DS 807 UNSTRUCTURED DATA

EXPLORING MUSIC GENRE PREFERENCES ON SPOTIFY

MAY 11, 2023

REETHIKA REDDY CHALLA, NAVEEN KUMAR KRISHNASAMY, DIKSHANT JOSHI

Table of Contents

Introduction

Music has been an integral part of human life. Over time, the way we listen to and discover music has evolved. There are many music streaming platforms where we can discover new music daily. Spotify is one such platform. Music is a mixture of different characteristics, and through these characteristics, it is differentiating in different genres. In this project, we are exploring different music genres by grouping the similar audio features of the songs and identifying the characteristics of these groups using several clustering algorithms.

Dataset:

The Spotify dataset is downloaded from Kaggle. The dataset consists of 114000 observations with 20 different variables. It has 14 numerical variables, 1 binary variable, and 5-character variables. The numerical variables are the audio features of the songs on Spotify. This dataset is a supervised dataset but to make it unsupervised, we are not considering the ID and track id variables in our analysis.

Variables:

Variable	Description
Artists	The artists' names who performed the track. If there is more than one artist, they are separated by ";"
Album name	The album name in which the track appears
Track name	Name of the track
Popularity	The popularity of a track is a value between 0 and 100, with 100 being the most popular
Duration	Length of the time in ms(milli second)
Explicit	Whether the track has explicit lyrics(binary)
Danceability	Danceability describes how suitable a track is for dancing (A value of 0.0 is least danceable and 1.0 is most danceable)
Energy	Measure of intensity and activity (A value of 0.0 is least energy and 1.0 is most danceable)
Key	The key the track is in. Integers map to pitches using standard Pitch Class notation.
Loudness	The overall loudness of a track in decibels (dB)

Mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
Speechiness	Speechiness detects the presence of spoken words in a track.
Acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
Instrumentalness	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context.
Liveness	Detects the presence of an audience in the recording
Valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive
Tempo	The overall estimated tempo of a track in beats per minute (BPM).
Time signature	An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar
Track genre	The genre in which the track belongs
Year	Year in which the song released

Pre-processing Data:

The dataset consists of NAs in the column Year. Since the year in which the song has been released isn't an important variable in our analysis, we did not consider the variable year. The variables in the dataset are completely different scales. Clustering algorithms are very sensitive toward the scale of the variables considered. If the scales of the variables are different, the clustering can be biased towards the larger scale. To balance out our data, we scaled the dataset by taking the required variables.

Problem Objective

The aim is to group similar songs on Spotify by analyzing their audio features and identifying the characteristics of these clusters. This information can then be used to understand user preferences.

Exploratory Data Analysis

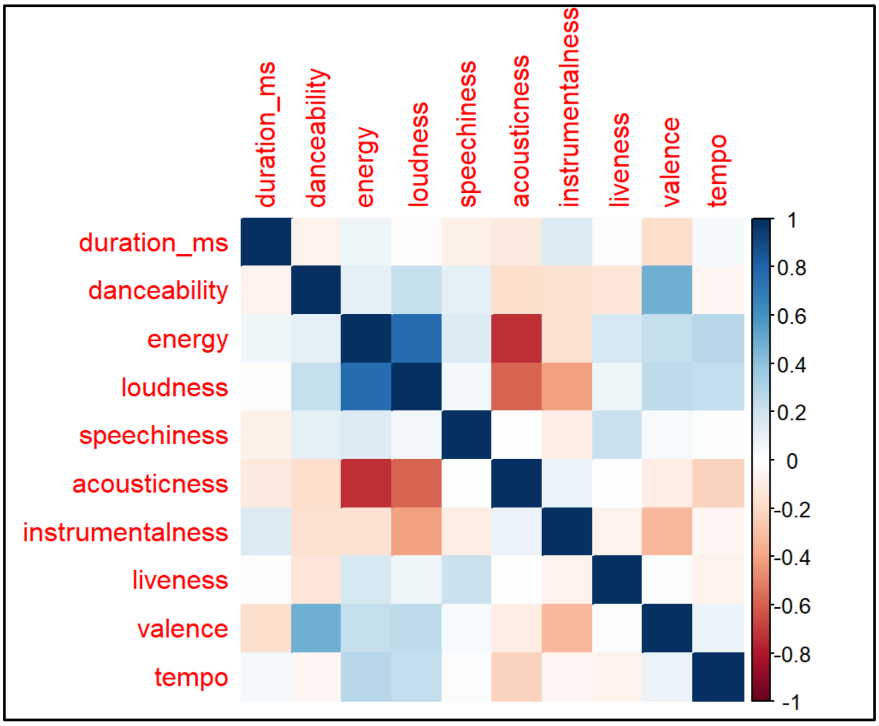

Fig 1: Correlation matrix between numerical variables

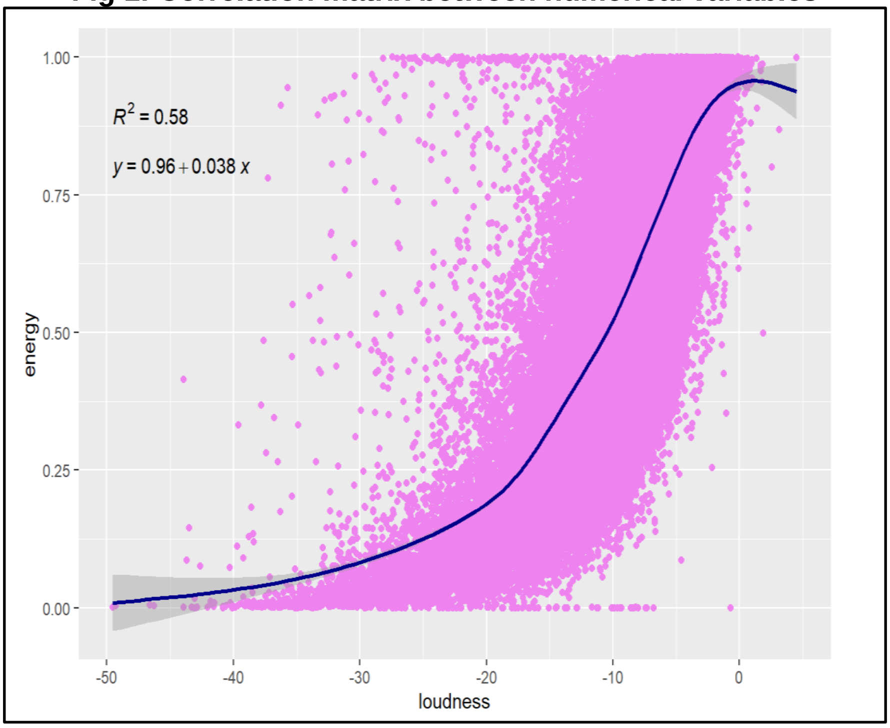

Fig 2: Regression plot between Loudness and Energy

We created a correlation matrix between numerical variables to see if there is a correlation between the characteristics of the songs that we will be using for clustering. The correlation matrix shows a high correlation between Energy & Loudness, Acousticness & Energy. Since both loudness and acousticness have a correlation with energy, we removed Energy variable for our analysis.

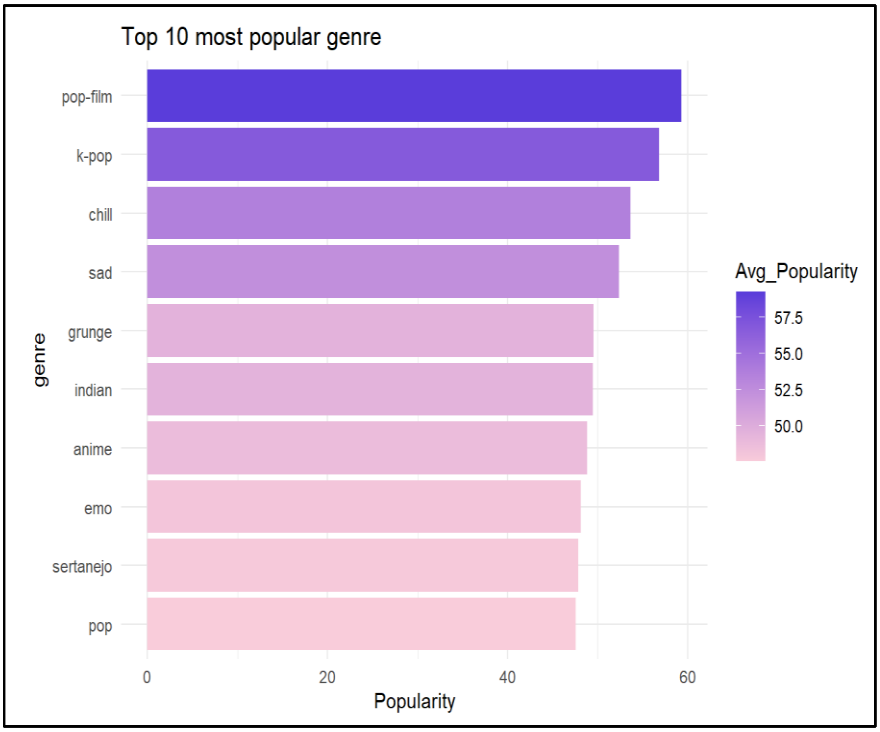

Fig 3: Top 10 popular genres from the list of songs

We created an inverted bar chart to see what the top 10 most popular genres are. We took a mean of the Popularity score for each genre to check the Avg_popularity. The top 5 popular genres which we obtained are pop film, k-pop, chill, sad, and grunge.

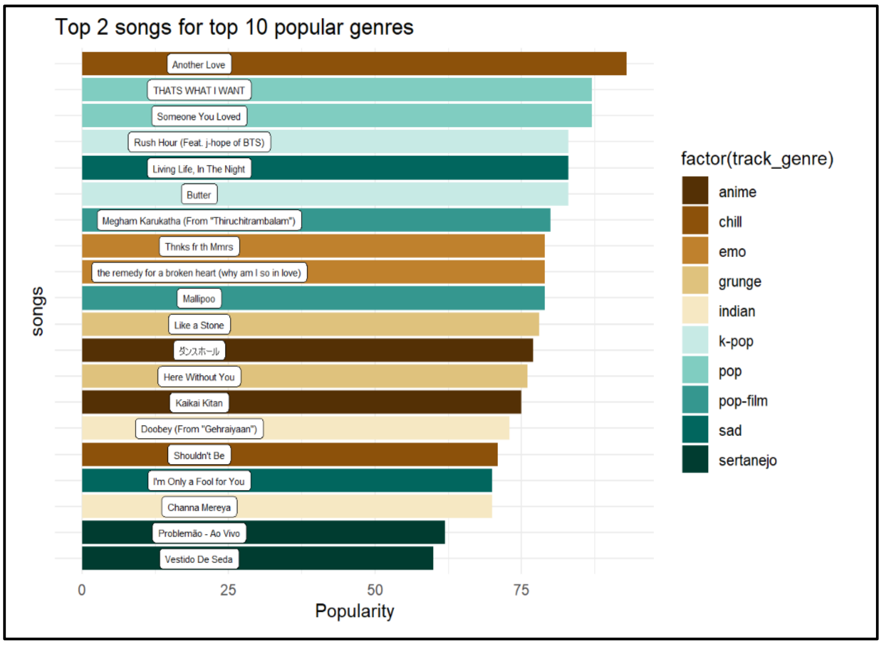

Fig 4: Top 2 songs in each popular genre

We created an inverted stacked bar chart with the top 2 songs in each of the above top 10 genres based on the Popularity score. It's interesting to know and analyze practically to see the songs we actually know fit into what kind of genres. This gave us a better understanding of how the genres are differentiated.

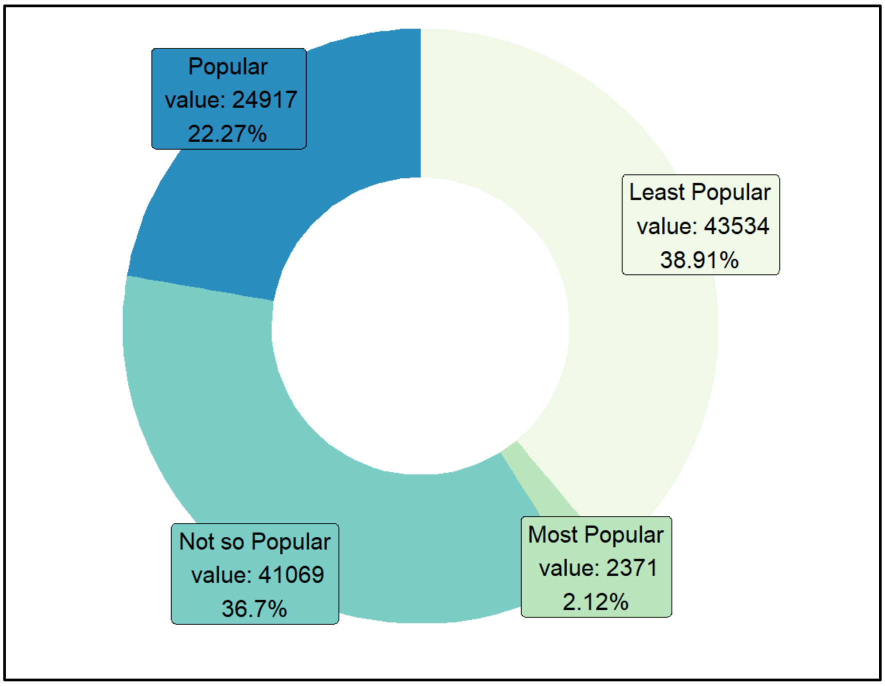

Fig 5: Proportions of the popularity of the songs

We created a donut chart to check how many songs in our dataset accounts into four categories of popularity.
Most Popular: (75-100) Popularity Score
Popular: (50-75) Popularity Score
Not so Popular: (25-50) Popularity Score
Least Popular: (0-25) Popularity Score

Deep Learning

Feed Forward Neural Network Algorithm

Our dataset had music elements like key, mode (major/minor), time signature, and duration of the song. We wanted to test and predict the song characteristics based on these elements. The characteristics become the dependent variables and they are,

- Speechiness
- acousticness
- instrumentalness
- liveness
- valence
- danceability
- energy
- loudness
- tempo

We filtered out the numeric variables mentioned above and created a data frame. Data is then split into training and testing data in a 7:3 ratio.
Our input layer was to units of 64 (meaning training the model with 64 values at a time) and the hidden layer to 32 units and used the ReLU activation function. We set our output layer activation function to Linear.

```
# Define MLP model architecture
model=keras_model_sequential()
model %>%
  layer_dense(units=64, activation='relu', input_shape=ncol(x_train)) %>%
  layer_dropout(rate = 0.4) %>%
  layer_dense(units = 32, activation = 'relu') %>%
  layer_dropout(rate = 0.4) %>%
  layer_dense(units =8, activation='linear')
```

```
Model: "sequential"
________________________________________________________________________________
Layer (type)                    Output Shape                     Param #
================================================================================
dense_2 (Dense)                 (None, 64)                       384
dropout_1 (Dropout)             (None, 64)                       0
dense_1 (Dense)                 (None, 32)                       2080
dropout (Dropout)               (None, 32)                       0
dense (Dense)                   (None, 8)                        264
================================================================================
Total params: 2,728
Trainable params: 2,728
Non-trainable params: 0
```

Model

We chose MSE as the loss function and rmsprop as the optimizer as we learned that it accelerates the optimization process. The metric chosen is mae (mean absolute error) to understand the performance of the algorithm.

```
# Compile model
model %>%
  compile(loss = 'mse',
          optimizer = 'rmsprop',
          metrics=c('accuracy', 'mae')
          )

# Train model
history<-model %>%
  fit(x_train, y_train, epochs = 50, batch_size = 64,verbose=1,validation_split = 0.2)
```

Result

The prediction evaluation metric, mae=0.578, which means the prediction differs from test data by 0.578. This difference is high and further, the prediction Vs test data plot showed the difference in scale in them. So, we conclude that the FFNN algorithm dint provides us with good results.
MAE= 0.5784452

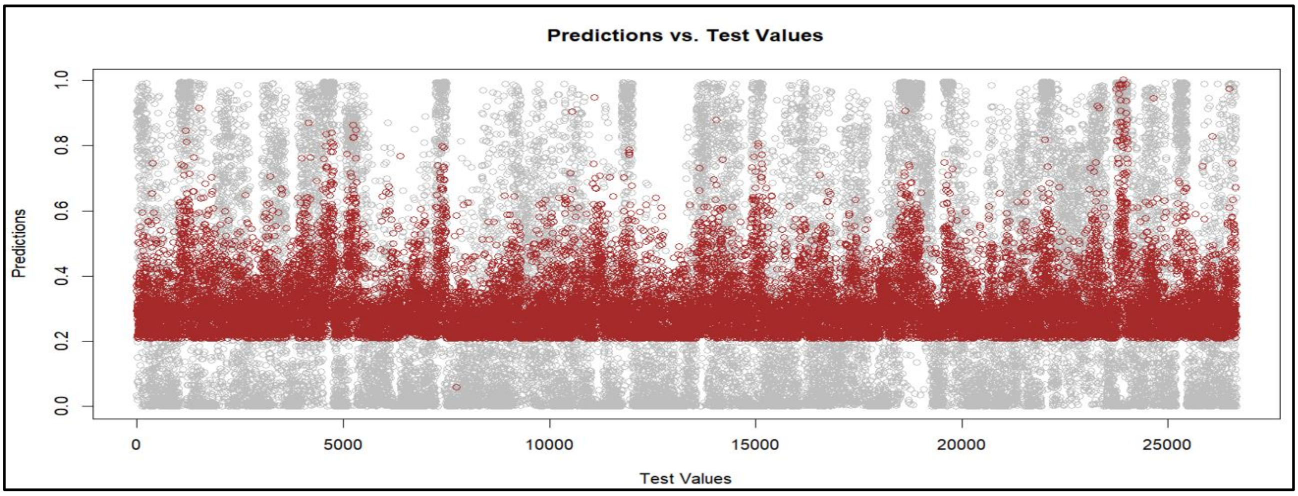

Fig 6: Prediction vs Test results for FFNN

Clustering

K-means Clustering

To classify the songs based on their characteristics, we conducted k-means clustering. To decide on the number of clusters, we performed the elbow method, gap statistics, and silhouette method. But due to computational difficulty with big dataset, we checked ran and used only the elbow method on r studio.

Elbow Method for determining k:

The Elbow method looks at the total WSS as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't improve much better the total WSS. From our viz plot on R, we could see that the minimum number of clusters starts at 2. Since differentiating with only 2 clusters does not give much information, we chose our number of clusters as 3.

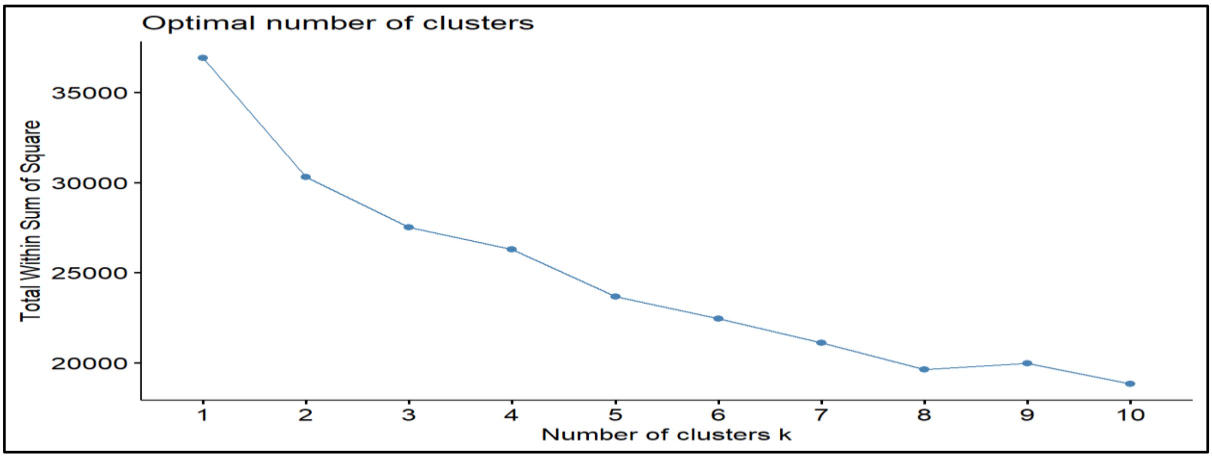

Fig 7: Optimal number of clusters using WSS method.

We cleaned the dataset to filter out numeric variables for the k-means analysis. Below are the clutter plots from the fviz_cluster function.

```
#K-means Clustering Method
set.seed(1994)
km=kmeans(spotify_knn, 3)
fviz_cluster(km, data =spotify_knn,palette = "jco", ggtheme = theme_minimal())
```

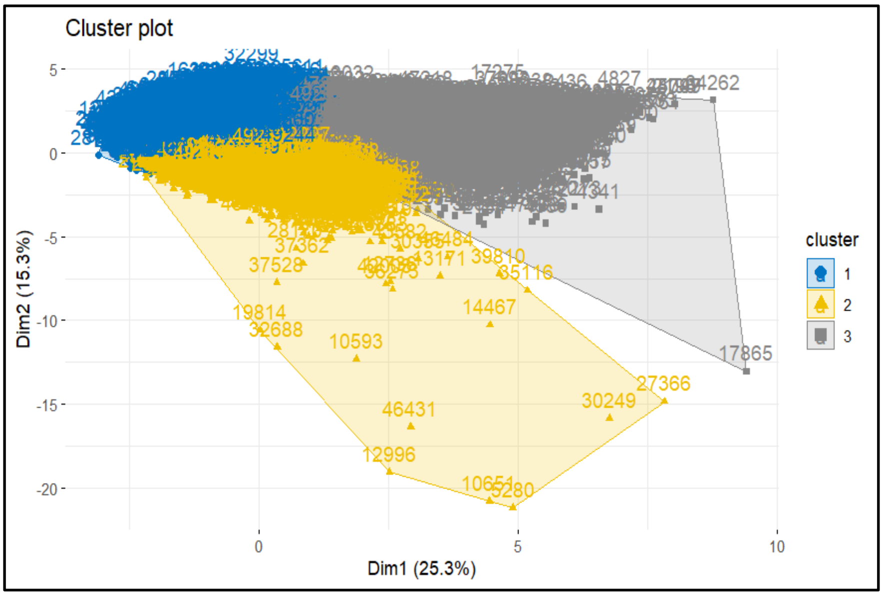

Fig 8: Plot showing different cluster data points.

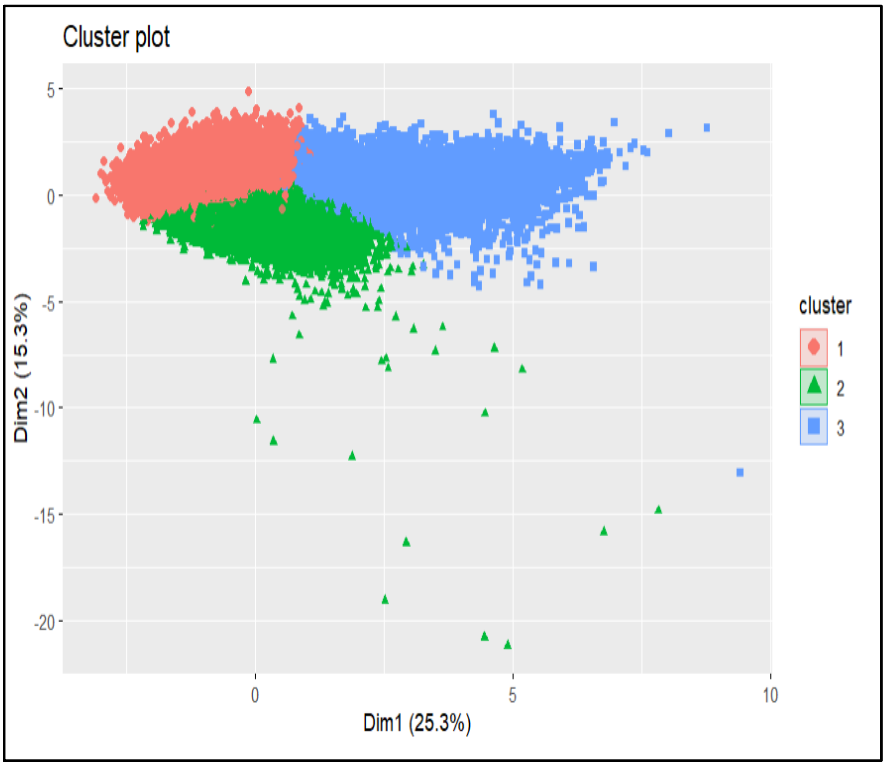

Fig 9: Plot showing three clusters.

Result

	duration_ms	tempo	speechiness	acousticness	instrumentalness	liveness	valence	danceability	loudness
1	-0.2058308	-0.1218442	0.1830637	-0.04989368	-0.3831004	-0.008183509	0.7376848	0.6251891	0.2554336
2	0.3343761	0.4260074	-0.1086889	-0.60397943	0.1390831	0.094752499	-0.5923243	-0.4451625	0.3659715
3	-0.1324047	-0.5731729	-0.2916271	1.45407744	0.8020150	-0.181987151	-0.8403088	-0.8354624	-1.5299044

km$clusters

Cluster 1: Songs high in danceability, loud and valance are grouped under cluster 1. Some songs randomly picked showed these characteristics and we could name them as positive, cheerful songs.

Cluster 2: Songs high in loudness, liveness, and tempo characteristics and can be related to inspirational songs.
Cluster 3: This cluster has songs that are high in acoustic, instrumental, and comparably less duration. This type of music emphasizes simplicity in its lyrics, harmonies, melodies, or melancholy.

	artists	track_name	cluster
1	Justin Bieber	Mistletoe	1
2	Omar S;L'Renee	S.E.X - C.G.P (Conant Gardens Posse) Remix	1
3	STU48	夏の"好き"はご用心	1
4	Keenan Te	Halfway There	1
5	BBS Paranoicos	La Rabia	2
6	Stabil	Kovala - Orijinal Film Müziği	1
7	Tale Of Us;Mind Against	Astral	2
8	Weston Estate	Silence	2
9	Bob Marley & The Wailers	Jamming	1
10	Rex Williams	You Are My Heart	1
11	MC Arraia;DJ Guih Da ZO	Tu Só Foi Mais Uma	1
12	Maneva	Vem Ver (Ao Vivo)	1
13	Vanna	Toxic Pretender	2
14	Alka Yagnik;Hariharan	Bahon Ke Darmiyan - Khamoshi - The Musical / Soundtrack ...	2
15	Melorman	Love in the 90's	3
16	Front Line Assembly	Killing Ground	2

We also noticed that some characteristics are repeated in clusters and would definitely look into having more clusters to classify them even better. We believe the mixture model to classify the songs would give more cluster options, hence we decided to do a mixture model next.

Mixture Model Clustering

Mixture models assume that the data is in different probabilities of distribution, unlike regular clustering algorithms. For the mixture models, we used mclust function as our dataset is in Gaussian distribution. Mclust function takes the Gaussian distribution by default. The optimal number of clusters is chosen by the mclust function by analyzing the distributions of the data points. For our analysis, through mclust we obtained 9 optimal clusters. This is because mixture models are more flexible and hence the algorithm is considering the ambiguity and complexity of the data. We can see below the probabilities of the number of songs that could go into each cluster. Cluster 7 has the highest probability.

```
Clustering table:
   1    2    3    4    5    6    7    8    9
 255 1382  193  242   49  127  219  361  528

Mixing probabilities:
         1          2          3          4          5          6          7          8          9
0.08305062 0.37072018 0.05756200 0.07192280 0.01486967 0.03826171 0.06904137 0.11248407 0.18208758
```

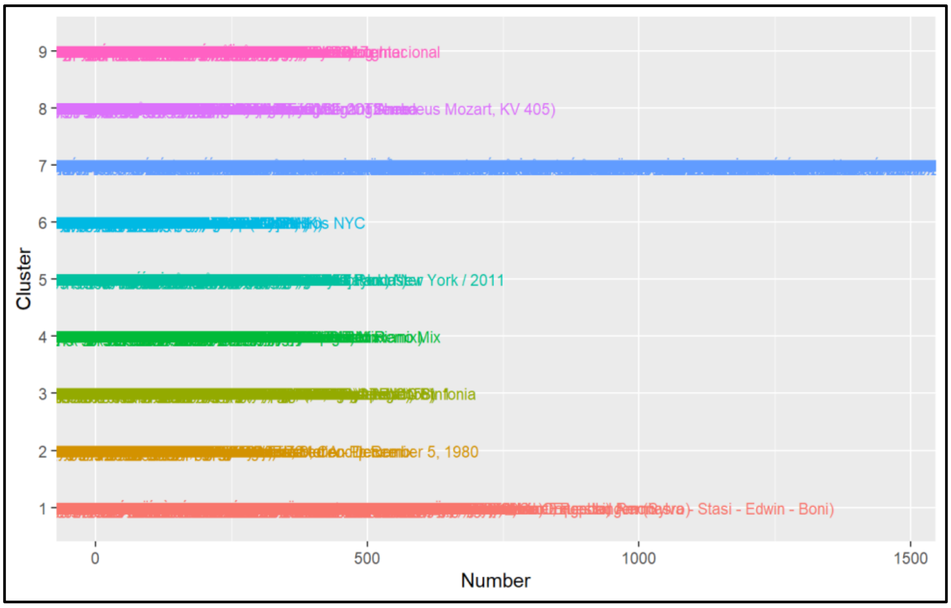

Fig 10: Probability of songs in each cluster

To understand the characteristics of each cluster, we observed the mean values of each characteristic.

```
Means:
                        [,1]        [,2]        [,3]        [,4]        [,5]       [,6]       [,7]        [,8]        [,9]
duration_ms      0.007217376 -0.13287845 -0.04554537  0.90535200  0.09093693 -0.6483557  0.1277918  0.17350544 -0.10279106
danceability    -0.288552824  0.14711159 -1.43522808  0.49829371  0.09690925  0.3878849  0.3503354 -0.55797246  0.21141696
energy          -1.141356565  0.05501465 -1.85331967  0.18222377 -0.21599547 -0.8421344  0.3444387  1.04452678  0.34121057
loudness        -0.642108347  0.27546812 -2.62545776 -0.23071265 -0.20982055 -0.7000997  0.2482227  0.55634152  0.37957438
speechiness     -0.439048776  0.25286878 -0.39771298 -0.29928167 -0.08683183  0.2591509 -0.2376797  0.05354536 -0.06095762
acousticness     0.949323879  0.06642938  1.71099764 -0.65984171  0.04641164  0.7792668 -0.3173131 -0.94166137 -0.31400272
instrumentalness -0.494583142 -0.50402829  2.07044280  1.77929309 -0.30252443  1.2335314 -0.4355367  0.53213722 -0.50364314
liveness        -0.407195331  0.15676366 -0.19097817 -0.34605552 -0.57830048 -0.4966837  0.1730620  0.09862990  0.08866673
valence         -0.415161557  0.26268516 -1.08673101 -0.23959588  0.09048564 -0.2624196  0.4625781 -0.51317462  0.28209294
tempo           -0.313334489  0.02957887 -0.69610106  0.04525991  0.34846045 -0.1157012 -0.1194940  0.35446576  0.10706167
Year            -0.017181818  0.01726525  0.01244297 -0.12832323  1.39002061 -0.1063216 -0.1716945  0.01812666 -0.01782951
```

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
0.206305	0.03537	0.04614	0.0762	0.08382	0.0331	0.39235	0.03884	0.08087
Valence, Loudness	Acousticness, Instrumentalness, Liveness	Acousticness, Instrumentalness	Danceability	Loudness, Speechiness	Danceability, Valence	Danceability, energy, loudness	Loudness, Speechiness	Instrumentalness, Loudness

Conclusion

K- means clustering did a great work in clustering the songs into 3 clusters.
We can see the results below our model clustered these songs into respective clusters as mentioned and they perfectly align with the characteristics mentioned with the cluster.

Clusters	Characteristics	Song
Cluster-1	Danceability, Loudness	Attention
Cluster-2	Liveness, Loudness, High Tempo	Caslte of the Glass
Cluster-3	Instrumentalness, Acousticness	Kun Faya Kun

Using these clusters, we can recommend other songs in the same cluster to users whose most of the songs in the playlist fall into any one of the three clusters. For instance, if most of the songs in the playlist of a user fall into cluster 1 we will recommend to him the songs in cluster 1 which are not in his playlist.

Limitations and Future Direction

From the results we obtained for Mixture models, the interpretation of clusters is very difficult since most of the clusters have similar characteristics. For future direction, we could build a recommendation system for personalized playlists to users based on the clusters of the songs. The music industry can gain better insights from the popular songs and their characteristics. Also, we could recommend a song to a user based on his most repeated songs and recommend songs from similar clusters.

References

[1] https://www.kaggle.com/datasets/vatsalmavani/spotify-dataset

[2] https://scikit-learn.org/stable/modules/mixture.html