Le mini projet

1 L'objectif

L'objectif du mini projet est d'éclairer un sujet d'intérêt public (météo, environnement, politique, vie publique, finance, transports, culture, santé, sport, économie, agriculture, écologie, etc) que vous choisissez librement. Vous utiliserez des données publiques Open Data, accessibles et non modifiées.

l'open data (ou donnée ouverte) est une donnée numérique d'origine publique ou privée. Elle peut être notamment produite par une collectivité, un service public (éventuellement délégué) ou une entreprise. Elle est diffusée de manière structurée selon une méthode et une licence ouverte garantissant son libre accès et sa réutilisation par tous, sans restriction technique, juridique ou financière. Wikipedia.

Le choix du sujet est libre.

2 Les livrables

Vous devez produire du code Python, structuré en plusieurs fichiers, permettant d'exécuter un dashboard dans un navigateur standard. Le dashboard contiendra a minima un histogramme et une représentation géolocalisée. Au moins un des graphiques doit être dynamique.

Le code devra recueillir et nettoyer les données, les sélectionner et les organiser pour les représentations graphiques interactives illustrant votre point de vue sur le sujet.

3 Les données

Les données doivent être accessibles publiquement, de façon reproductible, via une ressource statique ou une API publique. Dans tous les cas, le(s) pointeur(s) vers les ressources doi(ven)t être fourni(s).

Les données seront dites "statiques" si elles sont stockées localement sous la forme d'un fichier au format quelconque. Ce fichier sera téléchargeable localement avec:

> python get_data.py

Les données "dynamiques" seront valorisées. Les données seront considérées dynamiques si :

- elles sont stockées sur une page web statique nécessitant un processus de scraping ;
- elles sont générées à la volée par un serveur. On les obtient alors généralement par l'utilisation d'un browser headless, Selenium par exemple ;
- le dashboard met en cache une partie des données avec un mécanisme de rafraichissement.

Dans la très grande majorité des cas, l'ensemble des données (dataset) est présenté sous la forme d'une page de tableur, dont les lignes sont les observations et les colonnes, les V variables (numériques, ordinales ou catégorielles) de ces observations.

Ce dataset doit impérativement posséder les propriétés suivantes:

- il doit posséder un nombre suffisamment grand d'observations pour que le tracé d'un histogramme ait du sens (typiquement plusieurs centaines). Ainsi les statistiques sur les communes françaises sont éligibles (O > 36000), celles concernant les pays conviennent (200, variable selon les sources) mais celles concernant les département français ne conviennent pas (O < 100) ;
- parmi l'ensemble des données disponibles sur chacune des lignes, l'une au moins doit être numérique et non catégorielle. Une donnée non catégorielle posséde une relation d'ordonnancement (plus petit que, plus grand que). Exemple : la pression atmosphérique, le poids, le coût, le nombre, etc Attention à l'utilisation de l'année comme donnée numérique, le plus souvent cette dernière est utilisée comme donnée catégorielle ;

- les observations devront pouvoir être géolocalisées. Exemple : la température mesurée pour plusieurs stations météo, la taille relevée dans des zones géographiques différentes, le point de chute de météorites, etc.. Soit directement si les coordonnées géographiques sont incluses dans les observations, soit indirectement en faisant appel à une autre ressource.

Si le nombre d'observations est très grand, les observations peuvent être sous catégorisées pour donner du sens à l'analyse. Exemple : la température mesurée pour différentes heures du jour et de la nuit, la taille relevée pour chacun des deux sexes, les dépenses de fonctionnement des villes de moins de 5000 habitants, etc

En résumé, pour vérifier que le jeu de données choisi convient, vous devrez donc vous assurer que:

- le nombre d'observations est suffisamment grand ;
- la donnée à représenter sous forme d'histogramme n'est pas catégorielle ;
- une géolocalisation est possible.

Pour la géolocalisation, il est accepté qu'elle soit construite à partir d'un dataset différent de celui utilisé pour l'histogramme, du moment que le contexte des deux jeux de données est le même.

4 Contexte du développement

Le dashboard sera développé avec les versions les plus récentes disponibles au 1er septembre de l'année universitaire :

- du langage Python ;
- de pandas, dash et plotly ;
- et plus généralement des autres packages utilisés dans le mini projet.

C'est dans ces conditions que votre travail sera évalué.

5 Remise du travail

Gardez à l'esprit que votre travail sera déployé dans un contexte différent de celui de votre machine de développement.

Le dépôt Git doit être public et contenir :

- l'ensemble du code strictement nécessaire pour exécuter le projet ;
- un fichier main.py à la racine du dépôt tel que l'exécution de ce seul fichier construise le dashboard ;
- les données si celles ci sont statiques;
- le fichier README formatté en Markdown renseigné avec :
  - un "User Guide" qui permet de déployer et d'utiliser votre dashboard sur une autre machine ;
  - un rapport d'analyse qui met en avant les principales conclusions extraites des données ;
  - un "Developper Guide" qui permet de comprendre l'architecture du code et de modifier ou d'étendre celui ci.

Le fichier requirements.txt doit contenir la liste des seuls packages additionnels requis tel que la commande qui installe l'ensemble des dépendances nécessaires (et uniquement celles là).

> python -m pip install -r requirements.txt

Après clonage du dépôt et à l'exécution de main.py :

- le code ne doit pas produire de warnings dans la console ;
- le dashboard ne doit pas comporter pas d'erreur de call back.

6 Déploiement du travail

Le projet sera copié sur la machine d'évaluation avec la commande:
> git clone adresse_publique_de_votre_projet
Le dashboard sera lancé avec la commande:
> python main.py
Il sera prudent de reproduire les 2 étapes ci dessus sur une autre machine avant de soumettre votre travail.

7 Evaluation du travail

Le dashboard doit être fonctionnel à la première tentative. Chaque interaction supplémentaire entre l'évaluateur et le binôme sera assortie d'un point de pénalité. Les causes les plus fréquentes

- référence à un fichier local à la machine de développement ;
- utilisation d'un package obsolète ;
- utilisation d'un package non recensé dans requirements.txt.

Votre code doit être structuré en fonctions, classes, et modules.
Il doit être documenté :

- par des noms de variables explicites
- des commentaires précisant pourquoi on exécute les principales instructions
- des docstrings, voire du typing pour les fonctions/classes utilisées

Les structures de données doivent être adaptées.
Les "bonnes pratiques" du langage seront valorisées.

8 Grille évaluation

Ci dessous, une liste non limitative des critères pouvant être utilisés pour l'évaluation. Ces critères ne sont pas d'égale importance.

- Le dépôt
  - montre une utilisation régulière de git ;
  - le README est renseigné ;
  - contient les principales conclusions de l'étude ;
  - contient des informations sur les données utilisées ;
  - contient un User Guide correctement renseigné ;
  - contient un Developer Guide correctement renseigné.
- Le dashboard
  - l'instruction pour le produire est présente et il se lance sans erreur ;
  - contient les représentations graphiques demandées ;
  - les graphiques sont correctement documentés (titre, label des axes, etc.) ;
  - est dynamique et fluide ;
  - est de façon générale de bonne qualité.
- Le code
  - est structuré en modules, classes, fonctions ;
  - est lisible et commenté ;
  - met en uvre les bonnes pratiques du langage ;
  - les structures de données et les modules sont adaptés
- Les données
  - les données adressent un problème d'intérêt général ;

- les données sont accessibles dynamiquement ;
- les données sont riches et abondantes.

A chaque item, un grade est attribué:

- 4 : tout à fait d'accord
- 3 : plutôt d'accord
- 2 : ni vraiment d'accord, ni vraiment en désaccord
- 1 : plutôt en désaccord
- 0 : tout à fait en désaccord

Ces critères garantissent un projet complet, structuré et développé selon les bonnes pratiques.