Spotify Music Recommendation System

Yixin Chen, Shunqi Lei, Xinyue Cao

January 26, 2023

1 Abstract

Nowadays, many entertainment platforms are using machine learning to customize the recommendations for their users based on user history of choice of entertaining items like songs and movies. In this experiment, we build up four ML models: K-means, Gaussian Mixture Model, PCA plus K-means and PCA plus Gaussian Mixture Model to predict the possible songs users might like by randomly selecting songs in the same cluster as the song user previously picked. Our relatively best model is PCA plus either of K-means and GMM with a silhouette score of 0.4.

2 Introduction

This project is an experiment which aims to build a music recommendation system through unsupervised machine learning methods. In this project, we tackled the topic: application of machine learning to a practical problem. Nowadays, Spotify has been one of the most popular music streaming platforms worldwide. It has attracted 433 M users so far. Their music recommendation system is one of the most important aspects leading to Spotify's success. We are interested in studying how the music recommendation system works in Spotify, build our version of the Spotify recommendation system. In this experiment, we will use the unsupervised machine learning methods K-means and Gaussian Mixture Model (GMM) as our baseline. Besides, we will use PCA as a way of dimension reduction on top of K-means and Gaussian Mixture Model (GMM) to improve our models.

3 Background

3.1 Dataset

We use a dataset from Kaggle: Spotify Dataset 1921-2020. It contains 600k clean Spotify user data. The input includes track name, popularity, danceability, energy, etc.

3.2 Models based on Previous Work

Based on previous research, recommendation systems can be broken down into three categories: collaborative filtering systems, content based filtering systems, and hybrid systems. Collaborative filtering systems recommend songs based on the overlap of songs in playlists. Content-based filtering systems recommend songs that are similar to the otter songs in the dataset. Hybrid systems can be considered as the combination of the content based filtering systems and collaborative filtering systems.

The existing experiments on Spotify music recommendation systems mainly focus on content-based filtering systems method with naive K-means clustering to organize the songs in clusters and predict the songs users may like. Most of the previous work did not inspect the clustering quality. As for unsupervised learning models, it leaves the quality of recommendation system unknown.

In our experiment, we mainly use content-based filtering systems, which includes unsupervised learning methods of K-means clustering and Gaussian Mixture Model (GMM). Besides, we used Principal component analysis (PCA) for dimensionality reduction in order to get better-separated clusters.

4 Method

4.1 Pre-processing

First of all, we downloaded the dataset from Kaggle: Spotify Dataset 1921-2020 and imported the packages needed for our experiments, such as pandas, numpy, and random. Then, since the dataset is pretty large (586,672 tracks in total), we pre-processed the dataset by removing those tracks without names, trimming release dates to make them consistent, dropping some qualitative features since K-means ask for quantitative data, and we want to focus on the recent tracks (tracks released on or after 2016) for the preliminary experiment. After these pre-processing steps, we still have 52,950 tracks in the data frame. We also normalized the data to keep all features in the same scale for better results. We also split the processed data into training set and validation set, and used the validation set to determine the proper number of clusters.

4.2 K-means and Elbow method

First, we will use K-means clustering as our baseline model since K-means is the most flexible and scalable unsupervised machine learning model and can work on large datasets. The KMeans algorithm clusters data by trying to separate samples in n groups, minimizing the sum-of-squares of the distance between points x in the cluster j and the centroid μ_j of the cluster j, that is, the mean of all the points in the cluster j, as indicated below:

$$\sum_{i=0}^{n} min_{\mu_j \in C} ||x_i - \mu_j||^2$$

Since we don't know the exact number of clusters, we will make use of elbow method to figure out the possible number of clusters by plotting out the performance of the K-Means Model on validation set and capturing the turning point of the performance graph.

4.3 Gaussian Mixture Model (GMM)

Next, we will use Gaussian Mixture Model to explore the clusters within the dataset. GMM implements the expectation-maximization (EM) algorithm for estimating the parameters μ, Σ, ϕ of each cluster. The probability of the data x for each cluster is a multivariate Gaussian $P_\theta(x|y=k) = N(x;\mu_k,\Sigma_k)$. Since we also suffer from lack of knowledge of the exact number of clusters when applying GMM, we will again make use of elbow method to figure out the possible number of clusters. After deciding on the number of clusters, we will proceed to generate the recommended songs from the clusters picked out by Gaussian Mixture Model.

4.4 Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) is an unsupervised learning method to reduce the dimensionality of high-dimensional datasets. For example, GMM is very sensitive to data dimensionality. Thus, we applied PCA to improve the clustering quality. PCA also preserves the original structure and relationships inherent to the original dataset so that machine learning models are able to learn from them and make accurate predictions. PCA aims to minimize the reconstruction error

$$J_1(W) = \sum_{i=1}^{n} ||x^{(i)} - \tilde{x}^{(i)}||_2^2 = \sum_{i=1}^{n} ||x^{(i)} - WW^T x^{(i)}||_2^2$$

between each input $\tilde{x}^{(i)}$ and its approximate reconstruction

$$\tilde{x}^{(i)} = W \cdot z^{(i)} = W \cdot W^T \cdot x^{(i)}$$

4.5 Silhouette Coefficient

The Silhouette Coefficient is a metric used to calculate the goodness of a clustering method. Its value ranges from -1 to 1. Among them, "-1" means clusters are assigned in the wrong way, "0" means clusters are indifferent and the distance between clusters is not significant, and “1” means clusters are very apart from each other and clearly distinguished. The Silhouette Coefficient s for a single sample s is then given as:

$$s = \frac{b - a}{max(a, b)}$$

where a represents the mean distance between a sample and all other points in the same class, and b represents the mean distance between a sample and all other points in the next nearest cluster.

5 Experimental Analysis

5.1 Baseline and Elbow Method

We created the baseline model using K-Means clustering, and then used the validation set to get the best value K for K-means.

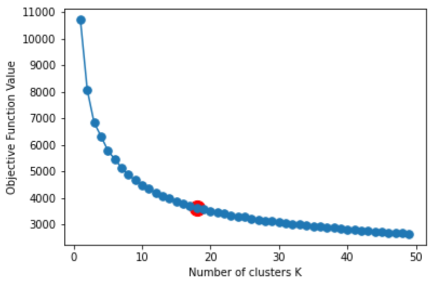


Figure 1: Elbow method with J score from K-Means

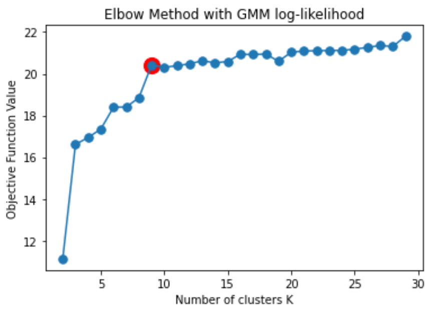


Figure 2: Elbow method with log-likelihood from GMM

From Figure 1, we chose $k = 18$ at the "elbow", where the rate of j-score decrease substantially slows down. Then we created the clustering model. The current K-means objective from the model is 14456.88. The objective seems large mainly because it is the sum of squares of the distances of points from their respective cluster centroids. Since we have a large dataset, the K-means objective is reasonably large given the dataset size. We also used $k = 9$ for GMM, based on the log-likelihood in Figure 2.

5.2 Silhouette Coefficient

However, the J score from K-Means clustering and the log-likelihood from GMM are not comparable. Thus, we used the Silhouette Coefficient for evaluation (Figure 3-4).

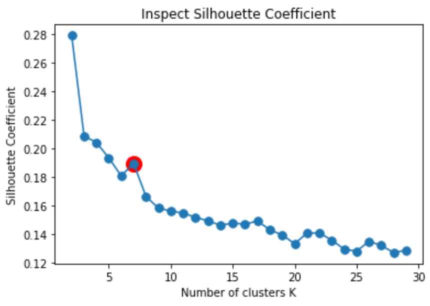


Figure 3: Silhouette Coefficient in K-Means clustering

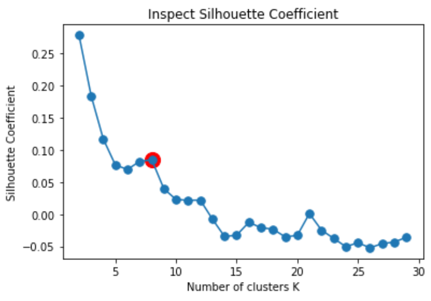


Figure 4: Silhouette Coefficient in GMM

The resulting Silhouette Coefficient for the K-Means model is 0.15, while the Silhouette Coefficient for GMM is 0.02. We visualized the clusters generated by these two models. Compared to the visualization generated by K-Means clustering (Figure 5), the differences between the clusters from GMM are less obvious. In GMM cluster visualization (Figure 6), cluster 2 and 5 look similar, while cluster 3 and 6 look nearly identical to each other.

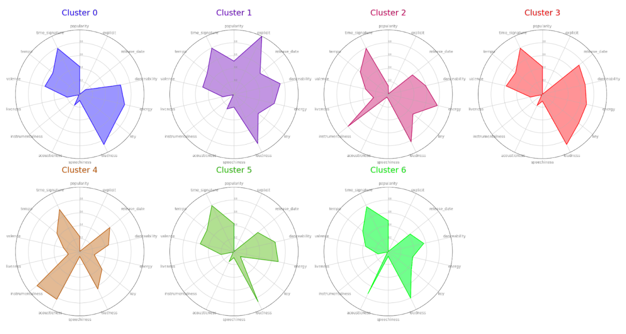


Figure 5: Average clusters generated by K-Means clustering

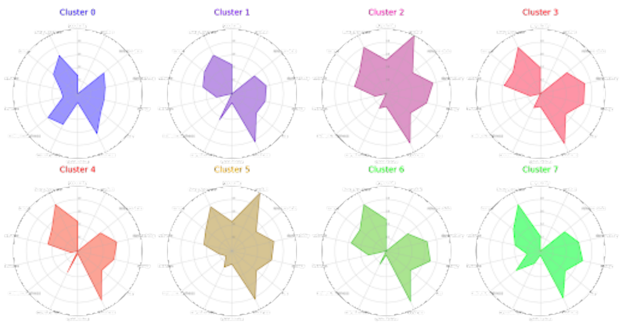


Figure 6: Average clusters generated by GMM

5.3 Dimensionality Reduction

5.3.1 Further Pre-processing

To improve the model, we re-inspected our pre-processing steps. duration_ms is not directly related to the characteristics of the tracks, while mode is a categorical feature with only 2 values (0 and 1), which might introduce more chaos given that our number of clusters are larger than 2. Thus, to improve the model performance, we dropped these two columns. After the inspection, the Silhouette Coefficient of K-Means clustering improved to 0.19.

5.3.2 Hyperparameter Tuning

We realized that dimensionality reduction may help us better separate those clusters. As a result, we applied PCA before using K-Means and GMM for clustering. Meanwhile, we have to figure out the hyperparameters for PCA and the clustering models. Given that the pre-processed data has 14 columns in total, we have experimented with the combinations of n_components (for PCA) and n_clusters (for K-Means) / n_components (for GMM). We chose the hyperparameters according to the following principles:

- We do not want to either overfit or underfit the model.
- We do not want to have too few clusters: there are many music genres, so having too few clusters does not really make sense when recommending songs to users.
- We want to make the clusters as well-separated as possible.

As a result, we proceeded with PCA n_components=2 and GMM n_components=7 for GMM clustering, PCA n_components=2 and K-Means n_clusters=7 for K-Means clustering.

5.3.3 Results after applying PCA

With PCA and the hyperparameters above, the Silhouette Coefficient of K-Means clustering has increased to 0.4, while the Silhouette Coefficient of GMM clustering has increased to 0.4 as well.

The visualizations from Figure 5-8 shows the averages on each dimension of the clusters. According to the Silhouette Coefficient and the visualizations, both clustering methods have better separations after applying dimensionality reduction. Especially for GMM in visualization, the clusters are separated much better compared to the baseline.

6 Discussion and Prior Work

As we are aware, most mainstream platforms build recommendation systems through supervised filtering which recommend items based on the usage history of users. However, we are exploring the recommendation mechanisms through content-based filtering with unsupervised learning, which recommend items based on the similarities of contents instead of the user, due to the limitations of lack of access to private user data. Unlike

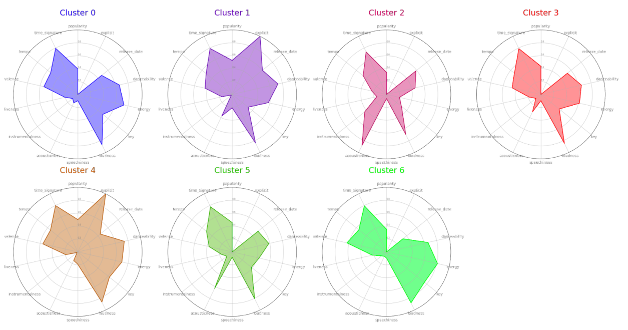


Figure 7: Average clusters generated by PCA + K-Means clustering

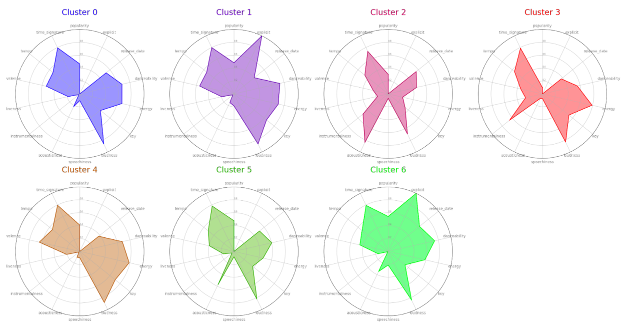
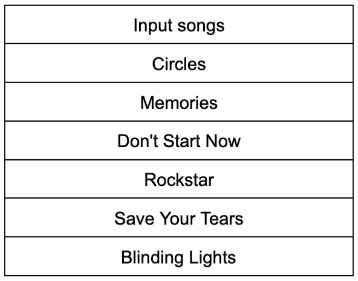


Figure 8: Average clusters generated by PCA + GMM

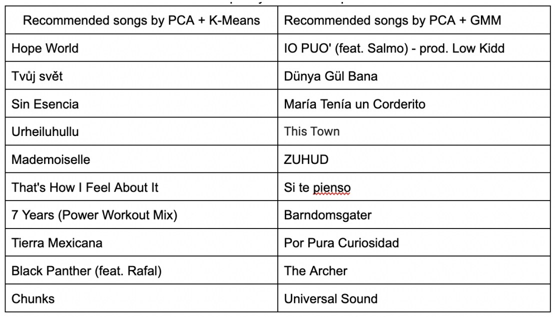
the existing unsupervised filtering in the market which just uses naive K-means, we aims for more accurate clustering result defined by high exclusivity of the clusters. We noted a significant improvement of performance when applying PCA to the clustering models in order to trim down some attributes, which leads to an increase of silhouette coefficient from 0.19 to 0.4 for K-means and 0.02 to 0.4 for GMM. Also, as we go about the problem, we noticed that the standalone Gaussian Mixture Model does poorly. We think standalone GMM cannot handle high dimensional data very well, because there might be too many parameters to estimate in EM step with so many attributes. With too many attributes, there are high chances of co-dependence between attributes, which could result in a non Gaussian dataset.

As a result, with unsupervised learning methods and experiments above, we got the recommended songs in the table below. We tried a few of them. These songs sound pretty similar to the tracks given, and are in better quality compared to the songs recommended by the naive model.



Input songs
Circles
Memories
Don't Start Now
Rockstar
Save Your Tears
Blinding Lights

Table 1: Input songs of the recommendation system



Recommended songs by PCA + K-Means	Recommended songs by PCA + GMM
Hope World	IO PUO' (feat. Salmo) - prod. Low Kidd
Tvůj svět	Dünya Gül Bana
Sin Esencia	María Tenía un Corderito
Urheiluhullu	This Town
Mademoiselle	ZUHUD
That's How I Feel About It	Si te pienso
7 Years (Power Workout Mix)	Barndomsgater
Tierra Mexicana	Por Pura Curiosidad
Black Panther (feat. Rafal)	The Archer
Chunks	Universal Sound

Table 2: Recommendation output by the two best performant models

7 Conclusion and Future Work

In conclusion, we approached designing recommendation systems by developing four machine learning models based on processed dataset and compared their performances. For the future experiments, model-wise, we would like to use a hybrid approach that combines the advantages of both collaborative and content-based filtering after we gather some user data by conducting user research, or having access to Spotify user data. With collaborative filtering, we could render more supervised learning models to gain a greater confidence in the derived recommendation result.

8 Reference

1. “About Spotify.” Spotify, 25 Oct. 2022, https://newsroom.spotify.com/company-info.

2. Ay, Yamac Eren. “Spotify Dataset 1921-2020, 600K+ Tracks.” Kaggle, 13 Mar. 2022, https://www.kaggle.com/datasets/yamaerenay/spotify-dataset-19212020-600k-tracks.

3. Holtz, Yan. “Use Faceting for Radar Chart.” The Python Graph Gallery, https://www.python-graph-gallery.com/392-use-faceting-for-radar-chart.

4. Kuleshov, Volodymyr. “Applied Machine Learning Lecture 8: Unsupervised Learning.” GitHub, https://github.com/kuleshov/cornell-cs5785-2022-applied-ml/blob/main/slides/lecture8-unsupervised-learning.ipynb.

5. Kuleshov, Volodymyr. “Applied Machine Learning Lecture 11: Dimensionality Reduction.” GitHub, https://github.com/kuleshov/cornell-cs5785-2022-applied-ml/blob/main/slides/lecture11-dimensionality-reduc
ipynb.

6. “Sklearn.metrics.silhouettes core.” Scikit, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.

7. “Overview of clustering methods.” Scikit, https://scikit-learn.org/stable/modules/clustering.html#overview-of-clustering-methods.

8. “Building a Song Recommendation System with Spotify.” Chang, https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7.