# What Makes A Playlist Successful?

By Venkat Aganya Mutupuri

# Agenda

Part A:

- Problem Statement And Dataset
- Summary Statistics
- Defining Success And Metrics To Measure Success
- Exploring Successful And Unsuccessful Playlists
- Summary

Part B:

- How would you change the current schema, or design a new one, to make it more analysis friendly?
- If this data was scaled up 1000x, across multiple tables, how would you have differently designed your solution and why? What about if the data was scaled up 1000000x?
- If you were asked to produce a version of your work which ran in an automated way, how would you do so?

# Problem Statement: What Makes A Playlist Successful?

- Dataset

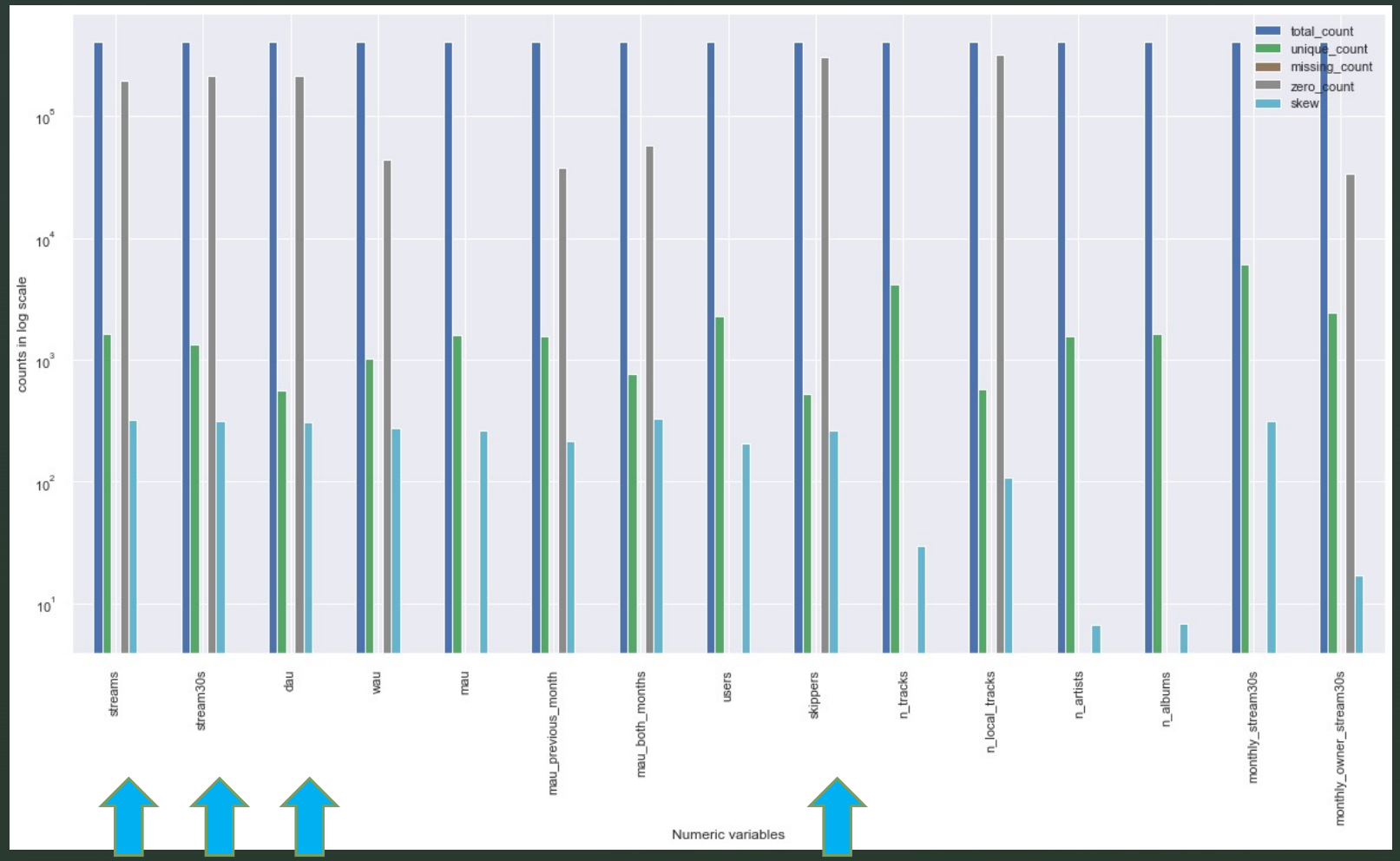
| | playlist_uri | owner | streams | stream30s | dau | wau | mau | mau_previous_month | mau_both_months | users | ... | n_albums | monthly_stream30s | monthly_owner_stream30s | tokens | genre_1 | genre_2 | genre_3 | mood_1 | mood_2 | mood_3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | spotify:user:977723f001dc75663d2fcfe55b9b4b70:... | 977723f001dc75663d2fcfe55b9b4b70 | 27 | 27 | 1 | 1 | 3 | 3 | 0 | 8 | ... | 7 | 30 | 27 | [ambient, music, therapy, binaural, beats, amb... | Dance & House | New Age | Country & Folk | Peaceful | Romantic | Somber |
| 2 | spotify:user:611d6958470da7b36b3a7b5700462f74:... | 611d6958470da7b36b3a7b5700462f74 | 0 | 0 | 0 | 1 | 2 | 1 | 1 | 3 | ... | 113 | 112 | 94 | [good, living] | Pop | Indie Rock | Alternative | Excited | Yearning | Defiant |
| 3 | spotify:user:e2ff3a8d0187e4bd221e64fd924e7ea9:... | e2ff3a8d0187e4bd221e64fd924e7ea9 | 4 | 2 | 1 | 1 | 7 | 5 | 0 | 15 | ... | 36 | 63 | 0 | [norte\u00f1a] | Latin | - | - | Lively | Upbeat | Romantic |
| 4 | spotify:user:6719b0a5dc93e068a3c4f04081bcce6b:... | 6719b0a5dc93e068a3c4f04081bcce6b | 12 | 12 | 1 | 1 | 4 | 6 | 1 | 10 | ... | 26 | 154 | 108 | [] | Dance & House | Electronica | Pop | Excited | Aggressive | Defiant |
| 5 | spotify:user:99d635f08ed668cdf7e36540fb653276:... | 99d635f08ed668cdf7e36540fb653276 | 20 | 4 | 1 | 1 | 2 | 1 | 1 | 2 | ... | 51 | 230 | 0 | [cheesy, pants] | Indie Rock | Alternative | Electronica | Excited | Defiant | Yearning |

- The dataset contains 403366 observations and 25 variables
- Data types are integer for numeric variables and string for text variables

```
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   playlist_uri             403366 non-null  object
 1   owner                    403366 non-null  object
 2   streams                  403366 non-null  int64
 3   stream30s                403366 non-null  int64
 4   dau                      403366 non-null  int64
 5   wau                      403366 non-null  int64
 6   mau                      403366 non-null  int64
 7   mau_previous_month       403366 non-null  int64
 8   mau_both_months          403366 non-null  int64
 9   users                    403366 non-null  int64
 10  skippers                 403366 non-null  int64
 11  owner_country            403366 non-null  object
 12  n_tracks                 403366 non-null  int64
 13  n_local_tracks           403366 non-null  int64
 14  n_artists                403366 non-null  int64
 15  n_albums                 403366 non-null  int64
 16  monthly_stream30s        403366 non-null  int64
 17  monthly_owner_stream30s  403366 non-null  int64
 18  tokens                   403366 non-null  object
 19  genre_1                  403366 non-null  object
 20  genre_2                  403366 non-null  object
 21  genre_3                  403366 non-null  object
 22  mood_1                   403366 non-null  object
 23  mood_2                   403366 non-null  object
 24  mood_3                   403366 non-null  object
dtypes: int64(15), object(10)
memory usage: 76.9+ MB
```


Summary
Statistics

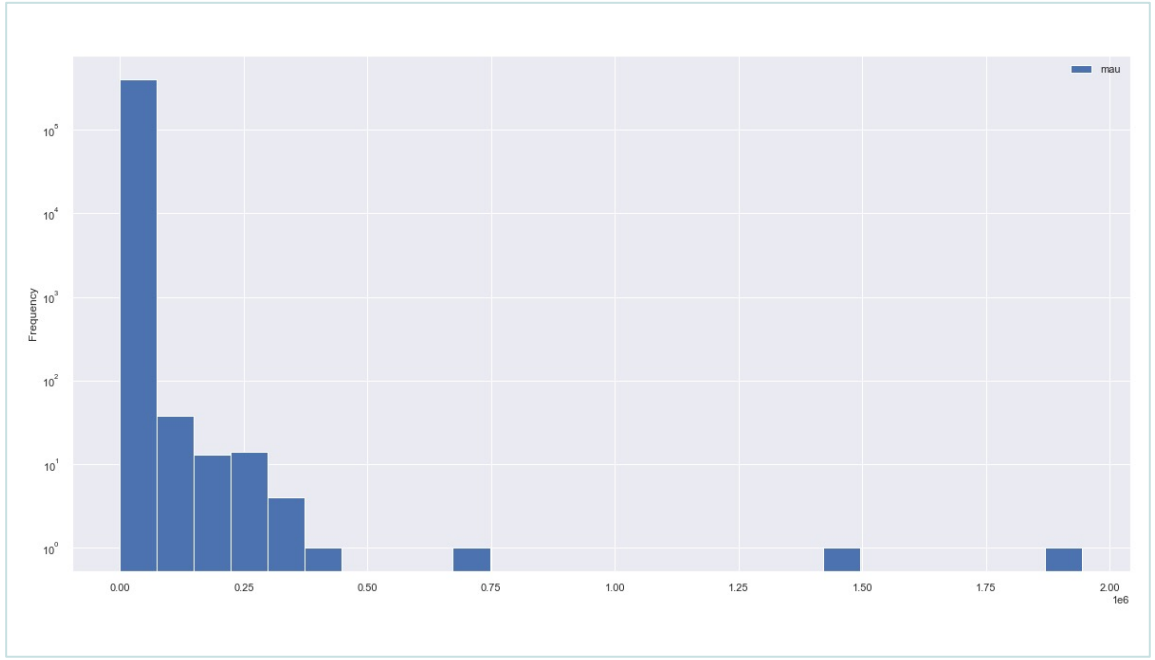
# Summary Statistics: Numeric Variables



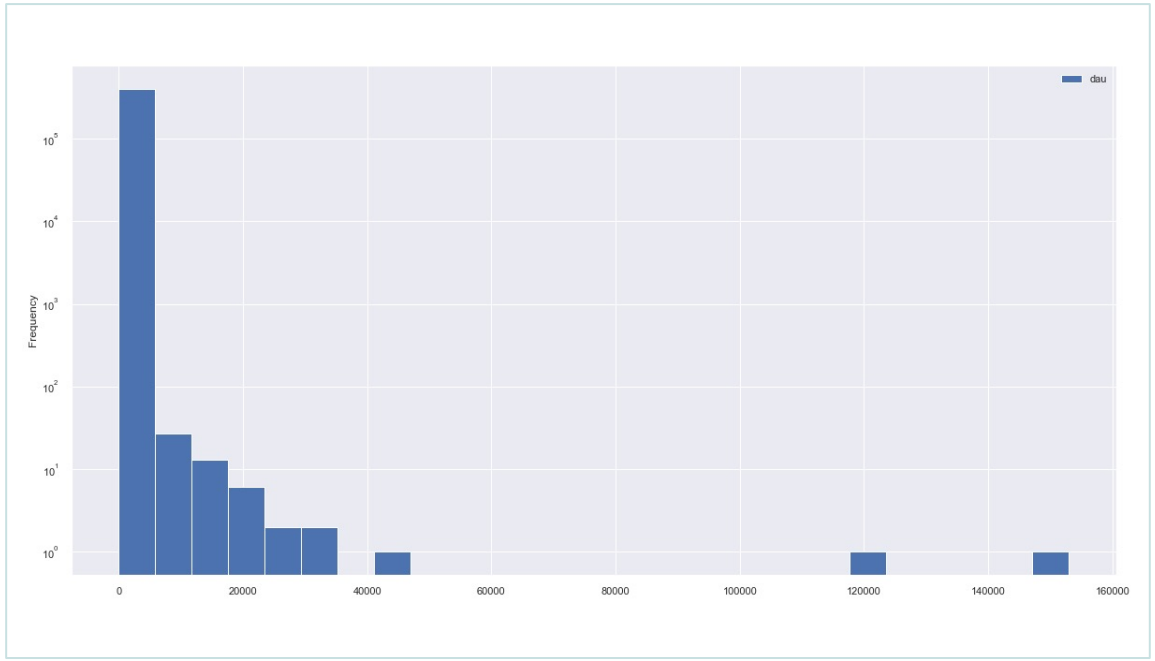

- Zero count is on the higher side for variables like streams, dau, skippers, n_local_tracks.

# Summary Statistics: Data Distribution

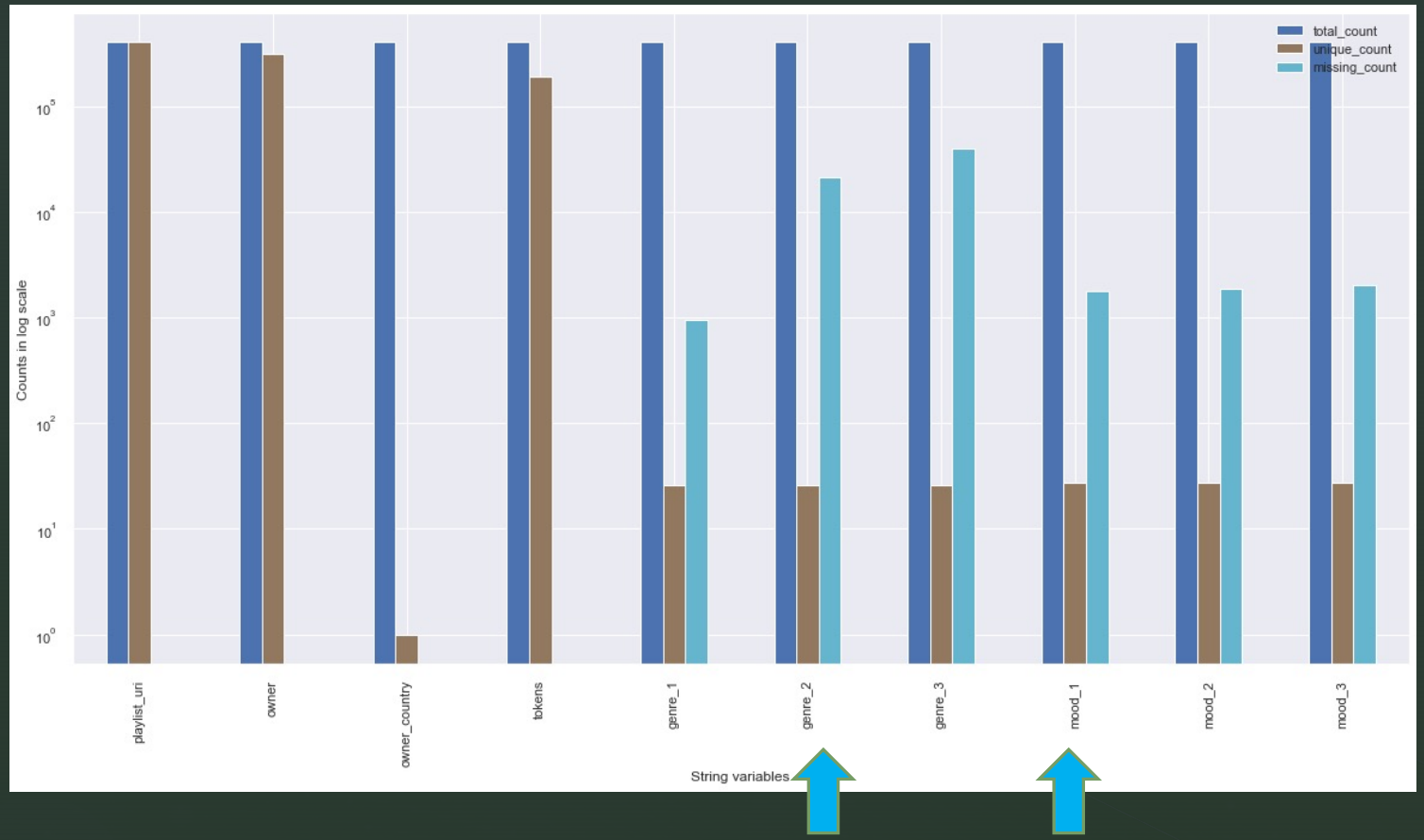
mau




dau



# Summary Statistics: String Variables




- Missing values indicated with '-' are on the higher side for variables like genre and mood compared to owner, country which do not have any missing values.

# Summary Statistics: Frequency Charts

Genre

| Value | Count | Frequency (%) |
|---|---|---|
| Defiant | 81079 | 20.1% |
| Excited | 64119 | 15.9% |
| Yearning | 38511 | 9.5% |
| Empowering | 33089 | 8.2% |
| Sensual | 25807 | 6.4% |
| Energizing | 24387 | 6.0% |
| Brooding | 18177 | 4.5% |
| Upbeat | 16189 | 4.0% |
| Urgent | 15023 | 3.7% |
| Cool | 11808 | 2.9% |
| Other values (17) | 75177 | 18.6% |

Mood

| Value | Count | Frequency (%) |
|---|---|---|
| Indie Rock | 70571 | 17.5% |
| Rap | 63523 | 15.7% |
| Pop | 61755 | 15.3% |
| Dance & House | 32111 | 8.0% |
| Rock | 28077 | 7.0% |
| Alternative | 23633 | 5.9% |
| Country & Folk | 21231 | 5.3% |
| R&B | 17422 | 4.3% |
| Latin | 15012 | 3.7% |
| Religious | 14074 | 3.5% |
| Other values (16) | 55957 | 13.9% |

# Observations From The Data

- Number of variables: 25, string variables: 10, numeric variables: 15
- Data is skewed to the right
- No duplicates in the dataset
- Variables like streams, sream30s, dau, skippers, n_local_tracks have 0 count > 50% of the total count
- Variable owner_country has 1 unique value
- String variables have < 10% missing values
- Datatypes are optimized, we can convert genre and moods to categories as there are approximately 27 unique values out of 403366 values in the data set

# Defining Success

# Defining Success And Metrics To Measure Success

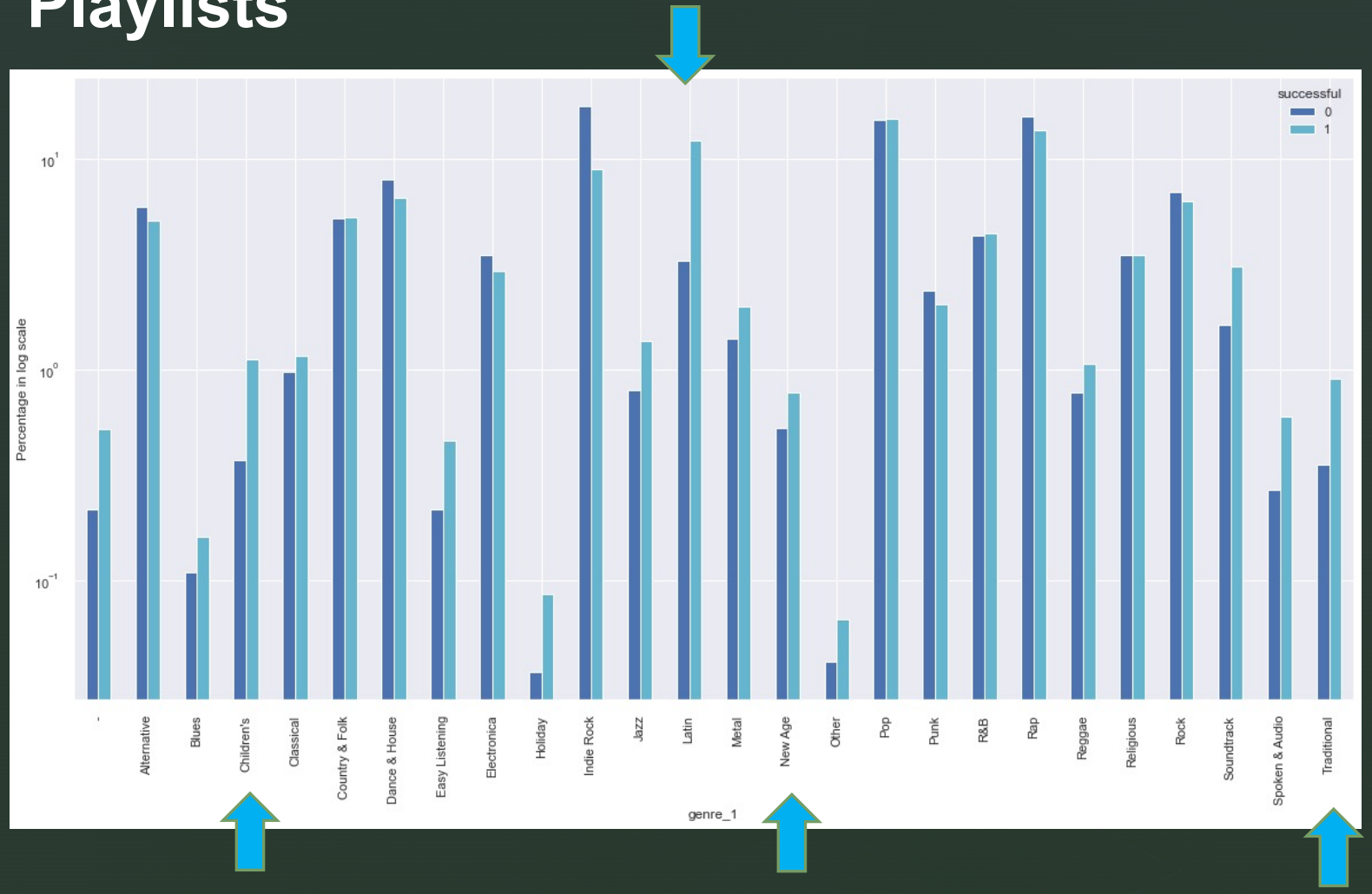
- Success from a product perspective is to have high active users on the platform and engage users for longer stream time.
- Metrics like dau, mau, streams30s, ma_streams30s will give us insight into active user streams over 30 seconds on the platform and give us insight into engagement with the playlists.

# Classifying Playlists As Successful And Unsuccessful

```
#metric1: DAU
dau_percent = df_playlist_analysis['dau'].quantile(0.75)
df_playlist_analysis['dau_success'] = df_playlist_analysis.apply(lambda x: 1 if x['dau'] > dau_percent else 0, axis=1)
#metric2: stream30s
dstreams_percent = df_playlist_analysis['stream30s'].quantile(0.75)
df_playlist_analysis['dstreams_success'] = df_playlist_analysis.apply(lambda x: 1 if x['stream30s'] > dstreams_percent else 0, axis=1)
#metric3: MAU
mau_percent = df_playlist_analysis['mau'].quantile(0.75)
df_playlist_analysis['mau_success'] = df_playlist_analysis.apply(lambda x: 1 if x['mau'] > mau_percent else 0, axis=1)
#metric4: monthly_stream30s
mstreams_percent = df_playlist_analysis['monthly_stream30s'].quantile(0.75)
df_playlist_analysis['mstreams_success'] = df_playlist_analysis.apply(lambda x: 1 if x['monthly_stream30s'] > mstreams_percent else 0, axis=1)
#Checking if a playlist is successful for all 4 metrics
df_playlist_analysis['successful'] = df_playlist_analysis.apply(lambda x: 1 if (x['dau_success'] == 1 & x['dstreams_success'] == 1 & x['mau_success'] == 1 & x['mstreams_success'] == 1) else 0, axis=1)
```

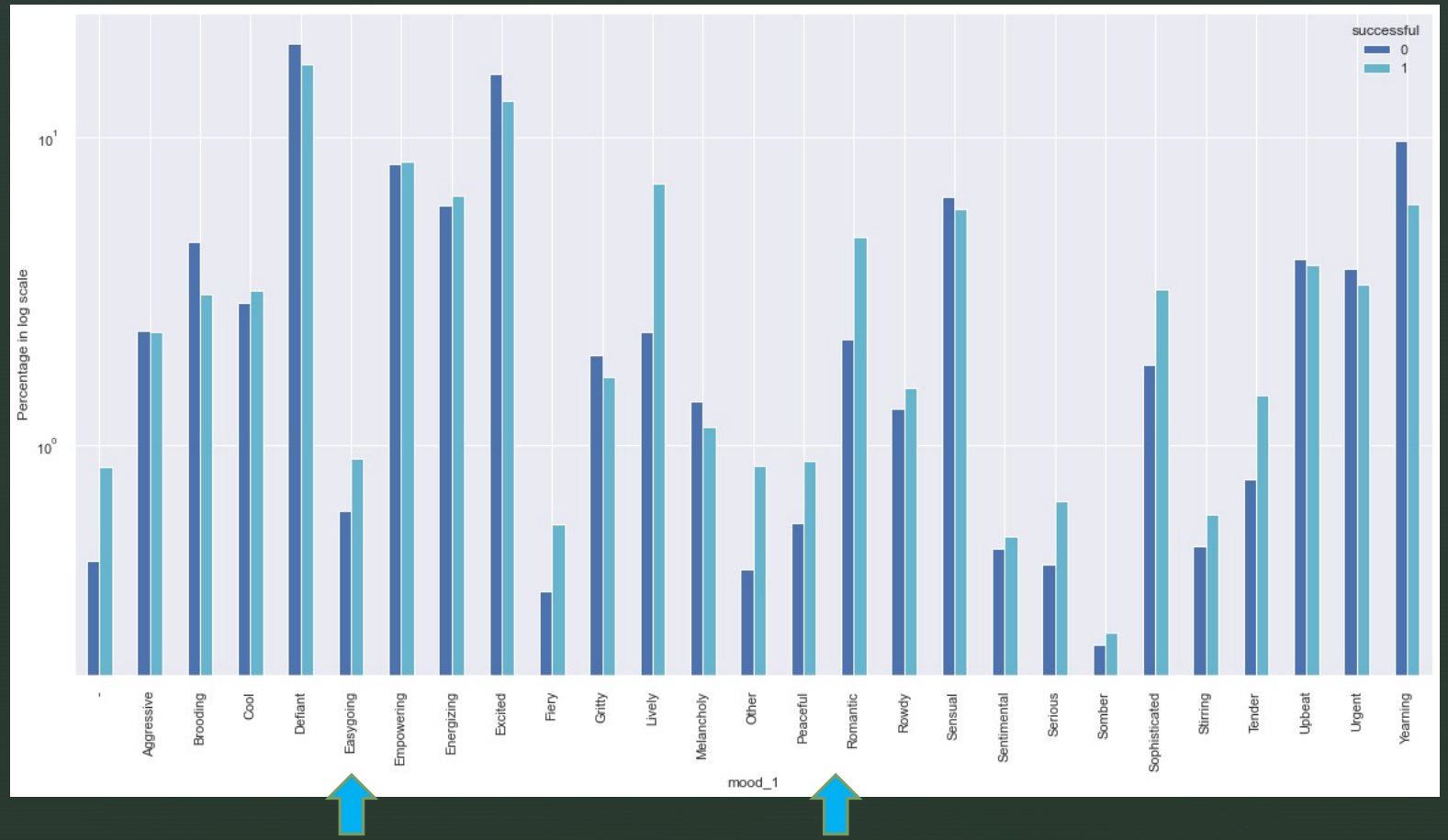
- Success for each metric: for each metric selected, data in the ¼ top quarter or top 25% is considered as successful, bottom ¾ quarter or bottom 75% is considered unsuccessful for the specific metric.
- Defining a successful playlist: once we indicate success for each metric if the playlist is successful on all chosen metrics, the playlist is tagged as overall successful.
- There are ~5% successful playlists of total playlists for all the chosen metrics.

# Genres For Successful And Unsuccessful Playlists



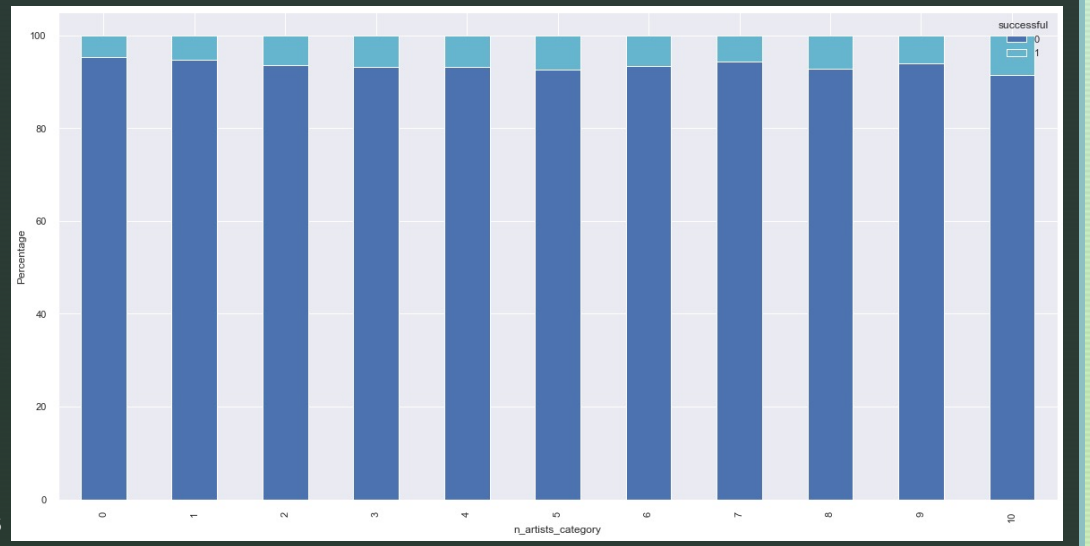

- genre 1: highest weighted genre

# Moods For Successful And Unsuccessful Playlists

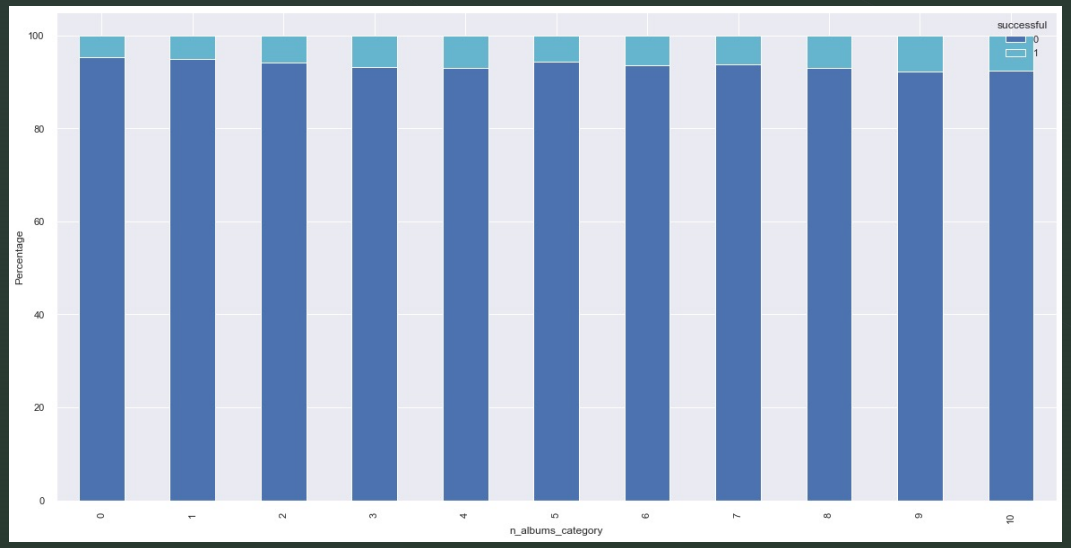



- mood 1: highest weighted mood

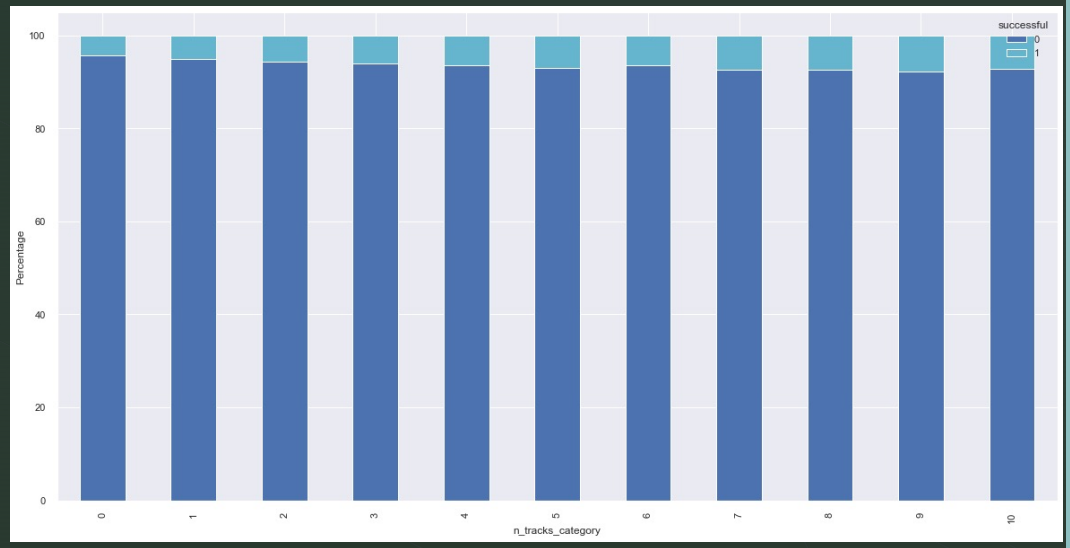
# Number Of Tracks, Artists And Albums For Successful And Unsuccessful Playlists




n_artists




n_albums




n_tracks

- As the number of tracks, albums and artists increase the percentage of successful playlists increase as well, in small numbers

# Summary

- There are a total of 403366 playlists and 19780 are successful for all the metrics, ~5% of the playlists.
- Genres like Blues, Latin and Traditional are popular over successful playlists and genres like Alternative, Dance & House, Rap are popular over unsuccessful playlists.
- Moods like Easy Going, Peaceful and Romantic are popular over successful playlists and moods like Defiant, Melancholy and Yearning are popular over unsuccessful playlists.
- Having a greater number of tracks, albums, artists shows some positive impact over successful playlists.

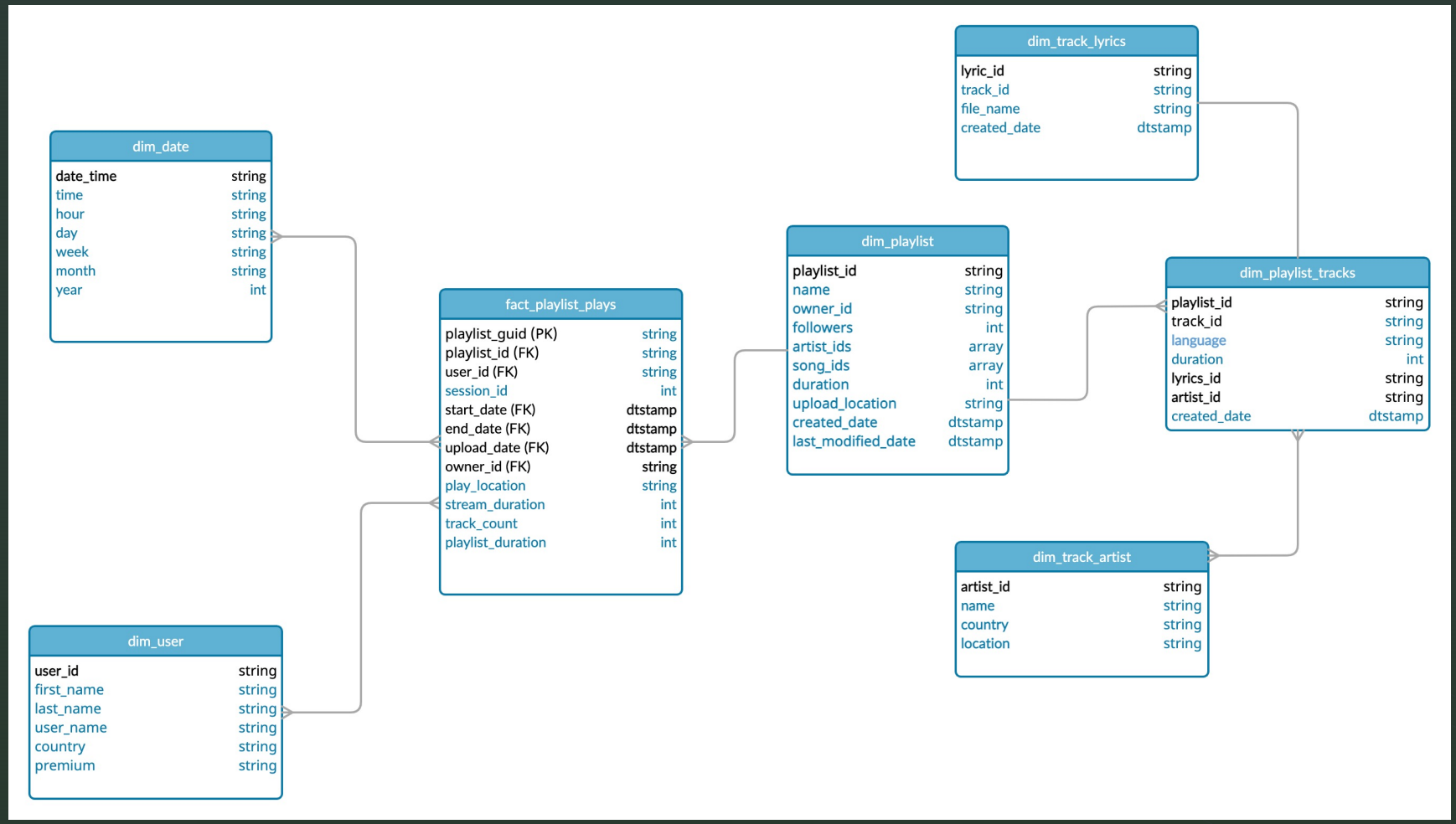
# Schema Design

# How would you change the current schema, or design a new one, to make it more analysis friendly?

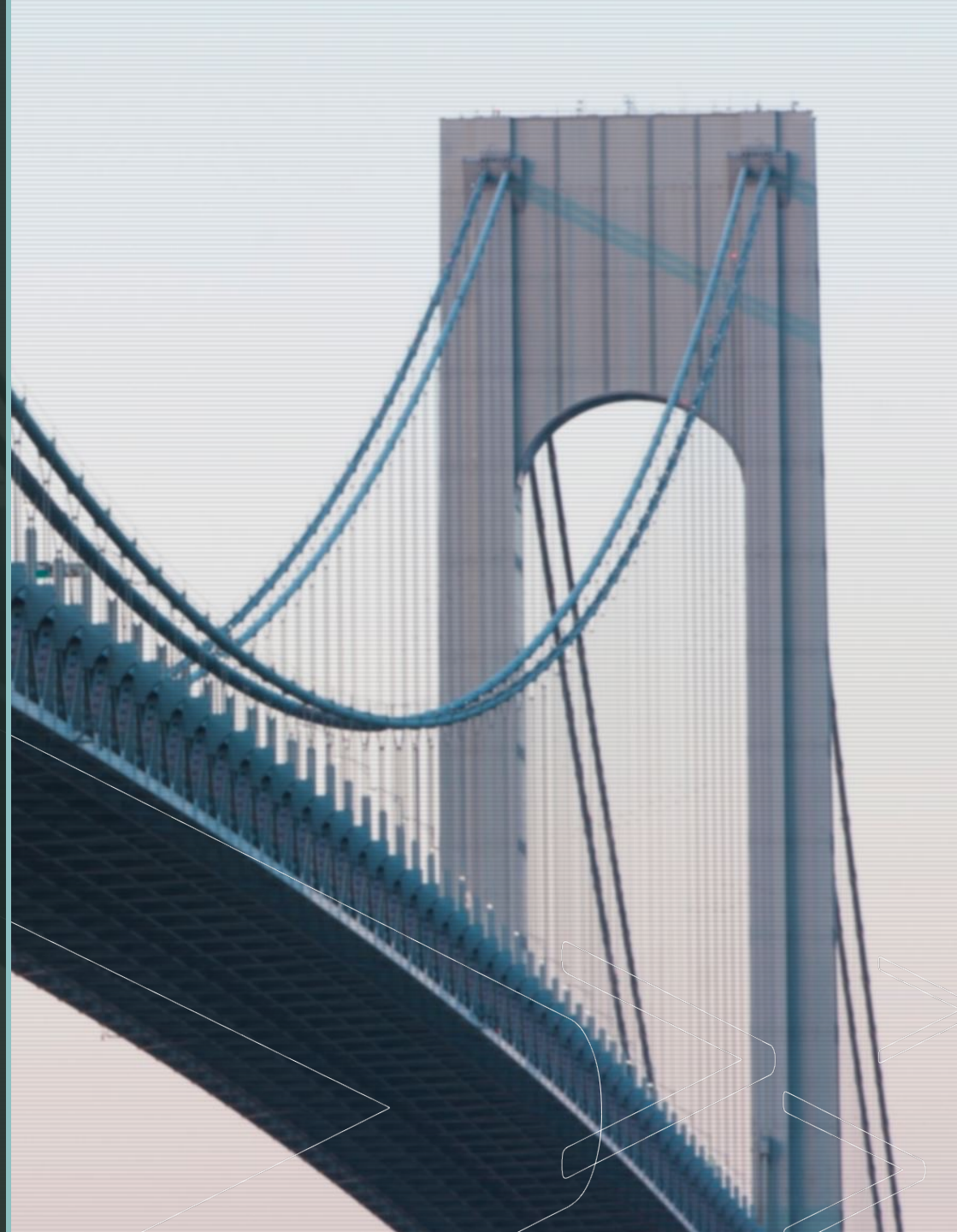
Considerations and Assumptions:

- Stared with the lowest grain for playlist analysis, definition of dimensions and facts will help slice the data across multiple dimensions.
- Understanding the structure of source data gives a clear picture on schema design.
- Extended discussions are necessary with the consumer of schema on how data must be presented so that the user may not have to do additional transformations for further analysis.
- The schema is created keeping presented data set in mind.
- Some assumptions were made while designing the schema like: we have tables for songs, song plays, tracks, playlist, albums, user information etc.

# Schema Design

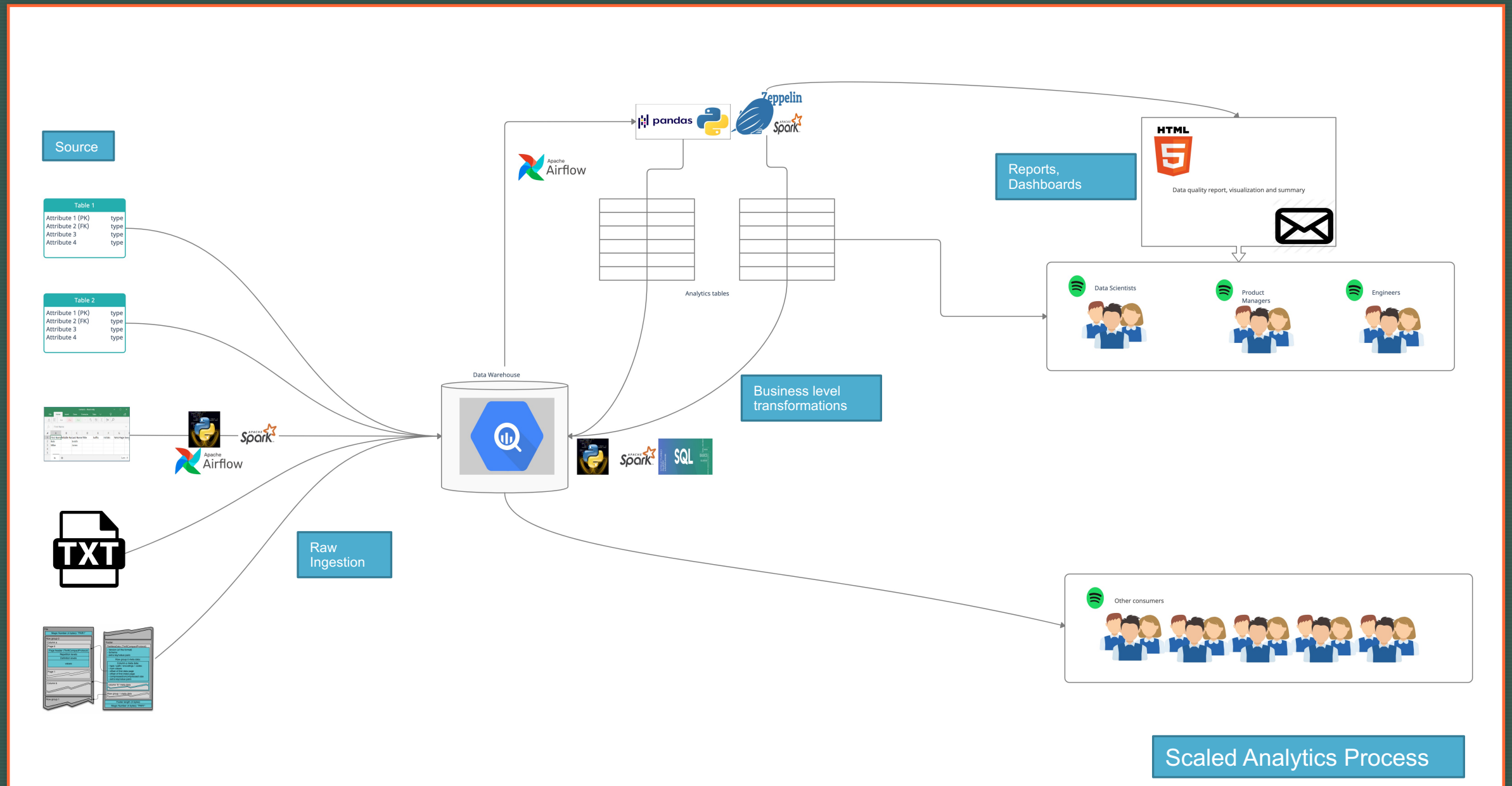


# Scaling Up



## If this data was scaled up 1000x, across multiple tables, how would you have differently
## designed your solution and why? What about if the data was scaled up 1000000x?

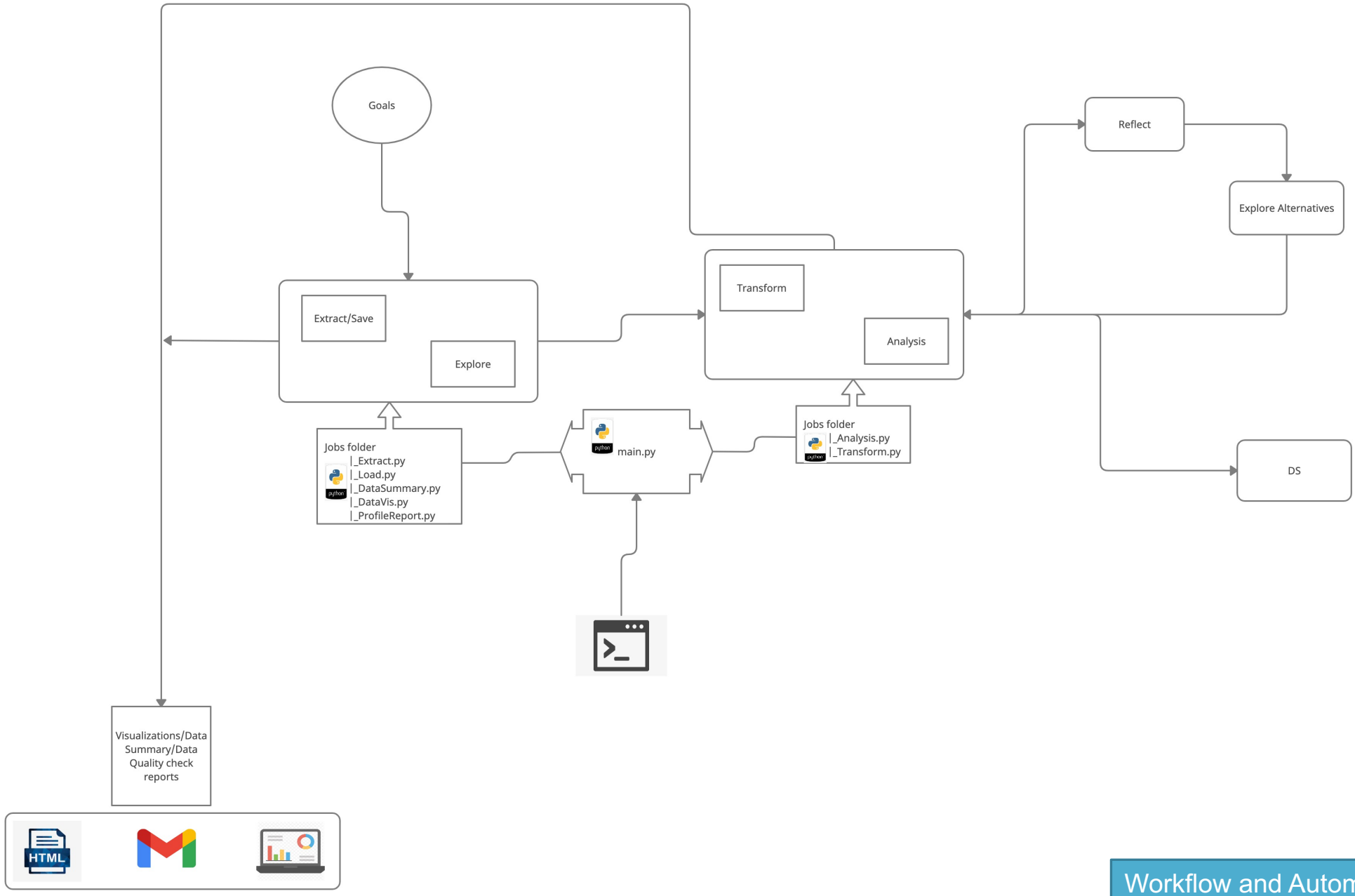
- Schema design:
  - Keeping tables compact by not adding additional attributes to the dataset outside the scope of analytic function.
  - Right level of normalization, additional normalizing of data will result in complicated schema design and affects the performance.
  - Right key design-surrogate keys, clustered keys, hashed keys for performance, lookups, data partitioning.
- Use column store databases/ file storages this will help of IO for reading and writing data.
- Currently my solution uses the local compute power and storage, using distributed storage and computing for faster data processing, engines like Spark and column store types like Parquet will decrease the data processing time and faster refresh rate of data.


Source
Table 1
Attribute 1 (PK) type
Attribute 2 (FK) type
Attribute 3 type
Attribute 4 type
Table 2
Attribute 1 (PK) type
Attribute 2 (FK) type
Attribute 3 type
Attribute 4 type
Apache Airflow
Spark
TXT
Raw Ingestion
pandas
Zeppelin
Spark
Apache Airflow
Analytics tables
Data Warehouse
Spark
SQL
Business level transformations
Reports, Dashboards
HTML
Data quality report, visualization and summary
Data Scientists
Product Managers
Engineers
Other consumers
Scaled Analytics Process

# Automation

# If you were asked to produce a version of your work which ran in an automated way, how would you do so?

- Assuming the analysis must be done on a data set similar size and we are tracking the same metrics.
- The code can be divided into its respective files of function e.g., extract.py, transform.py, summary.py, analysis.py etc.
- It must be a configuration driven process, passing arguments like file paths for read and write, email address for notifications or data summaries etc.
- A shell script can be used to pipeline the process by passing in necessary arguments.
- Log any exceptions to a file e.g., logging module.
- A high-level representation of the workflow can be found on next slide.




# Workflow and Automation

# Appendix

# Summary Statistics: Numeric Variables

Variable names

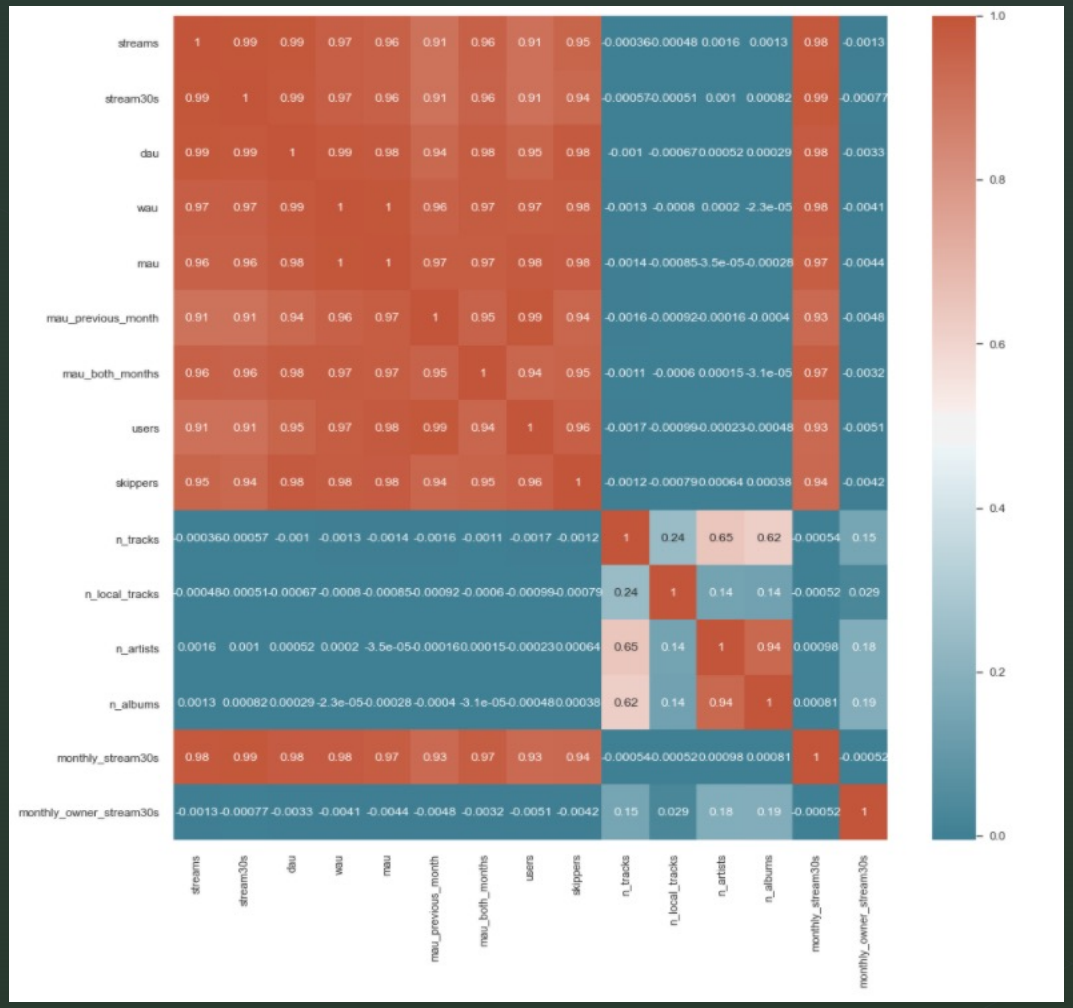
| | streams | stream30s | dau | wau | mau | mau_previous_month | mau_both_months | users | skippers | n_tracks | n_local_tracks | n_artists | n_albums | monthly_stream30s | monthly_owner_stream30s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| max_value | 2629715 | 1513237 | 152929 | 669966 | 1944150 | 1478684 | 578391 | 3455406 | 86162 | 79984 | 9117 | 5226 | 6397 | 42497334 | 25904 |
| min_value | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 1 | 2 | 0 |
| total_count | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 |
| unique_count | 1639 | 1329 | 566 | 1019 | 1597 | 1552 | 765 | 2293 | 522 | 4134 | 576 | 1560 | 1621 | 6094 | 2418 |
| unique_percentage | 0.41% | 0.33% | 0.14% | 0.25% | 0.4% | 0.38% | 0.19% | 0.57% | 0.13% | 1.02% | 0.14% | 0.39% | 0.4% | 1.51% | 0.6% |
| missing_count | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| missing_percentage | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| zero_count | 197044 | 212652 | 212652 | 43868 | 0 | 37771 | 57644 | 0 | 305908 | 0 | 315315 | 0 | 0 | 0 | 33730 |
| zero_count_percentage | 48.85% | 52.72% | 52.72% | 10.88% | 0.0% | 9.36% | 14.29% | 0.0% | 75.84% | 0.0% | 78.17% | 0.0% | 0.0% | 0.0% | 8.36% |
| skew | 324.608 | 317.473 | 305.228 | 276.759 | 265.448 | 215.702 | 329.167 | 207.229 | 264.832 | 29.7613 | 108.492 | 6.75244 | 6.87641 | 311.828 | 16.9408 |

# Summary Statistics: String Variables

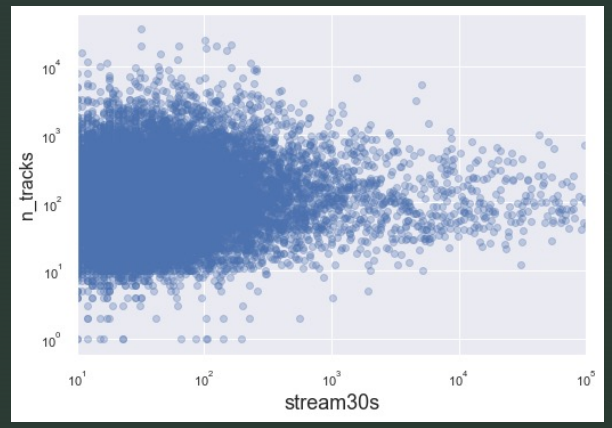
Variable names

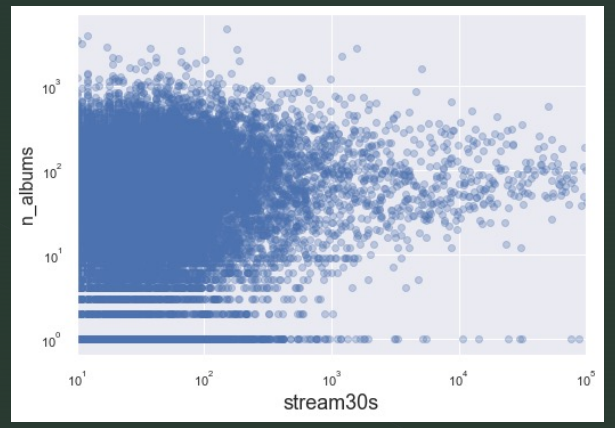
| | playlist_uri | owner | owner_country | tokens | genre_1 | genre_2 | genre_3 | mood_1 | mood_2 | mood_3 |
|---|---|---|---|---|---|---|---|---|---|---|
| max_value | spotify:user:ffffd7d68f005d7feafe85ef53e017a8:... | ffffd7d68f005d7feafe85ef53e017a8 | US | [zzzzzzzzzz] | Traditional | Traditional | Traditional | Yearning | Yearning | Yearning |
| min_value | spotify:user:00004454ac06ff1f42246d110b2d48ed:... | 00004454ac06ff1f42246d110b2d48ed | US | [\u00ba\u00ba\u00ba] | - | - | - | - | - | - |
| total_count | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 | 403366 |
| unique_count | 403366 | 314899 | 1 | 192107 | 26 | 26 | 26 | 27 | 27 | 27 |
| unique_percentage | 100.0% | 78.07% | 0.0% | 47.63% | 0.01% | 0.01% | 0.01% | 0.01% | 0.01% | 0.01% |
| missing_count | 0 | 0 | 0 | 0 | 944 | 21206 | 40123 | 1791 | 1868 | 2005 |
| missing_percentage | 0.0% | 0.0% | 0.0% | 0.0% | 0.23% | 5.26% | 9.95% | 0.44% | 0.46% | 0.5% |

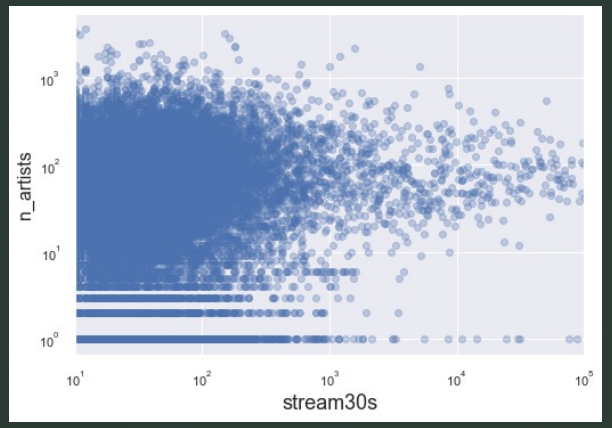
# Summary Statistics: Correlation Matrix

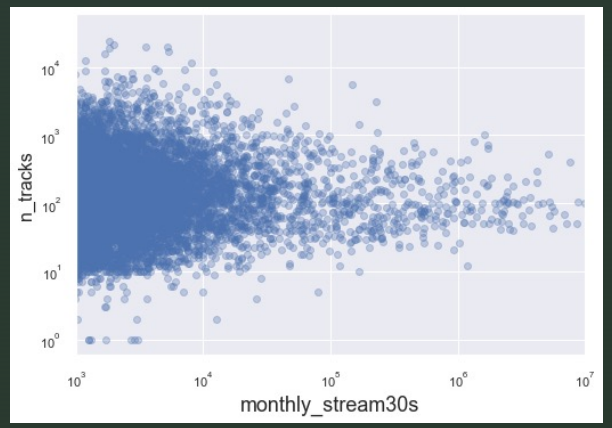


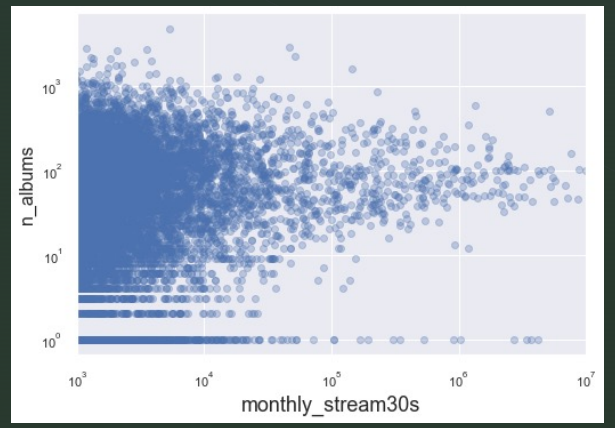
# Measuring success metrics with numeric variables

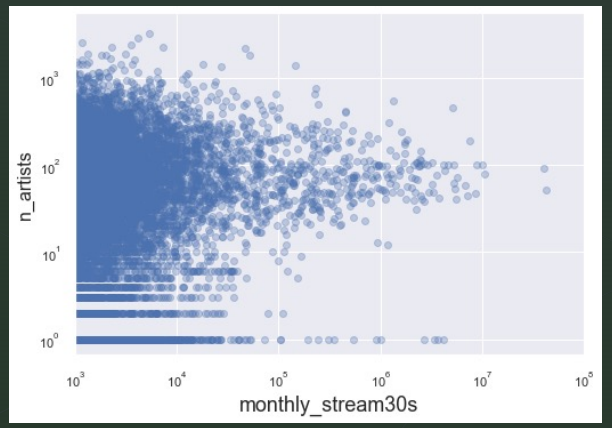


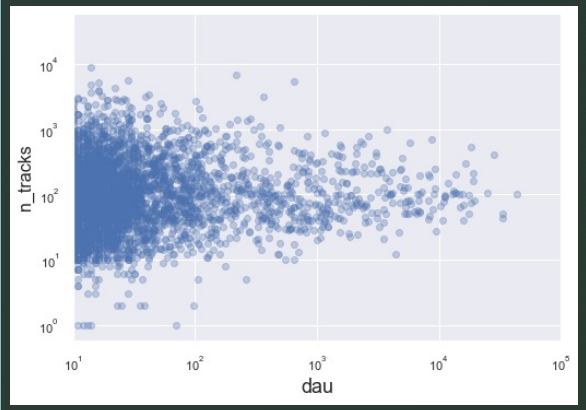



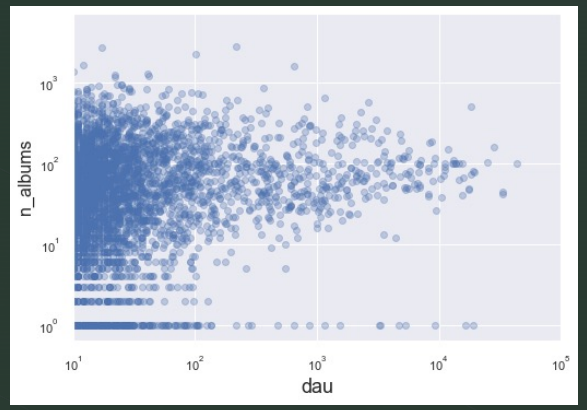



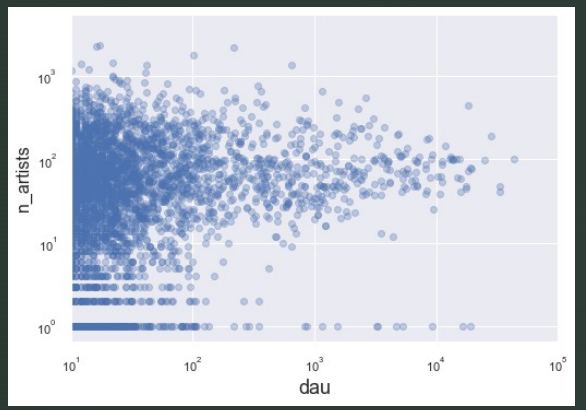



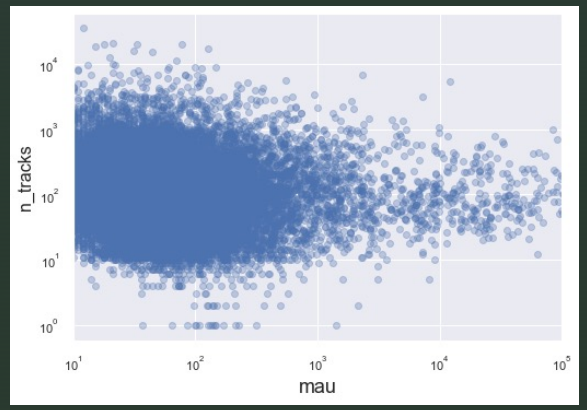



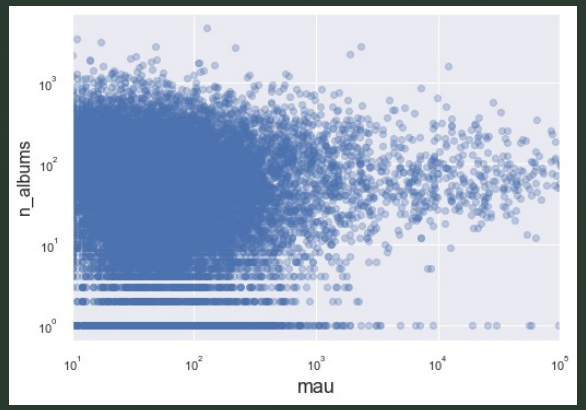
# Measuring success metrics with numeric variables

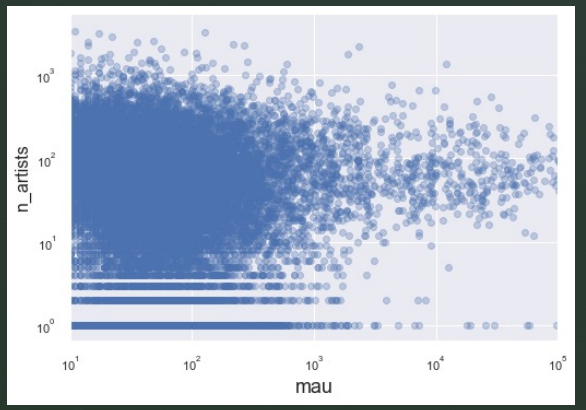


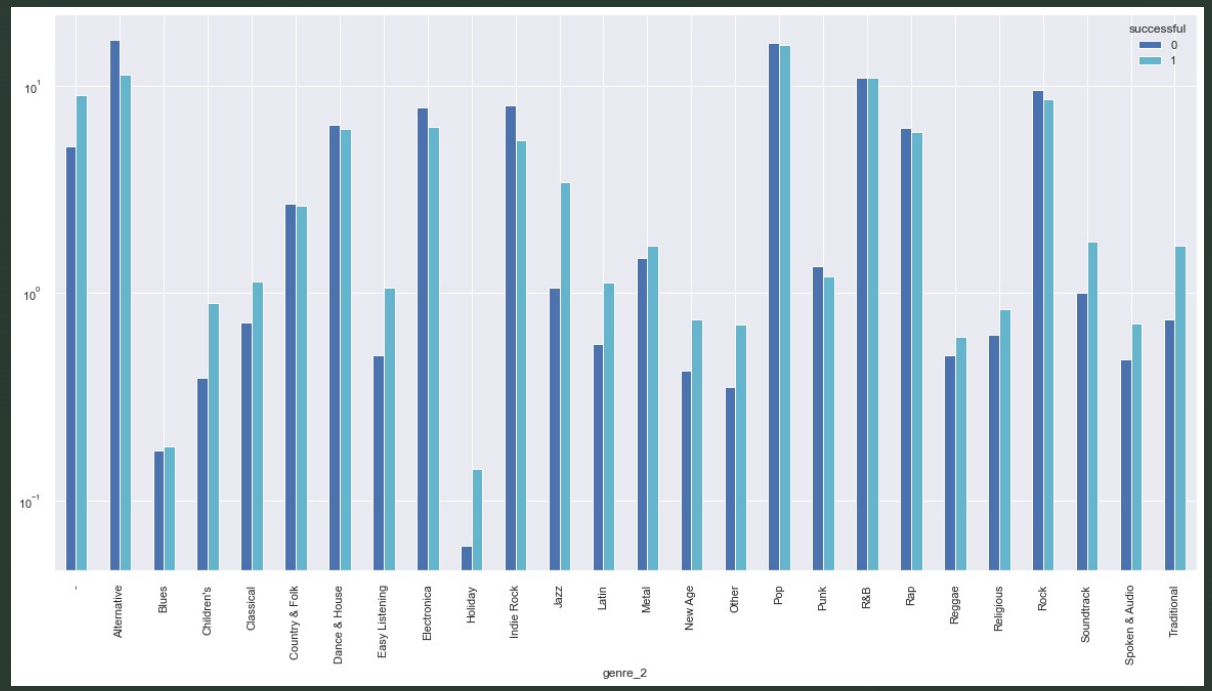



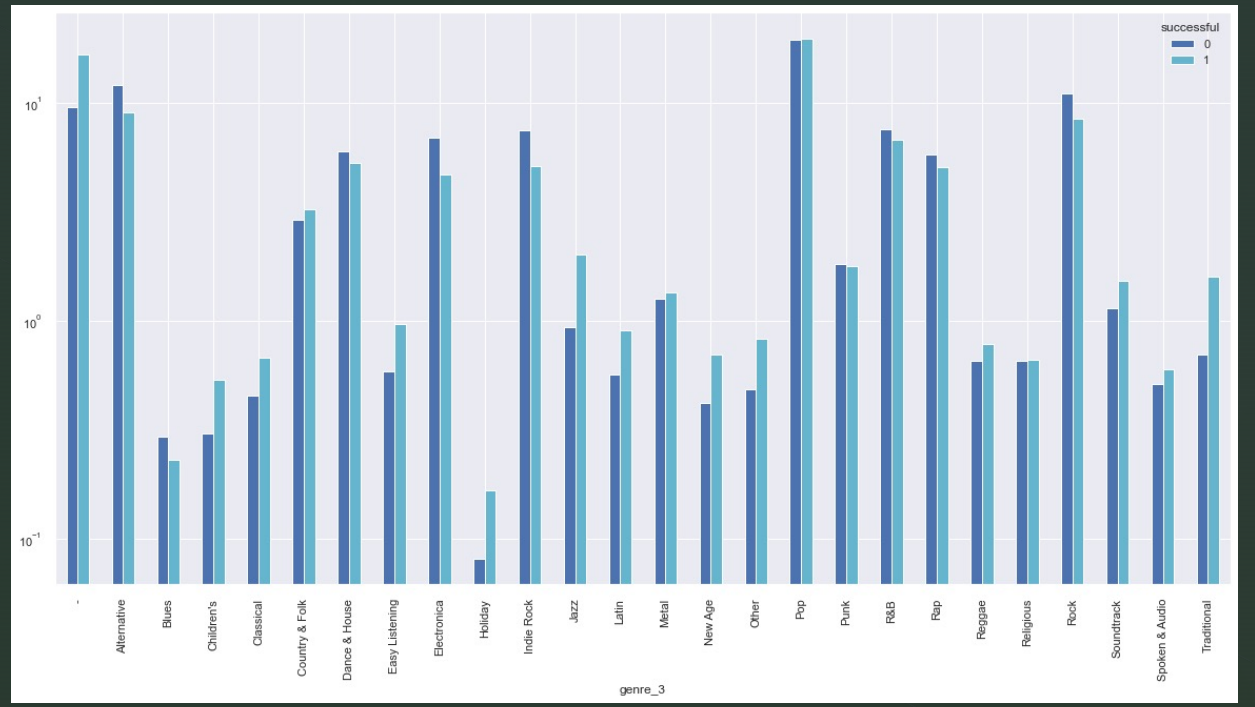



# Genres For Successful And Unsuccessful Playlists



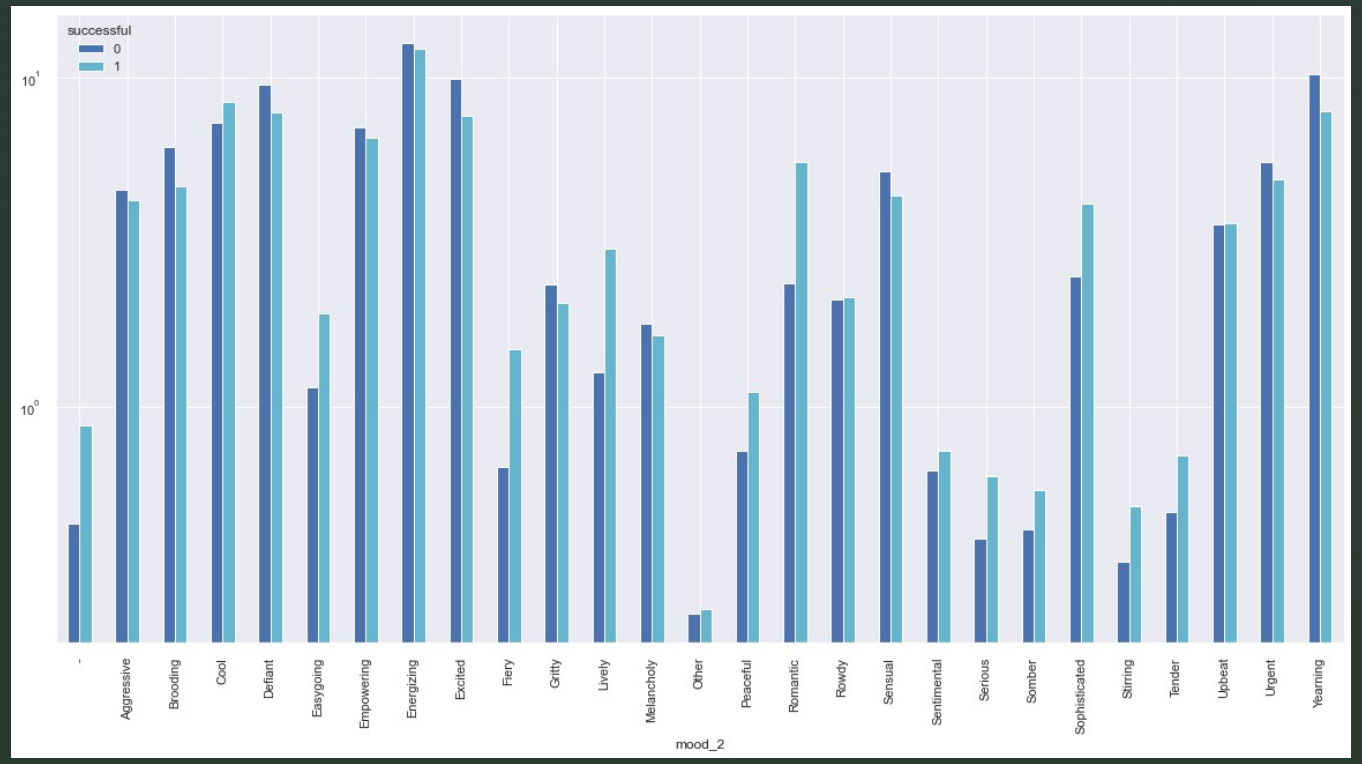

- genre 2: second highest weighted genre

# Genres For Successful And Unsuccessful Playlists



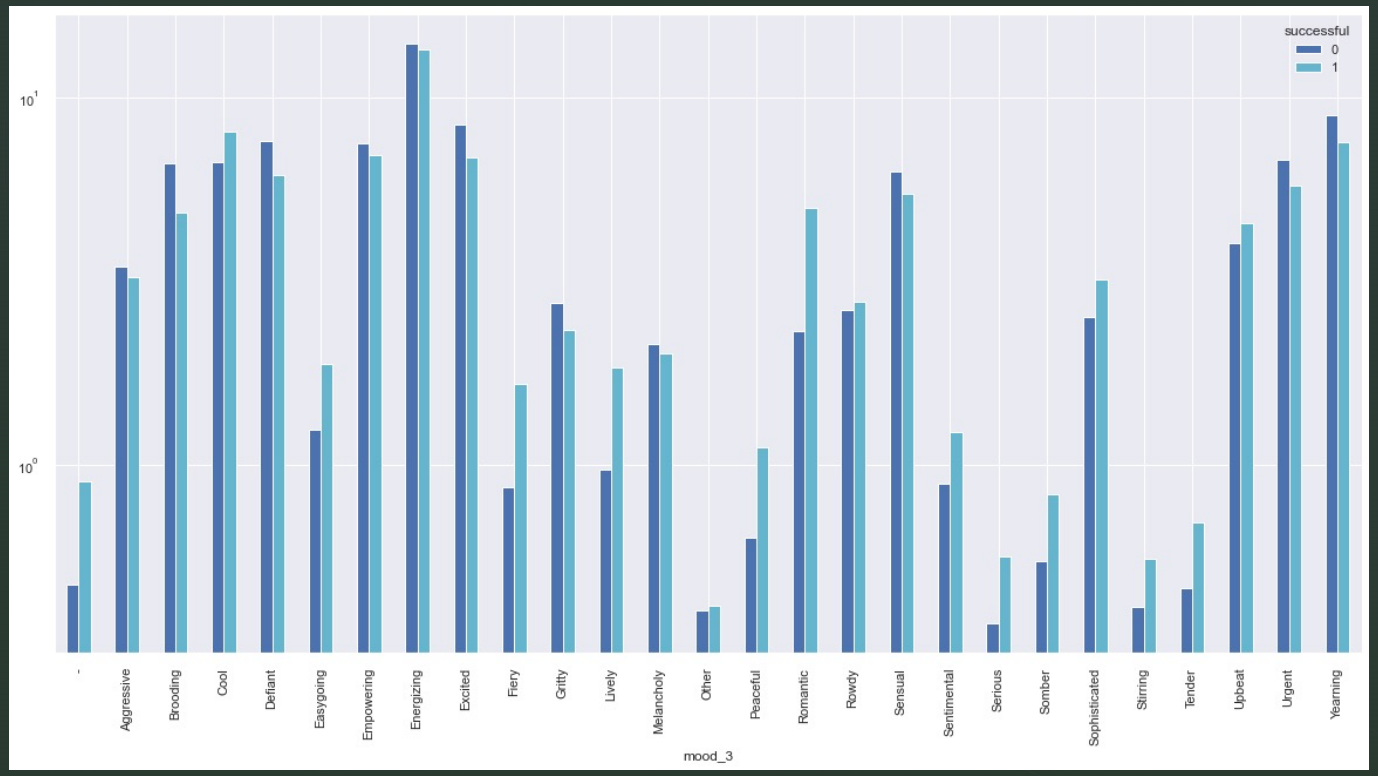

- genre 3: third highest weighted genre

# Moods For Successful And Unsuccessful Playlists



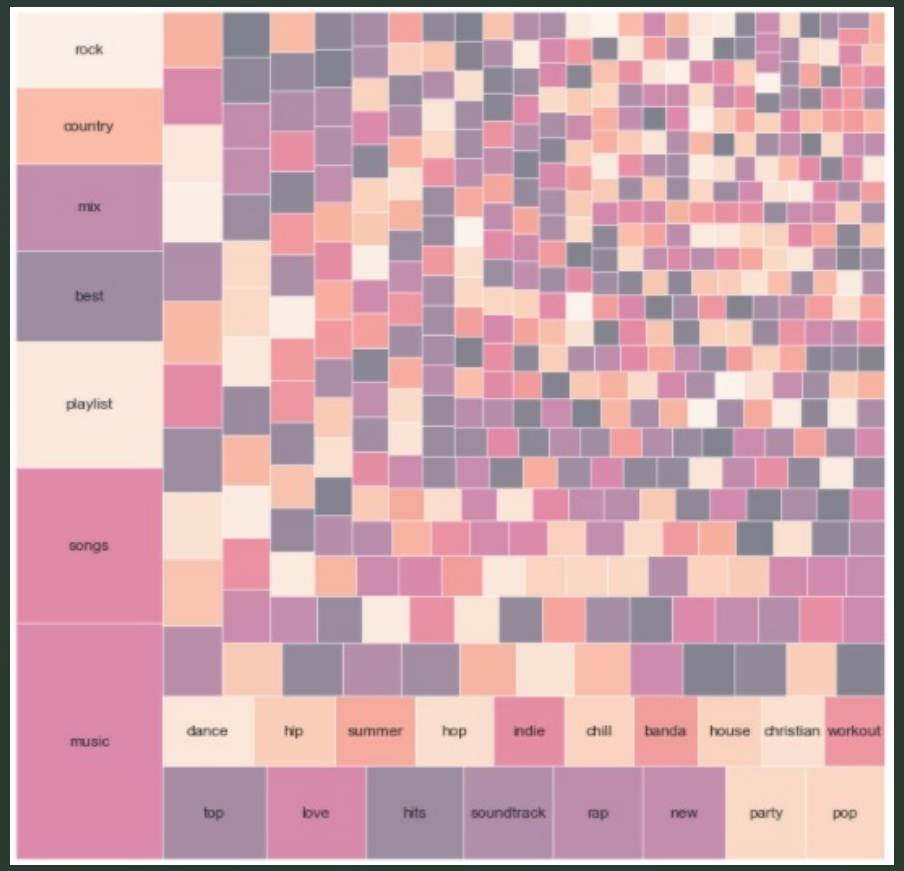

- mood 2: second highest weighted mood

# Moods For Successful And Unsuccessful Playlists

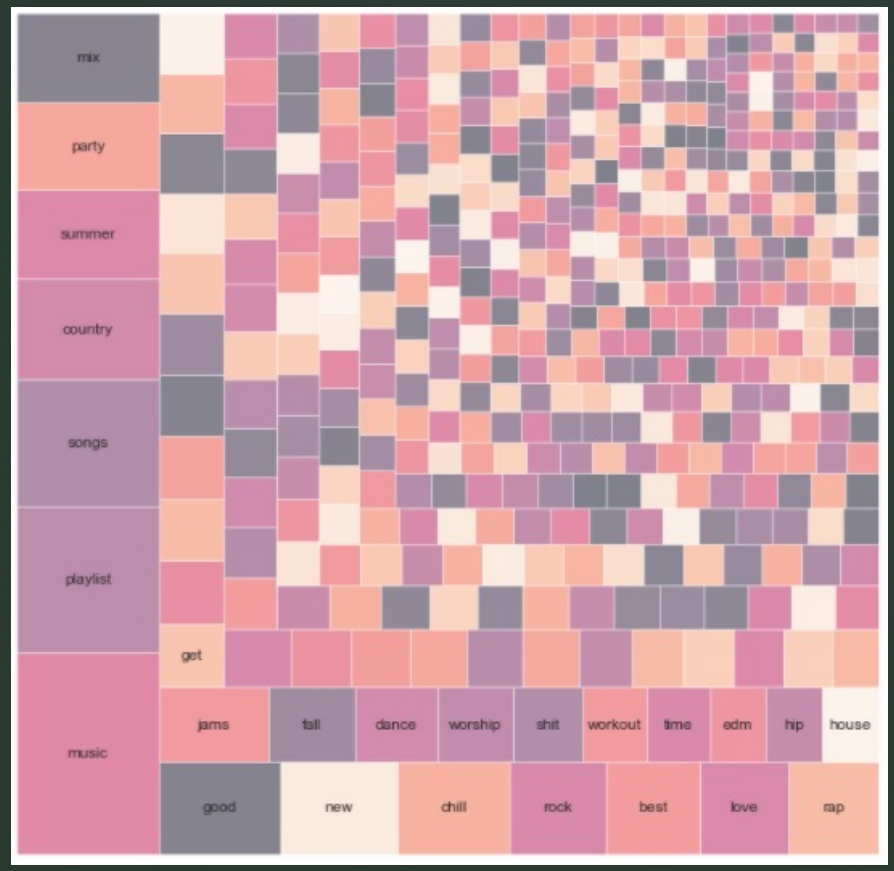



- mood 3: third highest weighted mood

# Playlist Tokens For Successful And Unsuccessful Playlists




Successful playlists




Unsuccessful playlists

# Playlist Owners For Successful And Unsuccessful Playlists

| | sucessful_owner_id | num_of_occurrences |
|---|---|---|
| 0 | 552c5ee1c8df46c75b6d473d68370189 | 392 |
| 1 | ad437dac3689492baba93a66205a2b04 | 36 |
| 2 | cb41ca2da57ecfd0381a5f2891735cc2 | 30 |
| 3 | 64559820ad024637e20047906adabeda | 29 |
| 4 | 8eecd7fb330e5217bfbe6c81967150e0 | 16 |
| 5 | 46bc904b127f9c0e6031c3f528497e38 | 15 |
| 6 | 9984b67a52aef608cdbbf066bd339e70 | 15 |
| 7 | e92ffbf848c7fda8675e7c0ec2b44929 | 13 |
| 8 | 526da0ac9890dbaa28309705f46364f6 | 13 |
| 9 | 0dce906b1b58e56c4e274d47066c650c | 12 |
| 10 | 80391b961385bfe15fa05748b230ee1b | 10 |

| | unsucessful_owner_id | num_of_occurrences |
|---|---|---|
| 0 | c126a48e5cd6f4846c457cb6ec435a6f | 44 |
| 1 | 1b62d3f6c6d274b8f80612e27d69d01f | 43 |
| 2 | 6addee40da677fe898ef1965cb679b15 | 40 |
| 3 | 0b5a1db0bb23f6395b8ee9d45d5ec45b | 40 |
| 4 | db61a268738eba333294c7a83d624709 | 36 |
| 5 | a38a804ea32c5ba94299b4d0181a7871 | 35 |
| 6 | 5cd51164a6828bc46ff335aa0ed6207a | 34 |
| 7 | e92ffbf848c7fda8675e7c0ec2b44929 | 30 |
| 8 | 917977c6b19c3862684aa519590d9594 | 29 |
| 9 | 81fa39edc05e7a7accb26507939b6491 | 29 |
| 10 | d9a8d67f6b711c928c798f6069b8c48b | 26 |

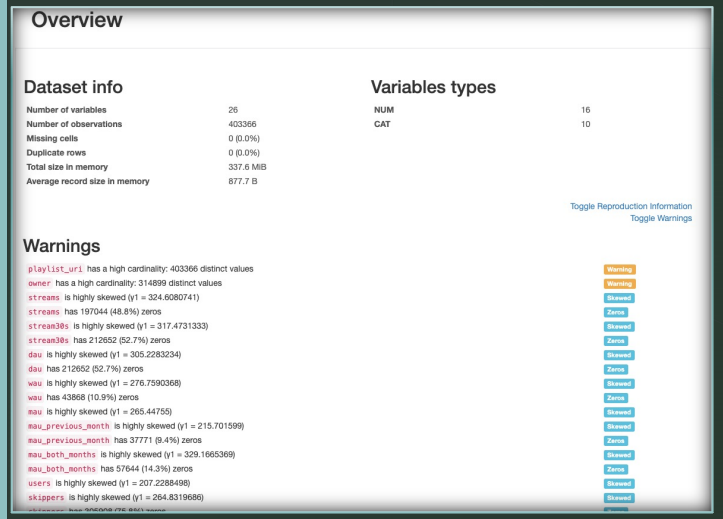
# Schema Design: Nested Data Structures

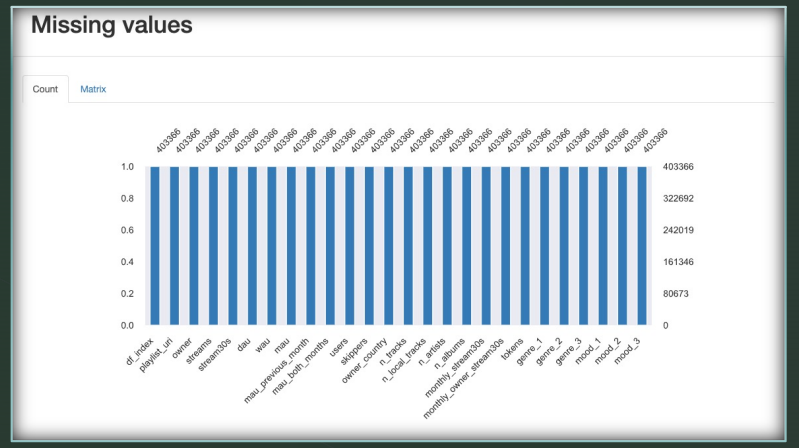
| spotify_playlist | datatype | description |
|---|---|---|
| playlist_id | string | Unique Id for playlist |
| playlist_name | string | play list name |
| upload_date | struct | upload date for the playlist |
| upload_date.upload_date | date time | date stamp |
| upload_date.hour | string | hour stamp |
| upload_date.day | string | day stamp |
| upload_date.week | string | week stamp |
| upload_date.month | string | month stamp |
| upload_date.year | string | year stamp |
| upload_location | struct | upload location of the playlist |
| upload_location.country | string | upload country |
| followers | int | number of followers/likes for the playlist |
| length | int | length of the playlist |
| owner | struct | owner_information |
| owner.owner_id | string | owner_id/uploaded_user_id, should be unique for each user |
| owner.first_name | string | owner first name |
| owner.last_name | string | owner last name |
| owner.owner_username | string | owner spotify username |
| owner.owner_country | string | owner country |
| owner.is_premium | bool | if the owner is a premium user or not |
| tracks | struct | tracks asscociated with the playlist |
| tracks.tracks_id | array | all track id's associated with the playlist |
| tracks.track_language | array | for each tack_id, track language |
| artists | struct | artist's associated with the track |
| artists.artist_id | int | artist_id's associated with track_id, sould relate with track information |
| artists.artists_name | string | artist's name associated to the artist_id |
| artists.artists_country | string | artist's country associated with artist_id |
| albums | struct | full album's associated with the playlist |
| album_id | array | distinct track_ids from album_id in tracks_id then album_id |
| lyrics | struct | each track that has lyrics associated we have lyrics_id |
| lyrics.lyrics_id | array | each lyric's id should be unique associated with the track id |
| user | struct | user information consuming the playlist |
| user.user_id | array | should be unique per user id who interacts with the playlist |
| start_time | struct | user start listening time of the playlist |
| start_time.start_time | date time | date stamp |
| start_time.hour | string | hour stamp |
| start_time.day | string | day stamp |
| start_time.week | string | week stamp |
| start_time.month | string | month stamp |
| start_time.year | string | year stamp |
| end_time | struct | user stop listening time of the playlist |
| end_time.end_time | date time | date stamp |
| end_time.hour | string | hour stamp |
| end_time.day | string | day stamp |
| end_time.week | string | week stamp |
| end_time.month | string | month stamp |
| end_time.year | string | year stamp |

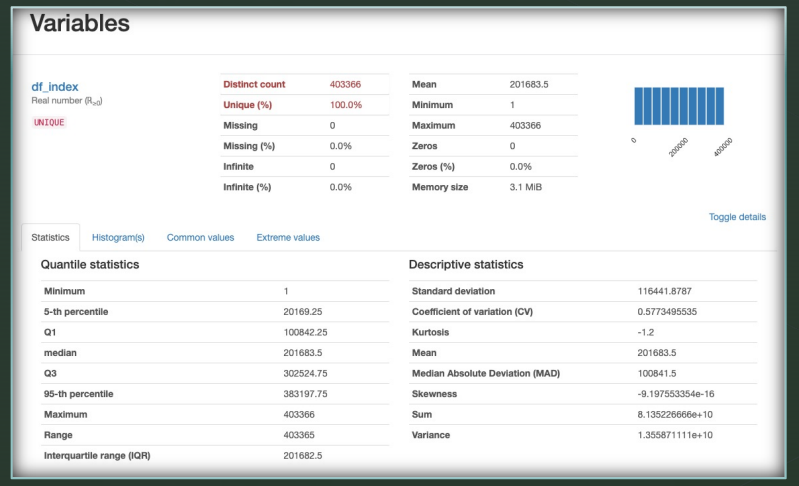
Integrate lyrics data, which currently exists as 1 text file per track. How would you structure this dataset
to allow data scientists to, in turn, flexibly perform analysis on how lyrics affect playlist performance?

- 23. Schema design : From this slide, the lyrics data has been integrated with tracks table.
- Assuming we do not need the text from the lyrics file.
- We create a unique id for each file and extract the file name.
- Each file per track can be associated with its respective track_id.
- We can aggregate this to the playlist level and create a new column as n_lyrics for each track_id that has an lyrics_id we can indicate True/False or 1/0 and count the number of True occurrences.
- E.g., playlist_id: 1234  n_tracks: 10  n_lyrics: 5
- n_lyrics can be used to check the influence on successful playlists.

# Data Profile Report




- An Interactive version of the profile report has been emailed separately.

# Summary

- Day of the week the daily metrics are collected will help understand usage of playlists at a granular level.
- Upload date can be used to check how old the playlist is and if there has been any activity on the playlist in recent time.
- Information like playlist created by Spotify vs users will help understanding the the group creating successful playlists and further analysis can be done to learn what factors are contributing to the success of playlists created by a particular group.
- Knowing number of likes the playlist can be added to the success metrics.
- Data from different countries will help understand usage and engagement with playlists, further analysis can be performed on how variables like local tracks and local artists are impacting to the success of playlists.
- Understanding where the playlists are placed in the product can also play a role. E.g.: is the playlist suggested on the home screen or if the user must search for the playlist.