Spotify Data Engineering & ML Project on GCP

Overview

This project demonstrates how to build a complete cloud-native data pipeline and machine learning model using Google Cloud Platform (GCP). The goal is to analyze and predict the popularity of Spotify tracks using BigQuery ML and audio feature data.

Step-by-Step Implementation

Step 1: GCP Project Setup

- **Created GCP Project**: `melodies`
- **Enabled APIs**:
  - Cloud Storage API
  - BigQuery API
  - IAM API
- **Created Storage Bucket**: `test513` (Multi-region, Fine-grained access)
- **Uploaded Dataset**: `dataset_playlist_2010to2022.csv`

Step 2: BigQuery Dataset & Table

- **Created BigQuery Dataset**: `spotify_data`
- **Imported CSV as Table**: `playlist_data` (auto-detect schema)

```
SELECT *
FROM `melodies-459020.spotify_data.playlist_data`
LIMIT 10;
```

Exploration Goal: Understand schema, spot nulls, count records.

Step 3: Data Cleaning

- **Checked Nulls** using `COUNTIF()` in SQL
- **Created Clean Table**:

```
CREATE OR REPLACE TABLE `melodies-459020.spotify_data.cleaned_playlist_data` AS
SELECT *
FROM `melodies-459020.spotify_data.playlist_data`
WHERE duration_ms IS NOT NULL
  AND energy IS NOT NULL
  AND danceability IS NOT NULL
  AND valence IS NOT NULL
  AND tempo IS NOT NULL;
```

Step 4: Trend Analysis

- **Top Artists by Popularity**:

```
SELECT artist_name, AVG(track_popularity) AS avg_popularity
FROM `melodies-459020.spotify_data.cleaned_playlist_data`
GROUP BY artist_name
ORDER BY avg_popularity DESC
LIMIT 10;
```

- **Genre Popularity by Year**:

```
SELECT year, artist_genres AS genre, AVG(track_popularity) AS avg_popularity
FROM `melodies-459020.spotify_data.cleaned_playlist_data`
GROUP BY year, genre
ORDER BY year, avg_popularity DESC;
```

Step 5: ML Model Training (BigQuery ML)

- **Model Type**: Linear Regression
- **Training Query**:

```
CREATE OR REPLACE MODEL `melodies-459020.spotify_data.track_popularity_model`
OPTIONS(model_type='linear_reg', input_label_cols=['track_popularity']) AS
SELECT danceability, energy, valence, tempo, duration_ms, track_popularity
FROM `melodies-459020.spotify_data.cleaned_playlist_data`
WHERE track_popularity IS NOT NULL;
```

Step 6: Model Evaluation

- **Evaluated Using**:

```
SELECT *
FROM ML.EVALUATE(MODEL `melodies-459020.spotify_data.track_popularity_model`);
```

Key Metrics:

- R^2 Score: ~0.014
- MAE: ~8.7
- Median AE: ~7.2

Step 7: Predictions

- **Predict Track Popularity**:

```
SELECT *
FROM ML.PREDICT(
  MODEL `melodies-459020.spotify_data.track_popularity_model`,
  (SELECT danceability, energy, valence, tempo, duration_ms
   FROM `melodies-459020.spotify_data.cleaned_playlist_data`
   LIMIT 5)
);
```

Business Value

- Predict track popularity for marketing and curation.
- Analyze musical trends and genre rise/fall by year.
- Enable AI tools (chatbots or dashboards) for music intelligence.

Next Steps

- Add advanced features (e.g., `speechiness`, `artist_genres`)
- Switch to `BOOSTED_TREE_REGRESSOR` for better predictions
- Build dashboard or chatbot interface for interactive use

Tools Used

- GCP: BigQuery, Cloud Storage, IAM
- SQL
- BigQuery ML

Diagram: Workflow Overview

[CSV Upload] --> [GCS Bucket] --> [BigQuery Table] --> [Data Cleaning] --> [ML Model] --> [Evaluation + Predictions]
