Table of Contents

ETL Process and Data Analysis for Spotify Data using AWS Services

Abstract

In the digital era, music streaming platforms like Spotify generate vast amounts of data on user preferences, artist popularity, and music trends. Analyzing this data can offer invaluable insights for the music industry, enabling data-driven strategies for artists, labels, and marketers. This project documents the development and deployment of a robust ETL (Extract, Transform, Load) pipeline for automating the ingestion, transformation, and storage of Spotify's music metadata using Amazon Web Services (AWS). The pipeline integrates AWS Lambda for automated extraction of data from Spotify's API, with Amazon CloudWatch triggering Lambda functions every minute to ensure real-time updates. The raw JSON data is transformed into structured CSV format using Python scripts in AWS Lambda, improving data accessibility and readability for analytical purposes. The transformed data is stored in Amazon S3 and cataloged using AWS Glue, allowing seamless querying in Amazon Athena, which enables SQL-based analytics on the structured data.

To enhance accessibility and provide a user-friendly exploration tool, an interactive Streamlit dashboard was developed. This dashboard visualizes key insights, including album release trends, distribution of song popularity, artist productivity, and song duration analysis. Through intuitive charts and graphs, stakeholders can analyze real-time music data, uncover trends, and gain actionable insights into the characteristics of popular music and artist engagement on Spotify. Key findings include a steady increase in album releases over time, a high concentration of popular songs within the dataset, and a lack of correlation between song duration and popularity. The Streamlit deployment of the dashboard allows for dynamic exploration of the dataset, empowering users to make informed decisions based on data trends.

This project demonstrates the capabilities of serverless cloud-based architectures and interactive visualization tools in handling large-scale, continuously updating data. The combination of AWS's scalable services and Streamlit's interactive visualization makes this solution cost-effective, efficient, and adaptable. Future extensions could include expanding the data source to incorporate additional streaming platforms, conducting sentiment analysis on song lyrics, or adding user engagement metrics for deeper insights. The project highlights how advanced data engineering and visualization can transform raw music metadata into meaningful insights, supporting data-driven decisions in the music industry.

Keywords: ETL pipeline, AWS Lambda, Amazon S3, Amazon Glue, Amazon Athena, Spotify API, Streamlit dashboard, music metadata, data-driven insights, serverless architecture, data visualization.

Introduction

In the modern digital landscape, music streaming platforms like Spotify generate vast amounts of data, ranging from song metadata to listener interactions. Analyzing this data can provide insights into music trends, artist popularity, and listener preferences. However, handling such high-volume, real-time data requires an efficient ETL (Extract, Transform, Load) pipeline that can ingest, process, and store data for analysis.

This project documents the design and deployment of an ETL pipeline for Spotify's music metadata using Amazon Web Services (AWS). By utilizing AWS's serverless services, such as Lambda, S3, Glue, and Athena, we achieved a cost-effective and scalable solution for continuously updating and analyzing music data. Additionally, we developed an interactive Streamlit dashboard to visualize insights, providing stakeholders with a user-friendly interface for data exploration and analysis. This report covers the objectives, methodology, analysis, and results of the project, showcasing how cloud-based ETL and visualization solutions can enhance data-driven decision-making in the music industry.

Project Profile

a. Objectives

The primary goals of this project are:

1. **Automate Data Collection**: Use AWS Lambda and CloudWatch to schedule and automate data extraction from Spotify's API, reducing manual intervention and ensuring up-to-date data.
2. **Efficient Data Transformation**: Convert complex JSON data from Spotify's API into a structured CSV format to facilitate data analysis and improve query performance.

3. **Scalable Storage and Querying**: Store structured data in Amazon S3 and catalog it in AWS Glue Data Catalog to make it accessible for SQL-based querying in Amazon Athena.
4. **User-Friendly Visualization**: Develop an interactive dashboard with Streamlit to display key insights, enabling users to explore data trends intuitively and gain actionable insights.

b. Dataset

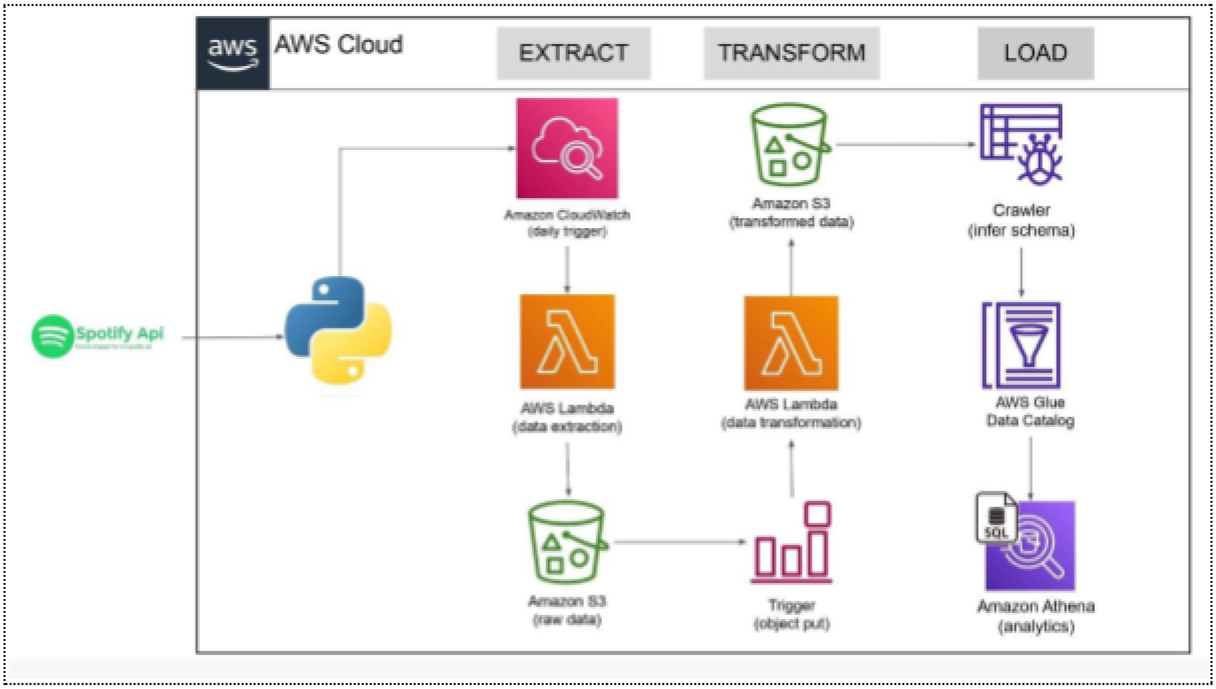
The dataset is sourced from Spotify's API and contains metadata on songs, albums, and artists. Key attributes include:

- **Albums**: Each album record contains album ID, name, release date, total tracks, and a URL to the Spotify page. This information enables analysis of release trends and album characteristics.
- **Artists**: Artist records include artist ID, name, and URL. This information is useful for studying artist popularity and productivity.
- **Songs**: Song records consist of song ID, name, duration (in milliseconds), popularity score, and associations with albums and artists. The song popularity score is a significant metric for understanding listener preferences.

The data is initially in JSON format, with a nested structure that represents relationships between albums, artists, and songs. Transforming this JSON data into CSV format allows for efficient analysis and querying, making it suitable for the analytical tools used in this project.

c. Methodology

Our ETL pipeline follows a structured methodology that leverages AWS's serverless architecture for efficiency and scalability:

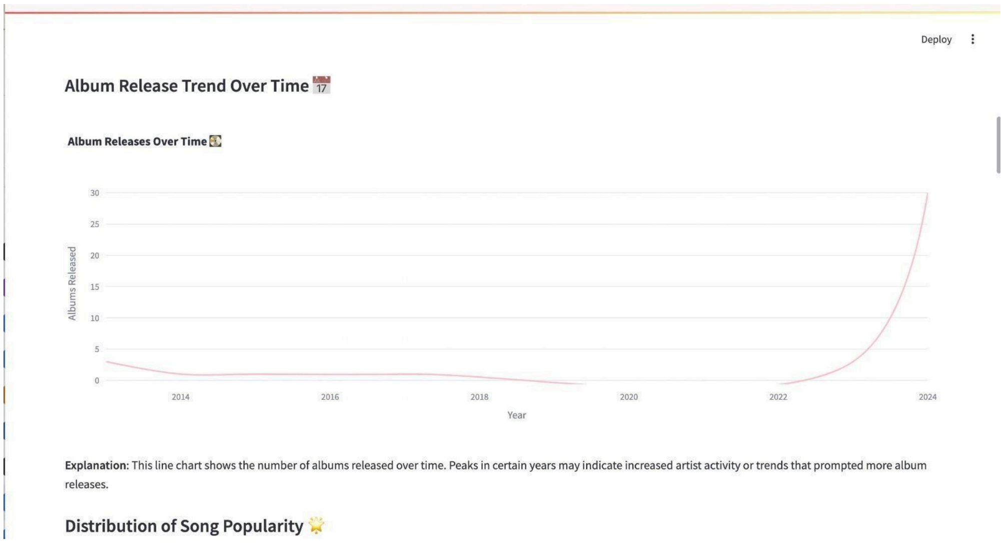


1. **Extraction**:
   - **Data Source**: Data is extracted from the Spotify API. We configured AWS Lambda functions triggered by Amazon CloudWatch to pull data every minute, ensuring near real-time data collection.
   - **Process**: The Lambda function makes API calls to Spotify, retrieves JSON data, and stores it in the “raw_data/to_processed” folder on Amazon S3. This automation reduces manual effort, allowing the system to operate continuously without intervention.
2. **Transformation**:
   - **Objective**: The JSON data format from Spotify’s API, while ideal for hierarchical storage, is not optimized for analysis. To address this, a second Lambda function processes and transforms the JSON data into CSV format.
   - **Process**: The transformation Lambda function reads raw data from the S3 “to_processed” folder, parses it using Python libraries like Pandas, and separates the data into structured tables for albums, artists, and songs. The transformed data is then saved in the “transformed_data” folder in S3, categorized by type.
   - **Benefits**: This transformation enables easier querying in Athena and improves data accessibility for analysis.
3. **Load**:
   - **Cataloging**: AWS Glue Crawler scans the transformed data, inferring its schema and adding it to the AWS Glue Data Catalog. This cataloging process organizes the data, making it accessible to Amazon Athena.
   - **Querying**: By leveraging Athena’s SQL interface, users can perform complex queries on the transformed data. This setup allows for easy and efficient data exploration, facilitating advanced analytics on Spotify’s music metadata.
4. **Data Visualization and Dashboard Deployment**:
   - **Streamlit Dashboard**: We created a user-friendly dashboard using Streamlit (coded in app2.py) to display key insights from the data. The Streamlit app is deployed to provide an interactive experience, allowing users to explore trends, artist popularity, and song characteristics visually.
   - **Visualization Elements**: The dashboard includes charts and graphs derived from the transformed data, allowing users to filter and interpret the data easily. This interactivity enhances accessibility for stakeholders, enabling data-driven insights.

Observations and Analysis

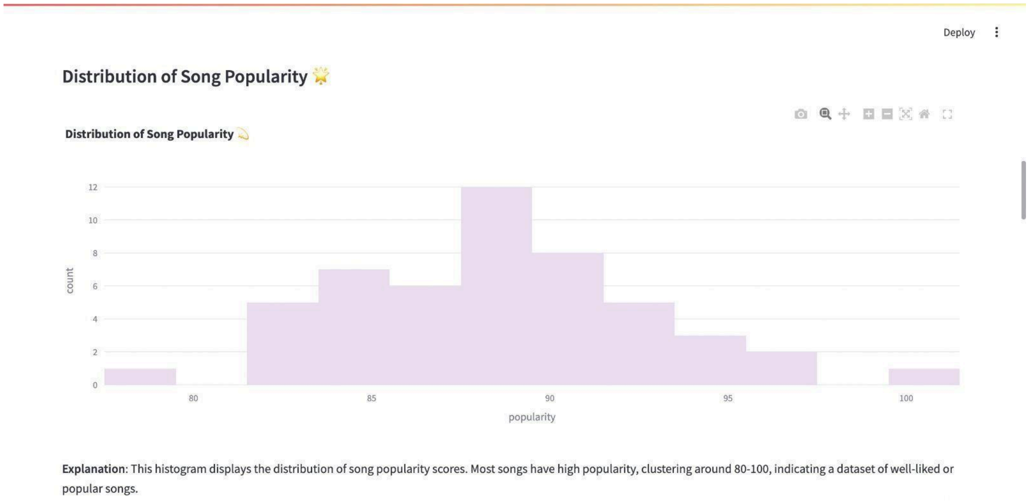
The analysis phase involved deriving insights from the transformed Spotify dataset, visualized through graphs on the Streamlit dashboard. Here are the main observations:

- **Automated, Real-Time Data Collection**: Our setup with CloudWatch and Lambda allows for continuous data extraction from Spotify's API every minute. This setup is essential for applications that require real-time data analysis, enabling us to capture recent trends in the music industry.
- **Transformation Process and Data Structure**: The transformation process effectively organizes the data into CSV files for albums, artists, and songs. This structured format is suitable for SQL queries, allowing users to drill down into specific aspects of the data, such as song popularity and artist productivity.
- **Interactive Dashboard Insights**:
    - **Album Release Trend Over Time**: This line chart on the dashboard shows a rising trend in album releases over recent years, indicating a surge in music production and distribution. This trend may reflect industry changes, such as increased accessibility to streaming platforms.


Deploy
⋮
Album Release Trend Over Time 📅
Album Releases Over Time 💿
30
25
20
15
10
5
0
Albums Released
2014
2016
2018
2020
2022
2024
Year
Distribution of Song Popularity 🌟

Explanation: This line chart shows the number of albums released over time. Peaks in certain years may indicate increased artist activity or trends that prompted more album releases.

- ○ **Distribution of Song Popularity**: The histogram shows that most songs have high popularity scores, clustering around 80-100. This indicates a dataset focused on well-liked songs, providing insights into general music preferences.

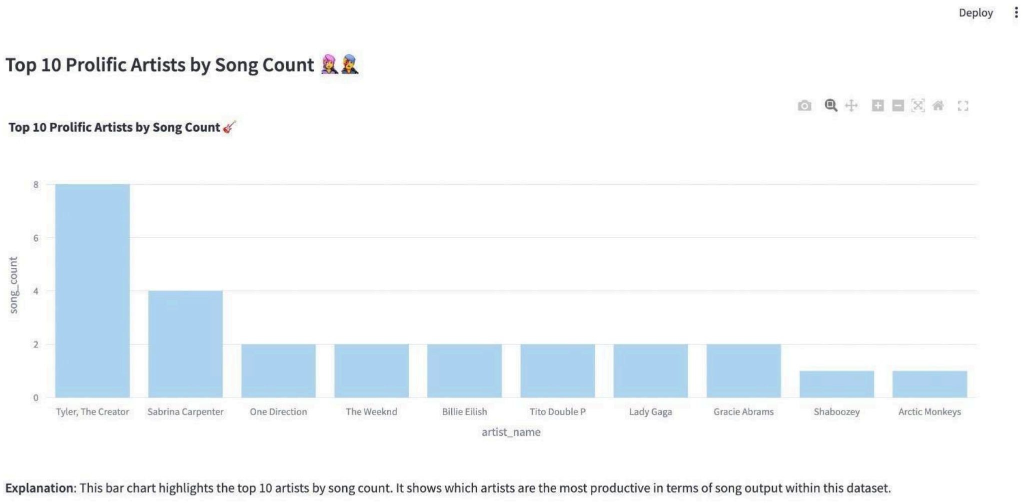


Distribution of Song Popularity 🌟

Distribution of Song Popularity 💫

Explanation: This histogram displays the distribution of song popularity scores. Most songs have high popularity, clustering around 80-100, indicating a dataset of well-liked or popular songs.

- ○ **Top 10 Prolific Artists by Song Count**: This bar chart highlights the most productive artists in the dataset, with Tyler, The Creator, leading in terms of song output. This metric can help stakeholders identify prolific artists.

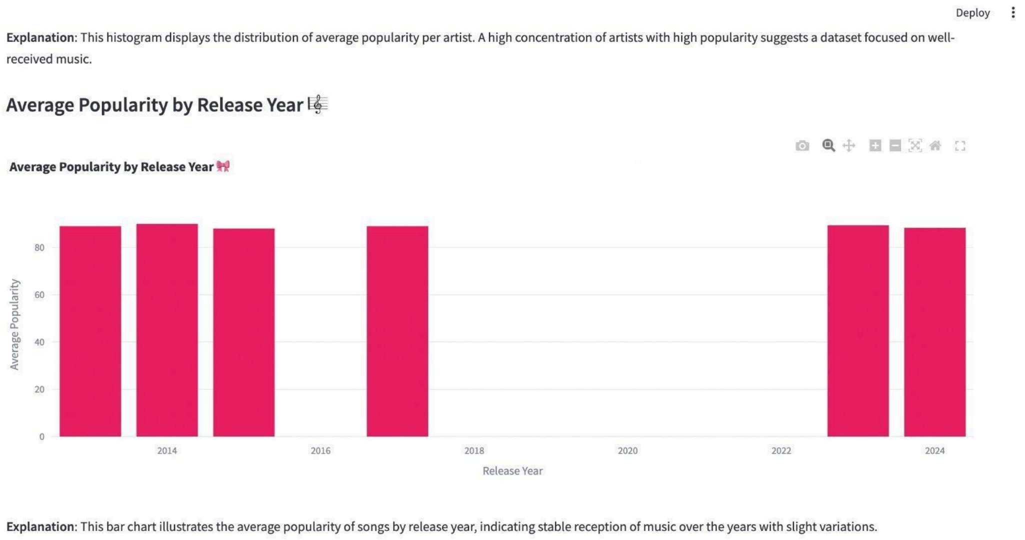


Top 10 Prolific Artists by Song Count 🧑‍🎤🧑‍🎤

Top 10 Prolific Artists by Song Count 🎸

Explanation: This bar chart highlights the top 10 artists by song count. It shows which artists are the most productive in terms of song output within this dataset.

- ○ **Average Popularity by Release Year**: This chart shows stable average popularity across release years, suggesting consistent listener engagement and music quality over time.



Explanation: This histogram displays the distribution of average popularity per artist. A high concentration of artists with high popularity suggests a dataset focused on well-received music.

Average Popularity by Release Year 🎼

Explanation: This bar chart illustrates the average popularity of songs by release year, indicating stable reception of music over the years with slight variations.

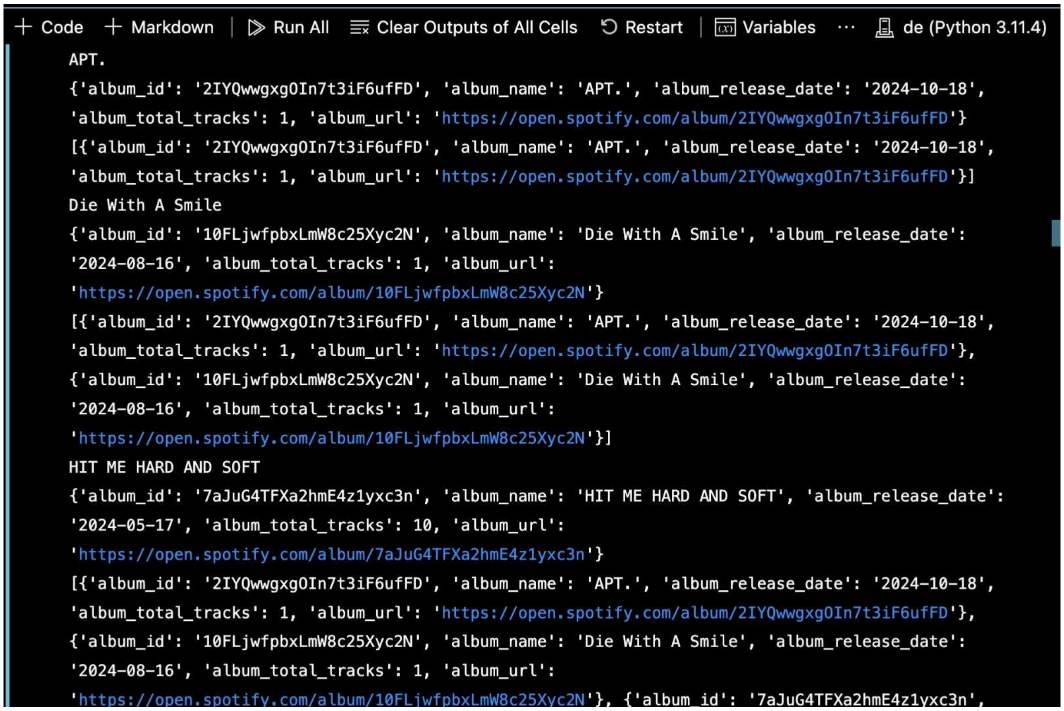
These observations underscore the effectiveness of the ETL pipeline and the value of visualization for data interpretation, enabling stakeholders to make informed decisions based on music trends.

Results

The ETL pipeline and visualization components achieved the following results:

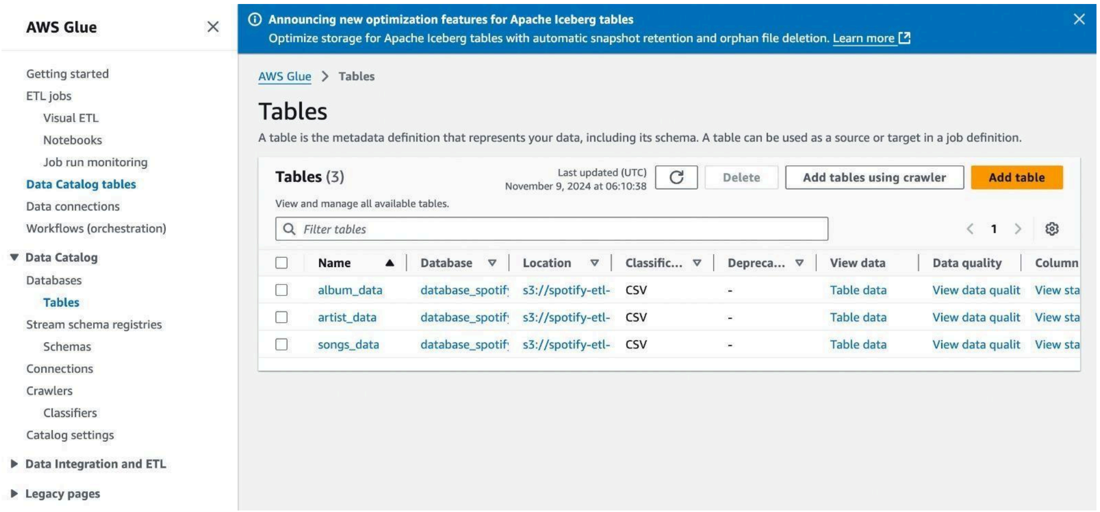
- **Organized, Accessible Data**: By transforming JSON data to CSV format and cataloging it in AWS Glue, we created a structured, easy-to-query dataset. This organization enhances the dataset's usability for analytical purposes.
- **Real-Time Data Updates**: The pipeline's one-minute update interval ensures that the data remains current, providing accurate, timely insights into Spotify's dynamic music database. This feature is essential for trend analysis and decision-making in real-time applications.
- **Comprehensive Insights through Streamlit Dashboard**: The Streamlit dashboard (using app2.py) provides a clear, accessible interface for exploring the Spotify data. The dashboard displays key metrics, such as the total albums, average tracks per album, unique artists, total songs, average song duration, and average popularity score, providing users with an overview of the dataset.

This deployment of the Streamlit dashboard not only makes the data accessible but also enhances the user experience, enabling stakeholders to explore and interpret the data without technical expertise.



```
APT.
{'album_id': '2IYQwwgxgOIn7t3iF6ufFD', 'album_name': 'APT.', 'album_release_date': '2024-10-18',
'album_total_tracks': 1, 'album_url': 'https://open.spotify.com/album/2IYQwwgxgOIn7t3iF6ufFD'}
[{'album_id': '2IYQwwgxgOIn7t3iF6ufFD', 'album_name': 'APT.', 'album_release_date': '2024-10-18',
'album_total_tracks': 1, 'album_url': 'https://open.spotify.com/album/2IYQwwgxgOIn7t3iF6ufFD'}]
Die With A Smile
{'album_id': '10FLjwfpbxLmW8c25Xyc2N', 'album_name': 'Die With A Smile', 'album_release_date':
'2024-08-16', 'album_total_tracks': 1, 'album_url':
'https://open.spotify.com/album/10FLjwfpbxLmW8c25Xyc2N'}
[{'album_id': '2IYQwwgxgOIn7t3iF6ufFD', 'album_name': 'APT.', 'album_release_date': '2024-10-18',
'album_total_tracks': 1, 'album_url': 'https://open.spotify.com/album/2IYQwwgxgOIn7t3iF6ufFD'},
{'album_id': '10FLjwfpbxLmW8c25Xyc2N', 'album_name': 'Die With A Smile', 'album_release_date':
'2024-08-16', 'album_total_tracks': 1, 'album_url':
'https://open.spotify.com/album/10FLjwfpbxLmW8c25Xyc2N'}]
HIT ME HARD AND SOFT
{'album_id': '7aJuG4TFXa2hmE4z1yxc3n', 'album_name': 'HIT ME HARD AND SOFT', 'album_release_date':
'2024-05-17', 'album_total_tracks': 10, 'album_url':
'https://open.spotify.com/album/7aJuG4TFXa2hmE4z1yxc3n'}
[{'album_id': '2IYQwwgxgOIn7t3iF6ufFD', 'album_name': 'APT.', 'album_release_date': '2024-10-18',
'album_total_tracks': 1, 'album_url': 'https://open.spotify.com/album/2IYQwwgxgOIn7t3iF6ufFD'},
{'album_id': '10FLjwfpbxLmW8c25Xyc2N', 'album_name': 'Die With A Smile', 'album_release_date':
'2024-08-16', 'album_total_tracks': 1, 'album_url':
'https://open.spotify.com/album/10FLjwfpbxLmW8c25Xyc2N'}, {'album_id': '7aJuG4TFXa2hmE4z1yxc3n',
```

Raw JSON data



Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more

AWS Glue > Tables

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Tables (3)

Last updated (UTC)
November 9, 2024 at 06:10:38

View and manage all available tables.

Filter tables

□	Name	Database	Location	Classific...	Depreca...	View data	Data quality	Column
□	album_data	database_spotif	s3://spotify-etl-	CSV	-	Table data	View data qualit	View sta
□	artist_data	database_spotif	s3://spotify-etl-	CSV	-	Table data	View data qualit	View sta
□	songs_data	database_spotif	s3://spotify-etl-	CSV	-	Table data	View data qualit	View sta

JSON data converted to CSV format

Visualizations and Key Insights

Here are the main visualizations and insights from the Streamlit dashboard:

1. **Album Release Trend Over Time**: This line chart shows album releases by year, with a noticeable increase in recent years, indicating a growing trend in music production and accessibility.
2. **Distribution of Song Popularity**: The popularity distribution histogram reveals that most songs are widely liked, with scores clustering around 80-100, suggesting a high overall quality of the dataset.
3. **Song Duration vs. Popularity**: The scatter plot shows a near-zero correlation between song duration and popularity, indicating that song length does not significantly impact its reception.
4. **Top 10 Prolific Artists by Song Count**: This bar chart identifies the most productive artists, with Tyler, The Creator, as a standout. This insight is useful for understanding prolific contributions within the music industry.
5. **Top 10 Albums by Average Popularity**: The chart displays albums with the highest average popularity scores, reflecting consistent listener engagement and preference.
6. **Song Duration Distribution**: This histogram shows that most songs are between 2 to 4 minutes long, aligning with typical commercial music norms.
7. **Monthly Trend of Songs Added**: This bar chart highlights potential seasonal patterns in song releases, which could be useful for scheduling marketing or release plans.
8. **Distribution of Average Artist Popularity**: This histogram displays the distribution of average artist popularity, showing a high concentration of popular artists in the dataset.
9. **Average Popularity by Release Year**: The bar chart displays consistent popularity over time, indicating stable listener engagement and music quality.

Conclusion

The ETL pipeline for Spotify data using AWS and Streamlit demonstrated an effective way to handle large-scale data processing, real-time updates, and interactive visualization. By automating data extraction, transformation, and storage, we created a robust solution for analyzing Spotify's music metadata. The Streamlit dashboard provided a user-friendly platform for stakeholders to explore data-driven insights, allowing for easy analysis of music trends and artist productivity.

This project underscores the power of cloud-based ETL solutions and the value of interactive data visualization. Future enhancements could include expanding the data source to cover multiple music platforms, incorporating sentiment analysis on song lyrics, or adding listener engagement metrics. This pipeline showcases how serverless architectures and visualization tools like Streamlit can empower data-driven decision-making in the music industry, offering new opportunities for strategic planning and audience understanding.