Spotify User Analysis

Uncovering Trends in Music Listening Behavior with Polars

Jesus L. Monroy

May, 2025

Abstract

This project explores a dataset of Spotify music listening habits, leveraging the power of the Polars DataFrame library for efficient and scalable analysis. By employing Polars' optimized data manipulation capabilities, we efficiently process and analyze Spotify data, extracting meaningful insights into user behavior and the dynamics of the music ecosystem. Furthermore, this work highlights the potential of Polars as a powerful tool for large-scale data analysis in the music industry and beyond.

Table of Contents

List of Figures

List of Tables

`Spotify`, a leading digital music, podcast, and video streaming service, has revolutionized the way individuals consume audio content. By offering a vast library of on-demand content through a freemium model, Spotify has significantly shifted music consumption from ownership to access. Spotify's expansion into podcasting and other audio content reflects its ambition to become a comprehensive audio platform.

Introduction

This study conducts a historical analysis of Spotify data, leveraging the efficiency of `Polars`, the analytical power of `DuckDB`, and the visualization capabilities of `Plotly`. By combining these tools, we examine trends in music consumption, artist popularity, and playlist dynamics over time.

Polars facilitates rapid data manipulation and cleaning of large Spotify datasets, while `DuckDB` enables complex `SQL` queries for in-depth analysis.

`Plotly` is then employed to create interactive visualizations, revealing patterns and insights that would be difficult to discern from raw data alone.

This analysis explores the evolution of musical genres, the impact of algorithmic recommendations, and the shifting landscape of artist distribution.

We demonstrate the efficacy of this toolchain for extracting meaningful narratives from Spotify's extensive historical data, providing a comprehensive understanding of the platform's influence on the music industry.

Environment settings

```
# import libraries
import numpy as np
import pandas as pd
import polars as pl
import duckdb as db
from plotnine import *
import os
from dotenv import load_dotenv
load_dotenv('.env')
from great_tables import GT, md
```

Database connection

```
# database connection
token = os.getenv('MD_TOKEN')
conn = db.connect(f"md:?motherduck_token={token}")
# create dataframe
df = conn.sql('select * from projects.spotify_history').pl()
conn.close()
```

`Duckdb` is a powerful tool for data analysts and developers who need to perform fast and efficient analytical queries on large datasets, especially in environments where simplicity and portability are crucial.

Dataset

`Polars` is a modern DataFrame library designed for speed and efficiency, offering a compelling alternative to traditional data manipulation tools.

```
  df.schema
```

```
Schema([('spotify_track_uri', String),
        ('ts', Datetime(time_unit='us', time_zone=None)),
        ('platform', String),
        ('ms_played', Int64),
        ('track_name', String),
        ('artist_name', String),
        ('album_name', String),
        ('reason_start', String),
        ('reason_end', String),
        ('shuffle', Boolean),
        ('skipped', Boolean)])
```

```
  df = (
      df.select(pl.exclude('spotify_track_uri'))
          .with_columns(year=pl.col('ts').dt.year())
  )
```

```
  (
      GT(df.head())
      .tab_header(title=md("### Spotify Songs History Dataset"))
      .fmt_date(columns='ts', date_style='iso')
      .fmt_number(columns='ms_played', decimals=0)
      .cols_hide(columns=['year','reason_start','reason_end'])
      .cols_label(
          ts='Date',
          platform='Platform',
          ms_played='Mins played',
          track_name='Song',
          artist_name='Artist',
          album_name='Album',
          reason_start='Reason start',
          reason_end='Reason end',
          shuffle='Shuffle',
          skipped='Skipped',
      )
      .tab_options(table_font_size='90%')
      .tab_options(table_width='80%')
      .tab_source_note(source_note='Source: https://mavenanalytics.io')
  )
```

Spotify Songs History Dataset

Date	Platform	Mins played	Song	Artist	Album	Shuffle	Skipped
2013-07-08	web player	3,185	Say It, Just Say It	The Mowgli's	Waiting For The Dawn	false	false
2013-07-08	web player	61,865	Drinking from the Bottle (feat. Tinie Tempah)	Calvin Harris	18 Months	false	false
2013-07-08	web player	285,386	Born To Die	Lana Del Rey	Born To Die - The Paradise Edition	false	false
2013-07-08	web player	134,022	Off To The Races	Lana Del Rey	Born To Die - The Paradise Edition	false	false
2013-07-08	web player	0	Half Mast	Empire Of The Sun	Walking On A Dream	false	false

Source: https://mavenanalytics.io

Table 1: A sample of data used in the analysis.

```
print(f'The dataset has {df.shape[0]:,.0f} rows.')
print(f'The year goes from {df['year'].min()} up to {df['year'].max()}.')
```

```
The dataset has 149,860 rows.
The year goes from 2013 up to 2024.
```

For further information about the fields of the dataset see Table 7

Findings

Unique Artists, Albums and Songs

```
# unique values per year
tops = (
        df.select('year','artist_name','album_name','track_name')
            .group_by('year', maintain_order=True)
            .n_unique()
)
```

```
(
    GT(tops)
    .tab_header(
        title=md("### Unique Artists, Albums and Songs Played by Year")
    )
    .fmt_number(columns=['artist_name','album_name','track_name'],
                decimals=0)
    .cols_label(
        year='Year',
        artist_name='Artists',
        album_name='Albums',
        track_name='Songs',
    )
    .tab_options(table_font_size='90%')
    .tab_options(table_width='80%')
    .tab_source_note(source_note='Source: Self-elaboration by author')
)
```

Unique Artists, Albums and Songs Played by Year

Year	Artists	Albums	Songs
2013	63	95	149
2014	21	22	23
2015	610	924	1,357
2016	458	940	2,535
2017	647	1,357	3,326
2018	439	947	2,917

Table 2: Unique Artists, Albums and Songs by Year.

Unique Artists, Albums and Songs Played by Year

Year	Artists	Albums	Songs
2019	493	1,029	2,934
2020	820	1,569	3,929
2021	1,578	2,674	5,126
2022	1,220	2,284	4,477
2023	1,456	2,336	4,042
2024	1,072	1,828	3,587

Source: Self-elaboration by author

Minutes Played by Month and Year

```
years = (
    df.group_by('year', maintain_order=True)
        .agg((pl.col('ms_played').sum()/60_000).round(2)
        .alias('total_mins'))
)
```

```
months = (
    df.group_by_dynamic('ts', every='1mo')
        .agg((pl.col('ms_played').sum()/60_000).round(2)
        .alias('total_mins'))
)
```

```
(
    ggplot(years, aes(x='year', y='total_mins'))
    + geom_col(color='black', fill='#1DB954')
    + geom_text(
        aes(label='total_mins'),
        size=8,
        va='bottom',
        format_string="{:,.0f}",
    )
    + labs(
        x='',
        y='Minutes played',
        title='Total Minutes Played per Year'
    )
    + theme(axis_text_x =element_text(angle=0))
    + scale_x_continuous(breaks=np.arange(2013, 2025, 1))
    + scale_y_continuous(breaks=np.arange(0, 60_000, 10_000))
    + theme(figure_size=(5.5, 4))
)
```

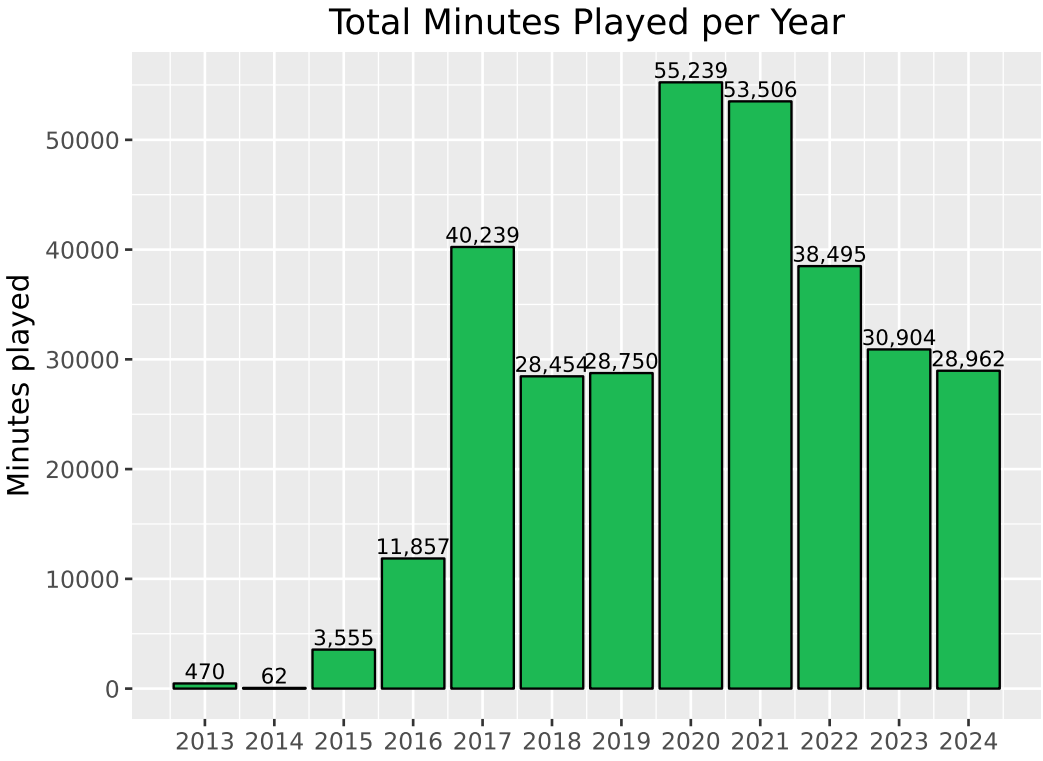

Figure 1: Total of Minutes Played By Year.

```
(
    ggplot(months, aes(x='ts', y='total_mins'))
    + geom_line(color='#1DB954', size=1.1)
    + scale_y_continuous(breaks=np.arange(0, 8_000, 1_000))
    + labs(
        x='',
        y='Minutes played',
        title='Total Minutes Played per Month'
    )
    + theme(axis_text_x =element_text(angle=0))
    + theme(figure_size=(5.6, 4))
)
```

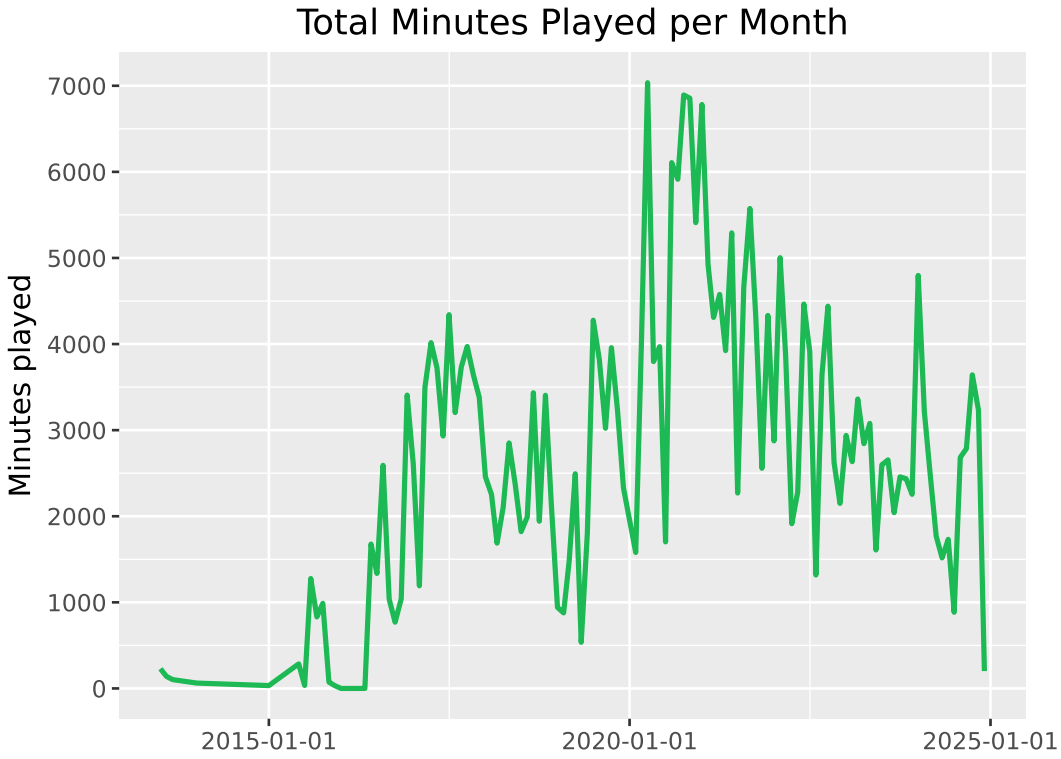

Figure 2: Total of Minutes Played By Year and Month.

Top 05 Artists by Year

```
top_artists = (
    df
        .group_by('year','artist_name')
        .agg(pl.len())
        .sort(by=['year','len'], descending=True)
        .select(
            pl.all().top_k_by('len', k=5).over('year')
        )
        .group_by('year')
        .agg(pl.col('artist_name'))
        .sort('year')
)
```

```
(
    GT(top_artists)
    .tab_header(title=md("### Top 05 Artists by Year"))
     .cols_width(
        cases={
            'year': '10%',
            'artist_name': '100%',
        }
    )
```

```
    .cols_label(
        year='Year',
        artist_name='Artist'
    )
    .cols_align(align='left', columns=('artist_name'))
    .tab_options(table_font_size='90%')
    .tab_options(table_width='80%')
    .tab_source_note(source_note='Source: Self-elaboration by author')
)
```

Table 3: Top 05 Artists by Year.

Top 05 Artists by Year

Year	Artist
2013	['John Mayer', 'The Kooks', 'Lana Del Rey', 'Coldplay', 'Passion Pit']
2014	['The Neighbourhood', 'Blur', 'Bloc Party', 'Freddie King', 'Funeral Suits']
2015	['The Script', 'The Rolling Stones', 'David Bisbal', 'Justin Bieber', 'We The Kings']
2016	['The Beatles', 'Led Zeppelin', 'Elvis Presley', 'Johnny Cash', 'Vampire Weekend']
2017	['The Beatles', 'John Mayer', 'The Killers', 'Bob Dylan', 'Radiohead']
2018	['The Beatles', 'Paul McCartney', 'Bob Dylan', 'John Mayer', 'The Killers']
2019	['The Beatles', 'Paul McCartney', 'The Killers', 'Bob Dylan', 'John Mayer']
2020	['The Killers', 'The Beatles', 'John Mayer', 'The Strokes', 'Bob Dylan']
2021	['The Beatles', 'The Killers', 'John Mayer', 'Bob Dylan', 'Kings of Leon']
2022	['The Beatles', 'Joaquín Sabina', 'The Killers', 'John Mayer', 'Howard Shore']
2023	['The Beatles', 'The Killers', 'Bob Dylan', 'John Mayer', 'Howard Shore']
2024	['The Beatles', 'John Mayer', 'The Killers', 'ABBA', 'Paul McCartney']

Source: Self-elaboration by author

Top 05 Albums by Year

```
top_albums = (
    df
        .group_by('year','album_name')
        .agg(pl.len())
        .sort(by=['year','len'], descending=True)
        .select(
            pl.all().top_k_by('len', k=5).over('year')
        )
        .group_by('year')
        .agg(pl.col('album_name'))
        .sort('year')
)
```

```
(
    GT(top_albums)
    .tab_header(title=md("### Top 05 Albums by Year"))
```

```
      .cols_width(
         cases={
             'year': '10%',
             'album_name': '100%',
         }
     )
     .cols_label(
         year='Year',
         album_name='Album'
     )
     .cols_align(align='left', columns=('album_name'))
     .tab_options(table_font_size='90%')
     .tab_options(table_width='80%')
     .tab_source_note(source_note='Source: Self-elaboration by author')
)
```

Table 4: Top 05 Albums by Year.

Top 05 Albums by Year

Year	Album
2013	['Inside In / Inside Out', 'Born To Die - The Paradise Edition', 'Gossamer', 'Where the Light Is: John Mayer Live In Los Angeles', 'Battle Studies']
2014	['Blur: The Best Of', "I'm Sorry...", 'LP', '30 Beatles Top Hits', 'Can't Get Better Than This']
2015	['Tú Y Yo', 'No Sound Without Silence', 'Ultimate Sinatra', 'Somewhere Somehow', 'Hozier']
2016	['Elvis At Sun', 'At Folsom Prison', 'New York', 'Ultimate Sinatra', 'The Beatles']
2017	['The Beatles', 'Past Masters', 'Abbey Road', 'The Wall', 'Help!']
2018	['The Beatles', 'Egypt Station', 'Past Masters', 'Beatles For Sale - Remastered', 'The Wall']
2019	['Abbey Road', 'The Beatles', 'Past Masters', 'Prismism', 'Egypt Station']
2020	['Imploding The Mirage', 'The New Abnormal', 'The Beatles', 'The Wall', 'Hot Fuss']
2021	['Pressure Machine', 'When You See Yourself', 'The Beatles', 'Abbey Road', 'Past Masters']
2022	['The Wall', 'Revolver', 'Past Masters', 'Pressure Machine', 'The Lord of the Rings: The Fellowship of the Ring - the Complete Recordings']
2023	['The Beatles', 'Abbey Road', 'Past Masters', "Sam's Town", 'Hot Fuss']
2024	['Arrival', 'Past Masters', 'The Beatles', 'Born and Raised', 'Paradise Valley']

Source: Self-elaboration by author

Top 05 Songs by Year

```
top_songs = (
    df
        .group_by('year','track_name')
        .agg(pl.len())
        .sort(by=['year','len'], descending=True)
        .select(
            pl.all().top_k_by('len', k=5).over('year')
```

```
        )
        .group_by('year')
        .agg(pl.col('track_name'))
        .sort('year')
)
```

```
(
    GT(top_songs)
    .tab_header(title=md("### Top 05 Songs by Year"))
     .cols_width(
        cases={
            'year': '10%',
            'track_name': '100%',
        }
    )
    .cols_label(
        year='Year',
        track_name='Song'
    )
    .cols_align(align='left', columns=('track_name'))
    .tab_options(table_font_size='90%')
    .tab_options(table_width='80%')
    .tab_source_note(source_note='Source: Self-elaboration by author')
)
```

Table 5: Top 05 Songs by Year.

Top 05 Songs by Year

Year	Song
2013	['Paper Doll', 'Mirrors', 'Young And Beautiful', 'Born To Die', "I Still Haven't Found What I'm Looking For"]
2014	['Awake Now', 'Walking By Myself - Remastered', 'Florida', 'I Want to Hold Your Hand', 'Sleeping Lessons']
2015	['Paraíso', 'What Do You Mean?', 'Diez Mil Maneras', 'Superheroes', 'Switzerland']
2016	['The River', 'Teenagers', 'Not Today', 'South Bound Saurez - Remaster', "I've Got You Under My Skin"]
2017	['Married with Children - 2014 Remaster', 'In the Blood', 'Perfect', 'The Man', 'Wildfire']
2018	['Dominoes', 'No Words - 2010 Remaster', 'Universal Gleam', 'Band On The Run - 2010 Remaster', 'Sé Que Te Duele']
2019	['Reminder', 'Once', "Maybe It's Time", 'Band On The Run - 2010 Remaster', 'Mariposa Traicionera']
2020	['Caution', 'Ode To The Mets', 'Dying Breed', 'Imploding The Mirage', 'My Own Soul's Warning']
2021	['Crucify Your Mind', '100,000 People', 'Not in Nottingham', 'When You See Yourself, Are You Far Away', 'Did My Best']
2022	['19 Dias y 500 Noches - En Directo', 'Postdata', 'Por el Bulevar de los Sueños Rotos', 'Se Me Olvidó Otra Vez - Unplugged; 2020 Remasterizado', 'Telefonía']
2023	['You Sexy Thing', 'Primera Cita', 'Qué Bonito Es Querer', 'Always Alright', 'Billy 4']

Top 05 Songs by Year

Year	Song
2024	['Why Did It Have To Be Me?', 'Superdeli', 'Does Your Mother Know', 'Knowing Me, Knowing You', 'Dancing Queen']

Source: Self-elaboration by author

```
platform = (
    df.group_by('platform')
        .agg(pl.len())
        .sort('len', descending=True)
        .with_columns((pl.col("len") / pl.sum("len")).alias("percent"))
)
```

```
(
    GT(platform)
    .tab_header(title=md("### Percentage of Use by Platform"))
    .fmt_number(columns='len', decimals=0)
    .fmt_percent('percent', decimals=2)
    .cols_label(
        platform='Platform',
        len='Total',
        percent='Percentage'
    )
    .cols_align(align='left', columns=('platform'))
    .tab_options(table_font_size='90%')
    .tab_options(table_width='80%')
    .tab_source_note(source_note='Source: Self-elaboration by author')
)
```

Percentage of Use by Platform

Platform	Total	Percentage
android	139,821	93.30%
cast to device	3,898	2.60%
iOS	3,049	2.03%
windows	1,691	1.13%
mac	1,176	0.78%
web player	225	0.15%

Source: Self-elaboration by author

Table 6: Total of Records Played by Platform.

Conclusions

We can conclude with the following findings:

Listening Habits Over Time

The total minutes played show an increase by getting a peak, since then it has been decreasing in listening time over the years, with a peak in **2020**. This could be attributed to reasons like new devices, changes in lifestyle, discovery of new music.

Monthly listening patterns reveal seasonal engagement, possibly due to events, or holidays.

Discovery and Exploration

The number of unique artists, albums, and songs played has remained relatively stable over the years, suggesting a shrinking interest in discovering new music.

A significant increase in unique songs played compared to unique albums in a particular year might indicate a trend towards listening to individual tracks rather than full albums.

Platform Usage

The primary platform for listening is **Android** Smartphones, accounting for around 93% of total playtime, indicating its importance in music consumption. Usage of iOS remains low over time.

Artist Loyalty and Discovery

The consistent presence of **The Beatles** in the top 05 artists across multiple years suggests a strong and enduring preference for their music.

The emergence of new artists in the top 05 each year indicates a continuous exploration and adoption of new musical tastes.

Album Engagement

The recurrence of specific albums in the top 05 albums across different years highlights their lasting appeal and potentially signifies comfort listening or revisiting favorite works.

A high turnover in the top 05 albums year-over-year might suggest a focus on new releases and less revisiting of older albums.

Song Popularity and Trends

The top 05 songs often include tracks from the top 05 artists of the same year, indicating a strong correlation between artist preference and song popularity.

The appearance of certain genres or moods within the top 05 songs for a specific year could reflect prevailing personal preferences or external influences during that period.

References

- Maven Analytics (2025). Maven Music Challenge Dataset. Retrieved from Maven Analytics Website.
- Yonak, R (2019). How Spotify has changed the way we listen to music. Retrieved from Audioxide Website.

Appendix

Table 7: Data dictionary, detailing field names and descriptions

Field	Description
spotify_track_uri	Spotify URI that uniquely identifies each track
ts	Timestamp indicating when the track stopped playing in UTC
platform	Platform used when streaming the track
ms_played	Number of milliseconds the stream was played
track_name	Name of the track
artist_name	Name of the artist
album_name	Name of the album
reason_start	Why the track started
reason_end	Why the track ended
shuffle	If shuffle mode was used when playing the track (boolean)
skipped	If the user skipped to the next song (boolean)

Contact

Jesus L. Monroy
Economist & Data Scientist

Linkedin | Medium | Twitter