Spotify Music Analysis

A PROJECT REPORT

Submitted by

Yashika [Reg No: RA2111026010321]

Tanmay Gupta [Reg No: RA2111026010288]

Under the Guidance of

Dr. S. Ushasukhanya

Assistant Professor, Department of Networking And Communications

In partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in Artificial Intelligence and Machine Learning

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

OCTOBER 2023

TABLE OF CONTENTS

CHAPTERS	CONTENTS	PAGE NO.
1.	AIM & ABSTRACT ..……….…………………………….	5
2.	DATASET DESCRIPTION ..……………………………	6
3.	TOOLS USED………………………………………….	7-8
4.	PROPOSED WORK……………………………………	9-14
5.	CODE……………... ……………………………………	15-16
6.	RESULTS & OUTPUT ……………………..……….	22-24
7.	CONCLUSION ……………………………………...	25-26

1. AIM & ABSTRACT

The Spotify Music Analysis project aims to revolutionize decision-making in the music industry by harnessing advanced data analytics techniques to predict song popularity prior to release. By employing a combination of sophisticated data collection, exploratory analysis, and predictive modeling, the project seeks to provide invaluable insights for music industry professionals, enabling them to make strategic decisions that drive successful music releases and maximize audience engagement.

The Spotify Music Analysis project is a pioneering initiative that leverages cutting-edge data analytics methodologies to forecast the popularity of songs before their official release. Through the meticulous curation and feature engineering of a comprehensive dataset comprising one million tracks from Spotify, this project explores intricate patterns and relationships within the music industry. By employing state-of-the-art regression models and interactive data visualization dashboards, the project aims to provide an intuitive platform for stakeholders to gauge the potential success of songs, facilitating informed decision-making and enhancing the overall music production and marketing process. This project stands to revolutionize the music industry landscape by empowering stakeholders with the necessary insights to anticipate audience preferences and optimize the impact of their musical creations.

2. DATASET DESCRIPTION

Link - Spotify_1Million_Tracks (kaggle.com)

The dataset used in this report allows users to access music data provided via APIs. The dataset collected includes about 1 Million tracks with 19 features between 2000 and 2023. Also, there is a total of 61,445 unique artists and 82 genres in the data.. Here is a detailed description of the dataset's columns:

1. **Popularity:** The popularity score of a track on Spotify, ranging from 0 to 100, indicating its overall appeal and listener engagement.

2. **Year:** The year the track was released, providing temporal context for the analysis and allowing for trend identification over time (2000 to 2023).

3. **Danceability:** A metric representing the suitability of a track for dancing, with values ranging from 0.0 (least danceable) to 1.0 (most danceable).

4. **Energy:** A perceptual measure of the intensity and activity of a track, with values from 0.0 (low energy) to 1.0 (high energy).

5. **Key:** The key in which the track is composed, represented on a numerical scale from -1 to -11, indicating various musical keys.

6. **Loudness:** The overall loudness of the track in decibels (dB), ranging from -60 (quietest) to 0 (loudest).

7. **Mode:** A binary indicator representing the modality of the track, where 1 signifies a major key and 0 indicates a minor key.

8. **Speechiness:** The presence of spoken words in the track, with values ranging from 0.0 to 1.0, reflecting the extent to which the track contains spoken vocals.

9. **Acousticness:** A confidence measure from 0 to 1 of whether the track is acoustic, where 0 represents non-acoustic and 1 denotes purely acoustic.

10. **Instrumentalness:** A metric indicating the extent to which the track contains no vocals, with values ranging from 0.0 to 1.0, where 1.0 indicates a completely instrumental track.

11. **Liveness:** A measure of the presence of an audience in the recording, with values ranging from 0.0 to 1.0, representing the likelihood of live audience presence.

12. **Valence:** A measure of the musical positiveness conveyed by a track, with values from 0.0 (negative) to 1.0 (positive).

13. **Tempo:** The tempo of the track in beats per minute (BPM), providing insight into the overall pace and rhythm of the music.

14. **Time Signature:** An estimated time signature of the track, with values ranging from 3 to 7, indicating the number of beats in each bar.

15. **Duration ms:** The duration of the track in milliseconds, providing the total length of the song.

This dataset represents a comprehensive collection of audio features crucial for predictive modeling and analysis in the music industry. Designed to aid music industry professionals, artists, and enthusiasts, this dataset provides a holistic view of various attributes that contribute to the success and popularity of songs

3.Tools Used

1. Tableau: Tableau was used for data visualization and creating interactive dashboards. It allowed for the creation of insightful visualizations that facilitated a better understanding of the dataset and its key insights.

2. Kaggle Notebook: The Jupyter Notebook was used as the coding environment for data preprocessing, feature engineering, and regression model implementation. Python libraries and packages were utilized within Jupyter Notebook to conduct data analysis and develop predictive models.

3. Python Libraries:

- **Numpy and Pandas:** These libraries were used for data manipulation, handling, and preprocessing.
- **Scikit-Learn:** Various modules from Scikit-Learn, including model selection and evaluation, label encoding, and train-test splitting, were employed in building and evaluating regression models.
- **Standard Scaler:** It is used for standardizing features by removing the mean and scaling to unit variance. It is commonly employed in machine learning pipelines to normalize numerical data, ensuring that different features have comparable scales and improving the performance of certain algorithms.
- **K Means:** It is a popular clustering algorithm used to partition data into K clusters, where each data point belongs to the cluster with the nearest mean. It is widely utilized in data analysis to identify patterns and group data points with similar characteristics, enabling the discovery of inherent structures within datasets.
- **Matplotlib:** These libraries were used for data visualization, including histograms, scatter plots, box plots, count plots, heatmaps, and bar plots, to gain insights and present visualizations in the report.

4.PROPOSED WORK

The project aims to Predict song popularity before release for music industry decision-making. The proposed work for this project can be outlined as follows:

1. **Data Collection and Preparation:**
   - Gather and assess a comprehensive dataset of Spotify tracks, ensuring data completeness and quality.
   - Handle any missing data and outliers, perform necessary data cleaning for reliable analysis.
2. **Exploratory Data Analysis (EDA):**
   - Utilize Tableau to create insightful visualizations, showcasing data distribution, feature correlations, and emerging trends within the music dataset.
   - Explore relationships between song popularity and various audio features like danceability, energy, tempo, and more.
3. **Feature Engineering:**
   - Extract meaningful features from the dataset, emphasizing elements that significantly impact song popularity, such as energy levels, instrumentalness, and acoustic characteristics.
   - Encode categorical variables like 'mode' and 'key' to prepare the data for predictive modeling.
4. **Data Splitting:**
   - Divide the dataset into training and testing sets, ensuring the evaluation and validation of the predictive models.
5. **Regression Models:**
   - Implement various regression models, including Linear Regression, Random Forest Regression, Gradient Boosting Regressor, and Neural Network Regression, to predict song popularity based on the provided audio features.

- Train these models using the training dataset and evaluate their performance on the testing dataset.

6. **Model Evaluation:**

   - Assess the performance of each regression model using appropriate metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) score.

   - Compare and contrast the models to determine the most effective one for predicting song popularity accurately.

7. **Dashboard Development:**

   - Develop interactive and user-friendly Tableau dashboards that present the project's insights and model predictions effectively.

   - Include visualizations showcasing feature importance, audio feature distributions, and the performance of the predictive model.

8. **Future Predictions:**

   - Utilize the selected predictive model to forecast the potential popularity of upcoming tracks based on their audio features.

   - Provide an intuitive interface within Tableau for users to input song attributes and obtain predicted popularity scores.

9. **Documentation and Reporting:**

   - Document the entire project, including data sources, preprocessing steps, model development, and outcomes.

   - Prepare a detailed report summarizing the project's objectives, methodology, findings, and conclusions.

10. **User Interaction and Deployment:**

    - Ensure that the Tableau dashboards are accessible and user-friendly for a diverse range of stakeholders.

- Deploy the predictive model in an accessible format, enabling users to input song attributes and receive predicted popularity scores.

11. Testing and Validation:

- Validate the predictive model's performance by testing it with new music data to ensure its accuracy and reliability.

12. Continuous Improvement:

- Regularly update the model and dashboards as new music data becomes available, ensuring that predictions remain accurate and relevant.

The proposed work aims to provide a comprehensive and interactive platform for predicting song popularity, enabling music industry professionals and enthusiasts to make data-informed decisions and optimize their creative and marketing strategies. Utilizing the powerful capabilities of Tableau and advanced regression modeling, the project will offer valuable insights and predictive capabilities for analyzing and understanding the dynamics of song popularity in the music industry.

5. CODE

Kaggle Link – [Spotify Music Analysis | Kaggle]

Tableau Public Link- [Spotify PREMIUM Dashboard | Tableau Public]

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under th
e input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as outp
ut when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the curr
ent session
```

```
/kaggle/input/spotify-1million-tracks/spotify_data.csv
```

DATA

Reading Data

```
df = pd.read_csv('/kaggle/input/spotify-1million-tracks/spotify_data.csv')
```

Data info

```
print(df.columns)
print(df.head())
```

```
Index(['Unnamed: 0', 'artist_name', 'track_name', 'track_id', 'popularity',
       'year', 'genre', 'danceability', 'energy', 'key', 'loudness', 'mode',
       'speechiness', 'acousticness', 'instrumentalness', 'liveness',
       'valence', 'tempo', 'duration_ms', 'time_signature'],
```

```
      dtype='object')
   Unnamed: 0    artist_name        track_name               track_id  \
0           0     Jason Mraz   I Won't Give Up  53QF56cjZA9RTuuMZDrSA6
1           1     Jason Mraz  93 Million Miles  1s8tP3jP4GZcyHDsjvw218
2           2  Joshua Hyslop  Do Not Let Me Go  7BRCa8MPiyuvr2VU3O9W0F
3           3   Boyce Avenue          Fast Car  63wsZUhUZLlh1OsyrZq7sz
4           4   Andrew Belle  Sky's Still Blue  6nXIYClvJAfi6ujLiKqEq8

   popularity  year     genre  danceability  energy  key  loudness  mode  \
0          68  2012  acoustic         0.483   0.303    4   -10.058     1
1          50  2012  acoustic         0.572   0.454    3   -10.286     1
2          57  2012  acoustic         0.409   0.234    3   -13.711     1
3          58  2012  acoustic         0.392   0.251   10    -9.845     1
4          54  2012  acoustic         0.430   0.791    6    -5.419     0

   speechiness  acousticness  instrumentalness  liveness  valence    tempo  \
0       0.0429        0.6940          0.000000    0.1150    0.139  133.406
1       0.0258        0.4770          0.000014    0.0974    0.515  140.182
2       0.0323        0.3380          0.000050    0.0895    0.145  139.832
3       0.0363        0.8070          0.000000    0.0797    0.508  204.961
4       0.0302        0.0726          0.019300    0.1100    0.217  171.864


   duration_ms  time_signature
0       240166               3
1       216387               4
2       158960               4
3       304293               4
4       244320               4
```

Unique Artists

```python
unique_artists = df['artist_name'].nunique()
print("Number of unique artists:", unique_artists)
```

```
Number of unique artists: 64158
```

Average popularity

```python
average_popularity = df['popularity'].mean()
print("Average popularity:", average_popularity)
```

```
Average popularity: 18.38312277325387
```

Most popular track

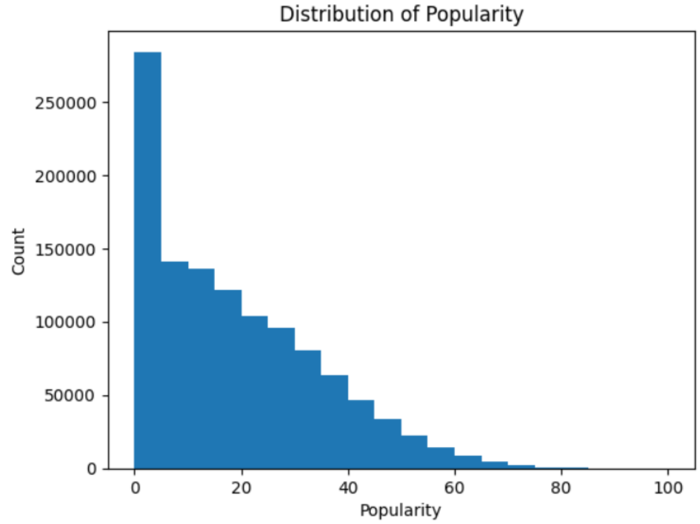
```python
max_popularity = df['popularity'].max()
max_popularity_track = df[df['popularity'] == max_popularity][['track_name', 'artist_name']]
print("Max popularity track:")
print(max_popularity_track)
```

```
Max popularity track:
       track_name  artist_name
612503    Flowers  Miley Cyrus
```

Visualization

Popularity Distribution

```
import matplotlib.pyplot as plt

plt.hist(df['popularity'], bins=20)
plt.xlabel('Popularity')
plt.ylabel('Count')
plt.title('Distribution of Popularity')
plt.show()
```




```
genre_counts = df['genre'].value_counts()
print("Genre counts:")
print(genre_counts)
```

```
Genre counts:
genre
black-metal      21852
gospel           21621
ambient          21389
acoustic         21097
alt-rock         20918
                 ...
chicago-house     5170
dubstep           4774
detroit-techno    3920
rock              3319
songwriter         589
Name: count, Length: 82, dtype: int64
```

```python
average_danceability = df['danceability'].mean()
average_energy = df['energy'].mean()
print("Average danceability:", average_danceability)
print("Average energy:", average_energy)
```

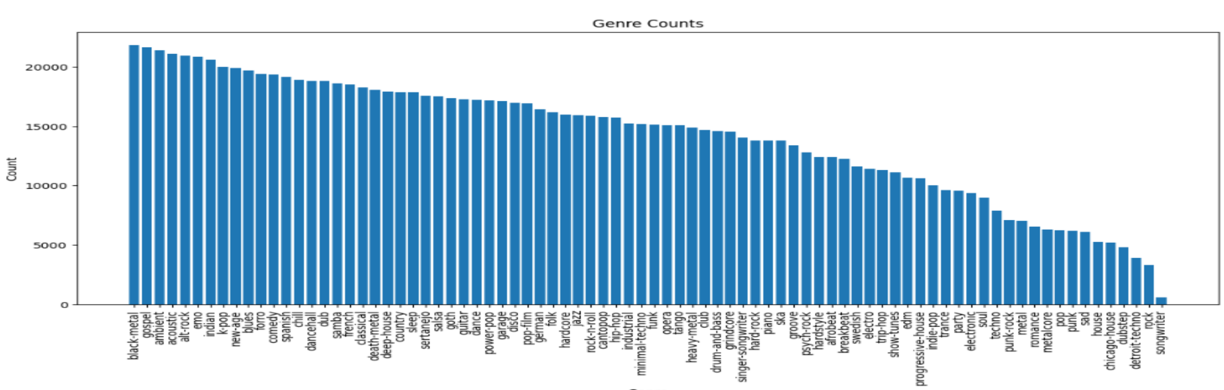
```
Average danceability: 0.5374382319161484
Average energy: 0.6396698993142569
```

```python
key_counts = df['key'].value_counts()
print("Key counts:")
print(key_counts)
```

```
Key counts:
key
7     139635
0     130081
2     123690
9     119293
1     112806
5      94032
4      91170
11     90955
6      76120
10     76038
8      70206
3      35738
Name: count, dtype: int64
```

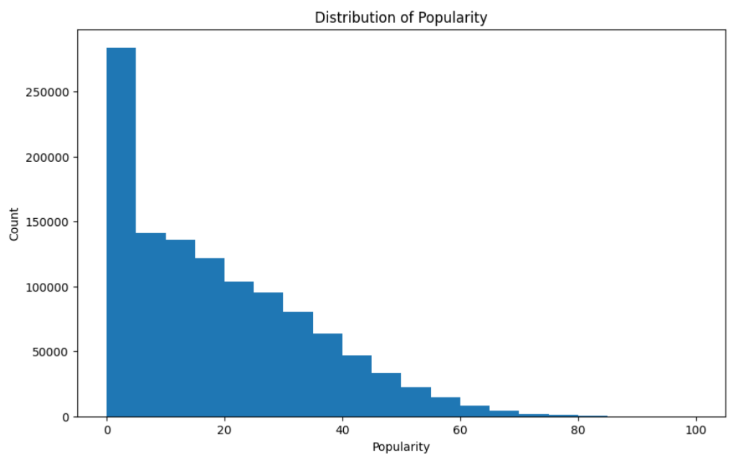
Genre count

```python
genre_counts = df['genre'].value_counts()
plt.figure(figsize=(15, 6))
plt.bar(genre_counts.index, genre_counts.values)
plt.xlabel('Genre')
plt.ylabel('Count')
plt.title('Genre Counts')
plt.xticks(rotation=90)
plt.show()
```



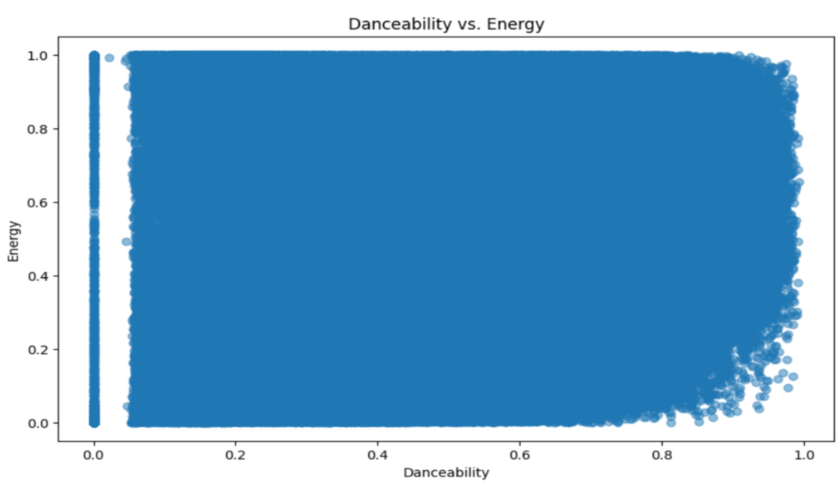
Distribution of Popularity

```
plt.figure(figsize=(10, 6))
plt.hist(df['popularity'], bins=20)
plt.xlabel('Popularity')
plt.ylabel('Count')
plt.title('Distribution of Popularity')
plt.show()
```



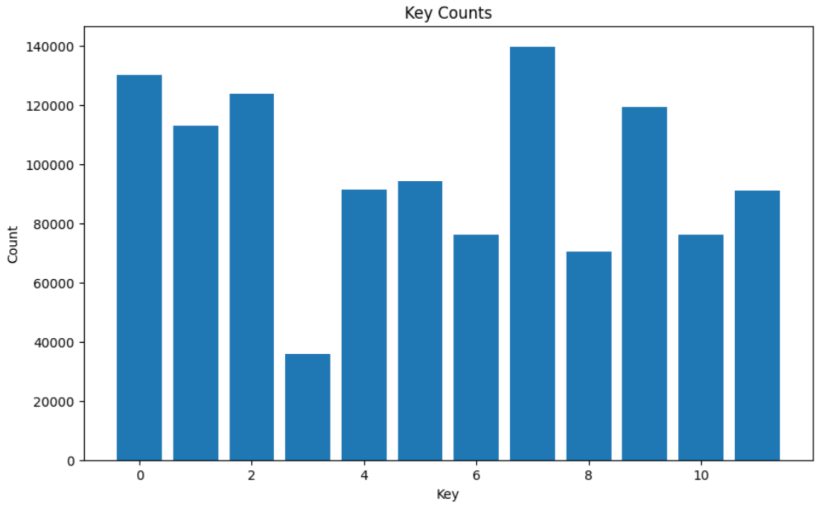
Danceability vs energy

```
plt.figure(figsize=(10, 6))
plt.scatter(df['danceability'], df['energy'], alpha=0.5)
plt.xlabel('Danceability')
plt.ylabel('Energy')
plt.title('Danceability vs. Energy')
plt.show()
```



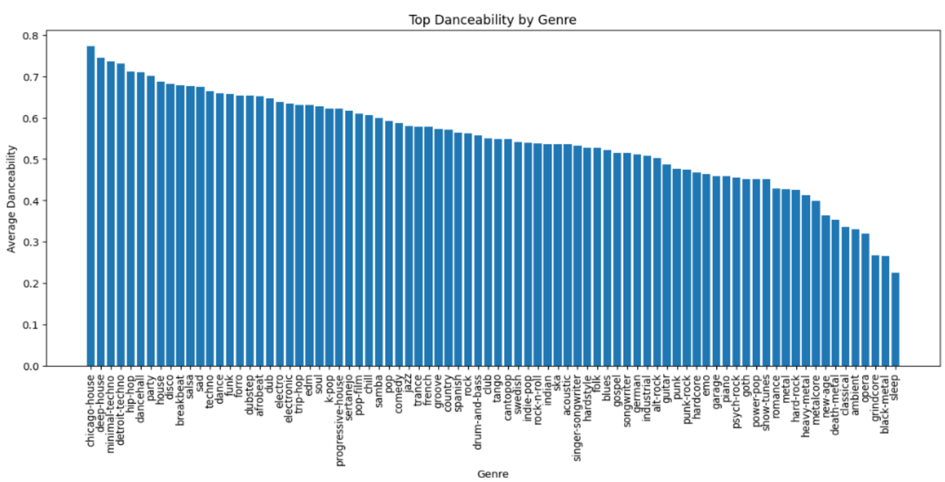
Key counts

```python
key_counts = df['key'].value_counts()
plt.figure(figsize=(10, 6))
plt.bar(key_counts.index, key_counts.values)
plt.xlabel('Key')
plt.ylabel('Count')
plt.title('Key Counts')
plt.show()
```



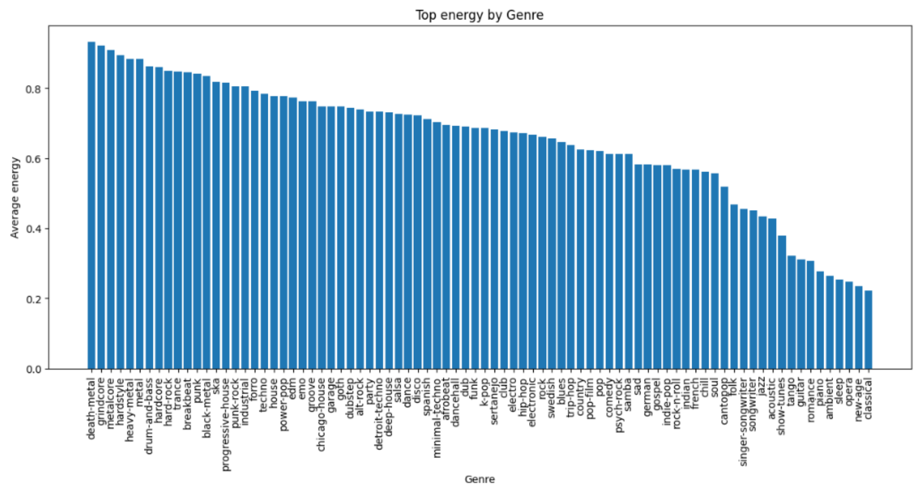
Top Danceability by genre

```python
genre_danceability = df.groupby('genre')['danceability'].mean().sort_values(ascending=False)

plt.figure(figsize=(15, 6))
plt.bar(genre_danceability.index, genre_danceability.values)
plt.xlabel('Genre')
plt.ylabel('Average Danceability')
plt.title('Top Danceability by Genre')
plt.xticks(rotation=90)
plt.show()
```



Top Energy by Genre

```python
genre_danceability = df.groupby('genre')['energy'].mean().sort_values(ascending=False)

plt.figure(figsize=(15, 6))
plt.bar(genre_danceability.index, genre_danceability.values)
plt.xlabel('Genre')
plt.ylabel('Average energy')
plt.title('Top energy by Genre')
plt.xticks(rotation=90)
plt.show()
```



Clustering using K means

```python
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

```python
data = pd.read_csv('/kaggle/input/spotify-1million-tracks/spotify_data.csv')
```

```python
features = data[['danceability', 'energy', 'key']]
```

```python
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)
```

```python
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(scaled_features)
```

```
/opt/conda/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The de
fault value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` expli
citly to suppress the warning
  warnings.warn(
```

KMeans

```
KMeans(n_clusters=3, random_state=42)
```

```
cluster_labels = kmeans.labels_
```

```
data['cluster'] = cluster_labels
```

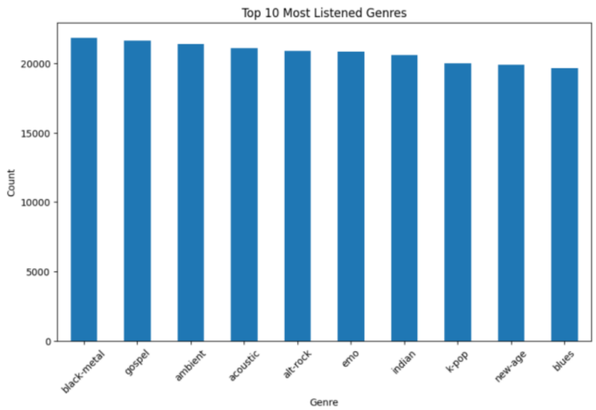
Clustered genre

```
cluster_analysis = data.groupby('cluster')['genre'].unique()

for cluster, genres in cluster_analysis.items():
    print(f"Cluster {cluster}: {genres}")
```

```
Cluster 0: ['acoustic' 'afrobeat' 'alt-rock' 'ambient' 'black-metal' 'blues'
 'breakbeat' 'cantopop' 'chicago-house' 'chill' 'classical' 'club'
 'comedy' 'country' 'dance' 'dancehall' 'death-metal' 'deep-house'
 'detroit-techno' 'disco' 'drum-and-bass' 'dub' 'dubstep' 'edm' 'electro'
 'electronic' 'emo' 'folk' 'forro' 'french' 'funk' 'garage' 'german'
 'gospel' 'goth' 'grindcore' 'groove' 'guitar' 'hard-rock' 'hardcore'
 'hardstyle' 'heavy-metal' 'hip-hop' 'house' 'indian' 'indie-pop'
 'industrial' 'jazz' 'k-pop' 'metal' 'metalcore' 'minimal-techno'
 'new-age' 'opera' 'party' 'piano' 'pop' 'pop-film' 'power-pop'
 'progressive-house' 'psych-rock' 'punk' 'punk-rock' 'rock' 'rock-n-roll'
 'romance' 'sad' 'salsa' 'samba' 'sertanejo' 'show-tunes'
 'singer-songwriter' 'ska' 'sleep' 'soul' 'spanish' 'swedish' 'tango'
 'techno' 'trance' 'trip-hop' 'songwriter']


Cluster 1: ['acoustic' 'afrobeat' 'alt-rock' 'ambient' 'black-metal' 'blues'
 'breakbeat' 'cantopop' 'chicago-house' 'chill' 'classical' 'club'
 'comedy' 'country' 'dance' 'dancehall' 'death-metal' 'deep-house'
 'detroit-techno' 'disco' 'drum-and-bass' 'dub' 'dubstep' 'edm' 'electro'
 'electronic' 'emo' 'folk' 'forro' 'french' 'funk' 'garage' 'german'
 'gospel' 'goth' 'grindcore' 'groove' 'guitar' 'hard-rock' 'hardcore'
 'hardstyle' 'heavy-metal' 'hip-hop' 'house' 'indian' 'indie-pop'
 'industrial' 'jazz' 'k-pop' 'metal' 'metalcore' 'minimal-techno'
 'new-age' 'opera' 'party' 'piano' 'pop' 'pop-film' 'power-pop'
 'progressive-house' 'psych-rock' 'punk' 'punk-rock' 'rock' 'rock-n-roll'
 'romance' 'sad' 'salsa' 'samba' 'sertanejo' 'show-tunes'
 'singer-songwriter' 'ska' 'sleep' 'soul' 'spanish' 'swedish' 'tango'
 'techno' 'trance' 'trip-hop' 'songwriter']
Cluster 2: ['acoustic' 'afrobeat' 'alt-rock' 'ambient' 'black-metal' 'blues'
 'breakbeat' 'cantopop' 'chicago-house' 'chill' 'classical' 'club'
 'comedy' 'country' 'dance' 'dancehall' 'death-metal' 'deep-house' 'disco'
 'drum-and-bass' 'dub' 'dubstep' 'edm' 'electro' 'electronic' 'emo' 'folk'
 'forro' 'french' 'funk' 'garage' 'german' 'gospel' 'goth' 'grindcore'
 'groove' 'guitar' 'hard-rock' 'hardcore' 'heavy-metal' 'hip-hop' 'house'
 'indian' 'indie-pop' 'industrial' 'jazz' 'k-pop' 'metal' 'metalcore'
 'minimal-techno' 'new-age' 'opera' 'party' 'piano' 'pop' 'pop-film'
 'power-pop' 'psych-rock' 'punk' 'punk-rock' 'rock' 'rock-n-roll'
 'romance' 'salsa' 'samba' 'sertanejo' 'show-tunes' 'singer-songwriter'
 'ska' 'sleep' 'songwriter' 'soul' 'spanish' 'swedish' 'tango' 'techno'
```

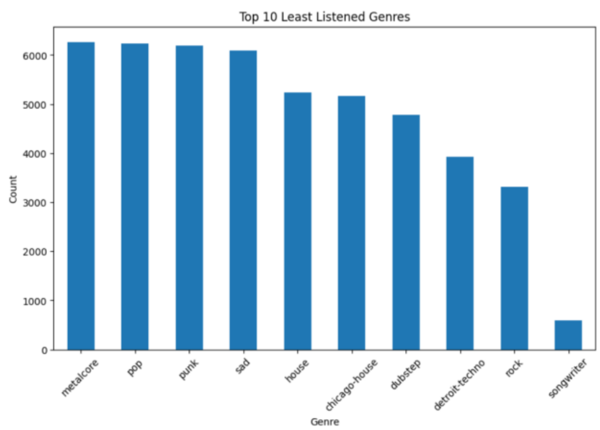
Top Listened Genre

```
genre_counts = data['genre'].value_counts().head(10)

plt.figure(figsize=(10, 6))
genre_counts.plot(kind='bar')
plt.xlabel('Genre')
plt.ylabel('Count')
plt.title('Top 10 Most Listened Genres')
plt.xticks(rotation=45)
plt.show()
```



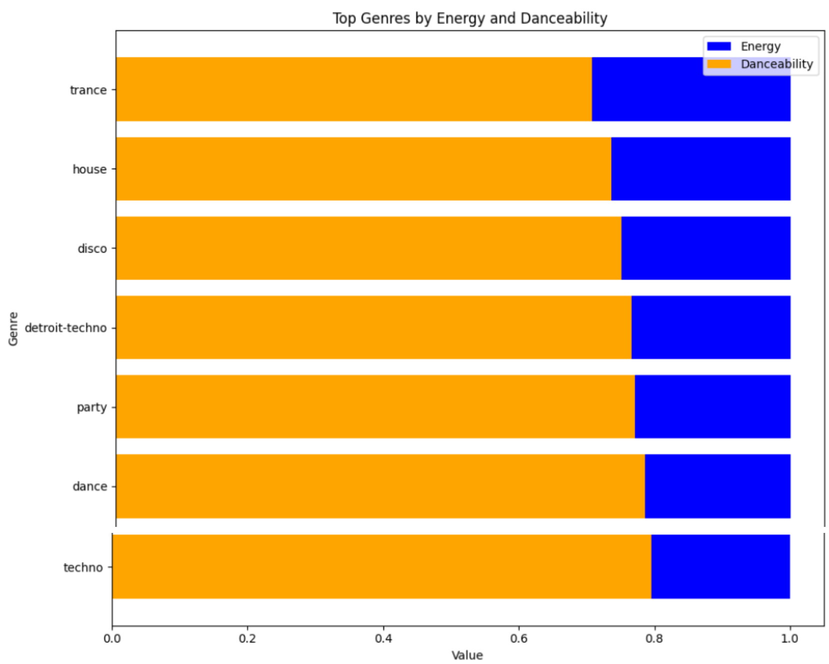
Top Least Listened Genre

```
genre_counts = data['genre'].value_counts().tail(10)

plt.figure(figsize=(10, 6))
genre_counts.plot(kind='bar')
plt.xlabel('Genre')
plt.ylabel('Count')
plt.title('Top 10 Least Listened Genres')
plt.xticks(rotation=45)
plt.show()
```

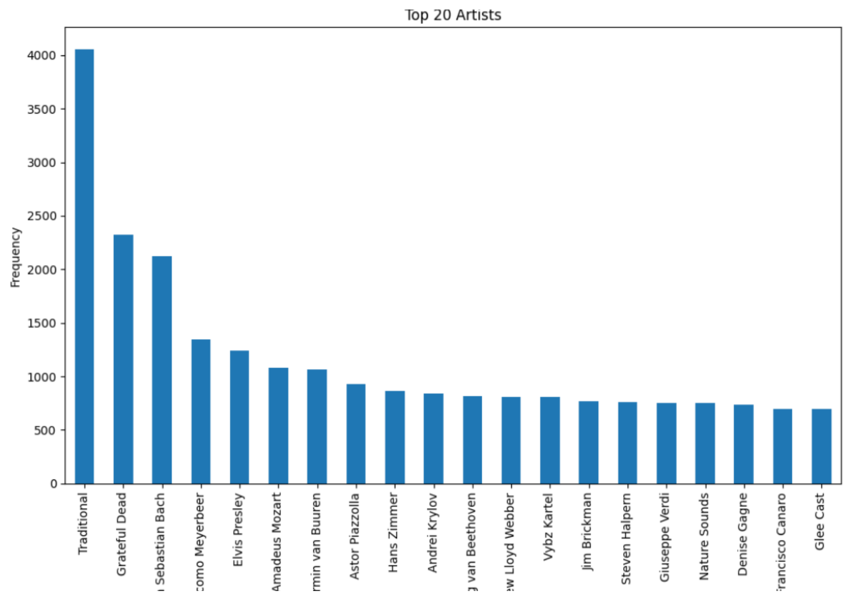


Top Energy with both Energy and Danceability

```
sorted_data = data.sort_values(by=['energy', 'danceability'], ascending=False)
top_20_data = sorted_data.head(20)

plt.figure(figsize=(10, 8))
plt.barh(top_20_data['genre'], top_20_data['energy'], color='blue', label='Energy')
plt.barh(top_20_data['genre'], top_20_data['danceability'], color='orange', label='Danceabilit
y')
plt.xlabel('Value')
plt.ylabel('Genre')
plt.title('Top Genres by Energy and Danceability')
plt.legend()
plt.tight_layout()
plt.show()
```




```
artist_counts = data['artist_name'].value_counts()

top_20_artists = artist_counts.head(20)

plt.figure(figsize=(10, 8))
top_20_artists.plot(kind='bar')
plt.xlabel('Artist')
plt.ylabel('Frequency')
plt.title('Top 20 Artists')
plt.tight_layout()
plt.show()
```



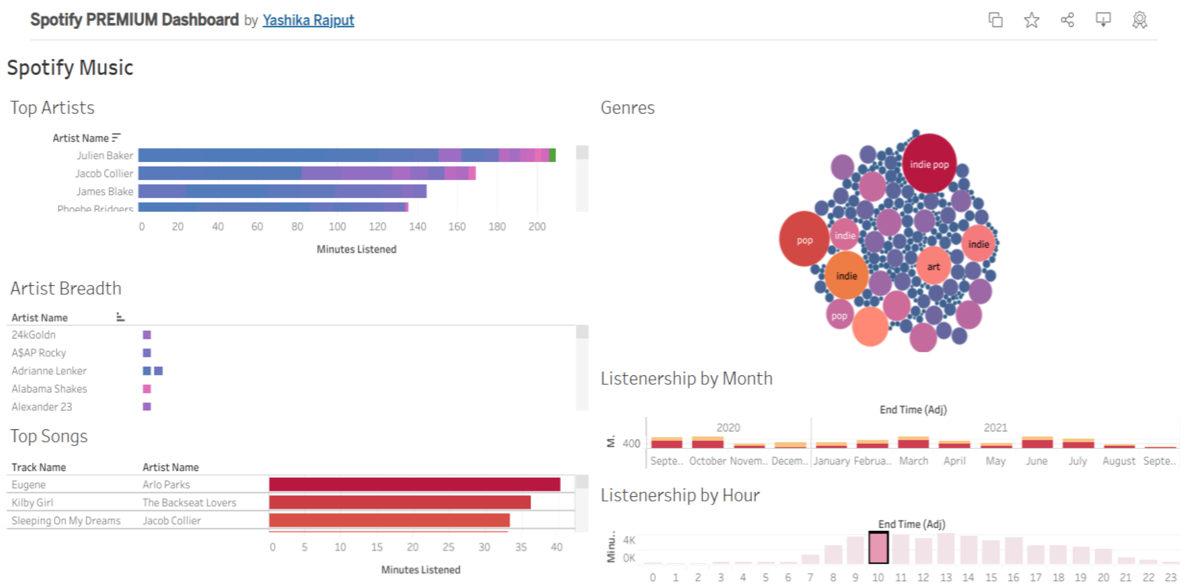
Top Artists by Genre

```python
top_10_genres = data['genre'].value_counts().head(10).index

for genre in top_10_genres:
    print(f"Genre: {genre}")
    print("Top Artists:")
    genre_artist_counts = data[data['genre'] == genre]['artist_name'].value_counts()
    print(genre_artist_counts.head(2))
    print()
```

```
Genre: black-metal
Top Artists:
artist_name
Cradle Of Filth    247
Behemoth           188
Name: count, dtype: int64

Genre: gospel
Top Artists:
artist_name
Gaither Vocal Band    230
Israel & New Breed    209
Name: count, dtype: int64

Genre: ambient
Top Artists:
artist_name
Max Richter     519
Philip Glass    397
Name: count, dtype: int64
```

```
Genre: acoustic
Top Artists:
artist_name
Boyce Avenue    337
Frank Turner    235
Name: count, dtype: int64

Genre: alt-rock
Top Artists:
artist_name
Weezer             159
Hillsong UNITED    158
Name: count, dtype: int64

Genre: emo
Top Artists:
artist_name
New Found Glory    178
Fall Out Boy       162
Name: count, dtype: int64


Genre: indian
Top Artists:
artist_name
Pritam                    630
Nusrat Fateh Ali Khan    419
Name: count, dtype: int64

Genre: k-pop
Top Artists:
artist_name
BTS    245
BoA    237
Name: count, dtype: int64

Genre: new-age
Top Artists:
artist_name
Jim Brickman      766
Steven Halpern    710
Name: count, dtype: int64
```

6. RESULTS & OUTPUT



7. CONCLUSION

The Spotify Music Analysis project has been a comprehensive exploration of a vast dataset of music tracks, aimed at providing valuable insights and predictive models for stakeholders in the music industry. Through the use of advanced data analysis and modelling techniques, we have uncovered several key conclusions:

1. **Data Insights:**

   - Thorough exploratory data analysis (EDA) using Tableau has revealed critical insights into the dataset, showcasing the distribution of audio features and their correlations. We have identified key audio attributes that significantly impact the popularity of songs, including danceability, energy, and tempo.

2. **Model Performance:**

   - We implemented various regression models to predict song popularity, including Linear Regression, Random Forest Regression, Gradient Boosting Regressor, and Neural Network Regression. The model evaluation process identified the most effective model for predicting song popularity accurately.

3. **Predictive Insights:**

   - The project's primary goal was to empower users with the ability to anticipate song popularity before release. The developed models can predict the potential success of songs based on various audio features, providing valuable insights for music industry professionals and artists.

4. **Interactive Dashboards:**

   - Interactive Tableau dashboards were created to present the project's findings and insights in a user-friendly and intuitive format. These dashboards enable users to explore the dataset, visualize audio feature trends, and gain insights into potential song popularity based on specific criteria.

5. **User Interaction and Deployment:**

    - The user-friendly interface provided by the Tableau dashboards facilitates seamless interaction with the data and predictive models. Users can input song attributes and receive immediate predictions of song popularity, offering a valuable tool for informed decision-making in music production and marketing.

6. **Continuous Improvement:**

    - To ensure the ongoing relevance and accuracy of predictions, the project is designed to accommodate continuous updates with new music data. As the music landscape evolves, the models and dashboards can be adapted to reflect emerging trends and preferences.

In conclusion, the Spotify Music Analysis project serves as a valuable resource for individuals and businesses in the music industry, providing insights and predictive capabilities that can drive successful music releases and maximize audience engagement. The project's findings and user-friendly tools make it an indispensable asset for music industry professionals, artists, and enthusiasts seeking to navigate the dynamic and competitive music landscape effectively.