Data Visualization

Alicia BENIDDIR - Elisa-Souad GOULAHSEN - Bao HA - Moubina MAVOULANA

I. Comment lancer notre projet

Ayant eu certaines difficultés avec d'autres logiciels, nous avons opté d'utiliser des outils qui demandent d'être lancés localement. Bien que vous nous aviez dit de ne pas mettre de dataset, étant un nouveau dataset modifié, nous devons vous le fournir pour que vous puissiez exécuter notre projet.

1. PostGreSql avec PgAdmin

Tout d'abord pour Windows, il va falloir charger le dataset sur pgadmin, pour cela tout d'abord il faut disposer de PostGreSQL. Voici le lien pour le télecharger et installer : https://www.postgresql.org/download/

Pendant l'installation, il vous sera demandé de créer un compte avec un mot de passe, à ne pas oublier ! Ensuite, il faut installer l'outil de gestion de base de données https://www.pgadmin.org/

Par la suite, il faudra créer une table sur pgadmin, sur le menu à gauche, sélectionnez Create puis Table qui va s'appeler "musique", puis cliquez sur la partie "columns" et entre les colonnes: artist de type text, album de type text, song de type text et date de type date. **Mettre les mêmes noms que la base de données est important**, cela permettra de lancer le prochain fichier avec le code plus facilement, sinon il faudra modifier les attribut dans les requêtes.

Avant de passer à la dernière partie, il faut récupérer le LastFM.zip, le décompresser pour obtenir le fichier csv.

Pour terminer avec la partie pgadmin, il manque plus qu'à faire un clique droit sur notre table qui va apparaître sur le menu et faire import csv en sélectionnant l'emplacement ou le fichier est stocké.

2. VsCode avec Dash Bootstrap Component

Maintenant que la base de données est créée, il faut récupérer le fichier DVIZ_1.py et l'ouvrir sur VsCode. Dans celui-ci, il va falloir modifier les lignes 9 et 10 avec les données de votre postgresql. Après avoir changé les attributs, il manque plus qu'à exécuter le fichier sur le terminal. Pour cela, il faut juste indiquer le chemin ou est stocker votre fichier python est appuyer sur entrée.

Le dashboard devrait se lancer sur une page internet.

II. Choix des outils

1. PostgreSQL (PgAdmin4)

Nous avons choisis PostgreSQL pour charger les données (les fichiers .csv). PostgreSQL est un choix parfait en fonctionnalités pour les données relationnelles. En effet, Postgres nous permet de stocker en toute sécurité des données volumineuses. Il nous aide également à créer les applications les plus complexes, à exécuter des tâches administratives et à créer des environnements intégraux.

Parmi les outils de gestion de PostgreSQL, nous avons choisis PgAdmin 4 en raison de son interface convivial.

2. Vscode

Vscode est un environnement de développement complet et accessible facilement pour tout le monde, en plus de permettre de travailler avec énormément de langages, comme python, C++, JS … Nous avons choisi Vscode car c'est un environnement sur lequel nous sommes habituées et que nous connaissons très bien. Il permet de lier facilement le lier plusieurs technologies ensembles, comme git, ou encore des base de données.

3. Dash / Bootstrap component

Pour l'affichage de notre dashboard on a opté pour Dash qui est un framework open source. Cela nous a permis de créer une petite application web très user friendly et facile d'approche afin d'afficher nos données qui ont auparavant été traiter sur vscode et pgadmin.
Afin d'avoir quelque chose de plus agréable visuellement parlant, on a utilisé Bootstrap component qui est une librairie pour les graphiques avec Dash.

III. Les demarches

1. **Le chargement du csv via pgAdmin**

   On a récupéré tous les csv qui ont été concaténés au préalable. On a créé une table “music” dans notre base de données et on a chargé en faisant attention de bien avoir un encodage utf-8 afin de pouvoir lire les caractères spéciaux.

2. **Connexion de la base de données avec Vscode**

   On a par la suite utilisé l’IDE Vscode pour pouvoir traiter les données (ETL) en effectuant une connexion à notre base de données pgAdmin.
   Une fois la connexion à la base de données réussie on a transformé notre csv en un dataframe en utilisant la bibliothèque pandas.n

3. **Nettoyage des données**

   Avant de passer aux requêtes, on s’est assuré que les données soient assez nettoyées. Les données vides n’ont pas été supprimées car lors du “count”, celui-ci ne les prend pas en compte. On a juste changé le type de la colonne date pour pouvoir avoir le même type de façon uniforme dans la colonne car nous allions effectuer des requêtes sur celle-ci.

4. **Les requêtes**

   Au vu de la grandeur du dataset, on a préféré passer par des requêtes en sql et par la suite de les transformer en dataframe cela facilite le temps de chargement des données.

5. **L’affichage du dashboard**

   Pour pouvoir afficher les différentes requêtes, on va utiliser le framework Dash qui va permettre d’afficher des graphiques. Pour rendre la chose un peu plus agréable visuellement on a opté pour une librairie de Bootstrap. Nos figures proviennent donc d’un mélange de plotly la librairie de python et de bootstrap ainsi que nos boutons afin de rendre les choses plus interactives.

IV. Remarques

Comme dit au début, nous nous sommes dirigés vers Databricks pour la partie ETL, puis Power BI pour la visualisation. Cependant, ayant rencontré des difficultés au long de la réalisation (nous expliquerons le processus à l’oral), nous avons finalement décidé de passer à la démarche expliquée précédemment.