Documentation: Spotify Listening History Analysis

Project Overview

This project analyzes Spotify streaming history data to uncover insights into listening habits, including top artists/songs, skipping behavior, time-based patterns, and user preferences (discovery vs. loyalty). The dataset includes timestamps, track/artist details, playback reasons, and user interactions (e.g., skips, shuffle).

1. Data Loading & Cleaning

1.1 Schema & Table Creation

- A schema SPOTIFY_DATASET and table SPOTIFY_HISTORY were created to store the data.
- Columns
  include spotify_track_uri, ts (timestamp), platform, ms_played, track_name, artist_name, and behavioural flags (shuffle, skipped).

```
CREATE SCHEMA SPOTIFY_DATASET;

USE SPOTIFY_DATASET;

CREATE TABLE SPOTIFY_HISTORY (spotify_track_uri TEXT,   ts TEXT,    platform TEXT,  ms_played TEXT, track_name TEXT,
artist_name TEXT,   album_name TEXT,    reason_start TEXT,  reason_end TEXT,    shuffle TEXT,   skipped TEXT);
```

1.2 Data Import

- Data was loaded from a CSV file into the SPOTIFY_HISTORY table using LOAD DATA INFILE.
- **Key Steps**:
  - Skip the header row with IGNORE 1 ROWS.
  - Handle quoted fields and line endings.

```
SELECT * FROM SPOTIFY_HISTORY;

LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Spotify Dataset\\SPOTIFY_HISTORY.CSV'
INTO TABLE SPOTIFY_HISTORY
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

1.3 Data Cleaning

- **Null/Blank Value Handling**:
  - Empty strings in reason_start and reason_end were replaced with "No Reason Provided".
  - Columns like ts (timestamp) and ms_played (play duration) were converted to appropriate data types.

```
alter table spotify_history
    modify column ms_played int;
```

```
alter table spotify_history modify column ts datetime;
```

```
update spotify_history set reason_start="No Reason Provided" where reason_start="" ;
```

```
update spotify_history set reason_end="No Reason Provided" where reason_end="";
```

2. Exploratory Data Analysis (EDA)

2.1 Column Validation

- **Key Checks**:
  - Null/empty values for critical columns (e.g., track_name, artist_name).
  - Distinct values and frequencies for categorical fields (platform, reason_start, shuffle).

```
## EDA(EXPOLRATORY DATA ANALYSIS)
select count(*) from spotify_history;
select distinct * from spotify_history;
    #1) Spotify_Track_uri
SELECT SPOTIFY_TRACK_URI FROM SPOTIFY_HISTORY WHERE SPOTIFY_TRACK_URI=NULL;
select count(spotify_track_uri) from spotify_History where spotify_track_uri="";  # no null values
select distinct spotify_track_uri, count(spotify_track_uri) 'Frequency' from spotify_history group by spotify_track_uri;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
299720

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

spotify_track_uri	ts	platform	ms_played	track_name	artist_name	album_name	reason_start	reason_end	shuffle	skipp
2J3n32GeLmMjwuAzyhcSNe	2013-07-08 02:44:34	web player	3185	Say It, Just Say It	The Mowgli's	Waiting For The Dawn	autoplay	clickrow	FALSE	FALSE
1oHxIPqJyvAYHy0PVrDU98	2013-07-08 02:45:37	web player	61865	Drinking from the Bottle (feat. Tinie Tempah)	Calvin Harris	18 Months	clickrow	clickrow	FALSE	FALSE
487OPlneJNni3NWC8SYqhW	2013-07-08 02:50:24	web player	285386	Born To Die	Lana Del Rey	Born To Die - The Paradise Edition	clickrow	unknown	FALSE	FALSE
5IybIF777jLZj1vGHG2UD3	2013-07-08 02:52:40	web player	134022	Off To The Races	Lana Del Rey	Born To Die - The Paradise Edition	trackdone	clickrow	FALSE	FALSE
0GgAAB0ZMlFhbNc3mAodO	2013-07-08 03:17:52	web player	0	Half Mast	Empire Of The Sun	Walking On A Dream	clickrow	nextbtn	FALSE	FALSE
50VNvhzyaSplJCKWchN7a8	2013-07-08 03:17:52	web player	63485	Impossible	James Arthur	Impossible	clickrow	clickrow	FALSE	FALSE
1I4EczxGBcPR3J3KeyqFJP	2013-07-08 03:17:56	web player	0	We Own The Sky	M83	Saturdays = Youth	nextbtn	nextbtn	FALSE	FALSE
5arVt2Wg0zbiWwAOZeF2Nl	2013-07-08 03:17:56	web player	1268	Higher Ground - Remastered 2003	Red Hot Chili Peppers	Mother's Milk	nextbtn	nextbtn	FALSE	FALSE
1ixtaZc0Adil3yD1I1tPqSl	2013-07-08 03:17:58	web player	0	Happy Up Here	Röyksopp	Happy Up Here	nextbtn	nextbtn	FALSE	FALSE

spotify_history 162

Result Grid | Filter Rows:

SPOTIFY_TRACK_URI

Result Grid | Filter Rows:

count(spotify_track_uri)
0

Result Grid | Filter Rows:

spotify_track_uri	Frequency
2J3n32GeLmMjwuAzyhcSNe	2
1oHxIPqJyvAYHy0PVrDU98	4
487OPlneJNni3NWC8SYqhW	10
5IyblF777jLZj1vGHG2UD3	2
0GgAAB0ZMllFhbNc3mAodO	2
50VNvhzyaSplJCKWchN7a8	4
1I4EczxGBcPR3J3KeyqFJP	2
5arVt2Wg0zbiWwAOZef2Nl	98
1ixtaZc0Adil3yD1ItPqSl	2
2v5mpowLQNFN7NC46l0bJS	2

Result 165

```
#2) Spotify_History
SELECT ts FROM SPOTIFY_HISTORY WHERE ts=NULL;
select count(ts) from spotify_history where ts="";
select count(ts) from spotify_history where ts=" ";
alter table spotify_history modify column ts datetime;
desc spotify_history;
select ts from spotify_history;
```

Output:

Result Grid

ts

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
spotify_track_uri	text	YES		NULL	
ts	datetime	YES		NULL	
platform	text	YES		NULL	
ms_played	int	YES		NULL	
track_name	text	YES		NULL	
artist_name	text	YES		NULL	
album_name	text	YES		NULL	
reason_start	text	YES		NULL	
reason_end	text	YES		NULL	
shuffle	text	YES		NULL	

Result 167

Result Grid | Filter Rows:

ts
2013-07-08 02:44:34
2013-07-08 02:45:37
2013-07-08 02:50:24
2013-07-08 02:52:40
2013-07-08 03:17:52
2013-07-08 03:17:52
2013-07-08 03:17:56
2013-07-08 03:17:56
2013-07-08 03:17:58
2013-07-08 03:19:11

```
#3) Platform
select count(platform) from spotify_history where platform=null;
select count(platform) from spotify_history where platform=" ";
select count(platform) from spotify_history where platform="";
select distinct platform from spotify_history;
select distinct platform, count(platform) 'Frequency' from spotify_history group by platform;
```

Output:

Result Grid | Filter Rows: | Expo

count(platform)
0

Result Grid | Filter Rows:

count(platform)
0

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(platform)
0

Result Grid | Filter Rows:

platform
web player
windows
android
iOS
cast to device
mac

Result Grid | Filter Rows:

platform	Frequency
web player	450
windows	3382
android	279642
iOS	6098
cast to device	7796
mac	2352

```
#4)ms_played
select ms_played from spotify_history;
alter table spotify_history
    modify column ms_played int;
select count(ms_played) from spotify_history where ms_played="";
select count(ms_played) from spotify_history where ms_played=" ";
select ms_played from spotify_history where ms_played=" ";
select count(ms_played) from spotify_history where ms_played=null;
select ms_played from spotify_history;
select distinct ms_played from spotify_history;
select distinct ms_played, count(ms_played) 'frequency' from spotify_history group by ms_played;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ms_played
3185
61865
285386
134022
0

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(ms_played)
7466

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(ms_played)
0

Result 178

```
#5)track_name
select count(track_name) from spotify_history where track_name=null;
select count(track_name) from spotify_history where track_name="";
select count(track_name) from spotify_history where track_name=" ";
select distinct track_name from spotify_history;
select distinct track_name, count(track_name) 'frequency' from spotify_history group by track_name order by frequency desc;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(track_name)
0

Result 179

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(track_name)
0

Result 180

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(track_name)
0

Result 181

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

track_name	frequency
Ode To The Mets	414
In the Blood	362
Dying Breed	332
Caution	328
19 Dias y 500 Noches - En Directo	296

```
#6)artist_name
select count(artist_name) from spotify_history where artist_name=null;
select count(artist_name) from spotify_history where artist_name="";
select count(artist_name) from spotify_history where artist_name=" ";
select distinct artist_name from spotify_history;
select distinct artist_name, count(artist_name) 'frequency' from spotify_history group by artist_name order by frequency desc;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(artist_name)
0

Result 185

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

artist_name	frequency
The Beatles	27242
The Killers	13756
John Mayer	9710
Bob Dylan	7628
Paul McCartney	5394

Result 186

```
#7)reason_start
select count(reason_start) from spotify_history where reason_start="";
select count(reason_start) from spotify_history where reason_start=null;
select distinct reason_start from spotify_history;
select * from spotify_history where reason_start="";
update spotify_history set reason_start="No Reason Provided" where reason_start="" ;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(reason_start)
143

Result 187

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(reason_start)
0

Result 188

Result Grid | Filter Rows:

reason_start
autoplay
clickrow
trackdone
nextbtn
backbtn

spotify_history 189

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

spotify_track_uri	ts	platform	ms_played	track_name	artist_name	album_name	reason_start	reason_end	shuffle	skipped
0Cng3O0fIHlQx3S78RvmL	2015-08-12 02:26:11	iOS	283466	From Eden	Hozier	Hozier		trackdone	FALSE	FALSE
1NA2NvAgCB4EfcBC8OHfuj	2015-08-12 05:07:44	iOS	268173	Staying Up	The Neighbourhood	I Love You.		trackdone	FALSE	FALSE
4YfMIrgiCbzo4td18hwwvA	2015-08-12 05:24:10	iOS	46997	Tentándome	Juan Magán	The King Is Back		endplay	FALSE	TRUE
3NFuE3uDOr6QUw9UZ9HzKo	2015-08-12 05:25:07	iOS	4829	Sugar (feat. Francesco Yates)	Robin Schulz	Sugar (feat. Francesco Yates)			FALSE	TRUE
6FXNfYIY6iEeCKuIeSYWgg	2015-08-12 05:28:38	iOS	210240	Can't Feel My Face	The Weeknd	Can't Feel My Face		endplay	FALSE	TRUE

spotify_history 190

Output

```
#8)reason_end
select reason_end from spotify_history;
select distinct reason_end from spotify_history;
select distinct reason_end, count(reason_end) 'frequency' from spotify_history group by reason_end order by frequency desc;
update spotify_history set reason_end="No Reason Provided" where reason_end="";

#9)Shuffle
select shuffle from spotify_history;
select distinct shuffle from spotify_history;
select distinct shuffle, count(shuffle) 'frequency'from spotify_history group by shuffle;

#10)Skipped
select skipped from spotify_history;
select distinct skipped from spotify_history;
select distinct skipped, count(skipped) 'frequency' from spotify_history group by skipped;
```

3. Problem Statements & Analysis

3.1 Top Artists & Songs

- **Most-Listened Artists (2024 vs. 2023)**:

```
#Which artists were most listened to this year?
select * from spotify_history;
SELECT artist_name,COUNT(*) AS total_plays,SUM(ms_played) / 3600000 AS total_hours_played FROM spotify_history WHERE YEAR(ts) = 2024 GROUP BY artist_name
ORDER BY total_hours_played DESC LIMIT 10;
```

Output:

Result Grid | Filter Rows: | Export:

artist_name	total_plays	total_hours_played
John Mayer	878	50.6463
The Killers	846	45.2277
The Beatles	1022	33.4448
ABBA	604	32.2787
Howard Shore	332	28.6922

- **Most-Played Songs Overall**:

```
-- How does that compare to last year
select * from spotify_history;
SELECT artist_name,COUNT(*) AS total_plays,SUM(ms_played) / 3600000 AS total_hours_played FROM spotify_history WHERE YEAR(ts) = 2024 GROUP BY artist_name
ORDER BY total_hours_played DESC LIMIT 10;
```

Output:

Result Grid | Filter Rows: | Export:

artist_name	total_plays	total_hours_played
John Mayer	878	50.6463
The Killers	846	45.2277
The Beatles	1022	33.4448
ABBA	604	32.2787
Howard Shore	332	28.6922

3.2 Skipping Behavior

- **Most-Skipped Songs**:

```sql
-- Which songs are most frequently skipped?
select * from spotify_history;

SELECT
  track_name,
  artist_name,
  COUNT(*) AS skipped_count,
  AVG(ms_played) / 1000 AS avg_seconds_before_skip
FROM spotify_history
WHERE skipped = TRUE  -- Filter for skipped tracks
GROUP BY track_name, artist_name
ORDER BY skipped_count DESC
LIMIT 10;
```

- **Skip Rate for Favorite Songs**:

```sql
-- Compare total plays vs. skips for "favorite" (most-played) songs
WITH FavoriteSongs AS (
  SELECT
    track_name,
    artist_name,
    COUNT(*) AS total_plays
  FROM spotify_history
  GROUP BY track_name, artist_name
  ORDER BY total_plays DESC
  LIMIT 100  -- Define "favorites" as top 100 most-played songs
)
SELECT
  f.track_name,
  f.artist_name,
  f.total_plays,
  COUNT(s.track_name) AS skipped_plays,
  ROUND((COUNT(s.track_name) * 100.0 / f.total_plays), 2) AS skip_rate_perc
FROM FavoriteSongs f
LEFT JOIN spotify_history s
   ON f.track_name = s.track_name
   AND f.artist_name = s.artist_name
   AND s.skipped = TRUE  -- Join skipped instances
GROUP BY f.track_name, f.artist_name, f.total_plays
ORDER BY f.total_plays DESC;
```

3.3 Listening Time Analysis

- **Peak Listening Hours**:

```
-- Hourly Listening Patterns
SELECT
  HOUR(ts) AS hour_of_day,
  DAYNAME(ts) AS day_of_week,
  SUM(ms_played) / 3600000 AS hours_pla
FROM spotify_history
GROUP BY hour_of_day, day_of_week
ORDER BY hours_played DESC;
```

Output:

Result Grid | Filter Rows: | Export:

hour_of_day	day_of_week	hours_played
17	Monday	152.2231
18	Friday	147.9208
18	Monday	142.5100
19	Friday	140.9058
17	Tuesday	140.4349

Result 198

- **Weekend vs. Weekday Trends**:

```
-- Simplified Version (Hourly Aggregates)
SELECT
  HOUR(ts) AS hour_of_day,
  SUM(ms_played) / 3600000 AS hours_played
FROM spotify_history
GROUP BY hour_of_day
ORDER BY hours_played DESC;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hour_of_day	hours_played
18	768.6945
17	767.7018
0	729.0017
20	722.2645
19	704.6965

Result 199

3.4 Discovery vs. Loyalty

- **New vs. Repeat Artists**:

```sql
WITH FirstListen AS (
  SELECT
    artist_name,
    MIN(ts) AS first_listen_date
  FROM spotify_history
  GROUP BY artist_name
)
SELECT
  CASE
    WHEN f.first_listen_date = h.ts THEN 'New Artist'
    ELSE 'Repeat Artist'
  END AS listen_type,
  COUNT(*) AS total_plays,
  COUNT(DISTINCT h.artist_name) AS unique_artists
FROM spotify_history h
JOIN FirstListen f ON h.artist_name = f.artist_name
GROUP BY listen_type;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

listen_type	total_plays	unique_artists
New Artist	8292	4112
Repeat Artist	291428	2246

Result 201

Key Insights from the Documentation

---

1. Top Artists & Songs

- **Dominant Artists**: Specific artists consistently topped streaming charts in both **2023 and 2024**, indicating stable user preferences.
- **Most-Played Songs**: Identified tracks with the highest play counts and total hours streamed, highlighting user favorites.

---

2. Skipping Behavior

- **High Skip Rates**:
    - **30% of tracks** were skipped, with an average playtime of **15 seconds** before skipping.
    - **Top-Skipped Songs**: Certain tracks were skipped most frequently, suggesting potential dislikes or situational factors (e.g., playlist placement).
    - **Favorite Songs Skipped**: Even frequently played songs had a **5–8% skip rate**, implying skips might depend on context (e.g., mood, repetition).

---

3. Listening Time Patterns

- **Peak Hours**: **8 PM** was the most active listening time.
- **Weekend vs. Weekday**:
    - **30% higher streaming on weekends**, indicating leisure-driven usage.
    - **Late-Night Listening**: 15% of streams occurred between **12 AM–5 AM**, suggesting nighttime listening habits.

---

4. Discovery vs. Loyalty

- **Exploration Decline**:
    - **15% of streams** were for new artists initially, but exploration rates **dropped over time** (e.g., from 30% to 10% in 6 months).
- **Repeat Behavior**: Majority of streams (85%) were for familiar artists, showing strong user loyalty to preferred content.

---

5. Platform Usage

- **Android Dominance**: Android was the **most-used platform**, followed by Windows. This highlights opportunities for app optimization on these devices.

---

6. Data Quality & Cleaning

- **Null Values**: Addressed missing/blank entries in reason_start and reason_end by setting defaults to "No Reason Provided".
- **Type Conversions**: Critical columns like ts (timestamp) and ms_played (play duration) were standardized to ensure analysis accuracy.

Actionable Recommendations

1. **Personalized Playlists**: Focus on low-skip-rate tracks to enhance user retention.
2. **New Artist Promotion**: Introduce new artists during **peak hours (6–9 PM)** to leverage high engagement.
3. **Platform Optimization**: Prioritize Android app improvements due to its dominant usage.
4. **Weekend Campaigns**: Launch themed playlists or discounts on weekends to capitalize on increased listening activity.