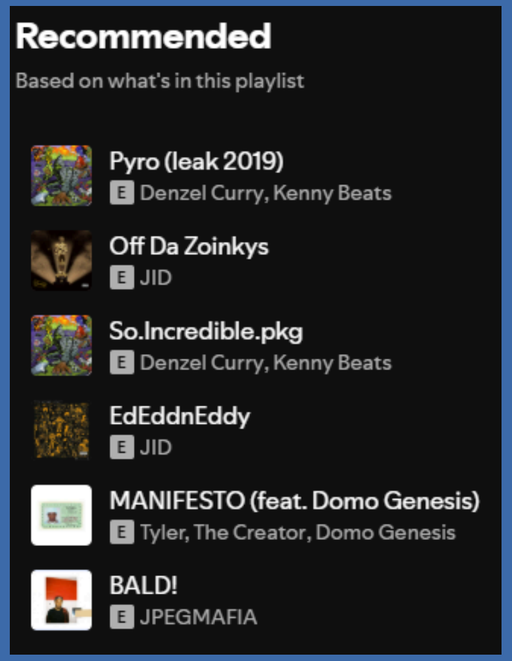
SC1015 Mini Project

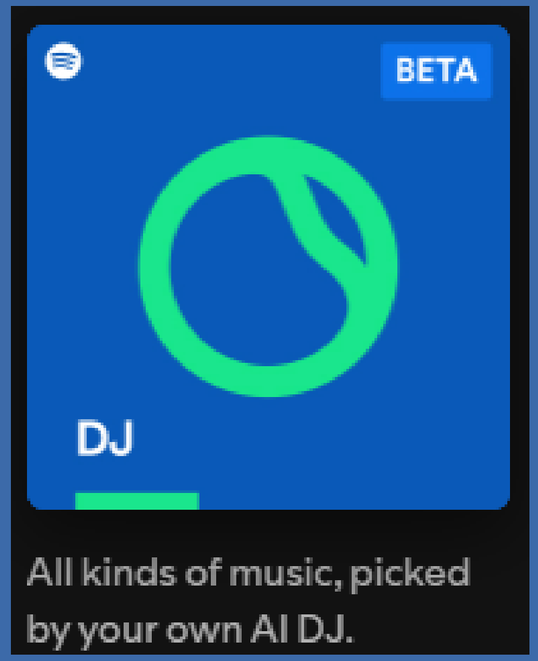
Spotify music recommendation model

by ECDS Group 6:
Ang Theng Wei
Wan Jun Jie Stanley
You Geo Yi

Motivation

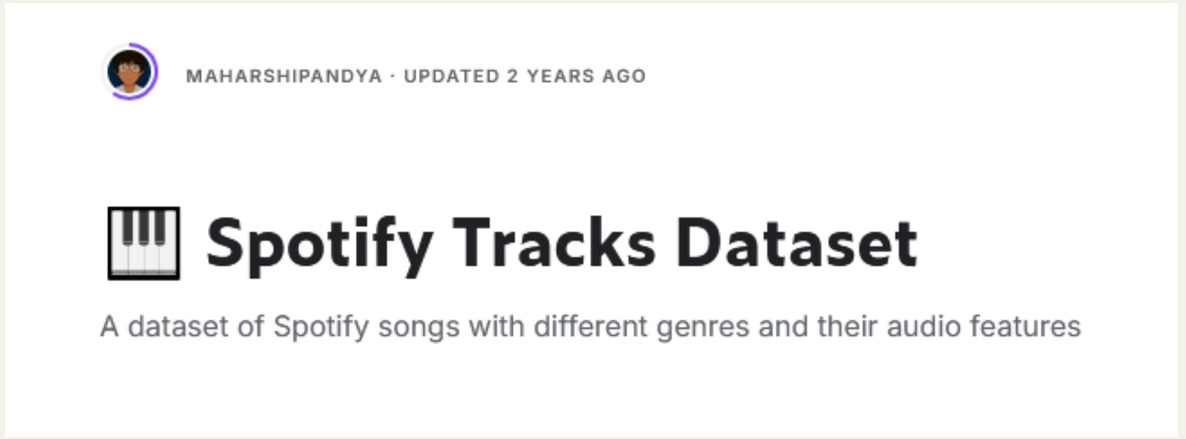
- Almost everyone uses music streaming services everyday
- Streaming services use AI to recommend songs to users
- Impacts the subscription choices for consumers
- Important for revenue





The Dataset

- Large dataset of 114k songs
- Has various sound features that can be used for our model



Data Cleaning

- The Dataset has duplicate songs
- After removing them, we have 81,344 unique songs

	artists	track_name
18	Jason Mraz;Colbie Caillat	Lucky
20	Jason Mraz	I'm Yours
22	A Great Big World;Christina Aguilera	Say Something
28	Jason Mraz	Winter Wonderland
29	Jason Mraz	Winter Wonderland
30	Jason Mraz	Winter Wonderland
31	Jason Mraz	Winter Wonderland
34	Brandi Carlile;Sam Smith	Party of One
35	Brandi Carlile;Sam Smith	Party of One
39	KT Tunstall	Lonely This Christmas

```
duplicate[['track_name', 'artists']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 32656 entries, 18 to 113991
```

```
cleaned_tracks = alltracks.drop_duplicates(subset=['track_name', 'artists'], keep='first')
```

```
cleaned_tracks.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 81344 entries, 0 to 113999
```

EDA

Variables in dataset:

- Abstract Features: Danceability, Energy, Speechiness (In decimal numbers)
- Other variables: Duration, Time Signature, Popularity

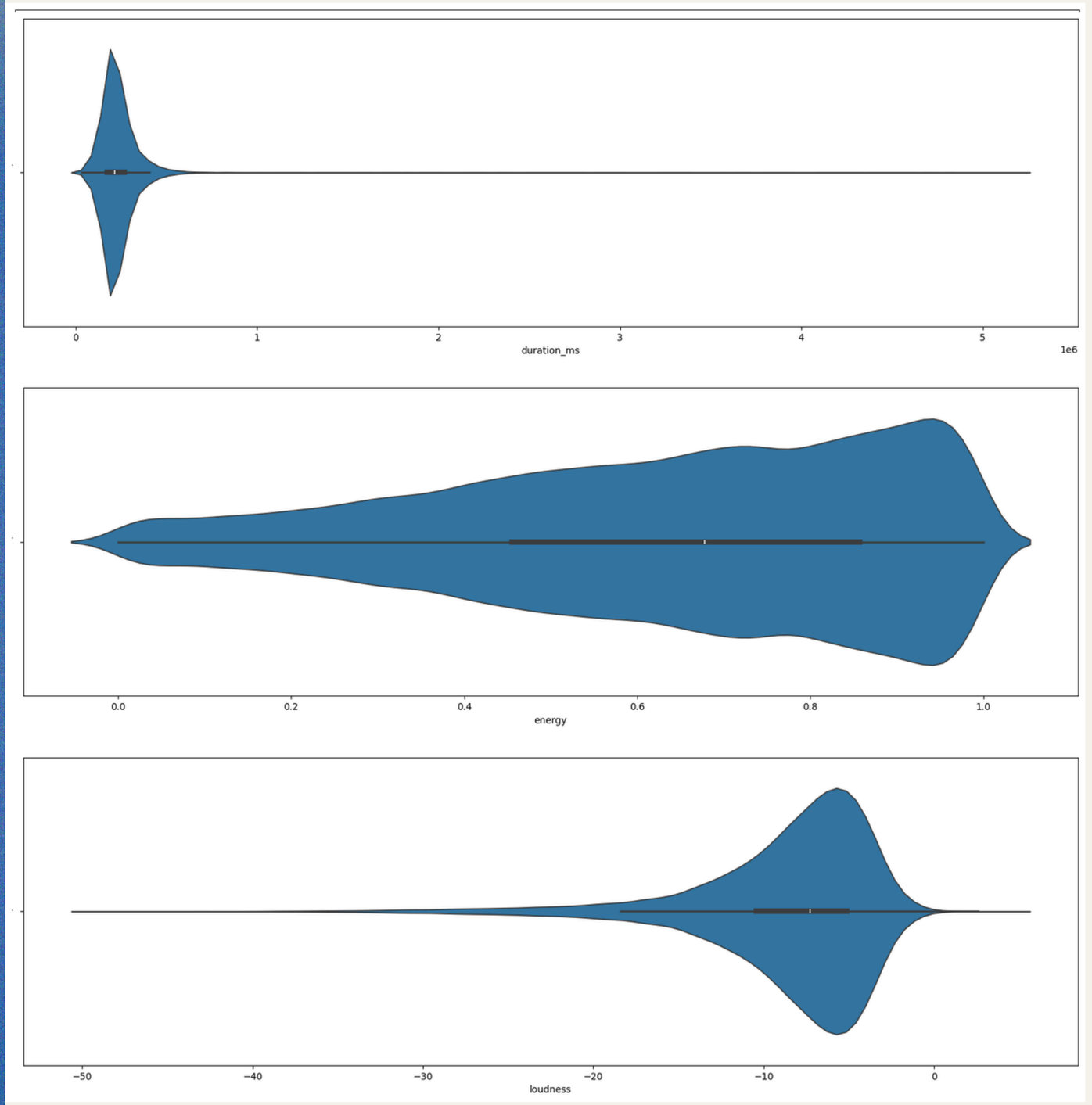
EDA

Plotting distribution of key sound features:

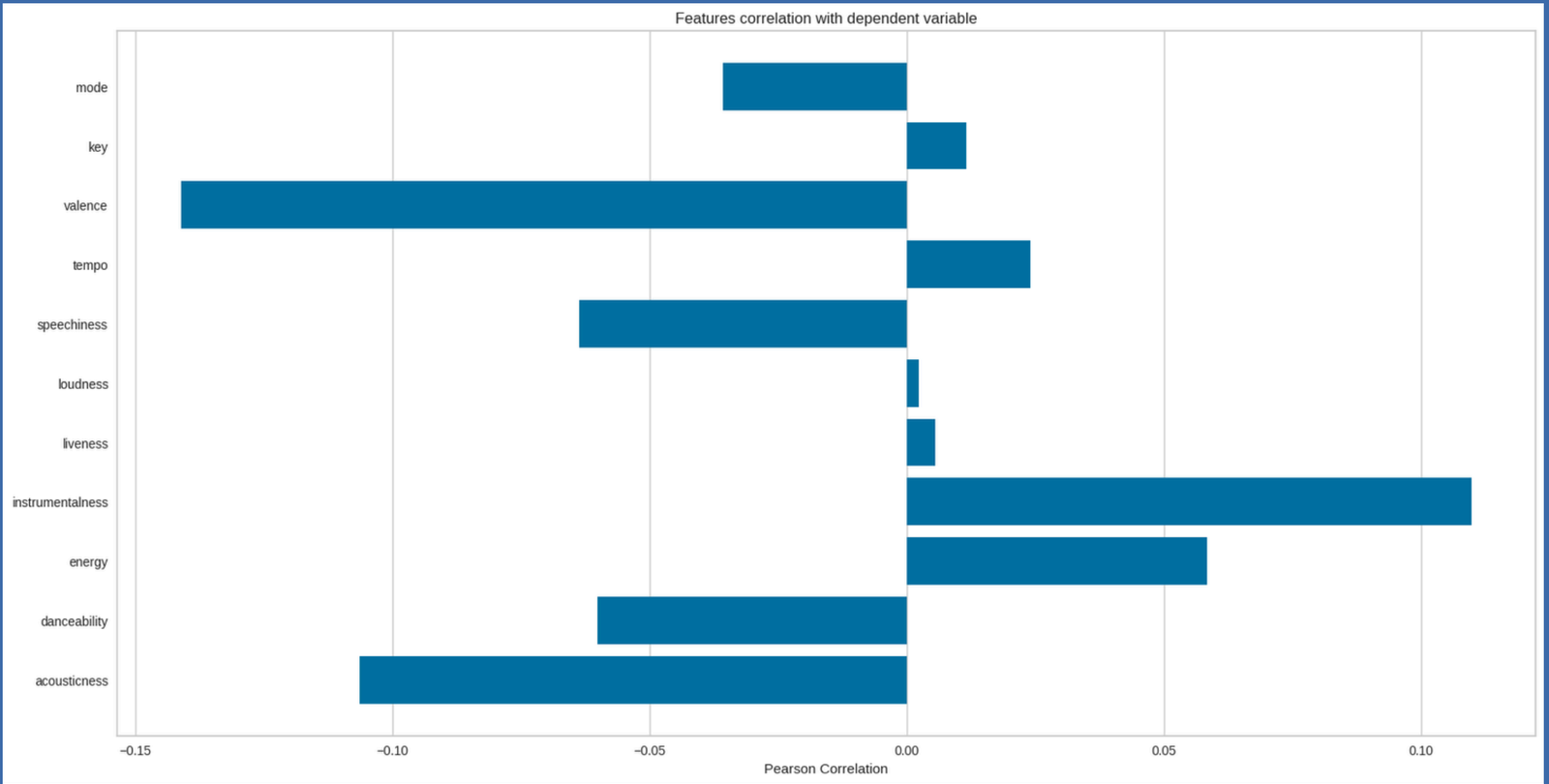
Loudness

Energy

Duration



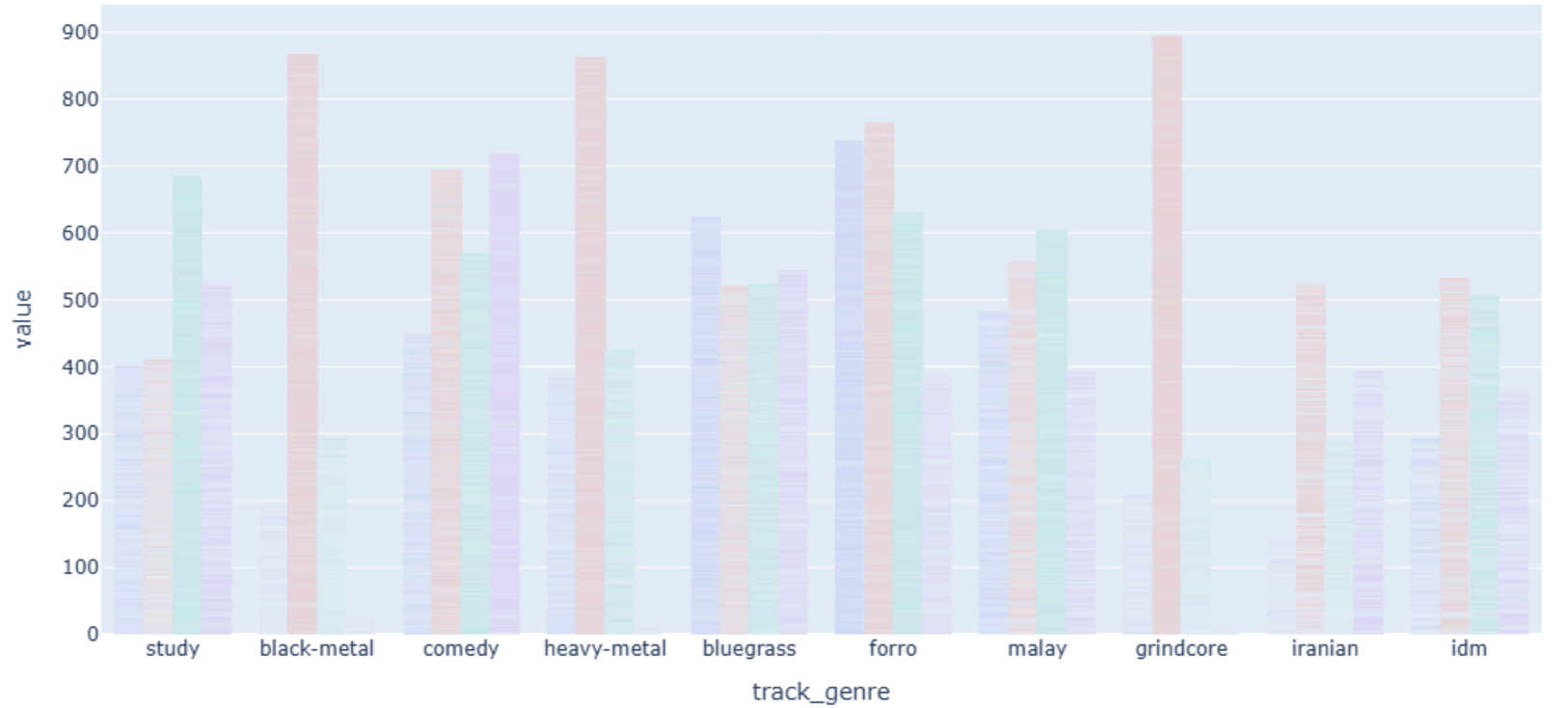
EDA: Does duration affect sound features?



Weak correlation between sound features and duration

Do not need to account for runtime when clustering sound features later on

EDA : Sound Features in Genres



Genre closely tied to musical style, consisting of sound features

EDA : Sound Features for clustering

- Use Decision Tree Classifier:

To build a prediction model to predict clusters

To check if prediction model works → sound features have enough structure for clustering to be meaningful

EDA : Sound Features for clustering

```python
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier



variables= {
    "sound_features": ['danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
                       'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo'],

}

results = []

for name, variable in variables.items():
    X = cleaned_tracks[variable].copy()
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)


    kmeans = KMeans(n_clusters=7, random_state=20)
    clusters = kmeans.fit_predict(X_scaled)


    cleaned_tracks.loc[:, "cluster"] = clusters #to prevent the SettingwithCopyWarning

    # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(X, clusters, test_size=0.25, random_state=20)

    # Decision tree
    dectree = DecisionTreeClassifier(max_depth=5)
    dectree.fit(X_train, y_train)

    test_accuracy = dectree.score(X_test, y_test)
    results.append((name, test_accuracy))

for variable_name, acc in results:
    print(f"variable: {variable_name}, Test Accuracy: {acc:.4f}")
```

```
variable: sound_features, Test Accuracy: 0.8224
```

Train-test split doesn’t test predictive power as we used unsupervised clustering labels, **but** shows that K-Means clustering is consistent enough for a decision tree to learn and generalize.

EDA : Tools used

Visualisation using

- matplotlib
- seaborn
- yellowbrick
- plotly

Techniques using

- KMeans (to be further explored)
- Decision Tree Classifier

(KEY INSIGHTS)

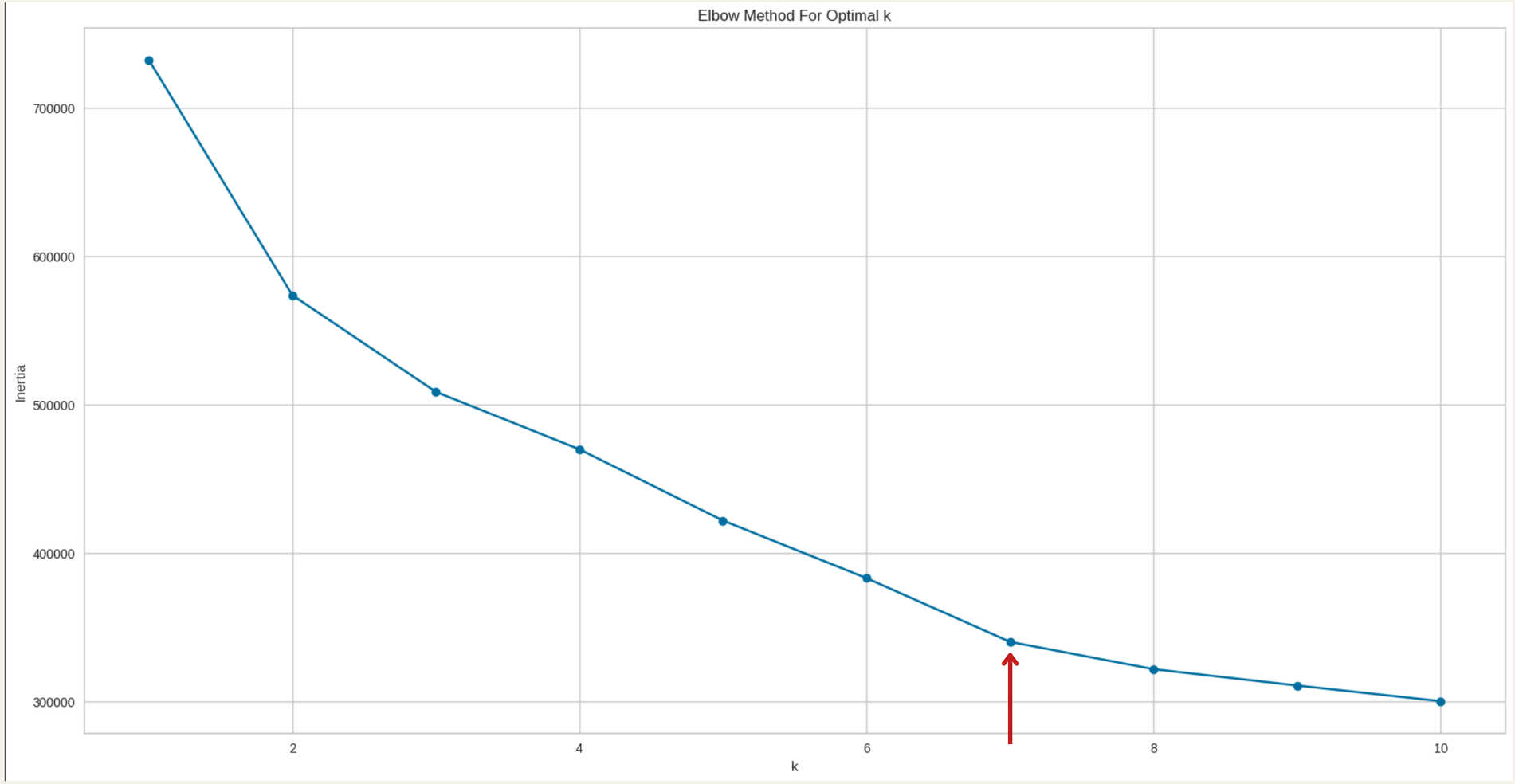
1. **Presence of many outliers and skewness → wide diversity of songs, meaningful to cluster**
2. **Duration is not a key variable in clustering**
3. **Validate using sound features for KMeans Clustering**

Recommendation Model

Clustering songs based on sound features:

1. Find optimal number of clusters
2. Group songs into respective clusters
3. Visualise clusters
4. Build a recommend_song function

Choosing no. of clusters



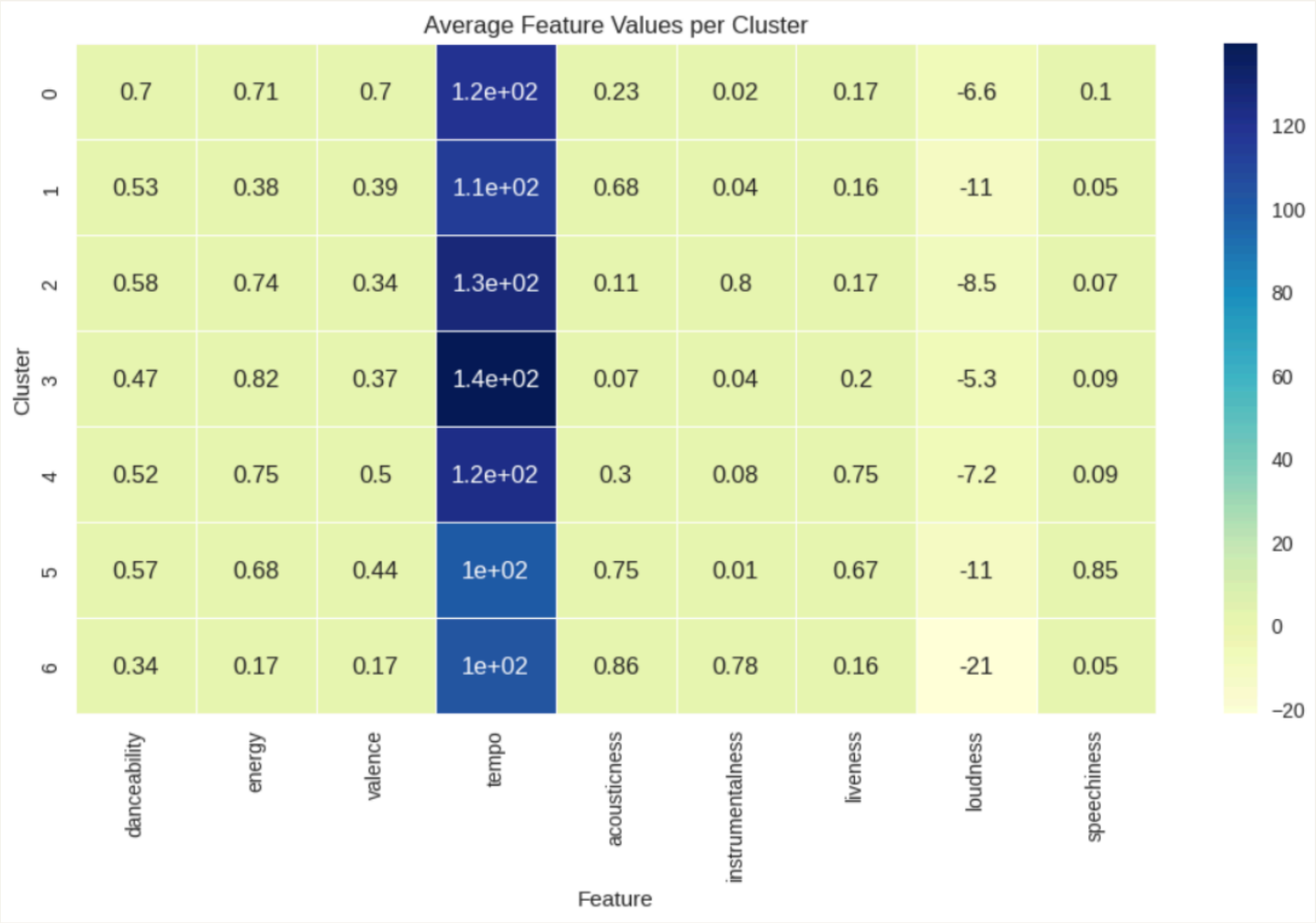
Finding x clusters, beyond which is meaningless to further cluster

Cluster Details

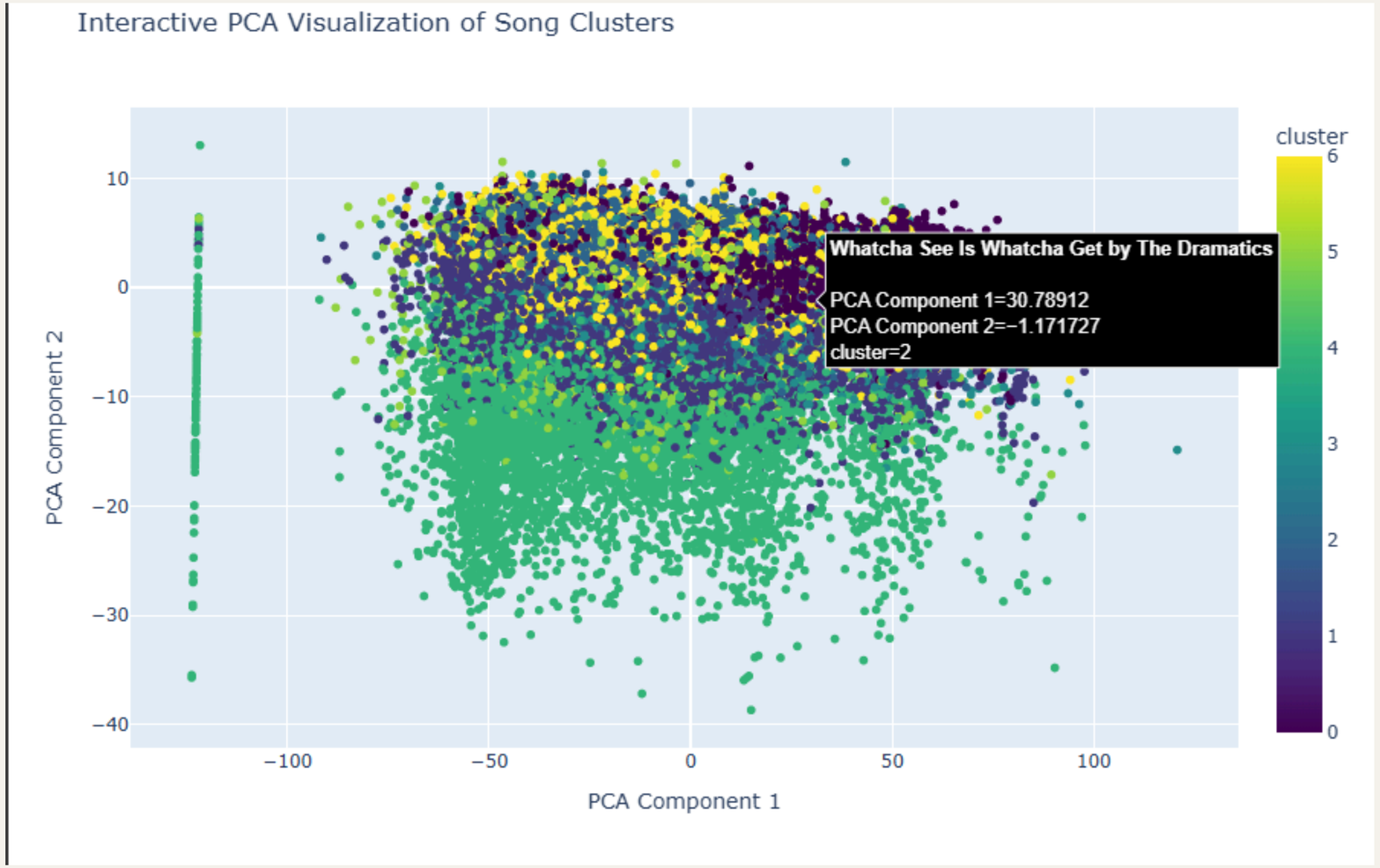
	track_name	artists	cluster
6457	Blessed He With Boils	Xanthochroid	4
96017	Contrato Vitalício - Ao Vivo	Akatu;Ferrugem	4
83594	Somebody Like You	Vicetone;Lena Leon	0
112631	Deli Divane	Eda Baba	1
59913	Chargah	Arash Pandi	2
92795	Caramuru Calibre 32	Hillbilly Rawhide	3
99813	Please Don't Go Home Yet	Stephen Sanchez	1
111817	Closer	Lamb	0
28513	Empires	Rogue	3
60608	Gladdest Night	Alkaline;Black Shadow	3

	danceability	energy	valence	tempo	acousticness
6457	0.269	0.795	0.121	119.958	0.000147
96017	0.641	0.752	0.691	149.901	0.580000
83594	0.620	0.851	0.816	122.011	0.136000
112631	0.760	0.421	0.855	100.012	0.872000
59913	0.262	0.592	0.192	139.777	0.225000
92795	0.398	0.952	0.882	154.504	0.605000
99813	0.602	0.131	0.269	98.412	0.850000
111817	0.863	0.651	0.869	134.039	0.004170
28513	0.349	0.774	0.398	173.277	0.001870
60608	0.658	0.733	0.285	133.836	0.042600

	instrumentalness	liveness	loudness	speechiness
6457	0.277000	0.7780	-7.508	0.0688
96017	0.000000	0.9500	-6.677	0.0439
83594	0.000144	0.0999	-2.727	0.0332
112631	0.000000	0.1340	-8.699	0.0496
59913	0.929000	0.0827	-6.903	0.0810
92795	0.000040	0.0783	-3.471	0.0786
99813	0.000000	0.0817	-10.587	0.0324
111817	0.143000	0.0724	-8.324	0.0907
28513	0.000134	0.3060	-6.312	0.0539
60608	0.000000	0.2260	-3.621	0.2320



Cluster visualisation using PCA



Distinct groups - successful clustering

Scattered - not as successful, could be due to limitation of dataset

recommend_songs

```
[ ]  print(recommend_songs("Unholy (feat. Kim Petras)", cleaned_tracks, features))
```

```
                                        track_name  \
856                                          White
69994     Parayathe Vayyen (From “Kottaram Vaidyan”)
98575  Monster - From "Frozen: The Broadway Musical"
90812                                        Engaño
97622                                Som de Cristal

                                              artists  distance  cluster
856                                      Tim Halperin  0.292382        1
69994                                          Sujatha  0.300523        1
98575  Caissie Levy;John Riddle;Original Broadway Cas...  0.350744        1
90812                                           Josue  0.513375        1
97622                                 Joaquim e Manuel  0.537286        1
```

Return 5 songs with the closest audio features to the input song within the same cluster

Areas for improvement

1. Expanding songs to Spotify Database
2. Customise to include user's Spotify ID
3. Look to other clusters for better accuracy

Thank you