EDA ON 30,000 SPOTIFY SONGS

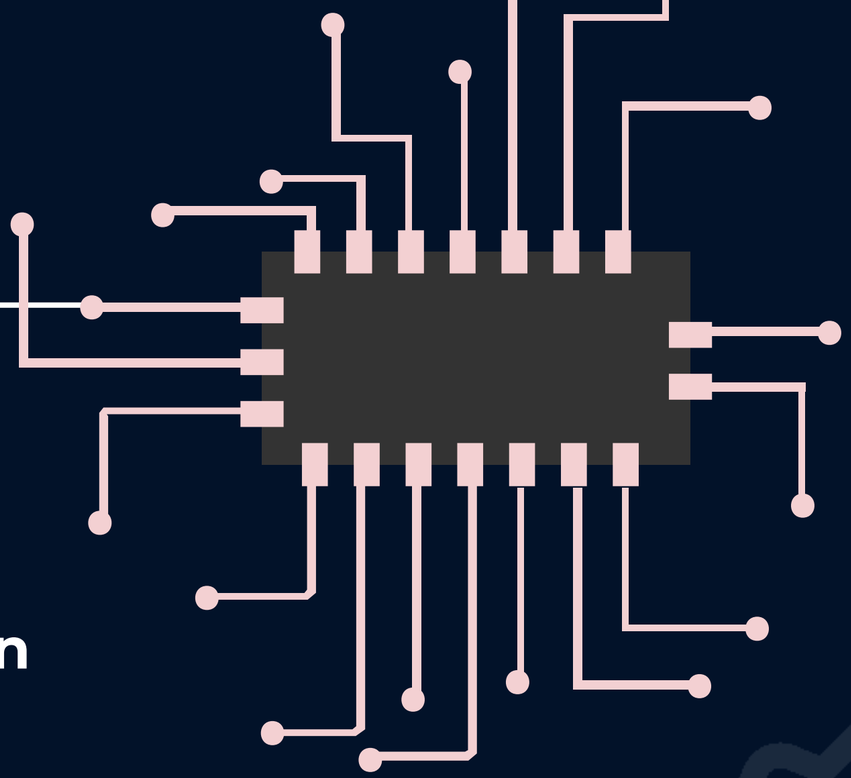
PRESENTATION BY GROUP 17

MEET OUR TEAM

21BAI10006	Deekshit Kashyap
21BAI10114	Devang Giri Goswami
21BAI10308	Sannidhya Srivastava
21BAI10339	Shubham Shailey
21BAI10118	Kshitij Singh

Data Visualization (A11+A12)

TOOLS USED



01

Dataset

Kaggle

02

Data Vizualization

Plotly

03

Coding

Python

04

Streaming & Hosting

Static.app

STEP 01

IMPORTING LIBRARIES

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
```

```
#loading dataset
df = pd.read_csv('spotify_songs.csv')
```

STEP 02

ANALYSING THE DATA SET

```
df.shape
```

```
(32833, 23)
```

```
df.head()
```

	track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date	playlist_name	playlist_id	playlist_genre	...	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms
0	6f807x0ima9a1j3VPbc7VN	I Don't Care (with Justin Bieber) - Loud Luxur...	Ed Sheeran	66	2oCs0DGTsRO98Gh5ZSl2Cx	I Don't Care (with Justin Bieber) [Loud Luxury...	2019-06-14	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop	...	6	-2.634	1	0.0583	0.1020	0.000000	0.0653	0.518	122.036	194754
1	0r7CVbZTWZgbTCYdfa2P31	Memories - Dillon Francis Remix	Maroon 5	67	63rPSO264uRjW1X5E6cWv6	Memories (Dillon Francis Remix)	2019-12-13	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop	...	11	-4.969	1	0.0373	0.0724	0.004210	0.3570	0.693	99.972	162600
2	1z1Hg7Vb0AhHDiEmnDE79l	All the Time - Don Diablo Remix	Zara Larsson	70	1HoSmj2eLcsrR0vE9gThr4	All the Time (Don Diablo Remix)	2019-07-05	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop	...	1	-3.432	0	0.0742	0.0794	0.000023	0.1100	0.613	124.008	176616
3	75FpbthrwQmzHlBJLuGdC7	Call You Mine - Keanu Silva Remix	The Chainsmokers	60	1nqYsOef1yKKuGOVchbsk6	Call You Mine - The Remixes	2019-07-19	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop	...	7	-3.778	1	0.1020	0.0287	0.000009	0.2040	0.277	121.956	169093
4	1e8PAfcKUYoKkxPhrHqw4x	Someone You Loved - Future Humans Remix	Lewis Capaldi	69	7m7vv9wlQ4i0LFuJiE2zsQ	Someone You Loved (Future Humans Remix)	2019-03-05	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop	...	1	-4.672	1	0.0359	0.0803	0.000000	0.0833	0.725	123.976	189052

STEP 02

ANALYSING THE DATA SET

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32833 entries, 0 to 32832
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   track_id                  32833 non-null  object 
 1   track_name                32828 non-null  object 
 2   track_artist              32828 non-null  object 
 3   track_popularity          32833 non-null  int64  
 4   track_album_id            32833 non-null  object 
 5   track_album_name          32828 non-null  object 
 6   track_album_release_date  32833 non-null  object 
 7   playlist_name             32833 non-null  object 
 8   playlist_id               32833 non-null  object 
 9   playlist_genre            32833 non-null  object 
 10  playlist_subgenre         32833 non-null  object 
 11  danceability              32833 non-null  float64
 12  energy                    32833 non-null  float64
 13  key                       32833 non-null  int64  
 14  loudness                  32833 non-null  float64
 15  mode                      32833 non-null  int64  
 16  speechiness               32833 non-null  float64
 17  acousticness              32833 non-null  float64
 18  instrumentalness          32833 non-null  float64
 19  liveness                  32833 non-null  float64
 20  valence                   32833 non-null  float64
 21  tempo                     32833 non-null  float64
 22  duration_ms               32833 non-null  int64  
dtypes: float64(9), int64(4), object(10)
memory usage: 5.8+ MB
```

STEP 02

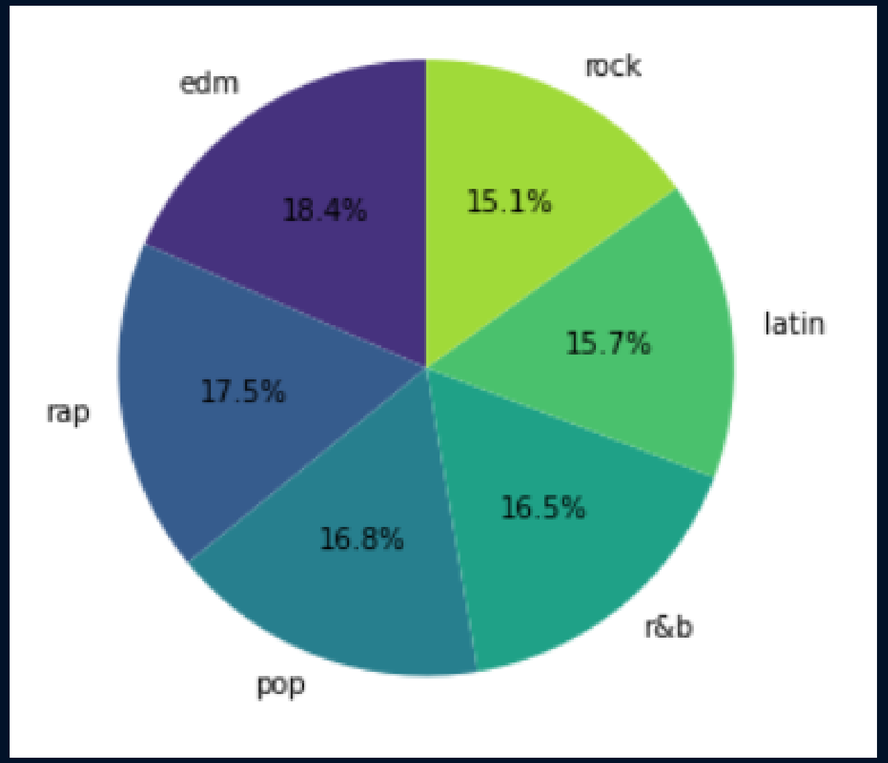
ANALYSING THE DATA SET

	track_popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms
count	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000
mean	42.477081	0.654850	0.698619	5.374471	-6.719499	0.565711	0.107068	0.175334	0.084747	0.190176	0.510561	120.881132	225799.811622
std	24.984074	0.145085	0.180910	3.611657	2.988436	0.495671	0.101314	0.219633	0.224230	0.154317	0.233146	26.903624	59834.006182
min	0.000000	0.000000	0.000175	0.000000	-46.448000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	4000.000000
25%	24.000000	0.563000	0.581000	2.000000	-8.171000	0.000000	0.041000	0.015100	0.000000	0.092700	0.331000	99.960000	187819.000000
50%	45.000000	0.672000	0.721000	6.000000	-6.166000	1.000000	0.062500	0.080400	0.000016	0.127000	0.512000	121.984000	216000.000000
75%	62.000000	0.761000	0.840000	9.000000	-4.645000	1.000000	0.132000	0.255000	0.004830	0.248000	0.693000	133.918000	253585.000000
max	100.000000	0.983000	1.000000	11.000000	1.275000	1.000000	0.918000	0.994000	0.994000	0.996000	0.991000	239.440000	517810.000000

STEP 03

GENRE DISTRIBUTION

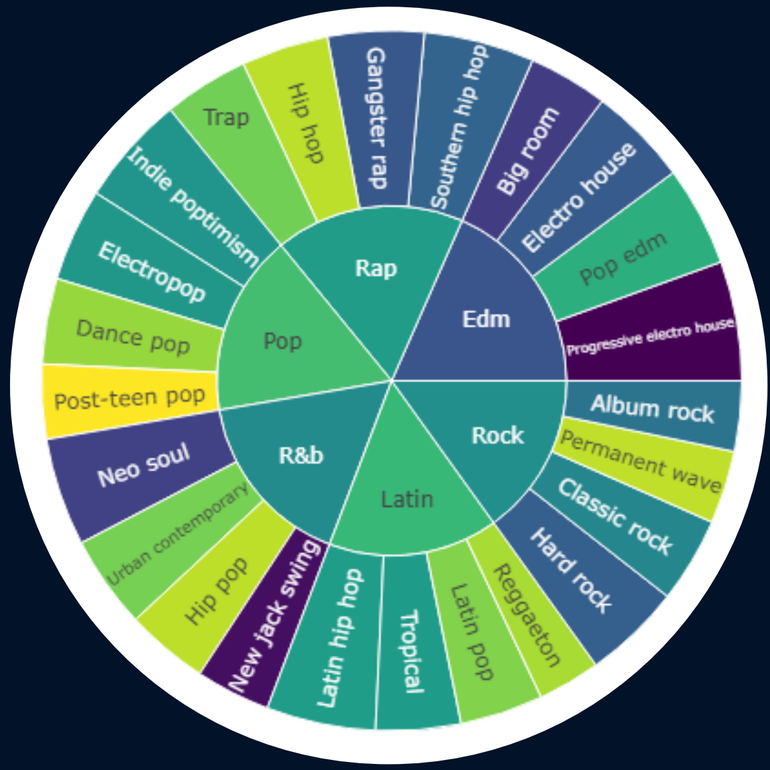
```
genre_counts = df['playlist_genre'].value_counts()
plt.figure(figsize=(5, 5))
plt.pie(genre_counts, labels=genre_counts.index, autopct='%1.1f%%', startangle=90, colors=sns.color_palette('viridis'))
plt.title('Genre Distribution in Playlists')
plt.show()
```



STEP 03

SUB-GENRE DISTRIBUTION

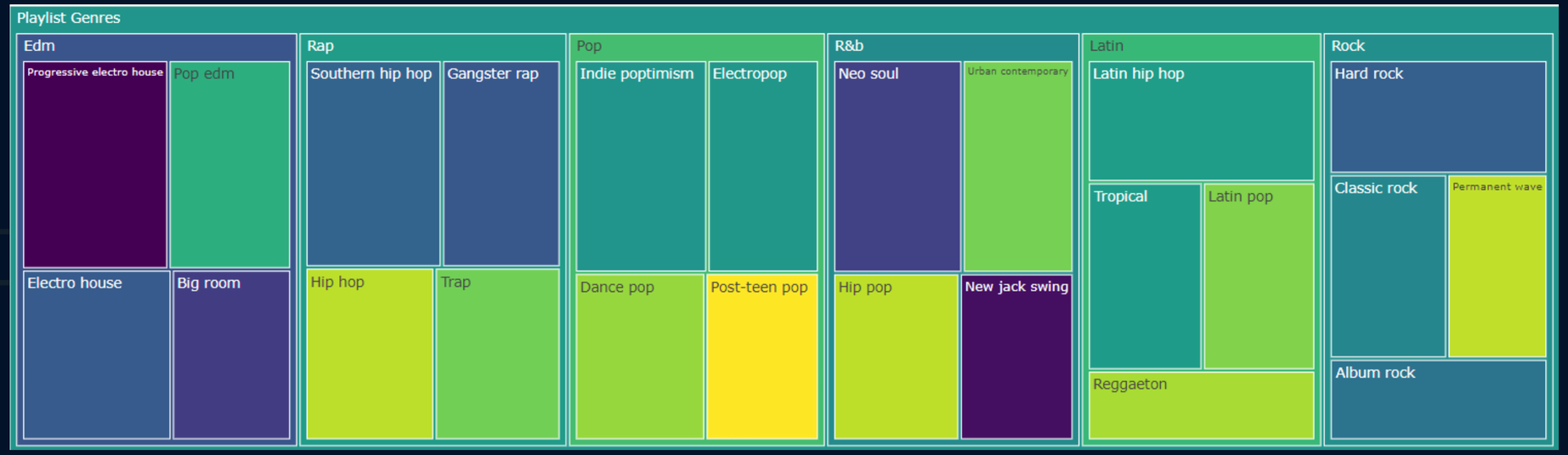
```
df[['playlist_genre', 'playlist_subgenre']] = df[['playlist_genre', 'playlist_subgenre']] \
                                                .apply(lambda x: x.str.capitalize(), axis=1)
fig = px.sunburst(df,
                  path=['playlist_genre', 'playlist_subgenre'],
                  color='track_popularity',
                  color_continuous_scale='viridis',
                  labels={'track_popularity'  'Popularity'})
fig.show()
```



STEP 03

SUB-GENRE DISTRIBUTION

```
fig = px.treemap(df.groupby(['playlist_genre', 'playlist_subgenre']).track_popularity.agg(['count', 'mean']).reset_index(),
                 path=[px.Constant('Playlist Genres'), 'playlist_genre', 'playlist_subgenre'],
                 values='count',
                 color='mean',
                 color_continuous_scale='viridis',
                 labels={'mean': 'Popularity'},
                )
fig.show()
```



STEP 03

SUB-GENRE DISTRIBUTION

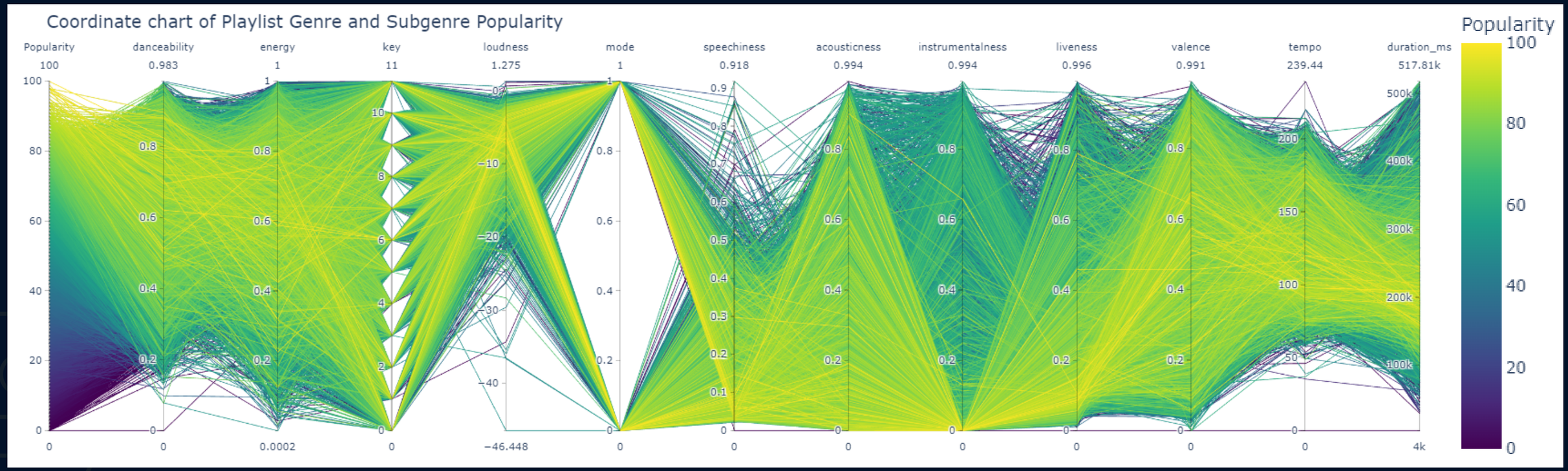
```
# Assuming you have a DataFrame df with columns 'playlist_genre', 'playlist_subgenre', 'track_popularity'
df[['playlist_genre', 'playlist_subgenre']] = df[['playlist_genre', 'playlist_subgenre']] \
    .apply(lambda x: x.str.capitalize(), axis=1)


# Parallel Coordinates Plot
parallel_coordinates_fig = px.parallel_coordinates(df, color='track_popularity',
                                                   color_continuous_scale='viridis',
                                                   title='Coordinate chart of Playlist Genre and Subgenre Popularity',
                                                   labels={'track_popularity': 'Popularity'})


parallel_coordinates_fig.show()
```

STEP 04

COORDINATE CHART OF PLAYLIST GENRE AND SUBGENRE POPULARITY



STEP 04

COORDINATE CHART OF PLAYLIST GENRE AND SUBGENRE POPULARITY

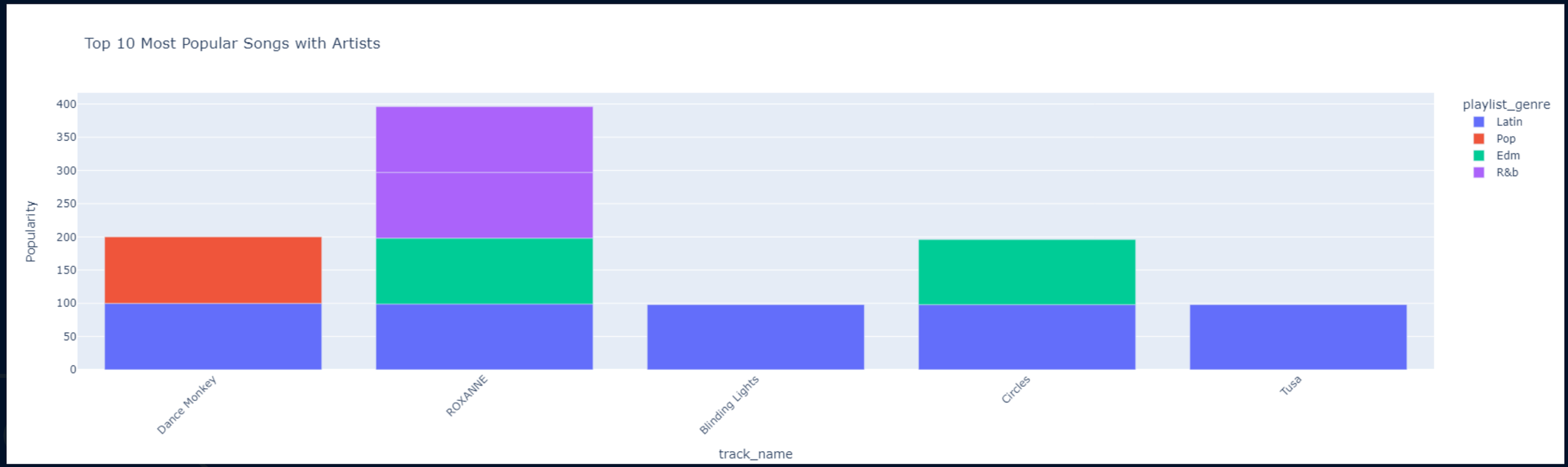
```
# Assuming you have a DataFrame df with the given structure
df[['playlist_genre', 'playlist_subgenre']] = df[['playlist_genre', 'playlist_subgenre']] \
    .apply(lambda x: x.str.capitalize(), axis=1)

# Create a grouped bar chart
fig = px.bar(df, x='playlist_genre', y='track_popularity', color='playlist_subgenre',
             title='Grouped Bar Chart of Playlist Genre and Subgenre Popularity',
             labels={'track_popularity': 'Popularity'})

# Show the grouped bar chart
fig.show()
```

STEP 05

CONCLUSION



CONCLUSION

1. Genre Distribution
2. Sub-Genre and Playlist Popularity
3. Artist and Track Popularity
4. Overall Genre Popularity:
    - Pop and Latin genres tend to be more popular among the analyzed playlists.
    - Electronic Dance Music (EDM) appears to be the less popular genre.
5. Top Trending Songs:
    - The analysis of the top 10 most popular songs indicates the names and popularity of these songs, providing a snapshot of the current trends on Spotify.

In summary, this EDA provides a comprehensive understanding of the dataset, revealing patterns and trends within the Spotify song data. These insights can be valuable for music enthusiasts, industry professionals, and anyone interested in understanding the factors influencing the popularity of songs on the platform.

THANK YOU

VISIT OUR WEBSITE FOR INTERACTIVE GRAPHS

https://spotify-eda.static.domains

SCAN ME