# Kmeans as a possible tool for efficient playlist creation

…

# PROTOTYPE - The Data

Initial dataset of Spotify with over 5,000 songs and 13 audio features

acousticness

danceability

energy

valence

tempo

loudness

355 outliers

speechiness

437 outliers

instrumentalness

~~mode~~

~~duration_ms~~

~~key~~

~~liveness~~

~~time_signature~~

# PROTOTYPE - The Data

8 remaining audio features and 4,421 songs for clustering

acousticness

valence

loudness

energy
0 - 1

tempo
0 - 214

danceability

speechiness

instrumentalness

StandardScaler → *scales features according to the standard deviation of the feature*

# PROTOTYPE - Techniques & Metrics

Simplifying data with Principal Component Analysis (PCA)

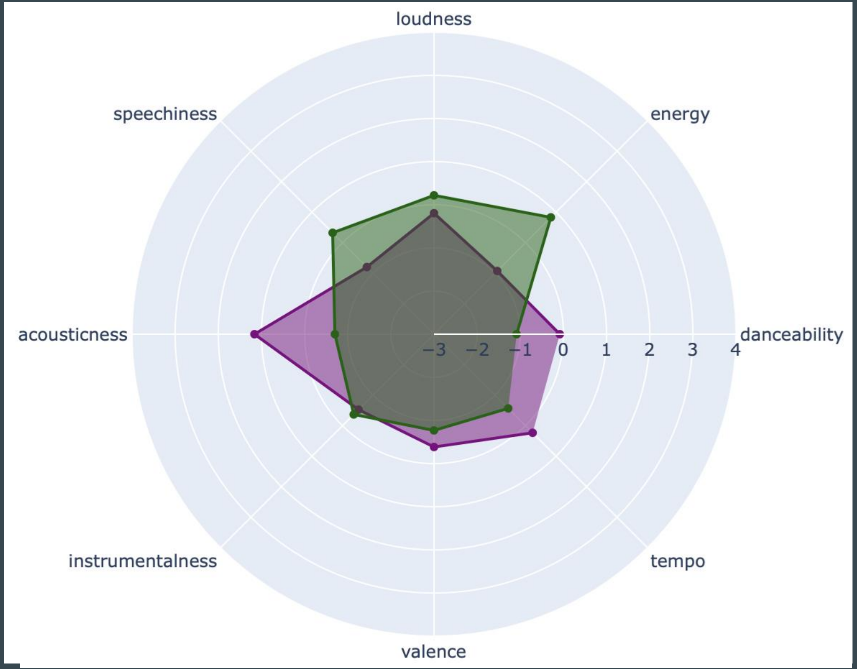
- ➢ 7 remaining principal components

Defining ideal amount of clusters with inercia score and silhouette score

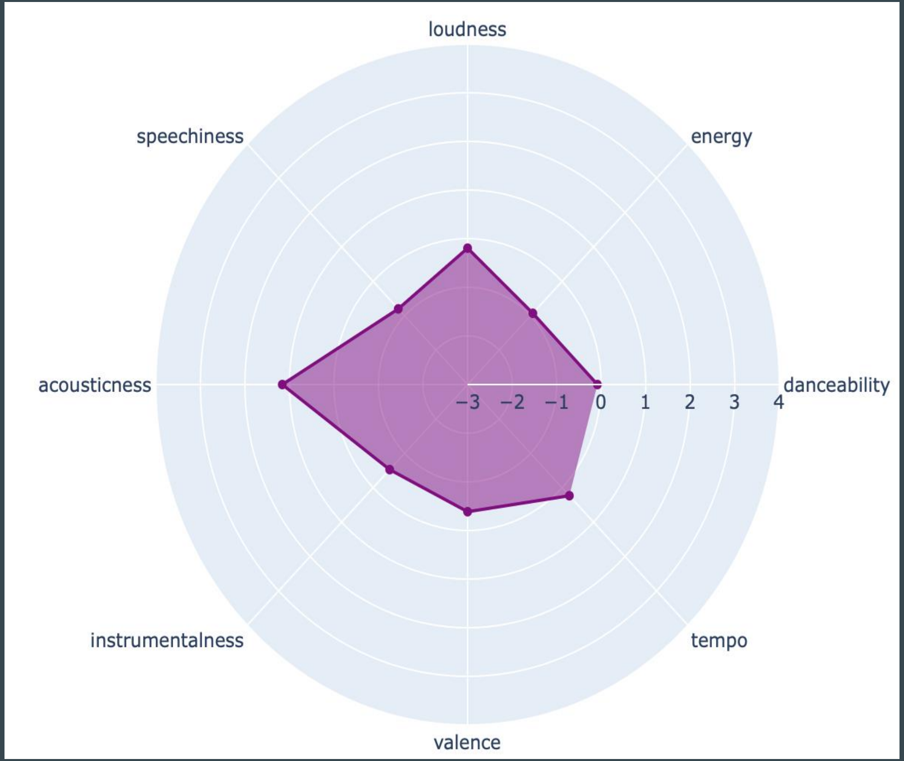
- ➢ 24 clusters for main dataset
- ➢ 9 clusters for outliers

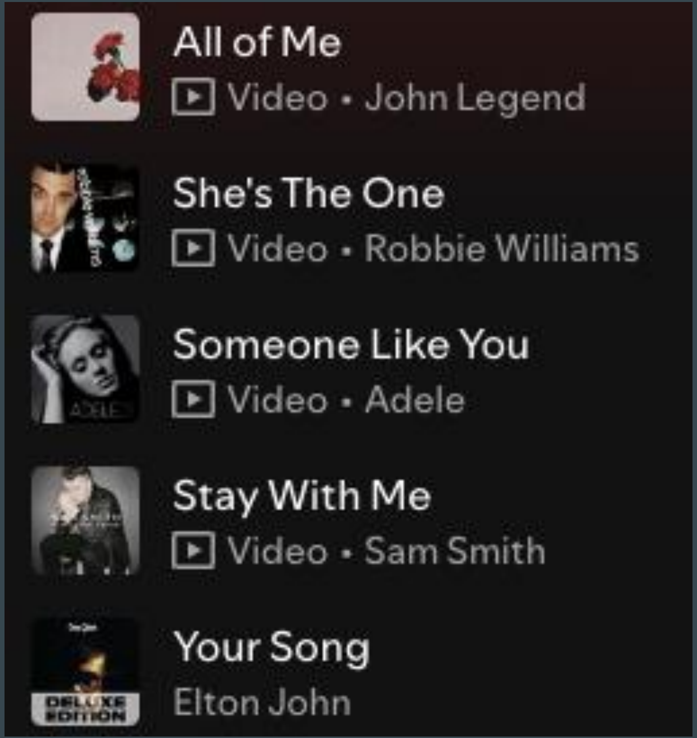
# The playlists

- Total amount: 33 playlists
  - **24 playlists** by clustering the main dataset
  - 9 additional playlists by clustering outliers
- Size: between 122 and 413 songs per playlist



# 1. Playlist: “Crying in the Car”





All of Me

▶ Video • John Legend

She's The One

▶ Video • Robbie Williams

Someone Like You

▶ Video • Adele

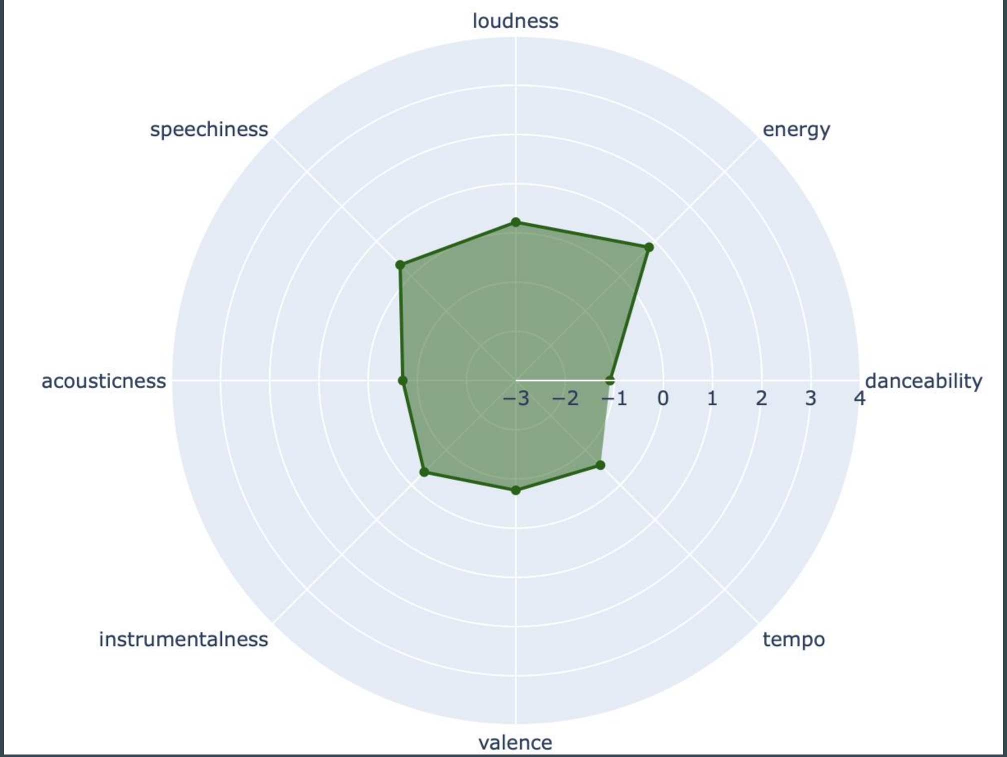
Stay With Me

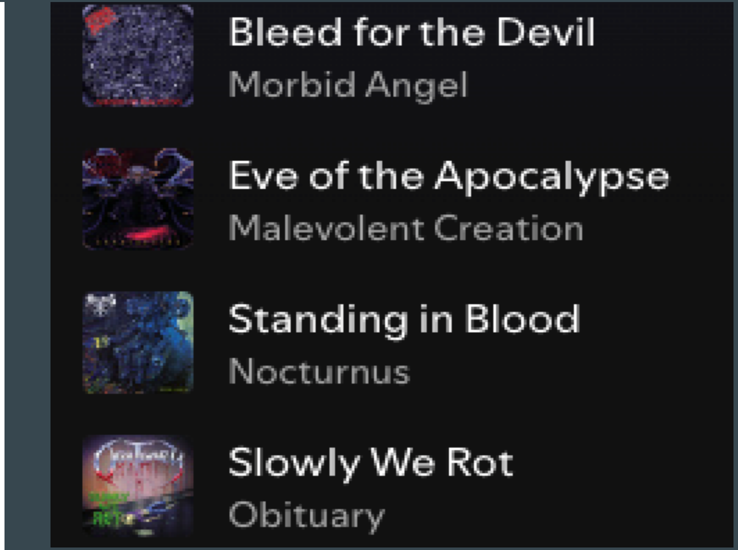
▶ Video • Sam Smith

Your Song

Elton John

# 2. Playlist: “Shattered Eardrums”



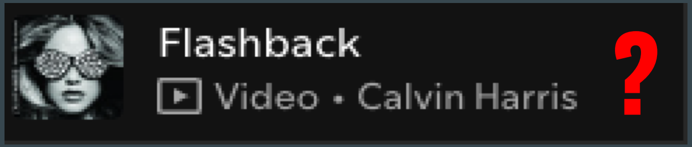


Bleed for the Devil
Morbid Angel

Eve of the Apocalypse
Malevolent Creation

Standing in Blood
Nocturnus

Slowly We Rot
Obituary



Flashback
Video • Calvin Harris

?

# SPOTIFY FEATURES are helpful, but we can’t rely on them

- Features give an general direction
- Sometimes it is not comprehensible why a song has a certain score

# KMEANS - an effective tool, but with limits

## PROS

- Clustering by similar feature ranges works well.
- Simplicity and faster convergence.

## CONS

- Does not account for differences like a human ear would do.
- Selection of K-value is difficult.
- Sensitive to data with high variance.

# Conclusion & Next steps

- **KMeans** algorithm **worked fine** in this scenario, but **human intervention was necessary**
- Since lot of data preprocessing was required, exploring **other algorithms** might be **effective**.
- Additional clustering with special target for the playlists
  (e.g. workout, studying etc.)

# Thank you

# Business questions

- **Are Spotify's audio features able to identify "similar songs", as defined by humanly detectable criteria?** When you listen to two rock ballads, two operas or two drum & bass songs, you identify them as similar songs. Are these similarities detectable using the audio features from Spotify?
- **Is K-Means a good method to create playlists?** Would you stick with this algorithm moving forward, or explore other methods to create playlists?

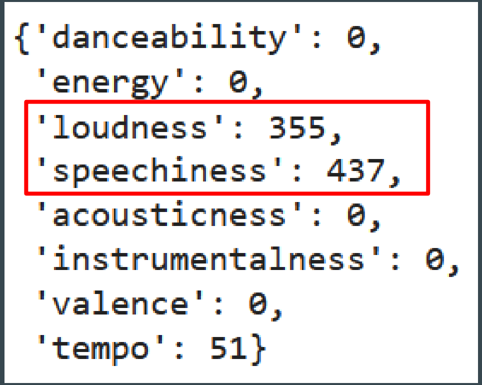
# Presentation Content:

You are **presenting your prototype** and answering these core questions:

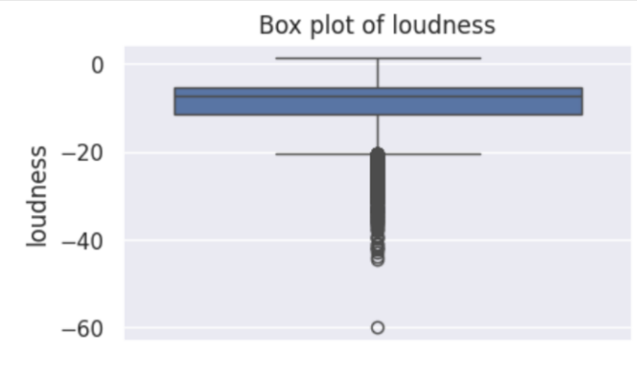
1. How did you create your prototype? (This does **not** mean showing off your code!)
   - How many playlists (clusters) are there?
   - What audio features did you use and what did you drop? Why?
2. Is the prototype effective at creating cohesive playlists?
   - Showcase one or two playlists to evaluate the prototype's performance.
3. Are Spotify's audio features capable of identifying "similar songs" as defined by humanly detectable criteria?
   - What kind of data might help us create better playlists?
4. Is K-Means a good method for creating playlists?
   - Provide pros and cons.
5. What would be your next steps if you continued with this project?

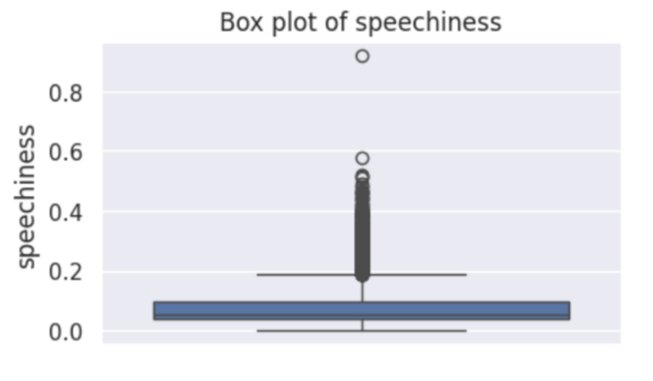
# PROTOTYPE - The Data

## Outliers



```
{'danceability': 0,
 'energy': 0,
 'loudness': 355,
 'speechiness': 437,
 'acousticness': 0,
 'instrumentalness': 0,
 'valence': 0,
 'tempo': 51}
```

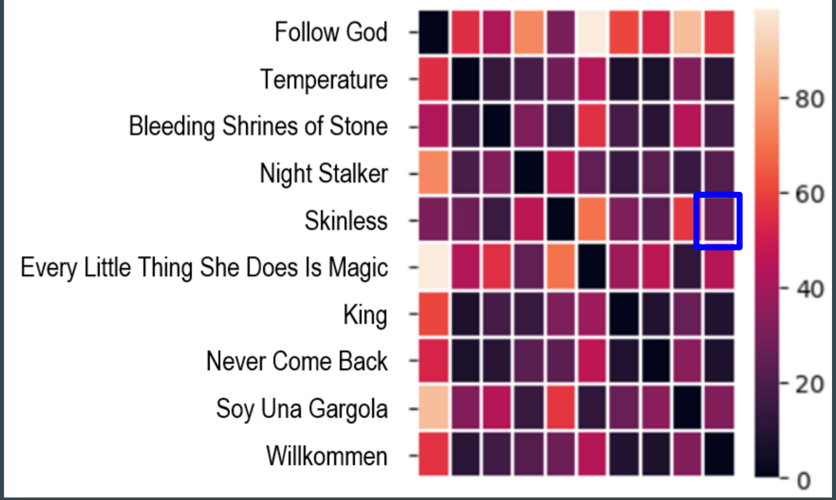




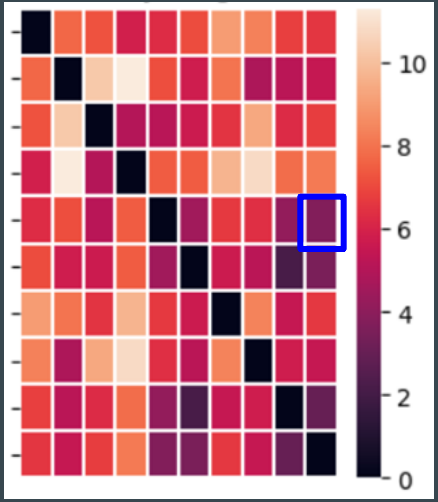
# PROTOTYPE - The Data

StandardScaler → *scales features according to the standard deviation of the feature*

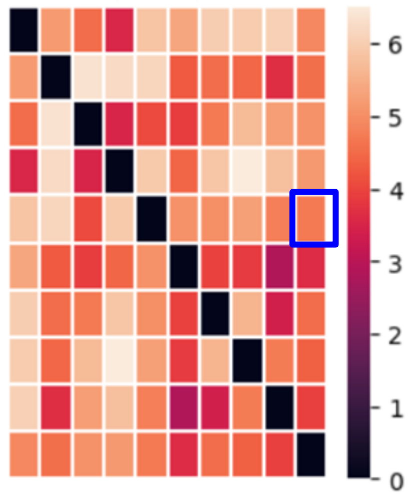
Without Scaling



RobustScaler

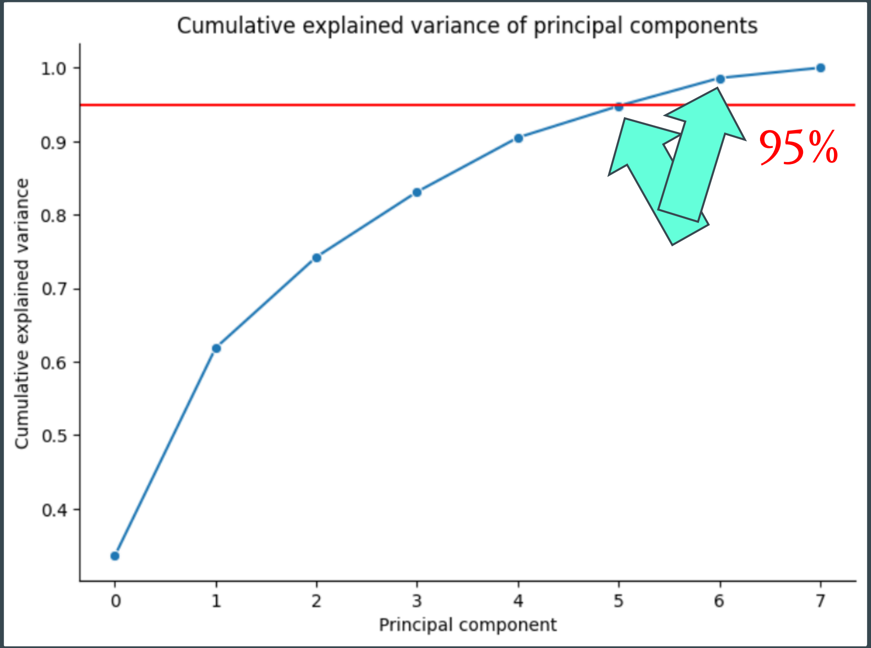


StandardScaler



# PROTOTYPE - Techniques & Metrics

## Using PCA to simplify our data before clustering

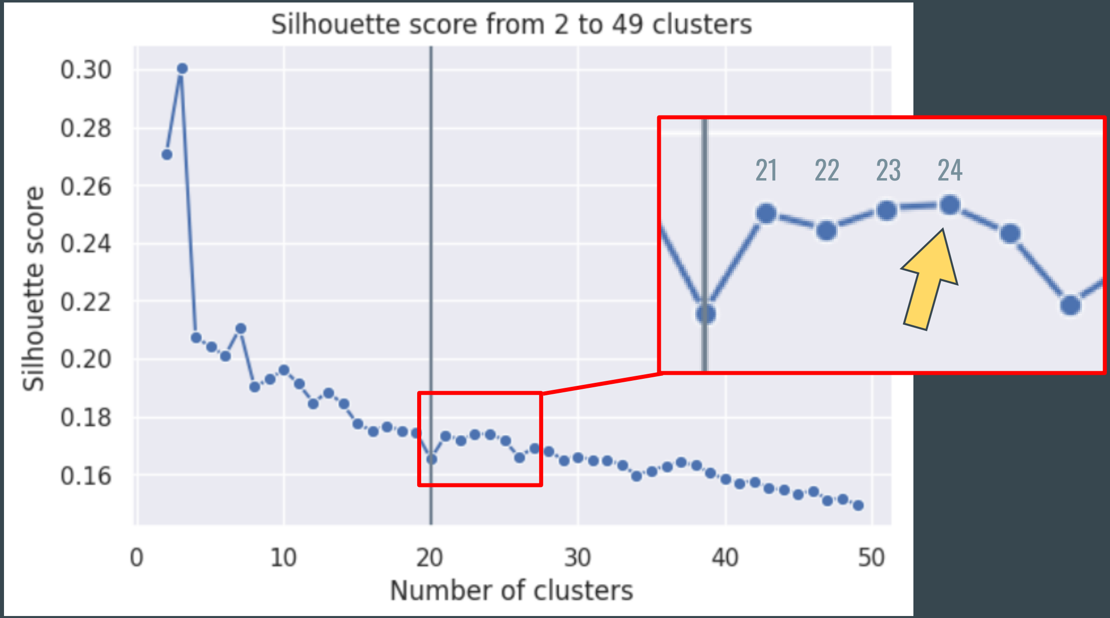


How many principal components should be kept?

↪ 7 principal components

# PROTOTYPE - Techniques & Metrics

How many Clusters? ⇒ 24 Clusters


Silhouette score from 2 to 49 clusters
Silhouette score
Number of clusters