

**Rapport projet « paris sportif »**

**Bet-py**

**Promotion Data Scientist décembre 2020**

**Groupe Projet : Matthieu Morand & Romain Miclo**

**Tuteurs : Maxime, puis Théophile à partir de juillet**

Contents

[1. Introduction 2](#_Toc80890379)

[Objectif du projet 2](#_Toc80890380)

[Données de départ 2](#_Toc80890381)

[2. Exploration et visualisation des données 3](#_Toc80890382)

[Présentation des données 3](#_Toc80890383)

[Visualisation des données 5](#_Toc80890384)

[3. Nettoyage, mise en forme et choix des features 9](#_Toc80890385)

[Première itération complète 9](#_Toc80890386)

[Seconde itération 10](#_Toc80890387)

[Définition jeux train / test et standardisation des données 12](#_Toc80890388)

[4. Modèle : création, application et évaluation 13](#_Toc80890389)

[Une première itération complète 13](#_Toc80890390)

[Seconde itération avec nouvelles features 14](#_Toc80890391)

[Conclusion du projet 19](#_Toc80890392)

[Rappel des fichiers à consulter 19](#_Toc80890393)

# Introduction

## Objectif du projet

L’objectif initial de ce projet est de tenter de battre les algorithmes des bookmakers sur l’estimation de la probabilité d’une **équipe gagnant un match**.

Ce projet permet de traiter l’ensemble des étapes d’un projet de Data Science :

* Exploration et visualisation des données
* Nettoyage et mise en forme des données
* Définition et choix des features
* Création, application et évaluation de modèles

## Données de départ

Pour ce projet, deux jeux de données étaient proposés :

* Un premier sur le football :

<https://www.kaggle.com/ayotomiwasalau/club-football-event-data?select=match.csv>

* Un second sur le tennis avec les matchs ATP :

<https://www.kaggle.com/edouardthomas/atp-matches-dataset>

Après avoir regardé le jeu de données, nous avons **choisi de partir sur le football** en estimant qu’il y avait un plus grand nombre de données et donc normalement la possibilité d’aller « plus loin » dans ce sujet de « battre les bookmakers ».

*Remarque :* Il est à noter que nous avions démarré le projet à 3, mais l’un des membres a malheureusement dû arrêter la formation et nous avons donc réalisé ce sujet en binôme.

**Les différents notebooks et fichiers réalisés pendant le projet sont cités au fur et à mesure de ce rapport. A la fin de la conclusion, vous avez également le récapitulatif des fichiers et des emplacements de ces derniers, si vous souhaitez les consulter dans notre espace Git.**

# Exploration et visualisation des données

La première phase du projet a eu pour but de décrire les étapes majeures d’exploration et visualisation des données.

L’archive de départ contenait les 5 fichiers suivants avec les différentes colonnes :

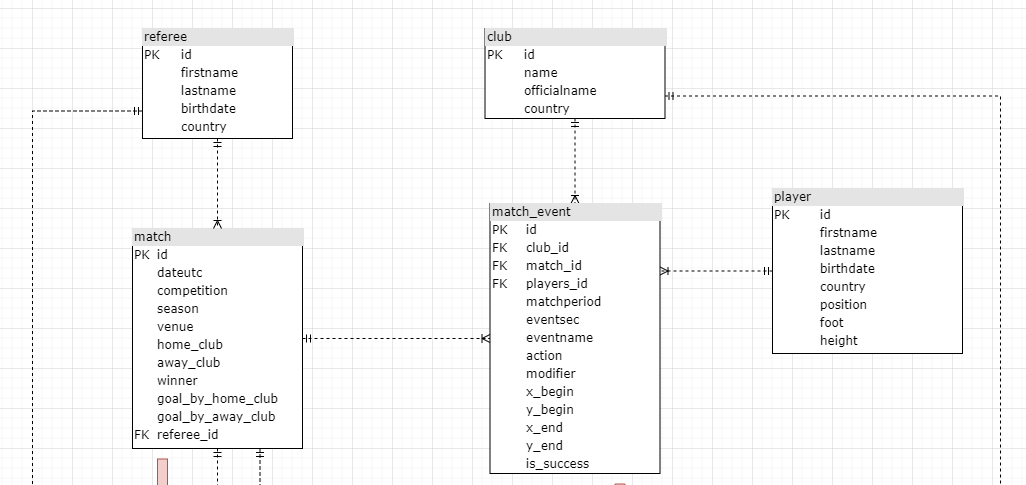


Figure 1: Vue DrawIO des 5 tables de départ fournies

Nous avons rapidement statué qu’à priori « seules » les tables « match » et « match\_event » allaient pouvoir nous aider.

La table « match » contient les informations principales résumant le match et surtout **notre label qui est le club gagnant**. La table « match\_event » représente quant à elle le cœur des informations disponibles avec des millions de lignes contenant le détail des matchs et permettant donc de créer potentiellement un grand nombre de features.

## Présentation des données

Les données fournies présentent les résultats de la saison 2017-2018 des 5 championnats majeurs de football en Europe. Seuls les résultats avec des victoires apparaissent. C’est un point majeur car **les matchs nuls n’y sont donc pas**, c’est pour cela que la base ne contient pas tous les résultats de la saison.

Voici les championnats représentés avec le nombre de matchs associés (on peut par exemple en déduire qu’en plus d’avoir un nombre de clubs moins important en Bundesliga, il y a eu beaucoup de matchs nuls dans le championnat allemand) :

Une image contenant carte

Description générée automatiquement

Figure 2: Vue Power BI du nombre de matchs dans la base sur la saison dans les 5 championnats

La table « match » contient les informations de 1 339 matchs et la table « match\_event » contient les informations d’environ 2,845M event sur ces matchs.

Au-delà des 5 pays où se jouent les matchs, les joueurs proviennent de nombreux pays différents dont une très grand majorité en Europe puis un grand nombre en Amérique du Sud et en Afrique :

Une image contenant carte

Description générée automatiquement

Figure 3: Vue Power BI - nationalité des joueurs ayant joué au moins un match

Cette information n’a pas été retenue pour le modèle mais pourrait être testée (voir si une équipe avec un grand nombre de nationalités ou certaines nationalités est plus performante statistiquement qu’une autre par exemple).

## Visualisation des données

Afin de prédire une victoire ou une défaite, une information essentielle est le nombre de buts marqués/encaissés, voici la répartition des buts marqués à domicile et à l’extérieur sur les jeux de donnés (cf. Figure 4).

On peut voir que le nombre de buts marqués à l’extérieur est plus faible sur chaque modalité qu’à l’extérieur (par exemple 19% au lieu de 25% des matchs avec 2 buts). Le fait de recevoir ou de se déplacer est donc déjà une feature importante à prendre en compte dans le modèle.

Afin de tester les futurs modèles, il est important de repérer les clubs ayant le plus de victoires dans leurs championnats respectifs afin de vérifier si l’algorithme se comporte bien pour ces clubs majeurs, tels que la Juventus en Italie, le PSG en France ou Manchester City en Angleterre (cf. Figure 5, 3 premiers clubs en abscisse).

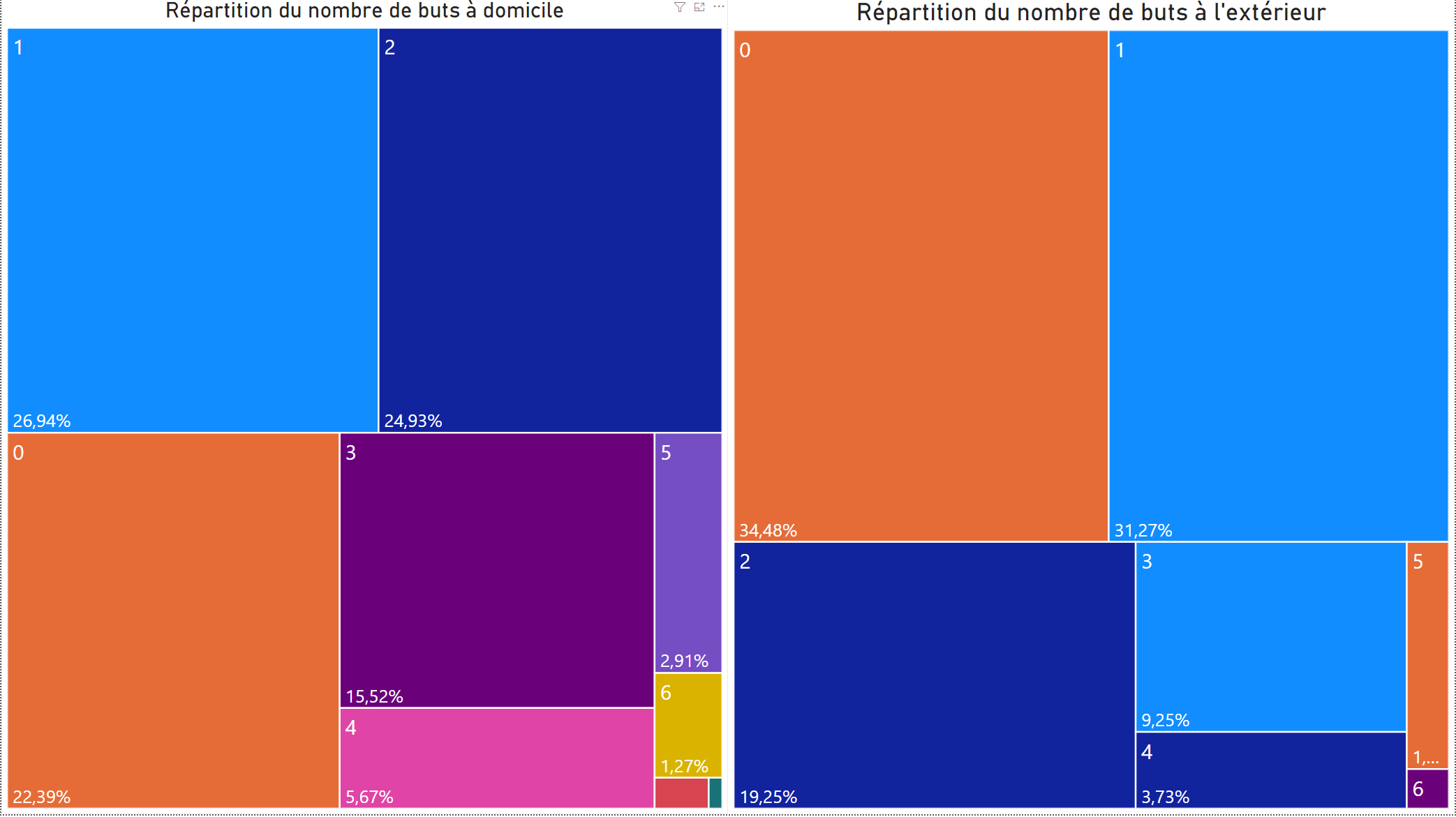


Figure 4: vue Power BI - répartition du nombre de buts marqués à domicile et à l'extérieur

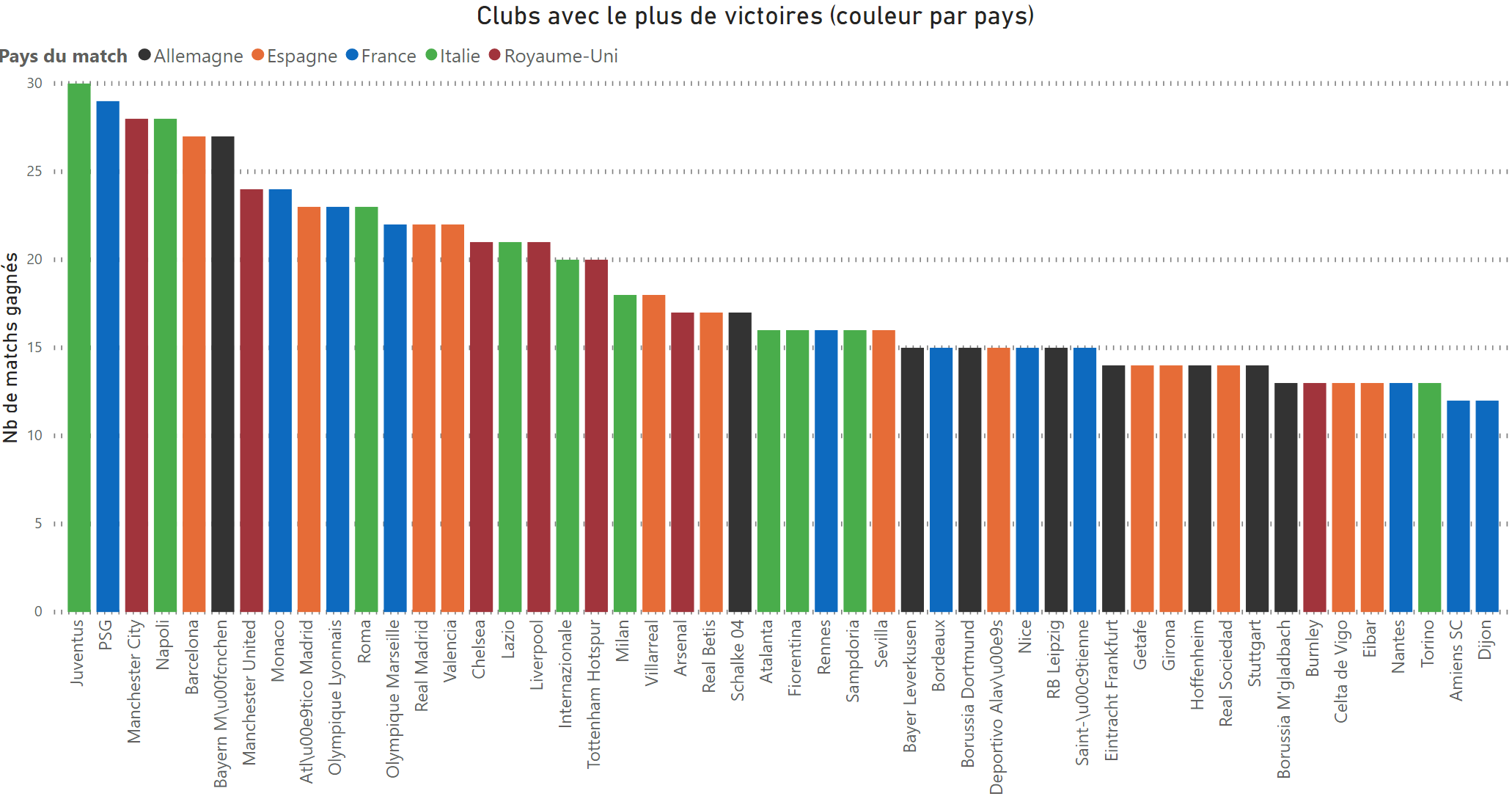


Figure 5: vue Power BI - clubs avec le plus grand nombre de victoires dans la saison, selon leur championnat

Le notebook « Bet-Py\_Notations.ipynb » contient un grand nombre de visualisation de données concernant la table match\_event selon :

* L’ensemble des events différents selon leur thématique (passes, duel, etc.)
* Une analyse spécifique sur un match par exemple

Sur le Notebook, les différents types d’event ont été étudiés selon les actions possibles selon le thème.

Une image contenant texte

Description générée automatiquement

Figure 6: différents évènements possibles et les actions associées

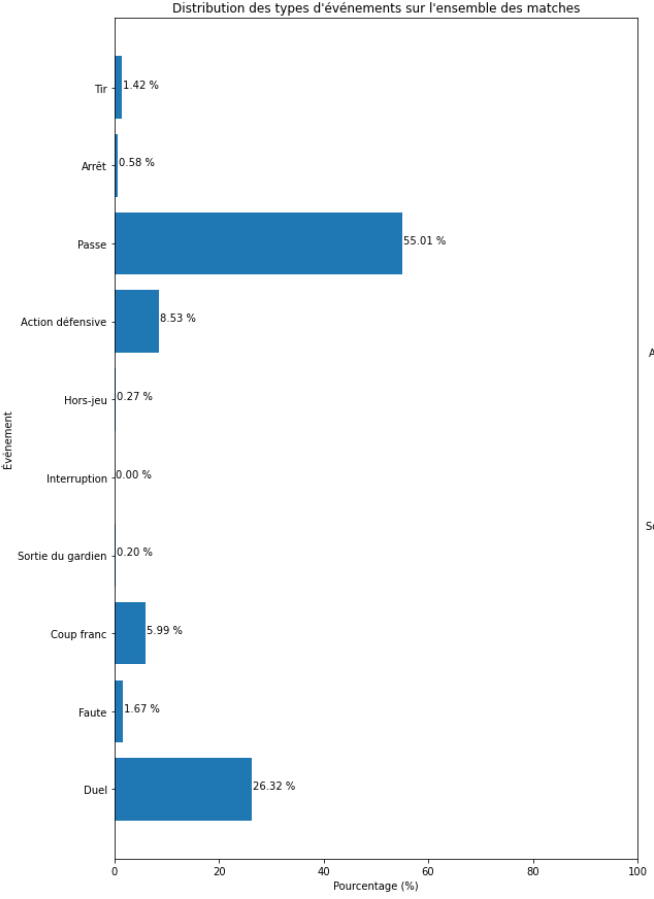
Nous pouvons par exemple regarder la distribution des différents events enregistrés dans la base avec plus de la moitié qui concernent des passes (55%) :

Figure 7: répartition du nombre d'évènements selon leur type dans « match\_event »

Le nombre d’évènements selon le poste du joueur peut aussi être une bonne indication. En effet, une équipe qui en proportion a beaucoup plus d’évènements avec ses attaquants doit avoir plus de chance de marquer et donc de gagner le match :

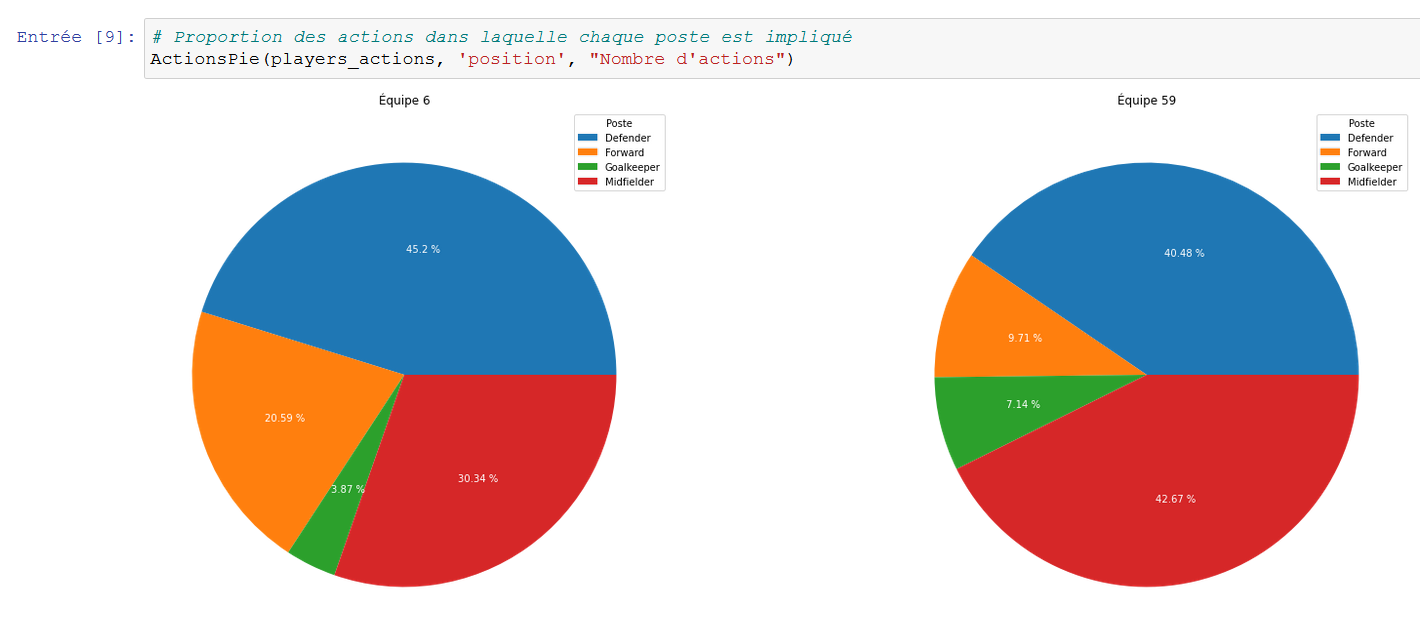


Figure 8: répartition des évènements par type de poste avec 2 équipes en exemple

*Compléments :*

D’autres vue Power BI ou graphiques Python sont disponibles respectivement dans le fichier « Rapport BI visualisation data.pbix » et le notebook « Bet-Py\_Notations.ipynb ». Ces 2 fichiers sont respectivement dans les dossiers « 1 – DataViz » et « 3 – Notebooks » du dossier sous GitHub.

# Nettoyage, mise en forme et choix des features

A partir des tables d’entrée, des dataframes « match\_results » ainsi que « match\_infos » ont été créés, et ce en deux temps comme le montre la vue ci-dessous :

Une image contenant table

Description générée automatiquement

Figure 9: Vue Draw IO des tables match\_results et match\_infos créées

## Première itération complète

L’objectif à ce stade du projet était **d’avoir « rapidement » une première version même incomplète de données afin de pouvoir tester un premier modèle.**

Pour **« être à jeu égal »** et « ne pas tricher » **avec** ce dont disposent **les bookmakers**, nous avons décidé de ne pas **utiliser** toute la base de données mais **uniquement des données des matchs précédents la rencontre où l’on veut prédire le gagnant.**

Lors de cette première itération, les features contenaient les informations « simples » de la table « match » ainsi qu’un grand nombre d’indicateurs contenant de nombreuses informations de statistiques sur les possession des équipes en 1ère et 2nd mi-temps, et ce selon le poste des joueurs ayant la possession (notés dans la table « match\_infos »).

Cette première itération a nécessité également du nettoyage avec des clubs ayant différentes syntaxes (Angers et Angers SCO par exemple), des caractères spéciaux donnant également des données d’entrées difficiles à lire ('Saint-\\u00c9tienne' pour Saint-Etienne par exemple) et des lignes sans un club renseigné où nous avons été récupérer l’information dans les calendriers passés de matchs.

## Seconde itération

Après parcours d’un état de l’art sur le sujet sur ce type de problème, dont l’article suivant[[1]](#footnote-1), nous avons défini que ce qui pourrait améliorer la performance de notre modèle serait l’obtention d’indicateurs de tendance sur le dernier, les 3 derniers ainsi que les 5 derniers matchs.

Avec notre connaissance également du football, nous avons défini comme élément importants :

* La série de victoire d’affilée sur les 5 derniers matchs
* Le nombre de buts marqués
* Le nombre de buts encaissés
* Le nombre de jours depuis le dernier match (si c’est grand, cela suggère un bon nombre de matchs nuls réalisés dans les derniers matchs)

A ces indicateurs, nous avons rajouté les indicateurs suivants (non décrits dans la vue DrawIO ci-dessous) présentés en écart entre le club évalué et ses adversaires du ou des derniers matchs :

* L’activité des attaquants
* L’activité des défenseurs
* Le nombre de fautes
* Le nombre de passes
* Le nombre de tirs

Tous ces indicateurs ont été calculés pour le club à domicile et celui à l’extérieur et comme précisé plus haut, cela a été réalisé sur le dernier, les 3 derniers et les 5 derniers matchs.

**La table complète « match\_all »** est décrite ci-dessous avec les features retenues en vert :

Une image contenant table

Description générée automatiquement

Figure 10: Table match\_all avec les champs en "vert" représentant **une partie** des features retenues pour la 2nd itération

## Définition jeux train / test et standardisation des données

Avec toutes ces données, quelle que soit l’itération (première ou seconde), un découpage en jeu d’entraînement et de test ainsi qu’une étape de standardisation ont été réalisées.

Une image contenant texte

Description générée automatiquement

Figure 11: étapes de découpages train/test et standardisation des données

# Modèle : création, application et évaluation

A partir des ces données complètes, nous avons donc pu attaquer la partie de modélisation.

## Une première itération complète

Pour cette première itération, nous avons souhaité tester différents algorithmes :

* SVM avec une grille de recherche (simple avec une combinatoire de 27 possibilités)
* KNN
* Random Forest

Une image contenant texte

Description générée automatiquement



Une image contenant texte

Description générée automatiquement

Avec le modèle SVM, notre meilleure paramétrage donnait un score de 69%.

Une image contenant texte

Description générée automatiquement

Le modèle des K plus proches voisins donnait quant à lui un score de 63%.

Une image contenant texte

Description générée automatiquement

Enfin le modèle Random Forest donnait quant à lui un score proche de 69% également.

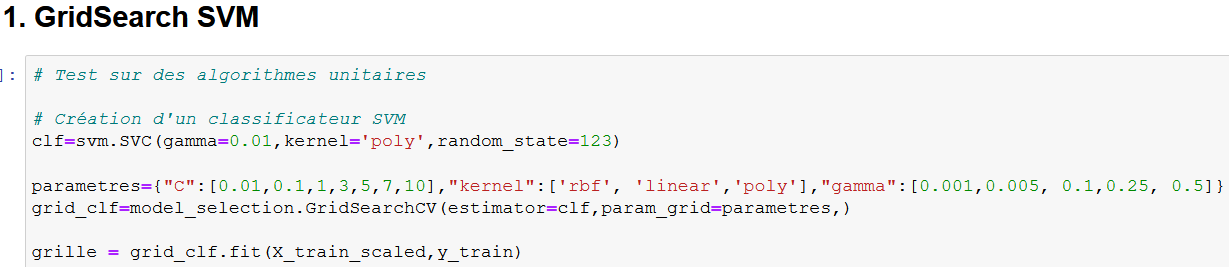
**Avec cette première itération, nous approchions des 70% d’accuracy, ce qui était très encourageant pour ce modèle.** **En effet, avec uniquement des indicateurs de possession, la performance semblait bonne.**

Après vérification des indicateurs d’entrée, nous nous sommes rendus compte que les indicateurs de possession avaient une petite erreur de calcul et surtout nous voulions intégrer un plus grand nombre d’indicateurs qui nous semblaient bien plus important pour aider à la prédiction dans une seconde itération. Une grande partie du travail de ce projet a donc été réalisée afin de les calculer (cf. [Nettoyage, mise en forme et choix des features / Seconde itération](#_Seconde_itération))

## Seconde itération avec nouvelles features

A partir d’un plus grand nombre de données et des modèles plus poussés, nous avons cette fois testé :

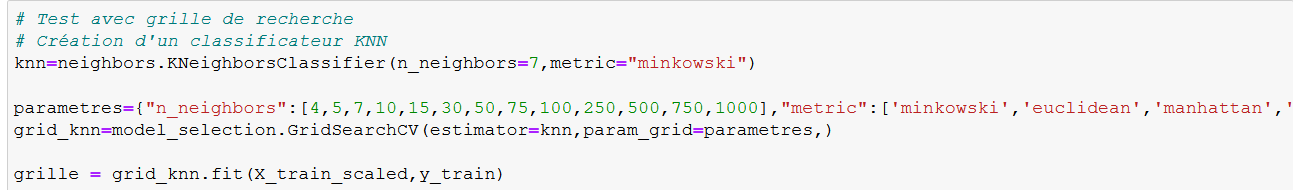
* SVM avec grille de recherche bien plus complexe (105 simulations)
* KNN avec également une grille de recherche (51 simulations)
* Random Forest
* Un réseau de neurones « simple » avec deux couches Dense



**Pour le modèle SVM avec la grille de recherche, le meilleur score était de 66,2%.**

Une image contenant texte

Description générée automatiquement



**Pour le modèle KNN avec la grille de recherche, le meilleur score était de 65,7%.**

Une image contenant texte

Description générée automatiquement

**Pour le modèle Random Forest, le score est de 65,7% également**, avec la matrice de confusion présentée ci-dessus.

 Une image contenant texte

Description générée automatiquement

Le réseau de neurones comporte deux couches Dense et l’entraînement a été réalisé de la manière suivante :

Une image contenant texte

Description générée automatiquement

**Malheureusement, les résultats n’étaient pas meilleurs, avec un score à 64%** comme le montre le tableau de metrics ainsi que la matrice de confusion ci-dessous.

Une image contenant table

Description générée automatiquement



Afin de creuser cette piste de réseau de neurones, nous avons tenter de créer des modèles plus complexes avec plus de couches et des couches de Dropout afin d’éviter au maximum l’overfitting.

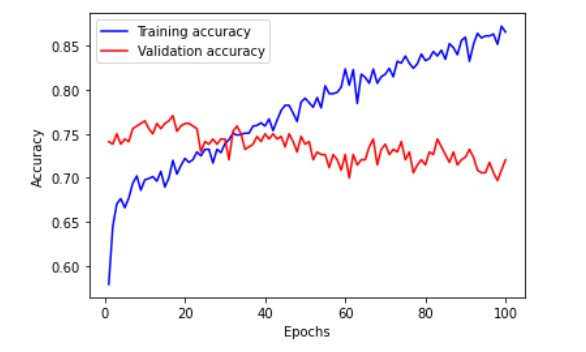
Une image contenant table

Description générée automatiquement

Nous avons également souhaité prendre un plus grand nombre d’epochs pour entraîner le modèle.

Une image contenant texte

Description générée automatiquement



Une image contenant table

Description générée automatiquement

Avec les différentes itérations faites sur cette seconde itération de neurones, les résultats ne sont que très peu améliorés **avec un score de 65%.**

**En conclusion de cette seconde itération avec les nouvelles features qui nous semblaient plus pertinentes ainsi que des tendances sur le derniers, les 3 derniers ou les 5 derniers matchs, nos résultats sont malheureusement moins bons que lors de la première itération.**

# Conclusion du projet

Ce projet complet nous a permis, comme prévu initialement, de parcourir l’ensemble des étapes d’un projet Data Science. Et comme souvent évoqué la partie « intéressante » de création des algorithmes et d’ajustement des paramètres et hyperparamètres n’est pas la plus longue.

Nous avons pu parcourir toutes ces étapes et obtenir des résultats de prédiction plutôt intéressants (avec un réseau de neurones appris dans les derniers modules), même si ça mériterait bien entendu de les améliorer en modifiant les features et/ou les algorithmes. Nous aurions souhaité aller plus loin sur ce sujet mais l’exploration, le nettoyage et la création des features a été longue et malheureusement un membre de l’équipe projet a dû se retirer rapidement du sujet.

Par rapport à nos 2 itérations, et par rapport au travail d’amélioration, il nous semblerait que la priorité pour améliorer les prédictions serait de passer du temps à ajuster, créer, sélectionner les features intéressantes. En effet, les différents algorithmes et paramétrages associés avaient finalement assez peu de différences. Le « challenge » fixé à la fin de la 1ère itération (même si ambitieux) d’obtenir 75% ne se ferait sans doute pas uniquement avec de meilleurs algorithmes sur ce jeu de données.

## Rappel des fichiers à consulter

**Pour rappel, voici sous notre dossier GitHub les différents dossiers et notebooks à consulter :**

* Visualisation :
  + **1 – DataViz / Rapport BI visualisation data** (fichier Power BI ou pdf associé si vous n’avez pas le logiciel)
  + **3 – Notebooks / Bet-Py\_Notations.ipynb**
* Création des données, nettoyage et réalisation des modèles :
  + **3 – Notebooks / Bet-Py\_Modélisation.ipynb** (pour arriver à la création de la table « match\_all » brute)
  + **3 – Notebooks / Bet-Py\_Machines\_Learning - Version finale.ipynb** (pour créer et sélectionner certaines features, faire la standardisation des données ainsi que les différents modèles et résultats associés)

1. : <https://towardsdatascience.com/machine-learning-algorithms-for-football-prediction-using-statistics-from-brazilian-championship-51b7d4ea0bc8> [↑](#footnote-ref-1)