TP #1 – JOUONS AVEC LES DONNEES

8PRO408 – Outils de programmation pour la science des données

Dans le cadre du cours, nous avons vu l'utilisation de NumPy, de Pandas, de Jupyter. Afin d'appliquer les concepts vus en cours et d'avoir un point de vue pratique dans le cours, il vous est demandé d'analyser un jeu de données afin d'en sortir diverses informations. Les jeux de données seront au format CSV, mais le résultat devra être fourni au format JSON.

Ainsi, le jeu de données à utiliser est un jeu de données relatif au jeu vidéo « League of Legends ». Vous pouvez le trouver sur le lien suivant : https://www.kaggle.com/datasets/park123/lol-data. Il contient deux fichiers contenant de l'information sur des matchs (complets ou au bout de 15mn), mais aussi des informations relatives aux champions (pour plus de détails, voir Capsule vidéo disponible durant tout le TP). Les fichiers nécessaires à la réalisation du TP sont tous disponible sur Moodle.

Ce TP doit être réalisé seul.

Travail à effectuer

1. Ouvrir le fichier de « match_15m.csv » et stocker dans un nouveau DataFrame les informations suivantes pour chaque match :
    - blue_advantage_gold : est-ce que l'équipe bleue est en avantage financier (plus d'argent que l'équipe rouge)
    - blue_advantage_buildings : est-ce que l'équipe bleue est en avantage de position (plus de bâtiment de l'équipe adverse détruit)
    - blue_drake : nombre de dragons tués par l'équipe bleue
    - red_drake : nombre de dragons tués par l'équipe rouge
    - blue_herald : est-ce que l'équipe bleue a tué le Rift Herald
    - red_herald : est-ce que l'équipe rouge a tué le Rift Herald
    - blue_win : est-ce que l'équipe bleue a gagné
2. **Ce DataFrame devra être exporté au format JSON avec le nom « df_ex1.json »**

3. A partir de ce fichier, créer un nouveau DataFrame avec l'information suivante :
    - Créer une répartition des écarts d'or de l'équipe bleue (*advantage_blue_category*) comme suit :
        - **Avantage fort :** $[2500, 5000]$
        - **Avantage faible :** $[500, 2500]$
        - **Pas d'avantage :** $[-500, 500]$
        - **Désavantage faible :** $[-2500, -500]$
        - **Désavantage fort :** $[-5000, -2500]$
    - Pour chaque écart, avoir une colonne « chance_win » indiquant le pourcentage de parties gagnées
    - Pour chaque écart, avoir une colonne « blue_structure » indiquant la moyenne du nombre de structures détruites dans l'équipe bleue
    - Pour chaque écart, avoir une colonne « red_structure » indiquant la moyenne du nombre de structures détruites dans l'équipe rouge
    - Pour chaque écart, avoir une colonne « blue_monster » indiquant la moyenne du nombre de monstres tués (dragons et herald combinés) pour l'équipe bleue
    - Pour chaque écart, avoir une colonne « red_monster » indiquant la moyenne du nombre de monstres tués (dragons et herald combinés) pour l'équipe rouge
    - **Ce DataFrame devra être exporté au format JSON avec le nom « df_ex3.json ».**
4. Que pouvez-vous déduire de ces données ? Inscrivez votre réponse dans le Notebook final.
5. A l'aide du dossier « champ » et de la colonne « matchId » disponible dans tous les fichiers CSV mis à disposition, retrouvez les champions utilisés dans chaque équipe.
    - Votre nouveau DataFrame devra être basé sur le fichier « match_full_time.csv »
    - Il devra avoir deux colonnes ajoutées « blue_team » et « red_team » comportant respectivement les champions (leur ID) dans chaque équipe, séparés par des virgules et encadrés par des [].
        - Exemple : ["Aatrox","DrMundo","Katarina","Lux","Zoe"]
    - Le DataFrame final devra être le même qu'initial, mais les deux colonnes doivent avoir été ajoutées dans l'ordre entre les colonnes « tier » et « blue_kill ».
    - **Ce Dataframe devra être exporté au format JSON avec le nom « df_ex5.json ».**
6. ***BONUS : 20 points bonus seront accordés si un autre DataFrame contenant les informations suivantes est ajouté (« df_bonus.json »):***
    - Les Index du DataFrame doivent être les ID des champions.
    - Une colonne par « tier » indiquant le pourcentage de victoire (ex. : bronze_win=0.53, silver_win=0.57)
    - Aggréger des données supplémentaires avec le jeu de données « champions_bonus.json » donné sur Moodle.
    - Une colonne par élément détenu dans la colonne « infos » avec la valeur de chaque élément :

        - Exemple : « info_attack : 8.666667 ; info_defense=5.33333 »
    - Une colonne par élément détenu dans la colonne « stats » avec la valeur de chaque élément.
        - Exemple : « stat_hp : 673.66667 ; stat_mp=100.00000 »
    - Analyse des données, que peut-on en tirer comme information ?

Format de rendu

Le projet rendu DOIT respecter les consignes suivantes :

- Fonctionner avec la dernière version de Python 3 (3.12.3 à l'heure actuelle)
- Fonctionner avec les librairies NumPy et Pandas (elles seront les seules installées sur l'environnement de test)
- Votre rendu doit être un unique fichier Jupyter Notebook, correctement rédigé avec des informations sur vos tests, sur la manière de générer vos données, etc.
- Celui-ci doit être capable de gérer des architectures comme celle-ci :
    - Votre fichier notebook
    - Le dossier « data »
        - « match_15m.csv »
        - « match_full_time.csv »
        - « champions_bonus.json »
        - Le dossier « champ »
            - Aatrox.csv
            - …
- Votre code sera lu par le correcteur au format Notebook
- Le fichier Notebook doit être nommé de la manière suivante : « NOM_PRENOM.ipynb ».

Tout non-respect de ces consignes de rendu peut se voir sanctionné par des pénalités.

Date de rendu

Le projet devra être remis avec **un seul fichier .ipynb au plus tard le 4 juin 2023 à 23h59**. Le rendu se fera sur Moodle via l'option créée à cet effet.

ATTENTION : Aucun retard n'est toléré. Si vous n'avez pas rendu le fichier zip, vous aurez automatiquement la note 0.

Répartition de la notation

- 5% : Nomenclature respectée pour l'ensemble des fichiers (donnés et générés)
- 5% : Le code fonctionne directement sans intervention annexe
- 20% : Le fichier « df_ex1.json » existe et correspond bien aux attentes
- 35% : Le fichier « df_ex3.json » existe et correspond bien aux attentes
- 5% : L'analyse réalisée pour la réponse à la question 4. est mise en évidence et est décente.
- 30% : Le fichier « df_ex5.json » existe et correspond bien aux attentes.
- *Le bonus de 20% sera accordé sur un autre TP si l'étudiant a déjà la note maximale à celui-ci.*