Aiman Madan

Recommendation

System s

Table of Contents

Introduction

Problem Statement

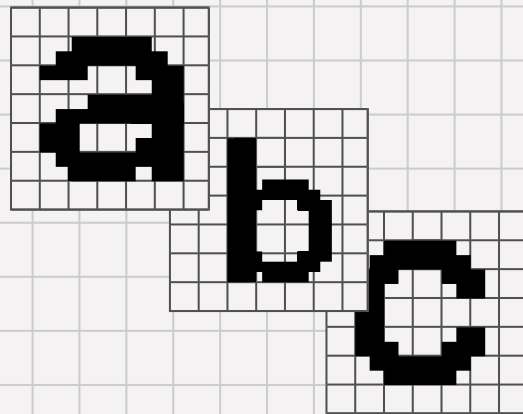Goal

Dataset

Processing

Conclusion

Introduction

What are recommendation systems?

Examples

Spotify®

Instagram

abc

Apple Music

TikTok

NETFLIX

amazon

YouTube

Problem

- Ken likes to chill

- Ken likes to listen to music

- Ken likes chill music

Problem

🔍

- Ken only likes chill music

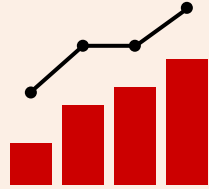- Not chill music makes Ken sad

- How can we make Ken Happy?

Goal

Create a content-based recommendation system that suggests music tracks to Ken based on the features of the tracks he like

Dataset

Features in the Dataset

Track	Artist	Danceability
Energy	Key	Loudness
Mode	Speechines	Acousticness
Liveness	Valence	Instrumentalness
Tempo	Duration	Time_signature
Chorus_hit		Sections

Source : [The Spotify Hit Predictor Dataset](#)

Dataset

12270 rows × 19 columns

LINDA LISTEN... YOU HAVE THE WRONG NUMBER

	track	artist	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	chorus_hit
0	Lucky Man	Montgomery Gentry	0.578	0.471	4	-7.270	1	0.0289	0.368000	0.00000	0.159	0.532	133.061	196707	4	30.88059
1	On The Hotline	Pretty Ricky	0.704	0.854	10	-5.477	0	0.1830	0.018500	0.00000	0.148	0.688	92.988	242587	4	41.51106
2	Clouds Of Dementia	Candlemass	0.162	0.836	9	-3.009	1	0.0473	0.000111	0.00457	0.174	0.300	86.964	338893	4	65.32887
3	Heavy Metal, Raise Hell!	Zwartketterij	0.188	0.994	4	-3.745	1	0.1660	0.000007	0.07840	0.192	0.333	148.440	255667	4	58.59528
4	I Got A Feelin'	Billy Currington	0.630	0.764	2	-4.353	1	0.0275	0.363000	0.00000	0.125	0.631	112.098	193760	4	22.62384

Time to **process the Dataset**

Dataset(Processed)

12270 rows × 19 columns

track	artist	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	artist_encoded	track_encoded
Lucky Man	Montgomery Gentry	0.559965	0.471339	0.363636	0.826531	1.0	0.006962	0.369478	0.000000	0.146656	0.541752	0.538881	0.043518	0.8	0.619040	0.514192
On The Hotline	Pretty Ricky	0.695858	0.854818	0.909091	0.863528	0.0	0.172022	0.018574	0.000000	0.135319	0.700611	0.308396	0.054562	0.8	0.709469	0.611798
Clouds Of Dementia	Candlemass	0.111303	0.836796	0.818182	0.914452	1.0	0.026671	0.000111	0.004579	0.162115	0.305499	0.273748	0.077744	0.8	0.152589	0.161224
Heavy Metal, Raise Hell!	Zwartketterij	0.139344	0.994994	0.363636	0.899265	1.0	0.153813	0.000007	0.078557	0.180666	0.339104	0.627335	0.057710	0.8	0.992337	0.347095
I Got A Feelin'	Billy Currington	0.616048	0.764706	0.181818	0.886720	1.0	0.005463	0.364458	0.000000	0.111615	0.642566	0.418310	0.042809	0.8	0.103883	0.384554

APPLICATION OF MACHINE LEARNING

Process

|	
Standardize the data	• features such as tempo (measured in BPM) and valence (a score between 0 and 1) are on vastly different scales. • Standardization ensures that all features contribute equally to the model, avoiding dominance by features with larger scales
Train PCA	• Autoencoder • Learn embeddings
Cosine Similarity	• Measures the angle between vectors, focusing on their direction rather than their magnitude

The Algorithms
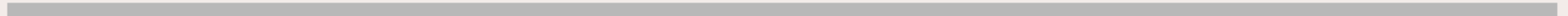
Single Song Recommendation

```python
# Find the index of the song
try:
    song_idx = df[df['track'] == song_name].index[0]
except IndexError:
    return f"Song '{song_name}' not found in the dataset."

# Print input song
print(f"Input Song:\nTrack: {song_name}\nArtist: {df.iloc[song_idx]['artist']}\n")

# Get similarity scores for the song
similarity_scores = list(enumerate(matrix[song_idx]))

# Sort by similarity scores in descending order
similarity_scores = sorted(similarity_scores, key=lambda x: x[1], reverse=True)

# Get the indices of the top_n most similar songs
recommended_indices = [i[0] for i in similarity_scores[1:top_n + 1]]

# Get recommended songs
recommendations = df.iloc[recommended_indices][['track', 'artist']]
```

Demo

```
Input Song:
Track: In Da Club
Artist: 50 Cent

Recommended Songs:
                                track                                 artist
0                          Look Alive            BlocBoy JB Featuring Drake
1                 Ma, I Don't Love Her          Clipse Featuring Faith Evans
2   I Just Wanna Love U (Give It 2 Me)                                 JAY-Z
3                          Open House                                  Guru
4                         It's So Hard   Big Punisher Featuring Donell Jones
```
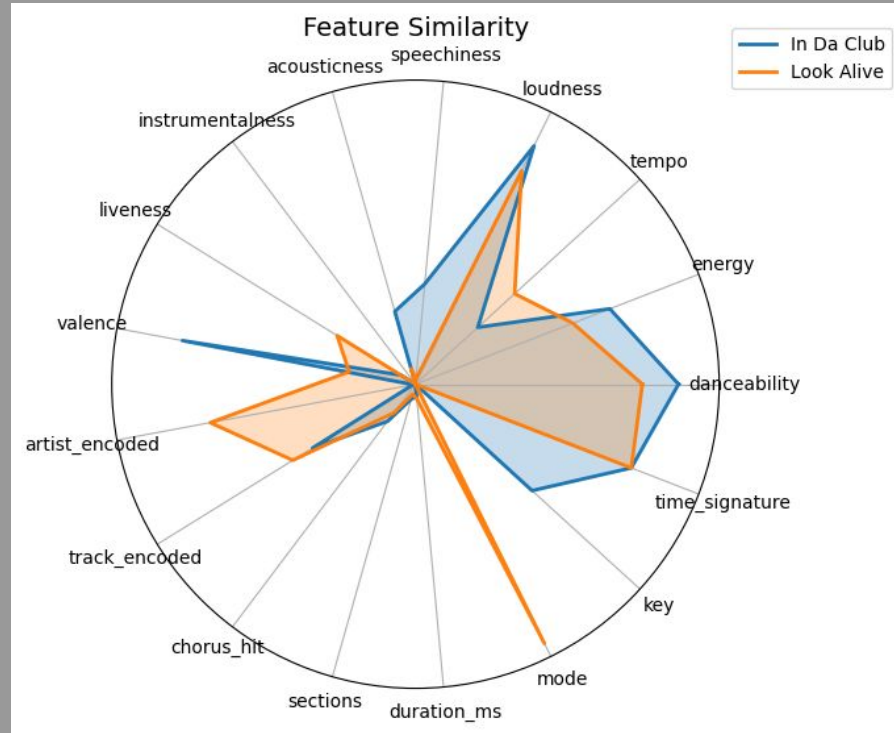
Feature Similarity

Multiple Songs Recommendation

```python
# Compute the mean of the user songs' similarity scores
aggregated_scores = sum(matrix[i] for i in song_indices) / len(song_indices)

# Get similarity scores for all songs
similarity_scores = list(enumerate(aggregated_scores))

# Sort by similarity scores in descending order
similarity_scores = sorted(similarity_scores, key=lambda x: x[1], reverse=True)

# Get the indices of the top_n most similar songs, excluding the input songs
recommended_indices = [
    i[0] for i in similarity_scores if i[0] not in song_indices
][:top_n]

# Return the track names and artists of the recommended songs
return df.iloc[recommended_indices][['track', 'artist']]
```

Demo

```
Input Songs:
Track: The Real Slim Shady, Artist: Eminem
Track: Funhouse, Artist: P!nk
Track: Goosebumps, Artist: Travis Scott


Recommended Songs:
                track                                          artist
0             Entourage                                       Omarion
1       I Need To Know                                  Marc Anthony
2                  Only                               Nine Inch Nails
3       Hole In The Head                                    Sugababes
4   I've Been Waiting  Lil Peep & iLoveMakonnen Featuring Fall Out Boy
```
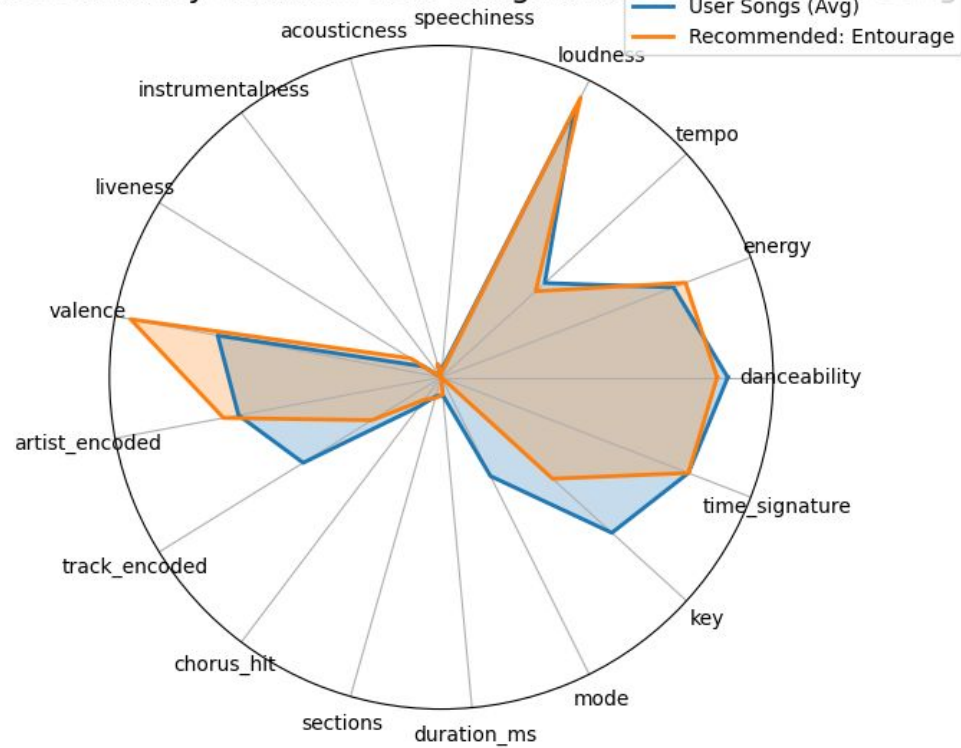
Feature Similarity Between User Songs and Recommended Song

THANK YOU