2. EDA

Ce projet s'inscrit dans une démarche complète de science des données, de la préparation des données brutes jusqu'au déploiement et à la surveillance du modèle, selon les meilleures pratiques de l'industrie. Il offre un cadre réutilisable et évolutif pour l'analyse prédictive dans le domaine sportif et au-delà.

Sources de données

VirtualBetsAPI

Présentation de l'outil

VBA est une API developper sur-mesure dans le cadre du projet PariVision, nous permettant d'appeler des données fictives aléatoirement. Cet outil exploite FastAPI et se base sur un fichier d'origine CSV, transformé en JSON.

Utilisation de Virtual Bets API

- **Appel : Demande de retour de 2 nodes**

```
http://parivision.heuzef.com:8800/getdata/2
```

Java

- **Résultat**


```
[
    {
        "Game ID": "0029700427",
        "Game Event ID": "389",
        "Player ID": "100",
        "Player Name": "Tim Legler",
        "Team ID": "1610612764",
        "Team Name": "Washington Wizards",
        "Period": "4",
        "Minutes Remaining": "11",
        "Seconds Remaining": "22",
        "Action Type": "Jump Shot",
        "Shot Type": "2PT Field Goal",
```

```
        "Shot Zone Basic": "Mid-Range",
        "Shot Zone Area": "Right Side(R)",
        "Shot Zone Range": "8-16 ft.",
        "Shot Distance": "15",
        "X Location": "117",
        "Y Location": "109",
        "Shot Made Flag": "1",
        "Game Date": "19980102",
        "Home Team": "WAS",
        "Away Team": "IND",
        "Season Type": "Regular Season"
    },
    {
        "Game ID": "0029700427",
        "Game Event ID": "406",
        "Player ID": "100",
        "Player Name": "Tim Legler",
        "Team ID": "1610612764",
        "Team Name": "Washington Wizards",
        "Period": "4",
        "Minutes Remaining": "9",
        "Seconds Remaining": "36",
        "Action Type": "Jump Shot",
        "Shot Type": "2PT Field Goal",
        "Shot Zone Basic": "Mid-Range",
        "Shot Zone Area": "Right Side(R)",
        "Shot Zone Range": "8-16 ft.",
        "Shot Distance": "14",
        "X Location": "143",
        "Y Location": "25",
        "Shot Made Flag": "0",
        "Game Date": "19980102",
        "Home Team": "WAS",
        "Away Team": "IND",
        "Season Type": "Regular Season"
    }
]
```


Java

- **Limite**

Le limite est fixée à 50000 nodes contenus dans le fichier dat_NBA.
L'URL `http://parivision.heuzef.com:8800/getdata/2` renvoie les deux premiers nodes du fichier. Pour renvoyer les 3 premiers nodes, il faut remplacer 2 par 3 dans l'URL

Si l'url est appelé deux fois, les deux noeuds suivants sont renvoyés. Un compteur est implémenté afin de ne pas fournir deux fois les mêmes nodes.

Pour réinitialiser le compteur à 0, il faut appeler l'URL suivante :
`http://parivision.heuzef.com:8800/reset`

Ressources

https://trello.com/c/xQWQqayp/22-ajout-nouvelle-source-de-data-pierreapi

https://github.com/jja4/nba_mlops/

BetsAPI

Utilisation du service BetsAPI

BetsAPI est un service RESTful pour les données sur tous les sports. C'est un service payant. L'avantage le plus important est qu'il fournit des vrais données de qualité, en temps réels.

Documentation : https://betsapi.com/docs/

Pour enregistrer le TOKEN dans une variable environment :

```
export TOKEN="SECRET-TOKEN-123"
```

Java

Afficher le TOKEN

```
env | grep TOKEN
```

Java

Tester l'API

Analyse exploratoire des données brutes

Les données brutes comportent 22 variables, couvrant à la fois des informations contextuelles (identifiants de match, joueur, équipe, période, date, etc, ...), des caractéristiques du tir (type de tir, zone du terrain, distance, coordonnées, etc.) et la cible à prédire (Shot Made Flag, indiquant si le tir a été réussi ou non).

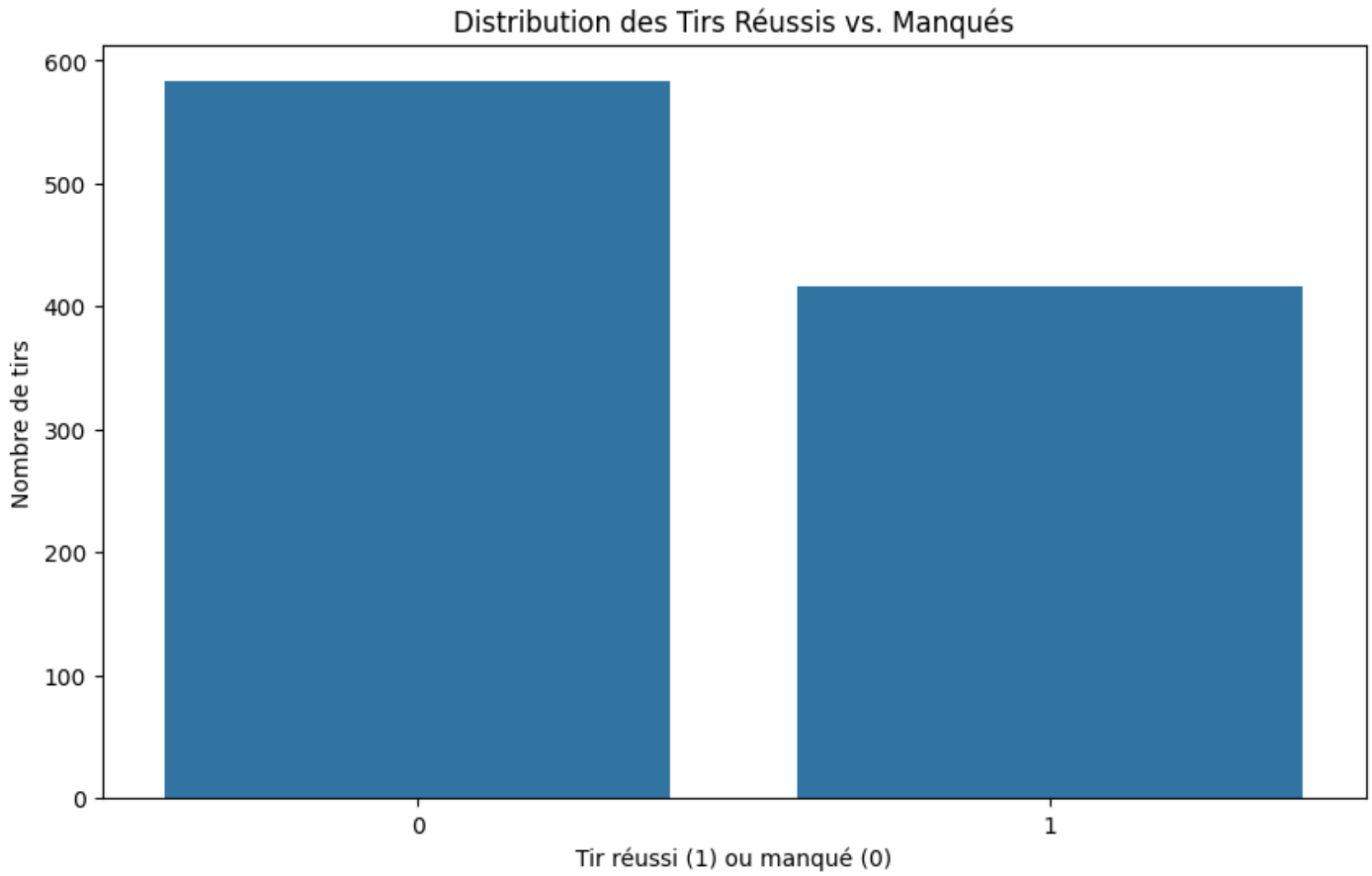
```
Data columns (total 22 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Game ID             1000 non-null   int64
 1   Game Event ID       1000 non-null   int64
 2   Player ID           1000 non-null   int64
 3   Player Name         1000 non-null   object
 4   Team ID             1000 non-null   int64
 5   Team Name           1000 non-null   object
 6   Period              1000 non-null   int64
 7   Minutes Remaining   1000 non-null   int64
 8   Seconds Remaining   1000 non-null   int64
 9   Action Type         1000 non-null   object
 10  Shot Type           1000 non-null   object
 11  Shot Zone Basic     1000 non-null   object
 12  Shot Zone Area      1000 non-null   object
 13  Shot Zone Range     1000 non-null   object
 14  Shot Distance       1000 non-null   int64
 15  X Location          1000 non-null   int64
 16  Y Location          1000 non-null   int64
 17  Shot Made Flag      1000 non-null   int64
 18  Game Date           1000 non-null   int64
 19  Home Team           1000 non-null   object
 20  Away Team           1000 non-null   object
 21  Season Type         1000 non-null   object
dtypes: int64(12), object(10)
memory usage: 172.0+ KB
```

	Game ID	Game Event ID	Player ID	Team ID	Period
count	1.000000e+03	1000.000000	1.000000e+03	1.000000e+03	1000.000000
mean	2.282636e+07	258.231000	1.664872e+05	1.610613e+09	2.533000
std	4.957243e+06	159.915436	3.803926e+05	8.676662e+00	1.153361
min	2.000001e+07	2.000000	1.500000e+01	1.610613e+09	1.000000
25%	2.050117e+07	116.750000	1.516500e+03	1.610613e+09	1.000000
50%	2.110061e+07	267.000000	2.548000e+03	1.610613e+09	3.000000
75%	2.170074e+07	382.250000	2.015872e+05	1.610613e+09	4.000000
max	4.990008e+07	743.000000	1.629673e+06	1.610613e+09	6.000000

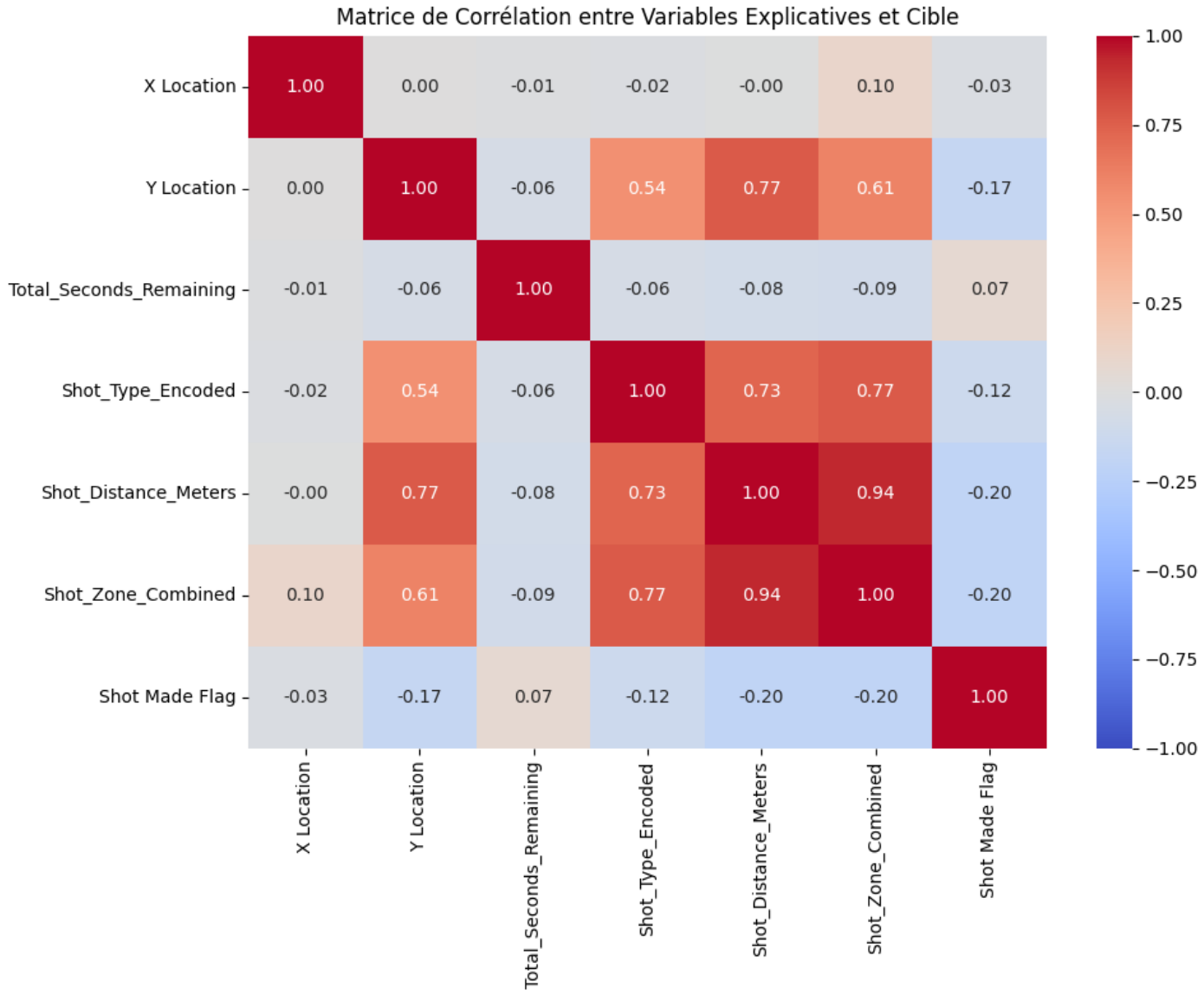
	Minutes Remaining	Seconds Remaining	Shot Distance	X Location
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	5.218000	28.378000	13.093000	2.278000
std	3.414108	17.452326	9.733195	113.872216
min	0.000000	0.000000	0.000000	-246.000000
25%	2.000000	13.000000	2.000000	-64.250000
50%	5.000000	28.000000	15.000000	0.000000
75%	8.000000	43.000000	23.000000	73.000000
max	11.000000	59.000000	58.000000	247.000000

	Y Location	Shot Made Flag	Game Date
count	1000.000000	1000.00000	1.000000e+03
mean	84.534000	0.41700	2.008836e+07
std	88.392477	0.49331	6.400691e+04
min	-31.000000	0.00000	1.997111e+07
25%	6.000000	0.00000	2.003122e+07
50%	53.000000	0.00000	2.009021e+07
75%	163.000000	1.00000	2.014113e+07
max	544.000000	1.00000	2.020031e+07

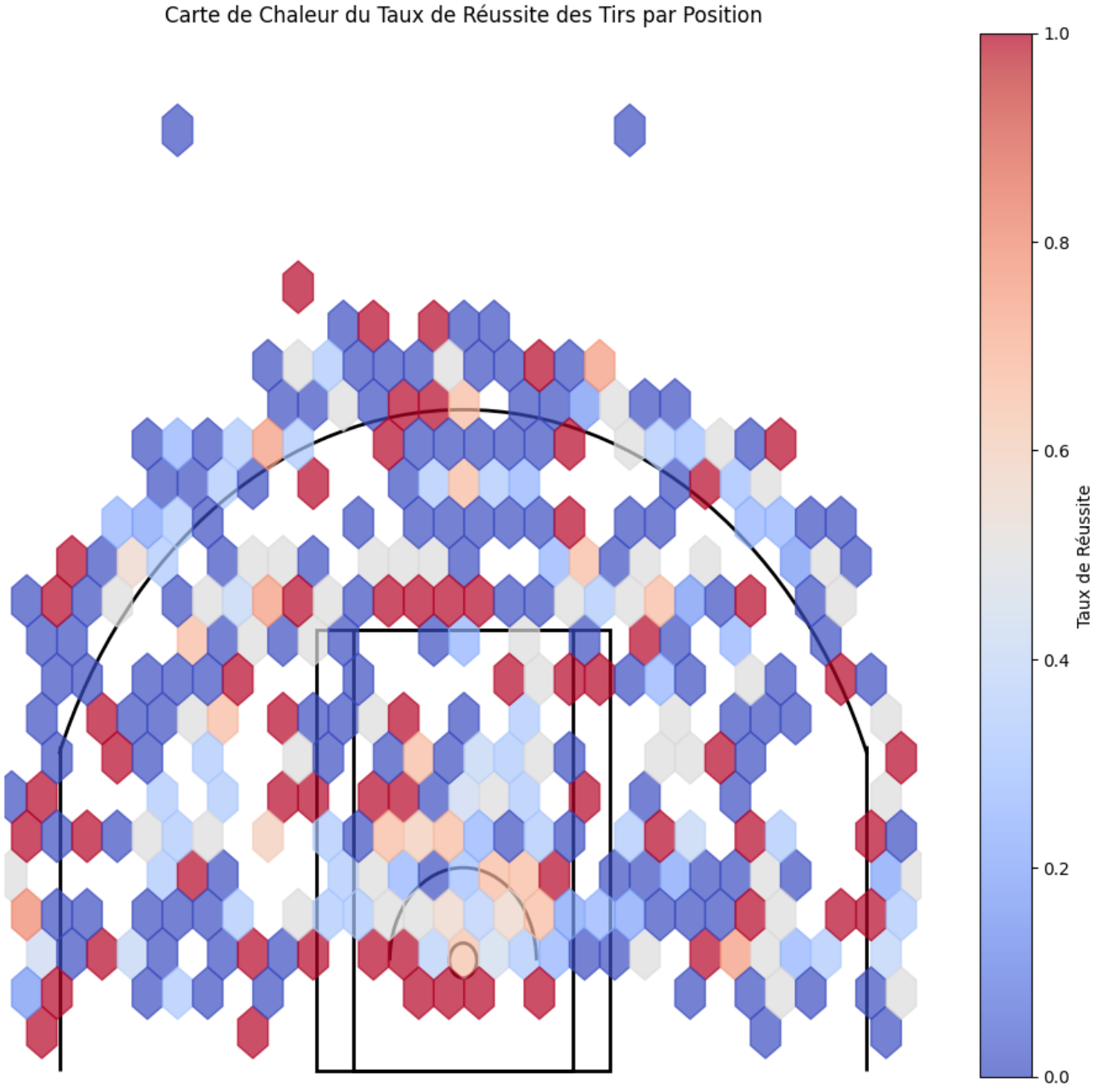
Statistiques descriptives (échantillons de 1000 entrées)



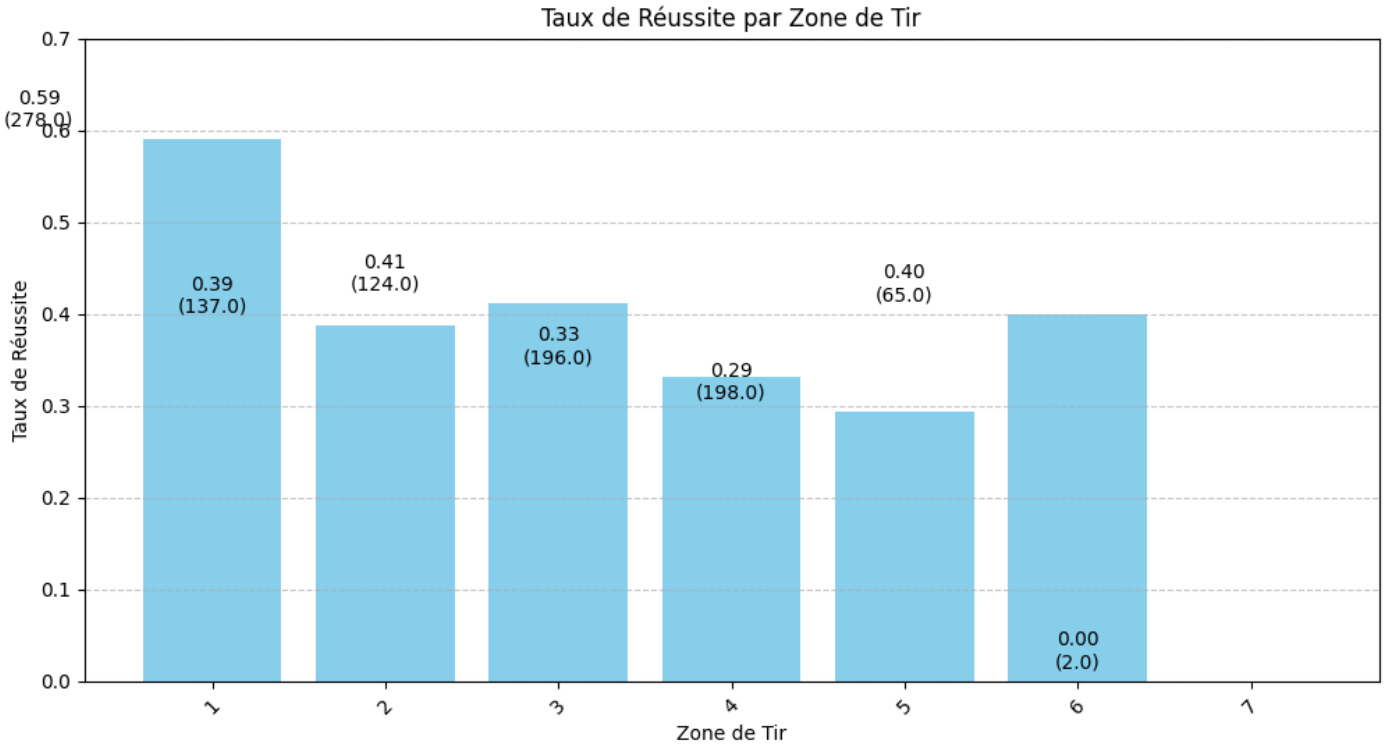
Visualisation de la distribution de la cible (Shot Made Flag)



Matrice de Corrélation



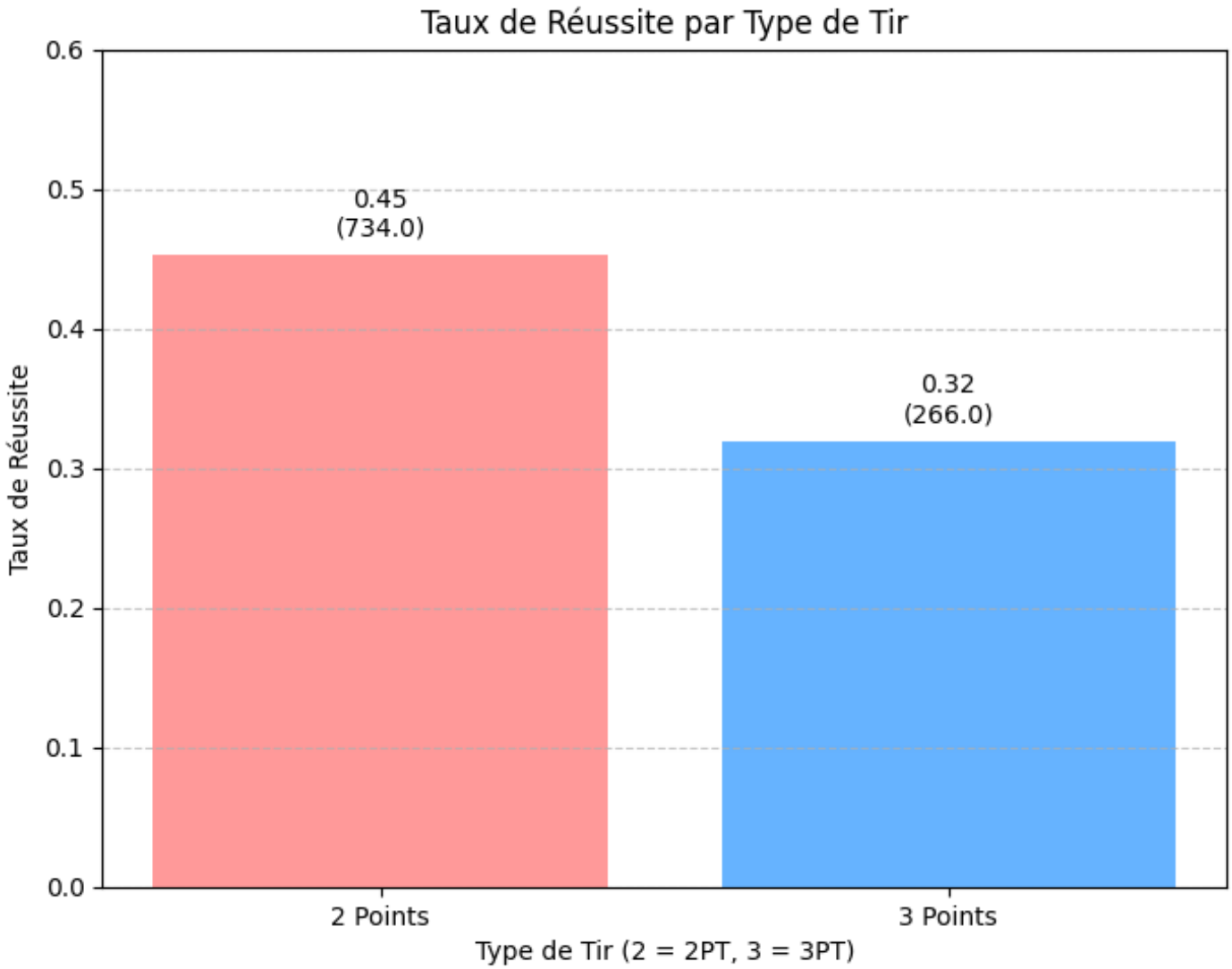
Analyse Spatiale des Tirs sur le Terrain



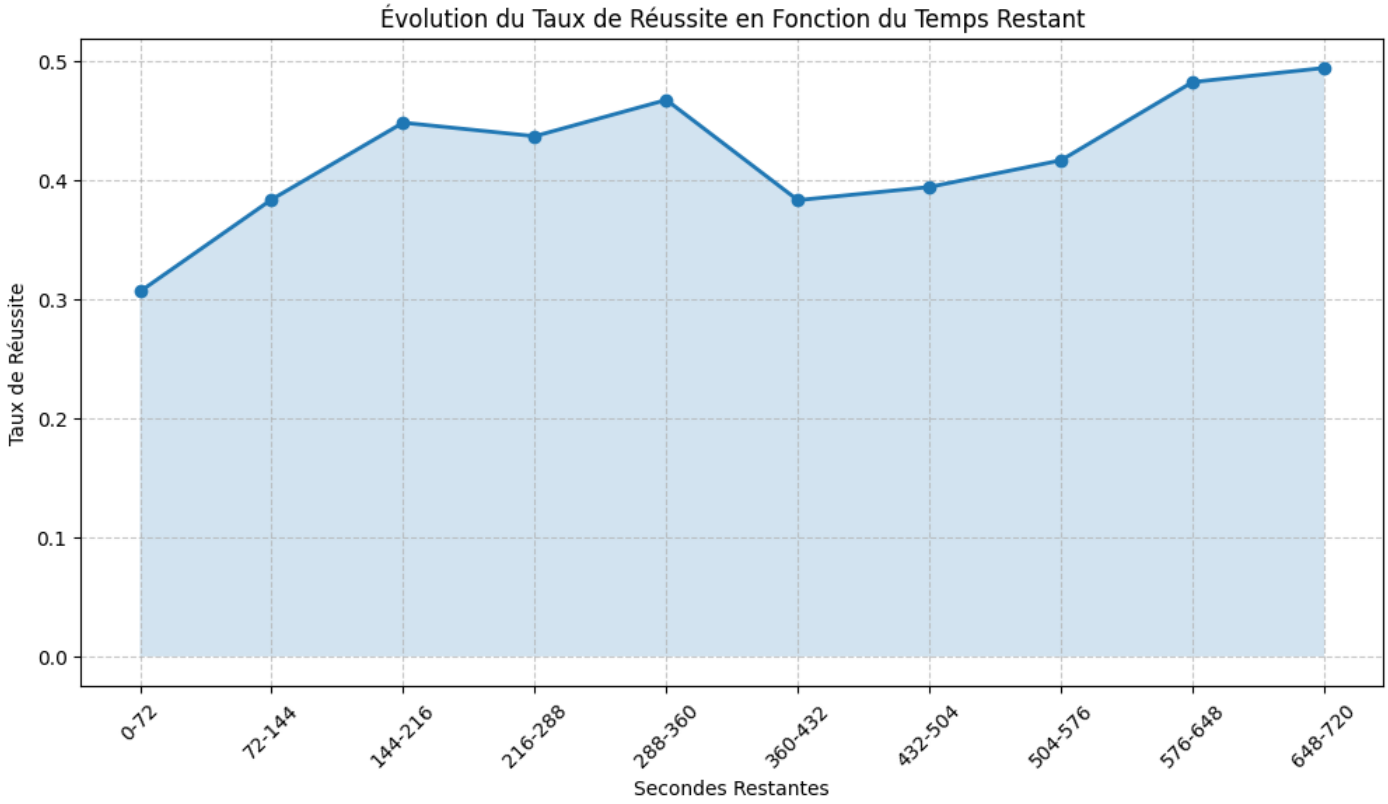
Analyse des zones

Analyse des zones :

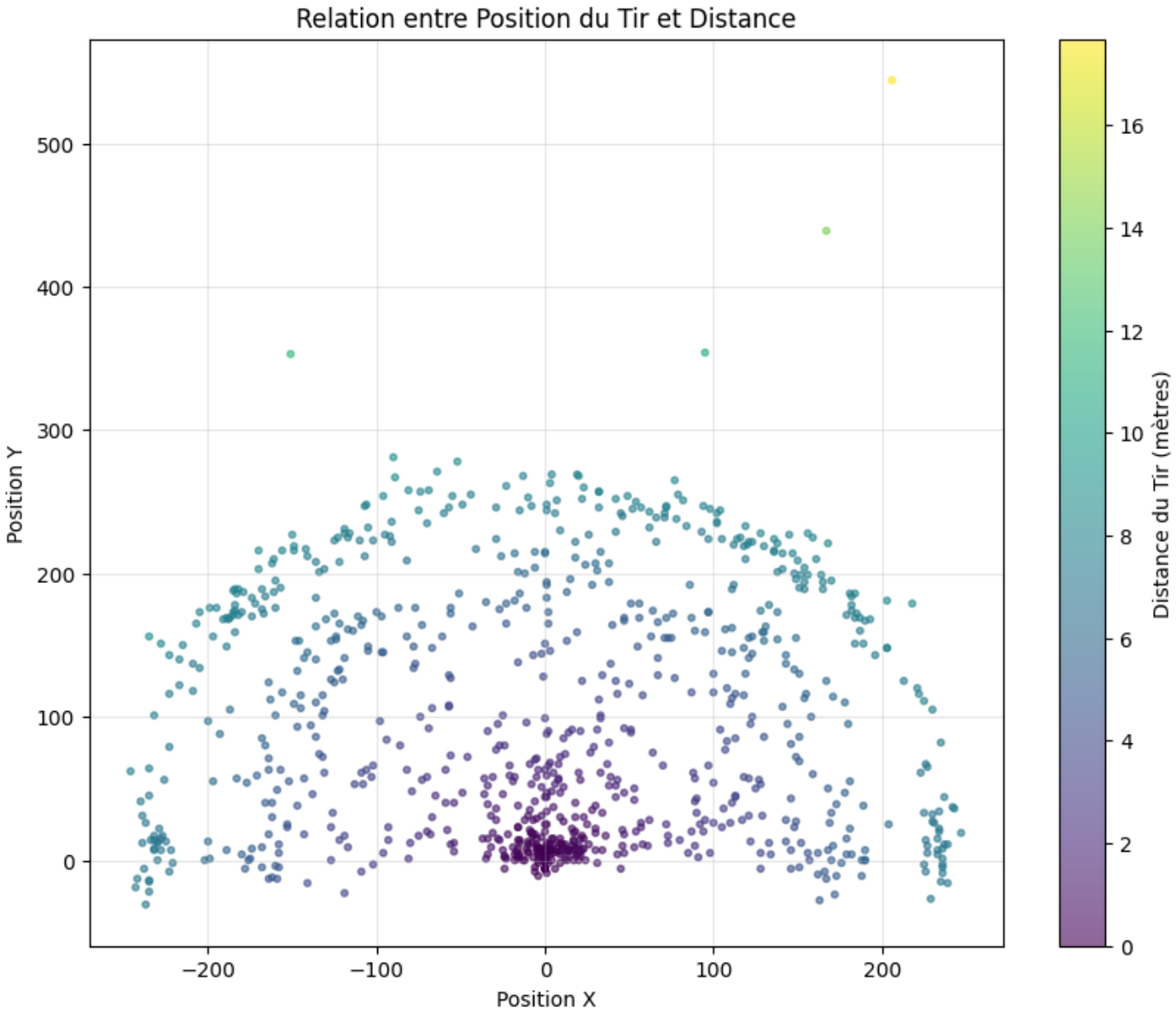
- **Zone 1 (Restricted Area):** réussite moyenne ~59%
- **Zone 2 (In The Paint):** réussite moyenne ~38%
- **Zones 3 (Mid-Range):** réussite moyenne ~34%
- **Zones 4 (Mid-Range):** réussite moyenne ~42%
- **Zone 5 (Above the Break 3):** plus faible réussite moyenne ~29%
- **Zone 6 (Corner 3):** meilleure réussite moyenne à 3 points ~40%



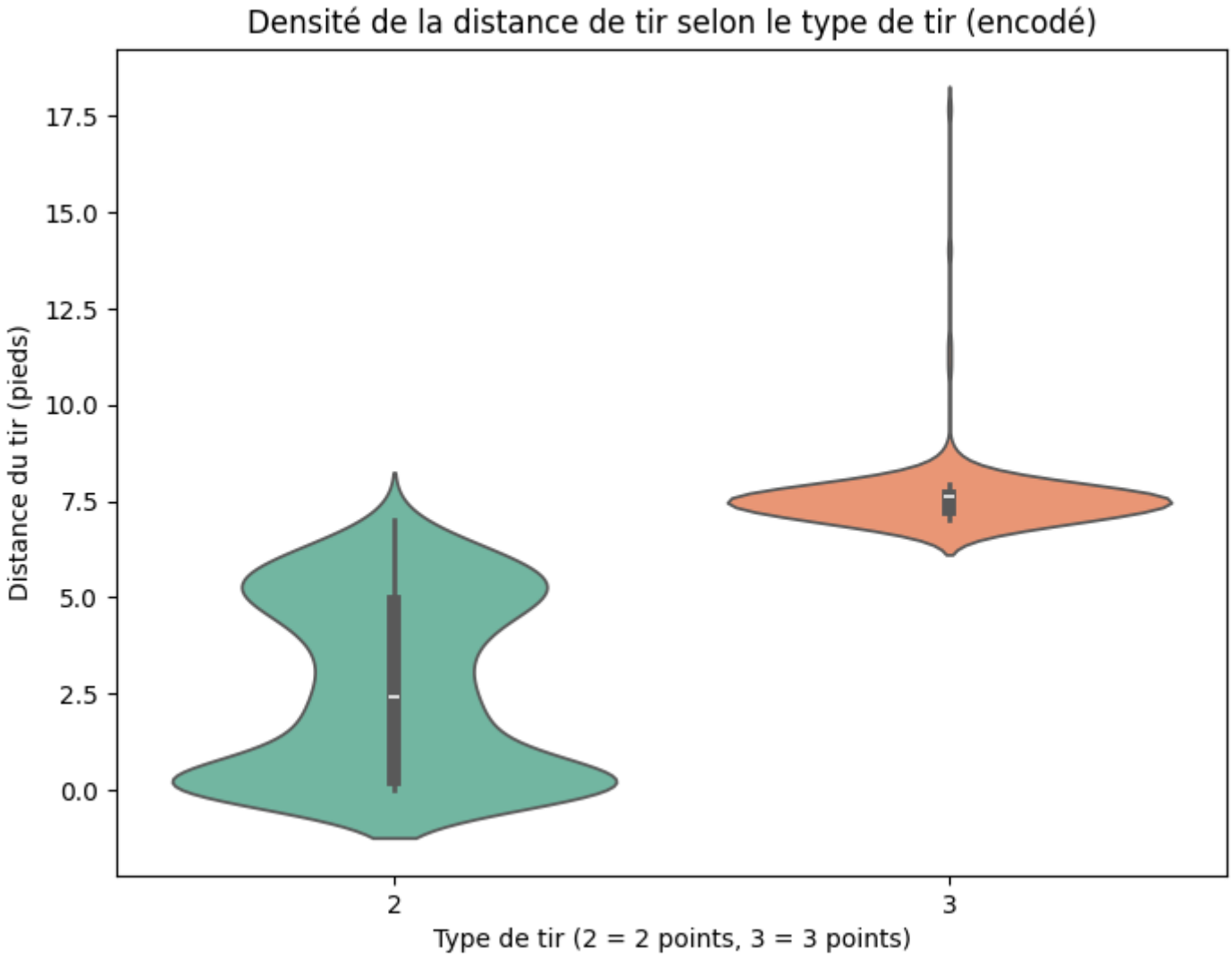
Relation entre type de tir et taux de réussite



Influence du Temps Restant



Relation entre distance et position du tir



Densité de la distance de tir selon le type de tir

Pré-traitement des données

Le pipeline de traitement des données mis en place comprend plusieurs étapes structurantes :

- **Pré-traitement et nettoyage** : suppression des colonnes non pertinentes, gestion des doublons, vérification et traitement des valeurs manquantes.
- **Ingénierie des fonctionnalités** : création de nouvelles variables plus informatives, telles que le temps restant dans la période, l'encodage numérique du type de tir, conversion des unités de mesure, etc …
- **Préparation à la modélisation** : sélection et transformation des variables explicatives, encodage des variables catégorielles, séparation des jeux d'entraînement et de test.

Variables finales

Variable	Description	Type

Player ID	Identifiant du joueur	Variable nettoyée
X Location	Position X sur le terrain	Variable nettoyée
Y Location	Position Y sur le terrain	Variable nettoyée
Total_Seconds_Remaining	Temps total restant	Nouvelle variables
Shot_Type_Encoded	Type de tir (2 points ou 3 points)	Nouvelle variables
Shot_Distance_Meters	Distance en mètres	Nouvelle variables
Shot_Zone_Combined	Zone de tir (7 zones)	Nouvelle variables
Shot Made Flag	Indicateur de réussite du tir	Variable cible

Tableau comparatif avant/après pré-traitement (Bronze → Silver)

Variable	**Bronze**	**Silver**
Nombre de colonnes	22	12
Colonnes présentes	Game ID, Game Event ID, Player ID, Player Name, Team ID, Team Name, Period, Minutes Remaining, Seconds Remaining, Action Type, Shot Type, Shot Zone Basic, Shot Zone Area, Shot Zone Range, Shot Distance, X Location, Y Location, Shot Made Flag, Game Date, Home Team, Away Team, Season Type	Player ID, Action Type, Shot Type, Shot Zone Basic, Shot Zone Area, Shot Zone Range, Shot Distance, X Location, Y Location, Shot Made Flag, Total_Seconds_Remaining, Shot_Type_Encoded
Colonnes supprimées	-	Game ID, Game Event ID, Player Name, Team ID, Team Name, Period, Game Date, Home Team, Away Team, Season Type
Colonnes créées	-	Total_Seconds_Remaining (Minutes Remaining × 60 + Seconds Remaining), Shot_Type_Encoded (2 ou 3 selon Shot Type)
Doublons	Non vérifié initialement	Vérifiés et supprimés (aucun doublon trouvé)
Valeurs manquantes	Non connues initialement	Vérifiées et remplacées par la médiane pour les numériques (aucune valeur manquante détectée)
Encodage	Variables catégorielles non encodées	Shot_Type_Encoded (2 = 2 points, 3 = 3 points, 0 = autre)
Colonnes temporelles	Minutes Remaining, Seconds Remaining	Fusionnées en Total_Seconds_Remaining
Colonnes de type de tir	Shot Type (texte)	Shot_Type_Encoded (numérique)
Cible	Shot Made Flag (0/1)	Shot Made Flag (0/1)
Format des variables	12 numériques, 10 catégorielles	Majoritairement numériques ou prêtes à l'encodage
Prêt pour l'apprentissage	Non	Oui

Ce tableau montre comment le pré-traitement améliore la structure et la qualité des données, facilitant ainsi la modélisation et la reproductibilité du projet.

BRONZE : Les données sont brutes, avec des colonnes redondantes, des variables catégorielles non encodées et des informations temporelles séparées.

SILVER : Le dataset est épuré, enrichi (feature engineering) et prêt pour l'apprentissage automatique, sans doublons ni valeurs manquantes.

Nettoyage des données

- Suppression des variables non pertinentes 'Game ID', 'Game Event ID', 'Player Name', 'Team ID', 'Team Name', 'Period', 'Season Type', 'Game Date', 'Home Team', 'Away Team'.
- Vérification des doublons : Aucun doublons
- Vérification des valeurs manquantes : Aucune valeurs manquantes

Ingénierie des fonctionnalités (Feature Engineering)

Pour améliorer la qualité des données, plusieurs transformations ont été effectuées lors de la phase de Feature Engineering.

Fusion des variables temporelles en une caractéristique continue

Les variables Minutes Remaining et Seconds Remaining ont été fusionnées en **Total_Seconds_Remaining** = (Minutes Remaining * 60) + Seconds Remaining.

Cela permet de représenter plus simplement le temps restant pour chaque tir, facilitant l'analyse par les modèles de machine learning.

Les colonnes d'origine ont été supprimées pour éviter la redondance.

Encodage numérique du type de tir

La variable catégorielle Shot Type a été convertie en numérique : 2 pour les tirs à deux points, 3 pour les tirs à trois points, et 0 pour les cas rares.

Cette transformation facilite l'utilisation de cette information dans les modèles numériques via une fonction d'encodage dédiée.

Transformation des variables de zone de tir lors du feature engineering

Le jeu de données original décrit la localisation du tir avec des variables catégorielles distinctes :

- **Shot Zone Basic** : zone principale (ex : Mid-Range, Restricted Area)
- **Shot Zone Area** : zone latérale ou centrale (ex : Left Side, Center, Right Side)
- **Shot Zone Range** : distance (ex : Less Than 8 ft., 8-16 ft.)

Les variables de zone de tir, telles que 'Less Than 8 ft.', '8-16 ft.', '16-24 ft.', '24+ ft.' et 'Back Court Shot', sont informatives mais redondantes. Elles ont été combinées en une seule variable, nommée **Shot_Zone_Combined**, pour simplifier la modélisation.

Synthèse

En résumé, ces transformations ont permis d'obtenir un jeu de données plus compact, plus informatif et directement exploitable par les algorithmes de machine learning, tout en conservant l'essentiel de l'information sportive et contextuelle utile à la prédiction du succès des tirs NBA.

Stockage des données

Le stockage des données brutes et pré-traités s'effectue au format JSON, dans un base de donné NoSQL.

Introduction

MongoDB est une base de données NoSQL orientée documents, qui se distingue par sa flexibilité, sa scalabilité, et sa capacité à gérer des données semi-structurées ou non structurées. Contrairement aux bases de données relationnelles traditionnelles (comme MySQL ou PostgreSQL), qui utilisent des tables pour stocker des données, MongoDB utilise des documents au format **JSON-like** (BSON en interne). Cette structure permet de modéliser les données de manière plus naturelle et intuitive pour les développeurs.

Créée en 2009 par MongoDB Inc., cette base de données est devenue l'une des solutions les plus populaires pour les applications modernes, notamment les applications web, mobiles, et les projets nécessitant un traitement rapide de gros volumes de données.

Installation sur Docker

Step 1 : Sur la machine host, vérifier que l'image de **mongoDB** n'est pas présente :

```
docker image
```

Java

Step 2 : Récupérer la dernière image de **mongoDB** disponible :

```
docker pull mongo:latest
```

Java

Step 3 : Vérifier que l'image est bien téléchargée :

```
docker image
```

Java

Step 4 : Créer un répertoire “mongoDB_PariVision” qui nous servira de volume pour la persistance des données :

```
mkdir mongoDB_PariVision
cd mongoDB_PariVision
```

Java

Step 5 : Lancer Docker avec l’image **mongoDB** en ligne de commande :

```
docker run -d -p 27017:27017 -v ~/mongoDB_PariVision/:/data/db --name
parivision_mongodb_container mongo:latest
```

Java

-d : dash mode, tourne en background

-p : association des ports de la machine host et du port sur lequel tourne **mongoDB** (27017)

-v : ajout du volume pour la persistance des données

–-name : nom du container **mongoDB**

Step 6 : Vérifier que le container est bien en RUN :

```
docker ps
```

Java

Step 7 : Accéder au container :

```
docker exec -it mongodb bash
```

Java

Step 8 : Une fois à l’intérieur du container, on démarre **mongoDB :**

```
mongosh
```

Java

Step 9 : Créer l’utilisateur

```
db.createUser({user: "parivision", pwd: "******************", roles: [ {
role: "readWrite", db: "parivision" }]})
```

Bash

Utilisation de mongoDB

a) Afficher les différentes bases de données

```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

Java

b) Création de la base de données pour PariVision

```
> use parivision
switched to db parivision
>
```

Java

c) Insertion d'un objet

```
> db.user.insert({"name":"Curry"})
WriteResult({ nInserted: 1 })
```

Java

d) Lister les objets

```
> db.user.find()
{ "_id" : ObjectId("223d432G56h765jo987d234a3"), "name":"Curry"}
```

Java

Utilisation de PyMongo

Documentation de PyMongo : https://pymongo.readthedocs.io/en/stable/tutorial.html

```
!pip3 install pymongo
from pymongo import MongoClient

client = MongoClient(
    host = parivision.heuzef.com,
    port = 27017,
    username = **********,
    password = **********
```

```
)

print(client.list_database_names())
```

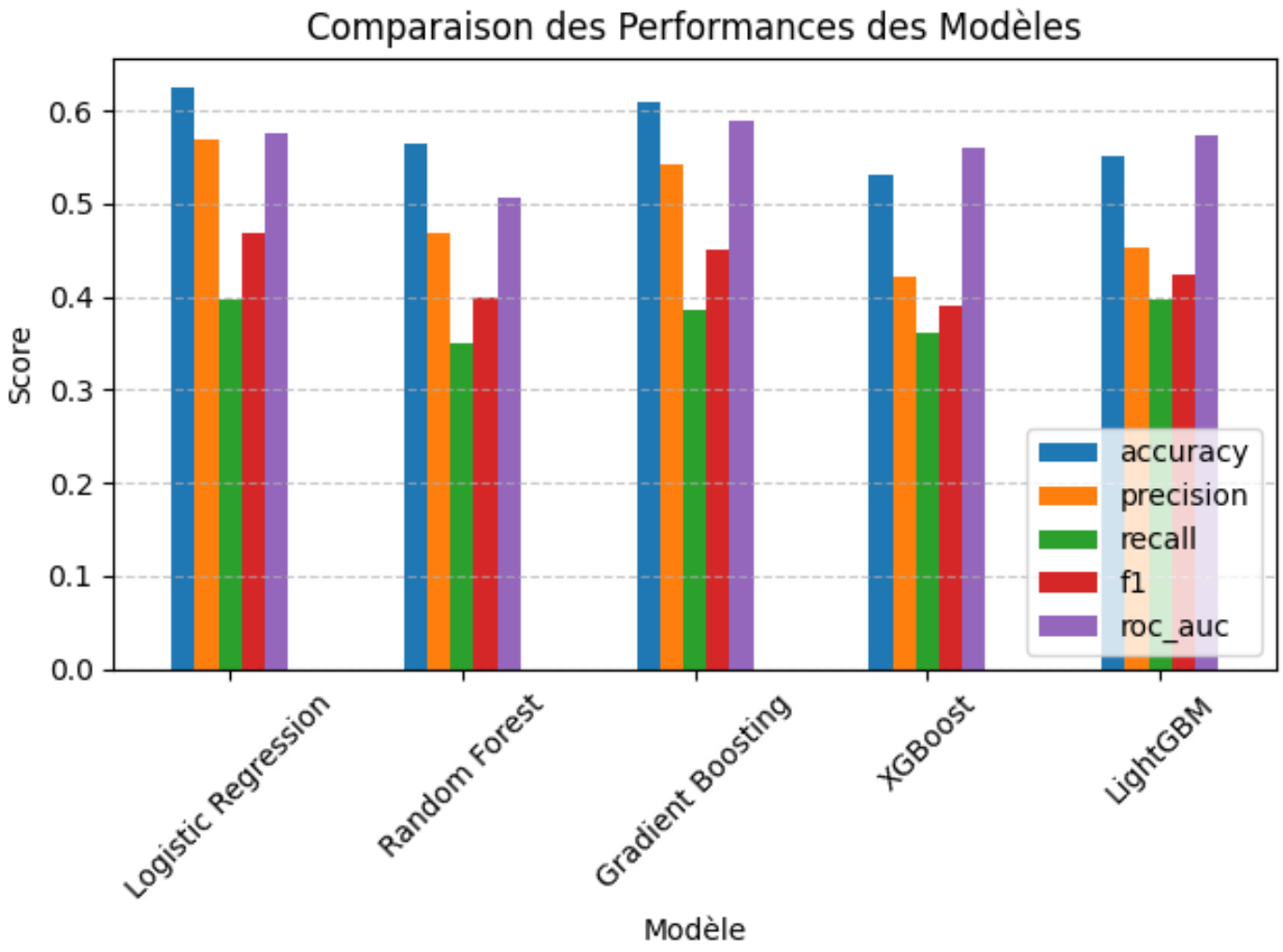
Python

Utiles mongoDB

Sheet cheat **mongoDB :**

Modélisation

La phase de modélisation s’appuie sur une sélection de modèles de classification robustes et complémentaires : régression logistique, Random Forest, Gradient Boosting, XGBoost et LightGBM. Chaque modèle est entraîné et évalué, avec un suivi rigoureux des paramètres et des métriques de performance grâce à MLflow.


Comparaison des Performances des Modèles
Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
accuracy
precision
recall
f1
roc_auc
Logistic Regression
Random Forest
Gradient Boosting
XGBoost
LightGBM
Modèle