TP – Installation environnement et Analyse de la donnée

Résumé

Mettre en place et appréhender un premier environnement d'expérimentation de Data Science sur Python
Prendre en main un Dataset de données simples. Générer des visualisations et des insights permettant de caractériser la donnée.

Consigne

- Pensez à vous entraider
- Utiliser toutes les ressources à votre disposition
- Prenez votre temps l'important est de comprendre ce que vous faites
- Il n'y a pas de mauvaises réponses
- Si votre travail a du sens peu importe la quantité

Ressource

- Documentation pandas : API reference — pandas 2.2.1 documentation (pydata.org)
- Galerie seaborn: Example gallery — seaborn 0.13.2 documentation (pydata.org)
- Documentation plotly : https://plotly.com/python/
- Comment choisir vos visualisations : https://www.data-to-viz.com/#chord
- Tutorial vidéo Python et Machine Learning : https://www.youtube.com/playlist?list=PLO_fdPEVlfKqMDNmCFzQISI2H_nJcEDJq
- Aller plus loin sur la théorie : https://www.youtube.com/@statquest/videos

Environnement

- Télécharger la distribution Anaconda sur le site officiel
- Installé l'application sur la machine
- Lancer « Anaconda Navigator »
- Installer et lancer Jupyter Notebook
- Naviguer et créer un dossier pour cette expérimentation.
- Créer un nouveau Notebook
- Copier le fichier data.csv dans le même folder
- Importer les librairies de traitements que vous souhaitez utiliser

```
#Utilisation d'un alias pour faciliter les appels suivants
import pandas as pd

df_1 = pd.import_csv("...")

#Import de fonction directement depuis une librairie
from sklearn.metrics import F1_score,accuracy_score

F1 = F1_score("...")
```

Librairie suggérée :

- Numpy
- Pandas
- Scipy
- Matplotlib.pyplot
- Seaborn
- Plotly.express

Import des données et Analyse de fond

- **Principales librairies :**
  - Pandas
  - Numpy
- **Lire le fichier Data grâce à la librairie pandas**
- **Afficher les premières lignes de votre Dataframe pour avoir une première idée de votre data**
- **Explorer les informations de la donnée**
  - De combien de lignes se compose le dataframe ?
  - Combien de variable compose le dataframe ?
  - Quel est la répartition des variables par type ?
  - Peut-on noter la présence de valeurs null (Vide) ?
  - Ces valeurs null sont-elles problématiques ?
- **Quelques mots clé : read, info, head**

Analyse univariée

Explorer les variables de la donnée de manière unitaire.

- **Proposer au moins :**
  - Deux visualisations représentant des variables Numériques
  - Deux visualisations représentant des variables Catégorielles
  - Des statistiques descriptives d'au moins 4 variables numériques
  - Des comptes de 2 variables catégorielles
- **Vos graphiques doivent être :**
  - Cohérent par rapport au type de donnée
  - Lisible et apportant de la valeur
  - L'esthétique est un vrai plus
- **Librairies :**
  - Pandas
  - Numpy
  - Matplot.lib + Seaborn (Graphique statique)
  - Plotly pour des graphiques dynamiques

Analyse multivariée

Explorer les variables les unes par rapport aux autres

- **Proposer au moins :**
  - Deux visualisations présentant deux variables numériques
  - Deux visualisations présentant deux variables catégorielles
  - Deux visualisations présentant une variable catégorielle et une variable numérique
- **Vos graphiques doivent être :**
  - Cohérent par rapport au type de donnée
  - Lisible et apportant de la valeur
  - L'esthétique est un vrai plus
- **Bonus :**
  - Un graphique présentant 3 variables ensemble
  - Selon le métier deux variables sont susceptibles de changer le reste des données drastiquement :
    - Is_legendary
    - Generation
  - Vous pouvez explorer l'impact de ces variables en appliquant différents filtres ou en ajoutant un aspect « Category » à vos graphiques
  - Pensez « Local » :
    - Les statistiques d'un Pokémon de type Fighting peuvent-elles vraiment être comparées aux statistiques d'un Pokémon de type Psychic ?
    - Les statistiques des légendaires ne pourraient-ils pas représenter un biais ?
  - Certaines valeurs sont difficiles à comparer par l'échelle pensez à une méthode pour gommer cela.
- **Librairies**
  - Pandas
  - Numpy
  - Matplot.lib + Seaborn (Graphique statique)
  - Plotly pour des graphiques dynamiques

Analyse de corrélation

Explorer les corrélations entre les différentes variables :

- Explorer les corrélations entre les variables numériques à l'aide d'une matrice de corrélation (Proposé par pandas : ici)
- Rendez cette visualisation plus agréable et « Visuelle » grâce à une « Heatmap »
- Quelles variables sont les plus corrélées dans le dataframe ?
- Produisez deux visualisations permettant de visualiser l'existence de cette corrélation
- Enfin montrer des distributions comparées par rapport à une variable catégorielle (Par exemple la variable « is_legendary » ou encore en comparant des Pokémon « Fighter » et « Psychic » sur leur valeur d'attaque spéciale)
- Bonus :
    - Etudier la corrélation des variables catégorielles en appliquant un test d'indépendance du Khi² (Grace à la librairie scipy et la fonction : chi²)

Rendu

Un rapport avec vos différentes visualisations expliquées par une phrase.

Bonus

Explorer les différentes options qui s'offrent à vous si vous vouliez présenter un rapport de votre analyse à votre client. (Librairie, outil etc…)

Essayer de rendre votre code propre et lisible :

- Ajout de commentaire
- Markdown de titre
- Structuration et utilisation d'alias
- Nom de variable explicite

Annexe : définition de la donnée

- **name:** The English name of the Pokemon
- **japanese_name:** The Original Japanese name of the Pokemon
- **pokedex_number:** The entry number of the Pokemon in the National Pokedex
- **percentage_male:** The percentage of the species that are male. Blank if the Pokemon is genderless.
- **type1:** The Primary Type of the Pokemon
- **type2:** The Secondary Type of the Pokemon
- **classification:** The Classification of the Pokemon as described by the Sun and Moon Pokedex
- **height_m:** Height of the Pokemon in metres
- **weight_kg:** The Weight of the Pokemon in kilograms.
- **capture_rate:** Capture Rate of the Pokemon
- **base_egg_steps:** The number of steps required to hatch an egg of the Pokemon
- **abilities:** A stringified list of abilities that the Pokemon is capable of having
- **experience_growth:** The Experience Growth of the Pokemon
- **base_happiness:** Base Happiness of the Pokemon
- **against_?:** Eighteen features that denote the amount of damage taken against an attack of a particular type
- **hp:** The Base HP of the Pokemon
- **attack:** The Base Attack of the Pokemon
- **defense:** The Base Defense of the Pokemon
- **sp_attack:** The Base Special Attack of the Pokemon
- **sp_defense:** The Base Special Defense of the Pokemon
- **speed**: The Base Speed of the Pokemon
- **generation:** The numbered generation which the Pokemon was first introduced
- **is_legendary: Denotes if the Pokemon is legendary.**