Azure Rhythm

Cloud Miners
SMU - 2023

The Idea…

Presentation Journey

1. R Shiny Application - Haitie
2. ETL - Mai
3. Azure Database – Lijo
4. Azure Machine Learning – Todd

Guided by Lani

Azure Rhythm

R Shiny Application

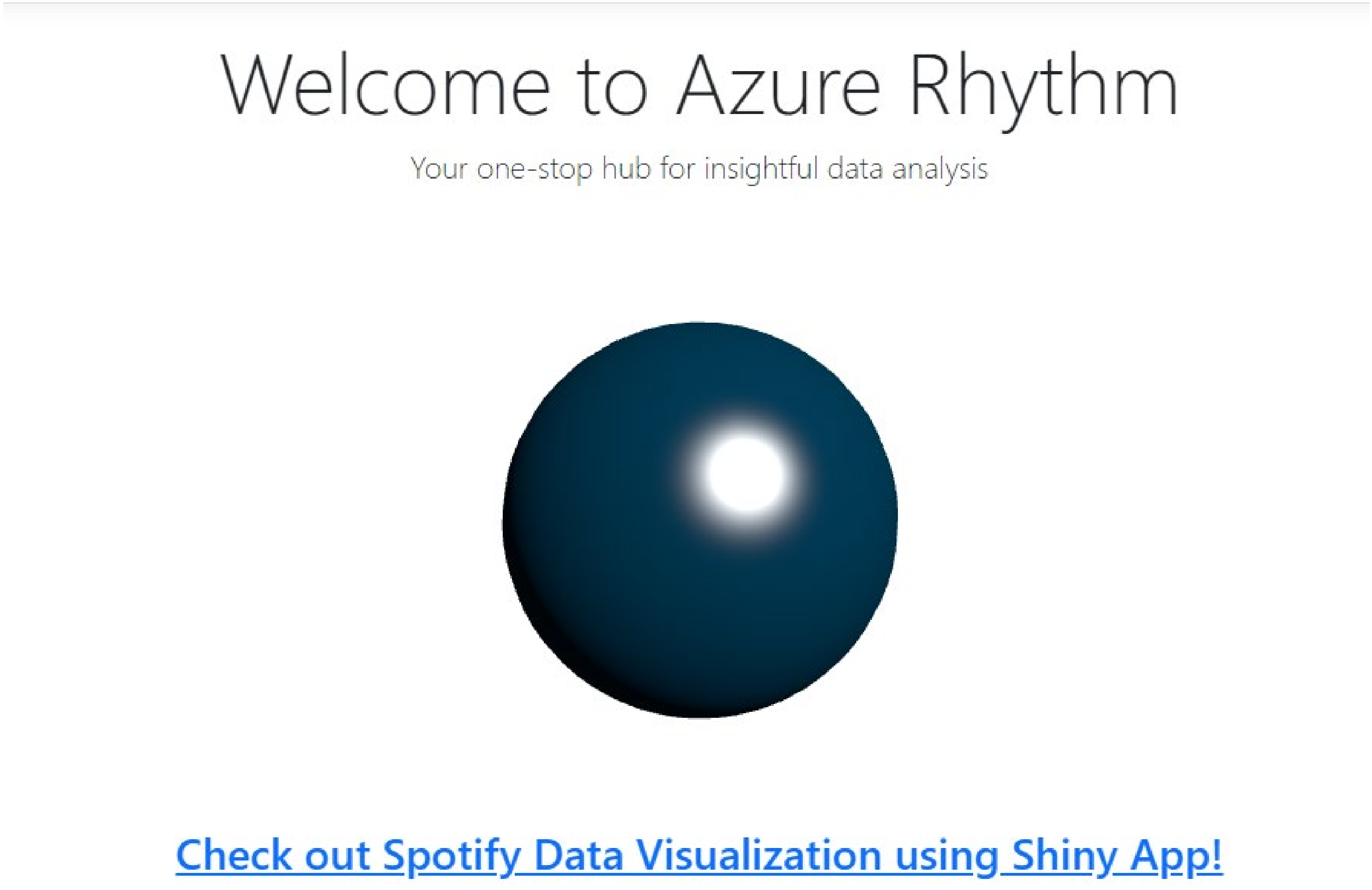
Haitie Liu

Azure Rhythm

R Shiny Application

https://azrhythmwebapp.azurewebsites.net/

Screenshots of our R Shiny App

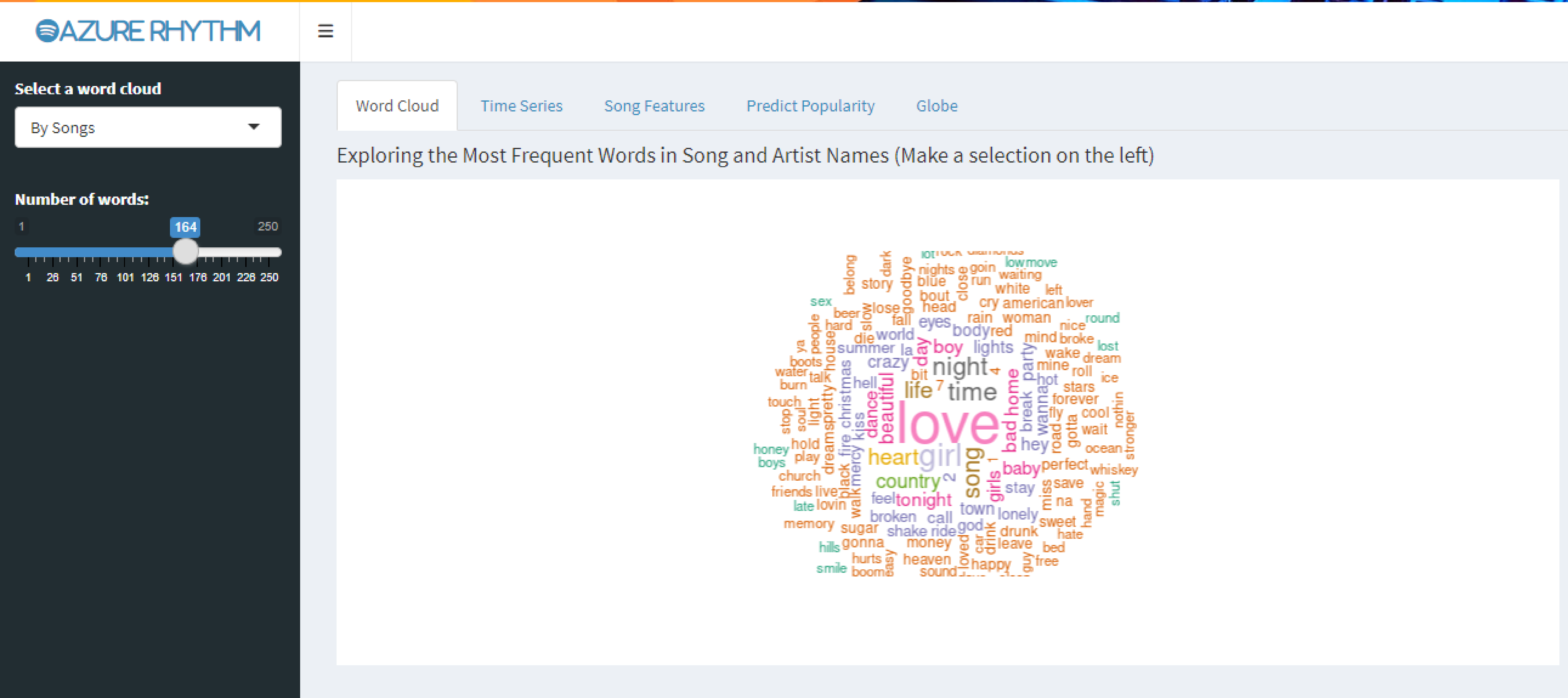


Welcome to Azure Rhythm

Your one-stop hub for insightful data analysis

Check out Spotify Data Visualization using Shiny App!

Screenshots of our ShinyApp

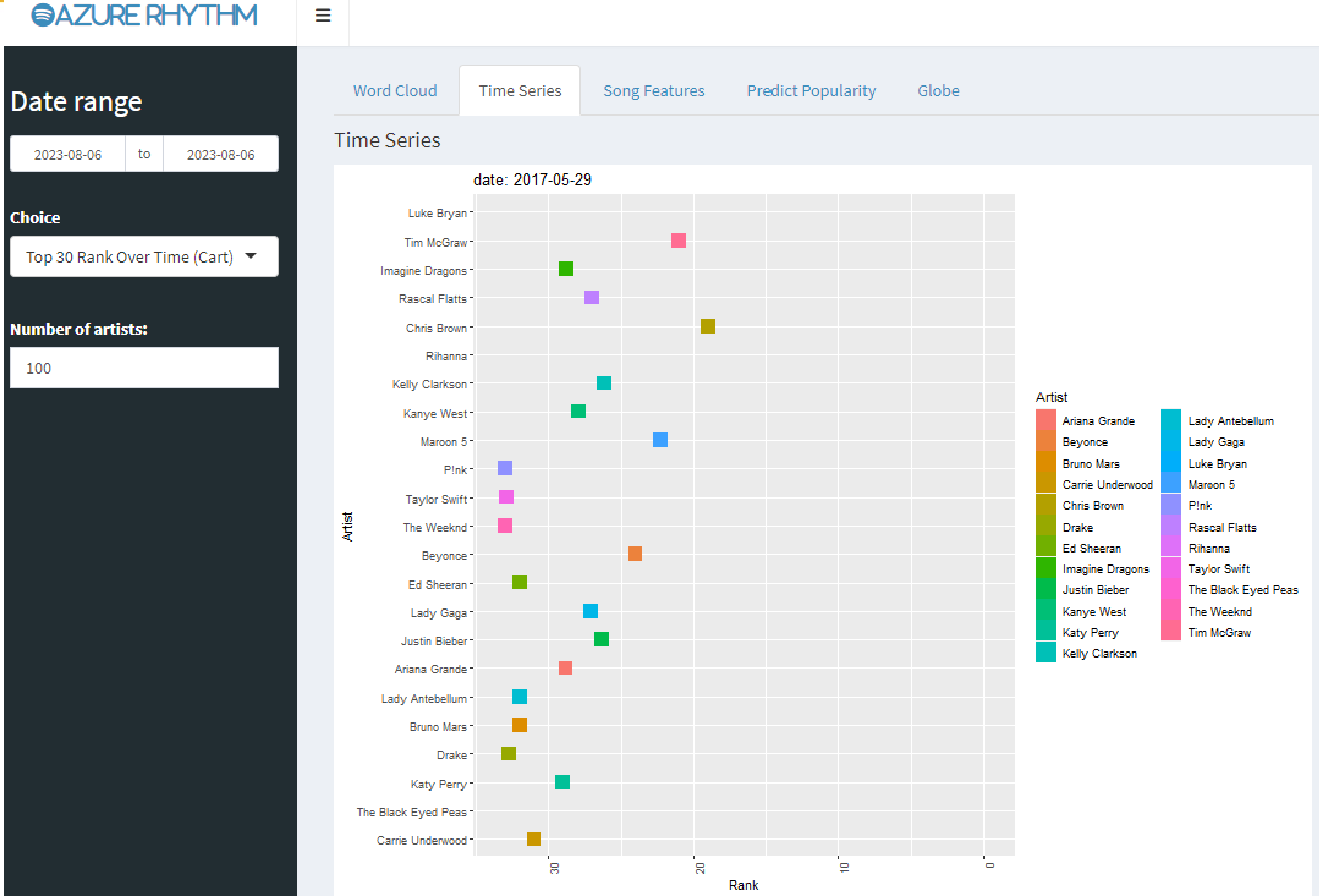


Select a word cloud: By Songs

Number of words: 164

Exploring the Most Frequent Words in Song and Artist Names (Make a selection on the left)

Screenshots – R Shiny App



AZURE RHYTHM

Date range

2023-08-06 to 2023-08-06

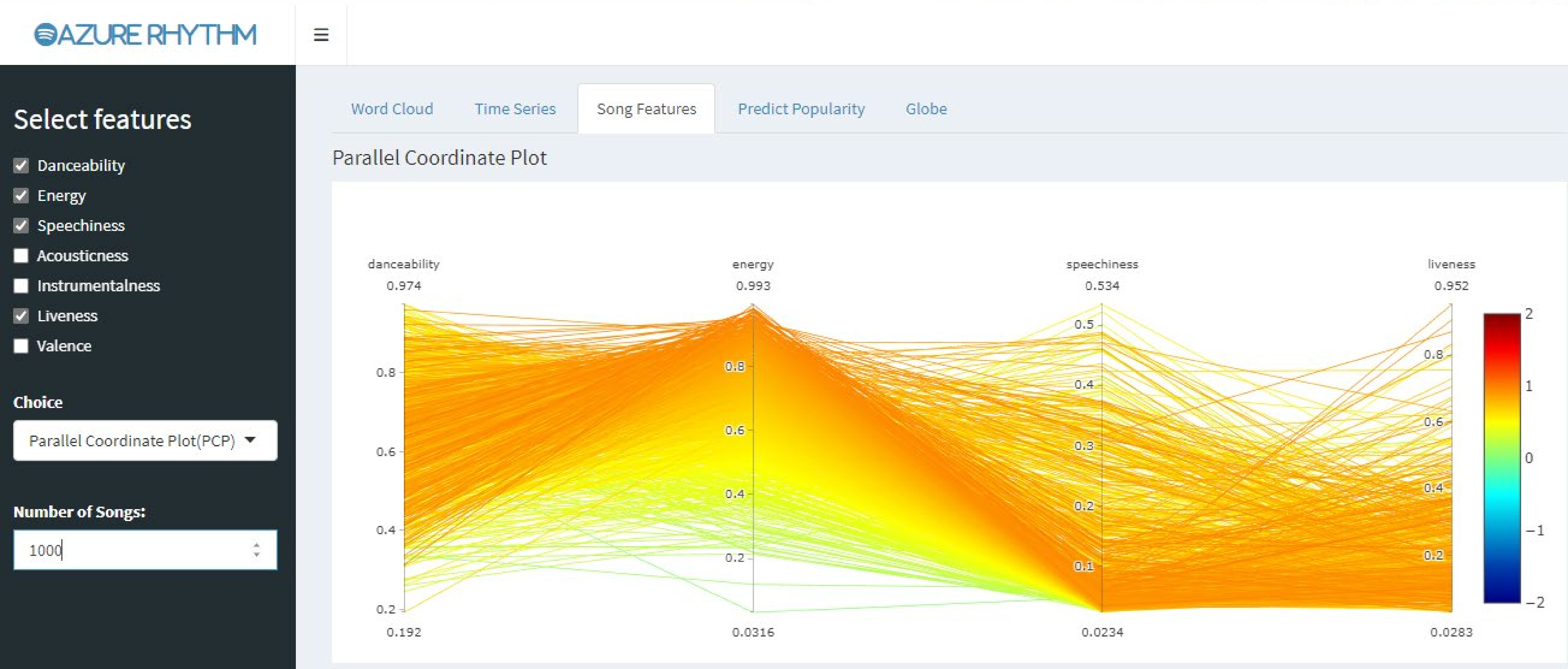
Choice

Top 30 Rank Over Time (Cart)

Number of artists:

100

Screenshots – R Shiny App



Select features

- ☑ Danceability
- ☑ Energy
- ☑ Speechiness
- □ Acousticness
- □ Instrumentalness
- ☑ Liveness
- □ Valence

Choice

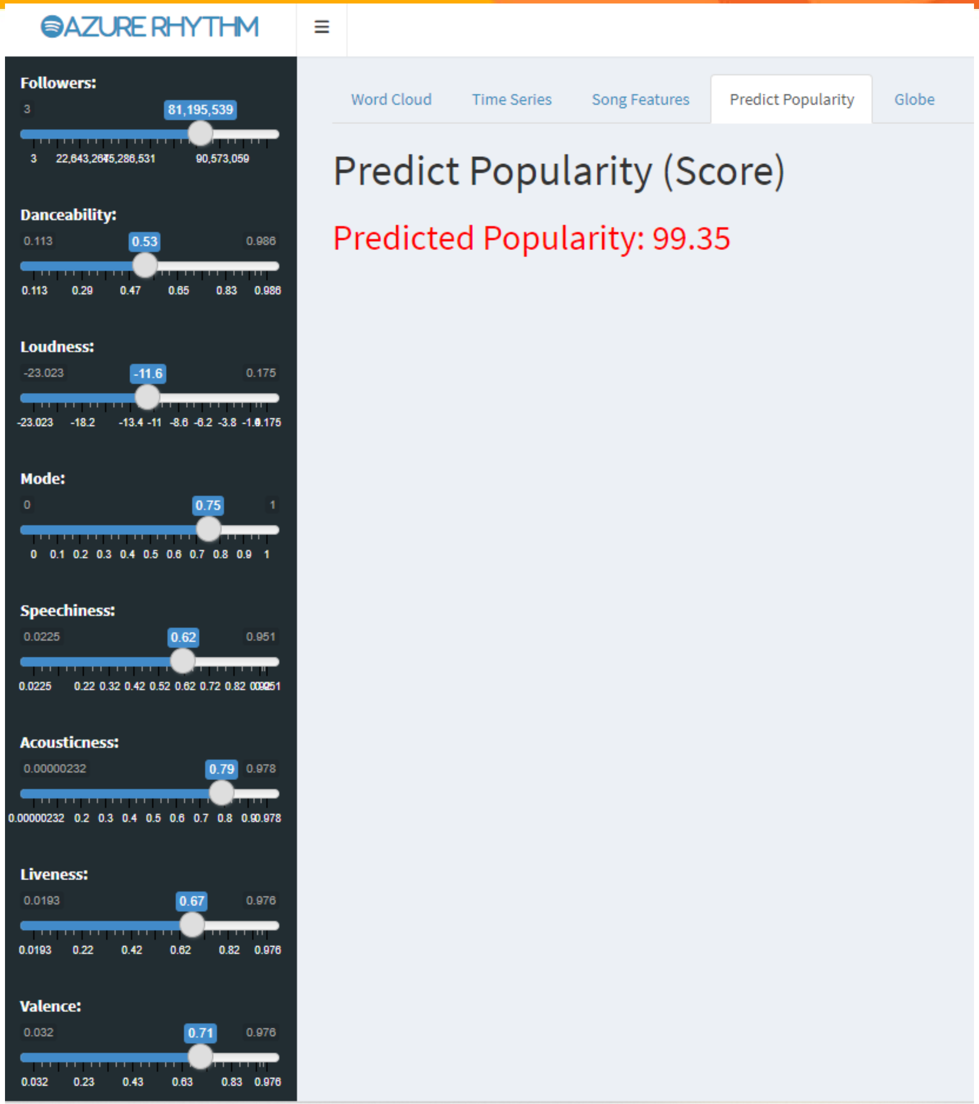
Parallel Coordinate Plot(PCP)

Number of Songs:

1000

Parallel Coordinate Plot

Screenshots – R Shiny App



Followers: 3 81,195,539

Danceability: 0.113 0.53 0.986

Loudness: -23.023 -11.6 0.175

Mode: 0 0.75 1

Speechiness: 0.0225 0.62 0.951

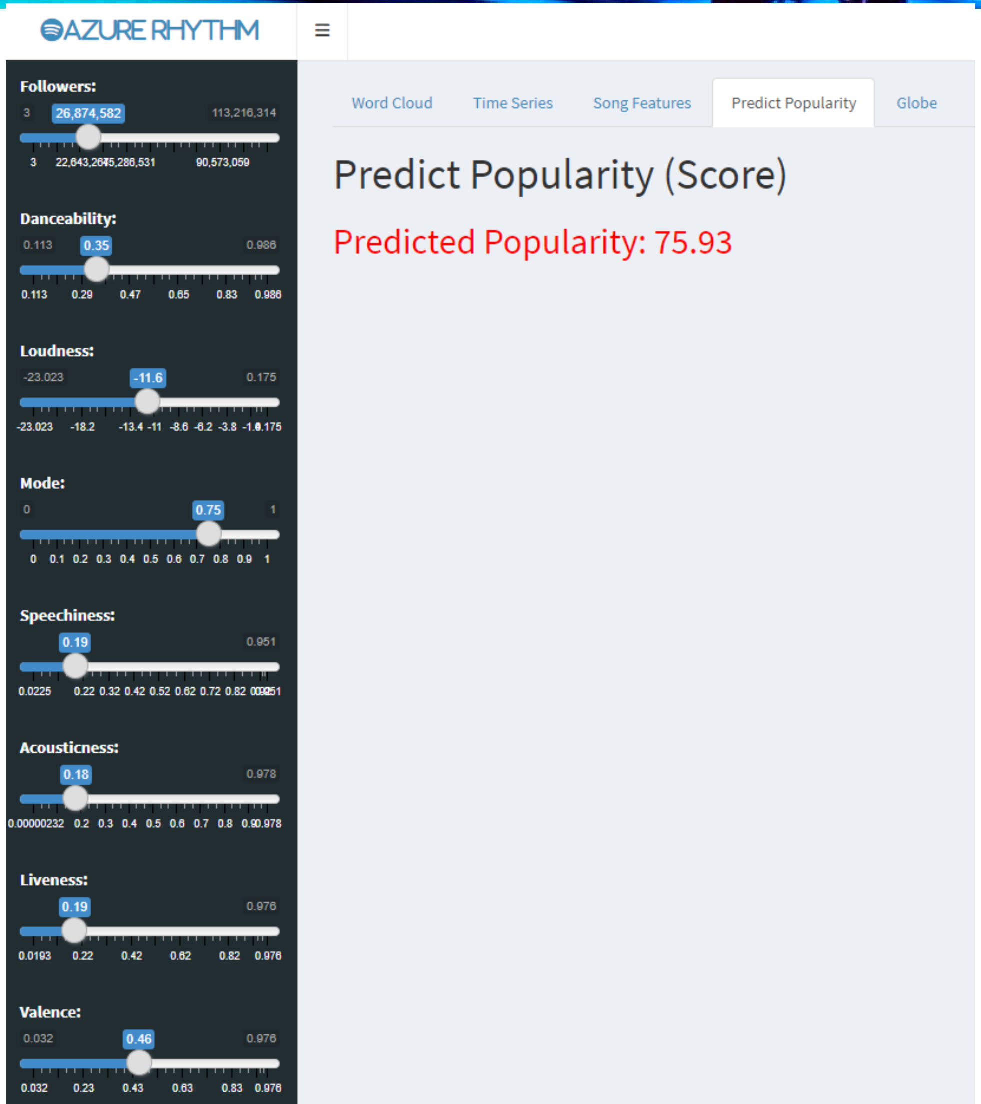
Acousticness: 0.00000232 0.79 0.978

Liveness: 0.0193 0.67 0.976

Valence: 0.032 0.71 0.976

Predict Popularity (Score)

Predicted Popularity: 99.35



Followers: 3 26,874,582 113,216,314

Danceability: 0.113 0.35 0.986

Loudness: -23.023 -11.6 0.175

Mode: 0 0.75 1

Speechiness: 0.19 0.951

Acousticness: 0.18 0.978

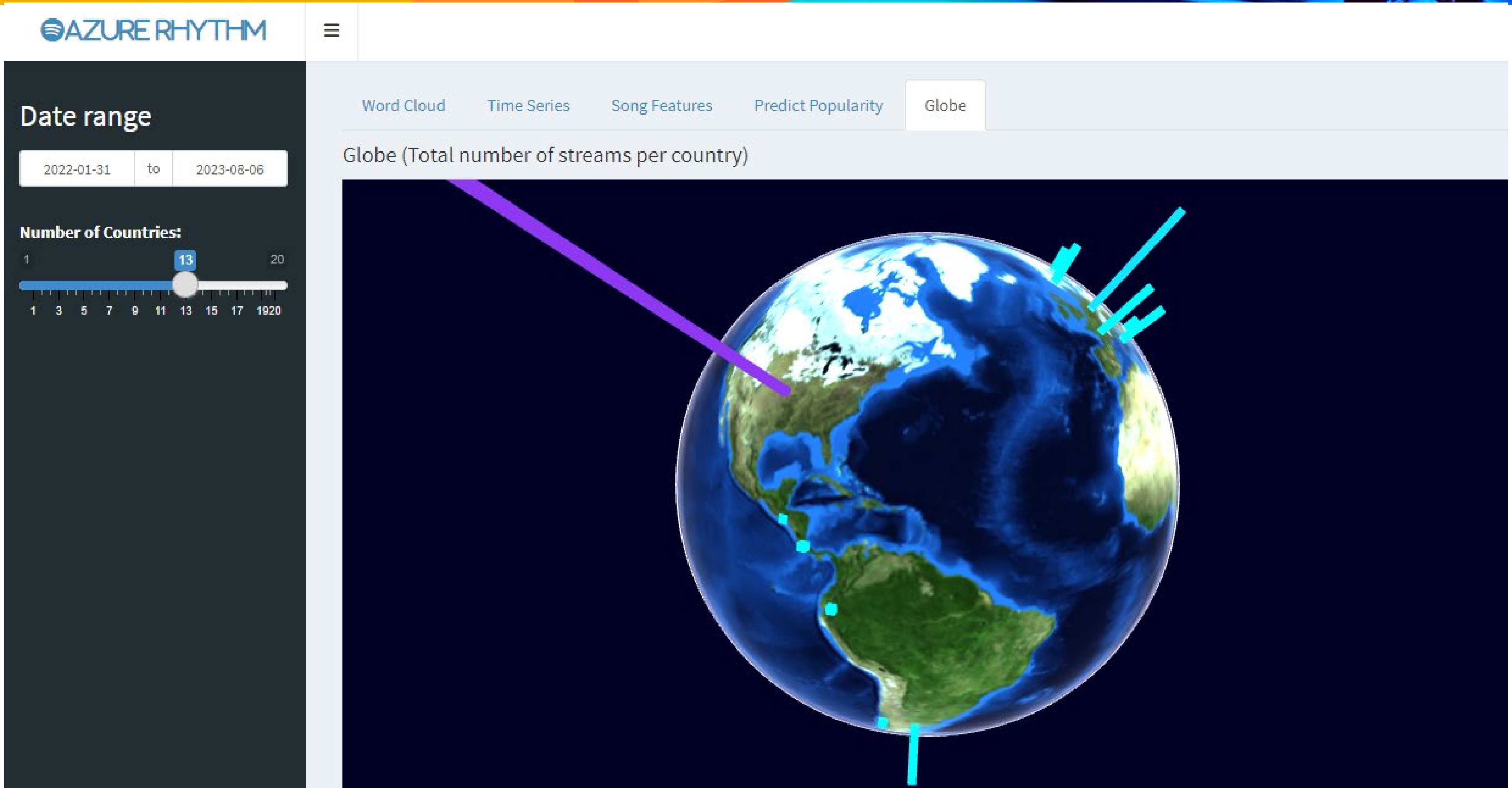
Liveness: 0.19 0.976

Valence: 0.032 0.46 0.976

Predict Popularity (Score)

Predicted Popularity: 75.93

Screenshots – R Shiny App



Date range

2022-01-31 to 2023-08-06

Number of Countries: 13

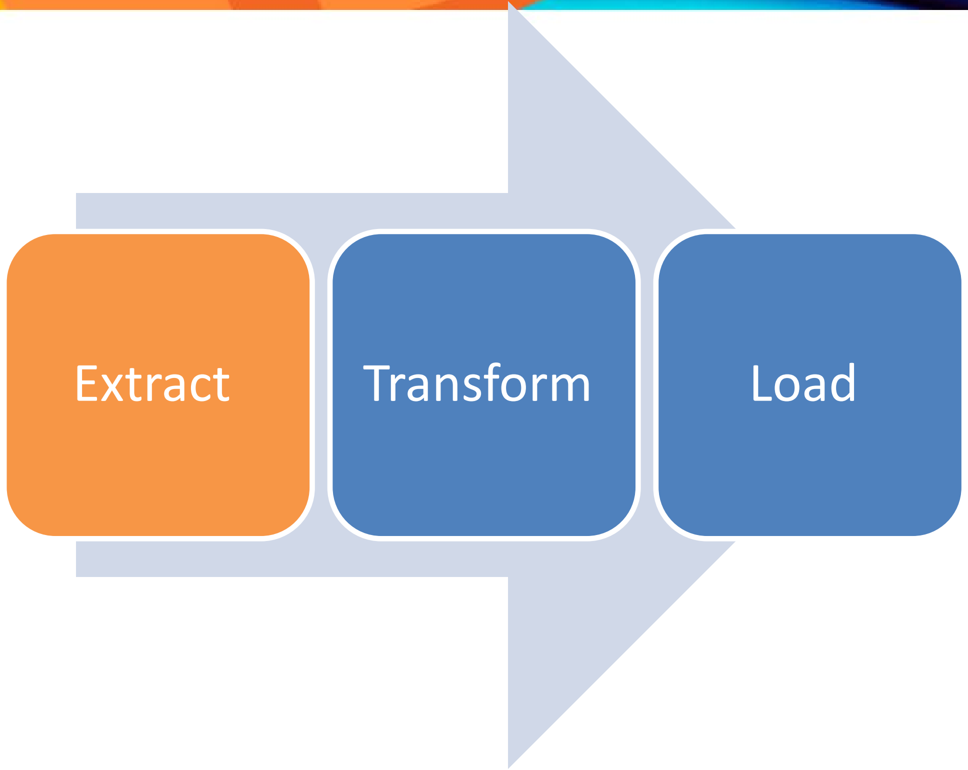
Globe (Total number of streams per country)

Extract, Transfer, Load (ETL)

Python

Mai Dang

ETL

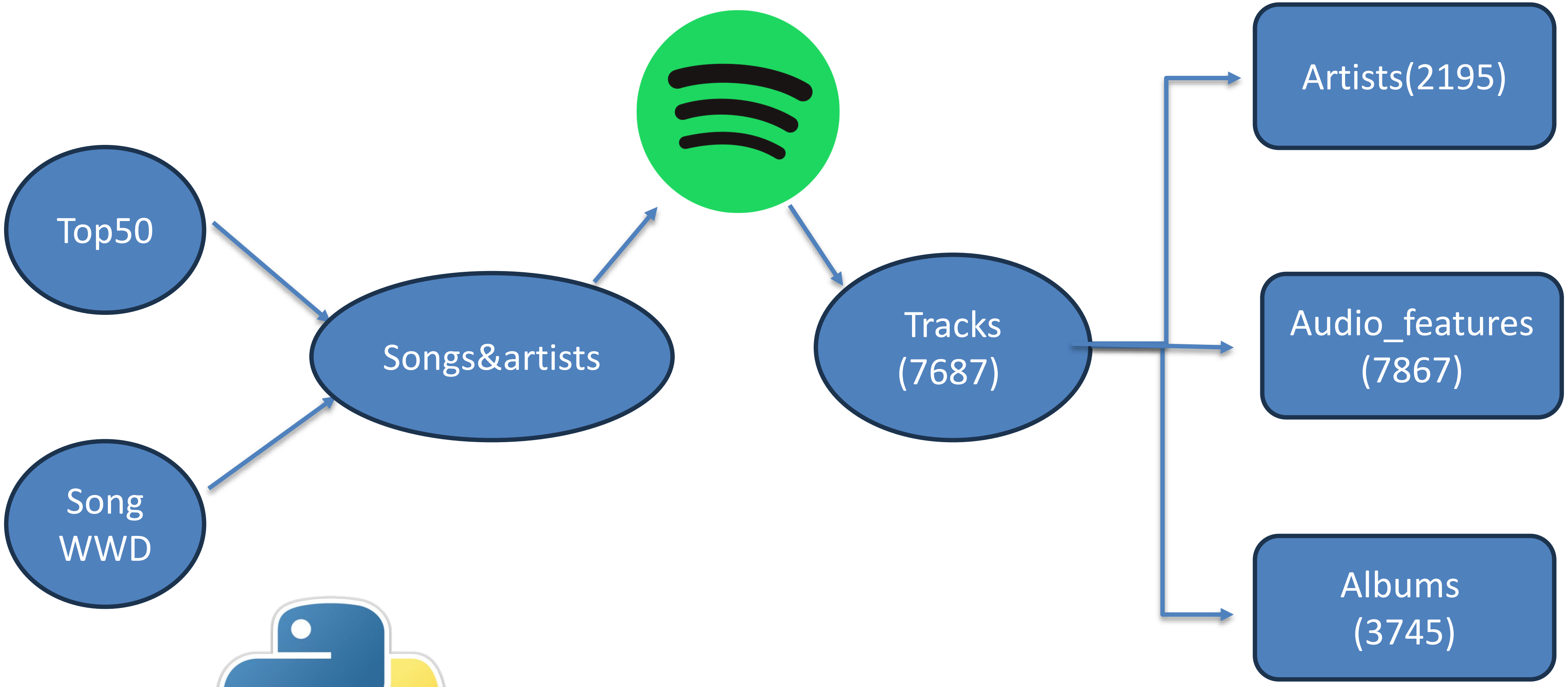


Extract

Transform

Load

Extract

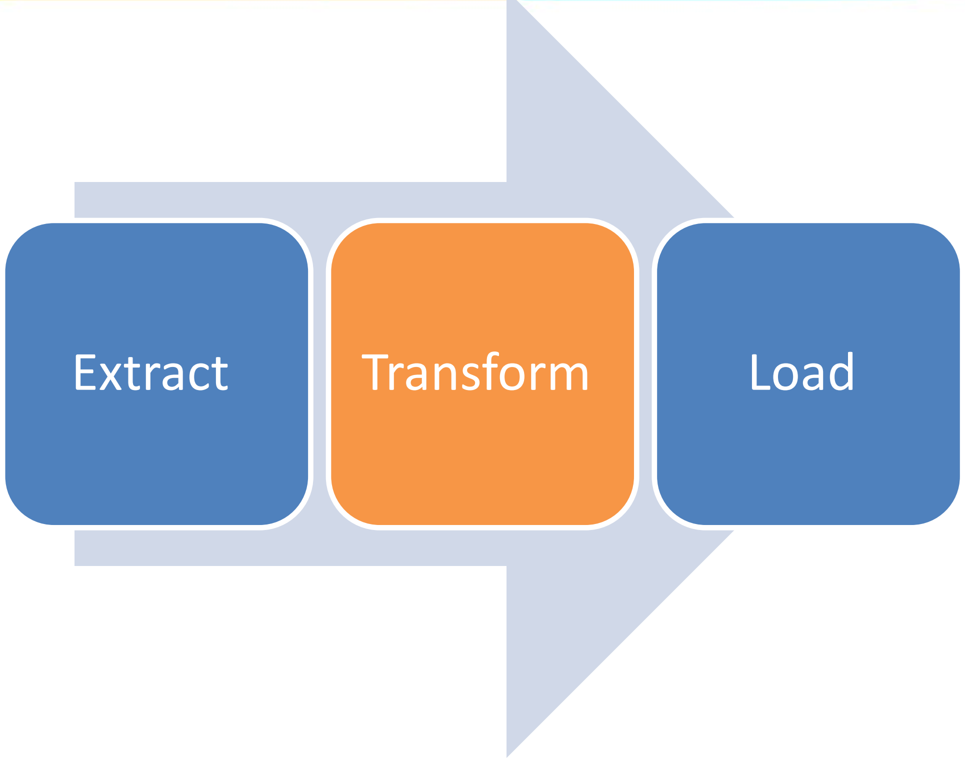


Using library Spotipy

Extract

Roadblocks	Solutions
• Spotify API limits: rate limit, token limit, etc.	• Client Credential authentication Break between loops Response Counter.
• Duplicated IDs	• Check and remove duplicates.
• Inconsistent artist names in 2 datasets	• Remove symbols and characters before normalizing the artist names.

ETL



Extract

Transform

Load

Transform

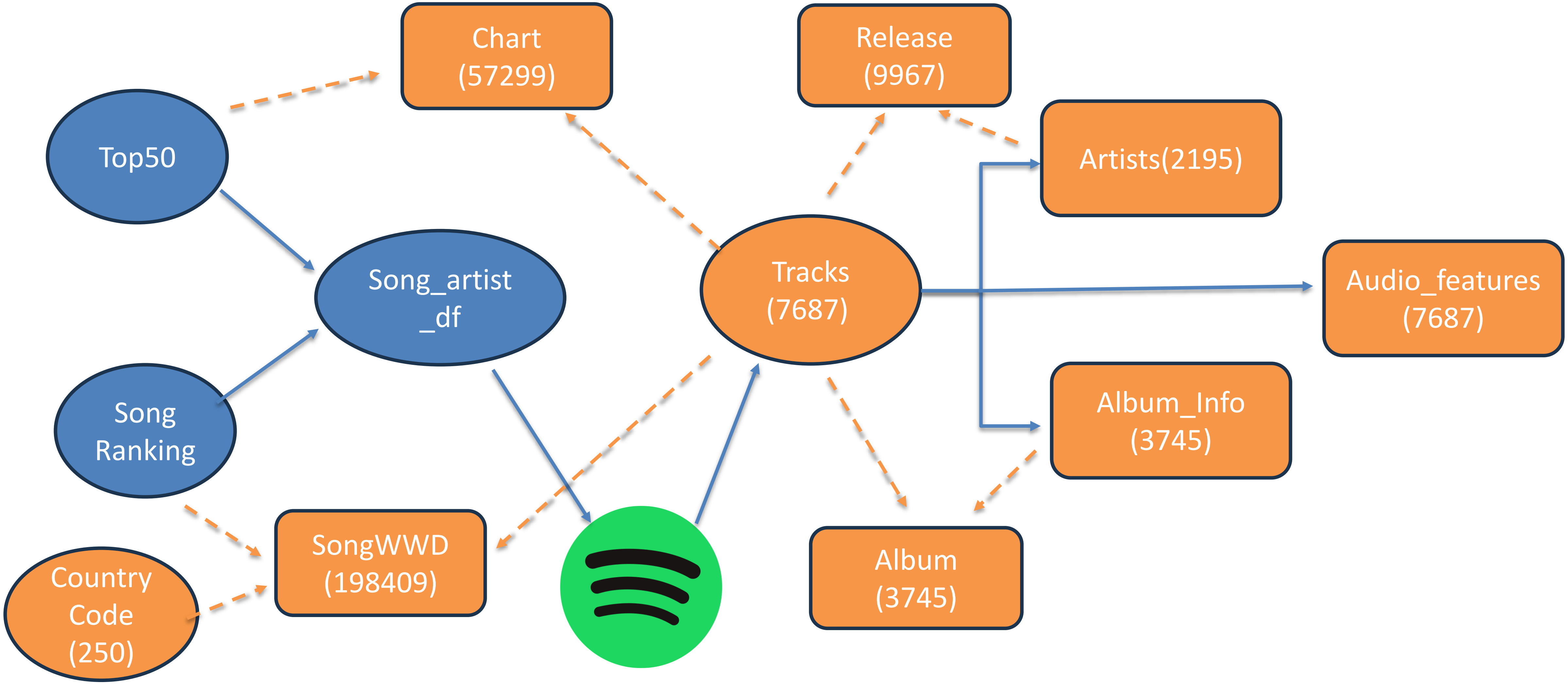
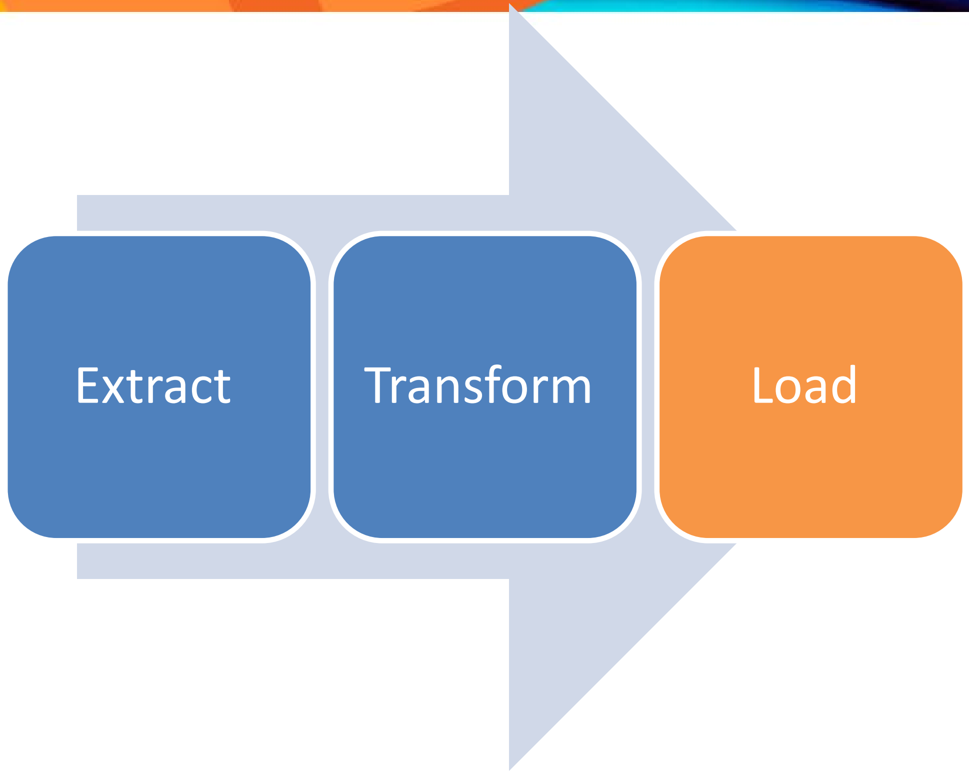


Chart
(57299)
Release
(9967)
Top50
Artists(2195)
Song_artist
_df
Tracks
(7687)
Audio_features
(7687)
Song
Ranking
Album_Info
(3745)
SongWWD
(198409)
Album
(3745)
Country
Code
(250)

ETL

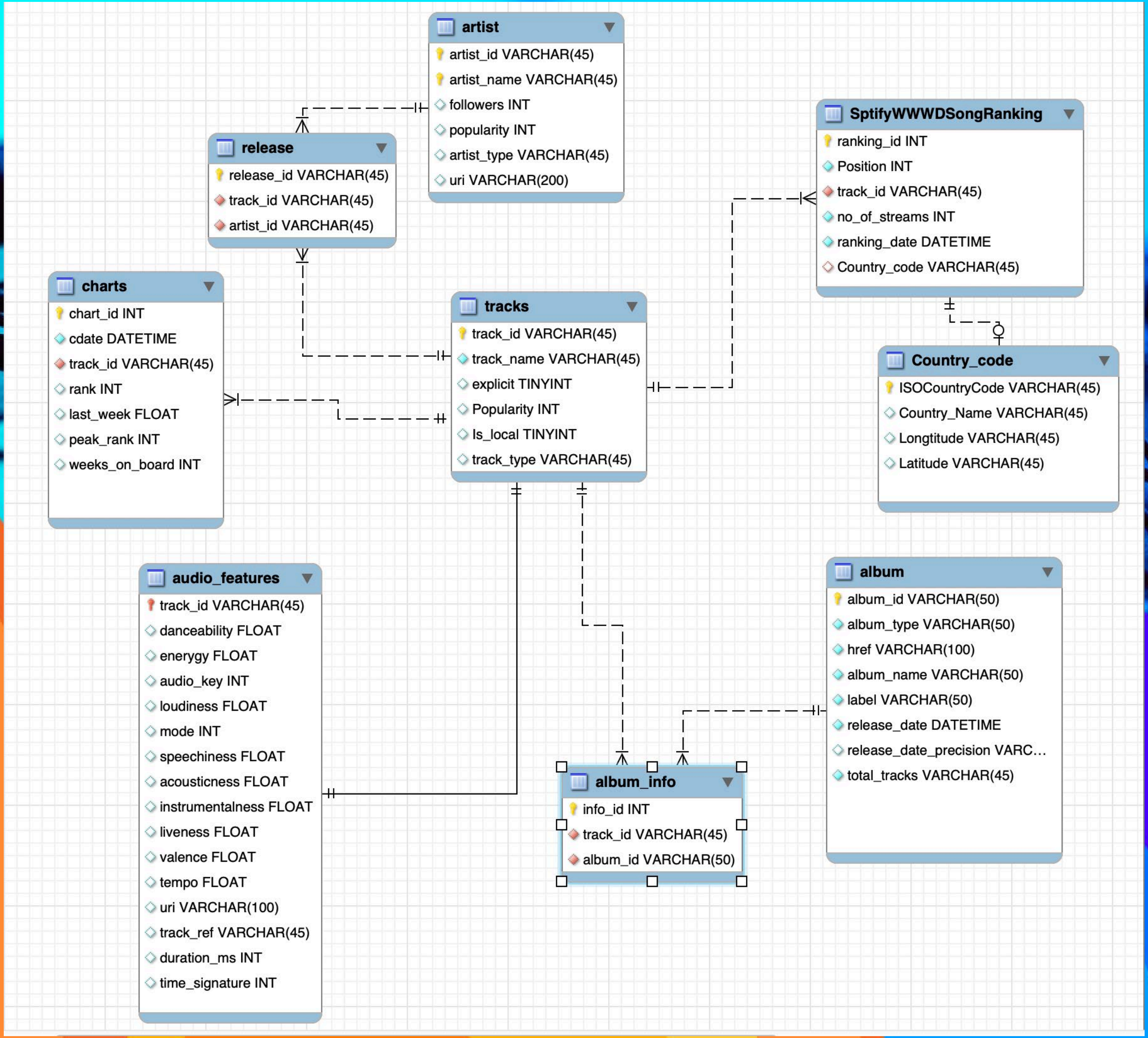


Extract

Transform

Load

ERD



artist
- artist_id VARCHAR(45)
- artist_name VARCHAR(45)
- followers INT
- popularity INT
- artist_type VARCHAR(45)
- uri VARCHAR(200)

release
- release_id VARCHAR(45)
- track_id VARCHAR(45)
- artist_id VARCHAR(45)

SptifyWWWDSongRanking
- ranking_id INT
- Position INT
- track_id VARCHAR(45)
- no_of_streams INT
- ranking_date DATETIME
- Country_code VARCHAR(45)

charts
- chart_id INT
- cdate DATETIME
- track_id VARCHAR(45)
- rank INT
- last_week FLOAT
- peak_rank INT
- weeks_on_board INT

tracks
- track_id VARCHAR(45)
- track_name VARCHAR(45)
- explicit TINYINT
- Popularity INT
- Is_local TINYINT
- track_type VARCHAR(45)

Country_code
- ISOCountryCode VARCHAR(45)
- Country_Name VARCHAR(45)
- Longtitude VARCHAR(45)
- Latitude VARCHAR(45)

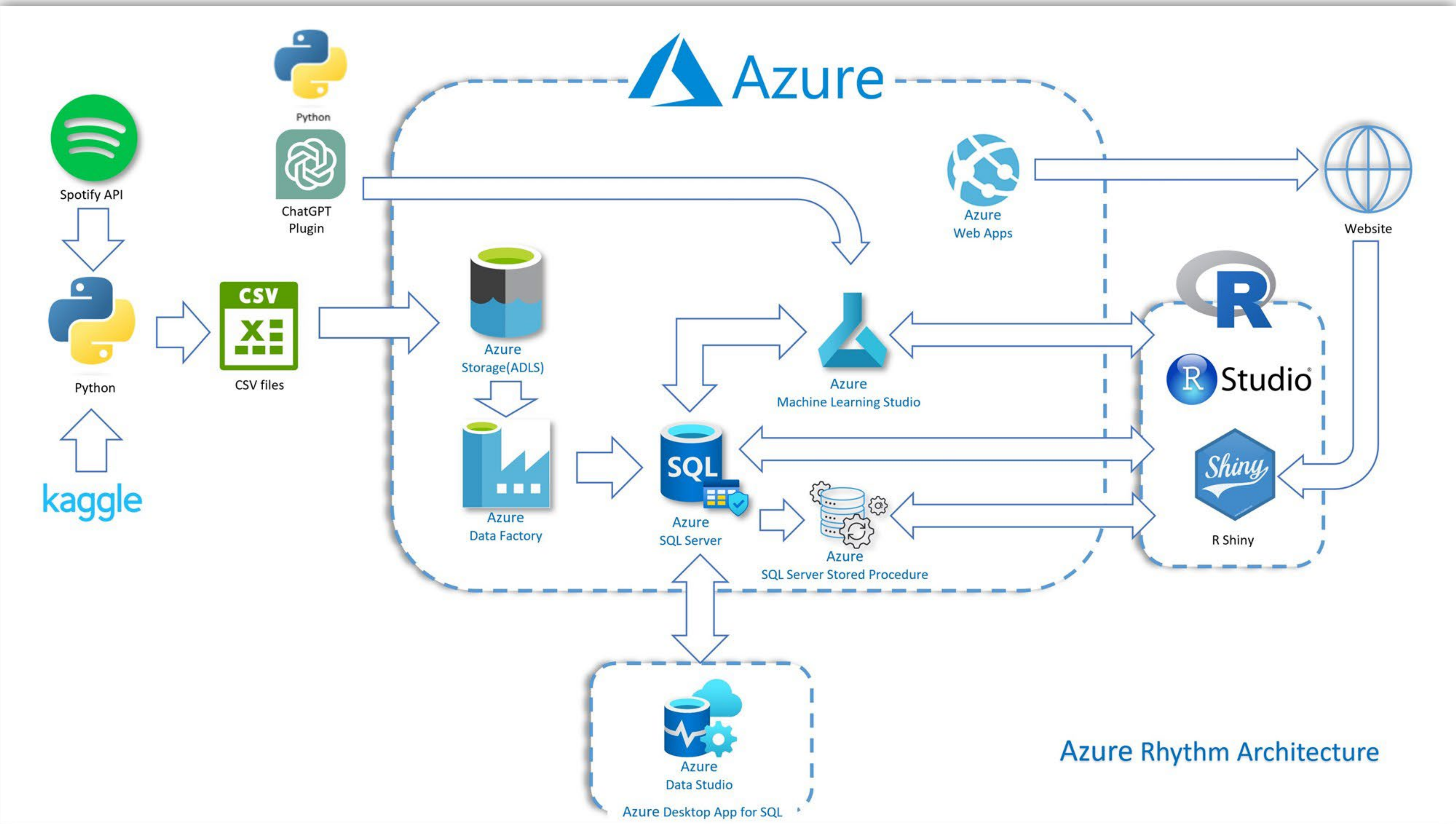
audio_features
- track_id VARCHAR(45)
- danceability FLOAT
- enerygy FLOAT
- audio_key INT
- loudiness FLOAT
- mode INT
- speechiness FLOAT
- acousticness FLOAT
- instrumentalness FLOAT
- liveness FLOAT
- valence FLOAT
- tempo FLOAT
- uri VARCHAR(100)
- track_ref VARCHAR(45)
- duration_ms INT
- time_signature INT

album_info
- info_id INT
- track_id VARCHAR(45)
- album_id VARCHAR(50)

album
- album_id VARCHAR(50)
- album_type VARCHAR(50)
- href VARCHAR(100)
- album_name VARCHAR(50)
- label VARCHAR(50)
- release_date DATETIME
- release_date_precision VARC…
- total_tracks VARCHAR(45)

Azure Database

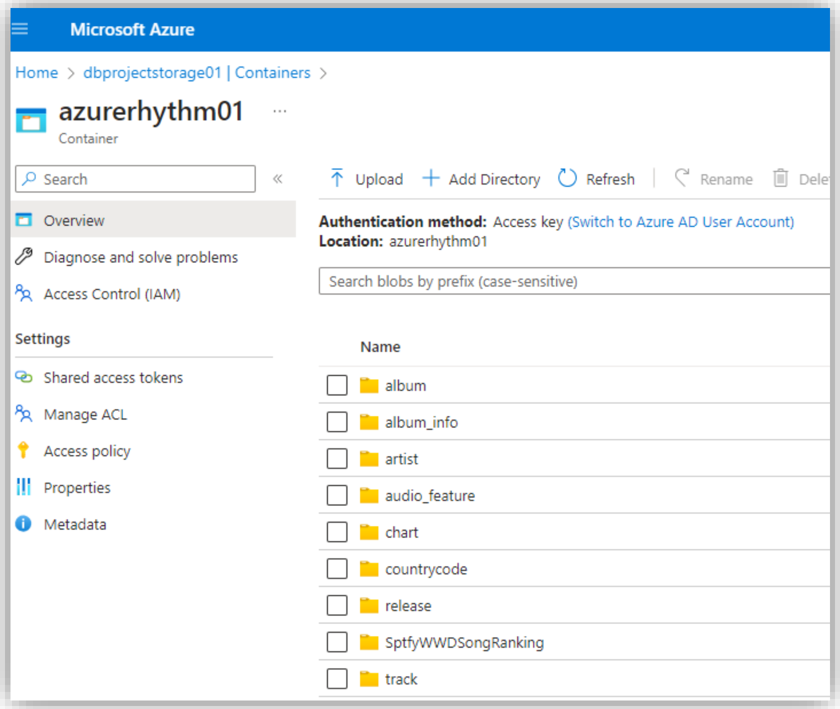
Lijo Jacob



Azure Rhythm Architecture

Azure Setup & Data Migration to Azure

- Setup Azure student account for each team member.
- Created resources under one of our subscriptions and assigned “Contributor” role to each team member.
- Setup a storage account and created a container
- Data stored in csv files were migrated to Azure data lake storage.



Microsoft Azure

Home > dbprojectstorage01 | Containers >

azurerhythm01 ···

Container

Search «

Upload + Add Directory Refresh Rename Dele

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Manage ACL

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Azure AD User Account)

Location: azurerhythm01

Search blobs by prefix (case-sensitive)

Name

- □ album
- □ album_info
- □ artist
- □ audio_feature
- □ chart
- □ countrycode
- □ release
- □ SptfyWWDSongRanking
- □ track

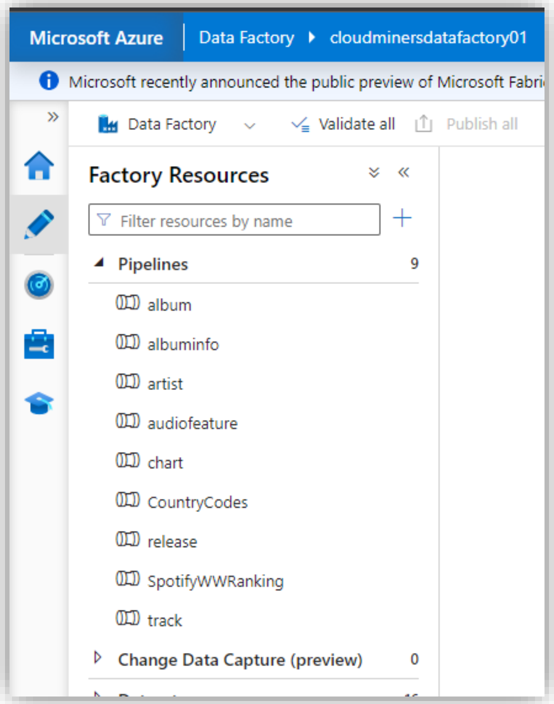
*

Microsoft Azure Data Lake

Data Ingestion to Azure SQL

- Data stored Azure data lake storage is ingested into Azure SQL using Azure Data Factory

Azure
Data Factory



Microsoft Azure | Data Factory ▸ cloudminersdatafactory01

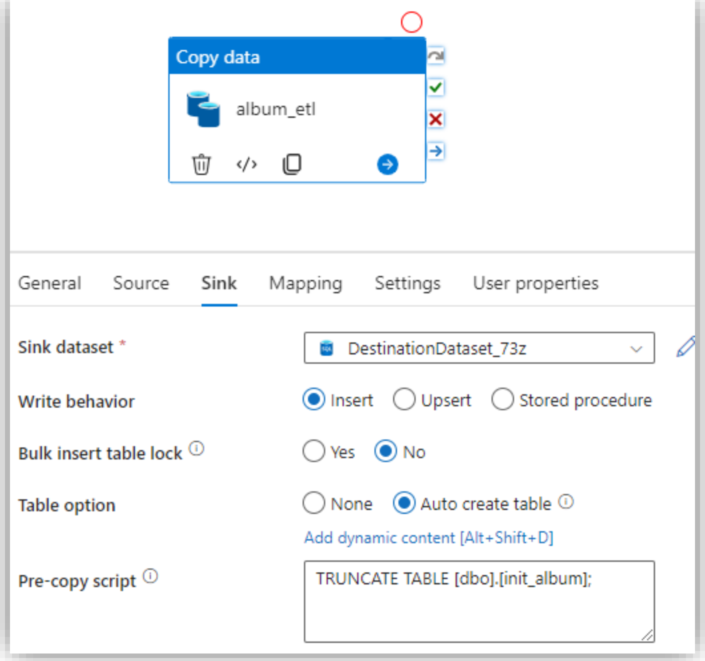
ⓘ Microsoft recently announced the public preview of Microsoft Fabri

Data Factory ⌄ Validate all Publish all

Factory Resources

Filter resources by name

- Pipelines 9
  - album
  - albuminfo
  - artist
  - audiofeature
  - chart
  - CountryCodes
  - release
  - SpotifyWWRanking
  - track
- Change Data Capture (preview) 0



Copy data

album_etl

General | Source | Sink | Mapping | Settings | User properties

Sink dataset * DestinationDataset_73z

Write behavior ● Insert ○ Upsert ○ Stored procedure

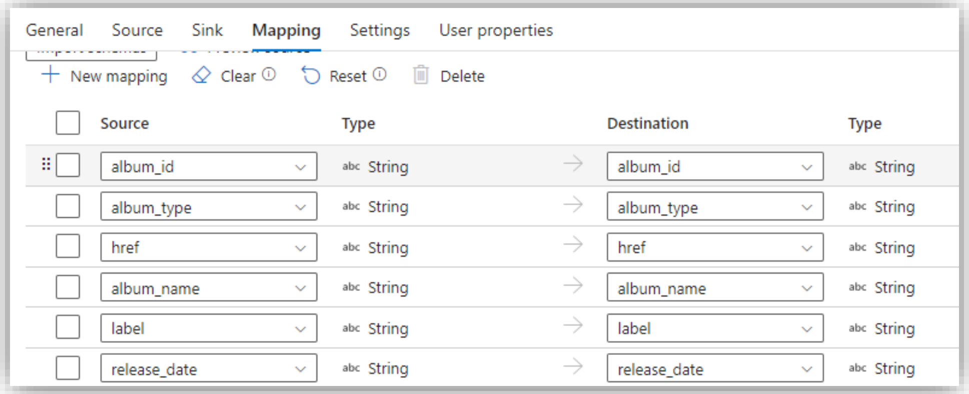
Bulk insert table lock ⓘ ○ Yes ● No

Table option ○ None ● Auto create table ⓘ

Add dynamic content [Alt+Shift+D]

Pre-copy script ⓘ

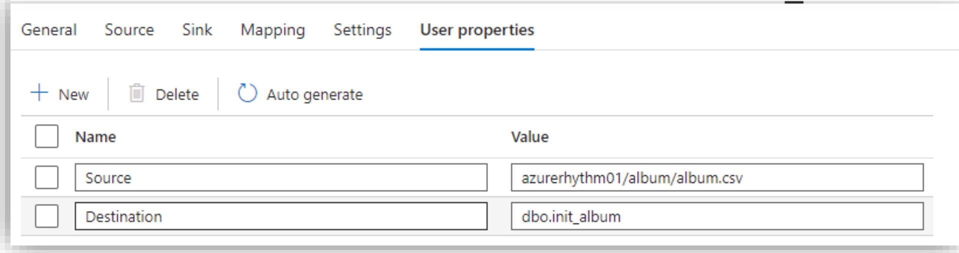
```
TRUNCATE TABLE [dbo].[init_album];
```



General | Source | Sink | Mapping | Settings | User properties

+ New mapping | Clear ⓘ | Reset ⓘ | Delete

	Source	Type		Destination	Type
□	album_id	abc String	→	album_id	abc String
□	album_type	abc String	→	album_type	abc String
□	href	abc String	→	href	abc String
□	album_name	abc String	→	album_name	abc String
□	label	abc String	→	label	abc String
□	release_date	abc String	→	release_date	abc String



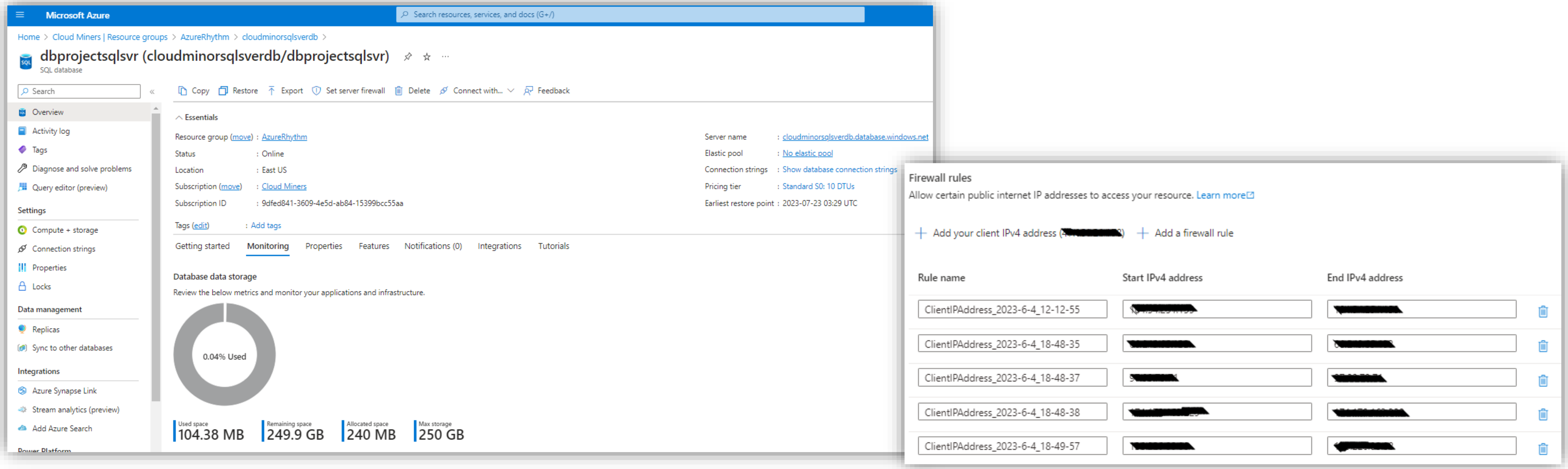
General | Source | Sink | Mapping | Settings | User properties

+ New | Delete | Auto generate

	Name	Value
□	Source	azurerhythm01/album/album.csv
□	Destination	dbo.init_album

Azure SQL Setup

- Created a sql server and sql database
- Setup firewall rules to allow our machines to access the sql server using Azure data studio and R.



Home > Cloud Miners | Resource groups > AzureRhythm > cloudminorsqlsverdb >

dbprojectsqlsvr (cloudminorsqlsverdb/dbprojectsqlsvr)

SQL database

- Overview
- Activity log
- Tags
- Diagnose and solve problems
- Query editor (preview)

Settings

- Compute + storage
- Connection strings
- Properties
- Locks

Data management

- Replicas
- Sync to other databases

Integrations

- Azure Synapse Link
- Stream analytics (preview)
- Add Azure Search

Power Platform

Essentials

Resource group (move)	: AzureRhythm
Status	: Online
Location	: East US
Subscription (move)	: Cloud Miners
Subscription ID	: 9dfed841-3609-4e5d-ab84-15399bcc55aa
Tags (edit)	: Add tags

Server name	: cloudminorsqlsverdb.database.windows.net
Elastic pool	: No elastic pool
Connection strings	: Show database connection strings
Pricing tier	: Standard S0: 10 DTUs
Earliest restore point	: 2023-07-23 03:29 UTC

Database data storage

Review the below metrics and monitor your applications and infrastructure.

Used space	Remaining space	Allocated space	Max storage
104.38 MB	249.9 GB	240 MB	250 GB

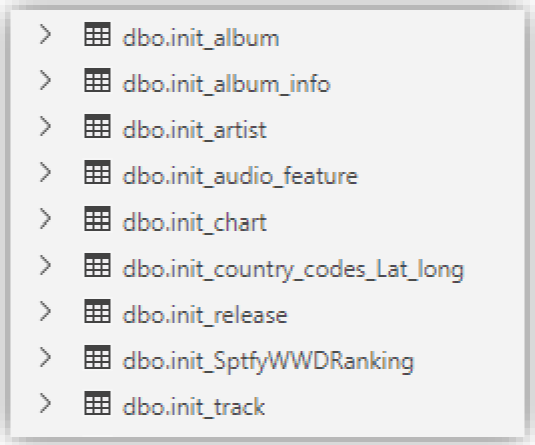
Firewall rules

Allow certain public internet IP addresses to access your resource. Learn more

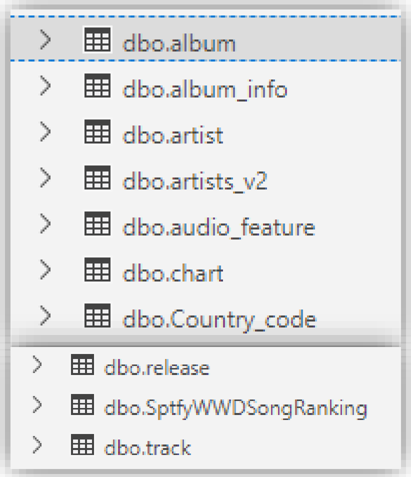
Rule name	Start IPv4 address	End IPv4 address
ClientIPAddress_2023-6-4_12-12-55	[redacted]	[redacted]
ClientIPAddress_2023-6-4_18-48-35	[redacted]	[redacted]
ClientIPAddress_2023-6-4_18-48-37	[redacted]	[redacted]
ClientIPAddress_2023-6-4_18-48-38	[redacted]	[redacted]
ClientIPAddress_2023-6-4_18-49-57	[redacted]	[redacted]

Relational Database Tables using Azure SQ

- For data ingested using Azure Data Factory, the landing tables are named as "init_<file_name>.
- Relational tables are created with key constraints and appropriate data types as per the ER Diagram.



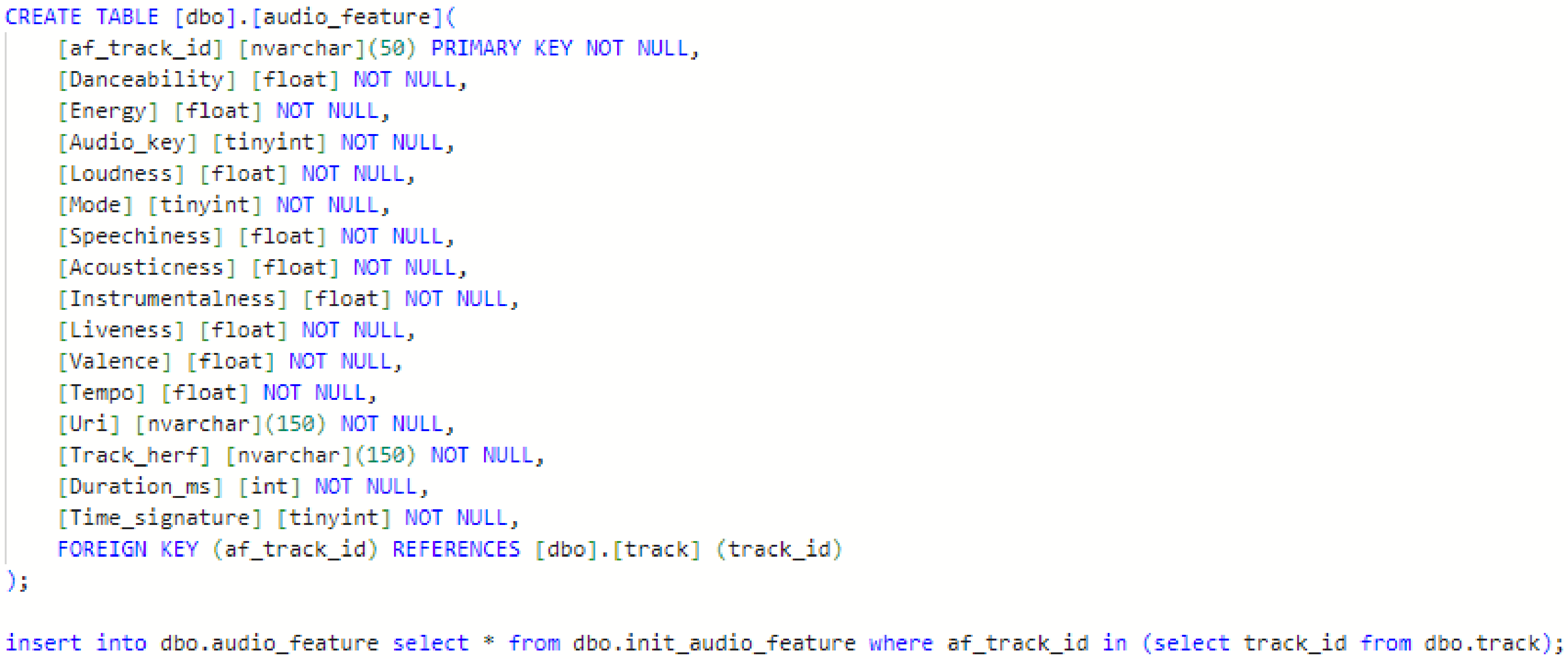
- dbo.init_album
- dbo.init_album_info
- dbo.init_artist
- dbo.init_audio_feature
- dbo.init_chart
- dbo.init_country_codes_Lat_long
- dbo.init_release
- dbo.init_SptfyWWDRanking
- dbo.init_track



- dbo.album
- dbo.album_info
- dbo.artist
- dbo.artists_v2
- dbo.audio_feature
- dbo.chart
- dbo.Country_code
- dbo.release
- dbo.SptfyWWDSongRanking
- dbo.track

Relational Table Creation and Data Insertion

- Table are created with Primary Keys and Foreign keys using create table statements.
- Data is inserted into the table from corresponding landing table with appropriate foreign key table lookup.



```
CREATE TABLE [dbo].[audio_feature](
    [af_track_id] [nvarchar](50) PRIMARY KEY NOT NULL,
    [Danceability] [float] NOT NULL,
    [Energy] [float] NOT NULL,
    [Audio_key] [tinyint] NOT NULL,
    [Loudness] [float] NOT NULL,
    [Mode] [tinyint] NOT NULL,
    [Speechiness] [float] NOT NULL,
    [Acousticness] [float] NOT NULL,
    [Instrumentalness] [float] NOT NULL,
    [Liveness] [float] NOT NULL,
    [Valence] [float] NOT NULL,
    [Tempo] [float] NOT NULL,
    [Uri] [nvarchar](150) NOT NULL,
    [Track_herf] [nvarchar](150) NOT NULL,
    [Duration_ms] [int] NOT NULL,
    [Time_signature] [tinyint] NOT NULL,
    FOREIGN KEY (af_track_id) REFERENCES [dbo].[track] (track_id)
);

insert into dbo.audio_feature select * from dbo.init_audio_feature where af_track_id in (select track_id from dbo.track);
```

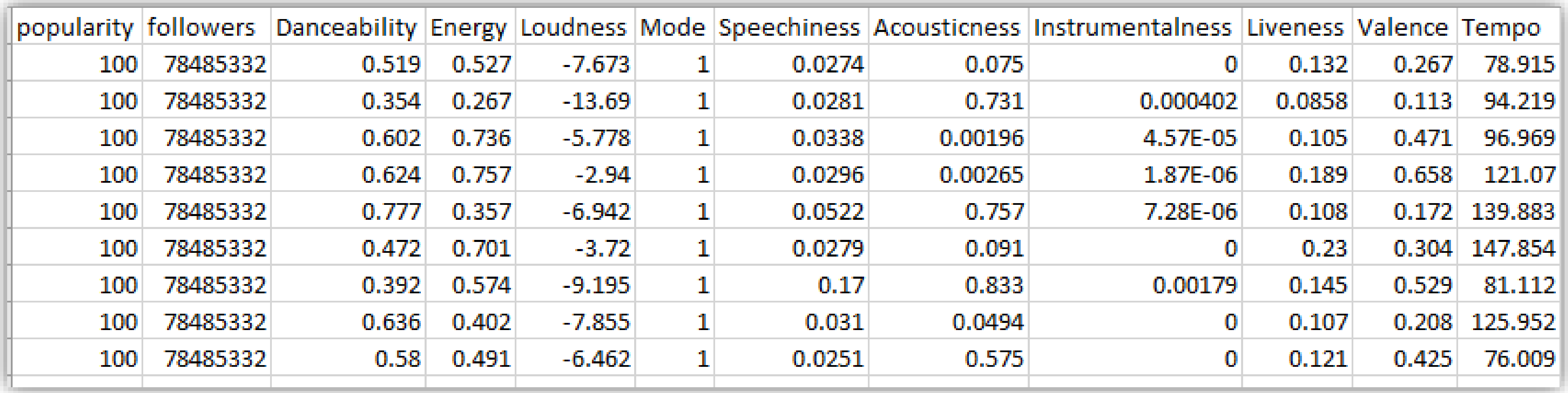
MACHINE LEARNING
THE FORMULA FOR SUCCESS!

Todd Garner

Why Azure Machine Learning?

Introduction and Problem Statement

- 11 independent variables from Spotify, one response variable (popularity).

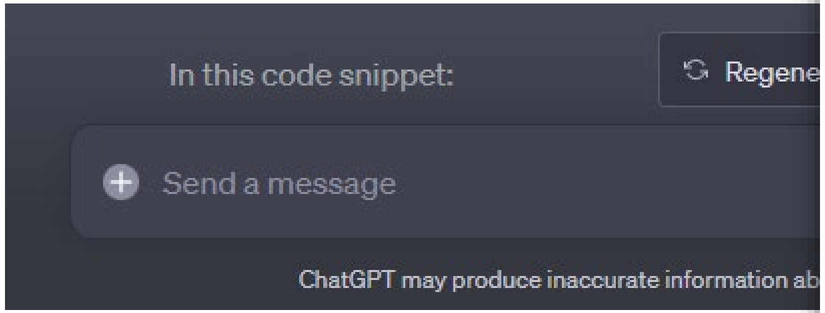


popularity	followers	Danceability	Energy	Loudness	Mode	Speechiness	Acousticness	Instrumentalness	Liveness	Valence	Tempo
100	78485332	0.519	0.527	-7.673	1	0.0274	0.075	0	0.132	0.267	78.915
100	78485332	0.354	0.267	-13.69	1	0.0281	0.731	0.000402	0.0858	0.113	94.219
100	78485332	0.602	0.736	-5.778	1	0.0338	0.00196	4.57E-05	0.105	0.471	96.969
100	78485332	0.624	0.757	-2.94	1	0.0296	0.00265	1.87E-06	0.189	0.658	121.07
100	78485332	0.777	0.357	-6.942	1	0.0522	0.757	7.28E-06	0.108	0.172	139.883
100	78485332	0.472	0.701	-3.72	1	0.0279	0.091	0	0.23	0.304	147.854
100	78485332	0.392	0.574	-9.195	1	0.17	0.833	0.00179	0.145	0.529	81.112
100	78485332	0.636	0.402	-7.855	1	0.031	0.0494	0	0.107	0.208	125.952
100	78485332	0.58	0.491	-6.462	1	0.0251	0.575	0	0.121	0.425	76.009

Methodology and Solution

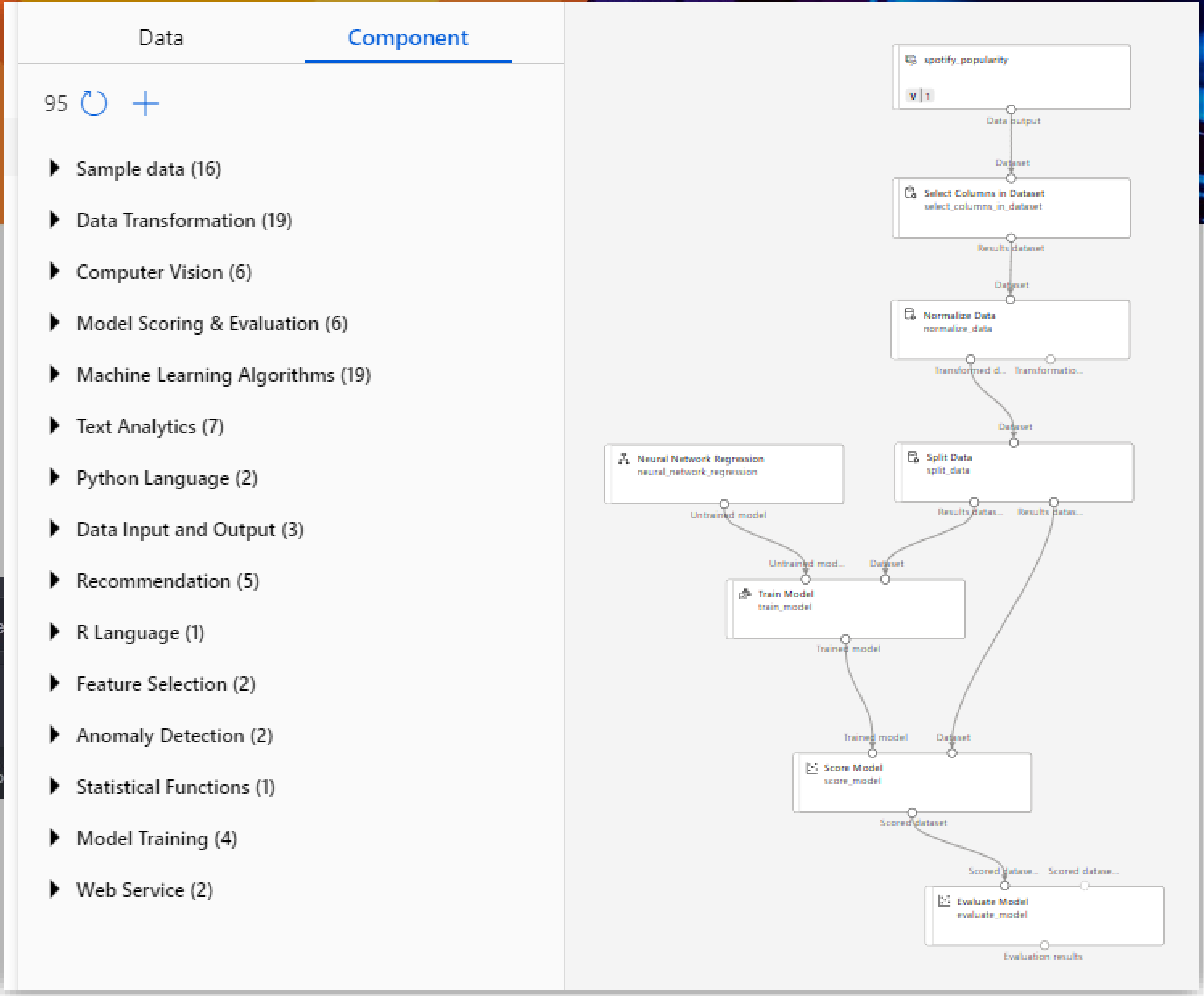
Derived Linear Regression Equation

Popularity = 61.3331 + ($3.93X10^{-7}$)×**Followers** + (11.5315)×**Danceability** – (0.0847)×**Loudness** – (1.5251)×**Mode** + (12.5477)×**Speechiness** – (2.2467)×**Acousticness** + (0.1258)×**Liveness** – (6.2922)×**Valence** + (0.0101)×**Tempo**



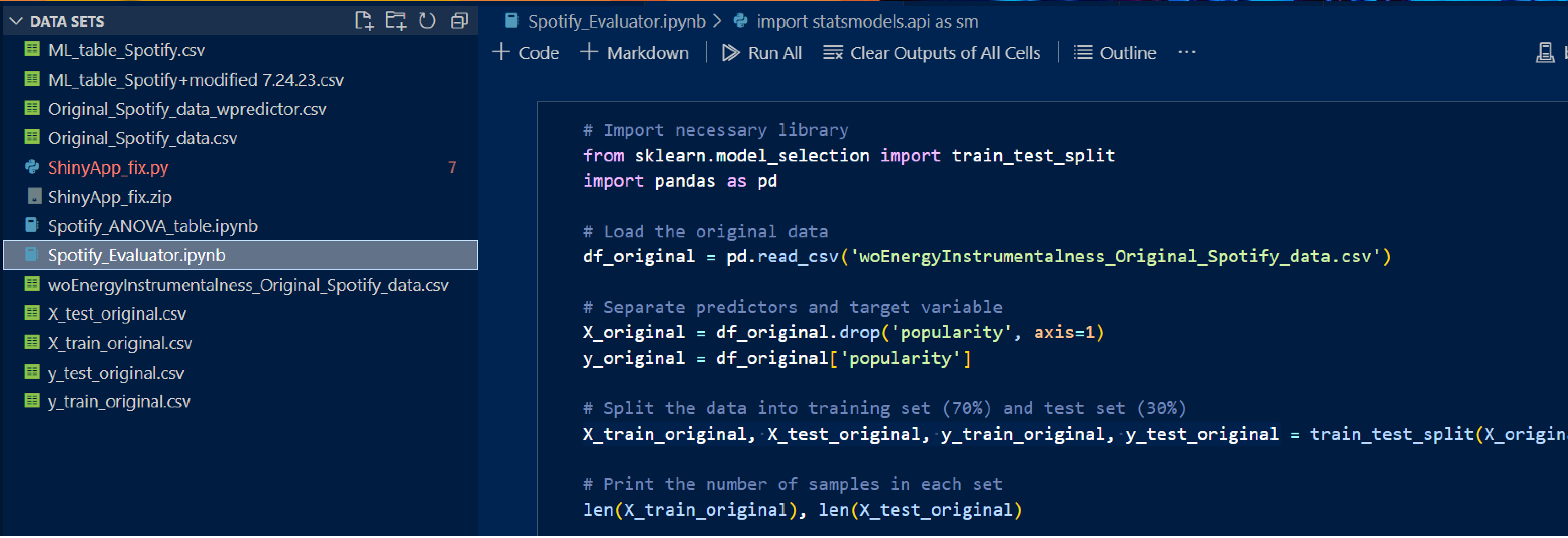
In this code snippet:

ChatGPT may produce inaccurate information ab



- Sample data (16)
- Data Transformation (19)
- Computer Vision (6)
- Model Scoring & Evaluation (6)
- Machine Learning Algorithms (19)
- Text Analytics (7)
- Python Language (2)
- Data Input and Output (3)
- Recommendation (5)
- R Language (1)
- Feature Selection (2)
- Anomaly Detection (2)
- Statistical Functions (1)
- Model Training (4)
- Web Service (2)

VS Code – Python – Linear Regression


DATA SETS
ML_table_Spotify.csv
ML_table_Spotify+modified 7.24.23.csv
Original_Spotify_data_wpredictor.csv
Original_Spotify_data.csv
ShinyApp_fix.py 7
ShinyApp_fix.zip
Spotify_ANOVA_table.ipynb
Spotify_Evaluator.ipynb
woEnergyInstrumentalness_Original_Spotify_data.csv
X_test_original.csv
X_train_original.csv
y_test_original.csv
y_train_original.csv
Spotify_Evaluator.ipynb > import statsmodels.api as sm
+ Code
+ Markdown
Run All
Clear Outputs of All Cells
Outline
···

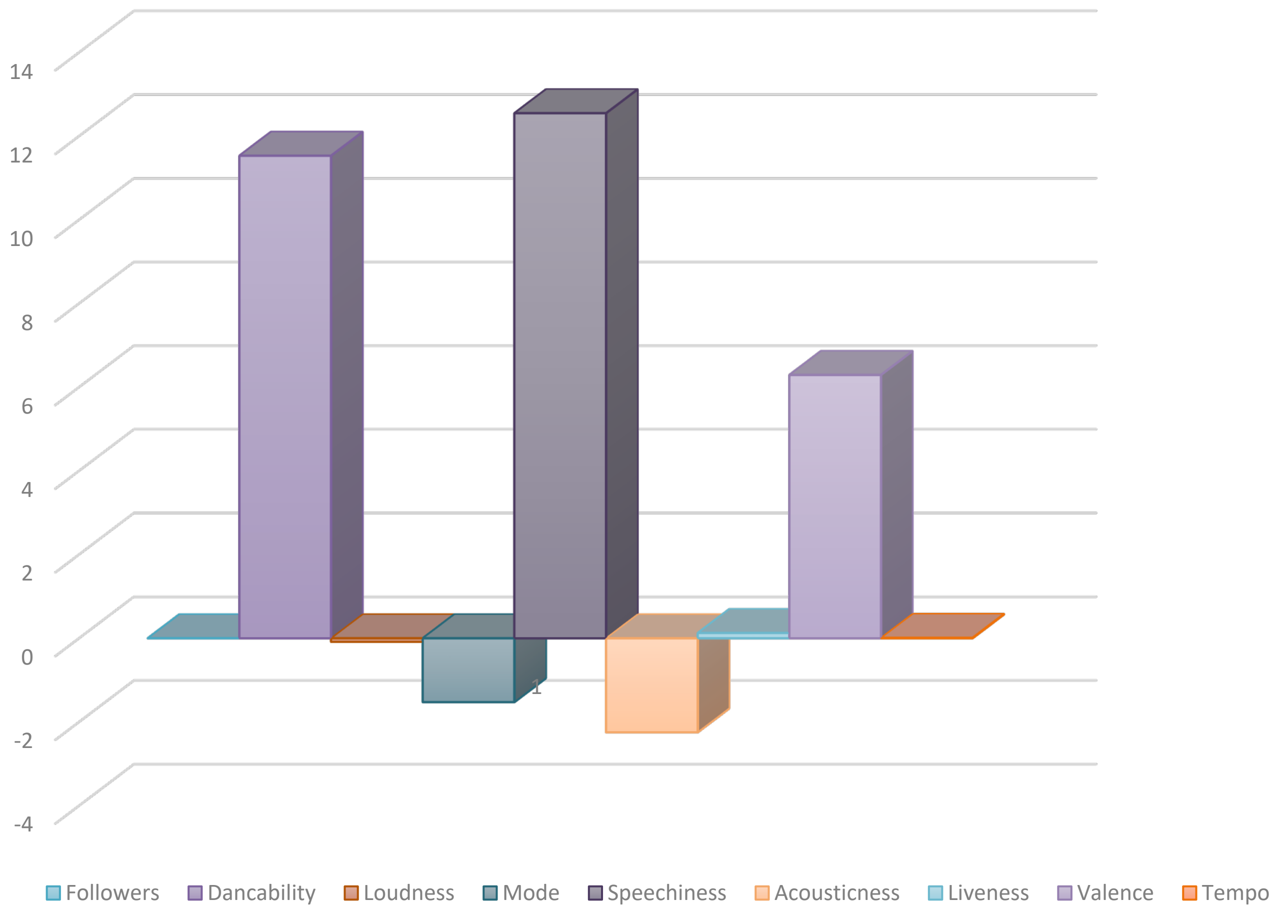
```python
# Import necessary library
from sklearn.model_selection import train_test_split
import pandas as pd

# Load the original data
df_original = pd.read_csv('woEnergyInstrumentalness_Original_Spotify_data.csv')

# Separate predictors and target variable
X_original = df_original.drop('popularity', axis=1)
y_original = df_original['popularity']

# Split the data into training set (70%) and test set (30%)
X_train_original, X_test_original, y_train_original, y_test_original = train_test_split(X_origin

# Print the number of samples in each set
len(X_train_original), len(X_test_original)
```

Finally, I used VS Code and wrote a Python script and obtained a solution

Results and Application

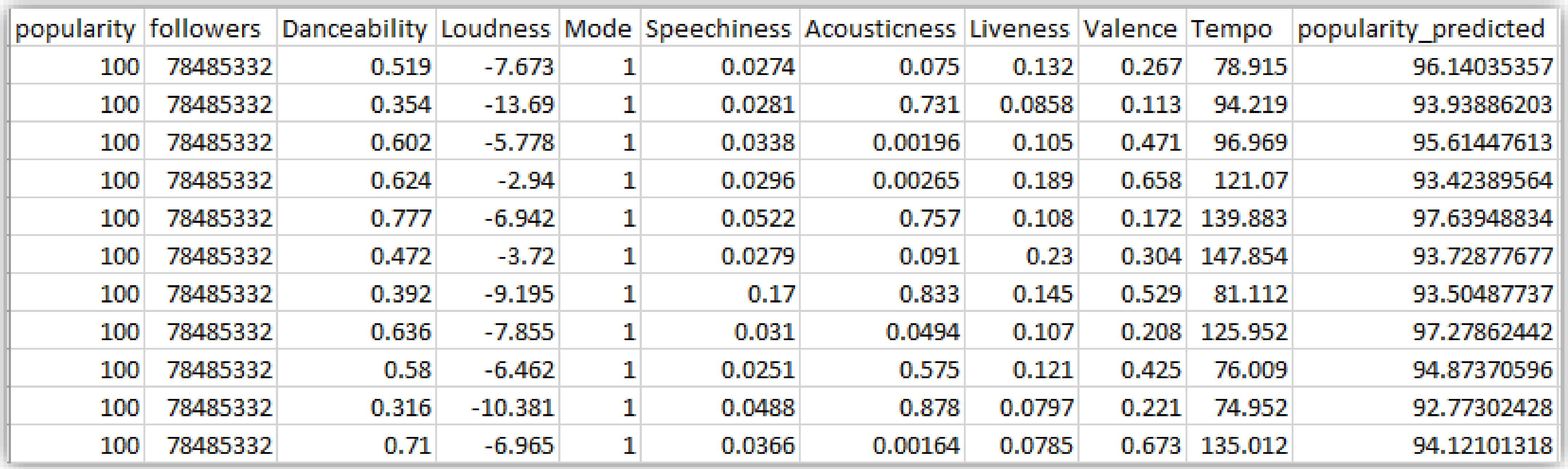
Popularity	
3.93E-07	Followers
11.5315	Dancability
-0.0847	Loudness
-1.5251	Mode
12.5477	Speechiness
-2.2467	Acousticness
0.1258	Liveness
6.2922	Valence
0.0101	Tempo

Popularity Equation parameter weights



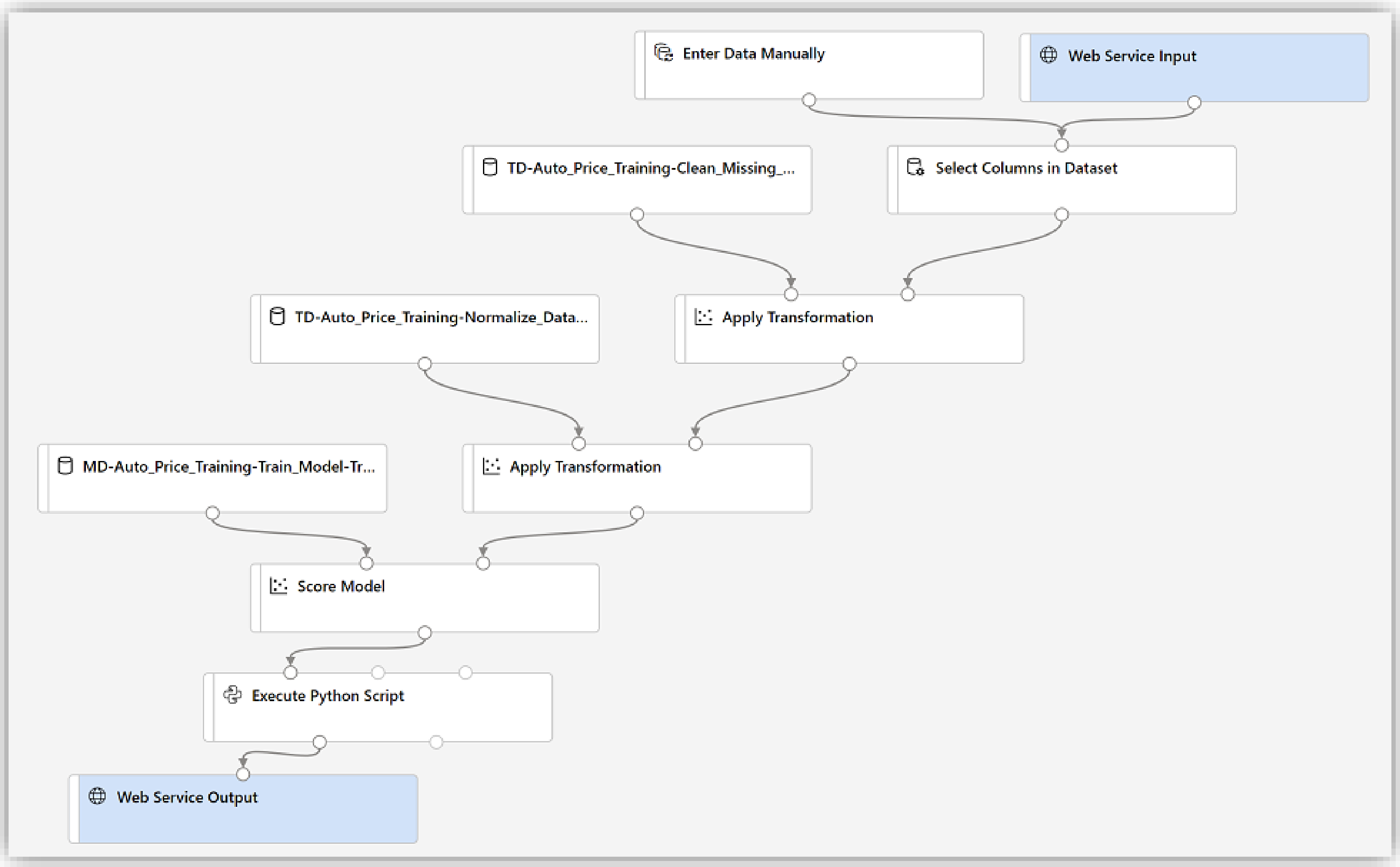
Introduction and Problem Statement

- 9 independent variables from Spotify, one response variable (popularity), plus predicted popularity.



popularity	followers	Danceability	Loudness	Mode	Speechiness	Acousticness	Liveness	Valence	Tempo	popularity_predicted
100	78485332	0.519	-7.673	1	0.0274	0.075	0.132	0.267	78.915	96.14035357
100	78485332	0.354	-13.69	1	0.0281	0.731	0.0858	0.113	94.219	93.93886203
100	78485332	0.602	-5.778	1	0.0338	0.00196	0.105	0.471	96.969	95.61447613
100	78485332	0.624	-2.94	1	0.0296	0.00265	0.189	0.658	121.07	93.42389564
100	78485332	0.777	-6.942	1	0.0522	0.757	0.108	0.172	139.883	97.63948834
100	78485332	0.472	-3.72	1	0.0279	0.091	0.23	0.304	147.854	93.72877677
100	78485332	0.392	-9.195	1	0.17	0.833	0.145	0.529	81.112	93.50487737
100	78485332	0.636	-7.855	1	0.031	0.0494	0.107	0.208	125.952	97.27862442
100	78485332	0.58	-6.462	1	0.0251	0.575	0.121	0.425	76.009	94.87370596
100	78485332	0.316	-10.381	1	0.0488	0.878	0.0797	0.221	74.952	92.77302428
100	78485332	0.71	-6.965	1	0.0366	0.00164	0.0785	0.673	135.012	94.12101318

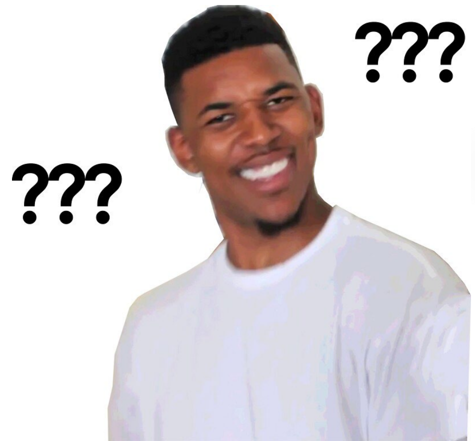
Phase 2: Enhancements


Enter Data Manually
Web Service Input
TD-Auto_Price_Training-Clean_Missing_...
Select Columns in Dataset
TD-Auto_Price_Training-Normalize_Data...
Apply Transformation
MD-Auto_Price_Training-Train_Model-Tr...
Apply Transformation
Score Model
Execute Python Script
Web Service Output

Limitations

Is there a different way to load your data to the database?

Yes! We can use ODBC Driver 17 for Sql Server.



The End!

Links

- Azure Rhythm Application | Landing Page
- Rshiny App Break Down
- Coursera Machine Learning

Contacts

Haitie Liu - haitiel@mail.smu.edu

Mai Dang - maid@mail.smu.edu

Lijo Jacob - lijoj@mail.smu.edu

Todd Garner - toddg@mail.smu.edu

Lani Lewis - lanil@mail.smu.edu

Thank You!