**Cahier des charges projet MLOps**

# 1. Contexte et Objectifs

## Problématique de l'Application

L'application a pour objectif de permettre à 2 profils d’utilisateurs (Standard & Premium) de maximiser les chances de gain sur des paris sportifs réalisés sur des matchs de football de “Premier League”.

Pour atteindre cet objectif, l’application devra tout d’abord permettre à l’utilisateur de saisir ou sélectionner les éléments suivants :

* Saisir son “bankroll”, c’est à dire le montant total d’argent qu’il est prêt à risquer dans ses activités de paris. [Utilisateur Premium]
* Sélectionner le nom de l’équipe à domicile et celui de l’équipe extérieure pour obtenir des prédictions. [Tout Utilisateurs]
* Obtenir un calendrier des matchs à venir, voir les prédictions de résultats et obtenir un conseil de paris. [Utilisateur Premium]

Dans un second temps, l’application devra exploiter un modèle de Machine Learning afin de calculer une probabilité ou une cote sur le résultat de ce match.

Dans un troisième et dernier temps, à partir d’une analyse statistique, l’application définira :

* La valeur d’un pari éventuel sur le match sélectionné. Un pari ayant de la valeur lorsque la probabilité estimée d’un évènement calculé par le modèle, est supérieure à la probabilité implicite des cotes des bookmakers (probabilité implicite = 1 / cote). [Tous utilisateurs]
* Le montant optimal de pari, c’est à dire la fraction du bankroll à risquer sur ce pari. Ce montant pourra par exemple, être calculer d’après le critère de Kelly. [Utilisateur Premium]

## Parties Prenantes

* Commanditaire de l'Application : L’entreprise “bankruptcy entertainment agency”, portée par un millionnaire jaloux des profits qu’engendrent les sites de paris sportifs. En maximisant les gains des utilisateurs de l’application son objectif est de diminuer les profits des sites de paris sportifs.
* Utilisateur de l'Application : Cette application cible la communauté de parieurs sportifs.
* Administrateur de l'Application : L'équipe responsable du déploiement, de la maintenance et de la gestion des données.

## Contexte d'Utilisation

L'application sera accessible en ligne, intégrée à une application Streamlit , avec une interface utilisateur conviviale. Elle s'appuiera sur des données statistiques et résultats de matchs provenant de la source de données fournie.

# 2. Modèle

## Type de Modèle

Un modèle de Machine Learning supervisé sera utilisé, basé sur des algorithmes de classification. L’attendu étant de prédire la probabilité de victoire d'une équipe dans un match.

## Explication du Fonctionnement

Le modèle analysera les statistiques des équipes (et des joueurs si possible), les résultats passés, et d'autres facteurs pertinents pour estimer les probabilités de résultats : Victoire - Défaite - Nulle.

## Métriques d'Évaluation

* Accuracy : Mesure de la précision globale du modèle.
* Robustesse : Capacité à maintenir des performances élevées sur de nouvelles données.
* Temps d'Entraînement et de Prédiction : Pour garantir la viabilité en production.

# 3. Base de Données

## Données relatives à la création de comptes utilisateurs

Un fichier JSON sert au stockage des informations d'identification des utilisateurs. Ce type de données pourra être ensuite stockée dans une base de données (MySQL, MariaDB) afin d'améliorer le service et la sécurité des données.

## Données relatives à l’entraînement et la prédiction du Modèle

L'interface Streamlit propose une mise à jour journalière des résultats de « English Premier League » et « France Ligue 1 ».

La base de ressources de notre modèle sera actualisée de manière jouranlière également, en collectant les résultats du championnat mentionné durant la semaine écoulée.

Ces données sont stockées dans un système de fichiers, source d'un pipeline d'ETL servant à l’entraînement du modèle et aux prédictions.

## Intégration entre les données et le Modèle

La gestion des données et l’entraînement du modèle sont gérés par le framework Airflow. La programmation des tâches inclura la récupération des données, la transformation et l’entraînement du modèle

En sus, une intégration MLFlow permet de monitorer l'évolution des entraînements du modèle et de ses métriques associées, de décider de quel modèle mettre en production.

# 4. API

**Fast API**

## Authentification

L'API doit prendre en charge l'authentification des utilisateurs et des administrateurs pour garantir un accès sécurisé.

## Endpoints

## Catégorie “home”

1. /status - Pour requêter le fonctionnement de l'API
2. /available\_championships – Pour obtenir les championnats disponibles

## Catégorie “Users”

1. /user\_info – Retourne les informations de l’utilisateur enregistré
2. /change/password - Pour changer son password
3. /change/risk - Pour changer son profil de risques
4. /change/bankroll - Pour changer son bankroll
5. /championship/{championship:str}/calendar" – Retourne un calendrier des matchs à venir, les prédictions et des conseils de paris. (utilisateurs premium)
6. /predict- Obtenir les dernières predictions pour un match donné (matchs à venir)

## Catégorie “Admin”

1. /admin/delete/user/{user\_to\_delete:str} – Permet de supprimer un utilisateur
2. /admin/change/{user\_to\_change:str} – Permet de modifier des informations utilisateur

## Catéhorie “Auth”

1. /signup - Pour créer un compte utilisateur
2. /login - Pour se connecter

Note :

- la mise à jour des données et l’entraînement du modèle seront gérés par un outil open-source de gestion et d’orchestration de workflows.

- l’api n’a pas accès au modèle, les prédictions sont enregistrées dans un fichier lu par l’API.

# 5. Testing & Monitoring

## Tests Unitaires

* Vérifier les fonctionnement des différentes fonctions relatives au pipeline ETL
* Tests des différents endpoints de l'API.
* Tester les containers clés (airflow, api, streamlit)

## Monitoring

* Évaluation périodique de la performance du modèle sur l'ensemble des tests.
* Détection des performances dégradées.
* Alertes en cas de besoin de ré-entraînement.
* Tag et mise en production des meilleurs modèles.

# 6. Schémas d'Implémentations

## Flux d’informations

# 

## Environnement

[Insérer diagramme fonctionnement blocs)