

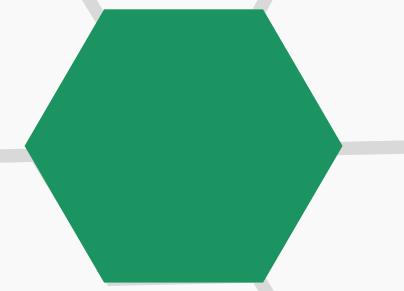
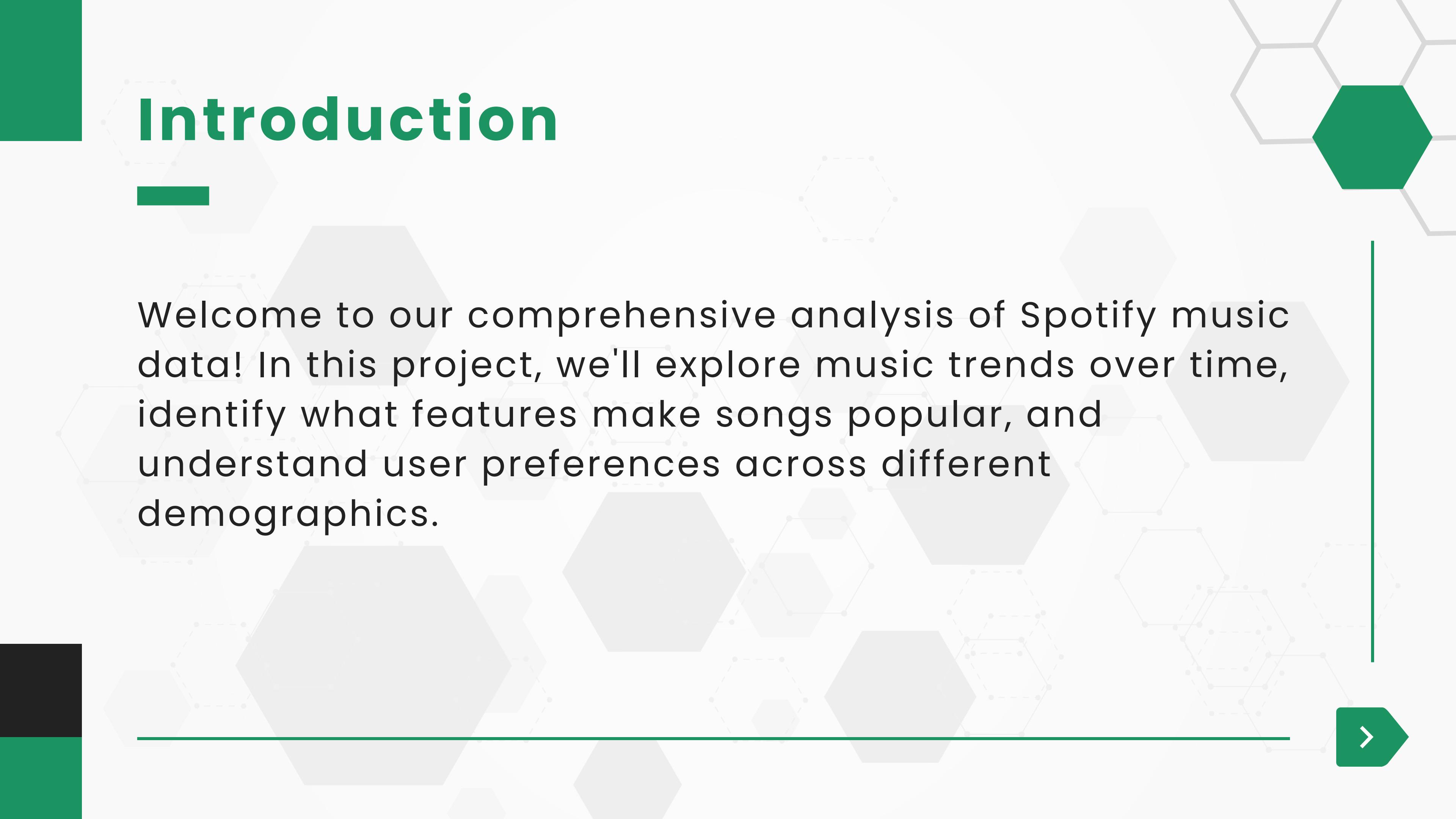


Data Analysis

Spotify Project



Introduction



Welcome to our comprehensive analysis of Spotify music data! In this project, we'll explore music trends over time, identify what features make songs popular, and understand user preferences across different demographics.



Project Overview

This analysis aims to uncover insights into how people interact with music on Spotify. We'll examine:

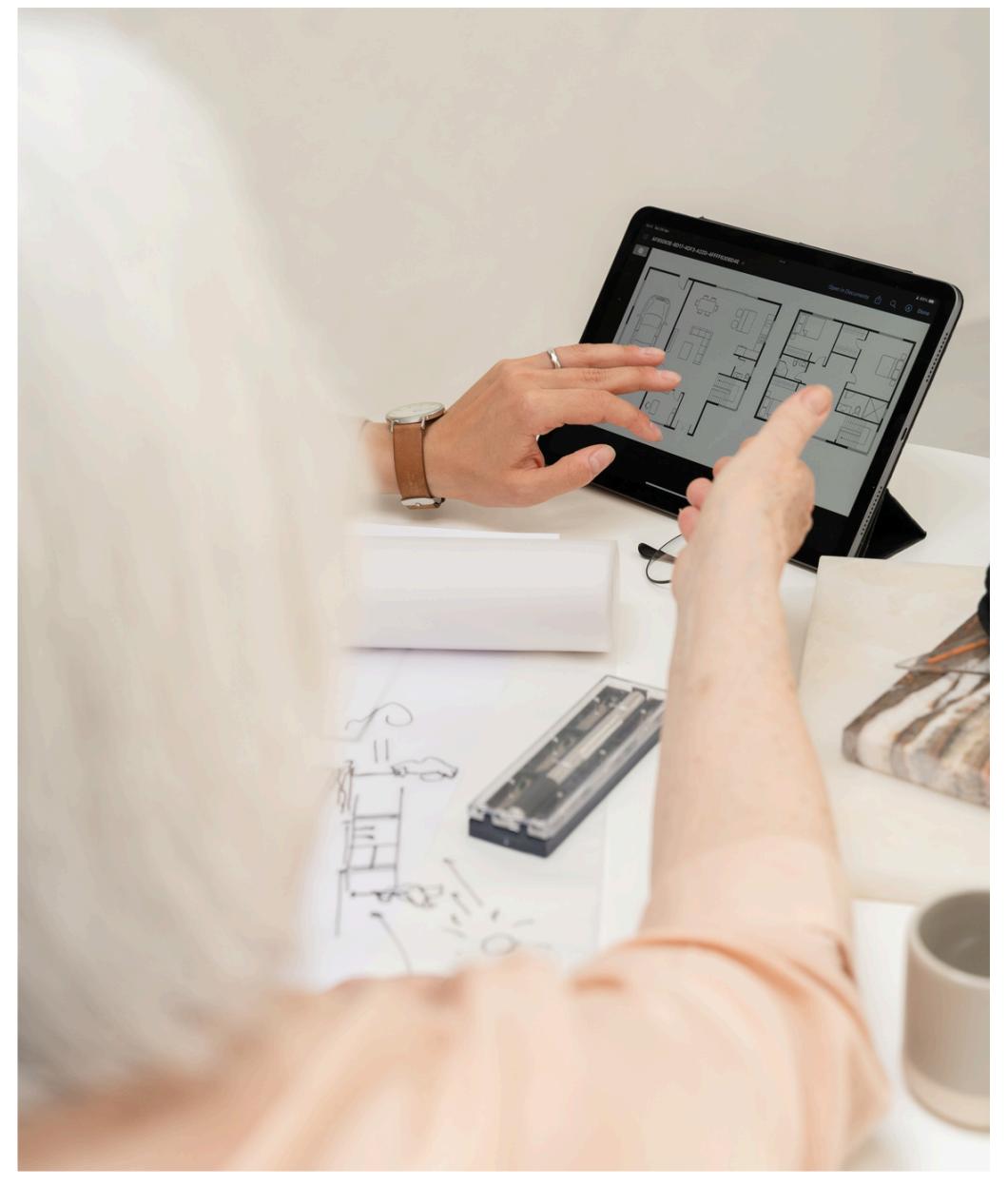
- How music trends have evolved over the years
- What audio features correlate with song popularity
- How user preferences vary by genre, artist, and time period



Datasets Used

We're working with five key datasets:

1. data.csv: Main Spotify tracks dataset with audio features
2. data_by_artist.csv: Metrics aggregated by artist
3. data_by_genres.csv: Metrics aggregated by genres
4. data_by_year.csv: Metrics aggregated by year
5. data_w_genres.csv: Track-level data with genre labels



Data Collection and Preprocessing

```
python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Loading all datasets
df1 = pd.read_csv("data_by_artist.csv")
df2 = pd.read_csv("data_by_year.csv")
df3 = pd.read_csv("data_w_genres.csv")
df4 = pd.read_csv("data.csv")
df5 = pd.read_csv("data_by_genres.csv")
```

Loading the Data
We began by
importing essential
Python libraries and
loading all datasets:



Data Inspection and Cleaning

Key cleaning steps included:

- Checking column names across datasets
- Examining dataset shapes (ranging from 100 to 170,653 rows)
- Handling missing values in critical fields like release_date
- Converting release_date to datetime format
- Removing duplicates
- Filling missing values appropriately

python

```
# Handling missing release dates  
df.release_date = df.release_date.fillna(df.release_date.mode()[0])  
df.release_date = pd.to_datetime(df.release_date, format='ISO8601')  
  
# Checking for duplicates  
df.duplicated().sum() # Found 0 duplicates  
  
# Filling other missing values  
df['count'] = df['count'].fillna(df['count'].median())  
df['artists'] = df['artists'].fillna('Unknown')  
df['year'] = df['year'].fillna(df['year'].median())
```



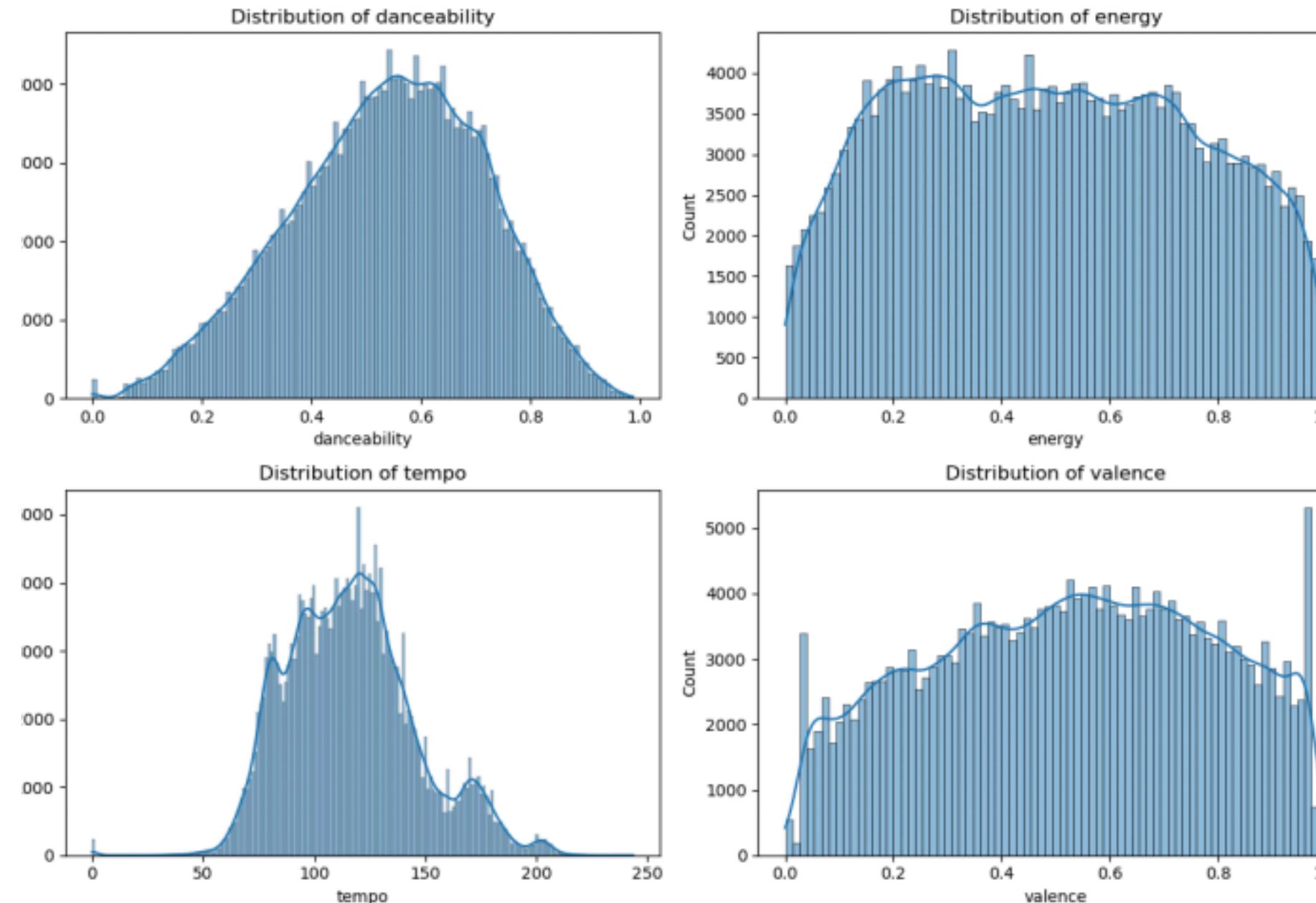
Exploratory Data Analysis (EDA)

We examined descriptive statistics for key audio features:

Feature	Mean	Min	Max
danceability	54	0	988
energy	487	0	10
tempo	11.664	0	243.507
valence	524	0	10
popularity	3.219	0	1.000



Chart



Data Analysis

Feature Distributions

- Key observations:
- Danceability shows a normal distribution centered around 0.55
- Energy is slightly left-skewed with most tracks having moderate energy
- Tempo has a bimodal distribution
- Valence (musical positiveness) is relatively evenly distributed

python

```
features = ['danceability', 'energy', 'tempo', 'valence']
for feature in features:
    plt.figure(figsize=(8,4))
    sns.histplot(df[feature], kde=True)
    plt.title(f'Distribution of {feature.capitalize()}')
    plt.show()
```



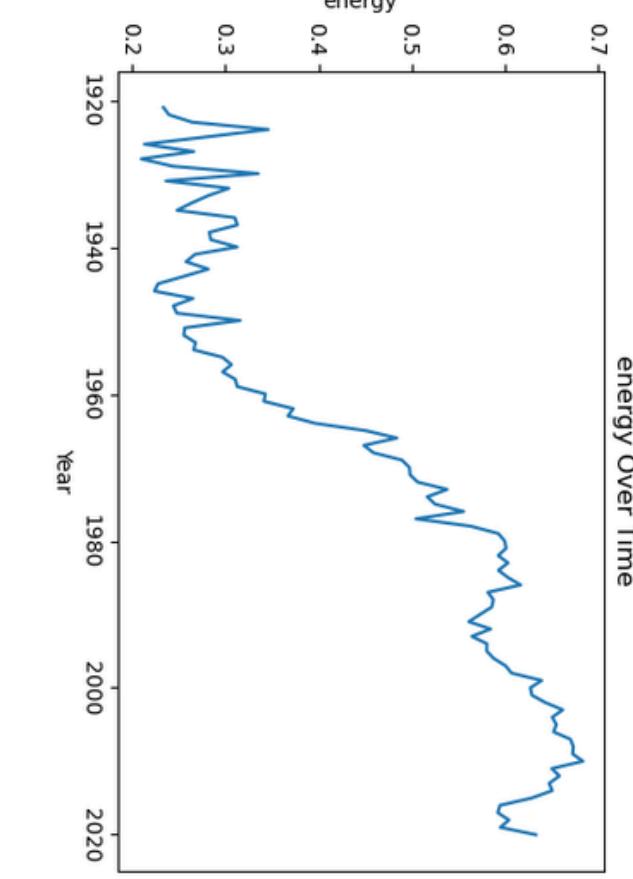
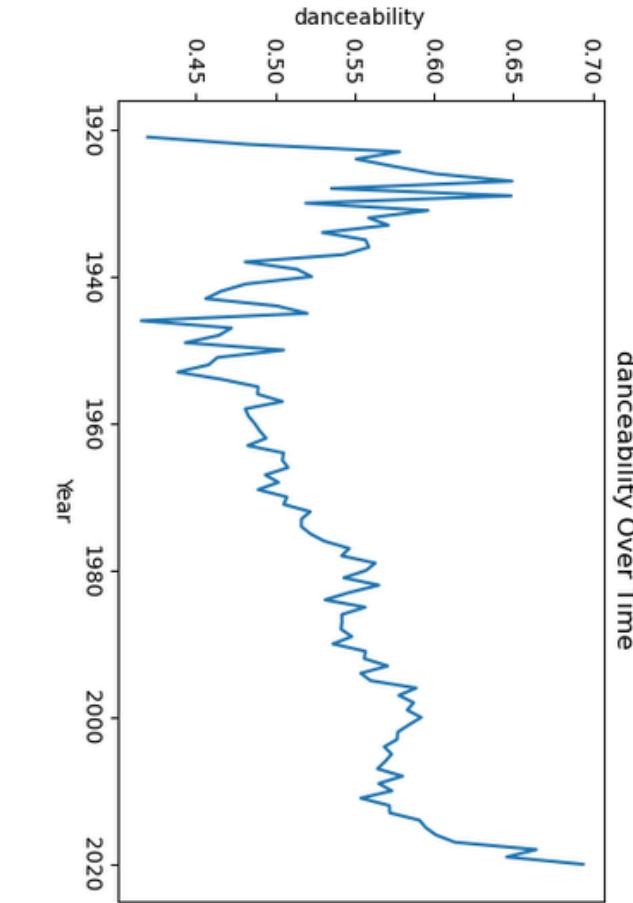
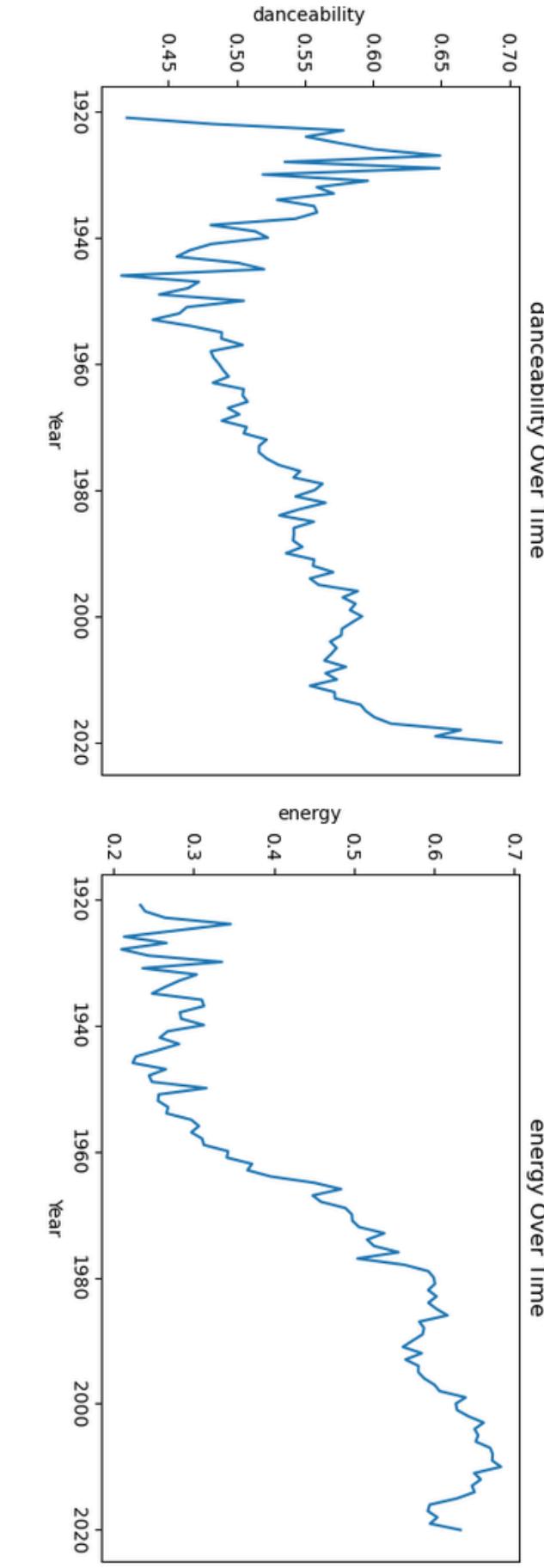
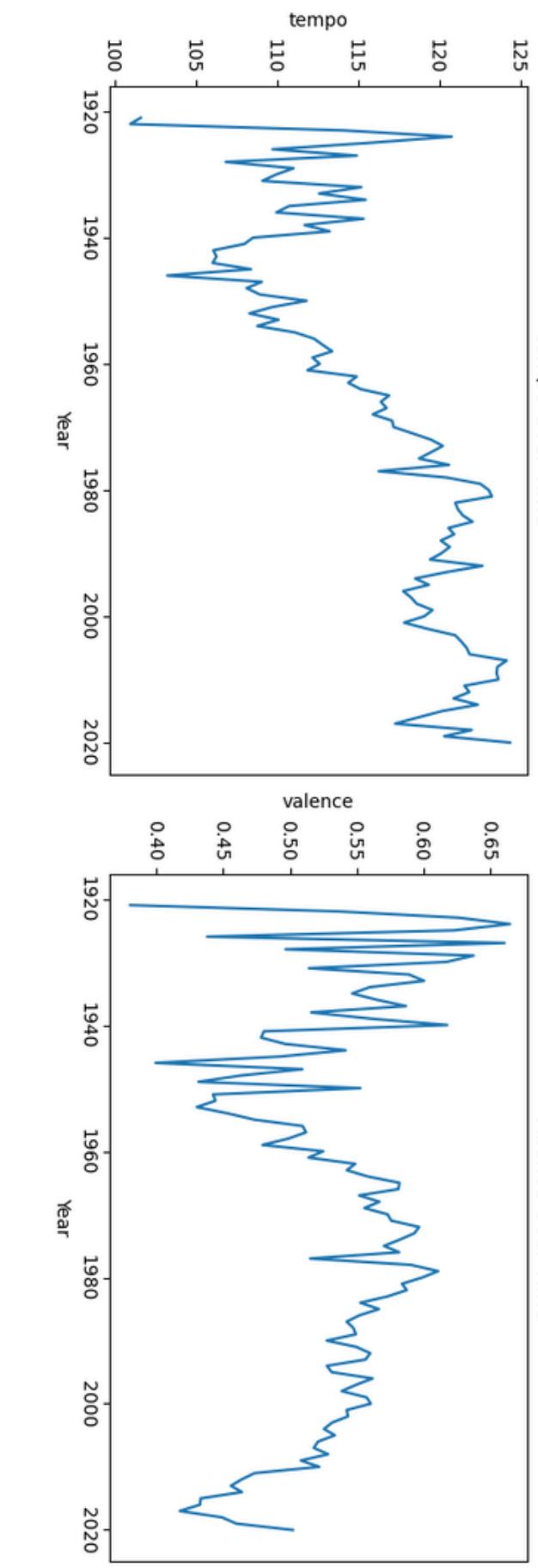
Key Insights

Popularity Trends Over Time

- Findings:
- Popularity shows an increasing trend over time
- Recent years (2010s) show higher average popularity scores
- Some fluctuations correspond to major music industry shifts

python

```
plt.figure(figsize=(12,6))
sns.lineplot(x='year', y='popularity',
             data=df2)
plt.title('Average Popularity by Year')
plt.show()
```



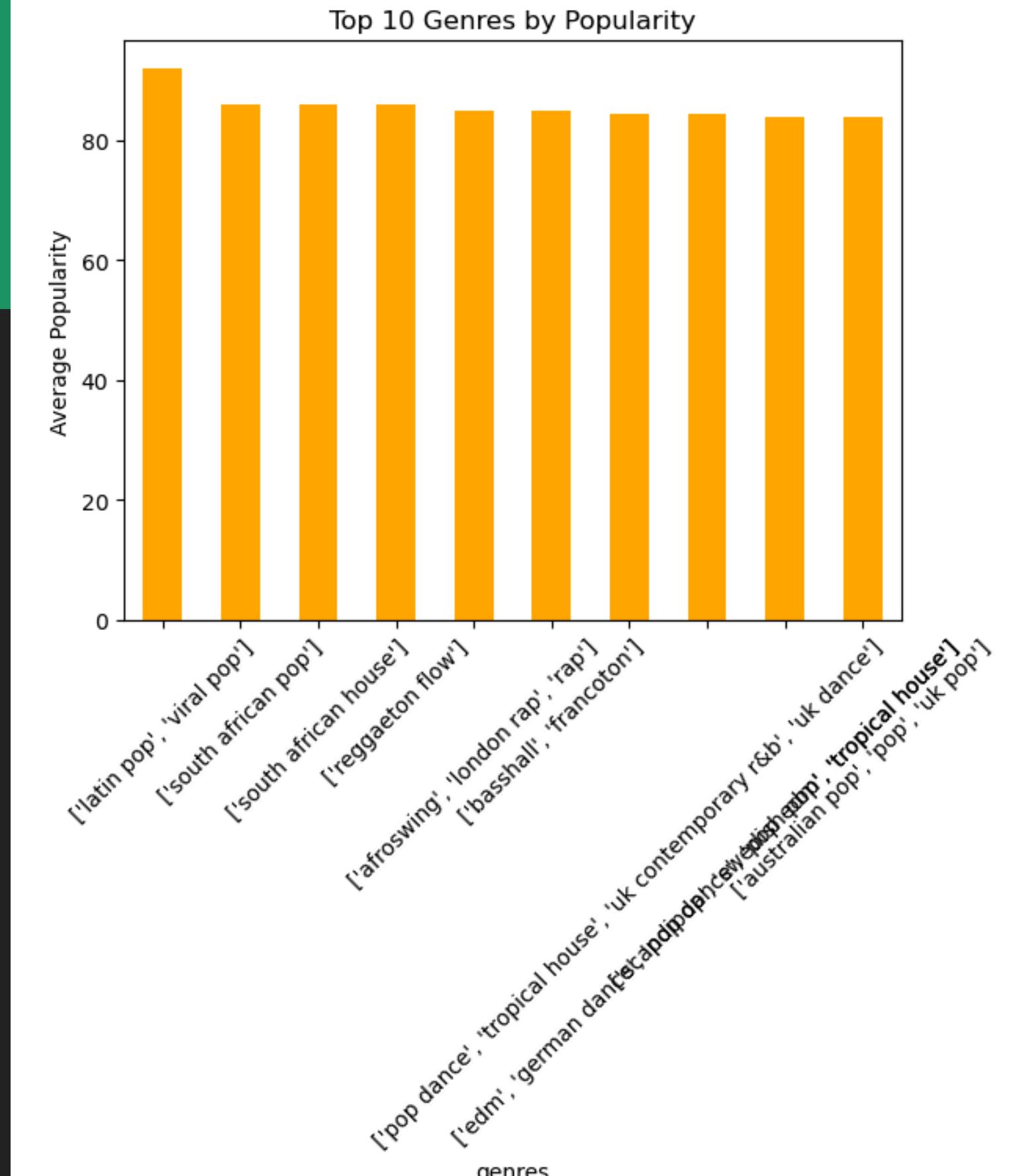
Genre Analysis

Examining popularity by genre revealed:

python

```
top_genres = df5.sort_values('popularity',
                               ascending=False).head(10)
plt.figure(figsize=(12,6))
sns.barplot(x='popularity', y='genres',
             data=top_genres)
plt.title('Top 10 Genres by Popularity')
plt.show()
```

- **Findings:**
- Pop, dance pop, and rap are consistently popular genres
- Some niche genres show high popularity within their segments
- Genre popularity correlates with mainstream appeal

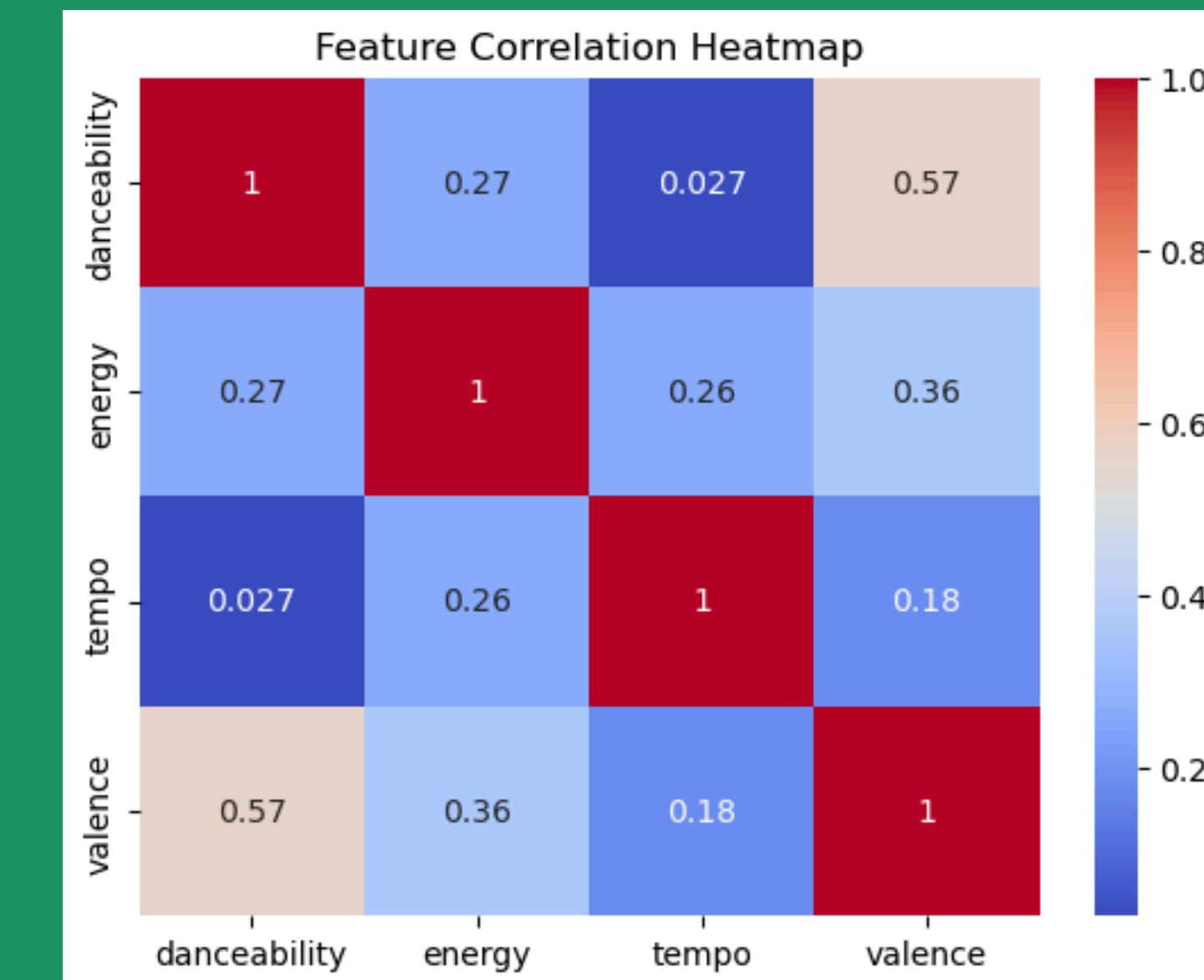


Audio Features vs. Popularity

python

```
corr_matrix = df[['danceability', 'energy', 'valence',
                  'tempo', 'popularity']].corr()
plt.figure(figsize=(10,8))
sns.heatmap(corr_matrix, annot=True,
            cmap='coolwarm')
plt.title('Correlation Between Audio Features and
Popularity')
plt.show()
```

- Interesting correlations:
- Danceability shows moderate positive correlation with popularity
- Energy has a weaker but still positive relationship
- Valence shows minimal direct correlation
- Tempo has almost no correlation with popularity



Recommendations for Artists and Industry

- Based on our analysis, we recommend:
- For Artists:
 - Focus on danceability in track production
 - Maintain moderate to high energy levels
 - Consider pop or dance pop genres for mainstream appeal
 - Experiment with tempo as it shows wide variation in popular tracks
- For Spotify:
 - Highlight danceable tracks in algorithmic recommendations
 - Consider energy levels when curating workout or focus playlists
 - Explore niche genres that show unexpected popularity
- For Listeners:
 - Explore beyond just popular tracks – many great songs exist across all popularity levels
 - Use audio features to discover new music matching your preferences

Future Work

Potential extensions for this analysis:

- Incorporate lyrics analysis for deeper insights
- Examine geographical trends in music preferences
- Build predictive models for song popularity
- Analyze playlist composition patterns
- Study the impact of collaborations on track success



Conclusion

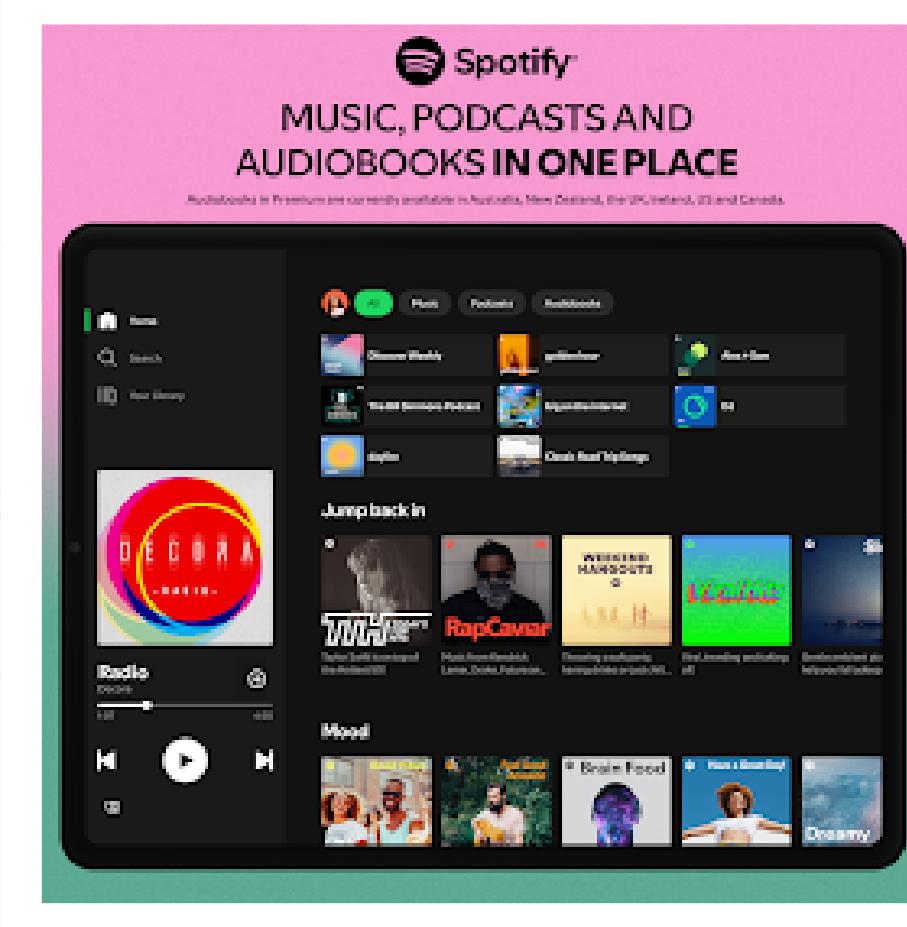
Our comprehensive analysis of Spotify data revealed fascinating insights into what makes music popular. Key takeaways include:

- Danceability and energy are important but not sole determinants of popularity
- Genre plays a significant role in track success
- Popularity has generally increased over time
- There's more to music than just popularity - many great tracks exist across all levels

This analysis provides valuable insights for artists, music industry professionals, and listeners alike to better understand and navigate the evolving music landscape.



Documentation



Thank You