Project – Music Streaming and Trend Analysis

This project analyzes music industry data using SQL for data management and Python for data analysis and visualization. It explores platform usage, genre preferences, user demographics, and other music-related metrics. The project uses various statistical and visual methods to gain insights into music trends over time.

Here are some potential analyses we are going to perform:

1. Platform Usage Analysis:

- **Market Share Comparison:** Assess the market share of leading streaming platforms like Spotify, Apple Music, and Amazon Music to understand their dominance and user preferences.
- **User Engagement Metrics:** Evaluate metrics such as average listening time, frequency of app usage, and user retention rates across different platforms to gauge user engagement.

2. Genre Preference Analysis:

- **Demographic Genre Preferences:** Identify which age groups prefer specific genres. For example, younger audiences may favor pop and hip-hop, while older demographics might lean towards rock or classical.
- **Regional Genre Trends:** Analyze how genre popularity varies across different regions, considering cultural influences and regional music scenes.

3. User Demographics Analysis:

- **Age Group Listening Habits:** Examine how different age groups engage with music streaming services, including subscription rates, listening duration, and preferred genres.
- **Gender-Based Preferences:** Explore if there are significant differences in music preferences and listening habits between male and female users.

4. Trend Analysis Over Time:

- **Yearly Growth Patterns:** Track the growth of music streaming services over the years, noting significant increases in user base and streaming hours.
- **Impact of Technological Advancements:** Assess how improvements in internet speeds and the adoption of 5G connectivity have influenced streaming quality and user experience.

5. Device Usage Analysis:

- **Device Preference Trends:** Analyze which devices (smartphones, tablets, desktops, smart speakers) are most commonly used for streaming and how this varies across demographics.
- **Impact of Device Integration:** Evaluate how the integration of streaming services with various devices (e.g., smart TVs, wearables) affects user engagement and content consumption.

6. Subscription and Revenue Analysis:

- **Subscription Growth:** Monitor the growth of paid subscriptions versus free tier users to understand monetization trends.

- **Revenue Streams:** Analyze revenue sources, including subscription fees, advertising, and partnerships, to identify the most profitable areas.

7. Social Media Influence Analysis:

- **Social Media Impact:** Examine how social media platforms influence music trends, such as viral songs or artists gaining popularity through platforms like TikTok.
- **User-Generated Content:** Assess the role of user-generated content, like playlists and reviews, in shaping music discovery and streaming habits.

Dataset Overview:

```
CREATE TABLE dataset(
year INT NOT NULL,
month VARCHAR(20) NOT NULL,
platform VARCHAR(65) NOT NULL,
region VARCHAR(65) NOT NULL,
genre VARCHAR(65) NOT NULL,
song_type VARCHAR(45) NOT NULL,
age_group_18_24 INT NOT NULL,
age_group_25_34 INT NOT NULL,
age_group_35_44 INT NOT NULL,
age_group_45_54 INT NOT NULL,
age_group_55_plus INT NOT NULL,
male_percentage INT NOT NULL,
female_percentage INT NOT NULL,
device_type VARCHAR(45) NOT NULL,
monthly_active_users INT NOT NULL,
monthly_streams INT NOT NULL,
revenue_usd INT NOT NULL,
arpu_usd FLOAT NOT NULL,
paid_subscribers_percentage INT NOT NULL,
free_subscribers_percentage INT NOT NULL,
likes INT NOT NULL,
shares INT NOT NULL,
comments INT NOT NULL,
peak_streaming_hours_utc VARCHAR(45) NOT NULL,
churn_rate_percentage INT NOT NULL
);
```

Database Connectivity:

To perform the analysis, we'll establish a connection to the MySQL database containing the ecommerce data. We'll use the appropriate Python libraries, such as pandas and mysql, to fetch and manipulate the data directly from the database.

```
import mysql.connector
import pandas as pd
# Connect to MySQL
connection = mysql.connector.connect(
   host="localhost",      # Replace with your host (e.g., "127.0.0.1")
   user="root",   # Your MySQL username
   password="1234", # Your MySQL password
   database="project" # Optional: specify database to connect to
)

# Check connection
if connection.is_connected():
   print("Connected to MySQL!")
#create cursor object to execute sql queries
cursor = connection.cursor()
# Query the data from walmart data sale
cursor.execute("SELECT * FROM dataset")
#After fetching data from the database we are storing it into Pandas DataFrame
dataset = pd.DataFrame(cursor.fetchall(), columns=[desc[0] for desc in cursor.description])
print(dataset.head(5))
```

Output:

Description:

customer_data = pd.DataFrame(cursor.fetchall(), columns=[desc[0] for desc in cursor.description])

cursor.fetchall():

This part of the code fetches all the rows from the result set obtained from a database query using the cursor object. The fetchall() method retrieves all the rows as a list of tuples.

columns=[desc[0] for desc in cursor.description]: This part creates a list of column names for the DataFrame. It uses a list comprehension to iterate over the cursor.description, which is a list of 7-item tuples describing the columns in the result set. The [desc[0] for desc in cursor.description] extracts the rst element (column name) from each tuple in the cursor.description and creates a list of column names.

pd.DataFrame(...): This part creates a Pandas DataFrame using the pd.DataFrame() constructor. It takes the fetched data (result of the query) and the list of column names as arguments. The DataFrame is assigned to the variable customer_data.

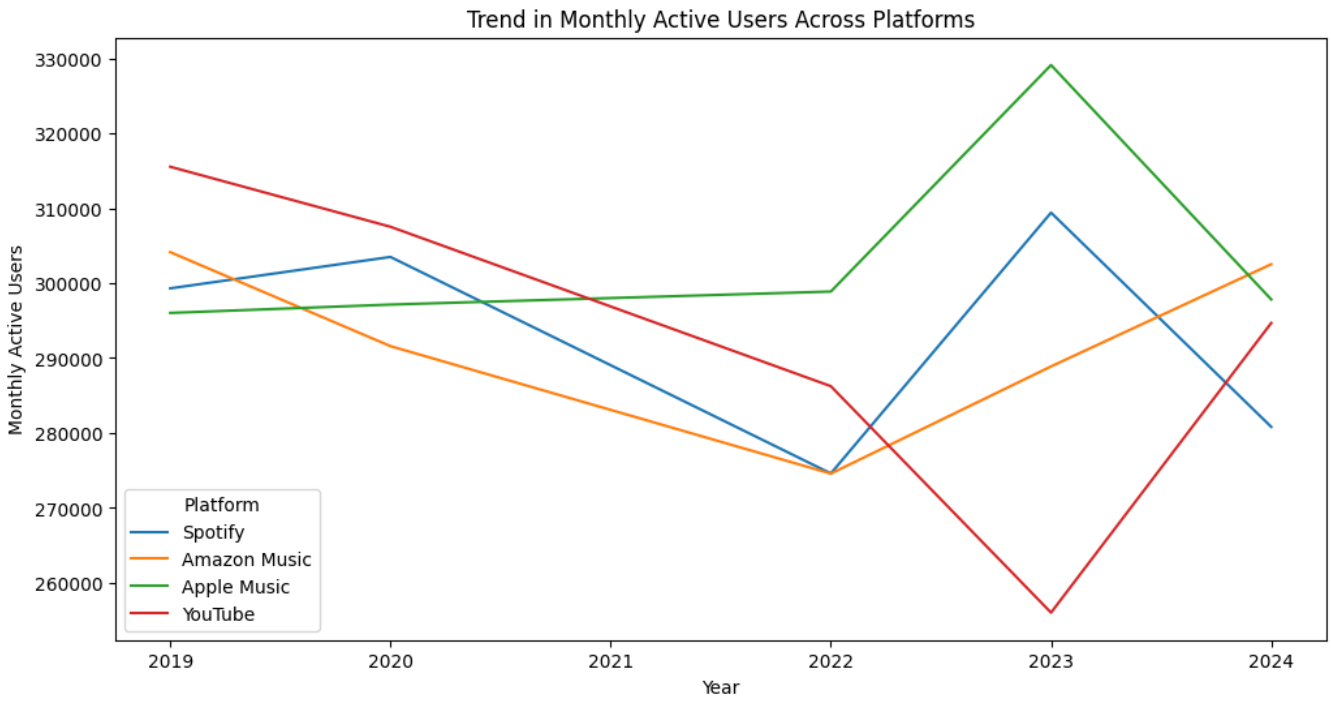
Data Cleaning:

Before proceeding with the analysis, let's perform some basic data cleaning:

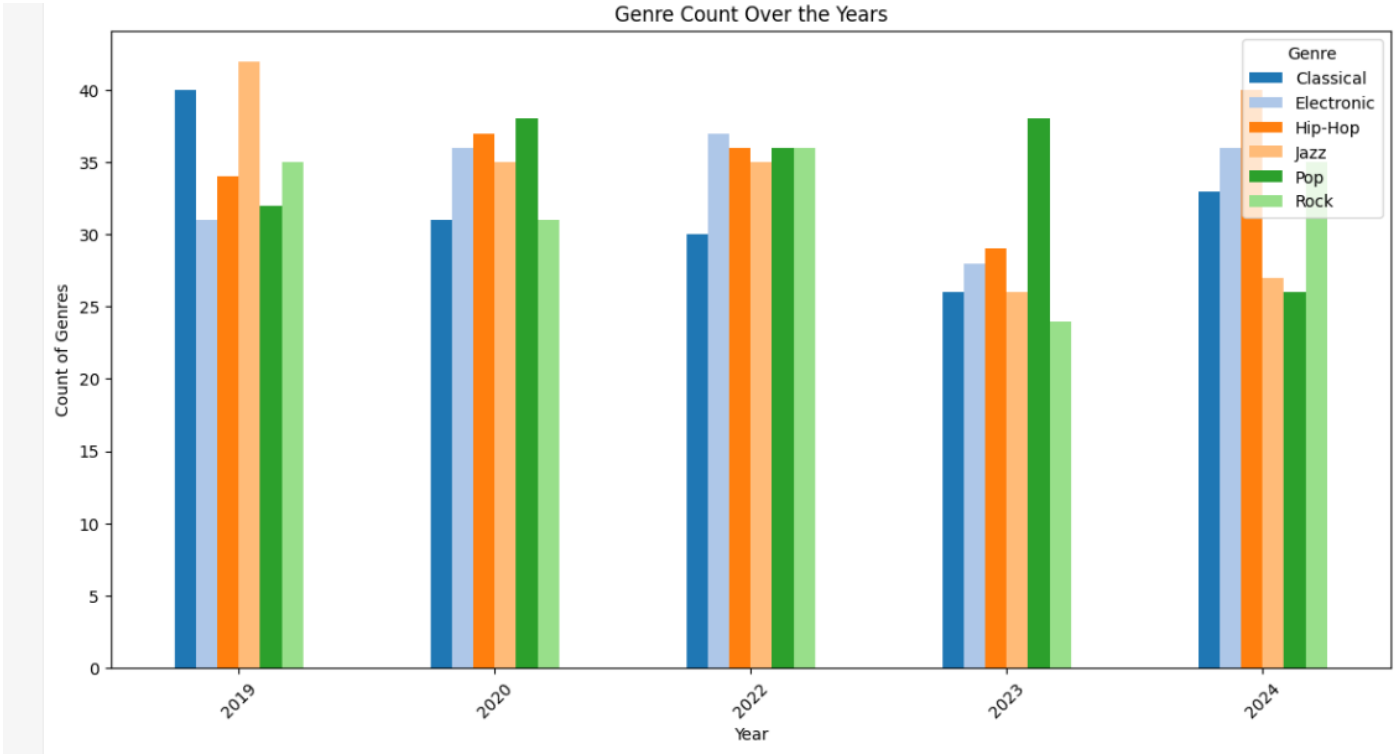
- Check for missing values in each table.
- Ensure data types are appropriate for each column.
- Handle any outliers or inconsistencies.

Exploratory Data Analysis (EDA) and Visualization:

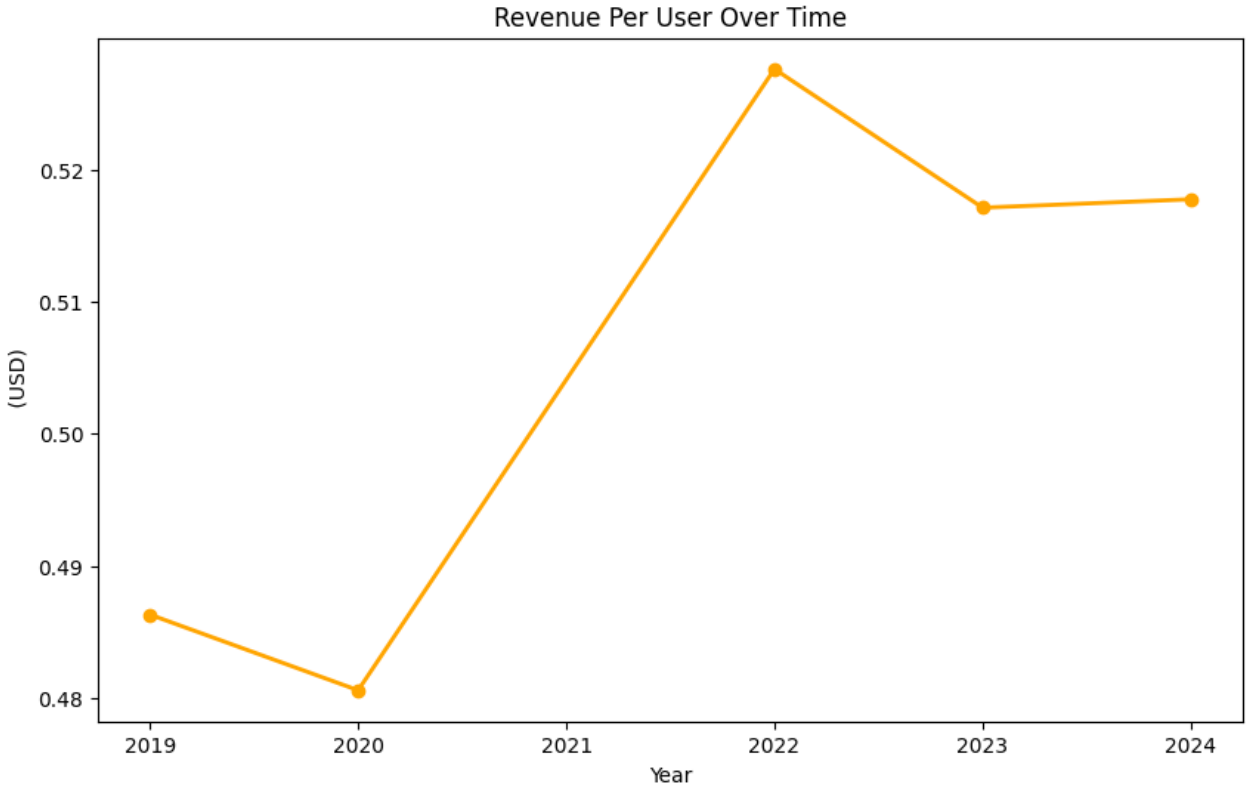
1. Analyzing the trend in Monthly Active Users Across Platforms



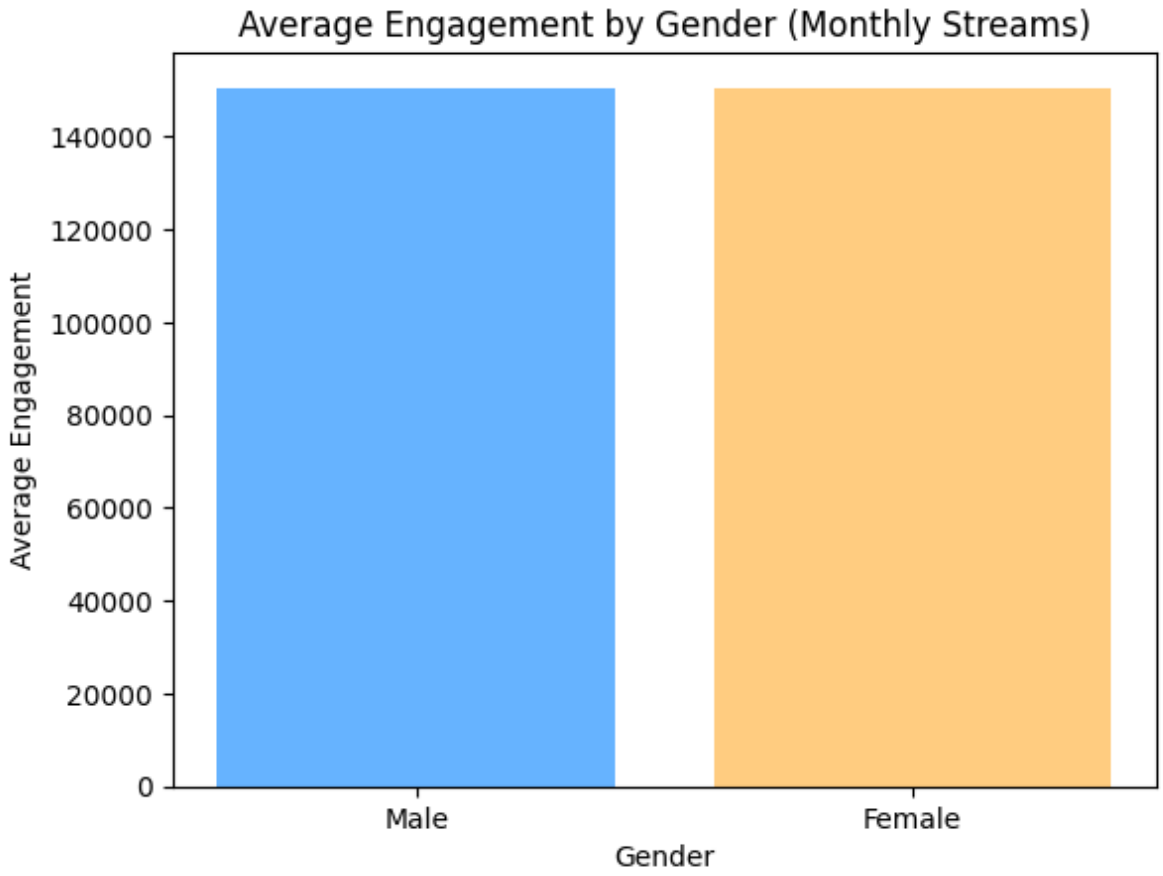
2. Analyzing the genre with highest engagement over the year



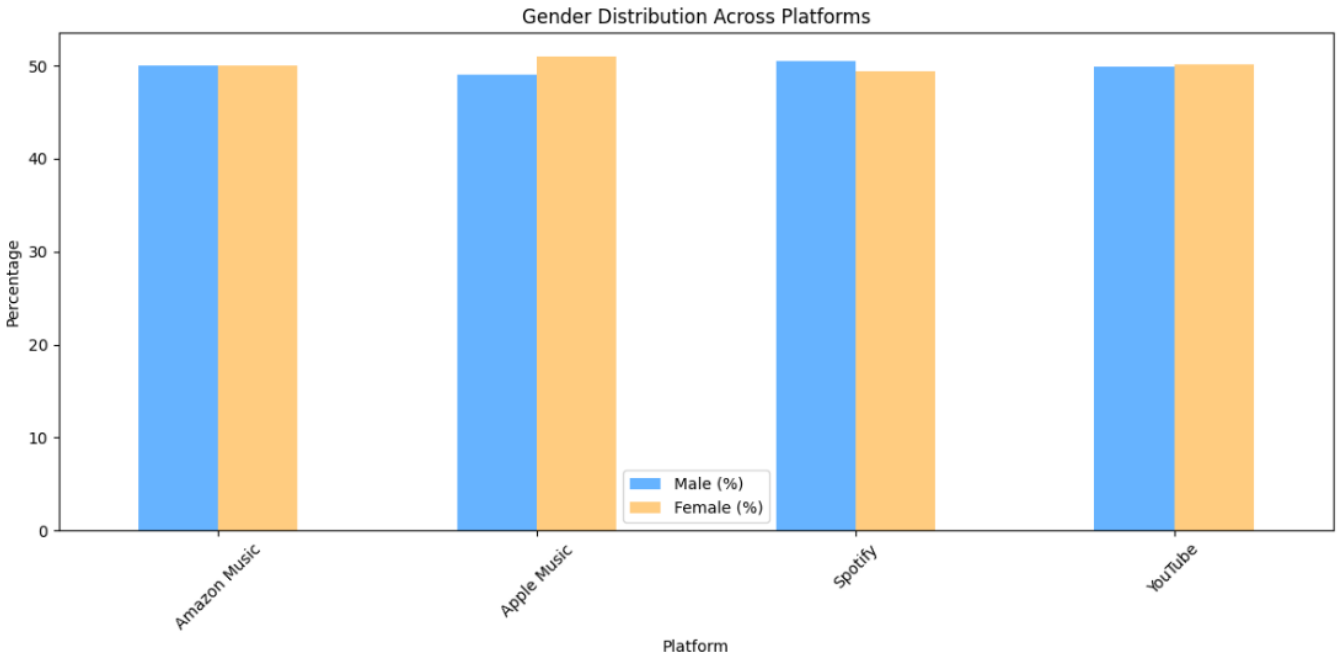
3. Analyzing the revenue per user over t



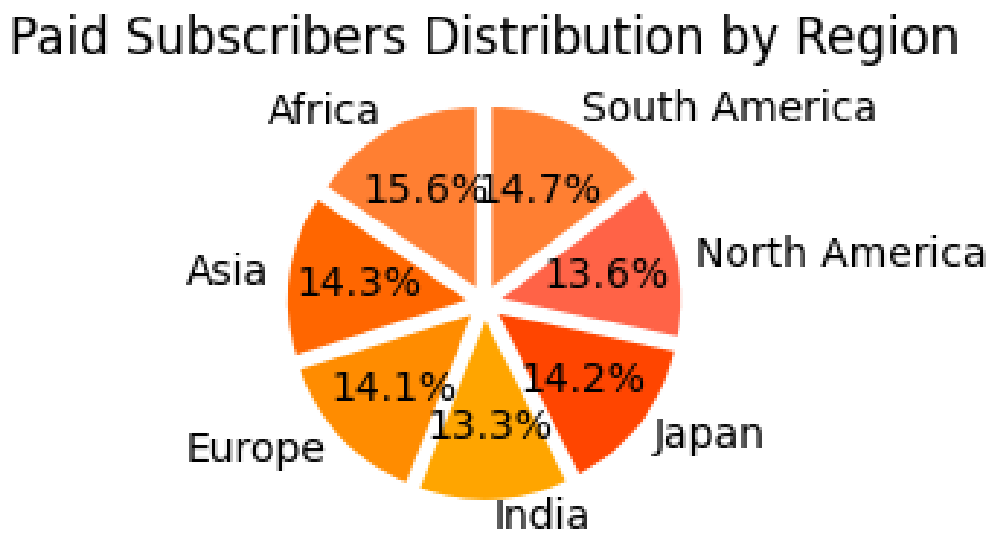
4. Analyzing average engagement based on gender



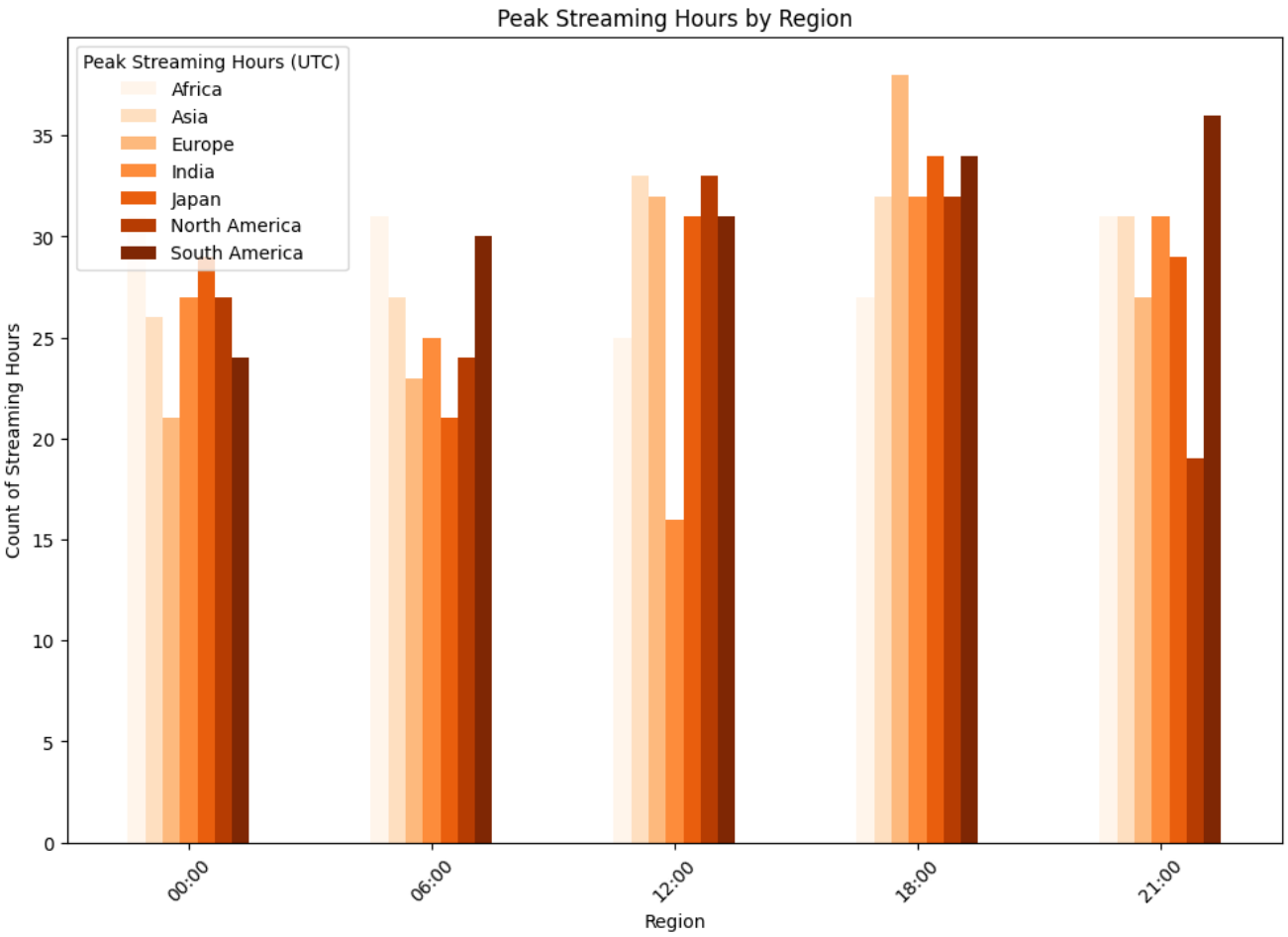
5. Gender Distribution Across Platforms



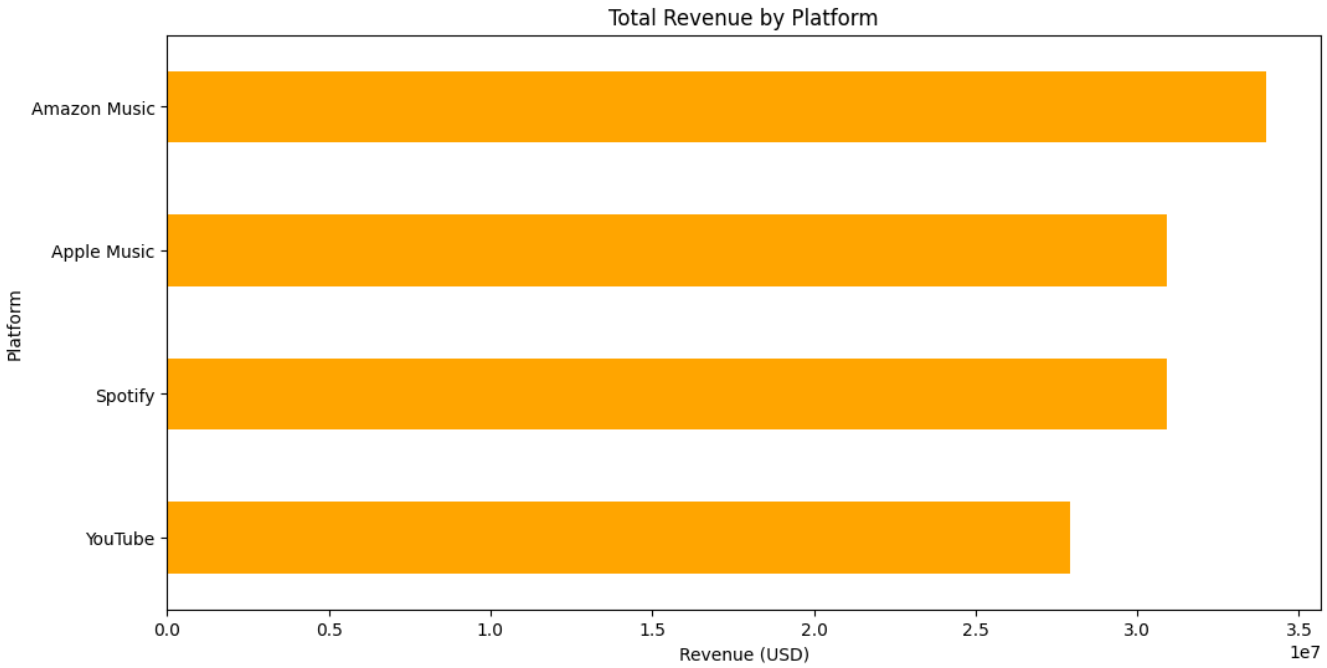
6. Analyzing the average Paid Subscribers percentage



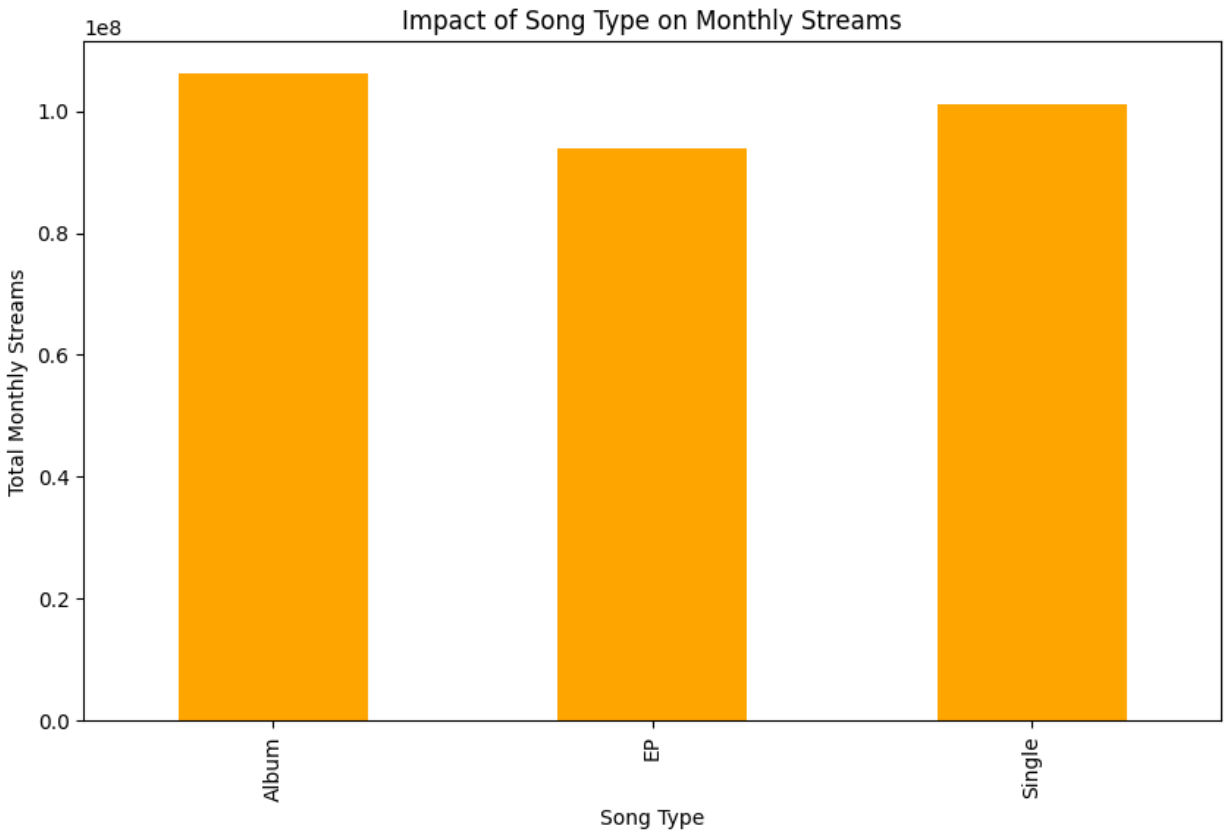
7. Finding Peak Streaming Hours and count occurrences



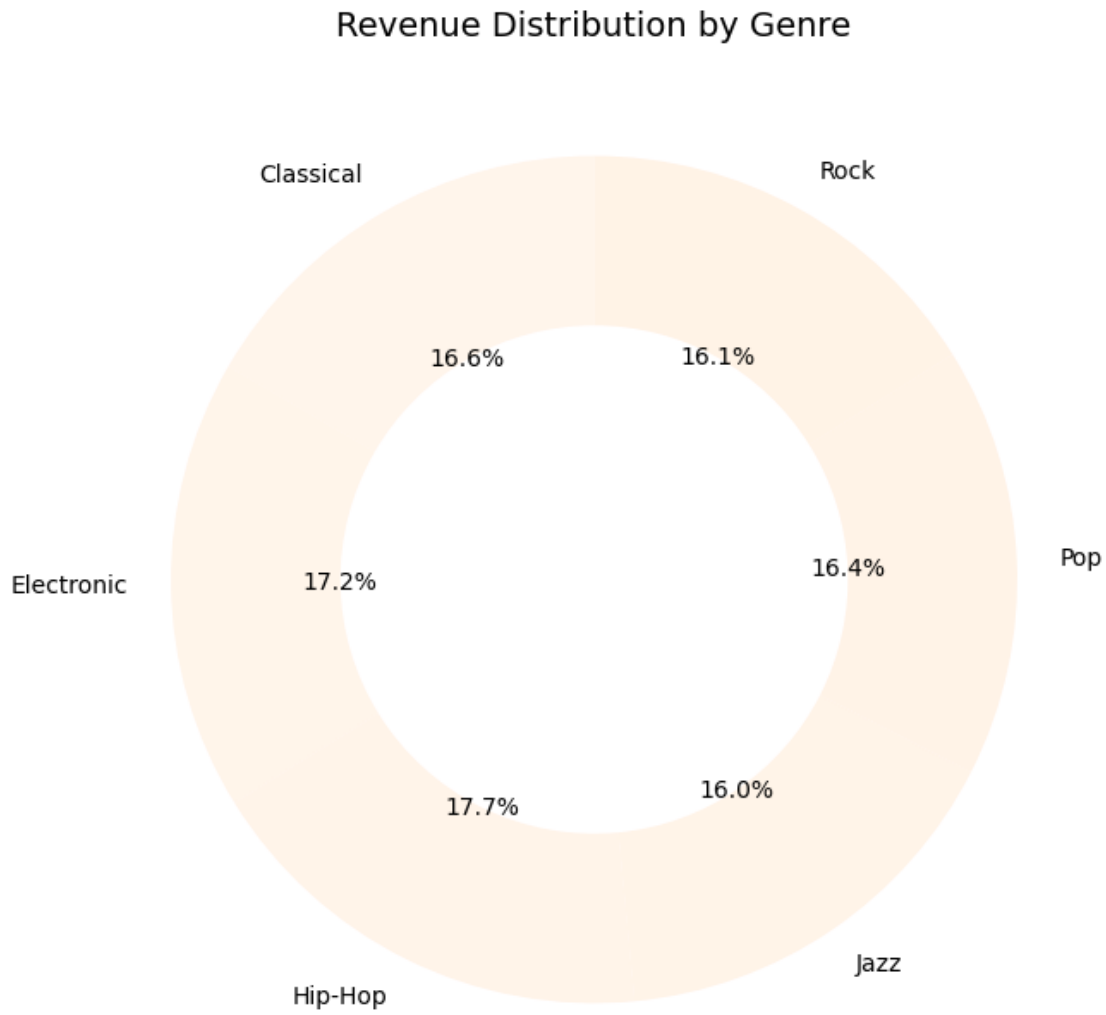
8. platform vs revenue



9. Impact of Song Type on Streams



10. Genre wise revenue



Conclusion:

Upon analysis, I observed several key trends in the dataset. First, there is a consistent and loyal user base for music streaming platforms, with stable engagement over time. Notably, there has been a marked rise in Hip-Hop culture in 2024, indicating a shift in genre preferences. Additionally, the period between 2020 and 2022 saw an increase in revenue per user, suggesting improved monetization strategies or premium subscriptions. Furthermore, the analysis revealed that engagement is nearly identical between male and female users, demonstrating a balanced user interaction across genders. These insights offer valuable perspectives for platform growth, content strategy, and user engagement.