SPOTIFY DATA ENGINEERING & ANALYTICS PIPELINE

By Vaishnavi Chinnala

Problem Statement

In the digital music landscape, Spotify generates massive volumes of data every minute across regions, genres, and user preferences. However, deriving actionable insights from such raw and semi-structured data is challenging without a robust pipeline. This project aims to build a complete ELT (Extract, Load, Transform) pipeline that fetches and processes Spotify data for meaningful analytics.

We focused on enabling music analysts to better understand patterns such as artist performance, track virality, listener behavior, and preferences across age groups. The processed data is visualized using Power BI dashboards to facilitate business and marketing decisions.

Key Focus Areas

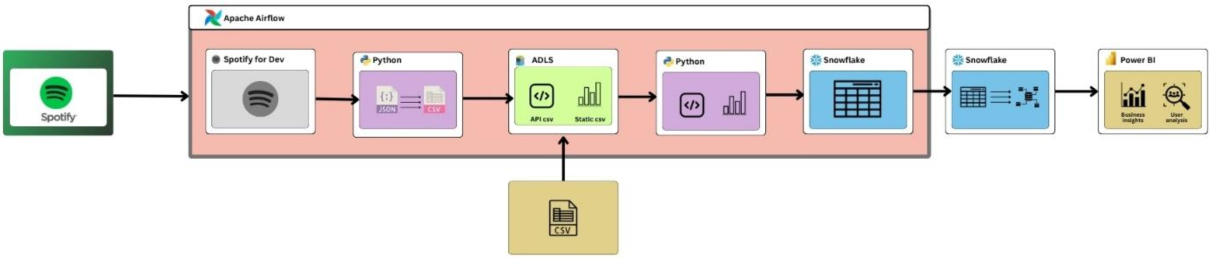
- **Automated Data Pipeline:** Leveraging Apache Airflow to orchestrate extraction, transformation, and loading processes.
- **Data Enrichment:** Applying techniques such as language detection and stream simulation to augment raw Spotify data.
- **Scalable Storage & Processing:** Using Azure Blob Storage for staging and Snowflake for structured storage and processing.
- **Dimensional Modeling:** Building normalized data models with star schema in Silver and Gold layers.
- **Interactive Visualization:** Connecting processed data to Power BI for detailed analytics dashboards.

Software Specifications

- Programming Language: Python 3.10
- Orchestration: Apache Airflow
- Cloud Storage: Azure Blob Storage (ADLS Gen2)

- Data Warehouse: Snowflake with Snowpark
- Visualization: Power BI Desktop
- Version Control: Azure DevOps
- Additional Libraries: Pandas, Requests, LangDetect, JSON

Architecture



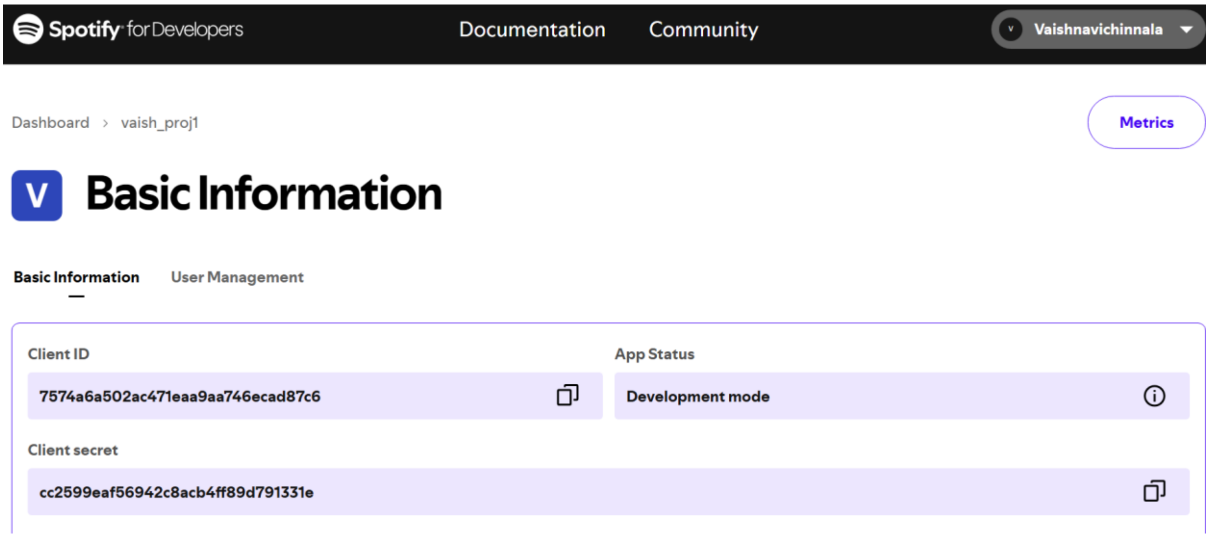
Flow Overview

1. Extract Spotify metadata via REST API
2. Clean and enrich with language detection and random stream generation
3. Store intermediate CSV in Azure Blob
4. Load into Snowflake using COPY commands
5. Transform using SQL scripts to build dimension and fact models
6. Visualize using Power BI linked to Snowflake

Project Implementation Steps

Step 1: API Integration

- Used Client Credentials flow to access Spotify API.
- Collected up to 1000 records using paginated API calls.



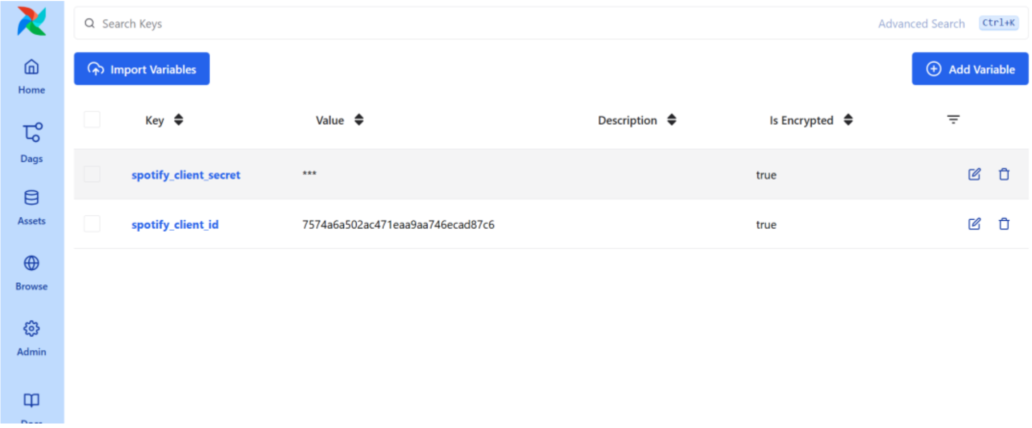
Client ID: 7574a6a502ac471eaa9aa746ecad87c6

App Status: Development mode

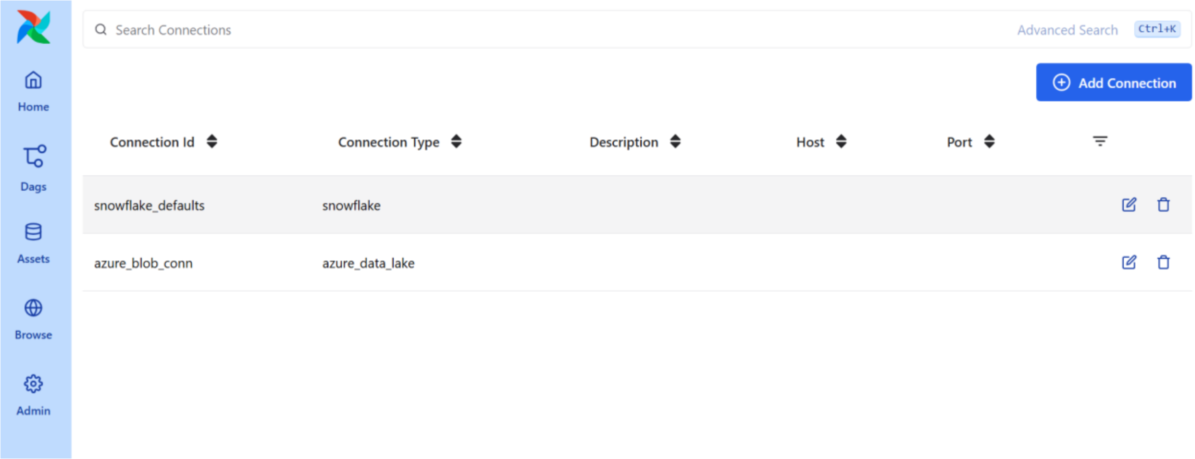
Client secret: cc2599eaf56942c8acb4ff89d791331e

Step 2: Airflow DAG Design

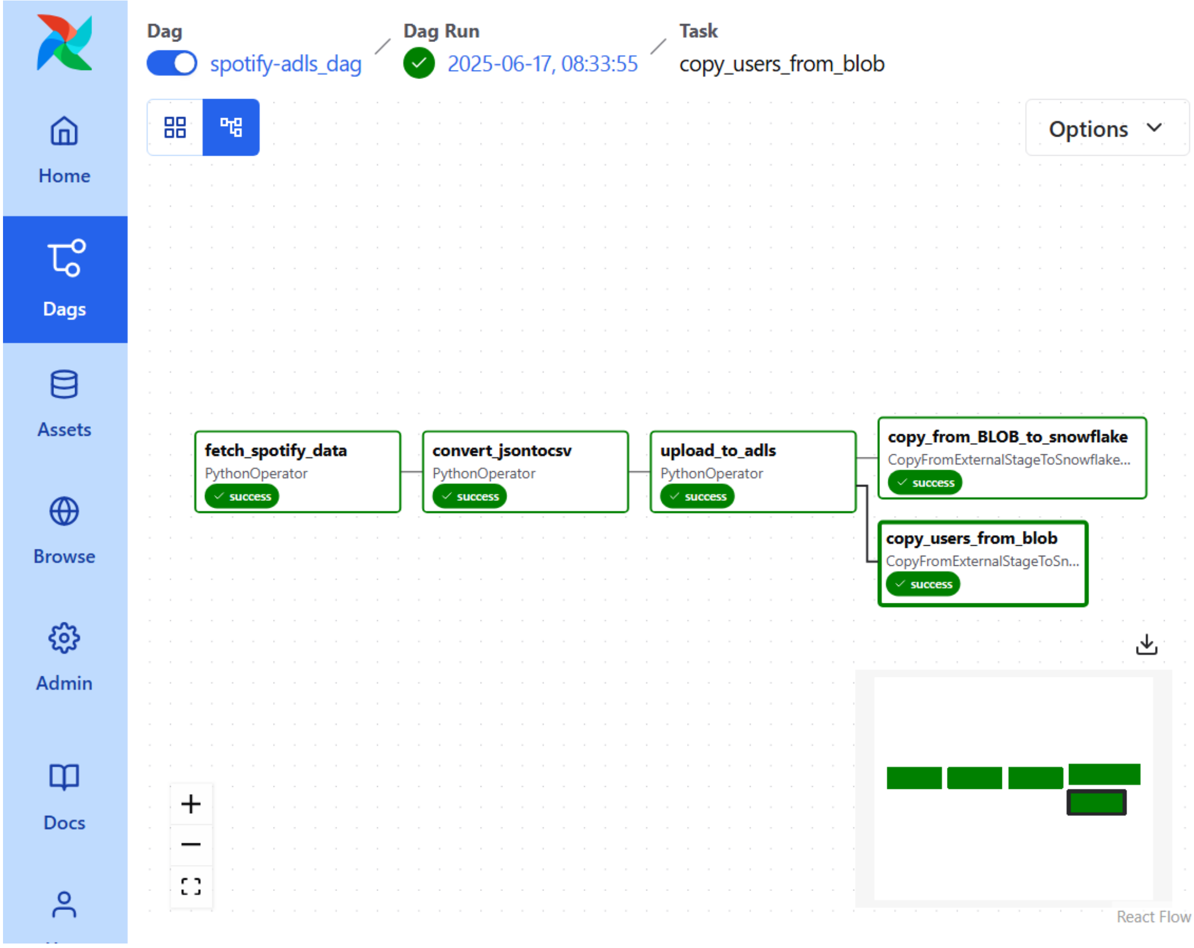
- Defined tasks: data fetching → transformation → upload → Snowflake load.
- Enabled XCom push/pull for inter-task communication.



Key	Value	Description	Is Encrypted
spotify_client_secret	***		true
spotify_client_id	7574a6a502ac471eaa9aa746ecad87c6		true



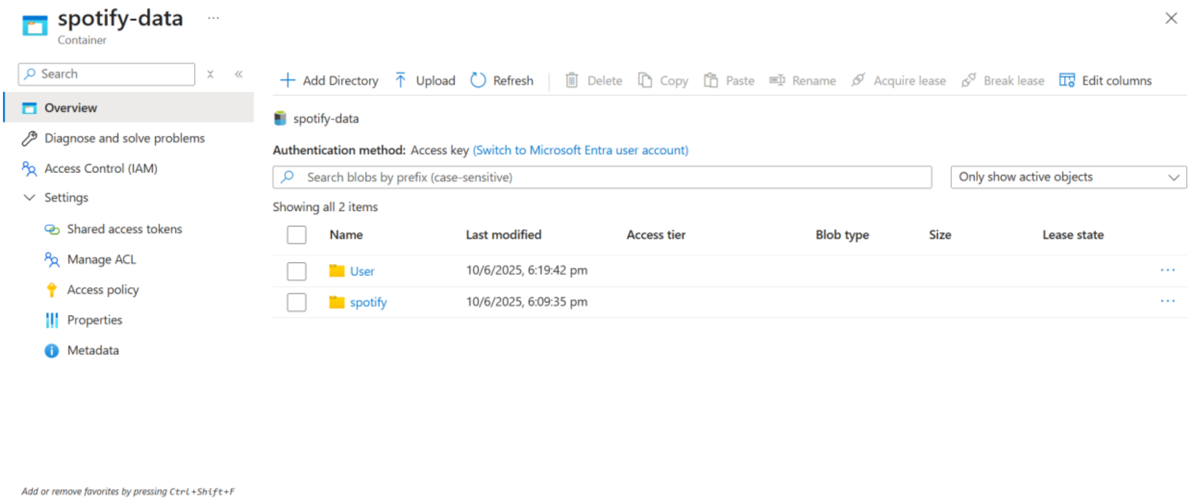
Connection Id	Connection Type	Description	Host	Port
snowflake_defaults	snowflake			
azure_blob_conn	azure_data_lake			



Dag spotify-adls_dag / Dag Run 2025-06-17, 08:33:55 / Task copy_users_from_blob

Step 3: Blob Storage Integration

- Uploaded enriched CSV to Azure using BlobServiceClient.
- Organized folder hierarchy for reusability and versioning.



spotify-data
Container

spotify-data

Authentication method: Access key (Switch to Microsoft Entra user account)

Showing all 2 items

□	Name	Last modified	Access tier	Blob type	Size	Lease state
□	User	10/6/2025, 6:19:42 pm				
□	spotify	10/6/2025, 6:09:35 pm				

Add or remove favorites by pressing Ctrl+Shift+F

Step 4: Snowflake Processing

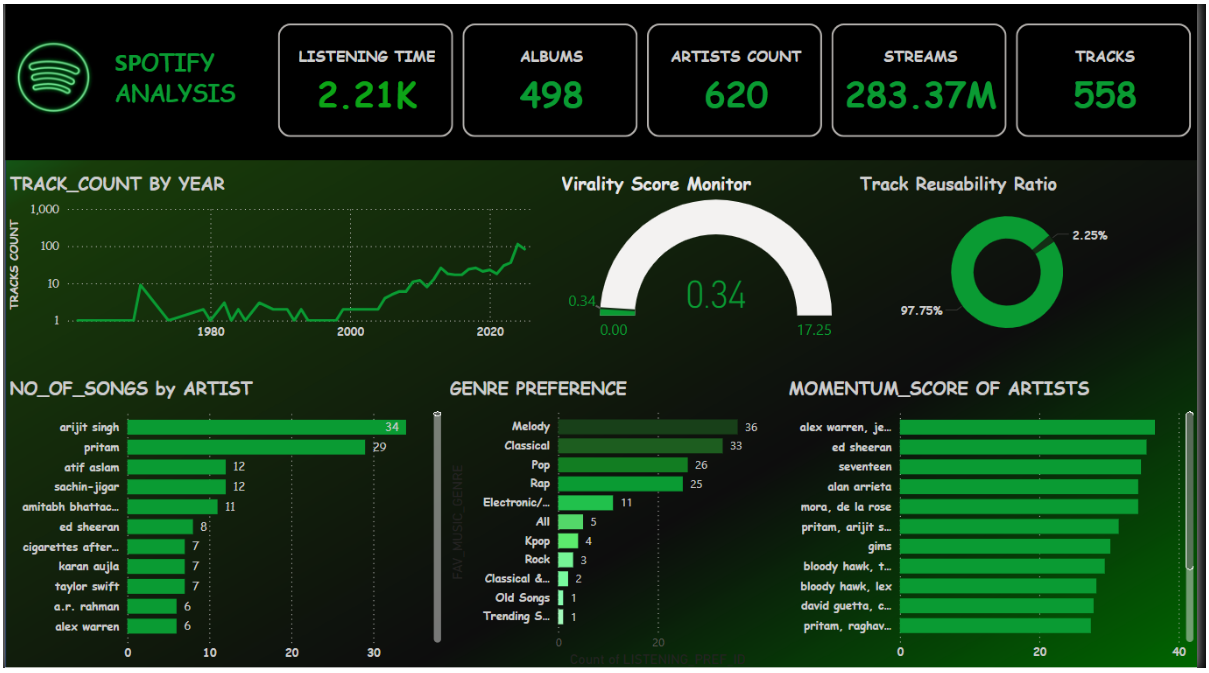
- Created two external stages and tables for raw Spotify and user data.
- Built Silver Layer tables using data cleansing logic.
- Created Gold Layer models using dimension-fact schema.

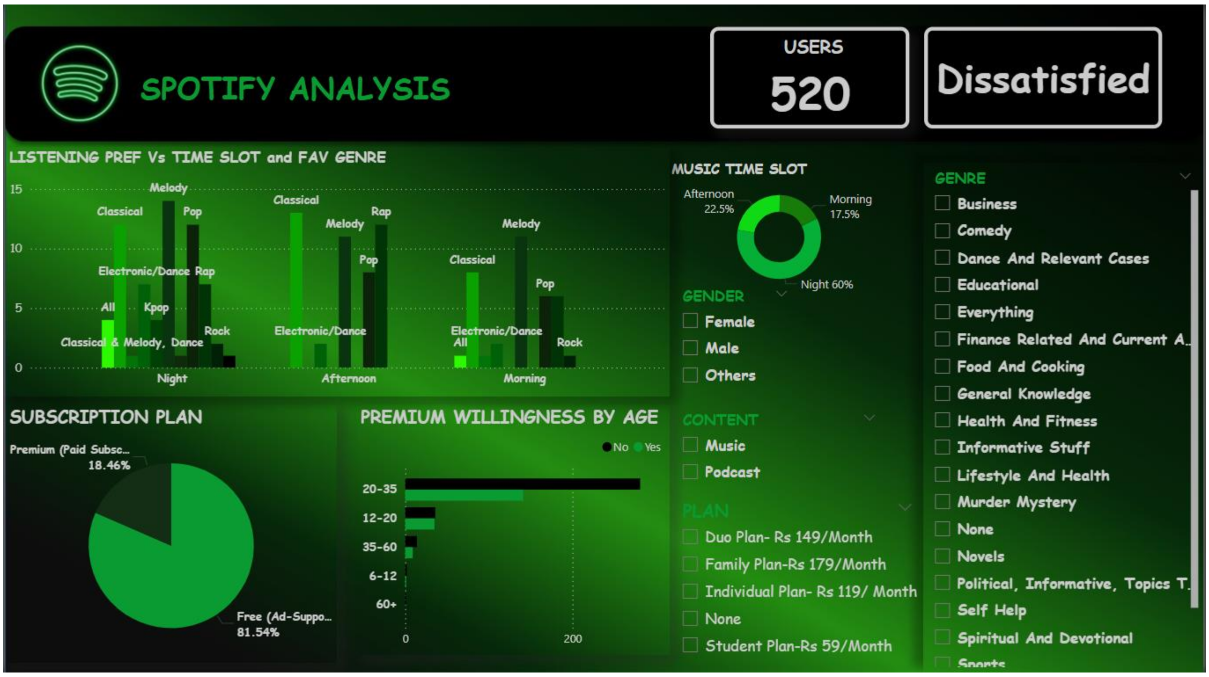
Step 5: Power BI Dashboards

- Built visuals: Top genres by age, virality score, premium conversion rate.
- Enabled dynamic filtering and drill-down options.

Results

- Created 15+ dimension and fact tables in Snowflake
- Automated Airflow pipeline with robust error handling
- Processed over 1000 tracks with metadata and enrichment
- Designed multiple interactive Power BI visuals
- Optimized storage and queries using columnar formats





Future Scope

- **Real-Time Integration:** Use Kafka or Azure Event Hub to stream live track data.
- **Predictive Analytics:** Train ML models on user behavior and genre preference.
- **Cross-Dataset Correlation:** Integrate with social media or event-based data for deeper insights.

Conclusion

This project demonstrates a scalable and automated data pipeline that transforms Spotify's raw data into actionable intelligence. By integrating open APIs, modern data warehousing, and BI tools, we've built a reusable framework that supports advanced analytics and storytelling.

The modular nature of the solution allows for further extension to accommodate real-time data, machine learning models, and crossplatform integration, making it a future-ready analytics ecosystem for digital music data.

References

- Spotify API Documentation
- Azure Data Factory Docs
- Snowflake Documentation
- Power BI
- LangDetect Library
- Airflow Docs