

Spotify Listening Insights Dashboard – Project Documentation

1. Project Overview

This project focuses on analyzing Spotify listening history data to uncover patterns in user engagement, track frequency, album preferences, and listening behaviors across different times and platforms.

The goal was to design a **clean, interactive, and branded Power BI dashboard** that not only provides insights but is also scalable and flexible for future analysis.

Key highlights:

- Data sourced from **Spotify Listening History (CSV format)**
 - Cleaning, transformation, and modeling done using **Power Query & DAX**
 - Designed a **Spotify-inspired UI** with consistent branding
 - Multiple report pages for **overview, listening patterns, and detailed drill-through analysis**
-

2. Data Preparation

2.1 Data Import

- Imported Spotify listening history data (CSV file) into Power BI.
- Removed unnecessary columns: **reason_start** and **reason_end** to reduce model size.

2.2 Data Profiling

- Checked data types for all fields.
- Identified missing values in removed columns.
- Ensured correct formatting for timestamps.

2.3 Creating Date Table

To enable **time intelligence functions**, a dynamic date table was created.

Steps:

1. Split the **ts** (timestamp) column into:

- **Track Played Date**
- **Track Played Time**

2. Created calculated table:

```
Date Table =
CALENDAR(
    MIN(spotify_history[Track Played Date]),
    MAX(spotify_history[Track Played Date])
)
```

3. Added supporting columns:

- **Year** → YEAR('Date Table'[Date])
- **Day Name** → FORMAT('Date Table'[Date], "DDD")
- **Weekday_Weekend** → IF(OR('Date Table'[Day Name] = "Sat", 'Date Table'[Day Name] = "Sun"), "Weekend", "Weekday")
- **Week_Number** → WEEKDAY('Date Table'[Date], 2)

4. Established a **one-to-many relationship** between `Date Table` and `spotify_history`.
-

3. Data Model

- **Fact Table:** `spotify_history`
 - **Dimension Table:** `Date Table`
 - Single direction, one-to-many cardinality
 - Optimized for performance and scalability
-

4. Report Design & UI/UX

- **Theme:** Inspired by Spotify's brand colors & logo
 - **Design Principle:** Group visuals by context for better readability
 - **Containers:** Used for organizing charts logically (Albums, Artists, Tracks)
 - **Navigation:** 3-page structure
-

5. Report Pages & Visualizations

Page 1: Overview (Albums, Artists, Tracks)

1. **KPI Cards** → Distinct count of Albums, Artists, Tracks
2. **Trend Analysis** → Area chart showing year-wise trends with min/max highlights
3. **YoY Comparison** → Cards for current year vs previous year with % change

4. **Weekend vs Weekday Listening** → Pie chart comparison
 5. **Top 5 Favorites** → Stacked bar chart for Albums/Artists/Tracks with Top N Visual level filter.
 6. **Slicers** → Skipped vs Not Skipped, Shuffled vs Not Shuffled
-

Page 2: Listening Patterns

1. **Heatmap** → Track count by Hour vs Day of Week
 - Matrix with conditional formatting for gradient intensity
 - Additional bar chart for hourly distribution
 2. **Scatter Plot** → Avg Listening Time vs Frequency per Track
 - Dynamic X/Y quadrant lines using Parameters
 - Conditional formatting to highlight High Frequency + High Avg Time tracks
 3. **Year Slicer** → For yearly comparison
-

Page 3: Drill-through Analysis

- Drill-through enabled from other visuals
 - Matrix table with: Album, Artist, Track, Platform, Date
 - KPIs included: No. of Albums, No. of Artists, No. of Tracks, ms_played, Avg Listening Time
 - Exportable data view for sharing insights with stakeholders
-

6. Key DAX Measures

Distinct Albums Count

```
No of Albums = DISTINCTCOUNT(spotify_history[album_name])
```

Dynamic Min/Max Highlight

```
Min Max value Album =  
VAR Max_value = MAXX(ALLSELECTED('Date Table'[Year]),  
    CALCULATE(DISTINCTCOUNT(spotify_history[album_name])))  
VAR Min_value = MINX(ALLSELECTED('Date Table'[Year]),  
    CALCULATE(DISTINCTCOUNT(spotify_history[album_name])))  
VAR Current_val = DISTINCTCOUNT(spotify_history[album_name])  
RETURN IF(Current_val = Max_value || Current_val = Min_value,  
Current_val, BLANK())
```

YoY Comparison

```
CurrentYearAlbums =  
VAR _LatestYear = MAX('Date Table'[Year])  
RETURN CALCULATE(DISTINCTCOUNT(spotify_history[album_name]), 'Date  
Table'[Year] = _LatestYear)
```

```
PreviousYearAlbums =  
VAR _LatestYear = MAX('Date Table'[Year])  
VAR _PreviousYear = _LatestYear - 1  
RETURN CALCULATE(DISTINCTCOUNT(spotify_history[album_name]), 'Date  
Table'[Year] = _PreviousYear)
```

```
PY and YoY KPI Album =  
VAR _latest = [CurrentYearAlbums]  
VAR _previous = [PreviousYearAlbums]  
VAR _YoY = IF(NOT(ISBLANK(_previous)),DIVIDE(_latest -  
_previous,_previous,0))  
RETURN IF(NOT(ISBLANK(_previous)),  
"vs PY: "& FORMAT(_previous, "#,##0") &  
" (" & FORMAT(_YoY, "0.00%") & ")", "No Data")
```

Conditional Formatting Quadrant (Scatter Plot)

```
CF Quadrant =  
VAR _AvgX = [Average Listening Time All Tracks]  
VAR _AvgY = [Average Frequency All Tracks]  
RETURN  
SWITCH(  
    TRUE(),  
    [Avg Listening Time] >= _AvgX && [Frequency] >= _AvgY, "High Time -  
- High Frequency",  
    [Avg Listening Time] >= _AvgX && [Frequency] < _AvgY, "High Time -  
Low Frequency",  
    [Avg Listening Time] < _AvgX && [Frequency] >= _AvgY, "Low Time -  
High Frequency",  
    "Low Time - Low Frequency"  
)
```

7. Business Value

- **Engagement Analysis** → Identify listening behavior across time, platform, and mood (weekday/weekend).
 - **Content Strategy** → Recognize top-performing tracks, albums, and artists.
 - **User Retention** → Detect patterns in skipped vs completed tracks.
 - **Scalable Design** → Easy to extend for new data sources or additional KPIs.
-

8. Conclusion

This project demonstrates **end-to-end BI capability**:

- Data preparation & cleaning
- Data modeling with relationships & calculated tables
- Advanced DAX for time intelligence and dynamic analysis

- Intuitive UI/UX aligned with branding
 - Business insights driving engagement and strategy
-

✨ **Author:** Shruti Bisht

✉ **Contact:** shrutibisht856@gmail.com