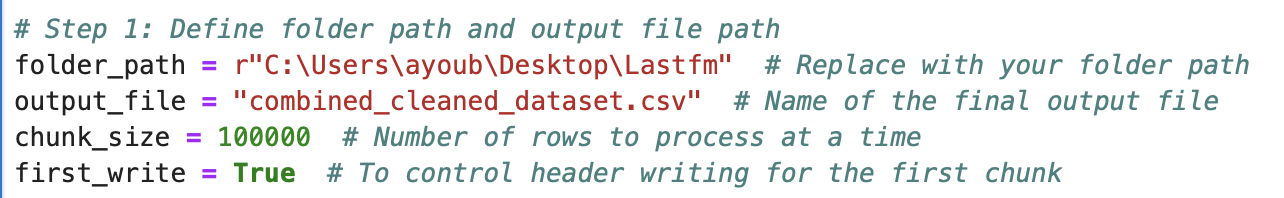
Project Report

SUBJECT 3 –DATA VISUALIZATION

Team members: Chen Renzhong, Adnane Ayoub, Zheng Zeyang, Zhang Letan

**ETL (Extract, Transform and Load the Data)**

The python code is designed to preprocess and combine multiple CSV files containing music listening history into a single cleaned dataset. It ensures data integrity, handles missing and invalid data, and produces a consolidated file ready for further analysis. Below is a detailed breakdown of each step.  
  
**Step 1: Define Folder Path and Output Configuration**

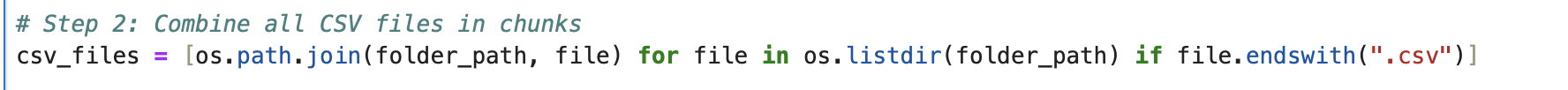
The script begins by defining the folder path containing the CSV files and the output file where the combined data will be stored.  
  
  
• folder\_path: Directory containing the input CSV files.

• output\_file: Name of the final output file.

• chunk\_size: Specifies the number of rows to process in each chunk, optimizing memory usage.

• first\_write: A boolean flag to manage writing headers only once during the first write operation.

**Step 2: Identify Relevant CSV Files**

The script lists all CSV files in the specified folder using the os library.  
• Filters for files ending with .csv.

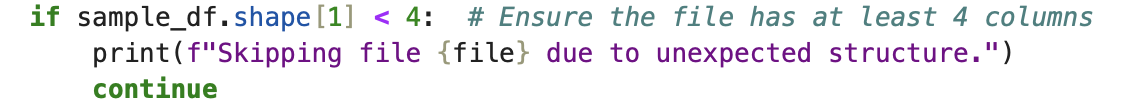
• Constructs full file paths for easier processing.

**Step 3: Process Each File**

Each file is processed individually in chunks to handle large datasets efficiently.

**3.1 Validate File Structure**

Before reading a file, the script checks its structure to ensure it has at least 4 columns.

  
• Reads the first 5 rows to validate the file’s structure.

• Skips files with fewer than 4 columns.

**3.2 Read Data in Chunks**

Files are read in chunks to reduce memory usage, with headers dynamically adjusted if necessary.  

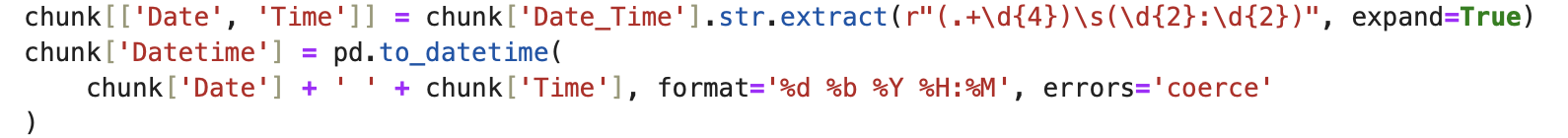

• Reads the file chunk-by-chunk using the specified chunk\_size.

• Skips lines with invalid formatting using on\_bad\_lines='skip'.

**Step 4: Data Cleaning and Validation**

**4.1 Validate and Parse Date\_Time**

The script ensures the Date\_Time column is valid and extracts date and time for conversion to a standard datetime format.

  
• Validates Date\_Time format using regex.

• Extracts and parses date and time components.

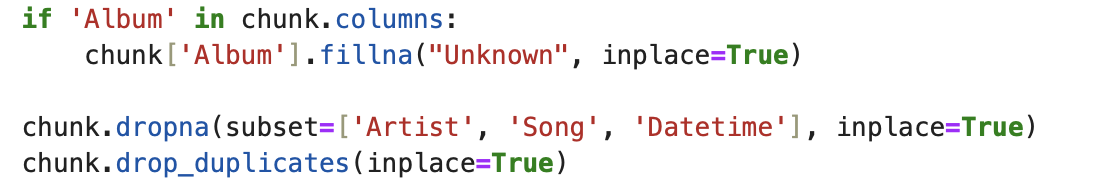
• Converts the data into a unified datetime format.

**4.2 Handle Missing Values**

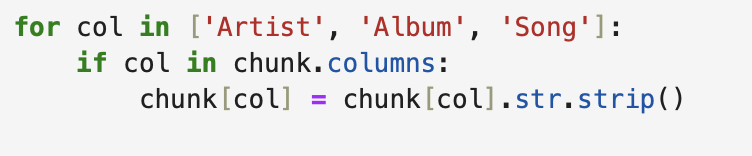
Missing values in the Album column are replaced with “Unknown,” and rows with essential missing data are dropped.

**4.3 Remove Duplicates**

Duplicate rows are removed to ensure data consistency.

  
**4.4 Clean String Columns**

Trailing and leading whitespace from text columns are stripped to maintain uniformity.



**Step 5: Save Processed Data**

Each processed chunk is appended to the output file, writing headers only during the first iteration.



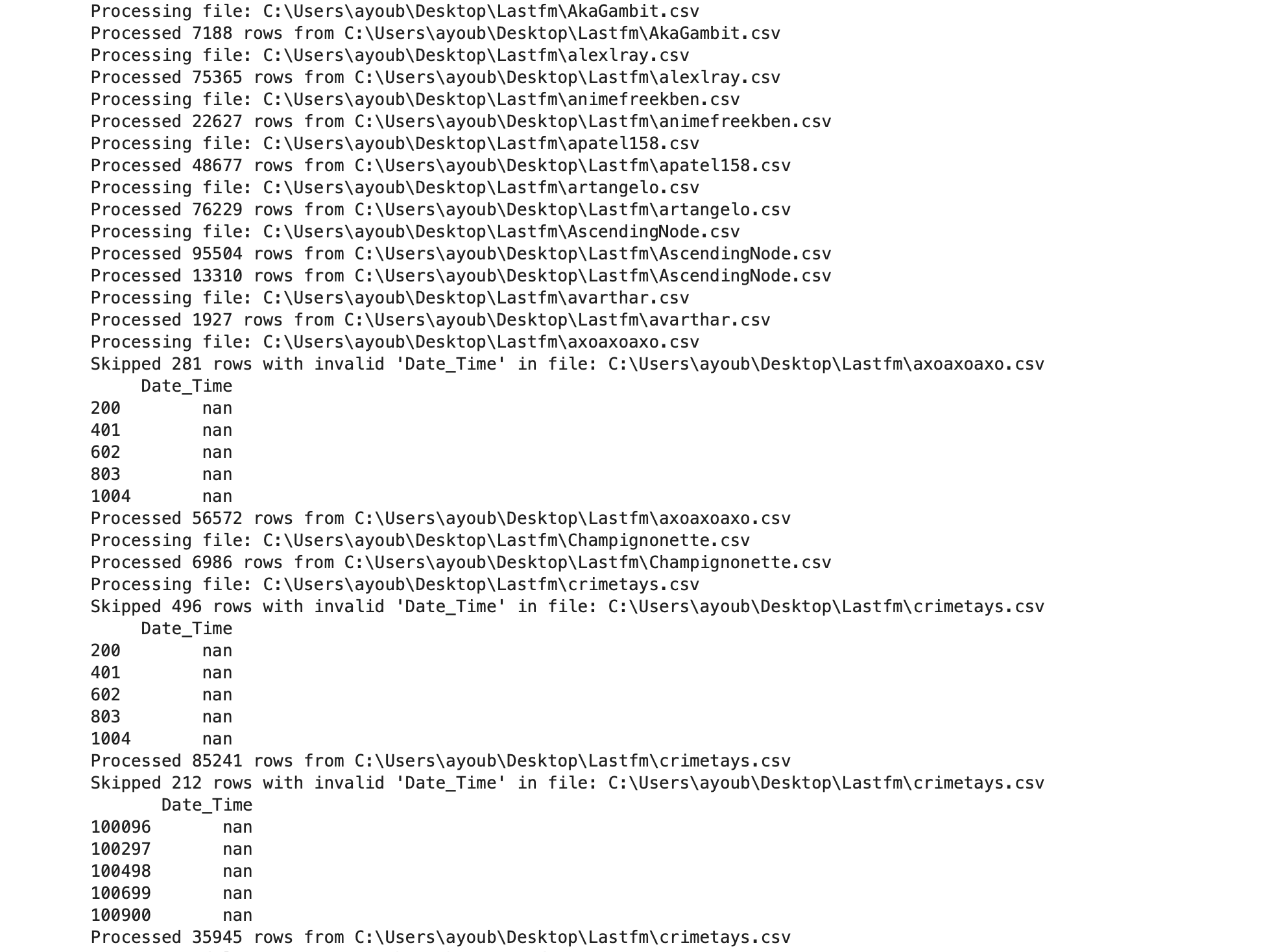
• Appends data if first\_write is False.

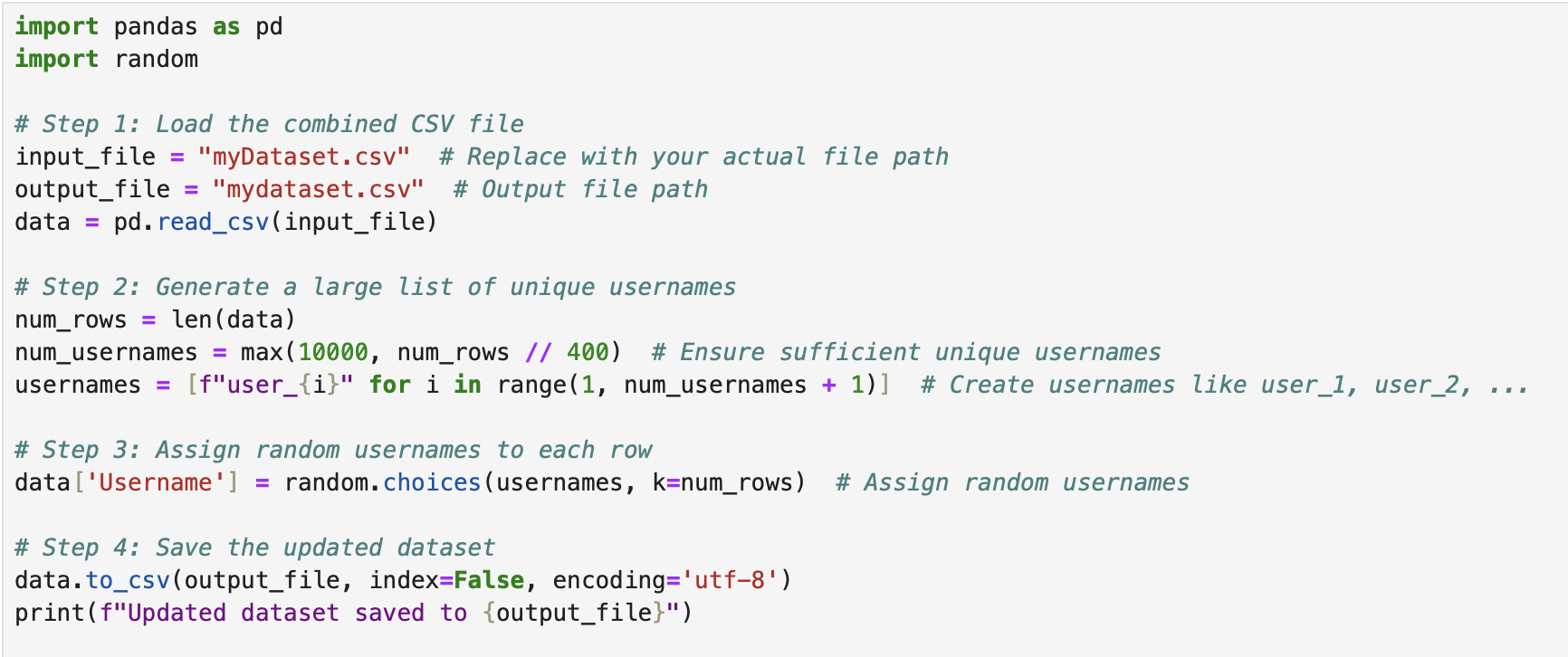
• Writes headers only on the first write.

**Step 6: Error Handling**

Errors encountered during file reading or processing are logged and skipped, ensuring the script continues processing other files.



Output:  


Step7:  


we created a dataset with usernames and use python code to merge it with the previous final dataset.

Usage of python (explaination)

Using Python for ETL provides us with flexibility, scalability, and efficiency. By leveraging libraries like Pandas for data manipulation, we can easily handle large datasets and automate various workflows, making Python an excellent choice for preprocessing our data. The use of Jupyter notebooks in this process also facilitated easy exploration and visualization, ensuring that the data was thoroughly cleaned and prepared before being loaded into Power BI for analysis.

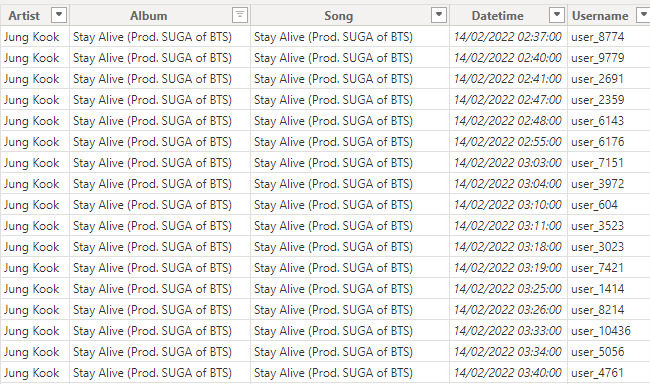
By using Python in this way, we can maintain full control over our ETL process, automate repetitive tasks, and ensure that the data is in optimal condition for our analytical needs.

**Data Visualisation:**

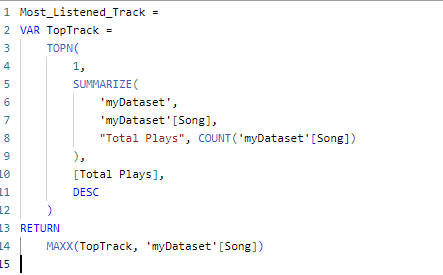
Usage of powerBi (explaination)

We use powerbi to do dashbord, Power BI is a powerful, flexible, and user-friendly tool that enhances data analysis, visualization, and reporting. It streamlines the process of turning raw data into meaningful insights, making it an ideal choice for projects that rely on data-driven decisions. By leveraging its capabilities, teams can improve productivity, collaboration, and the overall quality of their deliverables. **DAX (Data Analysis Expressions)** is the primary formula language used in Power BI, Excel Power Pivot, and SQL Server Analysis Services (SSAS). It enables users to create calculations, aggregate data, and enhance the analytical capabilities of Power BI reports.

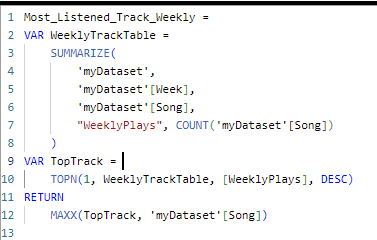
After we generated the preprocessed dataset into a csv file, we uplaoded the csv file into powerBi to create the table(**myDataset**) into powerBi enviremment.



KPI:  
  
Most listened track all time:



Most listened track for each week:



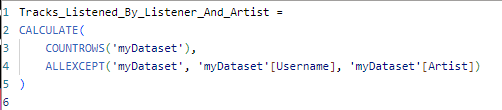
Most listened album all time:



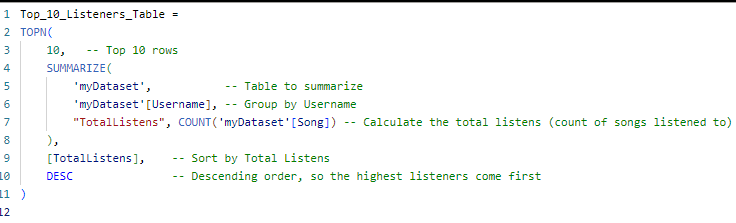
Most listened album for each week:



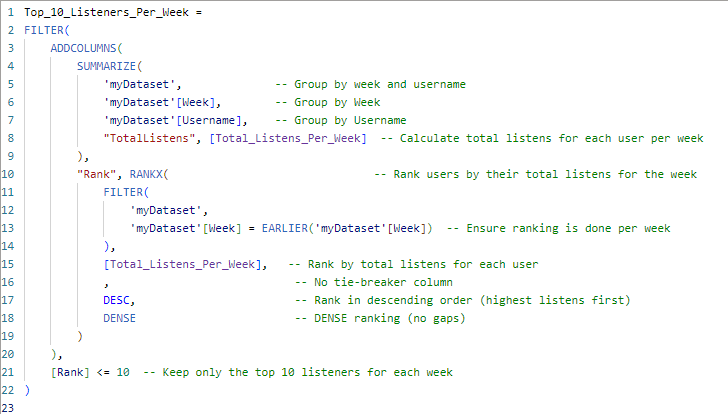
Cross Tabulation of the number of listened tracks by listener and by artist:



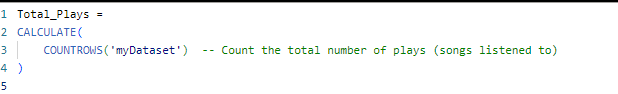
Ranking of 10 biggest listeners all time:



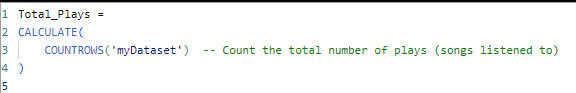
Ranking of 10 biggest listeners for each week:

Total Plays by Hour of the Day (Hourly Distribution):



total plays by year:





**Music listening insight Dashboard (Result)**