PLAYLIST 4 MooSIC

UNSUPERVISED MACHINE LEARNING PROJECT

By: Foram, Neha & Payal

Introduction

- **Moosic** startup that creates playlists.
- **Users** can subscribe to **website** and listen playlists through their preferred **Music App** (be it Spotify, Apple Music, Youtube Music…).
- Playlists have a **personal touch**, each playlist encapsulates a certain “mood” or “style”.
- **Degree of Atomisation** to playlist creation.
- Dataset that has been collected from the **Spotify API** contains audio features for **5000 songs.**

QUESTIONS FOCUSSED:

1. **Are Spotify’s audio features able to identify “similar songs”, as defined by humanly detectable criteria?**
2. **Is K-Means a good method to create playlists?**

Approach

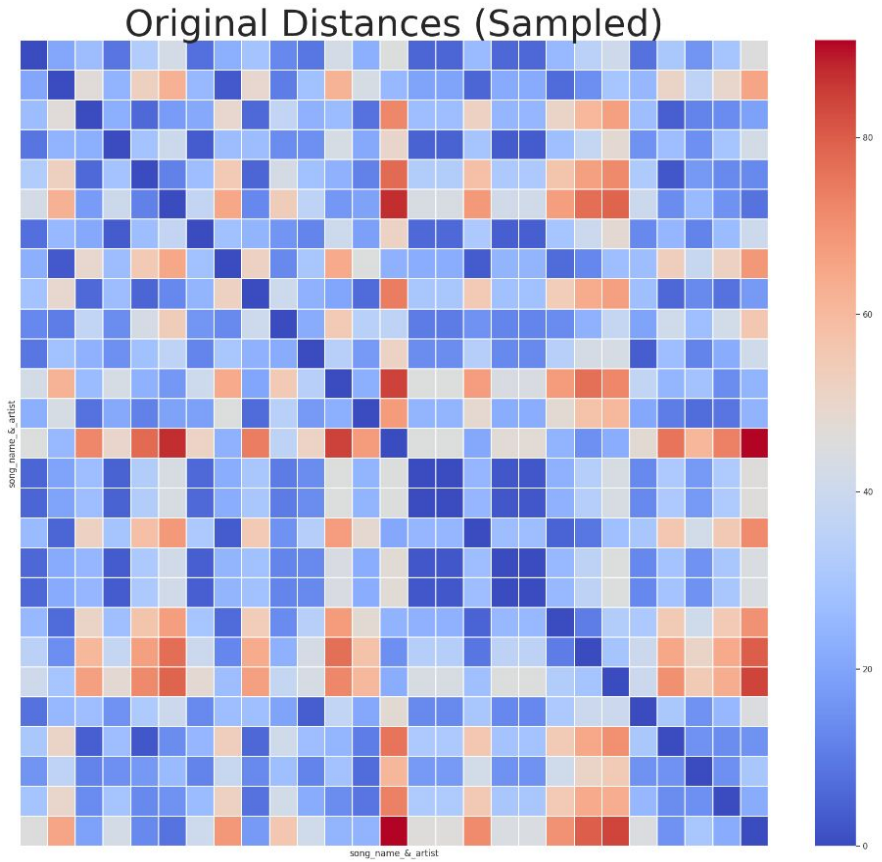
1. **Data Cleaning**
2. **Feature Selection**: Drop irrelevant or highly correlated features.
3. **Feature Scaling**: Choose an appropriate scaling method.
4. **PCA**: Apply PCA on the scaled data to reduce dimensionality before clustering.
5. **K-means Clustering**:
    - Determine the optimal number of clusters (k) using methods.
    - Apply K-means to the reduced-dimension data (from PCA).
6. **Analysis**: Evaluate and interpret the clustering results.

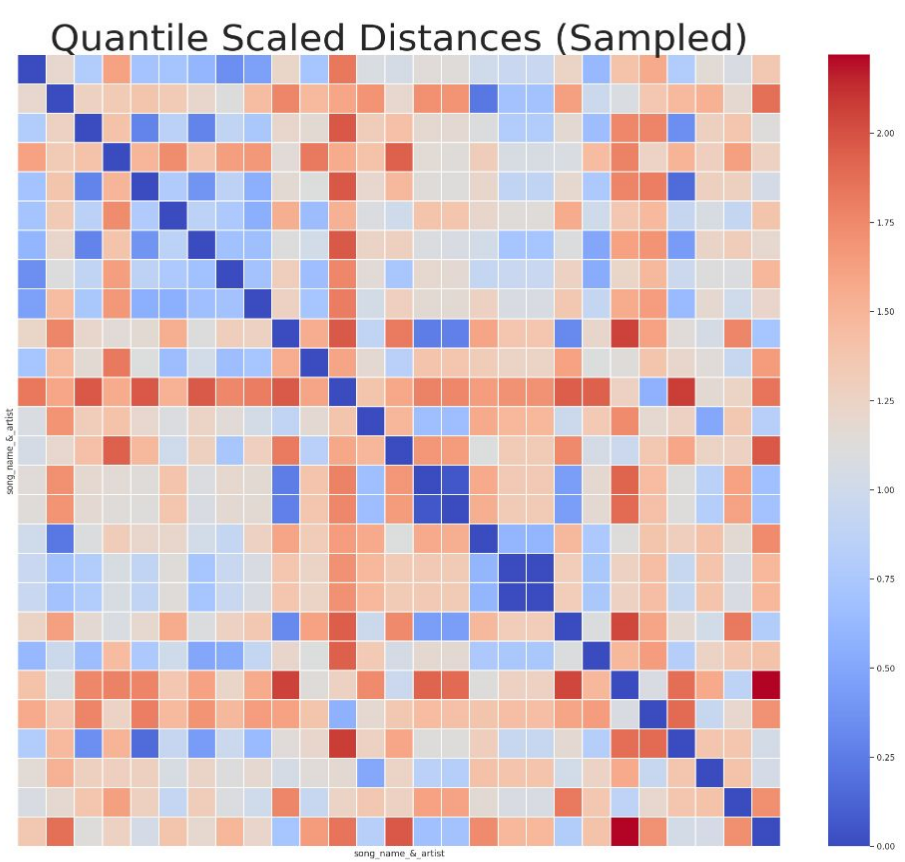
Feature selection

From the available features, we continued with the one that resembles more to human the mood

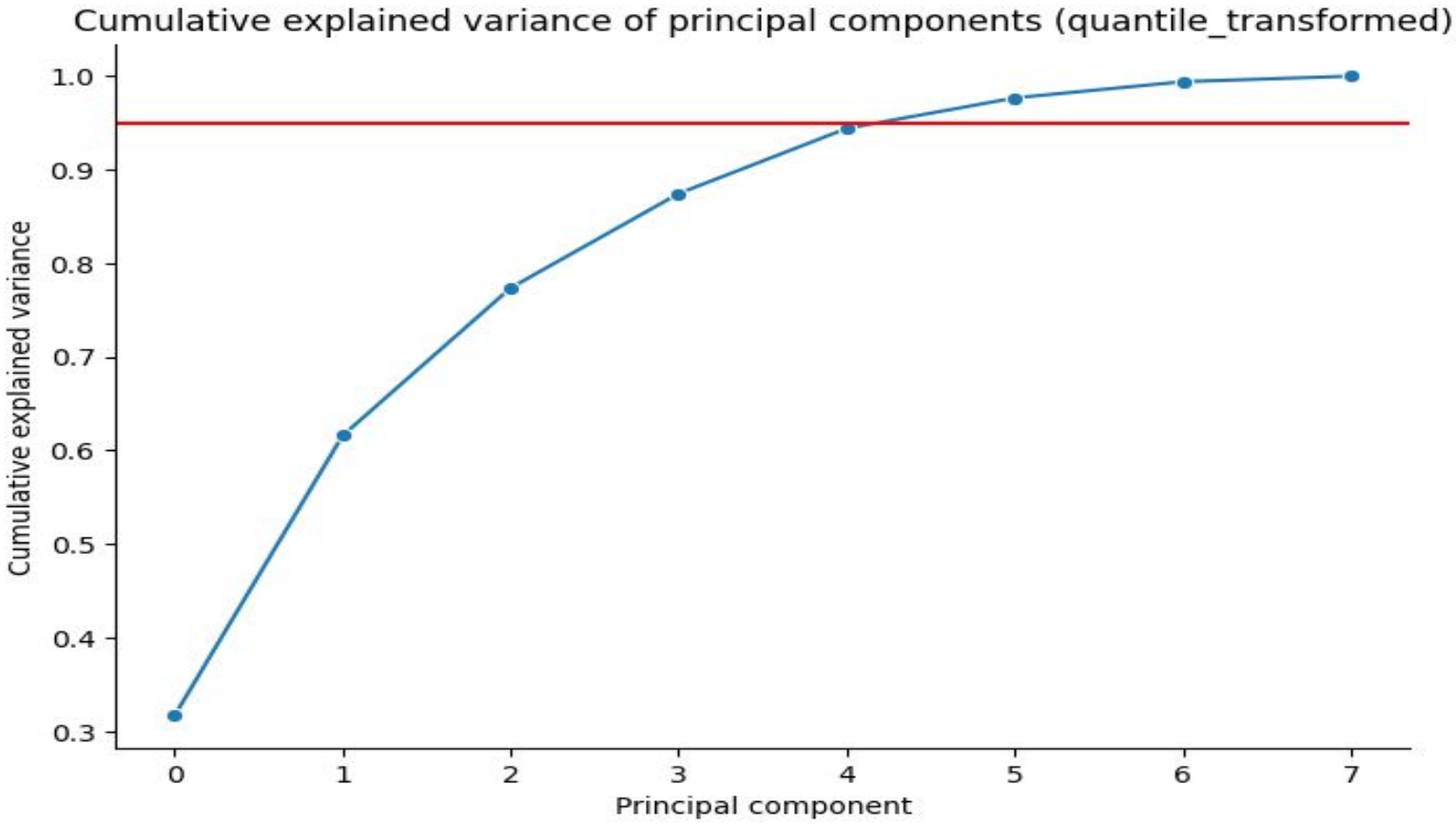
1. Danceability
2. Energy
3. Instrumentalness
4. Loudness
5. Mode (melodiness)
6. Speechiness
7. Tempo beats per minute
8. Valence (positiveness)

Heatmap: Song Similarity Analysis with Quantile transformed

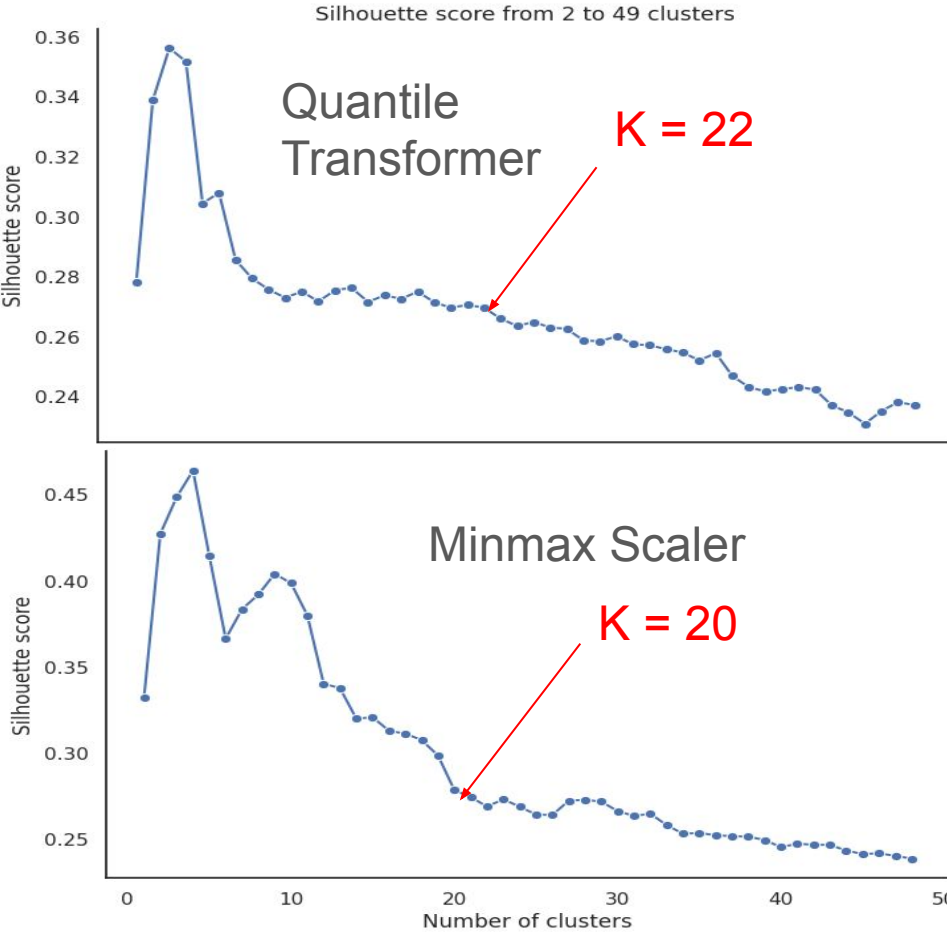


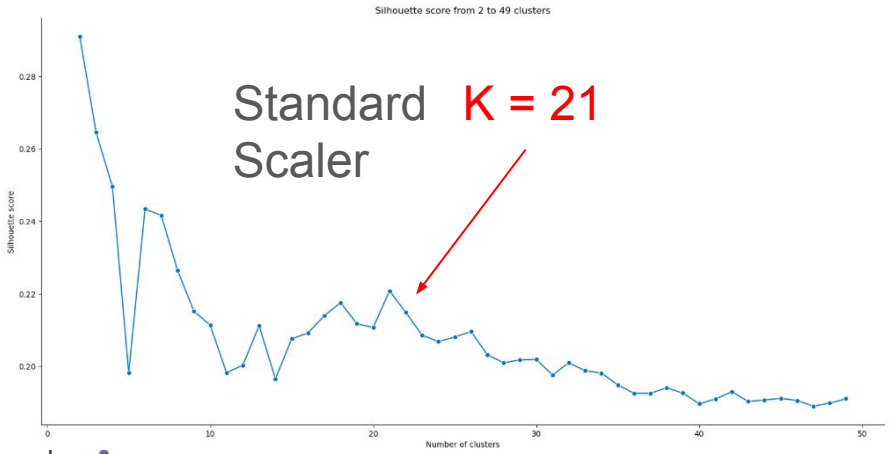


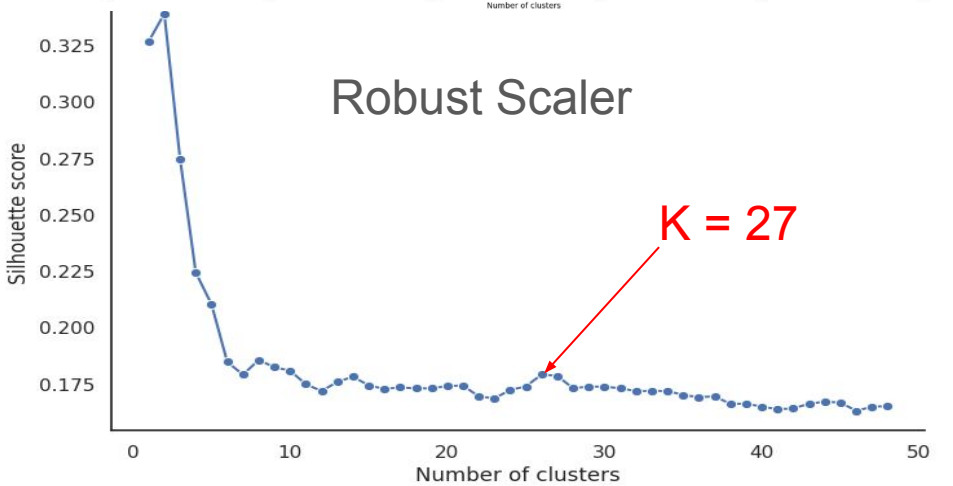
PCA to the Quantile transformed (loudness and tempo)



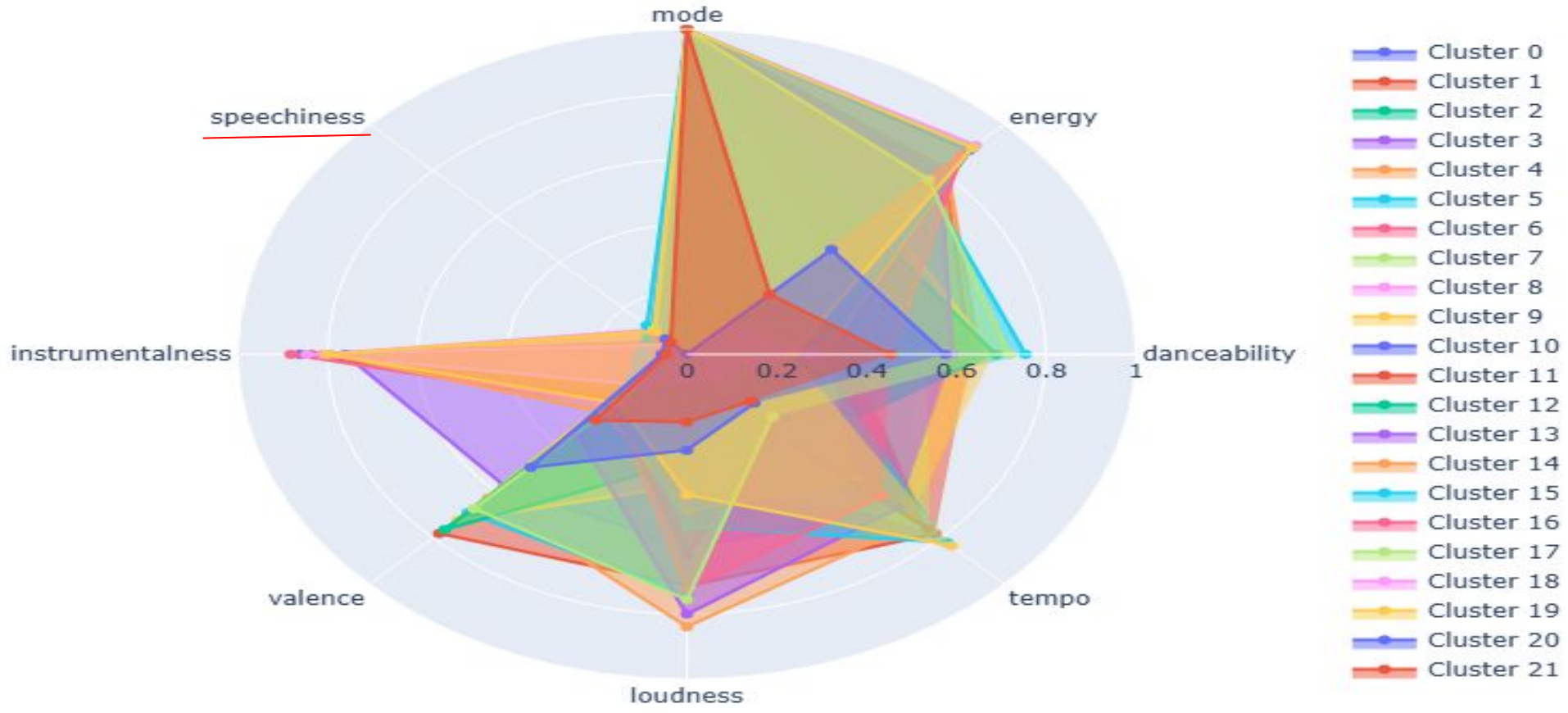
Silhouette score analysis with PCA, 95% variance

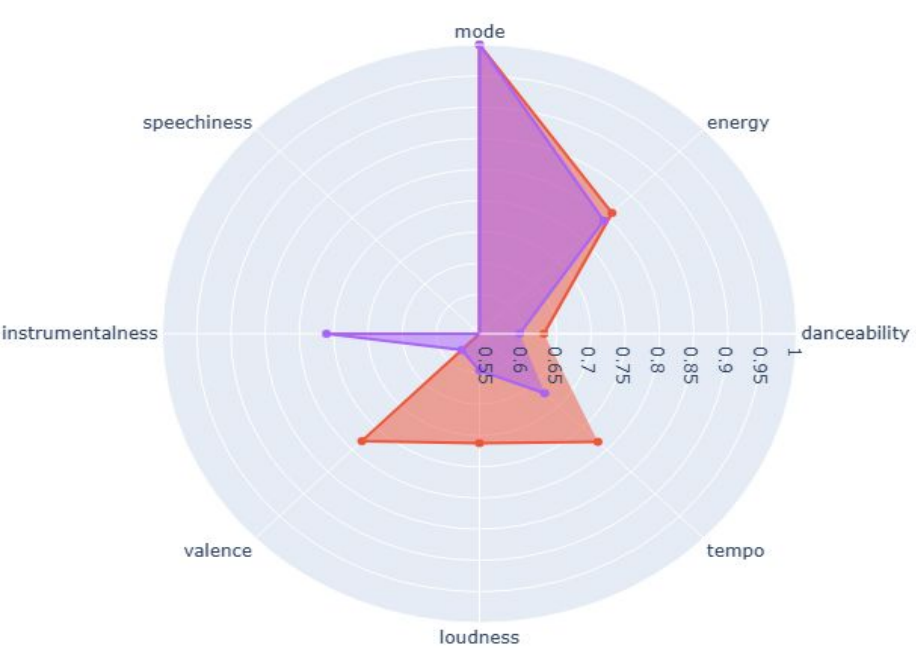


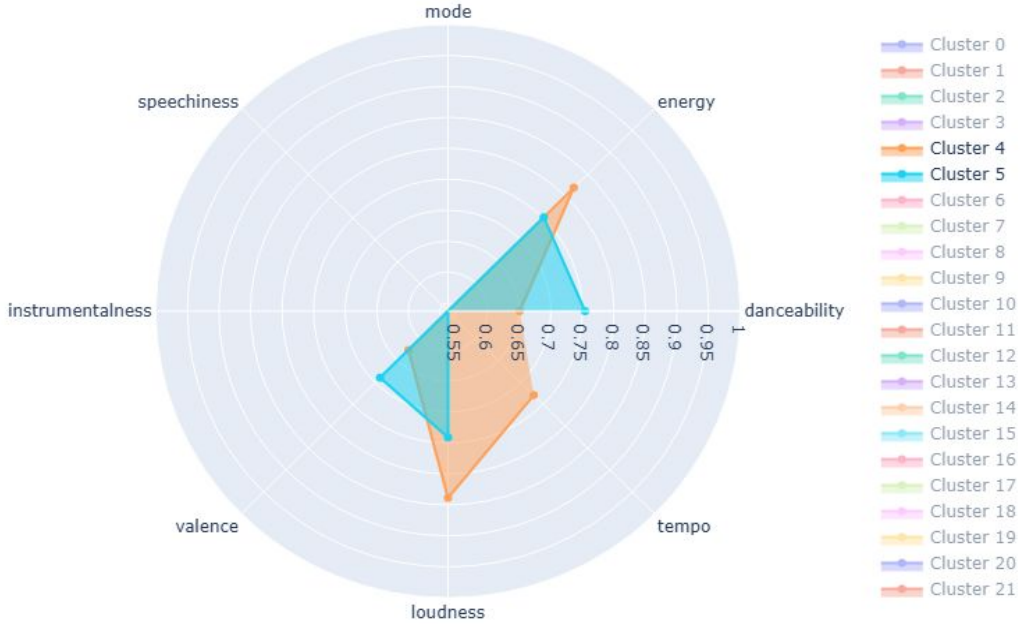




Radar chart for audio features by clusters using Quantile transform




mode
energy
danceability
tempo
loudness
valence
instrumentalness
speechiness
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9
Cluster 10
Cluster 11
Cluster 12
Cluster 13


mode
energy
danceability
tempo
loudness
valence
instrumentalness
speechiness
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9
Cluster 10
Cluster 11
Cluster 12
Cluster 13
Cluster 14
Cluster 15
Cluster 16
Cluster 17
Cluster 18
Cluster 19
Cluster 20
Cluster 21

Count of songs per cluster

Clusters	0	1	2	3	4	5	6	7	8	9	10
Total songs	215	381	88	93	277	301	106	290	297	178	250
Clusters	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Total songs	156	395	300	282	148	277	394	133	189	176	232

Observation and Conclusion

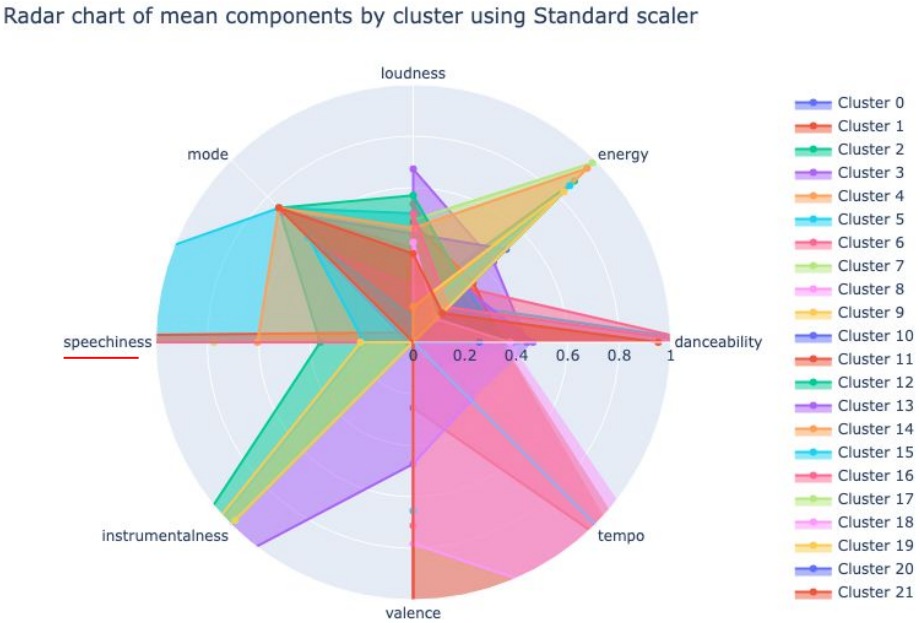
1. Not all the Playlists contain the same number of songs.
2. PCA may not be appropriate for small dimension data, as dimensionality reduction could loss the information.
3. PCA is good choice when we have large number of features as we will not lose much information.
4. K- means is the good choice when we have some idea about the data.
5. Analysis should be carried out through trial and error to achieve good results.

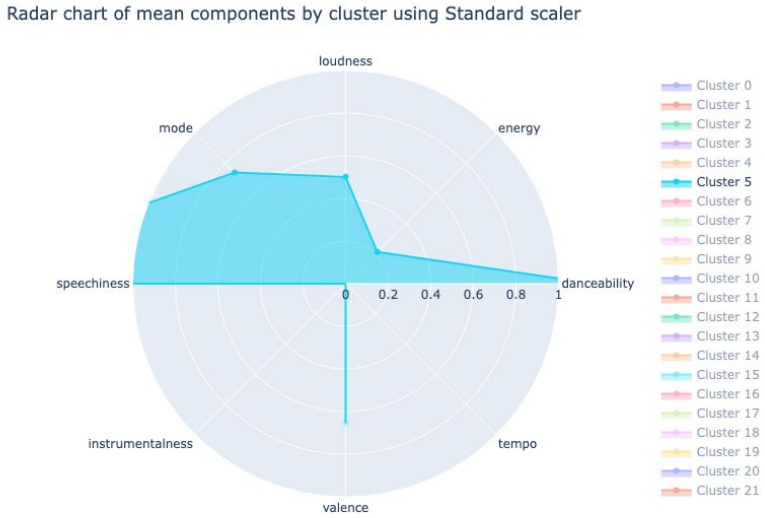
What would be your next steps if you continued with this project?

1. Collect User Feedback and Improve the Prototype
2. Implement Advanced Machine Learning Models
3. Test with a Larger User Base

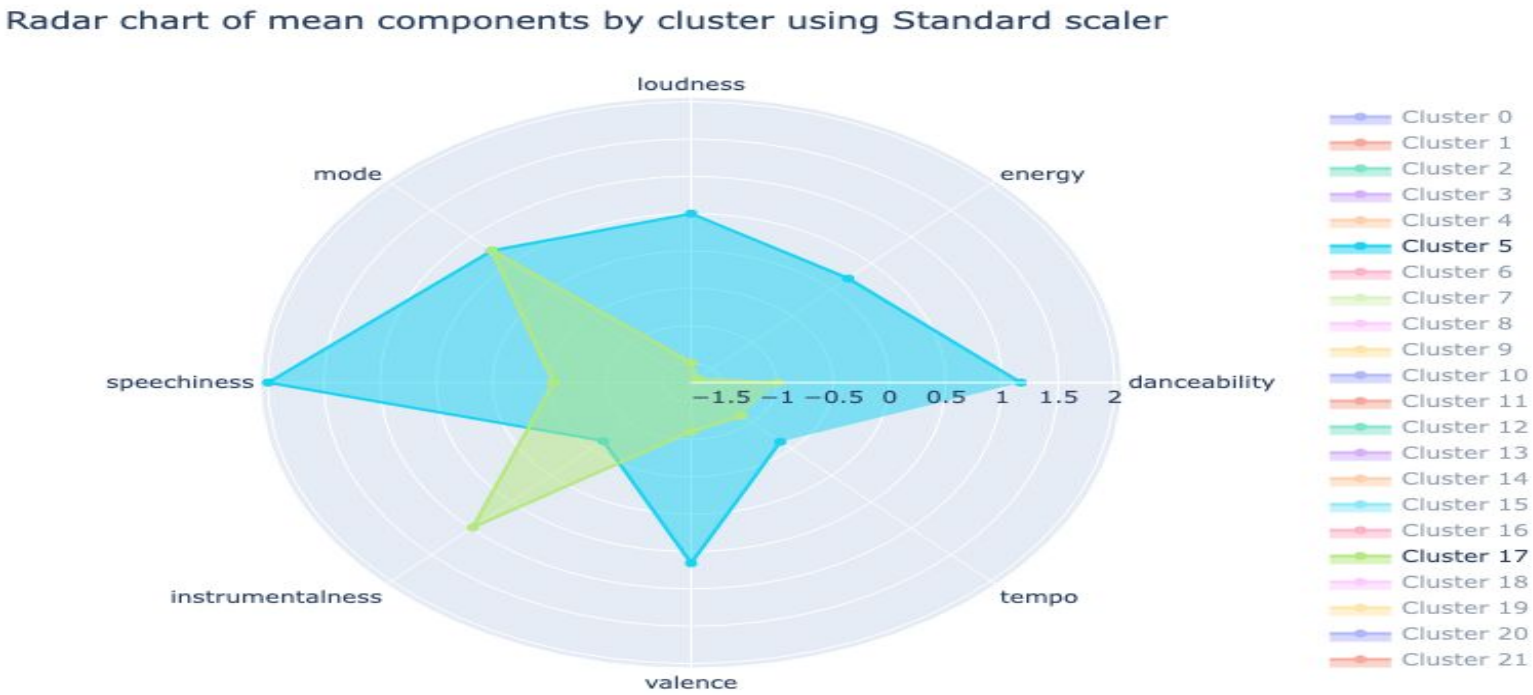
THANKYOU

Radar chart Using Standard Scaler





Radar chart Using Standard Scaler



Presentation Content:

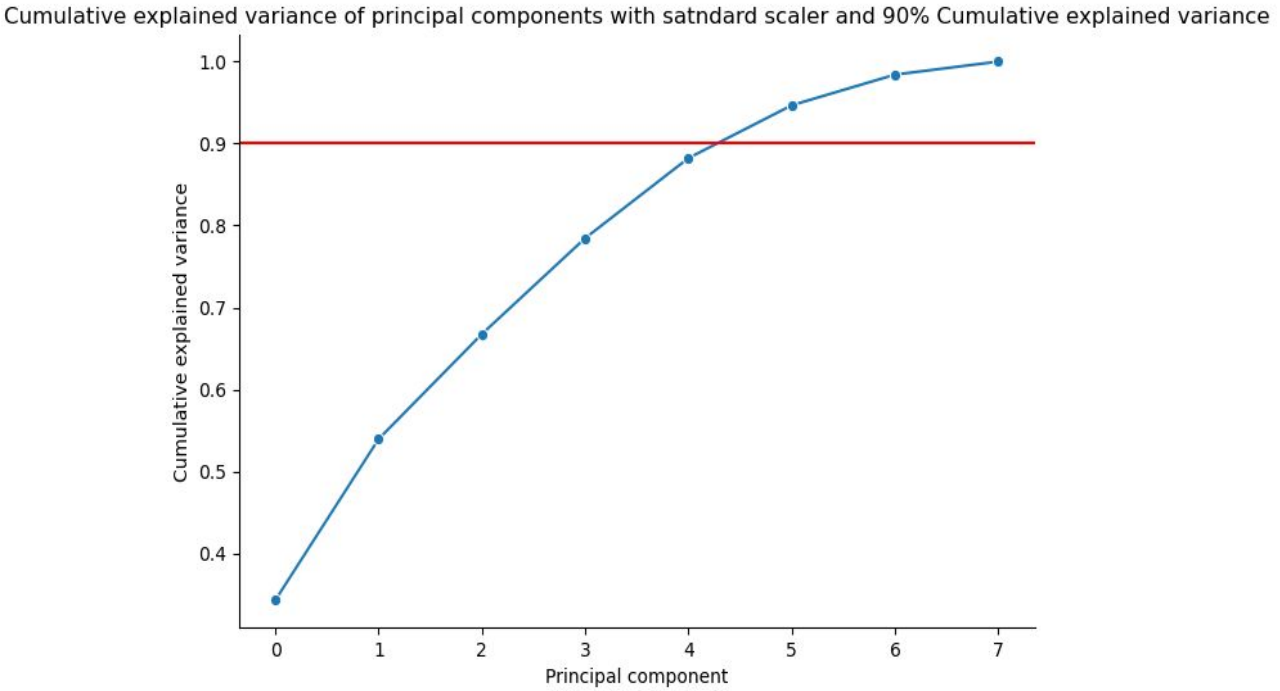
You are presenting your prototype and answering these core questions:

1. How did you create your prototype? (This does not mean showing off your code!)
    - How many playlists (clusters) are there? K = 22
    - What audio features did you use and what did you drop? Why? Slide = 9
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. Foram
5. What would be your next steps if you continued with this project? 1. Try with another method 2.Use machine learning models that can identify mood fit the emotional tone. 3 use social media insights and get data of feedback and behaviour

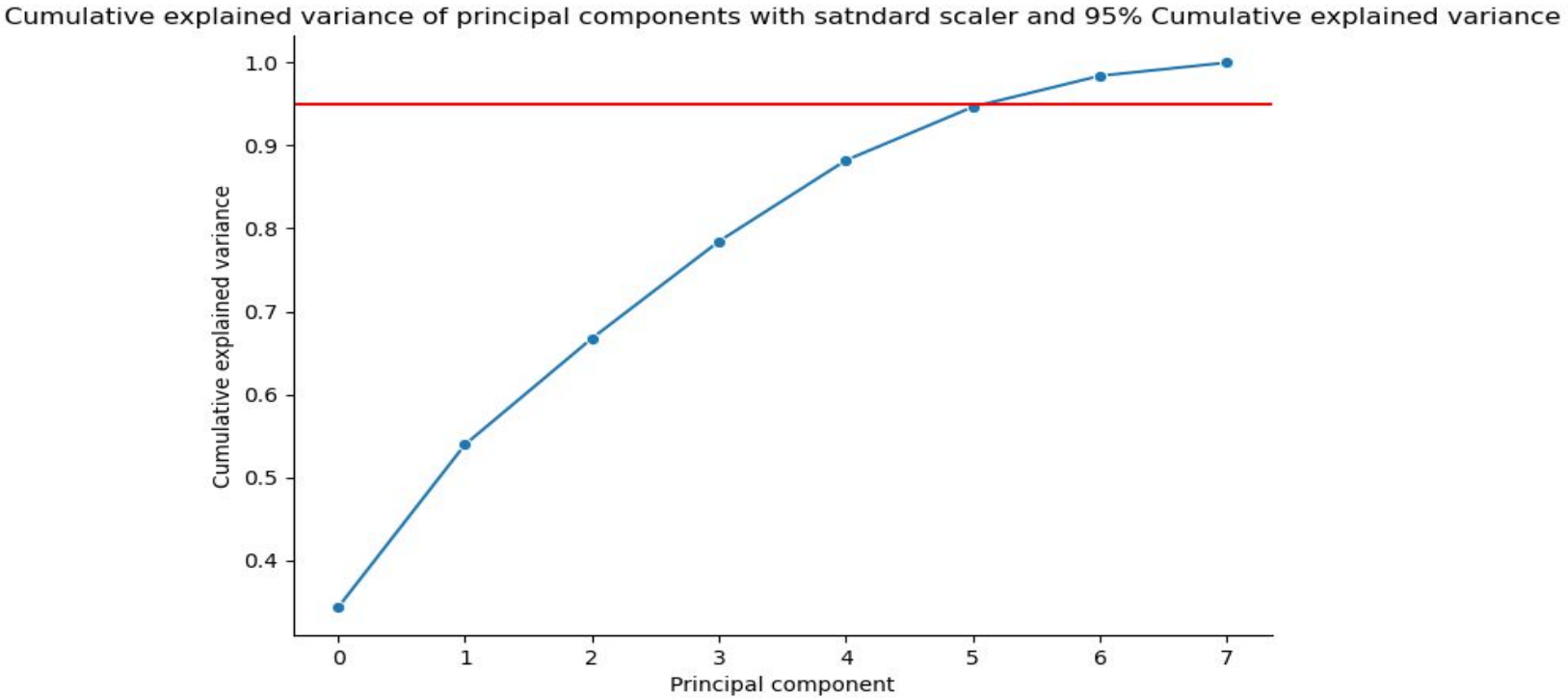
You will have 5 minutes to present, so you need not answer all these questions in detail.

PCA with standard scaler 90% cumulative explained variance

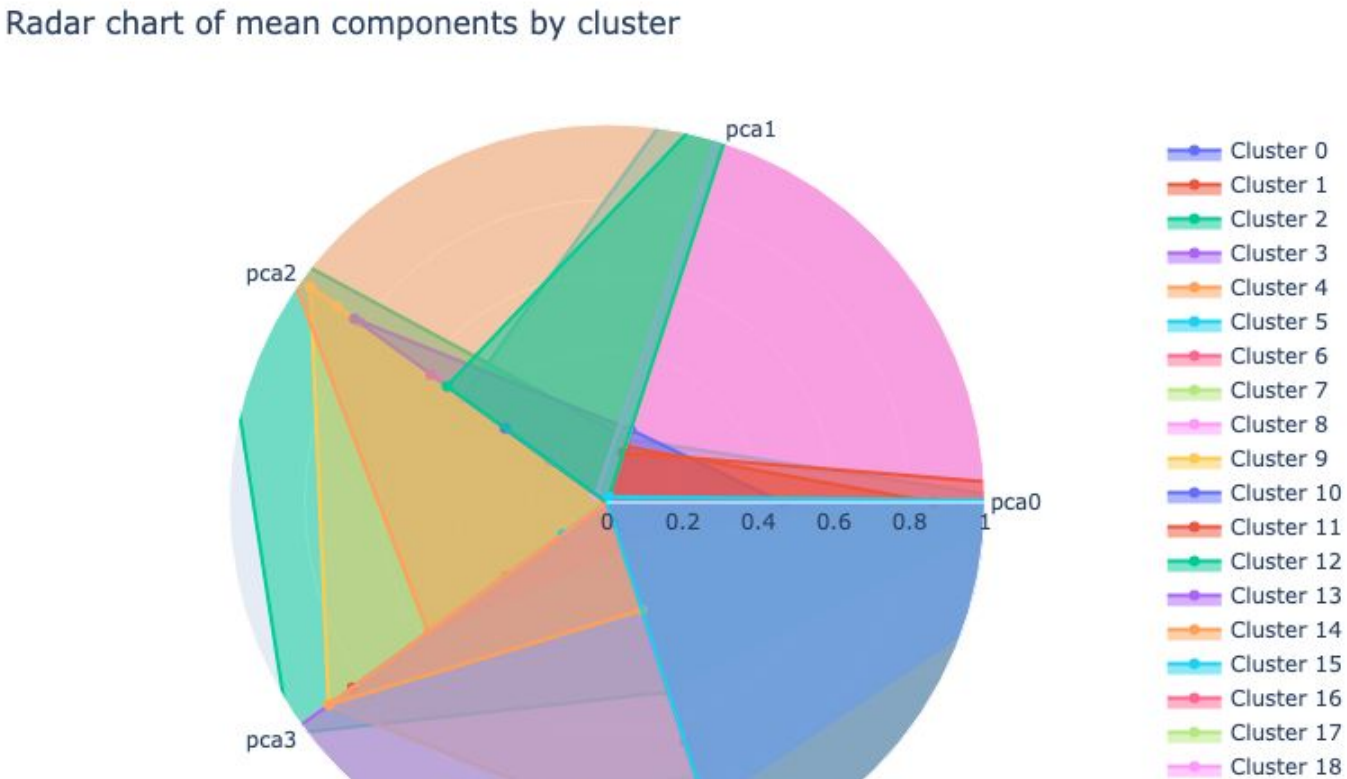
Components = 4

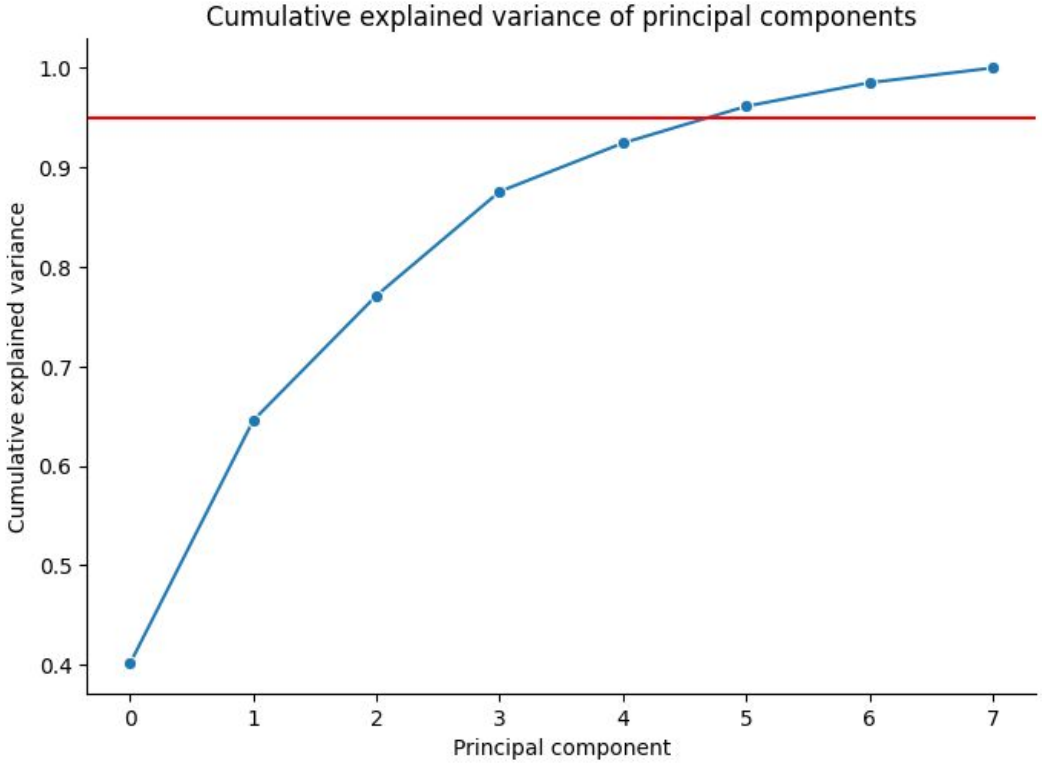


PCA with standard scaler 95% cumulative explained variance

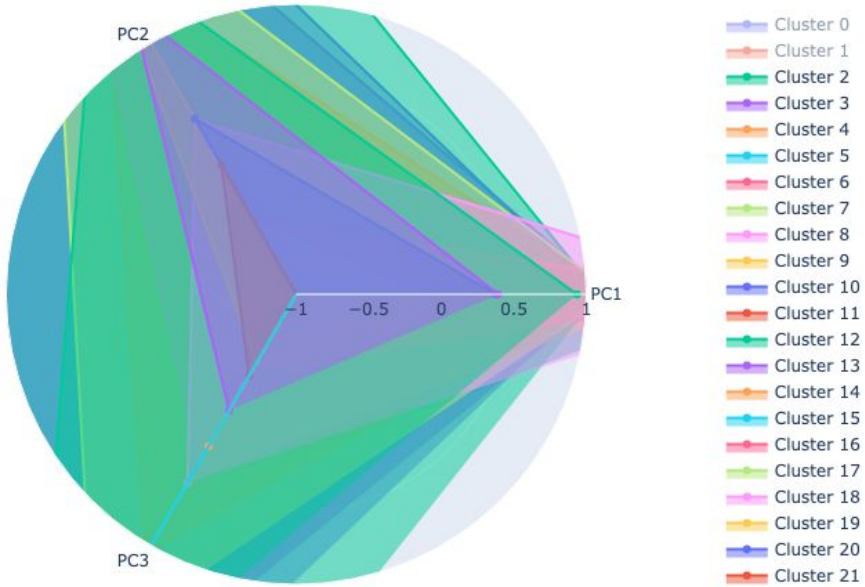


Radar for PCA with 80% components, clusters = 26 and random state 42

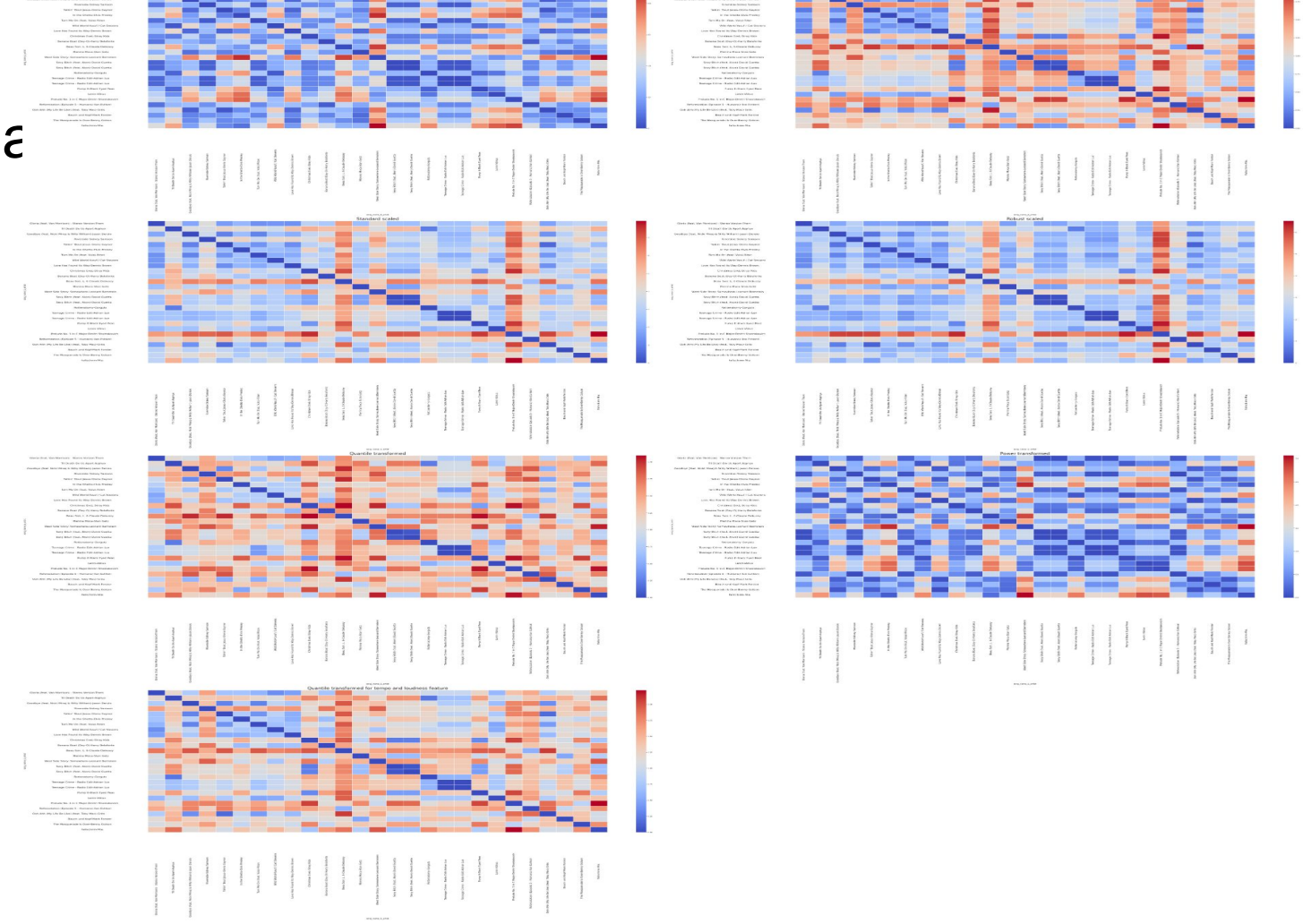



Cumulative explained variance of principal components
Cumulative explained variance
Principal component
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
1
2
3
4
5
6
7

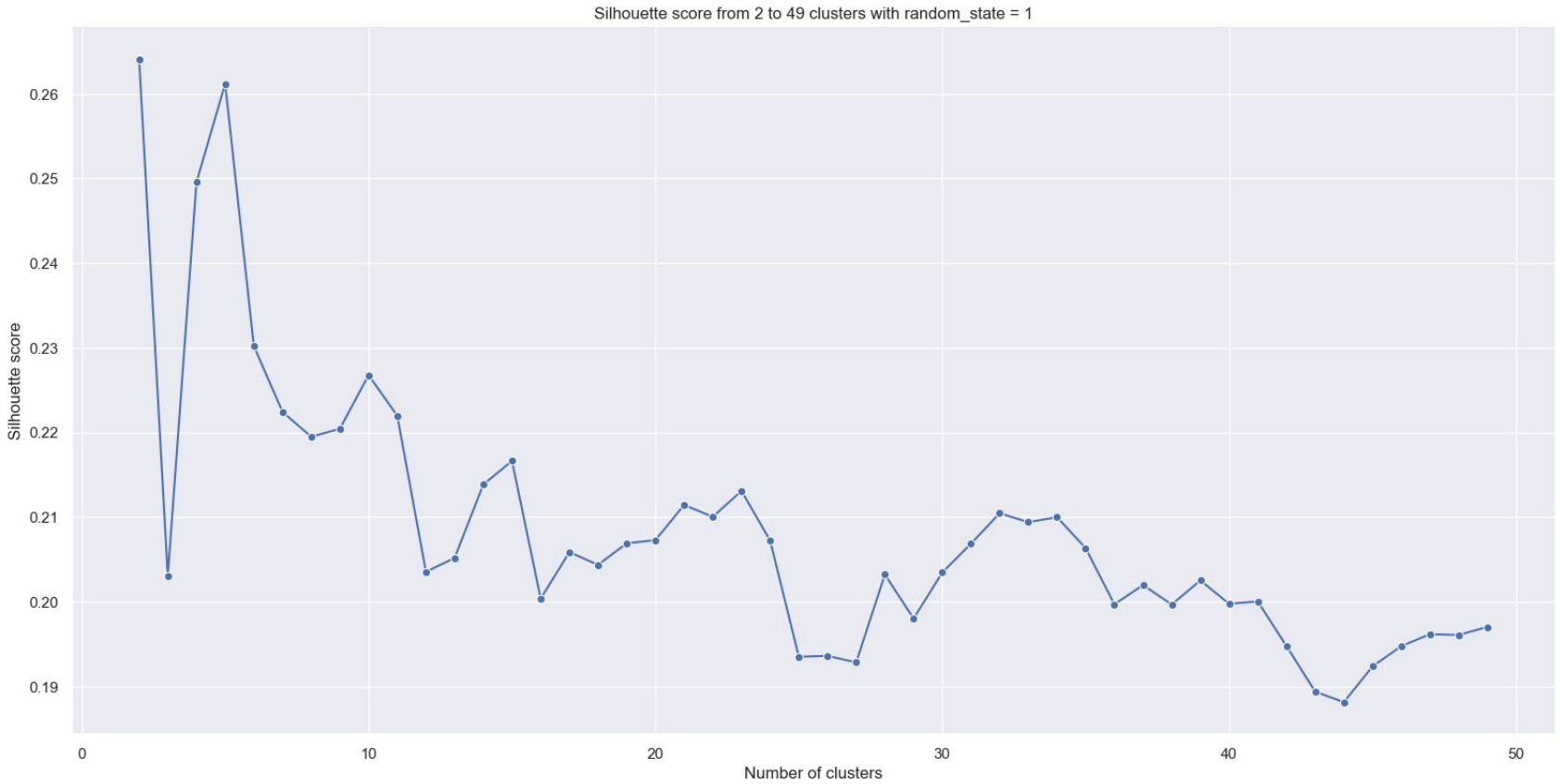
Radar Chart of PCA-transformed Clusters (RobustScaler)



Hea


Standard scaled
Robust scaled
Quantile transformed
Power transformed
Quantile transformed for tempo and loudness feature

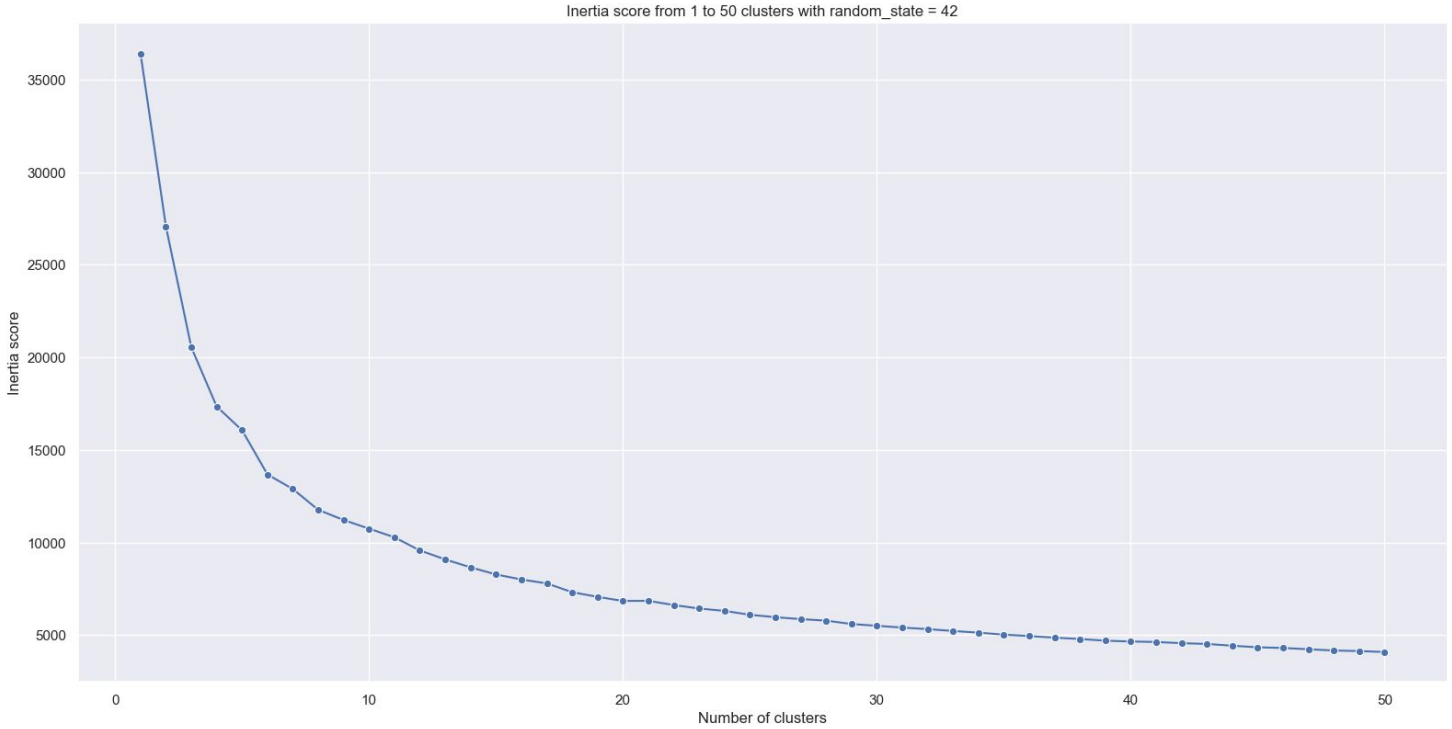
Silhouette score analysis with PCA, standard scaler, 95% variance



K means implementation

1. Import the KMeans class
2. `initialize the model specifying number of clusters (You have to change it depending on analysis and expectation)`
3. `fit the model to the data. :` performs the clustering process. This calculates the centroids (cluster centres) and determines which data point goes in which cluster.
4. `obtain the cluster output:` Obtain cluster labels for each data point. This tells you which cluster each data point belongs to based on the learned centroids.
5. Analyse the results.:
6. `attach the cluster output to our original DataFrame`
7. explore how your data is grouped and visualise the clusters using plotting techniques.

Inertia score analysis with PCA, standard scaler, 80% variance



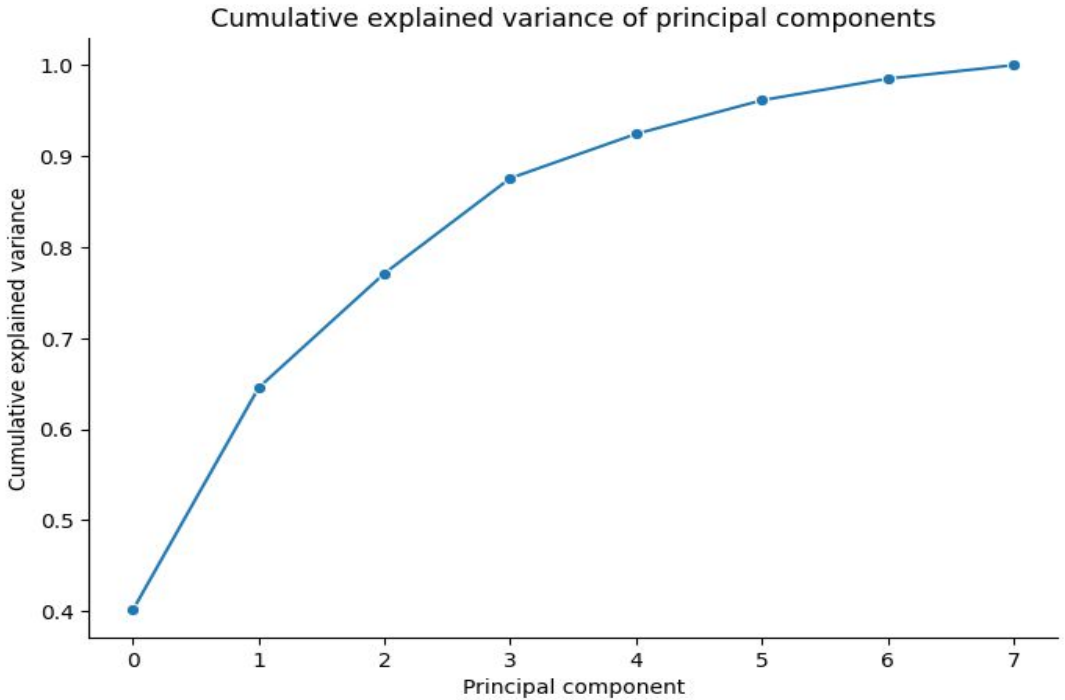
Analysing k-means: estimating the numbers of clusters

The number of clusters (k) is an important parameter for KMeans clustering. If k is too small, the clusters will be too large and will not accurately capture the underlying structure of the data. If k is too large, the clusters will be too small and will not be meaningful.

The inertia elbow method and the silhouette score are two popular methods for estimating the optimal number of clusters for KMeans.

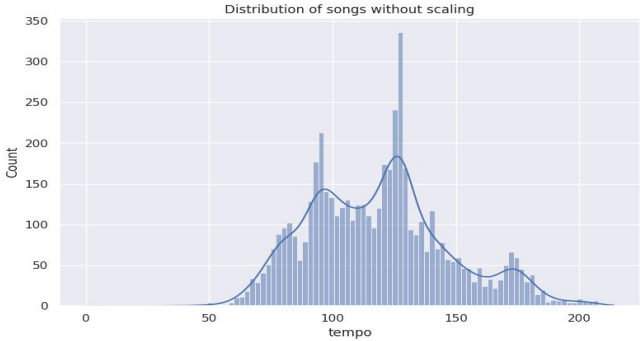
1. How many data points do you have? (around 5000 songs)
2. How many would be in a cluster if you chose this number of clusters?
3. What is the point of making these clusters? Is there a logical number of clusters, where if you make more clusters they just become too small for your business case?

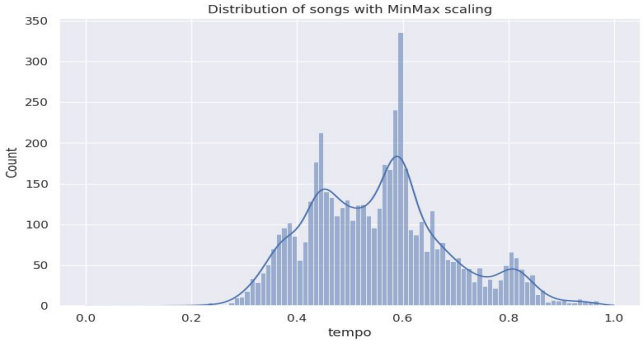
Foram

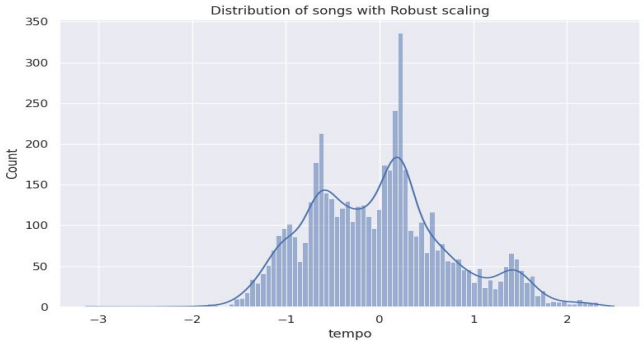

Cumulative explained variance of principal components
Cumulative explained variance
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
1
2
3
4
5
6
7
Principal component

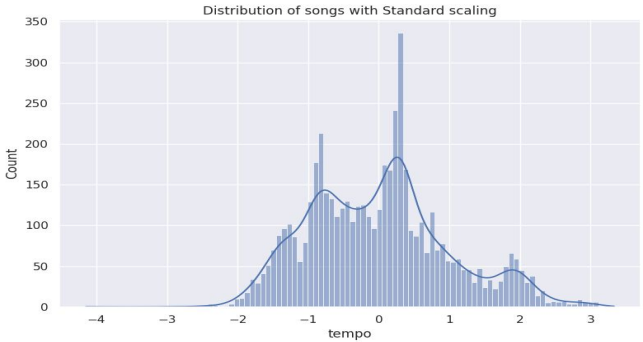
All hist

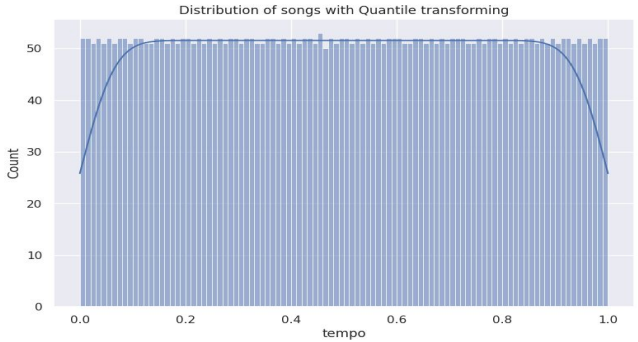
Bin = 100

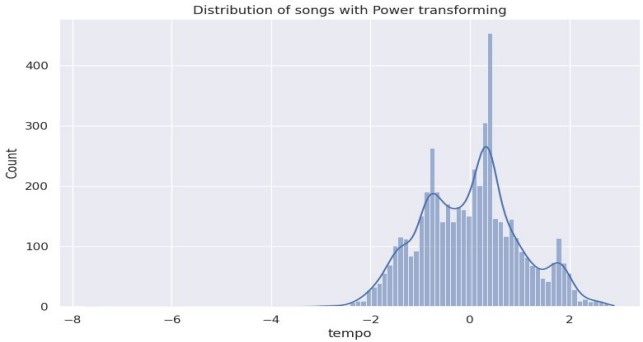




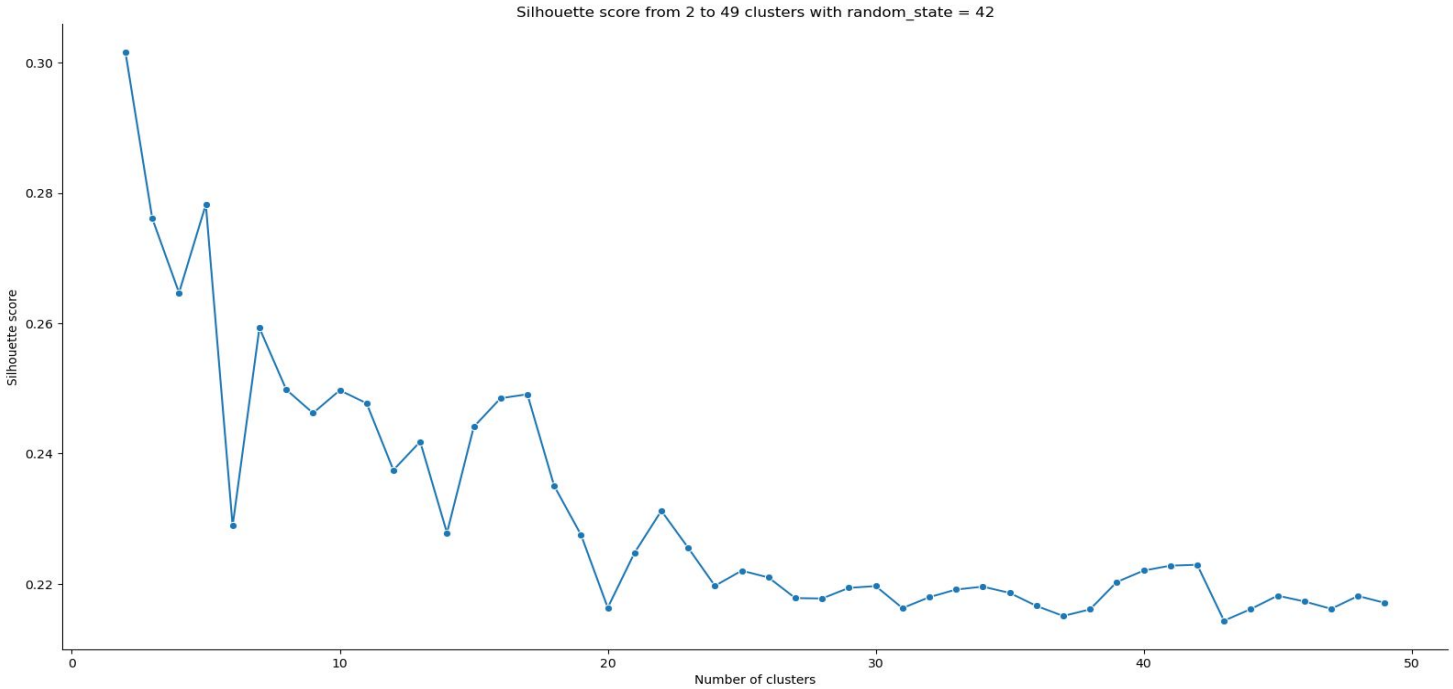




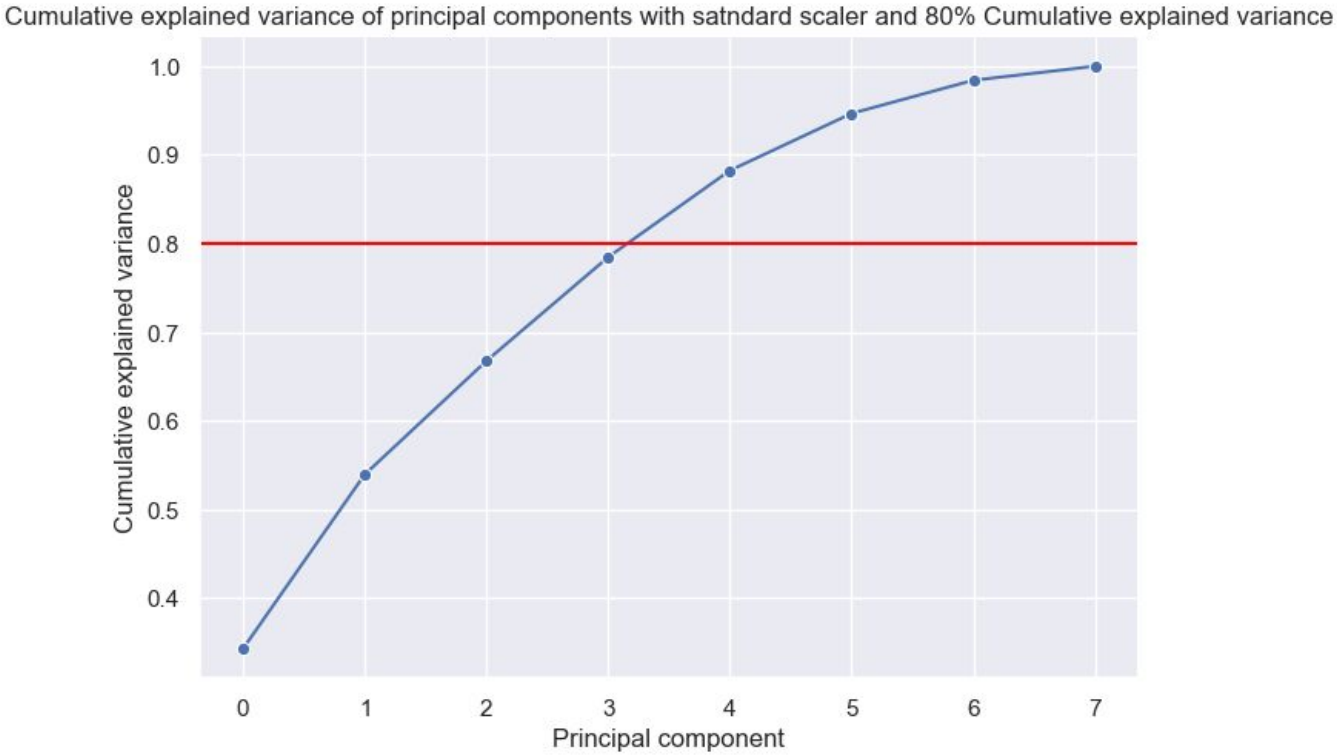




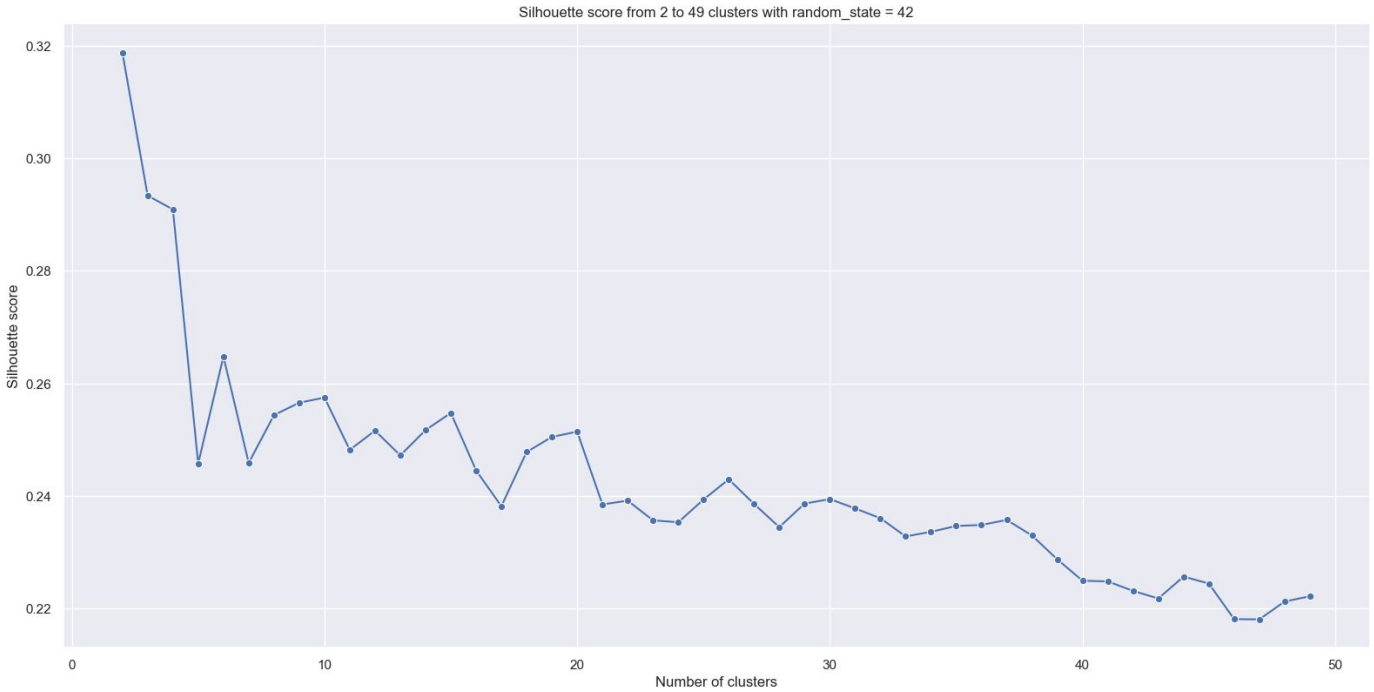
Silhouette score analysis with PCA, standard scaler, 90% variance



PCA with standard scaler 80% cumulative explained variance

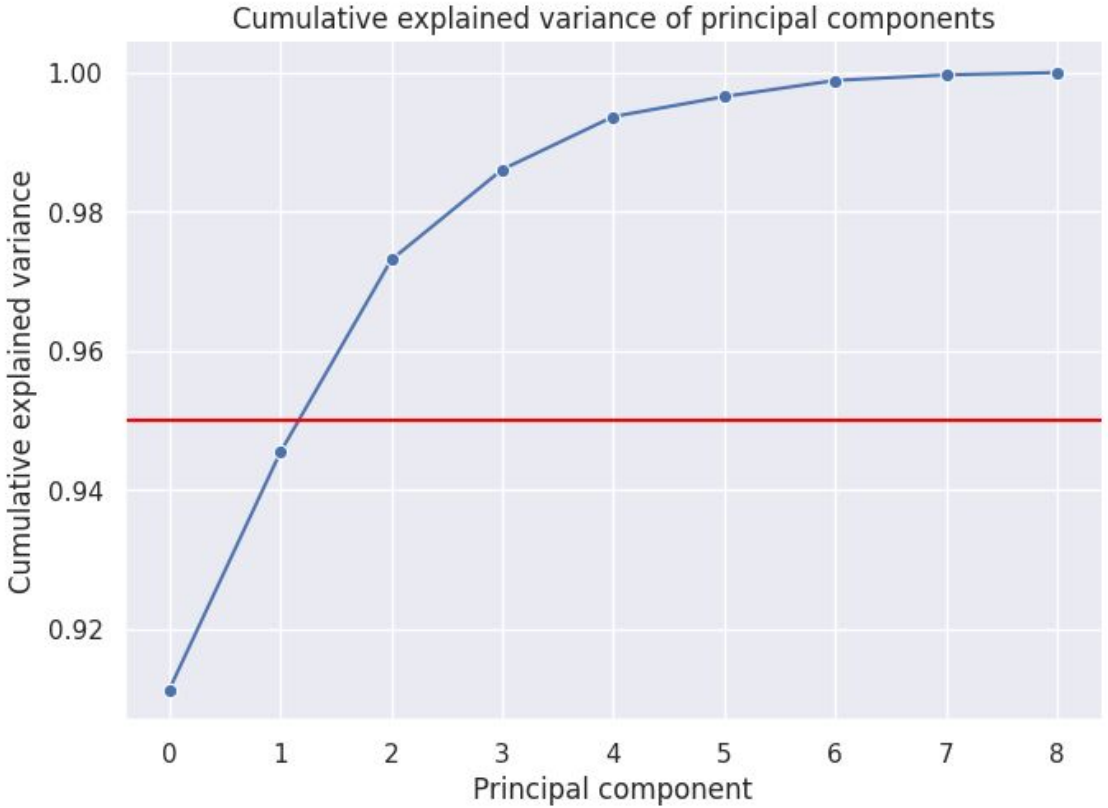


Silhouette score analysis with PCA, standard scaler, 80% variance



PCA and MinMax Scaler

Components = 2



Principal Component Analysis (PCA)

PCA is an unsupervised technique that processes unlabelled data to restructure the columns of a dataframe. The primary use of PCA is dimensionality reduction, which reduces the number of columns without losing significant information. This streamlining of data has a notable secondary effect: it reduces noise. Noise in data refers to random variations or errors that obscure underlying patterns or signals, making it harder to extract meaningful information.

In this project, we don't have many columns, so dimensionality reduction isn't a major concern. However, this technique will be particularly useful in our next project, supervised machine learning. For now, you'll find that the noise reduction achieved through PCA helps to create better clusters, demonstrating why PCA is such a valuable unsupervised learning tool.

PCA - variance (higher the better)

1. Elbow method - (decreasing trend - look at sudden change)
2. Cummulative variance - (increasing trend - we can consider upto 90 - 95%)

What is K- means

K-means clustering is an **unsupervised machine learning algorithm** that **groups** unlabeled **data** into a **predefined number of clusters**. It is one of the most popular clustering algorithms due to its simplicity and effectiveness.

Case study: Moosic

- **Dataset** (few thousand songs) collected from **Spotify API** contains audio features
- use **clustering** algorithm such as K-Means to divide the dataset into a few clusters (which will become playlists)
- **Are Spotify's audio features able to identify "similar songs", as defined by humanly detectable criteria?**
- **Is K-Means a good method to create playlists?**

Perspective 1: Skeptical View

Some members of the team doubt the ability of audio features to effectively capture the actual "mood" of a song. They argue:

- **Subjectivity**: Determining the mood of a song is inherently subjective and varies from person to person.
- **Human Judgment**: They believe that only human evaluation can truly understand and judge the emotional tone of music accurately.

Perspective 2: Optimistic View

Other team members are hopeful about leveraging Data Science to address the challenge. They believe:

- **Beyond Subjectivity**: A well-designed Data Science solution might uncover patterns and connections that go beyond human intuition.
- **Unexpected Connections**: Such an approach could reveal relationships between songs that may not be obvious initially but could provide fresh and meaningful insights.

Moosic

Feature Selection

Scaling

Training Models

`algorithm.fit(df)`

Cluster Analysis

Prediction

`model.predict(df)`

About k means….

- Explain the K-Means algorithm as if I were a 5 year old.
- **When choosing the right number of clusters, how should I balance mathematical criteria and business considerations?**
- Can K-Means handle different shapes or sizes of clusters well, or does it have limitations?
- Should I really spend time learning about K-Means, considering we’re all eventually going to die?

Scikit-Learn provides a powerful implementation of K-Means clustering through the `KMeans` class from the `cluster` submodule.