Collecte et Stockage de données

(Ce travail compte pour 25% de la note globale et à rendre le 3 janvier 2021)

Indications

- Le projet est à rendre le 3 janvier 2021 au plus tard 23h59.
- Le projet peut être fait de façon individuelle ou par un groupe de 3 participants au maximum
- Le projet est à envoyer à l'enseignante par courriel ou à travers Mio
- Le (s) participant(e)(s) sont amené(e)(s) à produire du code (programme).
- Veuillez utiliser les commentaires à l'intérieur de chaque programme pour mettre vos noms et vos matricules au tout début ainsi que la section et le numéro de question auxquels le code répond et cela en utilisant toujours des commentaires.

Travail à rendre par courriel

- Votre courriel comprendra dans son objet
  Collecte et Stockage de données 420-A53-BB_projet_nom(s)_participant(s)

- Vous devrez rendre en tout **4 fichiers (format compressé)**

1) Un 01 fichier **"solution.pdf"** qui ne devra pas dépasser 5 pages en incluant l'explication de la solution et les sorties

2) Un 01 fichier **"codepython.py / ipynb "** qui comprendra tout le code python avec des commentaires
   (Veuillez S.V.P commenter votre code)

3) Un 01 fichier **"codeSQL.sql"** qui comprendra tout le code SQL
   (Veuillez S.V.P commenter votre code)

4) Un 01 fichier **"data_sortie.csv"** qui représente votre table de données finale sur laquelle vous allez faire vos analyses préliminaires.

Critères d'évaluation:

1) Bonnes réponses aux questions posées.

2) Code concis, bien documenté, sans bugs ni erreurs et qui fonctionne

3) Projet fait par un groupe : Évaluation individuelle de chacun de vos coéquipiers
   (Note sur 10 pour ce qui est de la contribution de chaque coéquipier dans l'équipe).
   Envoyer un courriel pour vos co-équipiers

ATTENTION

- Le non-respect de ces règles peut entraîner des points en moins sur la note du projet

- AUCUNE FORME DE PLAGIAT NE SERA TOLÉRÉE dans ce projet.
  (Tout plagiat est dénoncé et peut causer l'échec à ce cours)

Collecte et Stockage de données

(Ce travail compte pour 25% de la note globale et à rendre le 3 janvier 2021)

Python (Importation et Manipulation de données)

1.1) Aller sur le site data.gov https://www.data.gov/ , charger et fusionner les jeux de données

- Chicago Park District: Movies in the Parks 2015
- Chicago Park District: Movies in the Parks 2016
- Chicago Park District: Movies in the Parks 2017

Calculer le nombre de lignes de votre table de données résultante (dataframe) (avec les fichiers fusionnés) sans et avec redondances des données (doublons s ils existent). Afficher des statistiques sur ces données. Préciser toutes les vérifications que vous avez effectuées sans faire de nettoyage sur les données.

1.2) Charger le jeu de données movies (movies.csv) dans une table de données (dataframe) et répondre aux questions suivantes

- Qui est l'acteur principal ayant été dans le film le plus couteux de la table de données? Quel est le montant du budget de ce film?
- Quels sont les 2 films ayant eu la plus grande rentabilité de notre table de données?
- Lister des titres de films dans lesquelles ont tourné votre acteur (actrice) préféré(e)?

1.3) Charger le jeu de données Film Locations in San Francisco du site data.gov https://www.data.gov/ Transposer l'affichage de la table de données et donner le nombre et les endroits de tournage (location) par film. Créer une nouvelle table de données n'ayant que ces trois attributs et afficher les 10 premiers enregistrements.

- Titre (title)
- Release year
- Production

1.4) Charger le jeu de données « data_employe.profiles.txt ».

- Déterminer la classe des travailleurs qui ont un capital gain (gain capital) le plus élevé et afficher uniquement l'âge, genre, classe de travail (class work) et le salaire ?

- Déterminer les employés qui sont susceptibles d'avoir un capital perte (capital loss) élevé (choisir les attributs qui vous semblent pertinents pour faire cette investigation et dites pourquoi ?

- Selon vous quelles sont les facteurs (combinaison d'attributs) ou le capital gain est au maximum et la perte en capitale (capital loss) est au minimum avec un salaire moyen par rapport à tous les employés ?

(Ce travail compte pour 25% de la note globale et à rendre le 3 janvier 2021)

SQL & Python

Pour faire cette 2eme partie du projet, il faudrait utiliser le module sqlite3. Vous n'avez pas besoin d'installer ce module. Voici ci-dessous un exemple d'utilisation

```
Exemple:
import sqlite3

## creer base de donnees si elle n existe pas
conn = sqlite3.connect('test.db')

conn.execute('''CREATE TABLE COMPANY
       (ID INT PRIMARY KEY     NOT NULL,
       NAME           TEXT    NOT NULL,
       AGE            INT     NOT NULL,
       ADDRESS        CHAR(50),
       SALARY         REAL);''')
print "Table created successfully"

conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
      VALUES (1, 'Paul', 32, 'California', 20000.00 )")

conn.commit()

cursor = conn.execute("SELECT id, name, address, salary from COMPANY")
for row in cursor:
   print "ID = ", row[0]
   print "NAME = ", row[1]
   print "ADDRESS = ", row[2]
   print "SALARY = ", row[3], "\n"

conn.close()
```

SQL- Partie 1:

1.5) Importer le jeu de données movies (movies.csv) dans une table de données qui se nommera data_movies en gardant seulement les variables suivantes :

- num_voted_users,
- country,
- movie_facebook_likes,
- director_facebook_likes,
- aspect_ratio,
- movie_title,
- actor_1_name,
- imdb_score,
- duration

Cette table de données comprendra seulement les films qui ont plus de 52000 personnes ayant voté.

1.6) Créer la variable « popularite » qui prendra les valeurs suivantes :

- Faible - Si le nombre de « Likes » sur FACEBOOK du film est en-dessous de 5000.
- Moyenne- Si le nombre de « Likes » sur FACEBOOK du film est compris entre 5000 et 24999.
- Forte- Si le nombre de « Likes » sur FACEBOOK du film est supérieur ou égale à 25000.

1.7) Renommer la variable « duration » en « temps » et la variable « movie_title » en « titre »

SQL & Python- Partie 2:

Au sein de votre programme, veuillez commenter les différentes opérations que vous avez effectuées

2.1) Charger les 6 fichiers (data_credit*_infos et data_credit*_socio) dans 6 tables

2.2) Fusionner les de façon à avoir toutes ces données toutes ensembles dans une table de base de données. Cette nouvelle table a comme nom data_credit
Aucun enregistrement (donnée) provenant des fichiers ne doit être perdu lors de la fusion et assurer qu'il n'y a aucun doublon dans cette nouvelle table (par doublons, nous voulons dire qu'aucune ligne de données ne se retrouve plus d'une fois dans la nouvelle table) ?

2.3) Afficher le nombre de clients qu'il y a dans la région de Toronto et après de Québec ainsi que l'âge moyen des clients de chaque région ?

2.4) Dans cette nouvelle table, veuillez afficher le nombre de produits moyen, le minimum du nombre de produit, le maximum du nombre de produits et cela grouper par ville. Nous voulons seulement la ville de Montréal et de Québec. Quelle est la ville qui a la plus petite moyenne ? Est-ce une grande différence?

2.5) Créer deux nouvelles variables pour grouper la variable « âge » avec un pas de 5 et après un pas de 10 en commençant par la valeur 10 (des groupes 10-15 pour les pas de 5) et (des groupes 10-20 pour les pas de 10, etc.) jusqu'à atteindre l'âge maximum de la table de données. Vous nommerez ces variables « age_group_5 et age_group_10 » (4 requêtes maximum).

2.6) Afficher la proportion totale de clients qui possèdent soit un mauvais crédit ou un bon crédit. Quelle est la catégorie où il l'y a le plus de mauvais crédit ?

2.7) Y a-t-il une préférence par territoire pour ce qui est des montants prêtés à la clientèle. Pour cela, vous devez exclure les personnes ayant entre 10 et 24 ans et les personnes de plus de 70 ans. Veuillez ordonner (ordre décroissant) cette sortie par le nombre de clients qu'il y a dans chacun des groupes

SQL -Partie 3:

SQLite ne gère pas les fonctions de fenêtrage. PostgreSQL est ici plus adapté.

3.1) Charger le jeu de données « data_employe.profiles.txt ».
3.2) Comparer le salaire d'un employé avec le salaire moyen des autres employés
3.3) Déterminer les rangs des employées par gros salaire
3.4) Détermine les 10 premiers rangs des employées par âge, capital-gain et sexe
3.5) Déterminer la somme de la de perte en capitale (capital loss) par sexe, âge et éducation