**Spotify Data Analysis and Visualization using Power BI and Python**

**1. Introduction**

In this project, we perform an in-depth analysis of Spotify music trends using a combination of Python (Pandas, Spotipy) and Power BI. The objective is to extract, clean, and transform Spotify track data, enhancing it with album image URLs using the Spotify Developer API. The processed dataset is then utilized to build an interactive Power BI dashboard, which enables us to analyze various aspects of song popularity, artist dominance, music genres, and acoustic characteristics. This report covers the complete workflow, from data preprocessing and transformation to data modeling, DAX calculations, and visualization.

**2. Data Preprocessing using Python**

Before importing the dataset into Power BI, Python was used for preprocessing. The Spotipy library, along with the Spotify Developer API, was leveraged to fetch album cover URLs for each track, allowing us to enhance the dataset visually in Power BI.

**2.1 Steps Performed in Python**

1. Loaded the dataset using Pandas to inspect its structure and identify necessary transformations.
2. Authenticated with the Spotify API using client credentials (Client ID and Secret Key) to access track metadata.
3. Extracted album cover URLs by querying the API using track\_name and artist\_name.
4. Added a new column (track\_album\_url) containing the retrieved image URLs.
5. Saved the updated dataset as a CSV file, which was then imported into Power BI for further transformation and analysis.

**PYTHON CODE:**

import pandas as pd

import spotipy

from spotipy.oauth2 import SpotifyClientCredentials

# Authenticate with Spotify API

client\_id = "your\_client\_id"

client\_secret = "your\_client\_secret"

sp = spotipy.Spotify(auth\_manager=SpotifyClientCredentials(client\_id, client\_secret))

# Load dataset

df = pd.read\_csv("spotify\_data.csv")

# Function to fetch album cover URL

def get\_album\_url(track\_name, artist\_name):

results = sp.search(q=f"track:{track\_name} artist:{artist\_name}", limit=1)

if results['tracks']['items']:

return results['tracks']['items'][0]['album']['images'][0]['url']

return None

# Apply function to dataset

df["track\_album\_url"] = df.apply(lambda row: get\_album\_url(row["track\_name"], row["artist\_name"]), axis=1)

# Save updated dataset

df.to\_csv("spotify\_data\_updated.csv", index=False)

This enhanced dataset was then loaded into Power BI for further transformation and visualization.

**3. Data Transformation and Cleaning using Power Query**

Once the dataset was enriched with album cover URLs, Power BI’s Power Query Editor was used to refine the data further. The following transformations were performed:

* Removed duplicate records to ensure data integrity.
* Handled missing values by filling or removing incomplete records.
* Converted data types (e.g., converting the popularity column to numerical values and date columns to proper DateTime format).
* Created new calculated columns to classify songs based on their musical key and mode.
* Filtered out irrelevant data to focus on meaningful insights.

These transformations ensured that the dataset was optimized for analysis and visualization.

**4. Data Modeling and Relationships**

To establish efficient querying and filtering, the dataset was structured into four main tables in Power BI:

* Tracks Table: Contains details about each song, including its popularity, tempo, and album cover.
* Artists Table: Stores information about different artists and their popularity.
* Albums Table: Includes album-related metadata such as release date and total track count.
* Genres Table: Categorizes tracks into different music genres for genre-based analysis.

**4.1 Establishing Relationships**

Relationships were created between these tables based on unique identifiers like artist\_id and album\_id. This allowed seamless cross-filtering and dynamic report interactions.

**5. DAX Measures for Analysis**

To extract meaningful insights, DAX (Data Analysis Expressions) measures were created:

* Total Streams: Calculates the total number of streams for all tracks.

Total Streams = SUM(SpotifyData[streams])

* Average Popularity: Computes the average popularity score of all songs in the dataset.

Avg Popularity = AVERAGE(SpotifyData[popularity])

* Major vs. Minor Key Classification: Creates a new column that combines the song key and mode.

Key Mode = CONCATENATE(SpotifyData[key], " ", SpotifyData[mode])

These calculations were essential in understanding song trends and popularity factors.

**6. Dashboard and Visualization Design**

An interactive Power BI dashboard was built, incorporating dynamic visuals and slicers to allow users to explore data effortlessly. The dashboard includes:

* Top Artists and Tracks (Bar Chart): Displays the most popular artists and songs based on their stream count and popularity.
* Streaming Trends Over Time (Line Chart): Shows how song popularity fluctuates over different time periods.
* Genre Distribution (Treemap): Categorizes songs by genre, giving an overview of genre dominance.
* Acoustic vs. Electronic Songs (Donut Chart): Segments tracks into acoustic (>50% acousticness) and electronic (<50% acousticness).
* Music Feature Impact on Popularity (Waterfall Chart): Analyzes how various musical features contribute to song popularity.
* Interactive Filters & Slicers: Enables users to explore data dynamically by filtering songs by artist, genre, or popularity.

Additionally, album image URLs were integrated into the report visuals to enhance interactivity and engagement.

**7. Key Insights and Conclusion**

Through this project, several key insights were discovered:

* Pop and Hip-Hop dominate the streaming charts, reflecting listener preferences.
* Danceability and energy are major contributors to song popularity, particularly in electronic and pop music.
* Some artists consistently outperform others, with a few dominating the top spots in terms of streams and engagement.
* Streaming activity fluctuates over time, influenced by album releases, viral trends, and seasonal changes.

Future Improvements

Future enhancements could include:

* Real-time API Integration: Connecting Power BI to the Spotify API for live data updates.
* Lyrical Sentiment Analysis: Performing NLP (Natural Language Processing) on song lyrics to identify mood and emotional tone.
* User Listening Habits Analysis: Incorporating user-specific listening history to generate personalized insights.

This project demonstrates the power of Python and Power BI in combining data transformation, analytics, and visualization to derive actionable insights from large datasets.