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

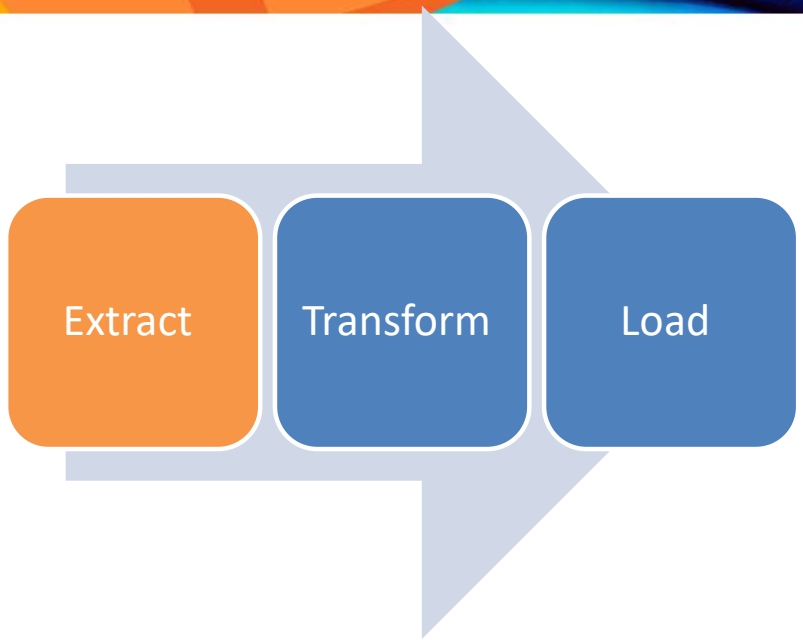
https://azrhythmwebapp.azurewebsites.net/

Extract, Transfer, Load (ETL)

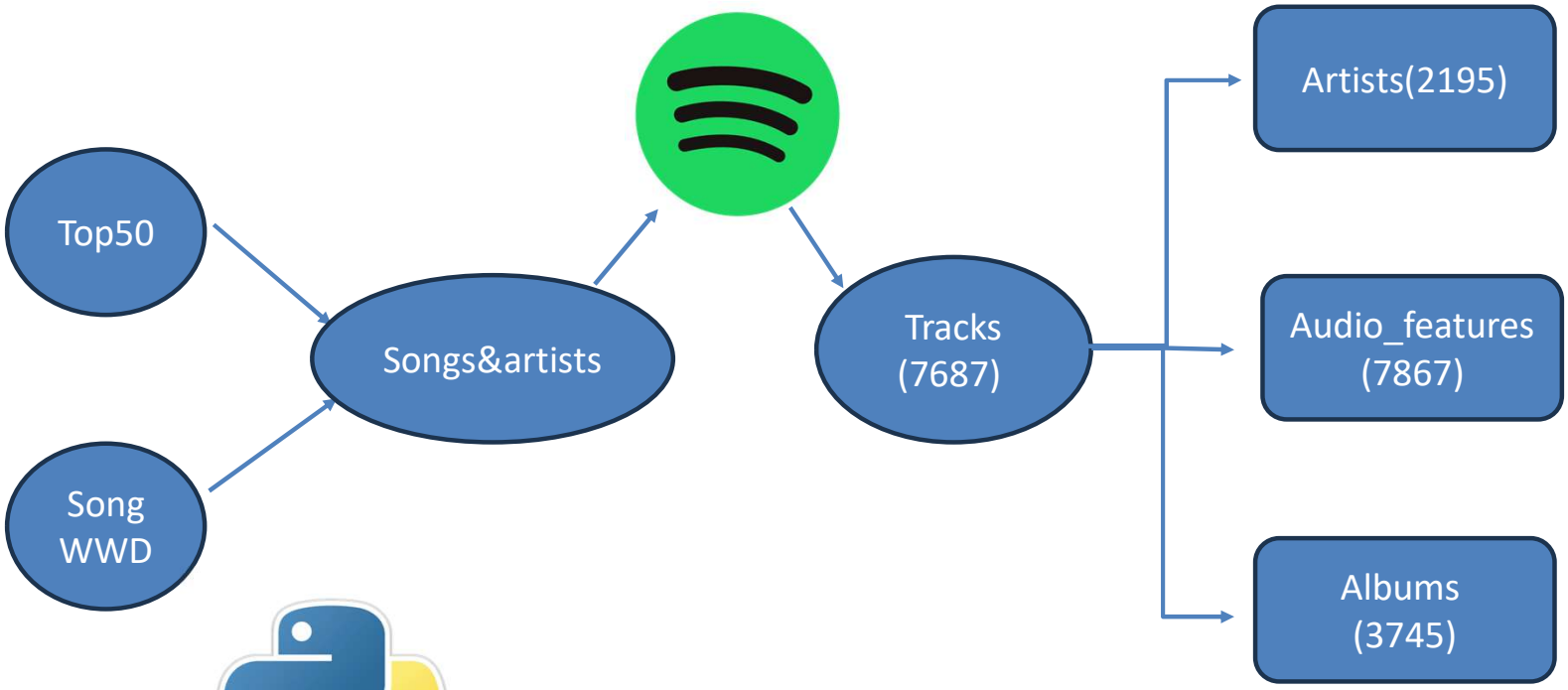
Python

Mai Dang

ETL



Extract

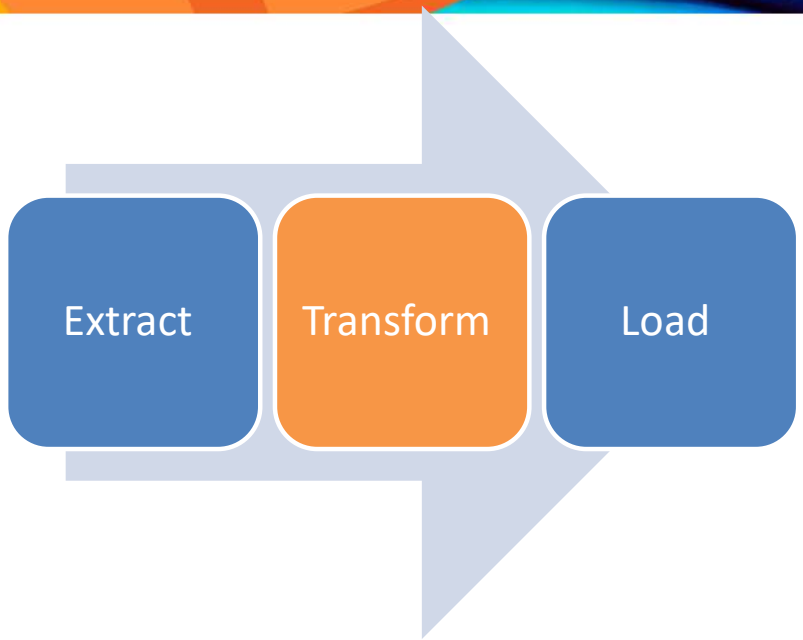


Using library Spotipy

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



Transform

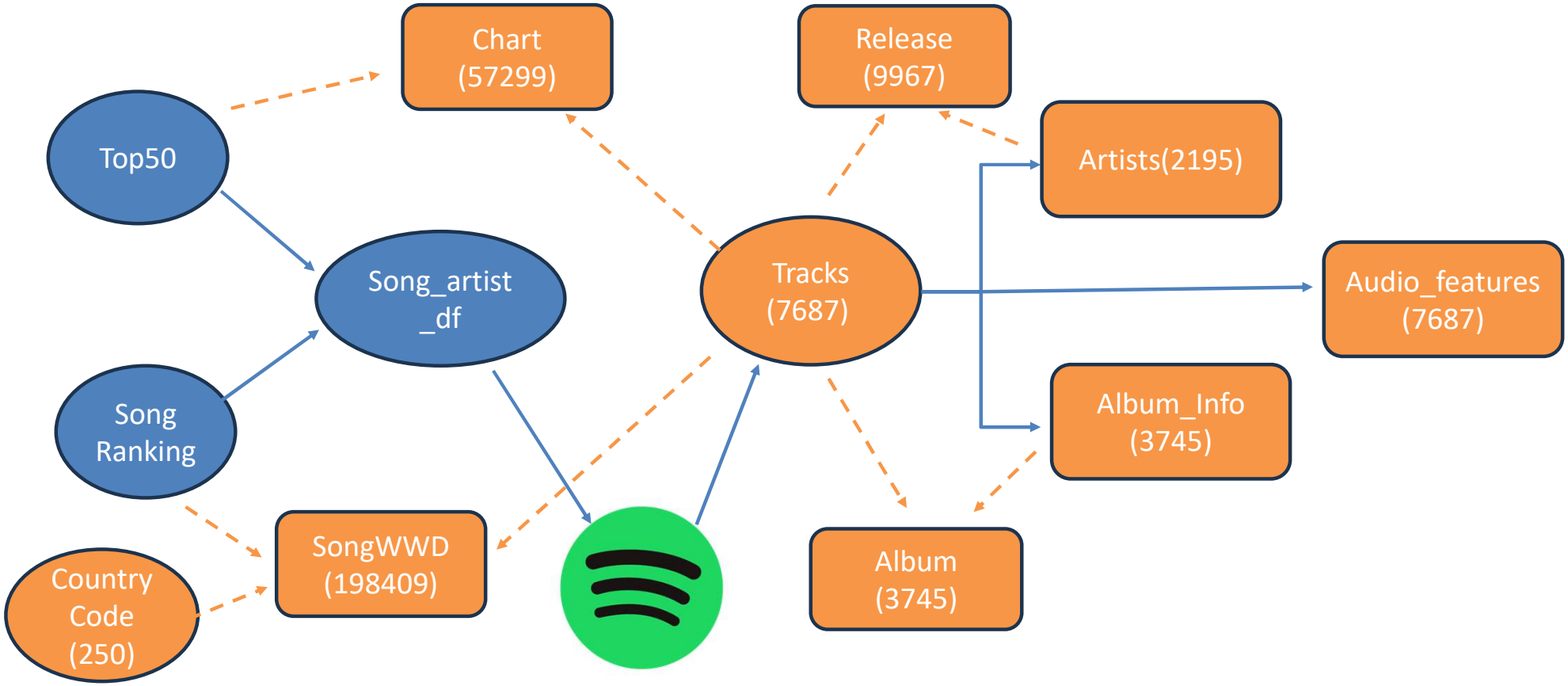
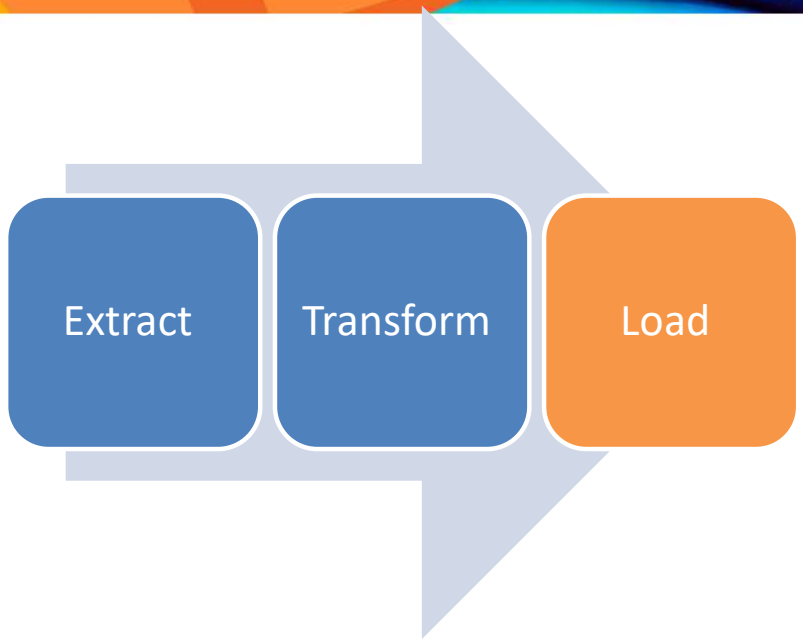
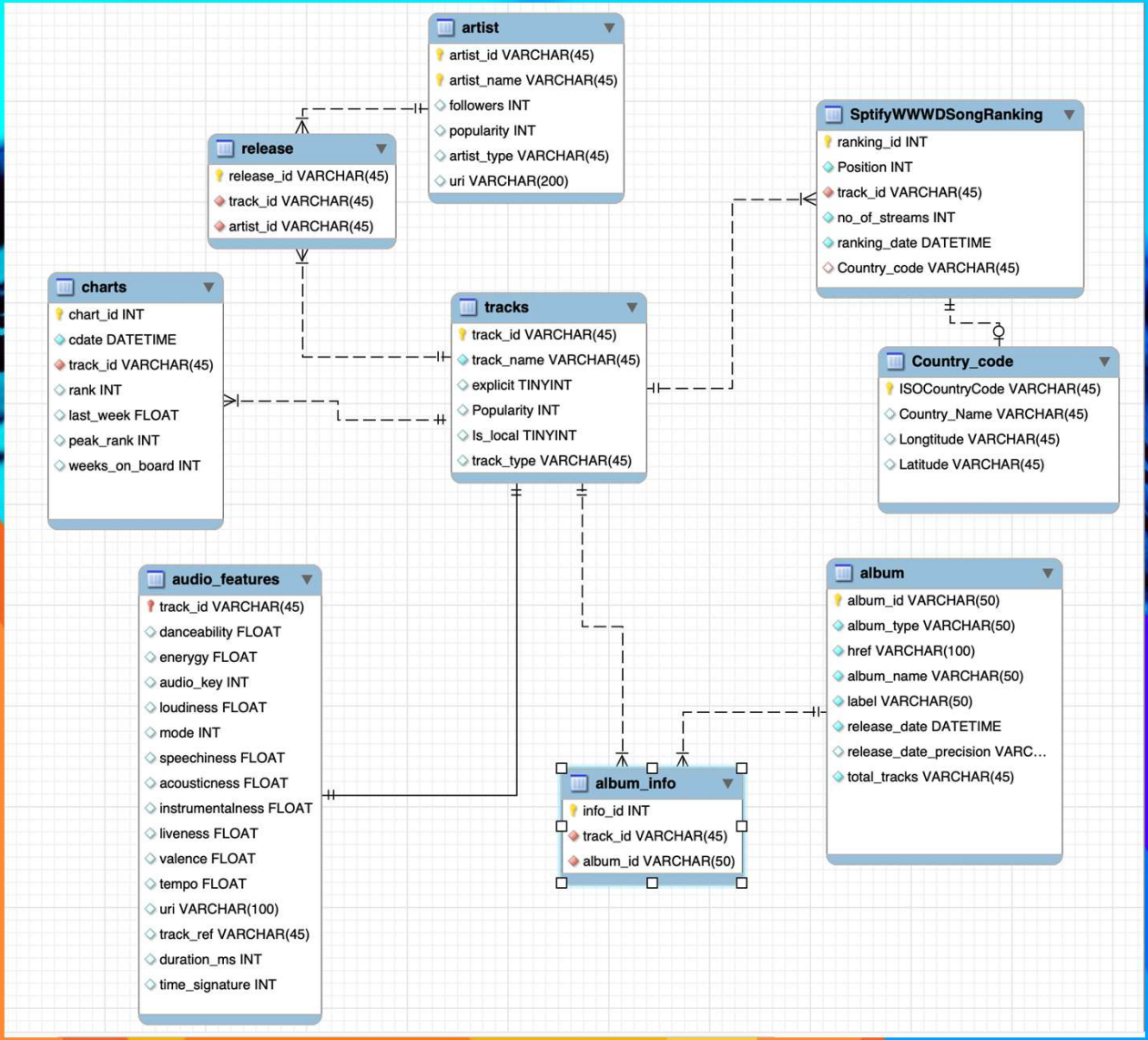


Chart
(57299)
Release
(9967)
Top50
Artists(2195)
Song_artist
_df
Tracks
(7687)
Audio_features
(7687)
Album_Info
(3745)
Song
Ranking
SongWWD
(198409)
Country
Code
(250)
Album
(3745)

ETL

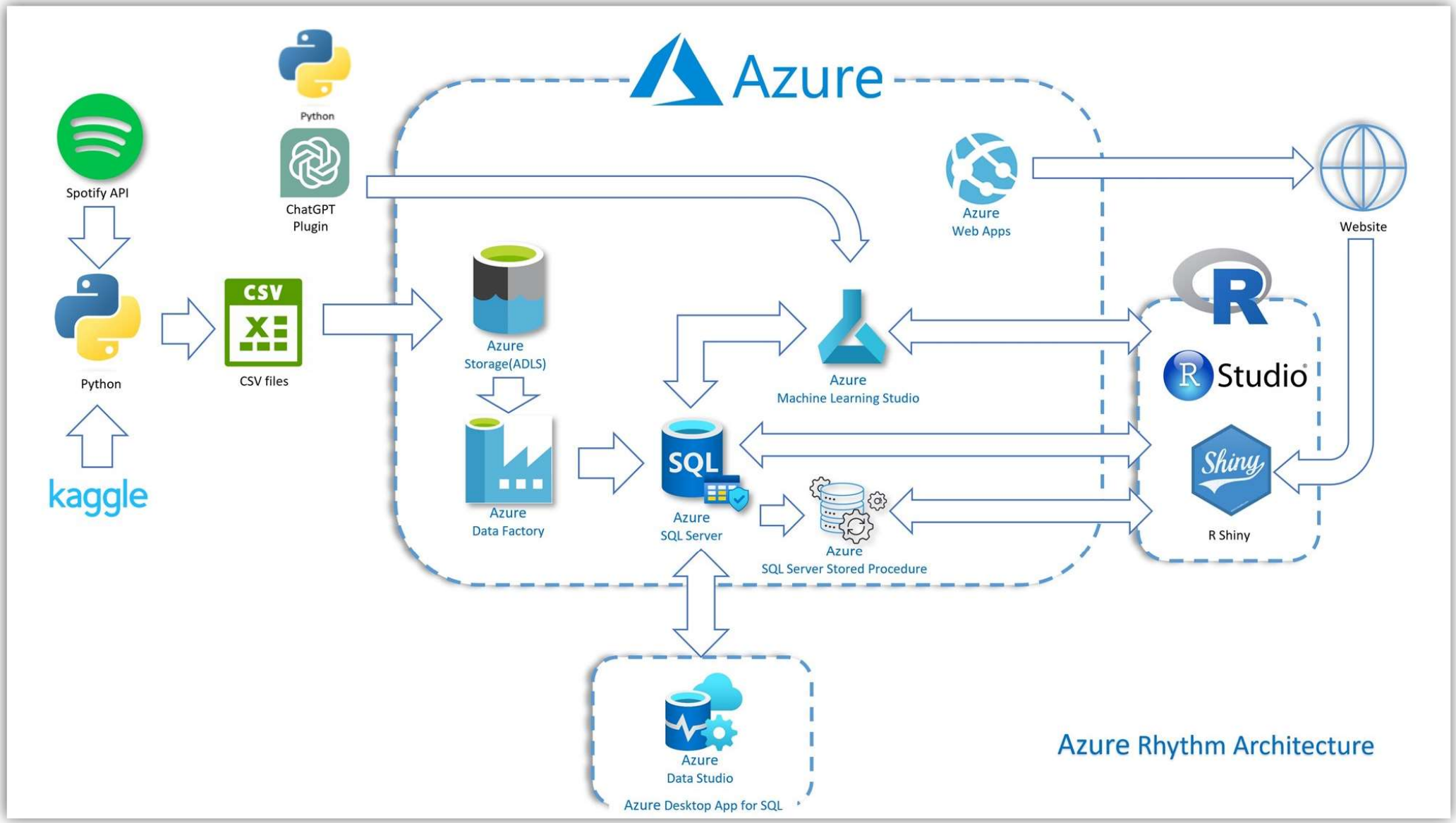


ERD


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

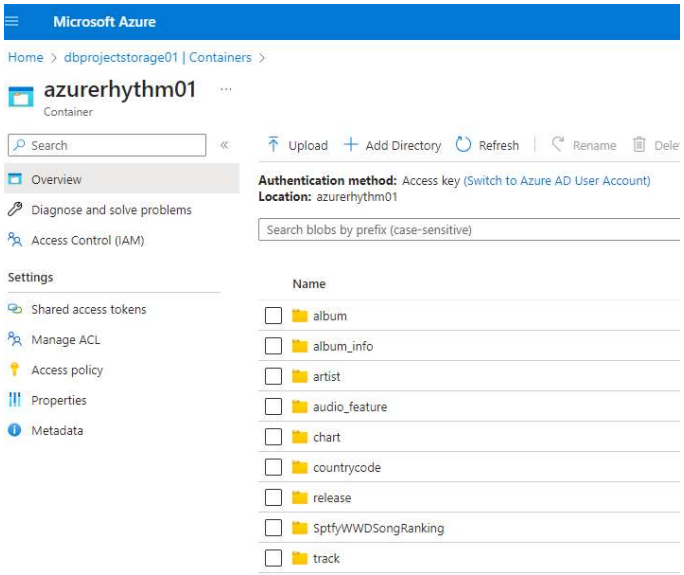
Azure Database

Lijo Jacob


Azure
Spotify API
Python
ChatGPT
Plugin
Python
kaggle
CSV
CSV files
Azure
Storage(ADLS)
Azure
Data Factory
SQL
Azure
SQL Server
Azure
Machine Learning Studio
Azure
SQL Server Stored Procedure
Azure
Web Apps
Website
R
R Studio
Shiny
R Shiny
Azure
Data Studio
Azure Desktop App for SQL
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.



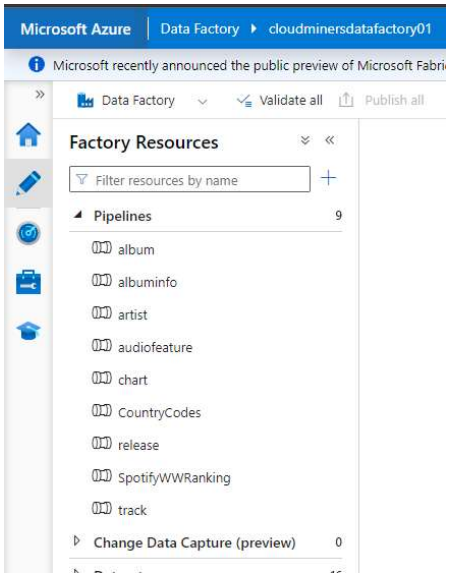
*

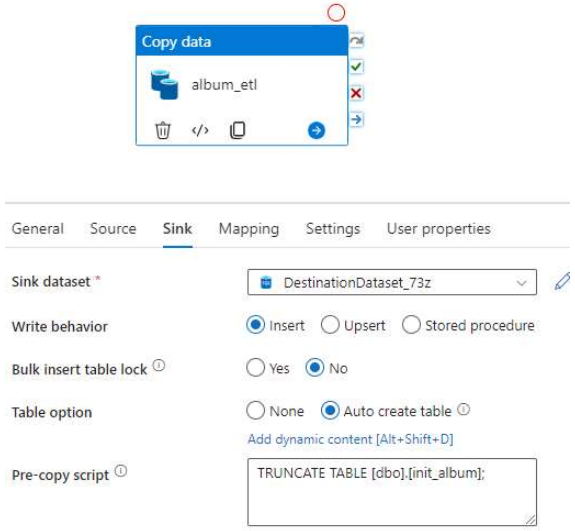
Microsoft Azure
Data Lake

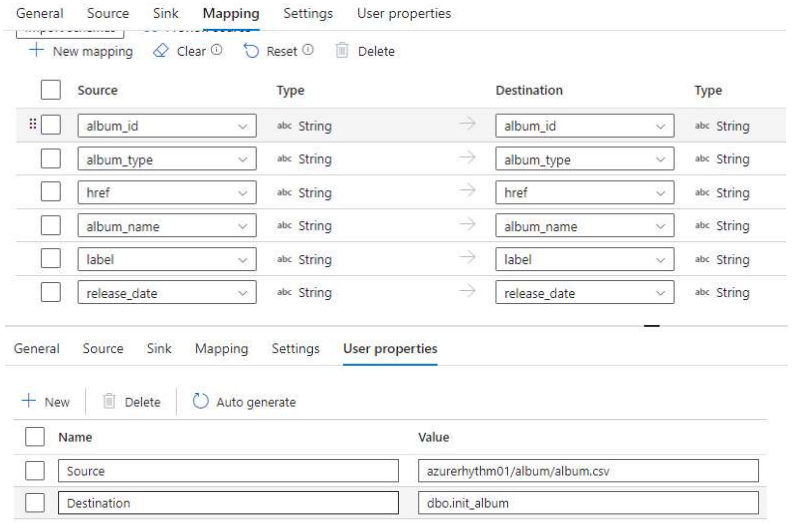
Data Ingestion to Azure SQL

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

Azure Data Factory

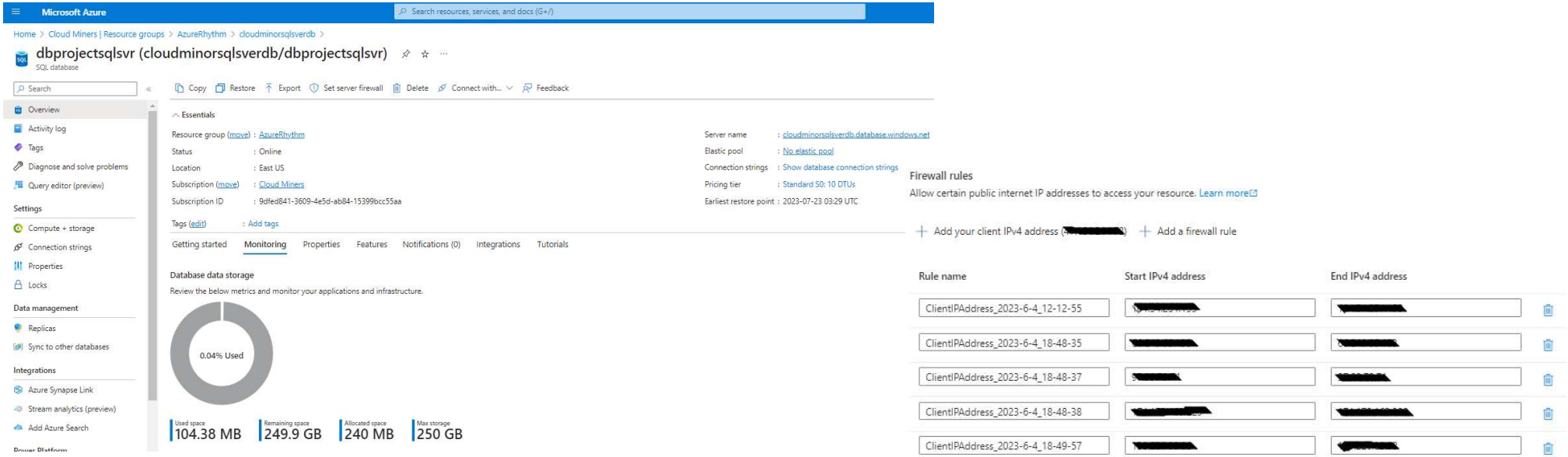






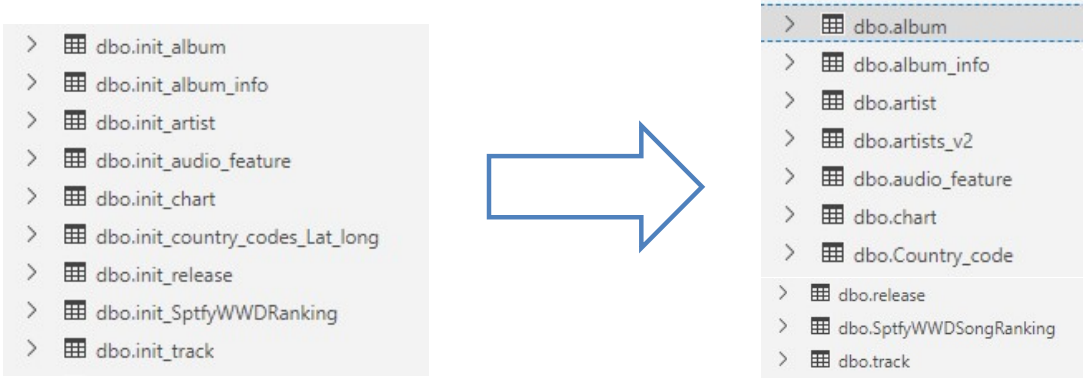
Azure SQL Setup

- 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.



Relational Database Tables using Azure SQL

- 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.



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.

```sql
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

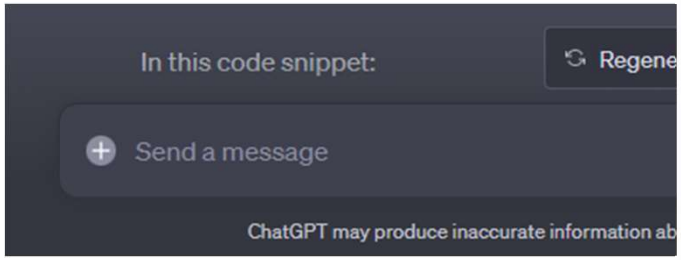
- 11 independent variables from Spotify, one response variable (popularity).

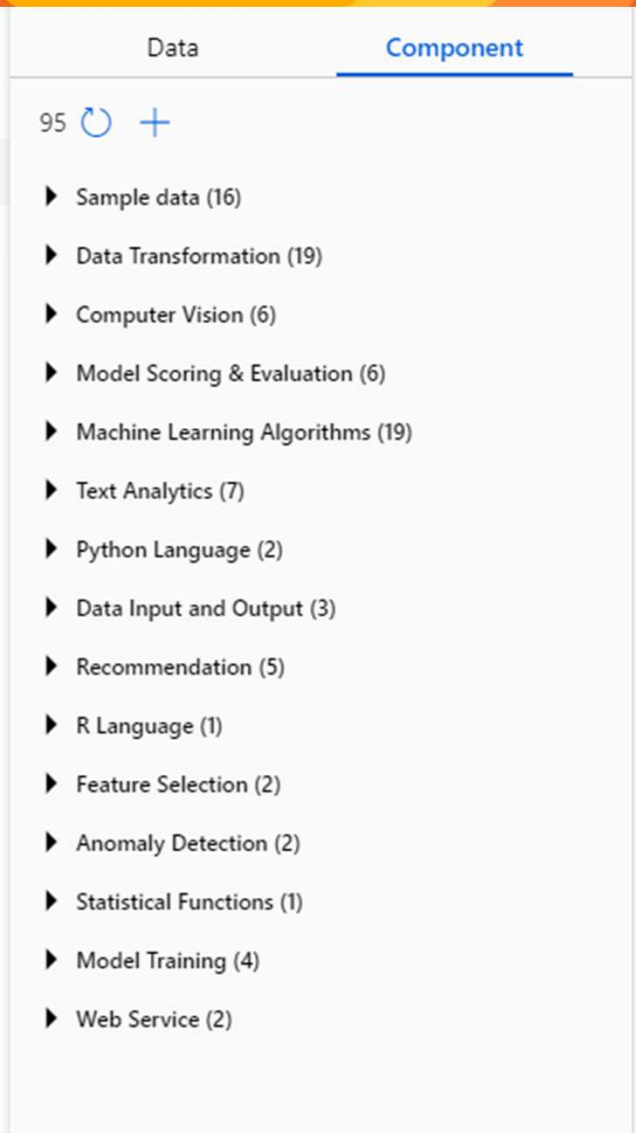
popularity	followers	Danceability	Energy	Loudness	Mode	Speechiness	Acousticness	Instrumentalness	Liveness	Valence	Tempo
100	78485332	0.519	0.527	-7.673	1	0.0274	0.075	0	0.132	0.267	78.915
100	78485332	0.354	0.267	-13.69	1	0.0281	0.731	0.000402	0.0858	0.113	94.219
100	78485332	0.602	0.736	-5.778	1	0.0338	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

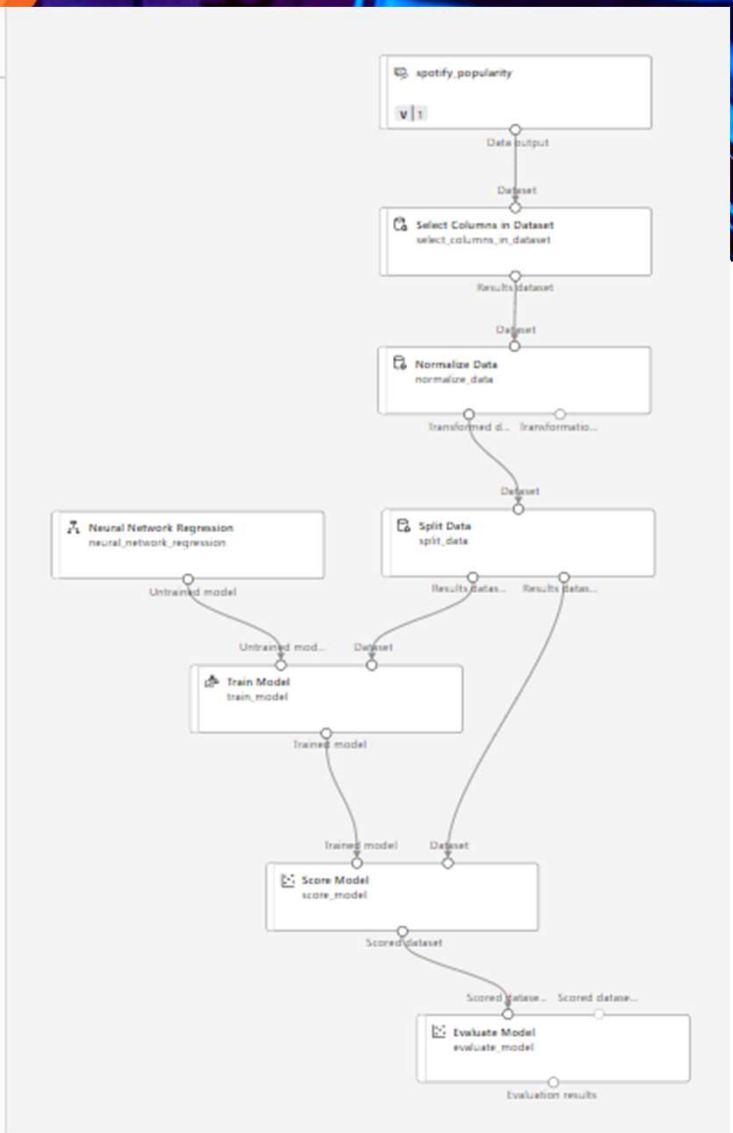
Methodology and Solution

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**







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

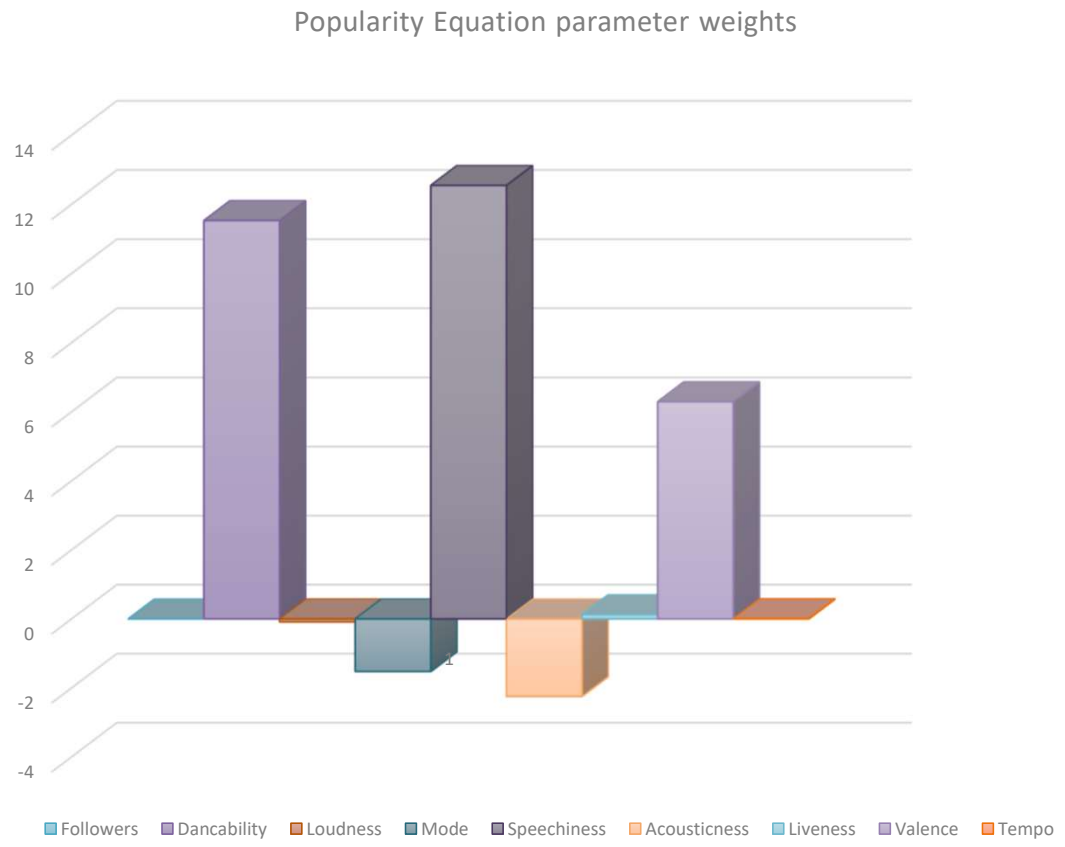
+ Code + Markdown | Run All | Clear Outputs of All Cells | Outline ···

```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

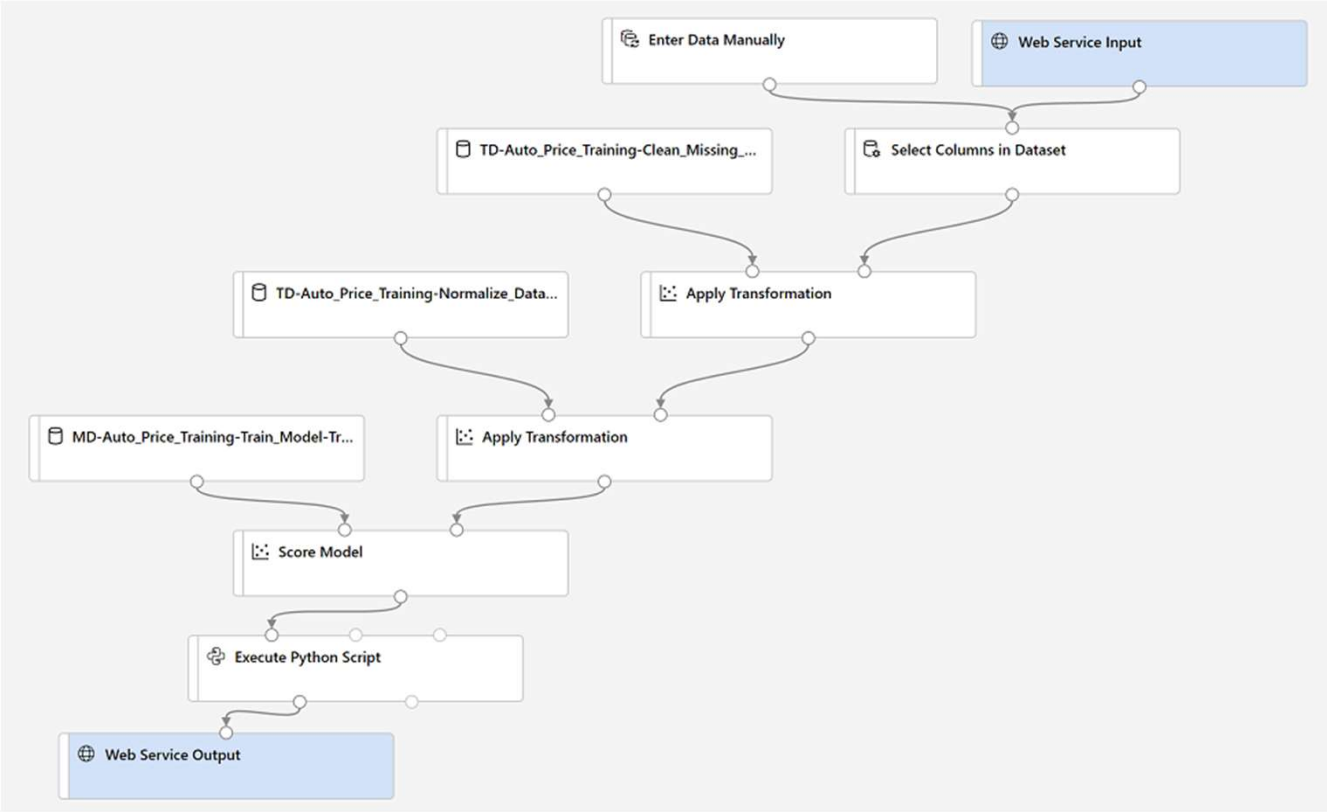


Introduction and Problem Statement

- 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

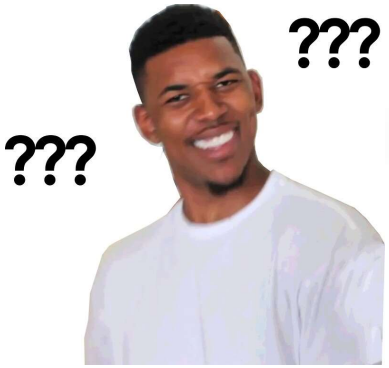
Phase 2: Enhancements



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!