## Data Application Report -SUBJECT 3 – DATA VISUALISATION

Git URL: https://github.com/ChessMartin/big-data-viz.git

## Our team

LE FORMAL, MARTIN

LIU, YINGDA

CHEN, YANG

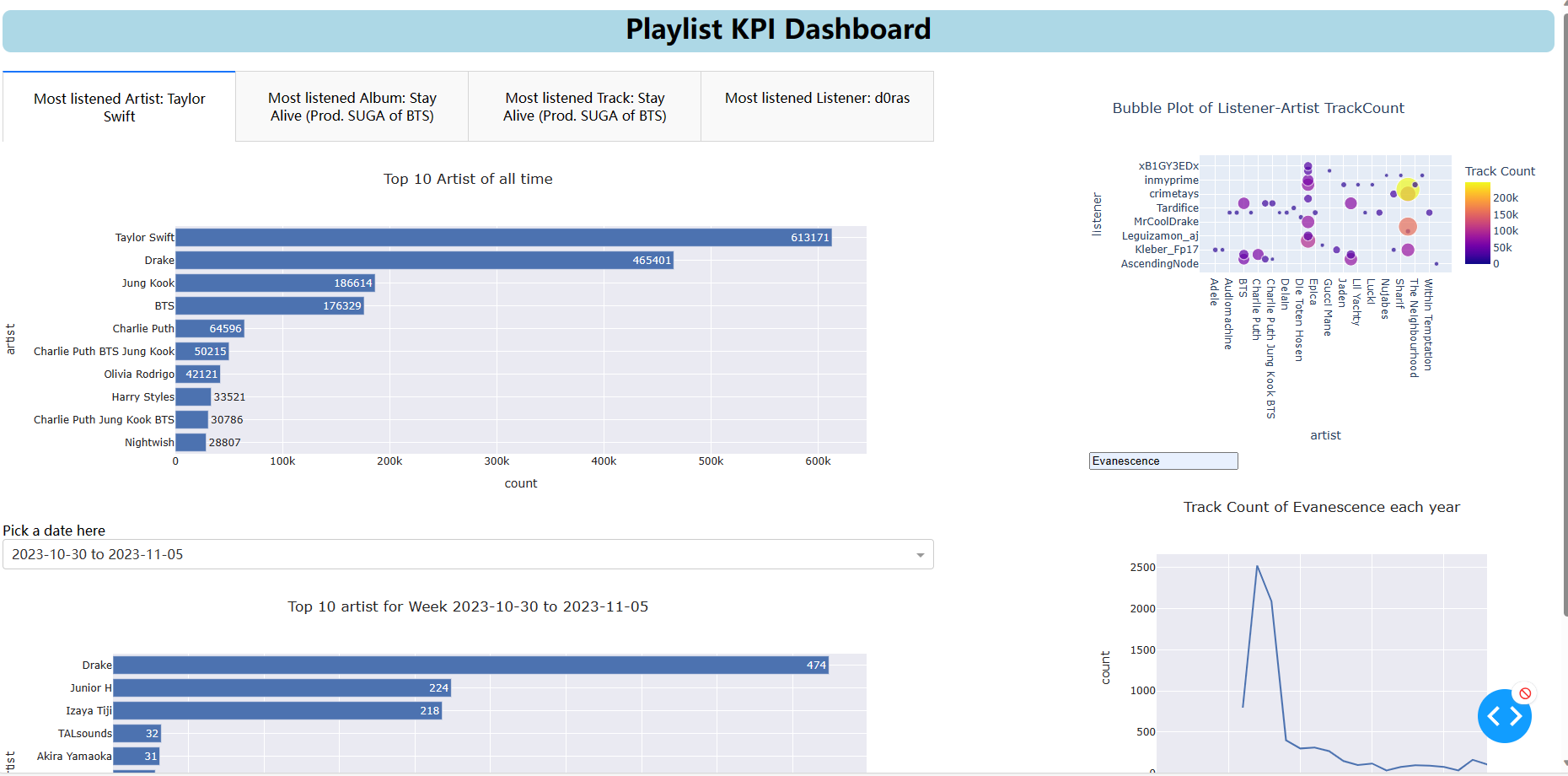
## What is it?

First, we get a set of CSV files, each named after a listener, containing key information about the music being played. We read each of these CSV files and imported them into a pre-created MySQL database. This step ensured that we could easily analyse the data in the database. In the MySQL database, we designed a table structure adapted to music playback KPI analysis. The key fields include artist, album, track, date, etc. for subsequent SQL queries. By writing a series of SQL statements, we calculated a variety of key KPIs related to music playback, such as most popular artists, albums, songs, and most active listeners.

Using the Plotly library in Python, we visualised the results of the SQL queries. This covers a variety of chart types, including bar charts, line charts, bubble charts, to visualise trends and statistics for music playback KPIs. With the interactive features provided by Plotly, users can easily interact with the charts to gain insight into the data. Finally, we created an interactive dashboard using the Dash library in Python. This dashboard summarises all the music playback KPI visualisations and allows users to easily switch to the KPI visualisation of interest by selecting a different tab.

## What it does?

Front-end:The dashboard below is divided into two parts, the left half can choose the topic you are interested in to view the kpi, for example, if you choose to view about artists, you can firstly see the most popular artists' names on the tab, and after clicking to select Artist Tab, you will be able to view the ranking of the top ten most popular artists, and you can also select the date of a certain week through a drop-down box to view the ranking of the top ten most popular artists within that date range. There are two visual charts in the right half, the bubble chart shows the relationship between listeners and artists and the corresponding number of songs, and the line chart shows the trend of music played by the selected artist with a yearly reference.



## What tools we use

Regarding the tools, we decided to go for a code oriented option more specifically only python frameworks. For the front end part we use the Dash framework provided by Plotly, that allows us to efficiently create a web app for data visualisation. We could have chosen another framework but it would require a perfect relation with the database which would be much harder to test since the communication only relies on drivers which we have less control over. For the ‘back-end’ part it is a simple python package that we called api that allows the user to populate and clear the database with very few risks of an error.

## How does it install and work?

There are three files in the Dashboard folder.

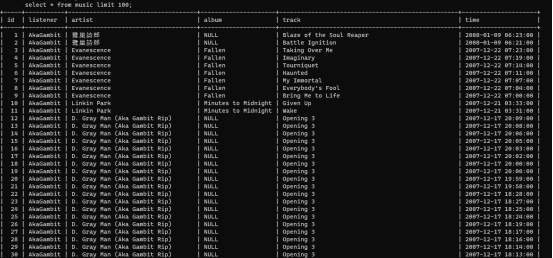
The viz\_database.py is the method for storing data in the database. If you do not have data in the database yet, you need to call this method first. In this part of the storage process, we can store various types of data we encounter into the database, such as null characters and long text.

The viz\_query.py file contains all functions for querying related KPIs, such as finding data such as Most listened track all time. This file provides all the api functions that the front end needs to call.

The viz\_app.py displays the dashboard.Here is the front end, which collects all the data queried using SQL and displays it on the dashboard.

Step1: Go into the viz\_database and viz\_query.py file and change the local mysql information and the data folder\_path to your local path.

The picture below is what the data looks like in the database



Step2:Store all data from the csv folder into the mysql database through the viz\_database.add\_data\_from\_csv() function.

Step3:Run the viz\_app.py file by using “python viz\_app.py” command.Then go to http://127.0.0.1:8050/ to view the dashboard.

## Generative Al

We used chatgpt as a tool to help us complete the project. Sometimes when we encounter some problems, we will explain the problem to chatgpt and ask him to provide some solutions. However, the methods provided by chatgpt sometimes cannot solve the problems we encounter, so we will also go to Google to find solutions.

## Improvements

The most sensible part of the project is the environment, since it relies on many libraries and drivers as well as the database, it would be preferable to encapsulate the whole app with the database in a single docker container via docker compose. It would make the environment more stable by strictly defining the database configuration as well as a defined versioning of the packages. For the development we used a mysql database integrated in a docker container but we did not manage to put the whole app within a single container.