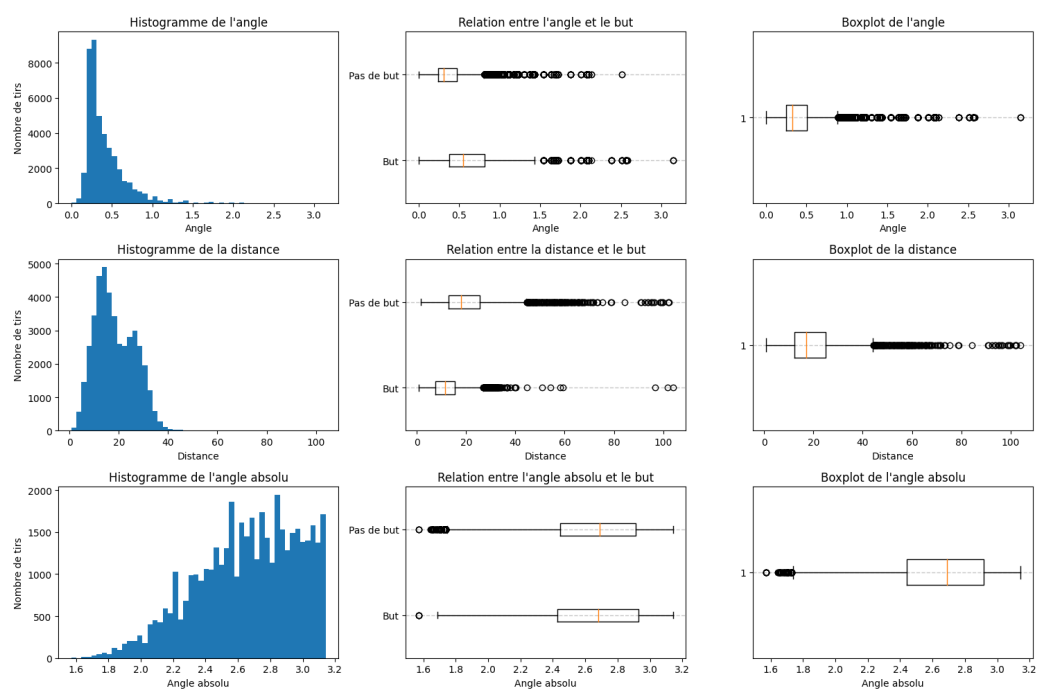


FIGURE 9 – Analyse des attributs quantitatifs

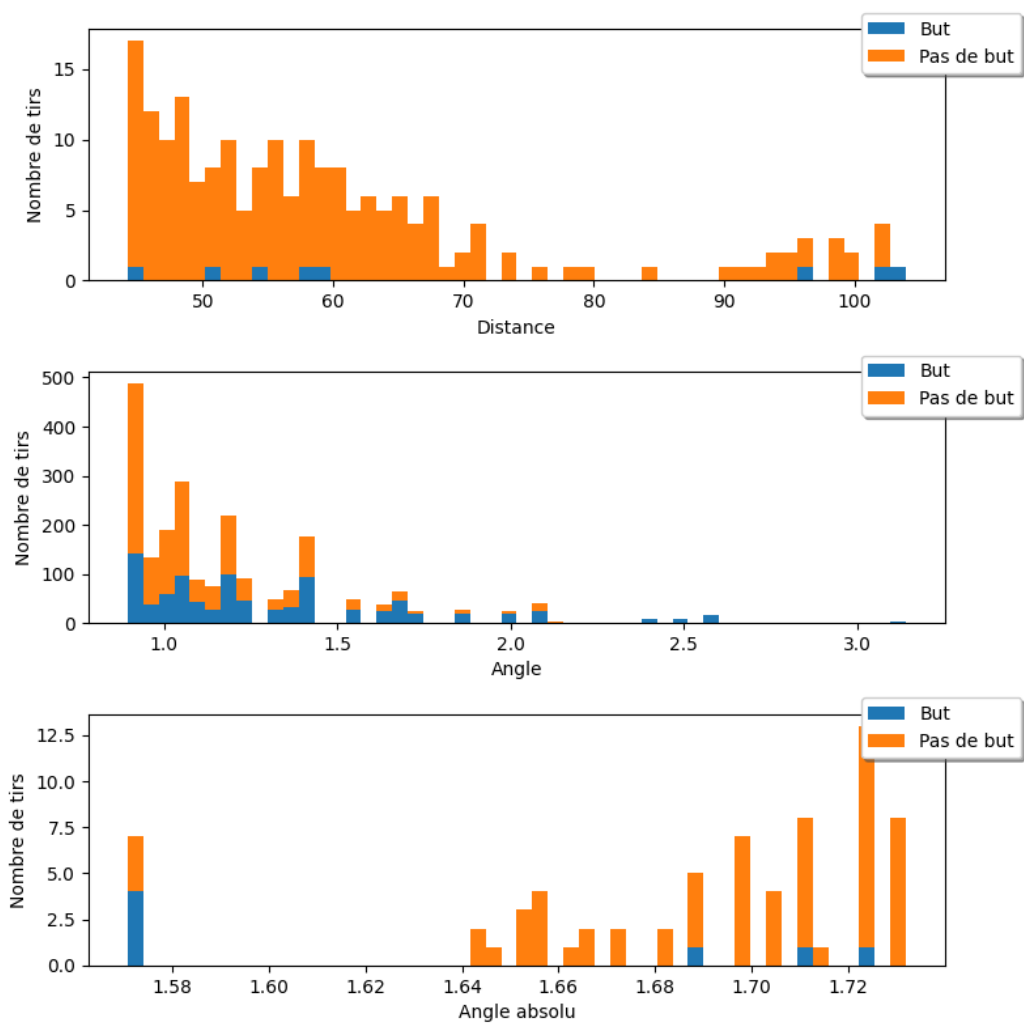


FIGURE 10 – Analyse des outliers

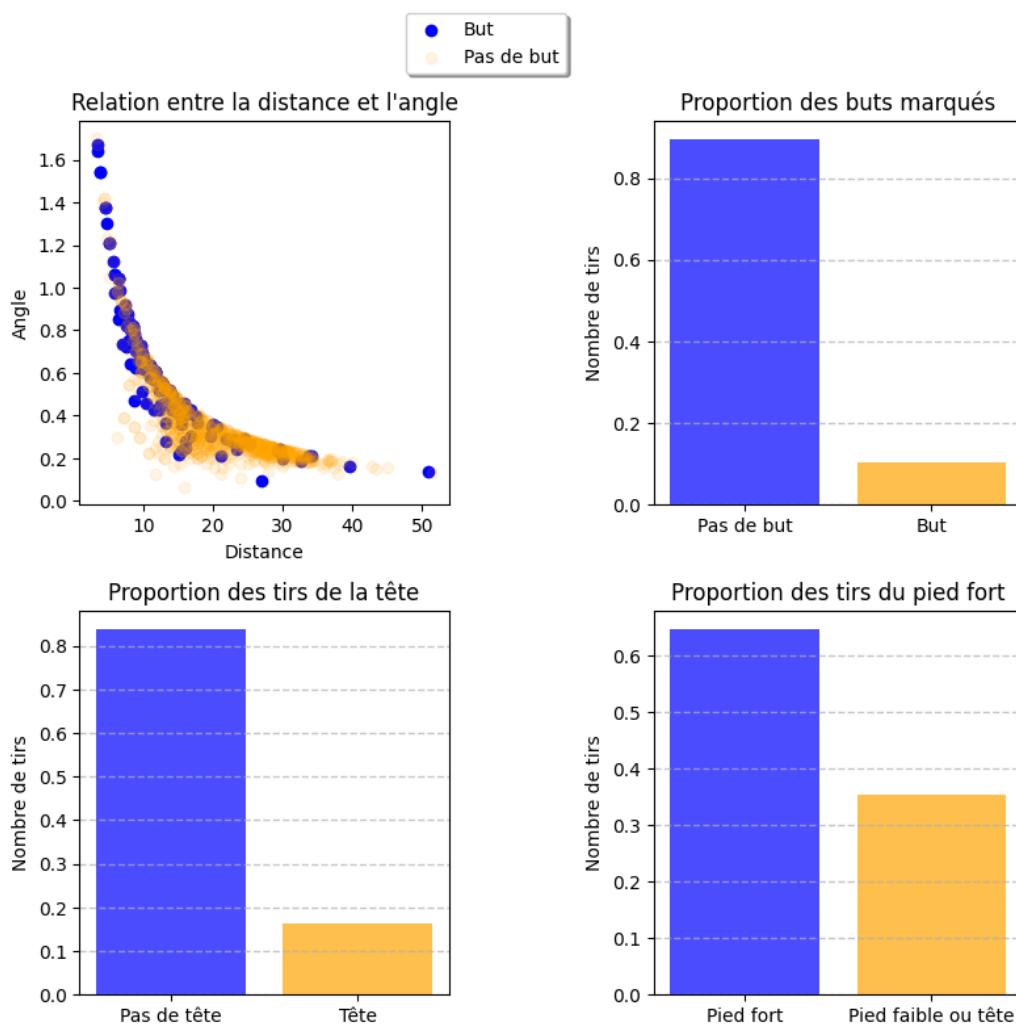


FIGURE 11 – Relation distance-angle et analyse des attributs qualitatifs

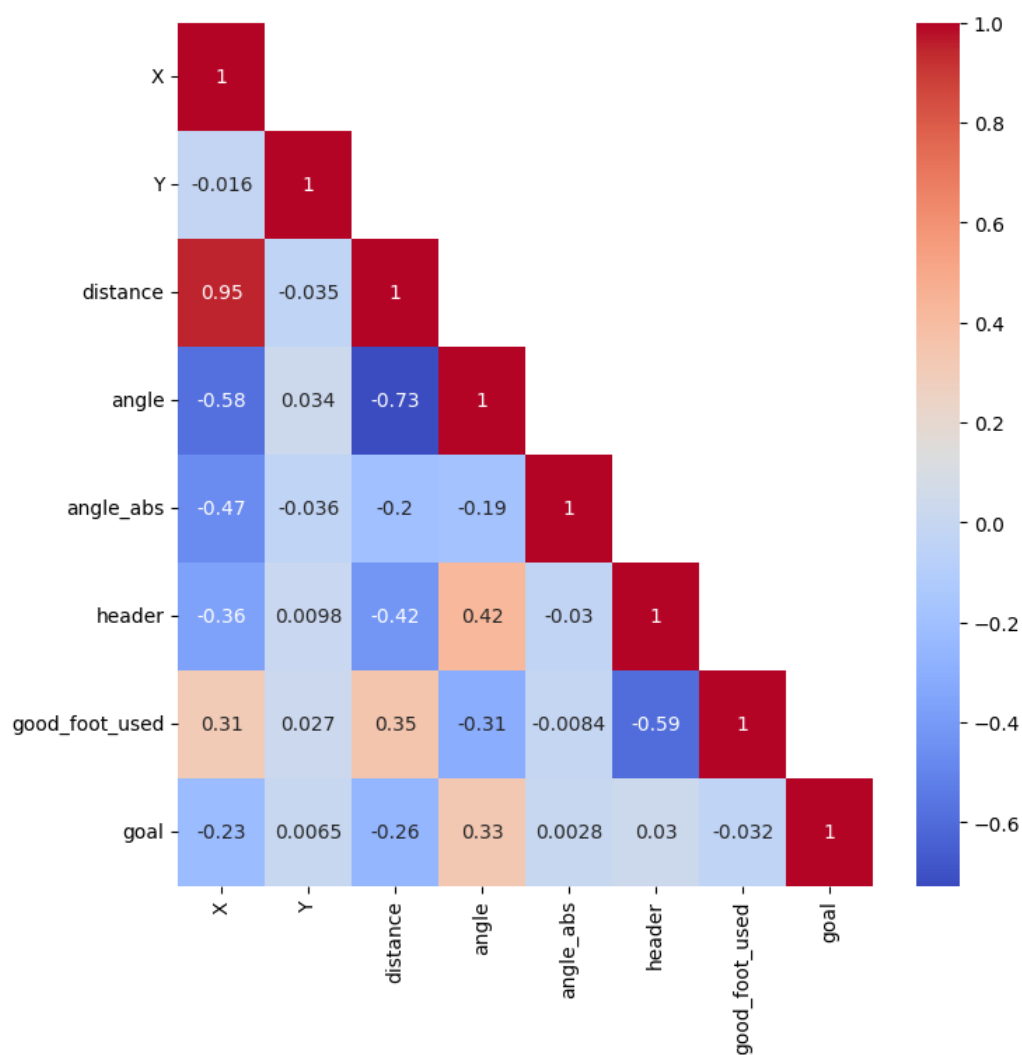


FIGURE 12 – Matrice de corrélation des attributs

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 essentiel de noter que la variable "distance" présente une corrélation significative avec la variable "X". Cette corrélation découle du fait que le calcul de l'attribut "distance" repose sur les coordonnées horizontales "X" et "C". Par conséquent, toute modification de la valeur de l'attribut "X" a un impact direct sur 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.

7 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 test sera utilisé pour évaluer la performance des modèles finaux. Tandis que le jeu d'entraînement, nous permettra d'entraîner notre modèle et de choisir le meilleur set d'hypers 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. Ayant environ 43'000 tirs enregistrés, cela nous donne approximativement 38'700 tirs pour le jeu d'entraînement et 4'300 tirs pour le jeu de test. Ainsi, nous avons un jeu de données d'entraînement suffisamment vaste pour entraîner nos modèles et un jeu de test de taille adéquate pour évaluer leur performance.

7.1 Choix des modèles

Plusieurs modèles vont être utilisés 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. Il est important de rappeler 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)

Le choix de ces modèles est basé sur les modèles les plus utilisés pour des problèmes de classification. Nous avons choisi ces modèles, car ils sont tous différents et permettent de tester différentes approches pour résoudre le problème. De plus, pour des questions de puissance de calcul, nous avons choisi des modèles qui ne sont pas trop complexes permettant d'être exécutés sur un ordinateur personnel.

7.2 Cross Validation

Pour chaque modèle, nous allons utiliser la cross validation pour choisir le meilleur set d'hypers paramètres. Elle représente 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 nous pouvons 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 (celui de "Validation"). 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. Celle-ci est utilisée dans des contextes de classification probabiliste [14](#) permettant d'évaluer la performance d'un modèle de classification. Ainsi, nous pouvons choisir le meilleur set d'hypers paramètres pour chaque modèle.

Après avoir choisi nos modèles et leurs hypers paramètres, nous allons les entraîner sur le jeu de données d'entraînement entier et les tester sur le jeu de données de test final. 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 nous pourrions déterminer quel modèle est le plus performant pour résoudre notre problème.

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

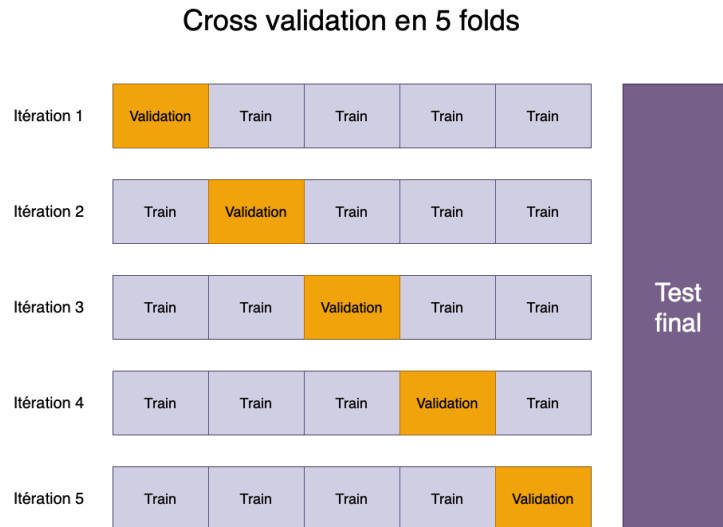


FIGURE 13 – Cross Validation avec 5 sous-ensembles

7.3 Sélection des meilleurs attributs

Pour la sélection des meilleurs attributs, nous allons utiliser la méthode de forward elimination. Elle consiste à tester tous les attributs un par un en les ajoutant au fur et à mesure au modèle et à 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](#), nous sommes amenés à 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](#), nous pouvons 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. Nous pouvons le remarquer d'ailleurs lorsque nous observons 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](#), nous ajoutons l'attribut "dis-

tance" au modèle. Cependant, nous avons supprimé l'attribut "Y" qui n'était pas significatif. Nous constatons 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](#), nous ajoutons l'attribut "angle" au modèle. Cependant, dans ce cas-ci, nous observons 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 leur lien 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. Cela est 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, nous ajoutons l'attribut "angle_abs" au modèle. Ici, nous pouvons apercevoir 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, nous pouvons 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 lorsque nous les additionnons 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 – Résultats 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 – Résultats 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 – Résultats 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 – Résultats 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 – Résultats 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 – Résultats de la régression logistique - Dernière itération

Pour conclure cette partie de sélection d'attributs, nous constatons que les attributs les plus significatifs parmi ceux que nous avons sélectionnés 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

Nous pouvons donc en conclure que les attributs que nous avons sélectionnés sont pertinents pour la prédiction de la réussite d'un tir. Il est également possible de voir que les attributs que nous avons sélectionnés sont cohérents avec les résultats de la littérature. En effet, nous retrouvons 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, nous avons pu répondre à la première question de la thèse :

Quels sont les paramètres qui influencent le plus l'expected goal ?

7.4 Jeu d'hypers paramètres et sélection

Pour le choix des hypers 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'hypers paramètres différent et passera par une validation croisée. Chaque résultat de validation croisée sera ensuite comparé pour trouver le meilleur jeu d'hypers paramètres pour chaque modèle. Comme indiqué dans la partie [7.2](#), nous utiliserons une validation croisée à 5 folds pour chacun des modèles en utilisant la métrique du log loss. Nous mettrons la méthode en grille pour trouver les meilleurs hypers paramètres pour chacun des modèles. Il est possible de trouver les hypers 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 maximales pour arriver à une convergence
solver	liblinear	L'algorithme utilisé pour résoudre le modèle

TABLE 16 – Jeu d'hypers 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'hypers 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	50	Nombre d'instances minimales acceptée en tant que feuille.
min_samples_split	10	Nombre d'instances minimales pour créer une séparation sur un noeud
n_estimators	50	Nombre d'arbres dans la forêt

TABLE 18 – Jeu d'hypers 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'hypers paramètres choisi pour KNN

Perceptron multi-couche		
Hyper paramètre	Valeur	Description
activation	logistic	Type de fonction d'activation
alpha	0.001	La force de la régularisation (pénalité)
hidden_layer_sizes	(100, 100, 100)	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'hypers paramètres choisi pour le perceptron multi-couche

8 Résultats

8.1 Comparaison des performances sur la log loss

Cette section vise à discuter des résultats. En effet, après avoir effectué la validation croisée sur tous les modèles pour trouver leurs meilleurs hypers 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	Log loss
Régression logistique	0.2882
Perceptron multi-couche	0.2885
Random Forest	0.2887
Arbre de décision	0.2903
KNN	0.4304
Classificateur naïf	3.8152

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 représente 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 perceptron multi-couche le suit de près. La différence entre ces deux modèles est très faible, ce qui indique qu'ils ont des performances similaires.

Le Random Forest 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, nous pouvons déduire que ce modèle n'est pas approprié pour ce problème.

8.2 Visualisation de la distribution des prédictions

Par la suite, il a été intéressant d'observer la distribution des résultats prédits sur les données de test. Cela nous permet de constater les différences parmi les modèles qui sont similaires en terme 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 cela est basé sur les données de test seulement. Autrement, nous pouvons 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-couche en terme de log loss, mais qui possède une distribution de résultats prédits différente.

Finalement, nous observons que le classificateur naïf prédit systématiquement la classe majoritaire¹⁵, ce qui était prévisible puisque cela représente le principe de ce classificateur.

15. Dans notre cas, elle prédit que le tir a été manqué et donc une probabilité de marquer à 0%

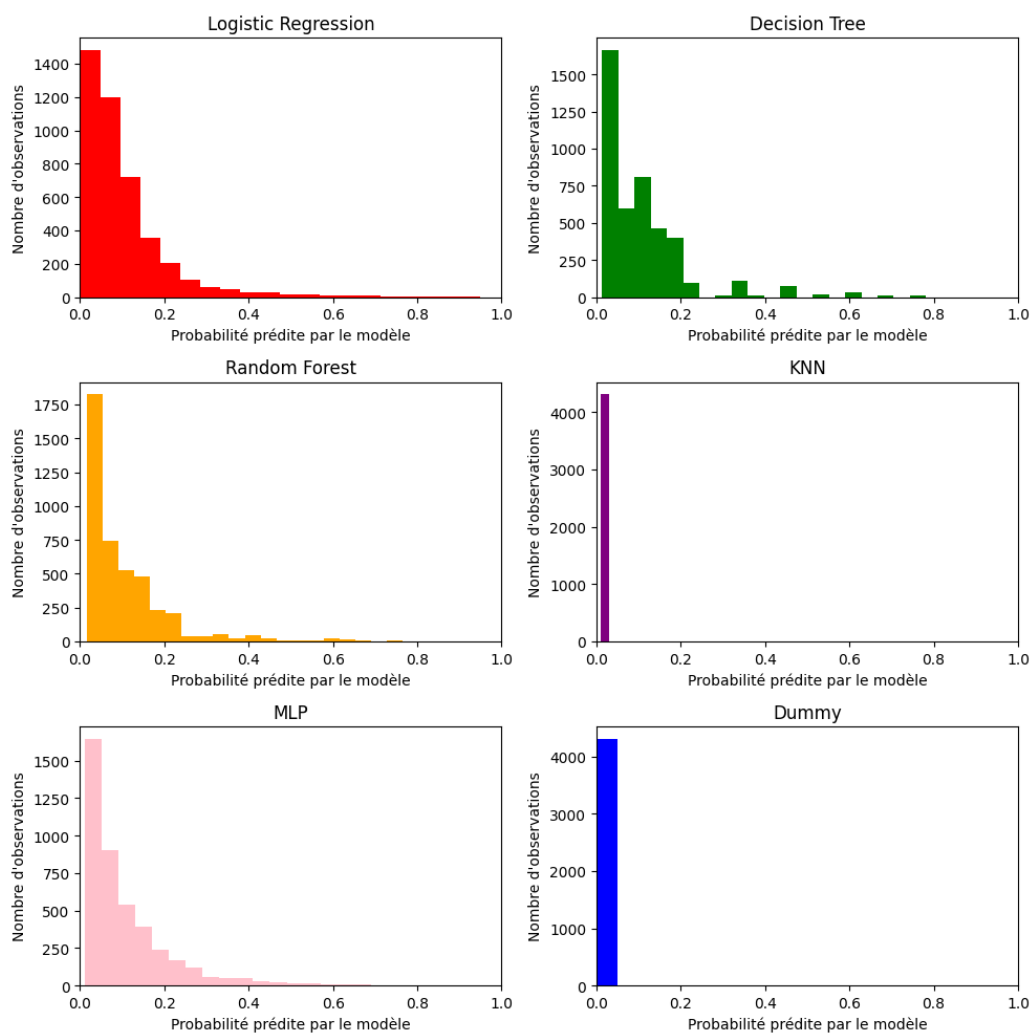


FIGURE 14 – Distribution des résultats prédits sur les données de test

8.3 Visualisation des résultats sur un terrain

Pour ajouter une nouvelle manière d'analyser et de visualiser les résultats, nous avons décidé de créer une heatmap pour chacune des positions du terrain de football. À partir de cela, nous calculerons la distance et l'angle. Nous avons également indiqué que tous les tirs effectués à chacune des positions n'ont pas été effectués de la tête et qu'ils ont été effectués avec le pied fort du joueur. Nous sommes amenés à voir que la majorité des tirs ont été effectués avec le pied fort du joueur et qu'ils n'ont pas été effectués 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.

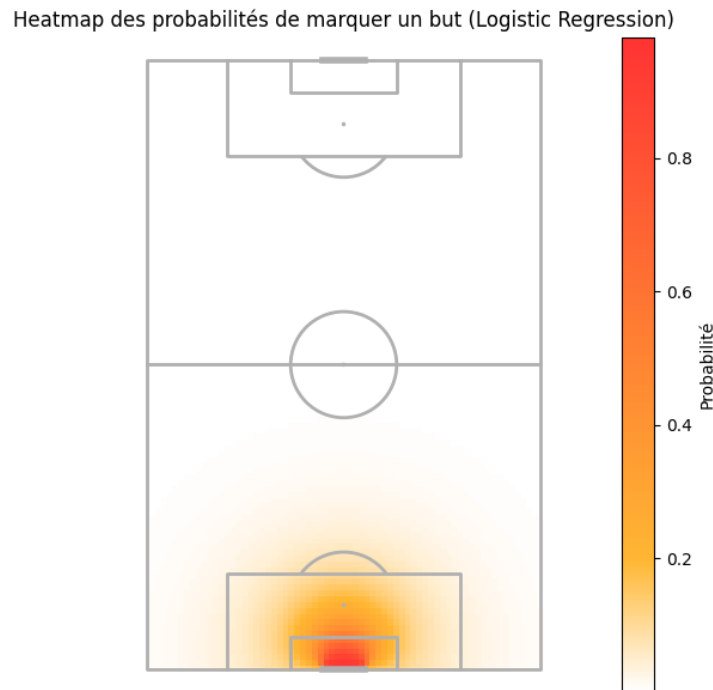


FIGURE 15 – Heatmap des résultats prédits par la régression logistique

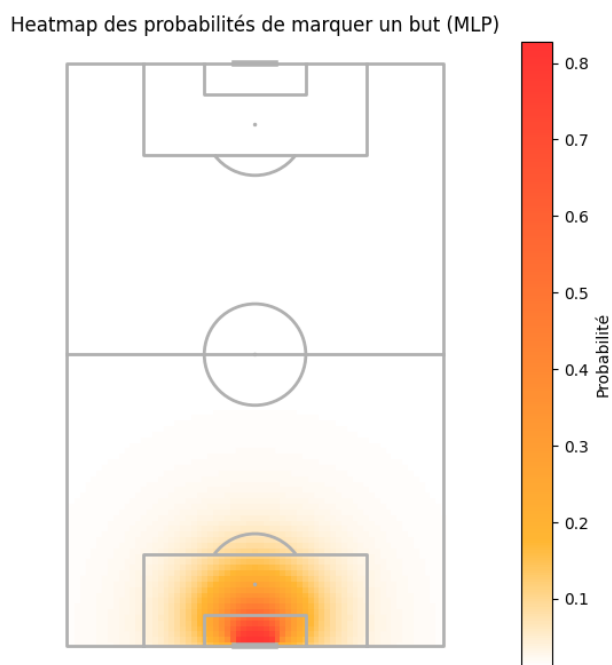


FIGURE 16 – Heatmap des résultats prédits par le perceptron multi-couche

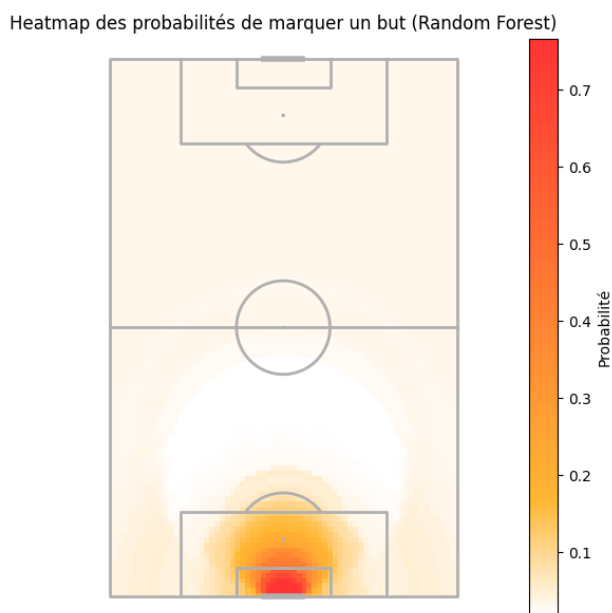


FIGURE 17 – Heatmap des résultats prédits par le modèle Random Forest

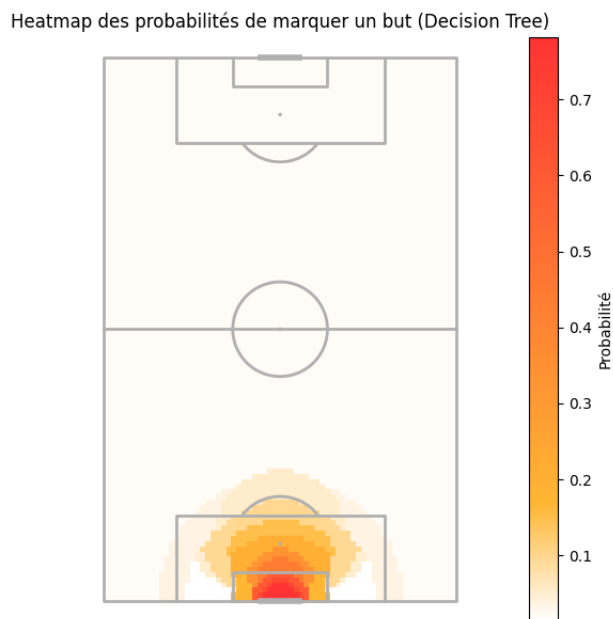


FIGURE 18 – Heatmap des résultats prédits par l'arbre de décision

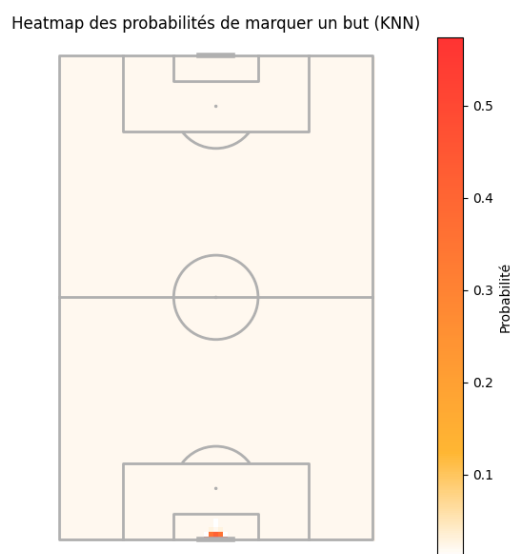


FIGURE 19 – Heatmap des résultats prédits par le modèle KNN

Lorsque nous nous penchons sur les figures [15](#), [17](#), [16](#), [18](#), nous constatons 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 transformés en but que ceux effectués à l'extérieur de la surface. Cependant, nous pouvons constater plusieurs phénomènes. Tout d'abord, nous nous apercevons que le modèle Random Forest prédit que les tirs effectués du côté du terrain de l'équipe attaquante ont une plus grande chance d'être marqués que ceux qui sont effectués 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. Ce phénomène peut être dû à un manque de tirs faits à très longue distance des modèles. En effet, lorsque nous observons la figure [8](#), nous constatons que les tirs sont majoritairement effectués dans la surface de réparation adverse et dans le camp adverse. Cela explique pourquoi ces deux modèles ont une bonne prédiction lorsque le tir est effectué dans ces zones. Cependant, nous pouvons voir que les tirs effectués dans le propre camp sont très peu nombreux. L'hypothèse est que ces modèles ont bien appris sur la majorité des tirs, mais que le manque de tirs effectués dans le propre camp a fait que ces deux modèles ont eu plus de difficultés à effectuer des prédictions correctes pour ces tirs.

Ensuite, nous pouvons voir que le modèle KNN prédit de manière incorrecte les probabilités. Nous avons été en mesure de le voir via sa log loss qui était relativement élevée par rapport aux autres modèles. Nous pouvons également le voir sur la figure [19](#) où nous constatons 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 nous pouvons constater et comparer les différences entre les modèles. Nous nous apercevons que les deux modèles les plus performants sont la régression logistique et le perceptron multi-couche. La régression logistique a une meilleure log loss que le perceptron multi-couche. Nous voyons notamment que, pour le modèle Random Forest, sa visualisation en heatmap fournit des résultats incohérents à très longue distance, contrairement au perceptron multi-couche et à la régression logistique. Nous avons vu via la tendance des prédictions des modèles que la chance de réussite d'un tir augmente plus le tir est effectué proche du but adverse. Sur

la figure 8, nous voyons que les joueurs professionnels de football ont déjà connaissance que leurs chances de tirs augmentent plus ils sont proches du but adverse.

Pour conclure cette section sur l'analyse des résultats, nous pouvons 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'hypers paramètres. Il est donc possible qu'un autre modèle soit plus performant que la régression logistique pour la résolution de ce cas.

9 Conclusion

9.1 Conclusion générale

Nous allons maintenant discuter la conclusion finale de ce travail. Tout d'abord, nous allons nous pencher sur la première question du travail qui était de savoir quels étaient les facteurs qui influençaient le plus la réussite d'un tir. Via nos recherches sur le sujet, nous avons pu établir un état de l'art sur le sujet. Nous avons pu observer, via cet état de l'art, que les facteurs les plus influents étaient la distance du tir ainsi que l'angle du tir. Ils sont d'ailleurs indissociables puisque le cumul des deux apporte une information plus pertinente que chacun d'eux pris séparément. L'un des articles scientifiques a également essayé d'introduire un facteur supplémentaire indiquant si le tir a été effectué de la tête. Cependant, rien n'est indiqué si ce facteur a une réelle influence sur la réussite d'un tir.

Dans notre travail, nous avons pu constater et confirmer que la distance et l'angle du tir étaient les facteurs les plus influents. En effet, ces deux facteurs sont corrélés (voir fig. 9) et ont une réelle influence sur la réussite d'un tir. Nous avons fait le travail de sélection des meilleurs attributs pour nos modèles dans la section 7.3. Cette recherche a consisté à faire une "forward elimination" sur les attributs de notre dataset. Le principe de cette méthode est de commencer avec un modèle vide et d'y ajouter un attribut à la fois. Ensuite, nous observons si l'ajout de cet attribut améliore les performances du modèle. Dans notre cas, nous observons si la p-value de l'attribut était inférieure à 0.05. Si c'était le cas, nous gardions l'attribut et nous recommandons l'opération avec un autre attribut. La p-value est une valeur qui indique si l'attribut est significatif pour le modèle. Dans notre cas, ce test a été fait avec une régression logistique. Lorsque la "forward elimination" a été faite, nous avons pu constater que les attributs les plus influents étaient :

- La distance du tir
- L'angle du tir
- Si le tir a été effectué de la tête
- Si le tir a été effectué avec le pied fort du joueur

Nous pouvons d'ailleurs le voir sur le tableau 15. Ce dernier affiche les résultats de la régression logistique avec les attributs sélectionnés. On peut y observer les p-values de chaque attribut ainsi que leurs coefficients. C'est donc ainsi que nous avons pu répondre à la première question de ce travail.

Ensuite, nous avons pu répondre à la deuxième question de ce travail qui était de savoir quel modèle est le plus performant pour résoudre ce problème. Pour répondre à cette question, il a fallu établir une bonne méthodologie.

Dans notre cas, nous avons décidé de faire une séparation du dataset en deux parties. La première partie était utilisée pour l'entraînement des modèles et la deuxième partie était utilisée pour tester les modèles. L'entraînement des modèles consistait à faire une validation croisée avec 5 folds pour trouver les meilleurs hypers paramètres pour chaque modèle. Ensuite, nous avons pu tester les modèles avec leurs meilleurs hypers paramètres trouvés sur la deuxième partie du dataset.

Les résultats sur la deuxième partie du dataset nous ont permis de comparer les modèles entre eux. Nous les avons comparés via la log loss et via la visualisation des probabilités prédites par les modèles. Nous avons pu constater des choses intéressantes sur les résultats. Tout d'abord, nous avons pu constater, via le tableau [21](#), que les 3 modèles les plus performants selon la log loss étaient la régression logistique, le perceptron multi-couche et le modèle Random Forest. Cependant, après avoir visualisé les probabilités prédites par les modèles sur un terrain de football, nous avons pu constater que le modèle Random Forest fournissait des résultats incohérents à longue distance (voir fig. [17](#)). Les deux autres modèles précédemment cités fournissaient des résultats plus cohérents par rapport à la réalité (voir fig. [15](#) et [16](#)).

Pour finir, nous pouvons dire que la régression logistique est le modèle le plus performant pour résoudre ce problème de prédiction de la réussite d'un tir. Ce modèle est utilisé avec la distance, l'angle, l'information du tir effectué de la tête et du pied fort.

9.2 Améliorations possibles

En terme d'améliorations possibles, nous pouvons dire que ce travail peut être amélioré de plusieurs manières. Tout d'abord, nous pourrions améliorer en ajoutant de nouveaux attributs. En effet, nous pourrions ajouter davantage de données pour apporter des détails supplémentaires sur le tir. Par exemple, il serait intéressant d'ajouter la hauteur du ballon au moment du tir, le type d'attaque effectuée (contre-attaque, attaque placée, etc.), le type de passe effectuée (passe en profondeur, passe en retrait, etc.), ou encore ajouter des informations quant à la qualité du joueur à terminer les actions. Effectivement, un défenseur a, en théorie, une moins bonne qualité de finition qu'un attaquant. Cette différenciation pourrait également se faire entre les différents attaquants. Par exemple, Cristiano Ronaldo a la capacité de marquer plus de buts dans des situations difficiles qu'un attaquant de Ligue 2. Il serait également envisageable d'ajouter le même type d'informations pour le gardien de but.

Ensuite, nous pourrions tester et comparer d'autres modèles non testés ici. Par exemple, nous pourrions tester un modèle de classification basée sur SVM. Finalement, nous pourrions également expérimenter d'ajouter de nouveaux hypers paramètres pour les modèles testés. En effet, dû à un manque de puissance de calcul et de temps, nous n'avons pas pu tester tous les hypers paramètres possibles pour chaque modèle.

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. (1).