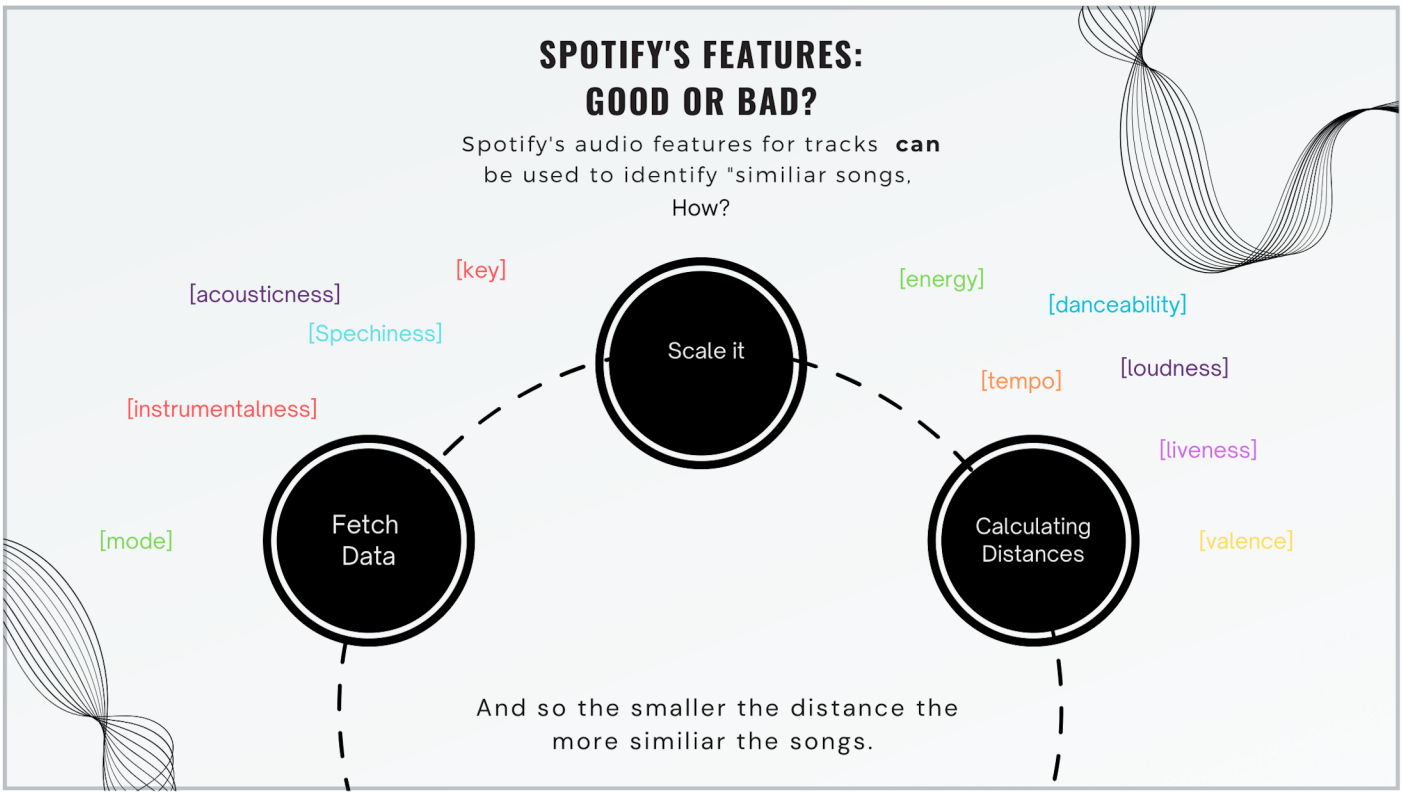
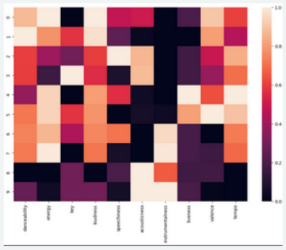
MOOSIC
PROJECT

On the First Goal: Talk about features being caracteristics of music that we are going to compare and see if the songs are trully alike when we categorize acording them

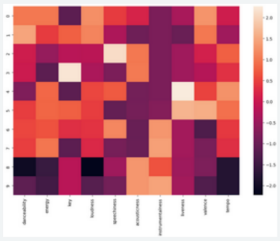
the company wants us to know if K-Means algorithm as method to group data acording to its audio features to build playlist.



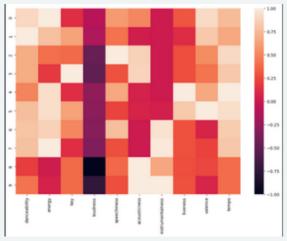
['danceability', 'energy', 'valence', 'tempo', 'acousticness', 'key', 'loudness', 'instrumentalness', 'speechiness']



MinMax scaler



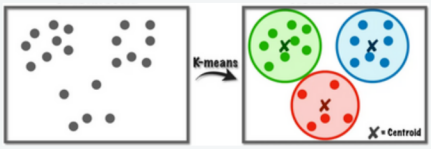
StandardScaler()



AbsoluteScaler()

K-MEANS

It can be a good method to create playlists.
As an algorithm that clasifies items into groups or "clusters" based on their features



Each cluster would then represent a playlist of musically similar songs, so here is what we tried.

Scaling Data

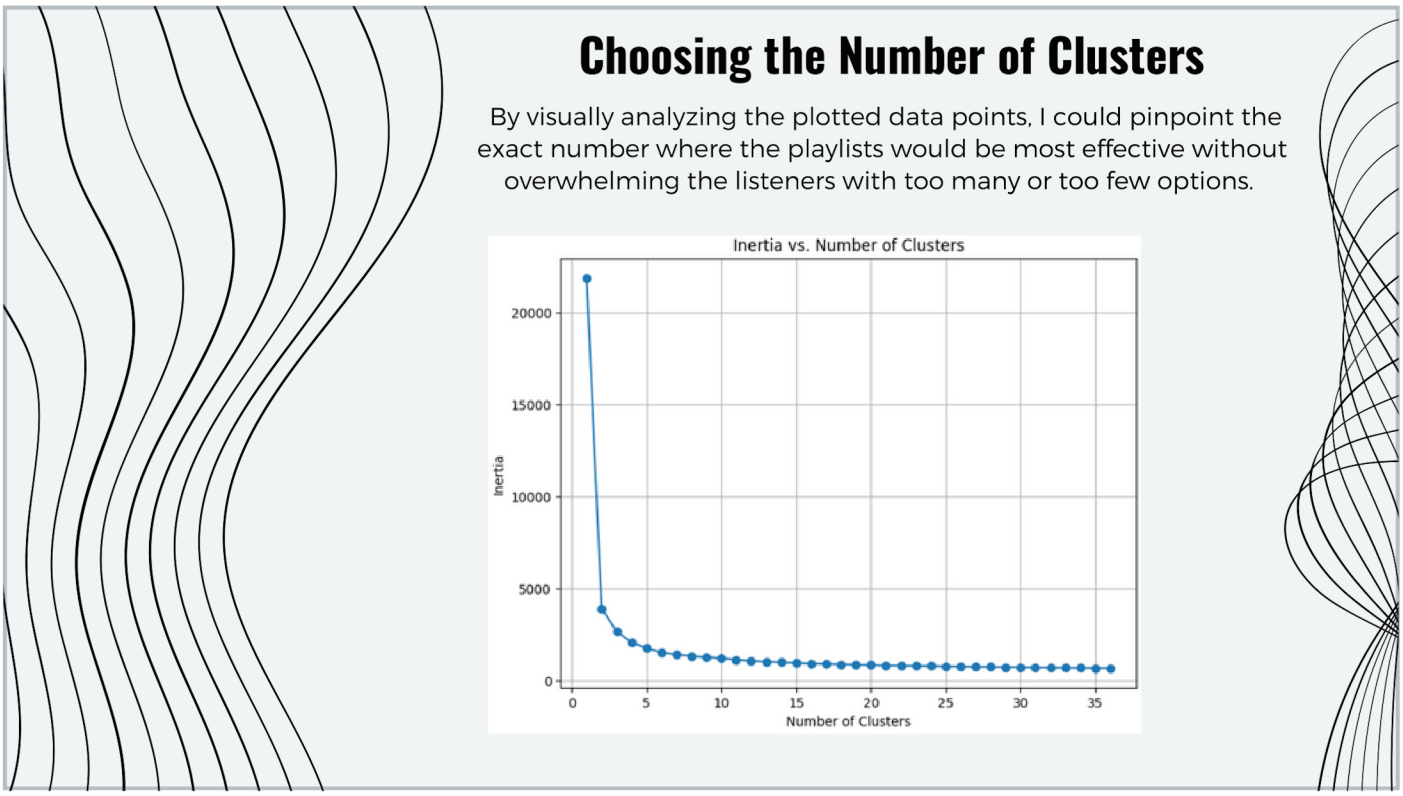
STEP 1

Applying K-Means

STEP 2

Playlist Evaluation

STEP 3



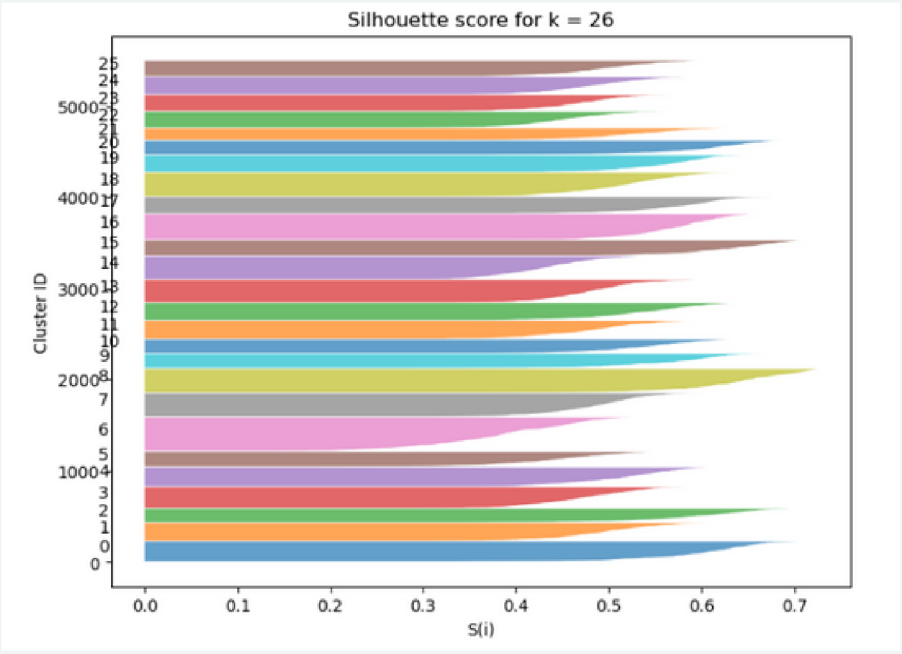
In the image, the WCSS (Within-Cluster Sum of Square) values are shown for cluster sizes ranging from 1 to 35. The WCSS(within-cluster sum of squares) values are shown as a line plot, with the x-axis representing the number of clusters and the y-axis representing the WCSS value.

The image shows that the WCSS values decrease as the number of clusters increases. This is because the within-cluster sum of squares is a measure of the similarity of the points within a cluster. As the number of clusters increases, the points within each cluster become more similar, which leads to a lower WCSS value.

the KNN plot is a graph of the distances to the kth nearest neighbor for each point in the dataset. The knee point in the plot is the point where the slope of the line changes significantly. This point is often used to determine the optimal value for the KNN parameter k.

In the code you provided, the KNN parameter k is set to 2. This means that the distance to the second nearest neighbor is plotted for each point in the dataset. The knee point in the plot is at a distance of approximately 23.

Checking the Clusters



A silhouette score of 0.19522482691662454 is a relatively low score. A silhouette score of 0 indicates that the sample is not well-clustered, while a score of 1 indicates that the sample is perfectly clustered. A score of 0.5 indicates that the sample is equally well-clustered in both of its nearest clusters.

Asim: with quantile scaler
The Silhouette Score is: 0.5163
for k=27

* For k = 26 the average silhouette score is: 0.5119
  - For cluster 0, the silhouette value is: 0.6
  - For cluster 1, the silhouette value is: 0.49
  - For cluster 2, the silhouette value is: 0.59
  - For cluster 3, the silhouette value is: 0.47
  - For cluster 4, the silhouette value is: 0.51
  - For cluster 5, the silhouette value is: 0.44
  - For cluster 6, the silhouette value is: 0.38
  - For cluster 7, the silhouette value is: 0.48
  - For cluster 8, the silhouette value is: 0.64
  - For cluster 9, the silhouette value is: 0.56
  - For cluster 10, the silhouette value is: 0.54
  - For cluster 11, the silhouette value is: 0.48
  - For cluster 12, the silhouette value is: 0.53

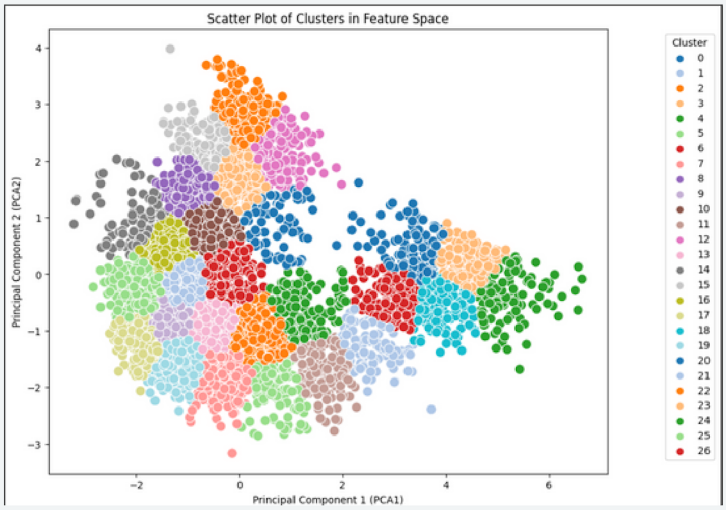
- For cluster 13, the silhouette value is: 0.48
- For cluster 14, the silhouette value is: 0.43
- For cluster 15, the silhouette value is: 0.63
- For cluster 16, the silhouette value is: 0.57
- For cluster 17, the silhouette value is: 0.57
- For cluster 18, the silhouette value is: 0.51
- For cluster 19, the silhouette value is: 0.56
- For cluster 20, the silhouette value is: 0.6
- For cluster 21, the silhouette value is: 0.52
- For cluster 22, the silhouette value is: 0.45
- For cluster 23, the silhouette value is: 0.46
- For cluster 24, the silhouette value is: 0.48
- For cluster 25, the silhouette value is: 0.49

The DBI is a metric that is used to evaluate the quality of clustering results. A lower DBI score indicates better clustering results.

The code you provided calculates the DBI score for cluster sizes ranging from 2 to 100. The DBI score decreases as clusters increase, reaching a minimum value of 0.19 for k=20. This suggests that the optimal number of clusters for this dataset is 20.

Running K-Means:

With the data prepared and the number of clusters defined,we apply the K-Means to group the songs into playlists based on their audio feature similarities. The algorithm iteratively assigns each song to the nearest cluster centroid based on the distance metric.



Dimensionality Reduction: Music features extracted from Spotify songs can often be high-dimensional, making it challenging to visualize and analyze the data. PCA helps reduce the number of dimensions while retaining as much variance as possible. By projecting the data onto a lower-dimensional space, it becomes easier to interpret and work with.

Faster Computation: With reduced dimensions, the computational complexity of the k-means algorithm decreases, leading to faster clustering, especially for large datasets.

Visual Representation: PCA allows you to visualize the clusters on a 2D or 3D plot, making it easier to understand the relationships between different songs and their cluster memberships.

Enhanced Interpretability: By reducing the features to the most important components, PCA can improve the interpretability of the clustering results. It can highlight the dominant musical characteristics that define each cluster.

However, it's essential to note that PCA is not always necessary, and its effectiveness depends on the nature of the data and the specific goals of the analysis. In some cases, you may achieve satisfactory clustering results without using PCA, especially if the data is already well-structured and the dimensionality is not excessively high. The decision to use PCA should be based on a careful analysis of the data and the objectives of the clustering task.

Playlist Evaluation

After running the K mean algorythm we checked the clusters and listend to its selection in order to appreciate it better and try to name it according it to the vibe it provide and what we were able to recognize.

Se Eu Quiser Falar Com Deus ...	Gilberto Gil	0.658	0.259	11	-13.141	0.0705	0.694	0.000059	0.3060	110.376	Latin Serenade: Soulful Ballads and Melodic Em...
Rosa Morena ...	Kurt Elling	0.651	0.119	6	-19.807	0.0380	0.916	0.000343	0.4020	120.941	Latin Serenade: Soulful Ballads and Melodic Em...
Madalena ...	Maria Gasolina	0.675	0.207	11	-13.820	0.0545	0.963	0.414000	0.6840	93.531	Latin Serenade: Soulful Ballads and Melodic Em...
The Girl From Ipanema ...	Stan Getz	0.641	0.140	8	-16.790	0.0390	0.867	0.001660	0.3880	129.318	Latin Serenade: Soulful Ballads and Melodic Em...
Rebel Rebel ...	Seu Jorge	0.775	0.206	6	-15.659	0.0530	0.879	0.000313	0.5530	120.460	Latin Serenade: Soulful Ballads and Melodic Em...

This cluster had inside ballads from diffrent style of latin music so whe name it :
Latin Serenade: Soulful Ballads and Melodic Embrace.

Aqua De Coco ...	Marcos Valle	0.755	0.732	10	-9.600	0.0658	0.487	0.000910	0.816	88.028	Tropical Grooves: Vibes of Sunny Paradise
Mas Que Nada ...	Rio Combo	0.618	0.539	5	-12.689	0.0312	0.156	0.002140	0.916	88.646	Tropical Grooves: Vibes of Sunny Paradise
Os Grilos ...	Marcos Valle	0.573	0.805	9	-11.327	0.0746	0.373	0.489000	0.694	100.029	Tropical Grooves: Vibes of Sunny Paradise
Orange Afternoon ...	Bebe	0.615	0.630	10	-10.809	0.0448	0.430	0.244000	0.557	91.122	Tropical Grooves: Vibes of Sunny Paradise
Agua De Beber ...	Sophie Milman	0.660	0.759	7	-8.550	0.0437	0.243	0.002270	0.700	94.477	Tropical Grooves: Vibes of Sunny Paradise

This cluster had inside more latin music but more cheerfull and faster rythms therefore the name:
Tropical Grooves: Vibes of Sunny Paradise

PROJECT CONCLUTIONS

Are Spotify's audio features able to identify "similar songs", as defined by humanly detectable criteria?

Yes, definitely. Spotify's audio features help our algorithm to identify noticeable similarities in songs.

Is K-Means a good method to create playlists?

Its good to sort music according to similarities and it can create playlists, but others methods might be more effective

elbow methods

DBSCAN

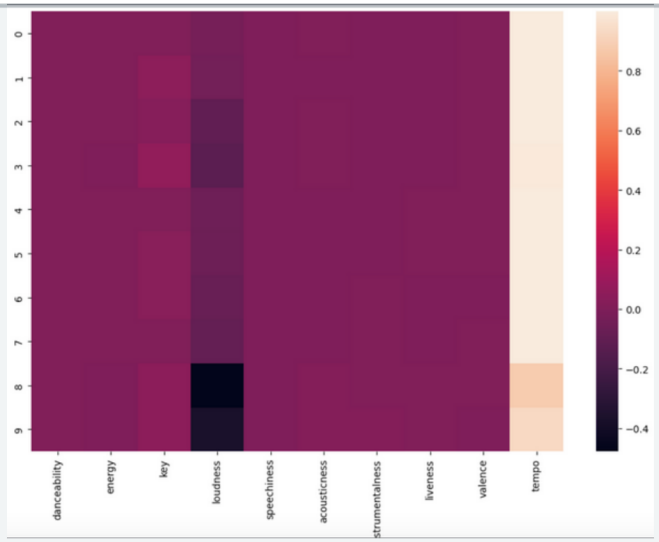
hierarchical_clusters

Davies-Bouldin Index

The silhouette score

The silhouette score is a metric that is used to evaluate the quality of clustering results. A higher silhouette score indicates better clustering results.

The Elbow Method is a heuristic method that is used to identify the optimal number of clusters by looking for the point where the within-cluster sum of squares(WCSS) starts to decrease



Unit Vector Scaling