SPOTIFY SONG RECOMMENDATION MINI PROJECT

- **Objective:** Cluster Spotify songs by audio features and build a recommendation system.
- **Pre-processing:** Removed missing values and duplicates; scaled numeric audio features.
- **EDA & Visualization:** Plots for popularity, danceability, energy, valence; top genres; feature correlation.
- **Clustering:** K-Means (k=5) with PCA for 2D cluster visualization; analyzed cluster features and top artists.
- **Recommendation System:** Suggests similar songs from the same cluster with random sampling.
- **Outcome:** Visual insights, cluster summaries, and actionable song recommendations.

Program code -

```
# Spotify Songs Genre Segmentation and Recommendation System

# ----------------------------
# Import Required Libraries
# ----------------------------
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# ----------------------------
# Load Dataset
# ----------------------------
dataset_path = 'spotify.csv'
try:
    spotify_df = pd.read_csv(dataset_path)
    print("Spotify dataset loaded successfully!\n")
except FileNotFoundError:
    print(f"Error: '{dataset_path}' not found.")
    exit()

# ----------------------------
# Explore Dataset
```

```
# ----------------------------
print(spotify_df.head())
print("\nDataset Info:")
print(spotify_df.info())
print("\nStatistical Summary:")
print(spotify_df.describe())

# ----------------------------
# Data Pre-processing
# ----------------------------
# Remove missing values
spotify_df.dropna(inplace=True)

# Remove duplicates based on track_name
spotify_df.drop_duplicates(subset=['track_name'], inplace=True)
print(f"\nUnique songs retained: {len(spotify_df)}")

# Select numeric audio features for clustering
features = ['acousticness', 'danceability', 'energy', 'instrumentalness',
            'liveness', 'loudness', 'speechiness', 'tempo', 'valence']
X = spotify_df[features]

# Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
print("\nFeature scaling completed!")

# ----------------------------
# Data Visualization
# ----------------------------
plt.figure(figsize=(20,5))

# 1. Track Popularity
plt.subplot(1, 4, 1)
sns.histplot(spotify_df['track_popularity'], kde=True, color='blue')
plt.title('Track Popularity')
plt.xlabel('Popularity')
plt.ylabel('Frequency')

# 2. Danceability
plt.subplot(1, 4, 2)
sns.histplot(spotify_df['danceability'], kde=True, color='purple')
plt.title('Danceability')
plt.xlabel('Danceability')
plt.ylabel('Frequency')

# 3. Energy
plt.subplot(1, 4, 3)
```

```
sns.histplot(spotify_df['energy'], kde=True, color='purple')
plt.title('Energy')
plt.xlabel('Energy')
plt.ylabel('Frequency')

# 4. Valence
plt.subplot(1, 4, 4)
sns.histplot(spotify_df['valence'], kde=True, color='purple')
plt.title('Valence')
plt.xlabel('Valence')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()

# Top 10 Playlist Genres
plt.figure(figsize=(12, 6))
top_genres = spotify_df['playlist_genre'].value_counts().nlargest(10)
sns.barplot(x=top_genres.values, y=top_genres.index, palette='viridis')
plt.title('Top 10 Playlist Genres')
plt.xlabel('Number of Tracks')
plt.ylabel('Genre')
plt.show()

# Correlation Matrix of Audio Features
plt.figure(figsize=(10, 8))
sns.heatmap(spotify_df[features].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Audio Features')
plt.show()

# ----------------------------
# Clustering using K-Means
# ----------------------------
optimal_k = 5
kmeans = KMeans(n_clusters=optimal_k, init='k-means++', random_state=42)
spotify_df['cluster'] = kmeans.fit_predict(X_scaled)
print(f"\nK-Means clustering applied with {optimal_k} clusters.")

# PCA Visualization
pca = PCA(n_components=2)
pca_result = pca.fit_transform(X_scaled)
pca_df = pd.DataFrame(pca_result, columns=['PCA1', 'PCA2'])
pca_df['cluster'] = spotify_df['cluster']

plt.figure(figsize=(10, 7))
sns.scatterplot(data=pca_df, x='PCA1', y='PCA2', hue='cluster',
palette='Spectral', s=70, alpha=0.8)
plt.title('2D PCA Visualization of Song Clusters')
```

```
plt.legend(title='Cluster')
plt.show()

# ----------------------------
# Cluster-wise Genre Distribution
# ----------------------------
plt.figure(figsize=(12, 8))
for i in range(optimal_k):
    plt.subplot(2, 3, i+1)
    cluster_data = spotify_df[spotify_df['cluster'] == i]
    top_genres_cluster =
cluster_data['playlist_genre'].value_counts().nlargest(5)
    sns.barplot(x=top_genres_cluster.values, y=top_genres_cluster.index,
palette='magma')
    plt.title(f'Cluster {i} Top Genres')
    plt.xlabel('Number of Tracks')
    plt.ylabel('Genre')
plt.tight_layout()
plt.show()

# ----------------------------
# Cluster-wise Playlist Distribution
# ----------------------------
plt.figure(figsize=(12, 8))
for i in range(optimal_k):
    plt.subplot(2, 3, i+1)
    cluster_data = spotify_df[spotify_df['cluster'] == i]
    top_playlists_cluster =
cluster_data['playlist_name'].value_counts().nlargest(5)
    sns.barplot(x=top_playlists_cluster.values, y=top_playlists_cluster.index,
palette='cool')
    plt.title(f'Cluster {i} Top Playlists')
    plt.xlabel('Number of Tracks')
    plt.ylabel('Playlist Name')
plt.tight_layout()
plt.show()

# ----------------------------
# Cluster Analysis
# ----------------------------
cluster_features_avg = spotify_df.groupby('cluster')[features].mean()
print("\nAverage Audio Features per Cluster:")
print(cluster_features_avg)

if 'duration_ms' in spotify_df.columns:
    spotify_df['duration_min'] = spotify_df['duration_ms'] / 60000
    cluster_duration = spotify_df.groupby('cluster')['duration_min'].mean()
    print("\nAverage Song Duration (minutes) per Cluster:")
```

```
    print(cluster_duration)

top_artists = spotify_df.groupby('cluster')['track_artist'].apply(lambda x:
x.value_counts().head(3))
print("\nTop Artists per Cluster:")
print(top_artists)

# ----------------------------
# Recommendation Function
# ----------------------------
def recommend_songs(song_title, data=spotify_df, num_suggestions=5):
    if song_title not in data['track_name'].values:
        return f"'{song_title}' not found in dataset."

    cluster_id = data.loc[data['track_name'] == song_title, 'cluster'].iloc[0]
    similar_songs = data[(data['cluster'] == cluster_id) & (data['track_name']
!= song_title)]

    if similar_songs.empty:
        return "No other songs found in the same cluster."

    similar_songs = similar_songs.reset_index(drop=True)

    return similar_songs.sample(n=min(num_suggestions, len(similar_songs)),
random_state=None)[
        ['track_name', 'track_artist', 'playlist_genre']]

# ----------------------------
# Example Recommendations
# ----------------------------
examples = ["bad guy", "Bohemian Rhapsody - Remastered 2011"]
for song in examples:
    print(f"\nRecommendations for '{song}':")
    print(recommend_songs(song))

# Random recommendations multiple times
for i in range(3):
    print(f"\nRandom Recommendations Attempt {i+1}:")
    print(recommend_songs("I Don't Care (with Justin Bieber) - Loud Luxury
Remix"))

# ----------------------------
# End of Project
# ----------------------------
print("\nProject Execution Completed Successfully!")
```

Output-

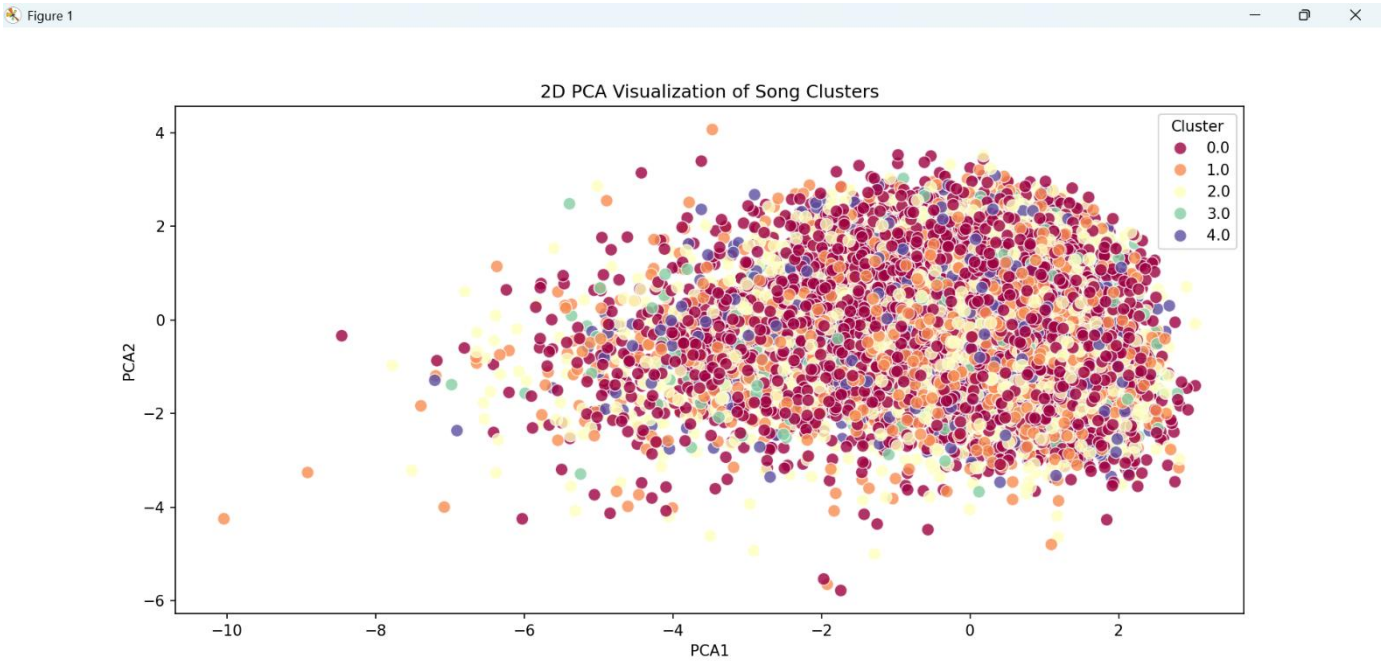


Figure 1
2D PCA Visualization of Song Clusters
Cluster
0.0
1.0
2.0
3.0
4.0
PCA2
PCA1

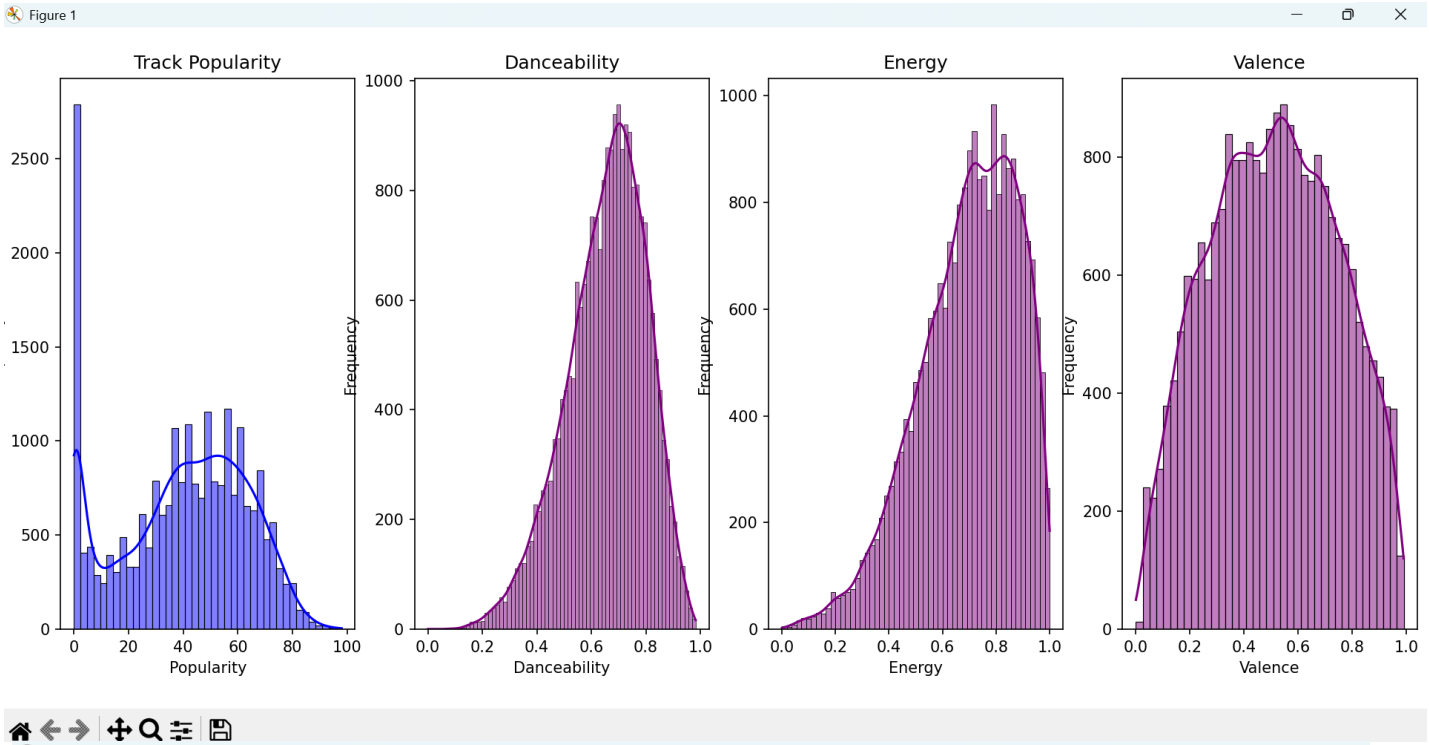


Figure 1
Track Popularity
Danceability
Energy
Valence
Frequency
Popularity
Danceability
Energy
Valence

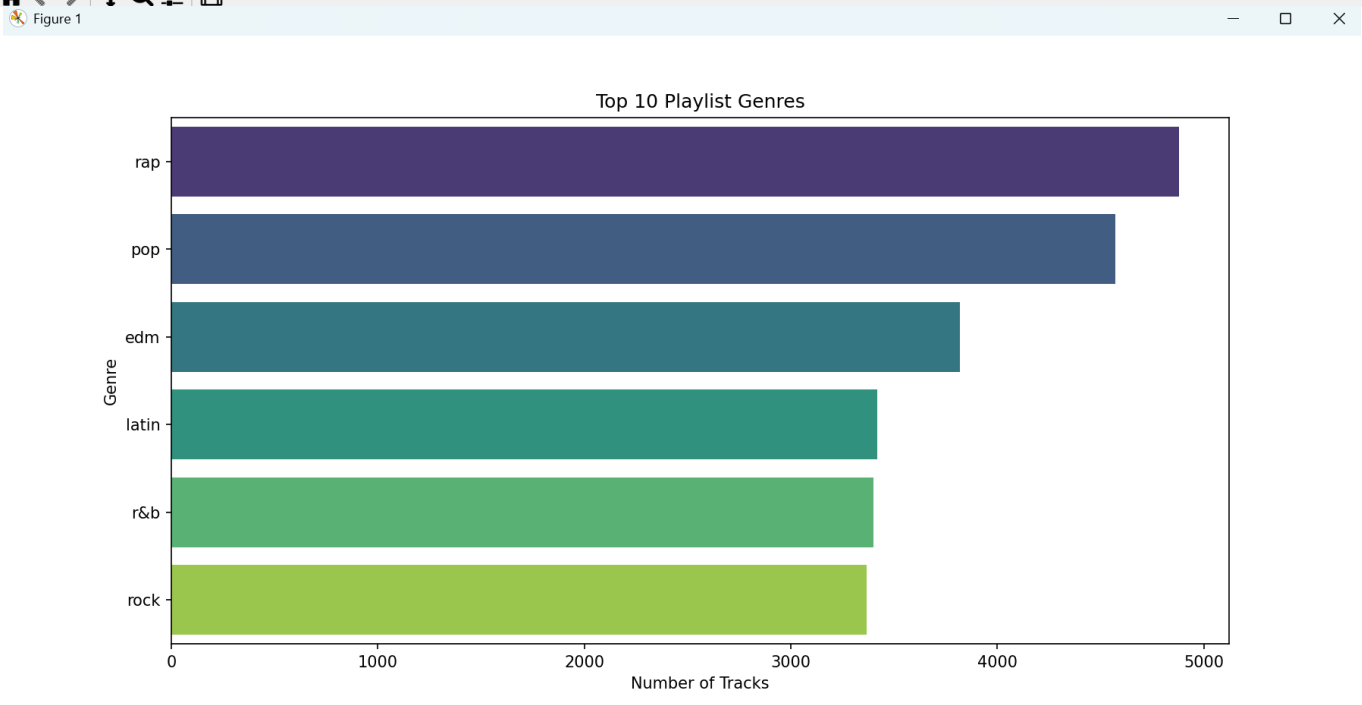


Figure 1
Top 10 Playlist Genres
rap
pop
edm
latin
r&b
rock
Genre
Number of Tracks

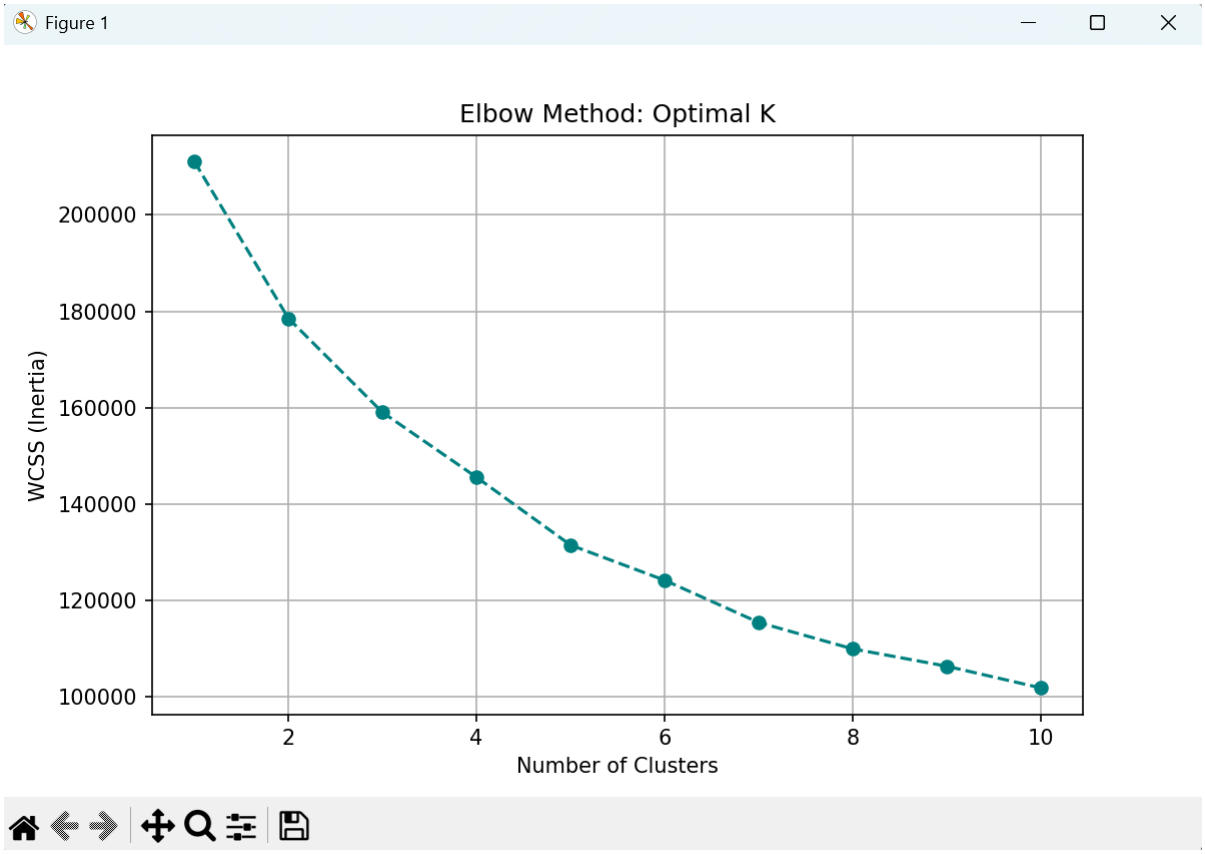


Figure 1
Elbow Method: Optimal K
WCSS (Inertia)
200000
180000
160000
140000
120000
100000
2
4
6
8
10
Number of Clusters

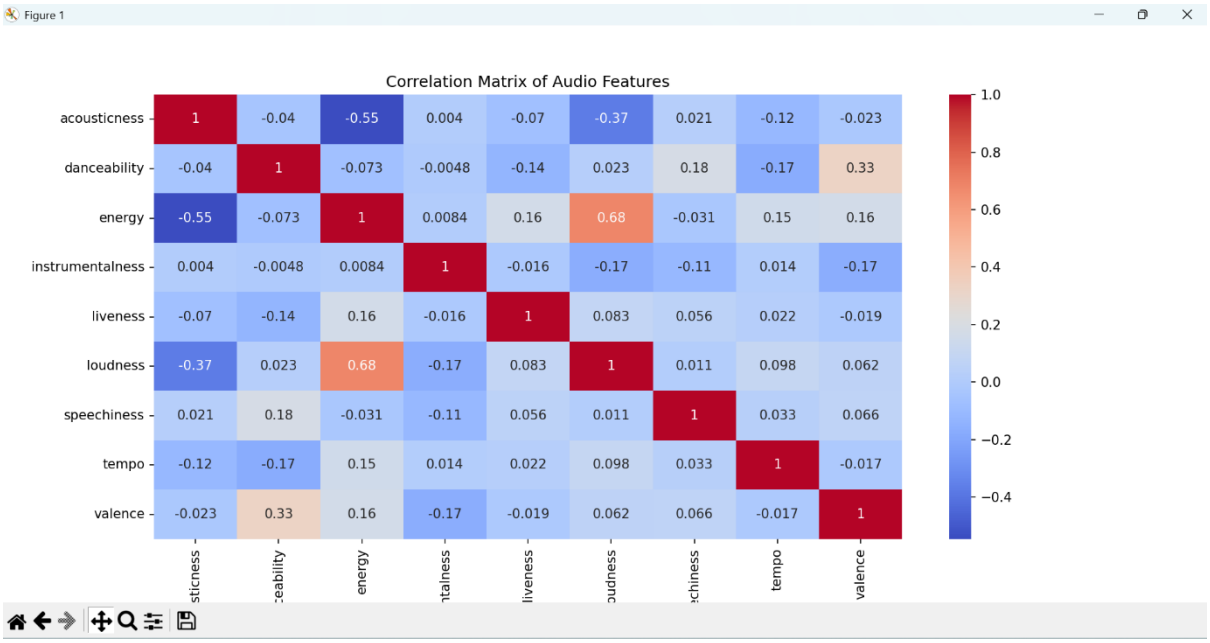


Figure 1
Correlation Matrix of Audio Features
acousticness 1 -0.04 -0.55 0.004 -0.07 -0.37 0.021 -0.12 -0.023
danceability -0.04 1 -0.073 -0.0048 -0.14 0.023 0.18 -0.17 0.33
energy -0.55 -0.073 1 0.0084 0.16 0.68 -0.031 0.15 0.16
instrumentalness 0.004 -0.0048 0.0084 1 -0.016 -0.17 -0.11 0.014 -0.17
liveness -0.07 -0.14 0.16 -0.016 1 0.083 0.056 0.022 -0.019
loudness -0.37 0.023 0.68 -0.17 0.083 1 0.011 0.098 0.062
speechiness 0.021 0.18 -0.031 -0.11 0.056 0.011 1 0.033 0.066
tempo -0.12 -0.17 0.15 0.014 0.022 0.098 0.033 1 -0.017
valence -0.023 0.33 0.16 -0.17 -0.019 0.062 0.066 -0.017 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
tempo
valence

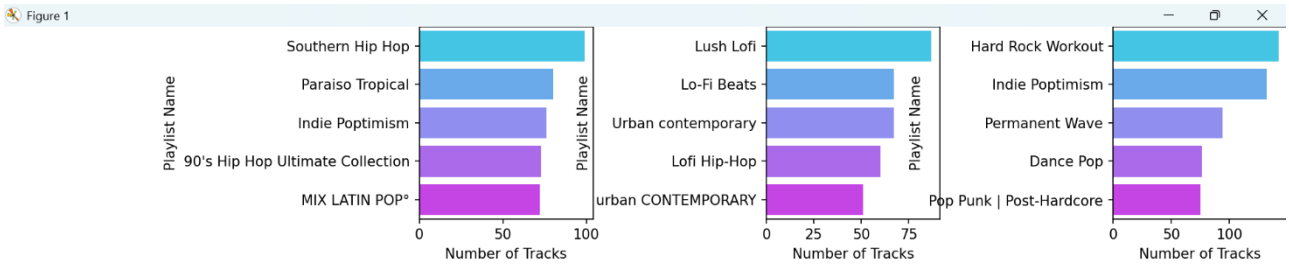
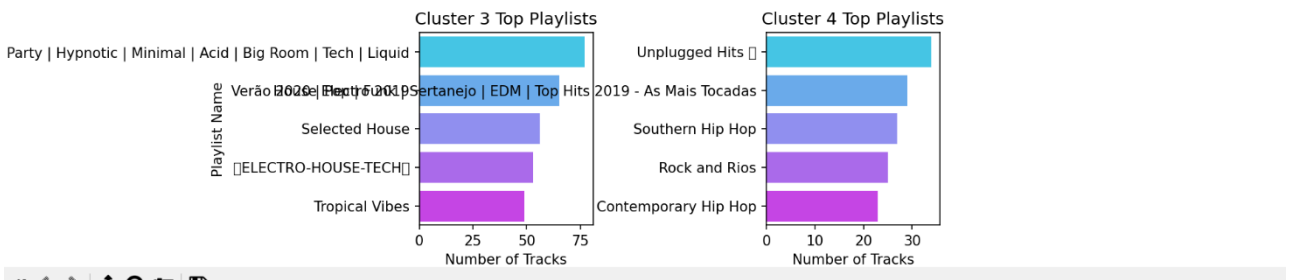
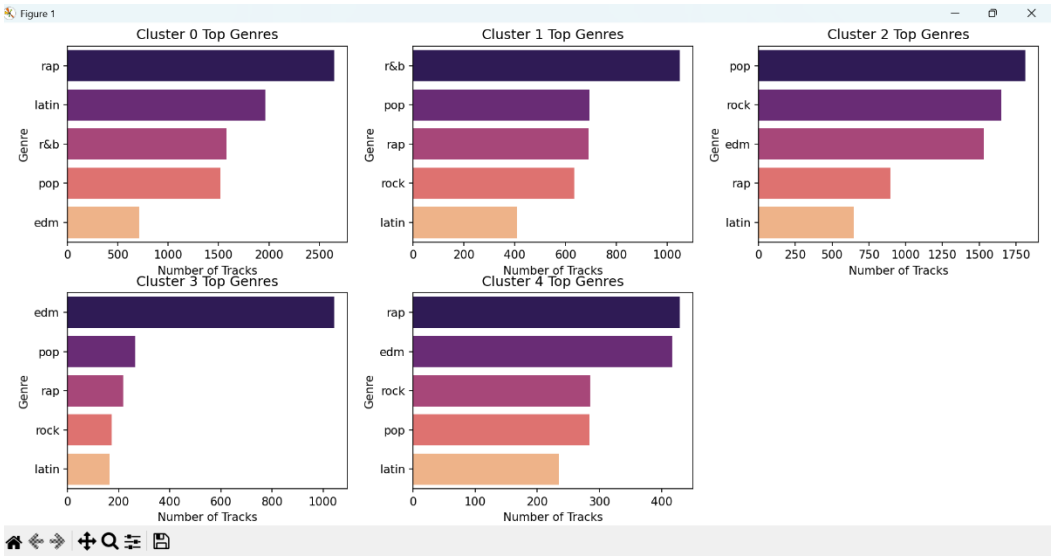
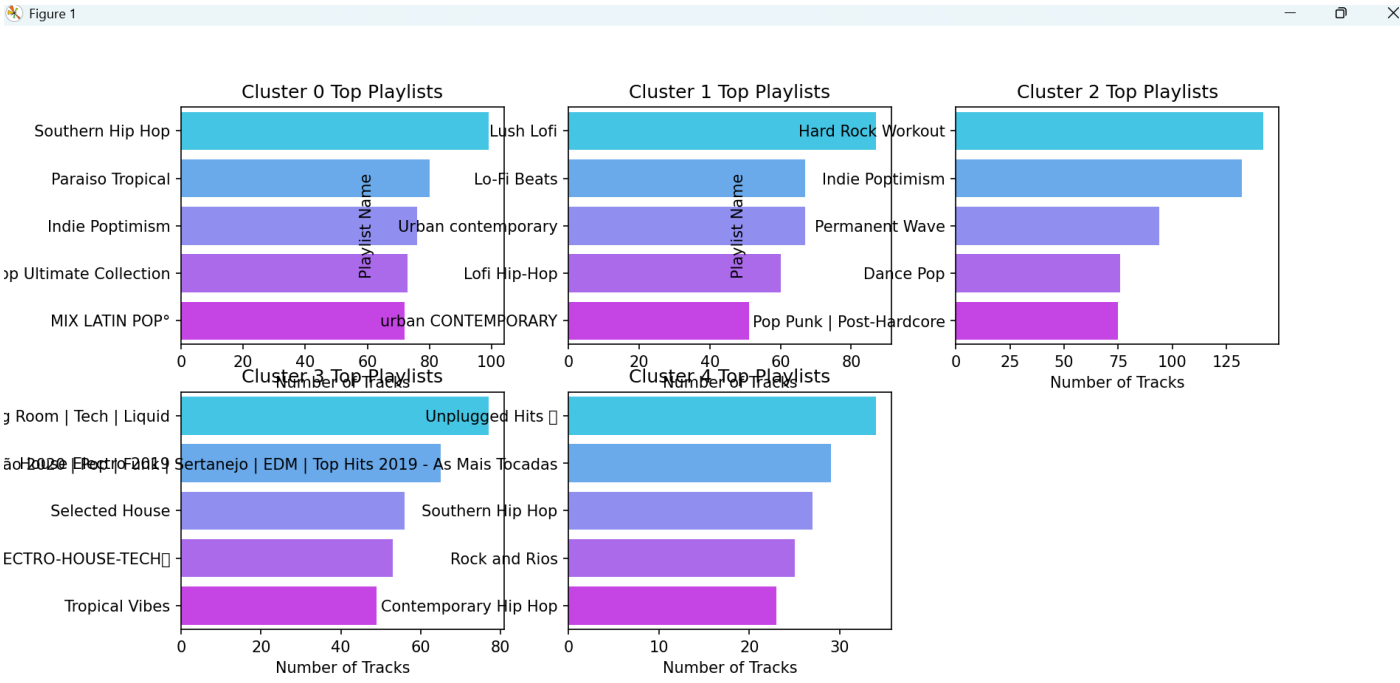


Figure 1
Playlist Name
Southern Hip Hop
Paraiso Tropical
Indie Poptimism
90's Hip Hop Ultimate Collection
MIX LATIN POP°
0
50
100
Number of Tracks
Lush Lofi
Lo-Fi Beats
Urban contemporary
Lofi Hip-Hop
urban CONTEMPORARY
0
25
50
75
Number of Tracks
Hard Rock Workout
Indie Poptimism
Permanent Wave
Dance Pop
Pop Punk | Post-Hardcore
0
50
100
Number of Tracks


Cluster 3 Top Playlists
Party | Hypnotic | Minimal | Acid | Big Room | Tech | Liquid
Selected House
ELECTRO-HOUSE-TECH
Tropical Vibes
0
25
50
75
Number of Tracks
Cluster 4 Top Playlists
Unplugged Hits
Sertanejo | EDM | Top Hits 2019 - As Mais Tocadas
Southern Hip Hop
Rock and Rios
Contemporary Hip Hop
0
10
20
30
Number of Tracks






```
PS C:\Anishka\AI coding> & C:/Users/Anish/AppData/Local/Programs/Python/Python312/python.exe c:/Anishka/AI coding/spotify mini.py"
Spotify dataset loaded successfully!

                track_id                                         track_name  ...    tempo  duration_ms
0  6f807x0ima9a1j3VPbc7VN  I Don't Care (with Justin Bieber) - Loud Luxur...  ...  122.036       194754
1  0r7CVbZTWZgbTCYdfa2P31                    Memories - Dillon Francis Remix  ...   99.972       162600
2  1z1Hg7Vb0AhHDiEmnDE79l                    All the Time - Don Diablo Remix  ...  124.008       176616
3  75FpbthrwQmzHlBJLuGdC7                  Call You Mine - Keanu Silva Remix  ...  121.956       169093
4  1e8PAfcKUYoKkxPhrHqw4x           Someone You Loved - Future Humans Remix  ...  123.976       189052

[5 rows x 23 columns]

Dataset 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
None

Statistical Summary:
       track_popularity  danceability        energy  ...       valence         tempo    duration_ms
count      32833.000000  32833.000000  32833.000000  ...  32833.000000  32833.000000   32833.000000
mean          42.477081      0.654850      0.698619  ...      0.510561    120.881132  225799.811622
std           24.984074      0.145085      0.180910  ...      0.233146     26.903624   59834.006182
min            0.000000      0.000000      0.000175  ...      0.000000      0.000000    4000.000000
25%           24.000000      0.563000      0.581000  ...      0.331000     99.960000  187819.000000
50%           45.000000      0.672000      0.721000  ...      0.512000    121.984000  216000.000000
75%           62.000000      0.761000      0.840000  ...      0.693000    133.918000  253585.000000
max          100.000000      0.983000      1.000000  ...      0.991000    239.440000  517810.000000

[8 rows x 13 columns]

Unique songs retained: 23449

25%           24.000000      0.563000      0.581000  ...      0.331000     99.960000  187819.000000
50%           45.000000      0.672000      0.721000  ...      0.512000    121.984000  216000.000000
75%           62.000000      0.761000      0.840000  ...      0.693000    133.918000  253585.000000
max          100.000000      0.983000      1.000000  ...      0.991000    239.440000  517810.000000
```

```
[8 rows x 13 columns]

Unique songs retained: 23449

25%           24.000000      0.563000      0.581000  ...      0.331000     99.960000  187819.000000
50%           45.000000      0.672000      0.721000  ...      0.512000    121.984000  216000.000000
75%           62.000000      0.761000      0.840000  ...      0.693000    133.918000  253585.000000
max          100.000000      0.983000      1.000000  ...      0.991000    239.440000  517810.000000

[8 rows x 13 columns]

Unique songs retained: 23449


[8 rows x 13 columns]

Unique songs retained: 23449


Feature scaling completed!
c:\Anishka\AI corizo\spotify mini.py:86: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable t
o `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_genres.values, y=top_genres.index, palette='viridis')

K-Means clustering applied with 5 clusters.

Average Audio Features per Cluster:
        acousticness  danceability    energy  instrumentalness  ...   loudness  speechiness       tempo   valenc
e
```

```
Average Audio Features per Cluster:
         acousticness  danceability    energy  instrumentalness  ...   loudness  speechiness       tempo   valen
e
cluster                                                          ...

0            0.156606      0.755865  0.700188          0.013590  ...  -6.572568     0.140899  113.560821  0.6573
1
1            0.533246      0.604401  0.421224          0.106430  ... -10.675379     0.088970  112.810247  0.3946
5
2            0.075696      0.559716  0.790547          0.021561  ...  -5.361819     0.083034  133.610926  0.4148
5
3            0.075044      0.665926  0.773070          0.753286  ...  -7.162814     0.070654  124.739497  0.3941
2
4            0.128950      0.613648  0.777167          0.052724  ...  -6.092147     0.147311  122.456936  0.5128
4

[5 rows x 9 columns]

Average Song Duration (minutes) per Cluster:
cluster
0    3.700296
1    3.707747
2    3.764871
3    4.160776
4    3.824458
Name: duration_min, dtype: float64

Top Artists per Cluster:
cluster
0        Logic                      36
         Don Omar                   35
         Daddy Yankee               30
1        Queen                      42
         Billie Eilish              20
         Daniel Caesar              18
2        Queen                      43
         David Guetta               37
         Martin Garrix              36
3        Dimitri Vegas & Like Mike  23
         Martin Garrix              17
         Semser                     14
4        Miguel Rios                25
         Soda Stereo                22
         Queen                      11
Name: track_artist, dtype: int64

Recommendations for 'bad guy':
          track_name          track_artist playlist_genre
2851         Red Wine        Quinn O'Donnell           r&b
2856         Twenty20             Toby Beck           r&b
745       Smile Again  Green Assassin Dollar           rap
2151  Peaceful Forest   The Sleep Specialist         latin
3013       Send It On              D'Angelo           r&b

Recommendations for 'Bohemian Rhapsody - Remastered 2011':
                                        track_name   track_artist playlist_genre
2915                       Come & Talk To Me         Jodeci           r&b
1273                                  LeBron           Duki           rap
3372                                   Truth    Chris Lewis           r&b
101   Playing Games (with Bryson Tiller) - Extended ...  Summer Walker           pop
3435                Tell Her You Belong To Me      Beth Hart           r&b

Random Recommendations Attempt 1:
                           track_name   track_artist playlist_genre
6149                Are You Listening      DubVision           edm
```

```
6149                 Are You Listening      DubVision           edm
6234            Polaroid - R3HAB Remix     Jonas Blue           edm
6142            Internet Friends - VIP    Knife Party           edm
6021  Beautiful Now - Dirty South Remix           Zedd           edm
5694           Sun Is Never Going Down  Martin Garrix           edm

Random Recommendations Attempt 2:
                     track_name    track_artist playlist_genre
962                     Outside   Calvin Harris            pop
5821  Don’t Stop (feat. RayRay)          Curbi            edm
2523           Mis Días Sin Ti  Rauw Alejandro            rap
1288         Heat Of The Moment            Asia            pop
5895              United We Are        Hardwell            edm

Random Recommendations Attempt 3:
                                        track_name  track_artist playlist_genre
4964              Turn Me On (Hold You) - Radio Edit   Kriss Raize          latin
1859                                      F.B.G.M.        T-Pain            rap
3923                                         Colors     Crossfade           rock
1578  ME! (feat. Brendon Urie of Panic! At The Disco)  Taylor Swift            pop
332             Que Calor (feat. J Balvin & El Alfa)   Major Lazer            pop

Project Execution Completed Successfully!
```