

# Data Visualization Dashboard

## 1. Project Overview

### Objective

The purpose of this project is to create a dashboard based on key performance indicators (KPIs) calculated from a given dataset and visualize it. The data is organized as csv files in a zip file, in which we can find 4 columns :

- The **Artist** name
- The **Album**
- The **Title** of the song
- The **Date** of listening.

Each csv file is named according to the name of the **User**.

---

## 2. Workflow

Here are the 3 main steps the application follows.

### 1. Data Retrieval

- The dataset is obtained by downloading the zip file on Teams. There are 63 csv files.
- The user can add any csv if it respects the same structure than the others such as : “artist name”, “Album”, “Title”, “Date”. The name of the file must be the username of the listener.

### 2. Data Processing

- Execute “app.py” to import new csv files in the dataset.

### Upload CSV Files

Select one or more CSV files:

Sélect. fichiers

Aucun fichier choisi

Upload

- Verify the presence of a specific file by entering its name

**Search for a CSV File**  
Enter CSV file name:

- Clean the Data and add the name of the columns
- Upload the dataset to the Relational Database MySQL

### 3. Data Visualization

- Use **Power BI** to create interactive visualizations by importing the SQL Database

---

## 3. Tools and Technologies

### Backend

- **Python Flask:** For developing the ETL pipeline and web application.
- **MySQL:** For storing and querying the dataset.

### Visualization

- **Power BI:** To create dynamic and interactive dashboards for data visualization.

---

## 4. KPIs

The following KPIs were calculated:

- **Most listened-to track of all time.**
- **Most listened-to track for each week.**
- **Most listened-to album of all time.**
- **Most listened-to album for each week.**
- **Cross-tabulation:** Number of listened tracks by listener and artist.
- **Top 10 biggest listeners:**
  - **All time.**
  - **Each week.**

## 5. Implementation Details

### 1. Data Extraction and Loading

- The dataset is fetched and uploaded using a Python Flask-based web interface.
- The data is cleaned and loaded into a **MySQL database**.

### 2. KPI Calculations

We calculated all KPIs with the Power Query Editor

- The most listened song of all time :

We copy the dataset from the original one so that we don't modify it.

We apply an aggregation to get the count of all songs.

We sort the resulting table by descending order.

We only keep the first row which is the most listened song.

Ex:

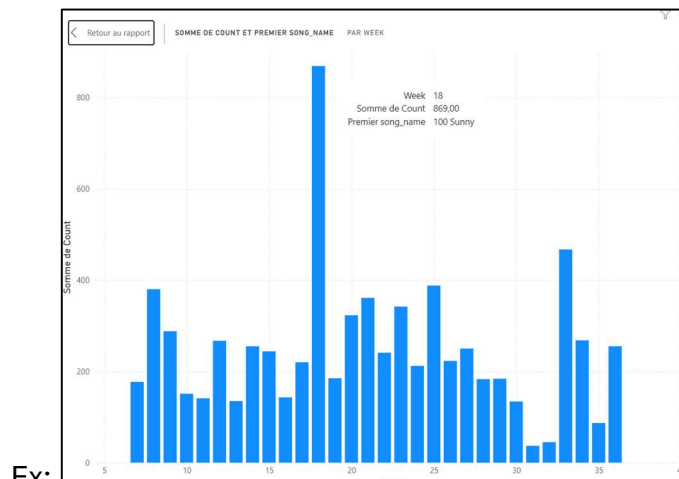
Count	song_name
186 073,00	Stay Alive (Prod. SUGA of BTS)

- The most listened songs per week :

We create a new column "week" that indicates the week number of the year.

We group by song title and by weeks to get the number of listening of each track per week.

We sort the resulting table in ascending order according to "Week" column.



- Most listened-to album of all time.

We apply an aggregation to get the count of all albums.

We sort the resulting table by descending order.

We only keep the first row which is the most listened album.

Ex:

album_name	Count
Stay Alive (Prod. SUGA of BTS)	185 780,00

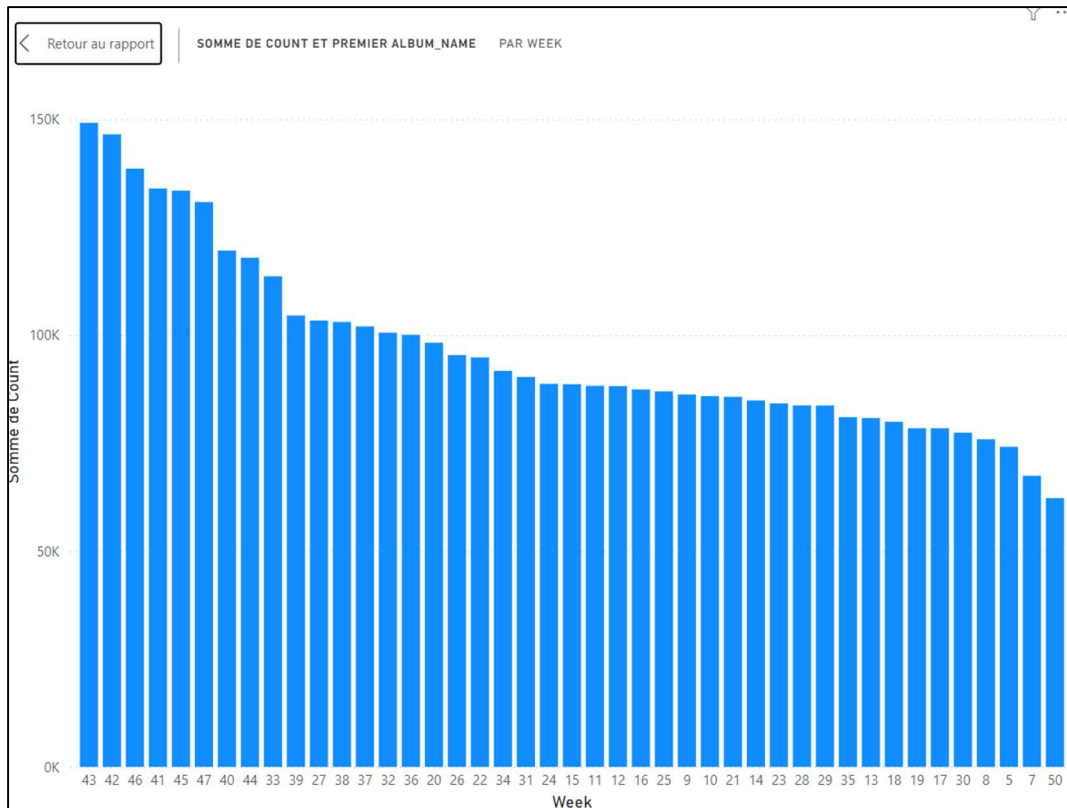
- Most listened-to album for each week.

We create a new column “week” that indicates the week number of the year.

We group by song title and by weeks to get the number of listening of each track per week.

We sort the resulting table in ascending order according to “Week” column.

Ex:



- Cross-tabulation: Number of listened tracks by listener and artist.

We calculate the total of listening per artists

We sort the artist per Total\_Count descending order and we keep the N most listened artists

Filter the initial data to keep only the selected artists

Group the data per user and artist

Group the total of listening per user

We add the column “UserTotal” to “GroupedData” and column for normalization

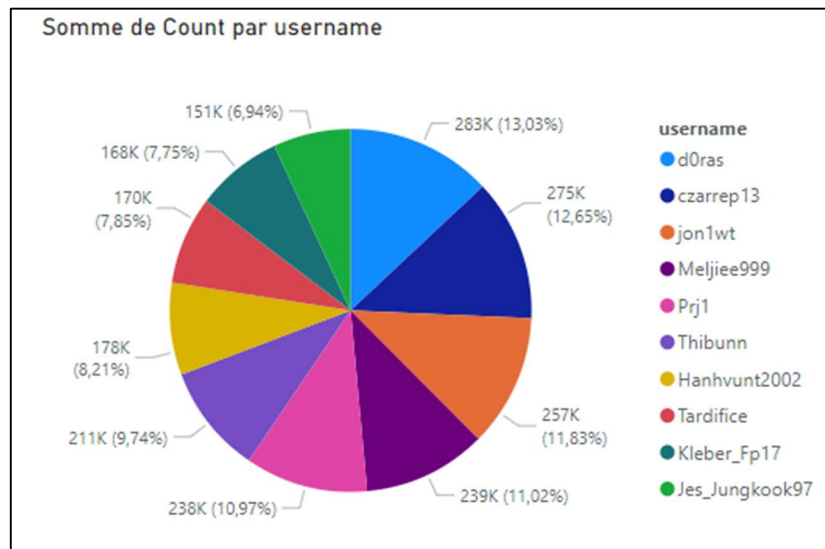
Ex:

username	Adele	Akira Yamaoka	ALT 236	ALT236	Amaranthe	Ariana Grande	Audius
AkaGambit							
alexray							
animefreakben	30	114				908	
apatel158	18					64	
artangelo	4					22	
AscendingNode	1	8		794			
avarthar							
Total	15667	6820	4223	5789	4640	16275	

- Top 10 biggest listeners All time :

We group by username

We sort the resulting table by descending order and we keep the 10 first rows



Ex:

- Top 10 biggest Each Week :

We add the column “Week”

We group by Week and we sort by week ascending order

We group every listener per week and only keep the top 10 so that we get a new column “Top10” that contains the 10 biggest listeners per week

We expand the table to print the name and the count of the biggest listeners

Week	username	Count
1	alexray	537
1	animefreekben	575
1	apatel158	546
1	artangelo	551
1	avarthar	68
1	czarrep13	729
1	d0ras	358
1	DuckDAWorld	1112
1	Guigt77	1207
1	Tardifice	406

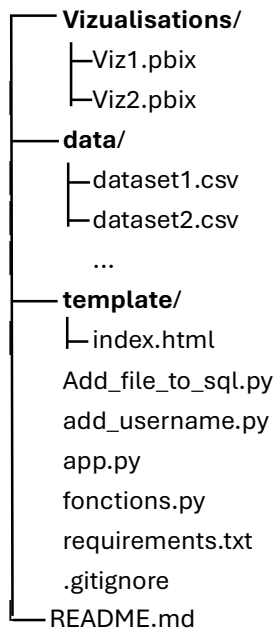
Ex:

### 3. Visualization with Power BI

- Data is imported into Power BI from the MySQL database using the **ODBC connector**.
- Dashboards include:
  - Bar charts for top listeners and albums.
  - Cross-tabulated tables for listeners vs. artists.

## 6. Folder Structure

Application\_of\_big\_data/



## 7. How to Run

### Step 1: Run the Flask Application

You need to modify the parameters in the `Add_file_to_sql.py`, by putting your mysql workbench username and password :

```
# Database connection
def connect_to_database():
    return pymysql.connect(
        host='localhost',
        user='root',
        password='root',
        database='dataviz_m2_proj',
        charset='utf8mb4',
        autocommit=False # Disable autocommit for better batch performance
    )
```

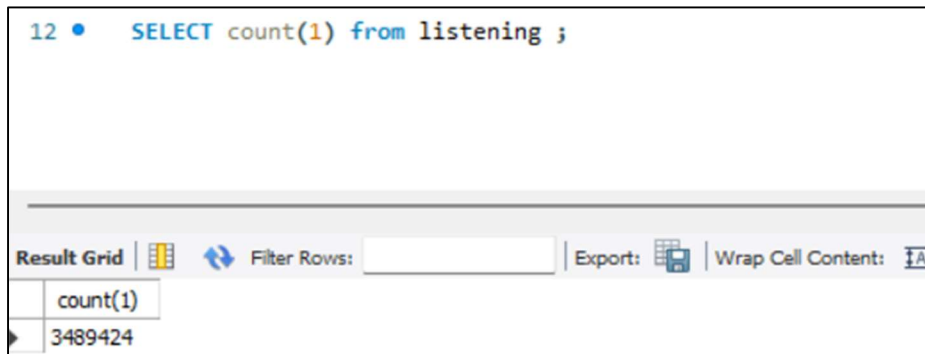
Then you can run it by using the command “python app.py” in your terminal :

```
PS C:\Users\romai\OneDrive\Bureau\S9\Applications of Big Data\VIZ\Dashboard> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 128-786-587
```

After that, you can add files to your database by going on the link : <http://127.0.0.1:5000/>

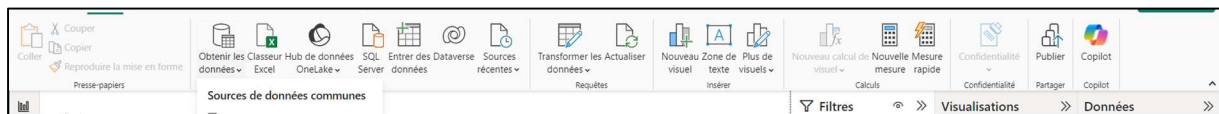
You can select files, one or several, then click on “upload”. Once it is uploaded, you can click on “start integration”. You will get a message once integration is done.

You can also make sure that all data was added to your database by counting data in mysql workbench :



## Step 2: View Dashboards in Powerbi Desktop

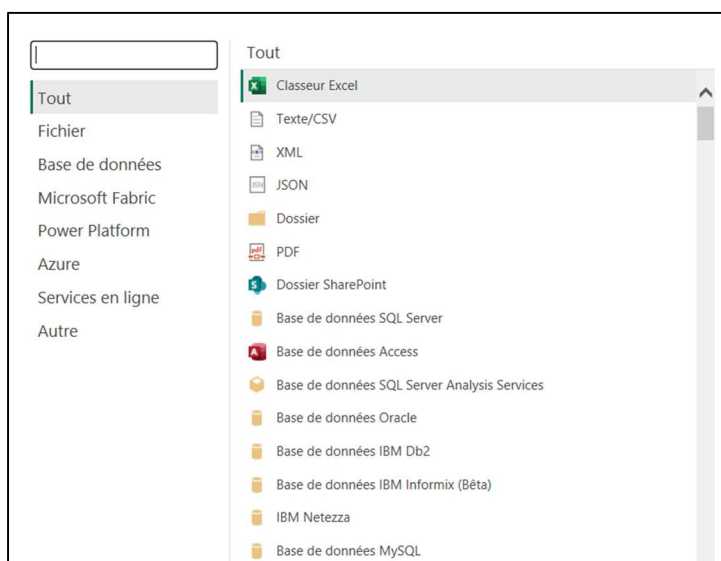
When you launch powerbi desktop, you first need to import data from you mysql workbench dataset. For that, you go in “obtenir les données” :



Then you go in the “plus” section :



You need to select “base de données mysql” :



Then you fill the sections “Serveur” and “base de données” with your informations :

## Base de données MySQL

**Serveur**

localhost:3306

**Base de données**

dataviz\_m2\_projet

After that, you just need to put your identification informations to connect :

Windows  
  
 Base de données

### Base de données MySQL

localhost:3306;dataviz\_m2\_projet

Utilisez les informations d'identification Windows pour accéder à cette base de données.

☐ Utiliser mes informations d'identification actuelles  
☒ **Utiliser d'autres informations d'identification**

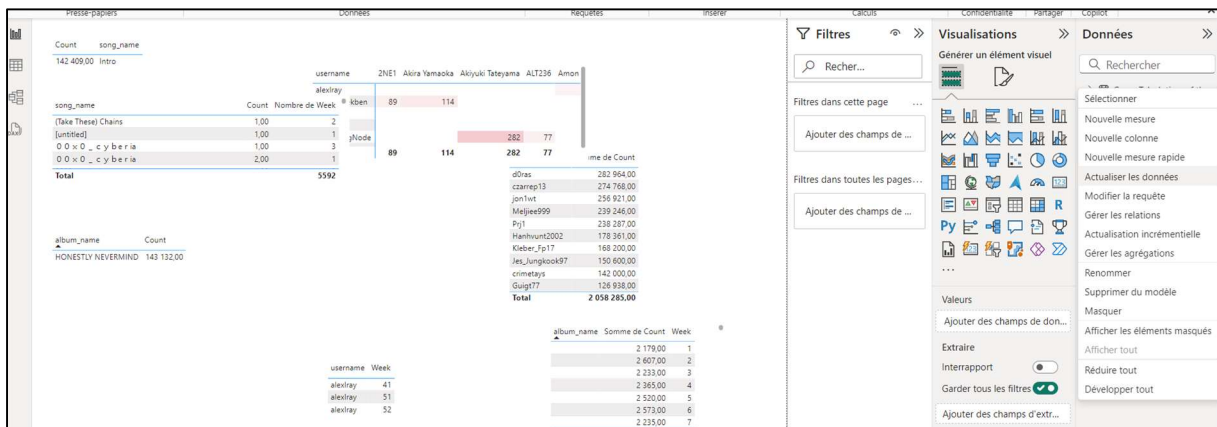
Nom d'utilisateur

Mot de passe

Sélectionner le niveau auquel appliquer ces paramètres

localhost:3306

Once it is done, you just need to refresh data for each request :



The screenshot shows the Tableau Desktop interface with the following components:

- Left Panel (Marks Card):** Contains a table with columns: Count, song\_name, album\_name. The data includes rows for 'Intro', 'Chans', 'cyberia', and 'NEVERMIND'.
- Right Panel (Filters and Visualizations):** Includes a 'Filtres' section with a search bar and a 'Visualisations' section with various chart types (Bar, Line, Map, etc.).
- Bottom Panel (Data Table):** Displays a table with columns: username, Week, and Sum of Count. The data includes rows for 'alexray', 'alexis', and 'alexis'.