Final Group Project Report

Music Predictive Analytics
CSP571- Data Preparation and Analysis

Group Members:
Shrey Jaradi (A20518260)
Chris Chen (A20502563)
Abdul Raqeeb Muhammad Yousaf (A20471861)
Sohaib Jawad (A20516885)

TABLE OF CONTENT

Abstract

The taste of music has changed throughout history. Now that access to new music is easier than ever, music platforms are finding specific trends and traits to outperform competitors and attract more users using machine learning. In this project, we want to dive into the field and figure out for ourselves how music platforms use the data they collected to give their users better experience in music recommendations, how they categorize music into micro-genres and much more. One motivational example could be that of Spotify mobile App, how they have segmented their archive into thousands of micro-genres.

Overview

Problem Statement

The following questions we are trying to answers in our project. Along with this if possible we might find out some more insights from the data. But as we have refer different datasources. We are thinking of answering these questions with the dataset that does have those features, which we require to predict.

1. Predict danceability of a song belong to based on specific features.
2. Predict the year of the song based on different characteristics like album cover etc.
3. Given a song, what are some other recommendations that would suit the same user's taste?
4. Can we predict the popularity of a song based on given features?
5. Which genre got famous/changed according to year and why?
6. What features affect the popularity of a song the most?
7. Who are the popular artists?
8. Predicting if a new artist is going to be popular based on its audio features and predict popularity score.

Proposed Methodology

The Approach taken towards answering our research question was that we loaded the data from the different dataset to answer each question. Since each question required features/information taht coil not be found in a single dataset, we were required to utilize different dataset to answer different research questions. Once the data was loaded, the data was then cleaned and then performed the analysis/visualization. In addition to the research question, we were also able to answer additional questions while working on the research question. These additional questions are mentioned in the analysis.

Where required, the dataset were divided into training and testing set and then were anayzed for regression, classification or clustering depending on the research question like for predicting the year of the song based on different characteristics, we used classification. Once we train the models, we determine the accuracy, MSE, R-square values to check the performance of the models to ensure that the selected model is the correct model.

Data Sources

Dataset 1:

Core Information:

Core information: The dataset is from Kaggle (https://www.kaggle.com/datasets/lehaknarnauli/spotify-datasets?resource=download&select=tracks.csv). This dataset has audio features of over 500,000 songs. This is an open source and can be downloaded as a CSV file from Kaggle.

Metadata: This dataset has two subsets of data, artists & tracks. Here are the details for each dataset:

Tracks: It consists of 20 columns and has 586,672 rows. Each column defines a specific feature of the song. Some of the relevant columns for our purpose are as follows:

Field Name	Type	Brief Description
ID	String	A unique identifier for the song
name	String	The name of the song
popularity	Numeric	Defines the popularity of the song. The value is between 0 to 100
duration_ms	Numeric	Defines the duration of the song in milliseconds
artists	String	The name of the artist
id_artists	String	A unique identifier for the artist
danceability	Numeric	Defines the danceability of the song. The value is between 0 to 1
energy	Numeric	Defines the energy of the song. The value is between 0 to 1
loudness	Numeric	Defines the loudness of the song. The value is between -60 to 6

speechiness	Numeric	Defines the speechiness of the song. The value is between 0 to 1
acousticness	Numeric	Defines the energy of the song. The value is between 0 to 1
liveness	Numeric	Defines the liveness of the song. The value is between 0 to 1
tempo	Numeric	Defines the tempo of the song. The value is between 0 to 250

Artists: It consists of 5 columns and has 1,104,349 rows. Each column defines a specific feature of the song. Some of the relevant columns for our purpose are as follows:

Field Name	Type	Brief Description
ID	String	A unique identifier for the artist. This column can be joined with the id_artist from the Tracks dataset
followers	Numeric	The number of followers the artist has
name	String	The name of the artist

According to Kaggle, this document has some missing or mismatched data in the 'Track' CSV file, specifically the 'name' column. This information can be found on this link. As we work on the dataset, we will verify this information as well.

Dataset 2:

Core information: The dataset is from Kaggle (https://www.kaggle.com/datasets/mrmorj/dataset-of-songs-in-spotify?resource=downloa d). This dataset again has audio features of different songs. This is an open source and can be downloaded as a CSV file from Kaggle.

Metadata: This dataset has two subsets of data, genres_v2 & playlists. Here are the details for each dataset:

genres_v2: It consists of 22 columns and has 42,306 rows. Each column defines a specific feature of the song. Some of the relevant columns for our purpose are as follows:

Field Name	Type	Brief Description
ID	String	A unique identifier for the song
genre	String	Defines the genre of the song

song_name	String	Defines the name of the song
danceability	Numeric	Defines the danceability of the song. The value is between 0 to 1
energy	Numeric	Defines the energy of the song. The value is between 0 to 1
loudness	Numeric	Defines the loudness of the song. The value is between -35 to 4
speechiness	Numeric	Defines the speechiness of the song. The value is between 0 to 1
acousticness	Numeric	Defines the acoustic-ness of the song. The value is between 0 to 1
liveness	Numeric	Defines the liveness of the song. The value is between 0 to 1
tempo	Numeric	Defines the tempo of the song. The value is between 55 to 220

playlists: It consists of 2 columns and only has 40 rows. The columns and its description are as follows:

Field Name	Type	Brief Description
playlist	String	A unique identifier for a playlist
genre	String	Defines the genre of the song

According to Kaggle, this document has some missing or mismatched data in the 'genres_v2' dataset, specifically the 'song_name' column. This information can be found on this link. As we work on the dataset, we will verify this information as well.

Dataset 3:

Core information: The dataset is from Kaggle (https://www.kaggle.com/code/akiboy96/spotify-song-popularity-genre-exploration/data?select=genre_music.csv). This dataset again has audio features of different songs. This is an open source and can be downloaded as a CSV file from Kaggle.

Metadata: This dataset has one file that is genre_music.csv. Here are the details for the dataset

genre_music.csv.: It consists of 20 columns and has 41,099 rows. Each column defines a specific feature of the song. Some of the relevant columns for our purpose are as follows:

Field Name	Type	Brief Description
ID	String	A unique identifier for the song
genre	String	Defines the genre of the song
track	String	Defines the name of the song
artist	String	Name of the Artist
danceability	Numeric	Defines the danceability of the song. The value is between 0 to 1
energy	Numeric	Defines the energy of the song. The value is between 0 to 1
key	numeric	Defines the key of the song, value is between 0 to 11
loudness	Numeric	Defines the loudness of the song. The value is between -35 to 4
mode	numeric	Define the mode of the song, values lies between 0 and 1
speechiness	Numeric	Defines the speechiness of the song. The value is between 0 to 1
acousticness	Numeric	Defines the acoustic-ness of the song. The value is between 0 to 1
liveness	Numeric	Defines the liveness of the song. The value is between 0 to 1
valence	numeric	Defines the valence of the song,value is inbetween 0 to 1
tempo	Numeric	Defines the tempo of the song. The value is between 55 to 220
duration_s	numeric	Defines the length of the song in millisecond
time_signature	numeric	Defines time signature of the songs, value between o to 5
chorus_hit	numeric	Defines the chorus hit of the song, value between 0 to 433
sections	numeric	Defines the sections of the song, value between 0 to 169
popularity	numeric	Defines the popularity of the song, value is 0 or 1
decade	string	Defines the decade in which song release

Data Processing

All the three datasets are CSV file and the size of all the files are approximately 350 MB combined. This dataset consists of string and numeric values, so importing them in R Studio would be easy. Some of the features in the dataset will not be relevant to our analyses and will be removed from the dataset. Before carrying out the analysis we will:

- Make sure that we do not have any duplicate rows.
- Remove any outliers.
- For any missing value, we will not consider that feature for that row in the analysis.

If the performance of the best model at the end still isn't up to the standard we will always go back to the data itself and do some more explorations and feature engineering like selecting only the relevant columns in the model.

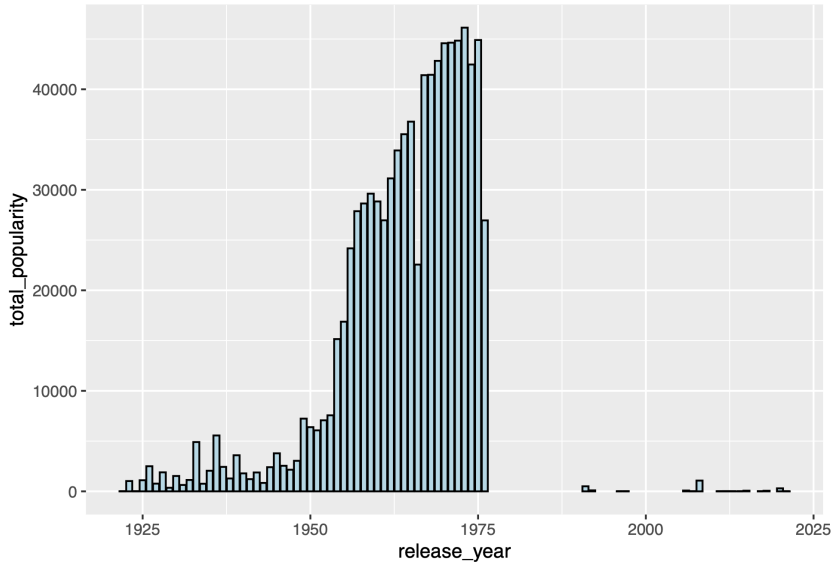
```
> summary(nData)
   popularity      duration_ms         explicit        release_year   danceability        energy           key            loudness      
 Min.   :  0.00   Min.   :   3344   Min.   :0.00000   Min.   :1900   Min.   :0.0000   Min.   :0.000   Min.   : 0.000   Min.   :-60.000  
 1st Qu.: 13.00   1st Qu.: 175093   1st Qu.:0.00000   1st Qu.:1974   1st Qu.:0.4530   1st Qu.:0.343   1st Qu.: 2.000   1st Qu.:-12.891  
 Median : 27.00   Median : 214893   Median :0.00000   Median :1992   Median :0.5770   Median :0.549   Median : 5.000   Median : -9.243  
 Mean   : 27.57   Mean   : 230051   Mean   :0.04409   Mean   :1989   Mean   :0.5636   Mean   :0.542   Mean   : 5.222   Mean   :-10.206  
 3rd Qu.: 41.00   3rd Qu.: 263867   3rd Qu.:0.00000   3rd Qu.:2007   3rd Qu.:0.6860   3rd Qu.:0.748   3rd Qu.: 8.000   3rd Qu.: -6.482  
 Max.   :100.00   Max.   :5621218   Max.   :1.00000   Max.   :2021   Max.   :0.9910   Max.   :1.000   Max.   :11.000   Max.   :  5.376  
      mode         speechiness      acousticness    instrumentalness       liveness         valence           tempo        time_signature  
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000000   Min.   :0.0000   Min.   :0.0000   Min.   :  0.0   Min.   :0.000  
 1st Qu.:0.0000   1st Qu.:0.0340   1st Qu.:0.0969   1st Qu.:0.0000000   1st Qu.:0.0983   1st Qu.:0.3460   1st Qu.: 95.6   1st Qu.:4.000  
 Median :1.0000   Median :0.0443   Median :0.4220   Median :0.0000245   Median :0.1390   Median :0.5640   Median :117.4   Median :4.000  
 Mean   :0.6588   Mean   :0.1049   Mean   :0.4499   Mean   :0.1134508   Mean   :0.2139   Mean   :0.5523   Mean   :118.5   Mean   :3.873  
 3rd Qu.:1.0000   3rd Qu.:0.0763   3rd Qu.:0.7850   3rd Qu.:0.0095500   3rd Qu.:0.2780   3rd Qu.:0.7690   3rd Qu.:136.3   3rd Qu.:4.000  
 Max.   :1.0000   Max.   :0.9710   Max.   :0.9960   Max.   :1.0000000   Max.   :1.0000   Max.   :1.0000   Max.   :246.4   Max.   :5.000  
```

This is the summary of all the numerical data in dataset 2. It will later be used to compare to each cluster of the KMeans model.

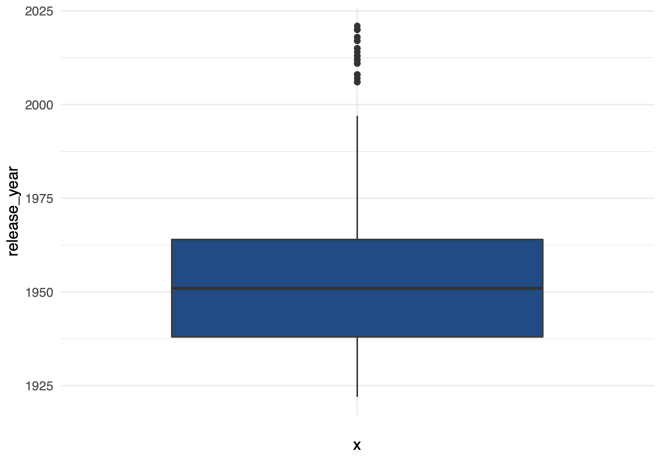
Data Analysis

Before training the model to evaluate and predict some of our question which we asked, we did some analysis, that also helps us to answer some of the questions.

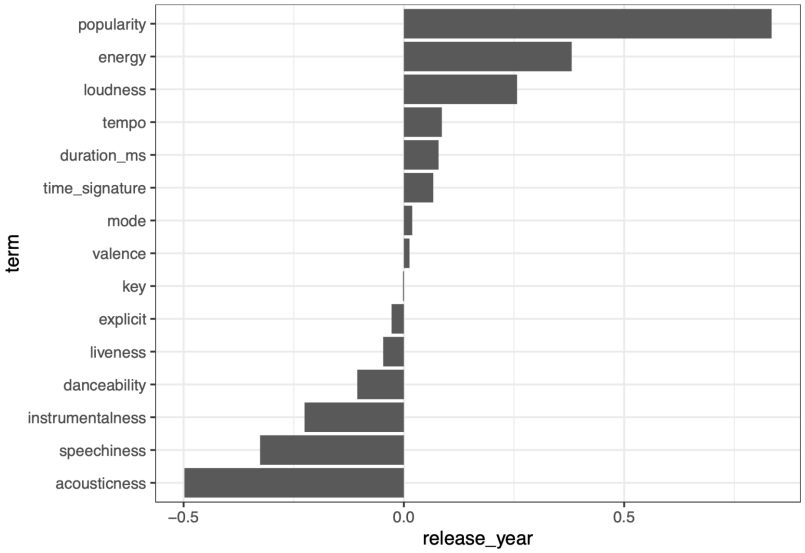
We tried to plot the popularity vary in each decade, how people start listening to music more. As after 1975 it shows less data because the size of the data was too large, so try to reduce it. We were facing memory issue so.



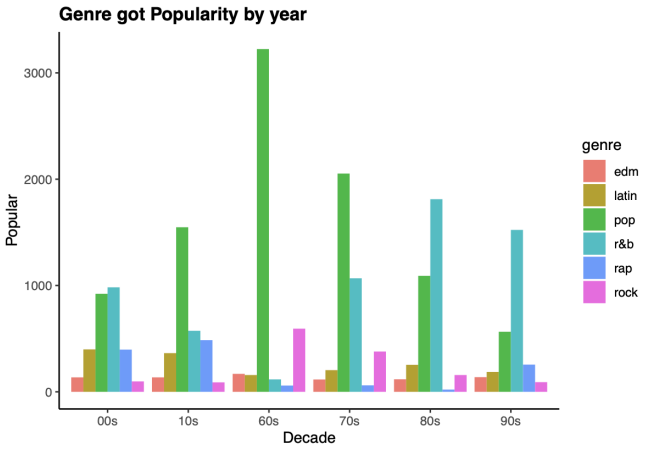
We found the outlier in our release_year feature, as we wanted to predict the year using different song feature, we removed these outliers.



Found the correlation focusing release_year and with other variables, how much correlated are they to the year, as we wanted to predict the year.



There is another question that we try to answer it by analyzing the dataset, is what are the popular genre in each decade?, as you can clearly see that in 60s pop was famous among all the different genres, and it varies with each decade though. To answer this , we took the dataset -3 , that was having all the three features like genre, popularity, and decade.



This plot answer our question of Popular genre by year This are the genre that were famous in each decade, - the most popular genre is pop in 1960s - in 2000s r&b genre was famous - in 2010s pop was famous - in 1970s again pop was famous - in 1980s r&b was famous - 1990s r&b was famous

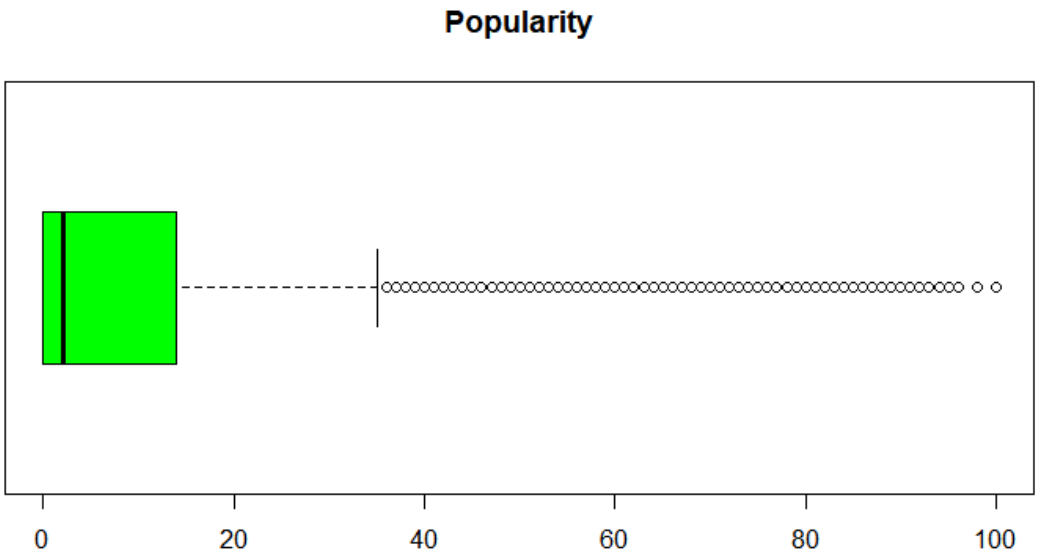
Who are the popular artists

We also found a list of artists that are famous based on the popularity. This was done by analyzing the popularity feature. We made sure that the data being analyzed does not have any missing data. For our analysis, artists with popularity greater than or equal to 90 were determined as famous artists. Out of 1,104,349 artists, only 51 artists had popularity greater than or equal to 90. The top 10 artists are as follows (sorted by popularity):

Name	Popularity
Justin Bieber	100
Bad Bunny	98
Taylor Swift	98
Drake	98
Juice WRLD	96
The Weeknd	96
BTS	96
Myke Towers	95
Ariana Grande	95

We also ensured that the list generated for the famous artists does not have any duplicate artist name by using the unique() function.

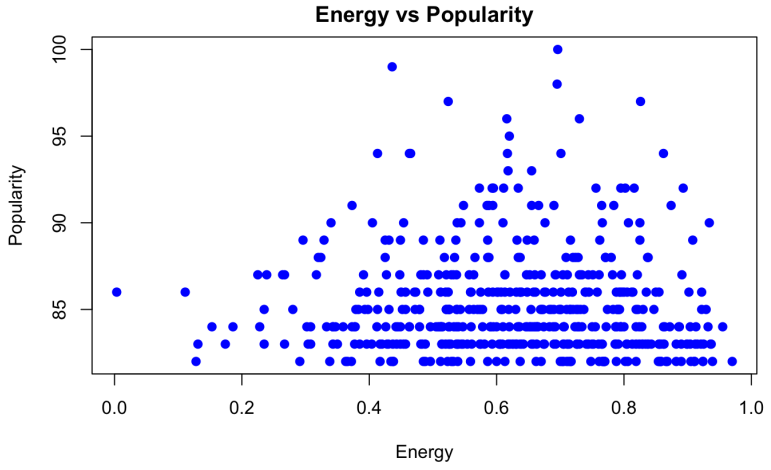
While analyzing this, we also noticed that most of the artists popularity was between 0 to 15. This shows that most artists dont become famous. The below plot shows the box and whiskers plot for the popularity for each artists:



What features affect the popularity of a song the most?

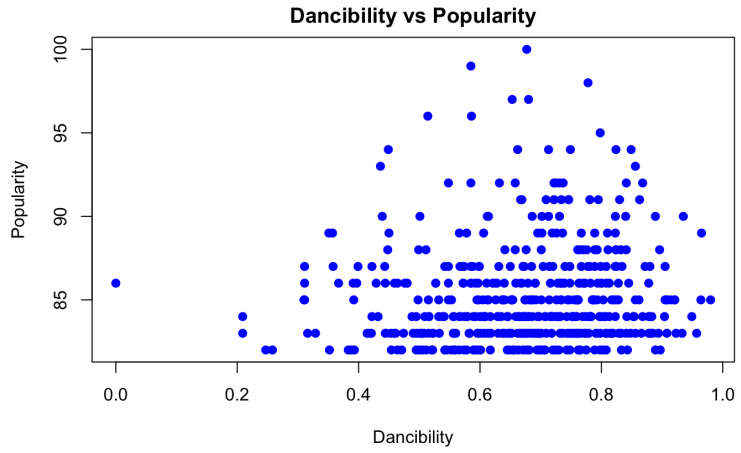
To answer this question, we analyzed different audio features in the dataset against the popularity variable and the findings are as follows:

Energy ranges from 0 to 1 and it shows the measure of intensity in the music. Usually you energetic songs feel loud and fast. Like metal music would have high energy and Bach prelude would have low. Looking at the scatter plot above, we can see that songs having energy level between 0.4 and 0.8 have higher chances of being comparatively popular.

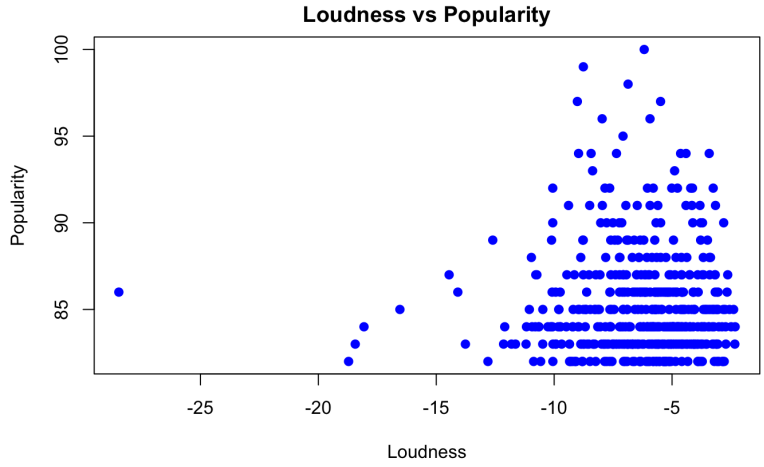


A song's danceability is determined by a number of musical factors, including tempo, rhythm stability, beat intensity, and overall regularity. The range from 0 to 1 represents the degree of danceability. Additionally, this scatter plot demonstrates that not all songs that are danceable are

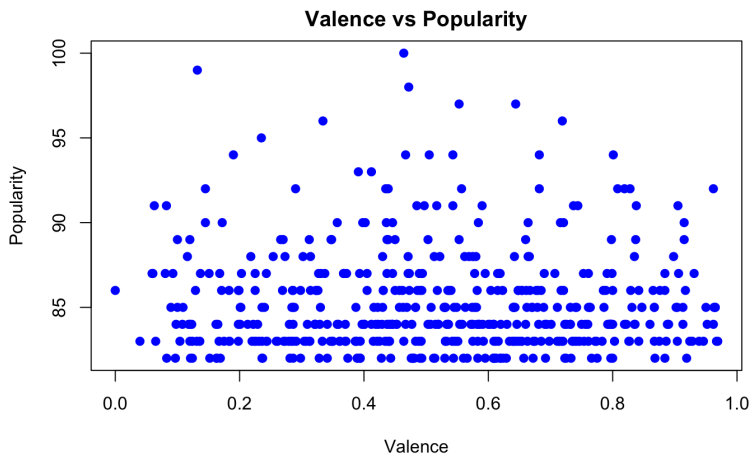
well-liked. Likewise, the most well-liked tracks on Spotify have a danceability rating between 0.5 and 0.85.



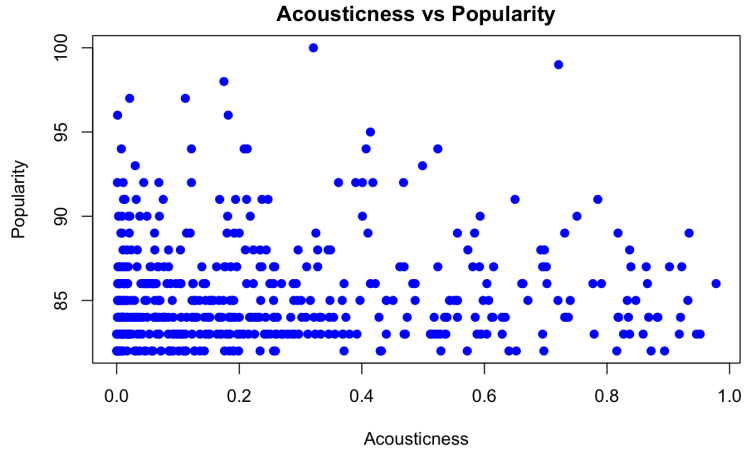
The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing the relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -25 and 0 db. Then the question arrived, which loudness is best for the song? The graph clearly shows that different loudnesses can make a song popular, but other factors must be considered, as some songs with the same loudness as popular songs are not popular.In addition, loudness around -12 to 0 is very common in the Top 500 songs on Spotify. However, loudness between -10 and -5 may result in better popularity of the song.



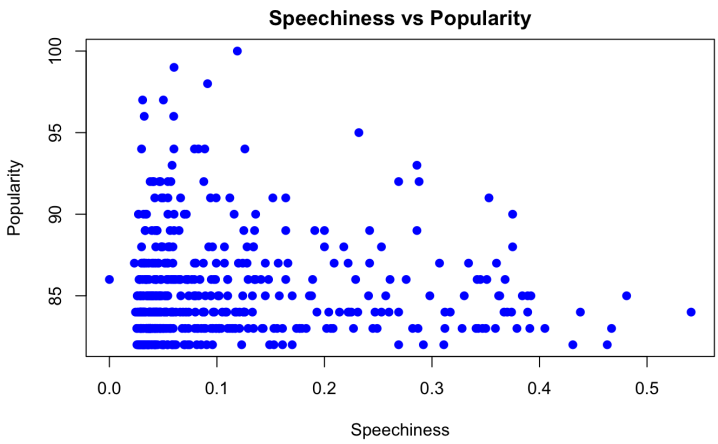
A measure from 0 to 1 describes the musical positiveness conveyed by a track. This scatter plot shows us that the valence doesn't have much effect on the popularity of the song. As you can see, almost all valences can become popular songs, even valence 0 can become popular.



Does acousticness affect the popularity of the song? A confidence measure from 0 to 1 of whether the track is acoustic. 1 represents high confidence the track is acoustic. So, by looking at the graph, do you think having more acoustic or less acoustic is good? The answer is that having acousticness more than 0.6 is not a good idea, but between 0 and 0.6 is a good range of acousticness. But sometimes people also don't like the acousticness around 1 to 0.3 which means acousticness also depends on the genre of the song.



How does speechiness affect the popularity of the song? Speechiness detects the presence of spoken words in a track. Therefore, it is seen that the less spoken words there are, the more popular the song would be. Moreover, having a presence of spoken words makes your song less popular as people usually want to listen to the lyrics of the song, not the spoken words.



In addition, I also found the average values for each audio feature: energy. danceability, loudness, valence, acousticness, speechness. These values could be used to help you create a song that has a higher chance of being featured in the Top 500 Spotify songs in the future.

speechiness	energy	danceability	loudness	acousticness	liveness	tempo
0.1047154	0.6298968	0.6754900	-6.3557380	0.2447356	0.1679984	120.5752480

Modeling

Regression Model - Predicting Year

To predict the year we trained linear regression model, and random forest model, though linear regression model gave us the R-square value on training dataset is 0.7195. We wanted to check it how the feature behaves in random forest, and that will also helps us to improve the model performance significantly.

To train our random forest model, we took few features only, those are “popularity” , “explicit” “danceability” , “energy” , “loudness”, “acousticness” ,“instrumentalness”

```
rf_model = randomForest(release_year ~ popularity +  explicit  + danceability  + energy + loudness  + a
summary(rf_model)
```

Random Forest Model to predict the year using different predictor variable

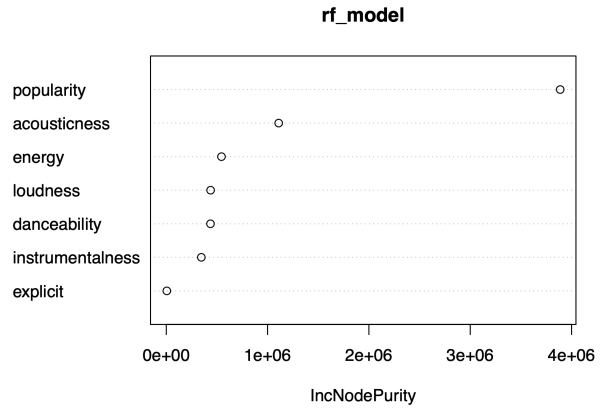
```
##                 Length Class  Mode
## call               3   -none- call
## type               1   -none- character
## predicted      33374   -none- numeric
## mse              500   -none- numeric
## rsq              500   -none- numeric
## oob.times      33374   -none- numeric
## importance         7   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            11   -none- list
## coefs              0   -none- NULL
## y              33374   -none- numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

After training the Random Forest model, we got significant increase in model performance, as you can see that RMSE values of 6.3 and R-square value of 0.81 on test dataset which is much better.

```
postResample(predicted_val, track_spotify_filter_test$release_year)
```

```
##      RMSE  Rsquared       MAE
## 6.3069904 0.8107416 4.8173554
```

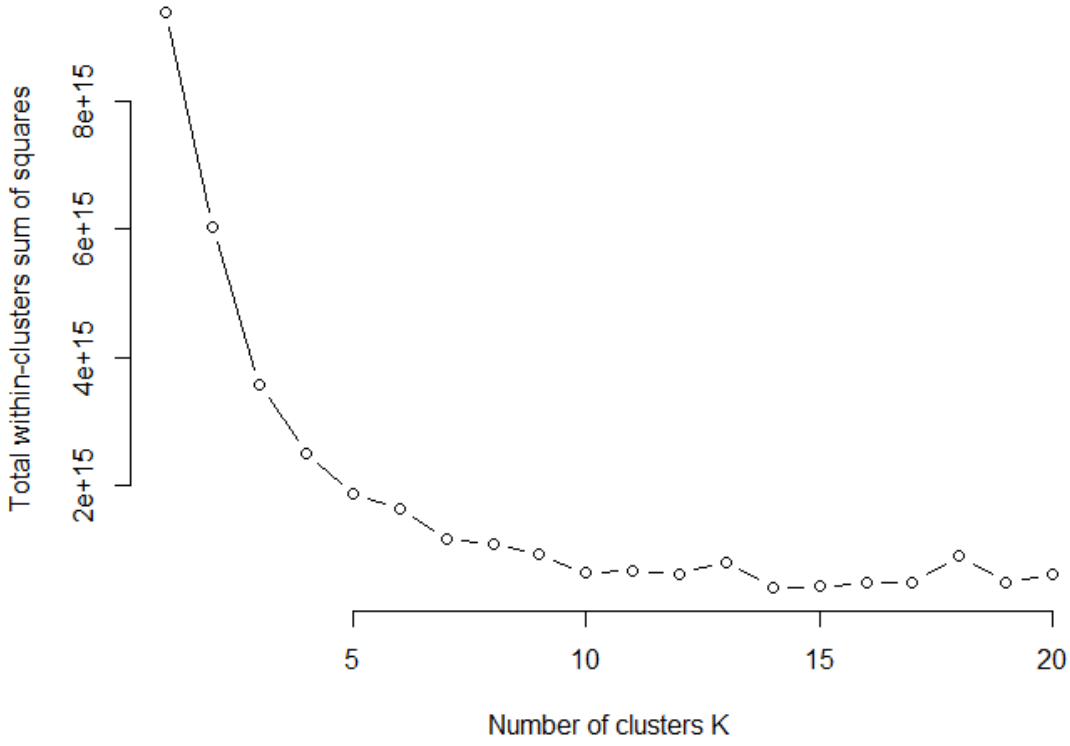
And through random forest model, we plot the Important feature also.



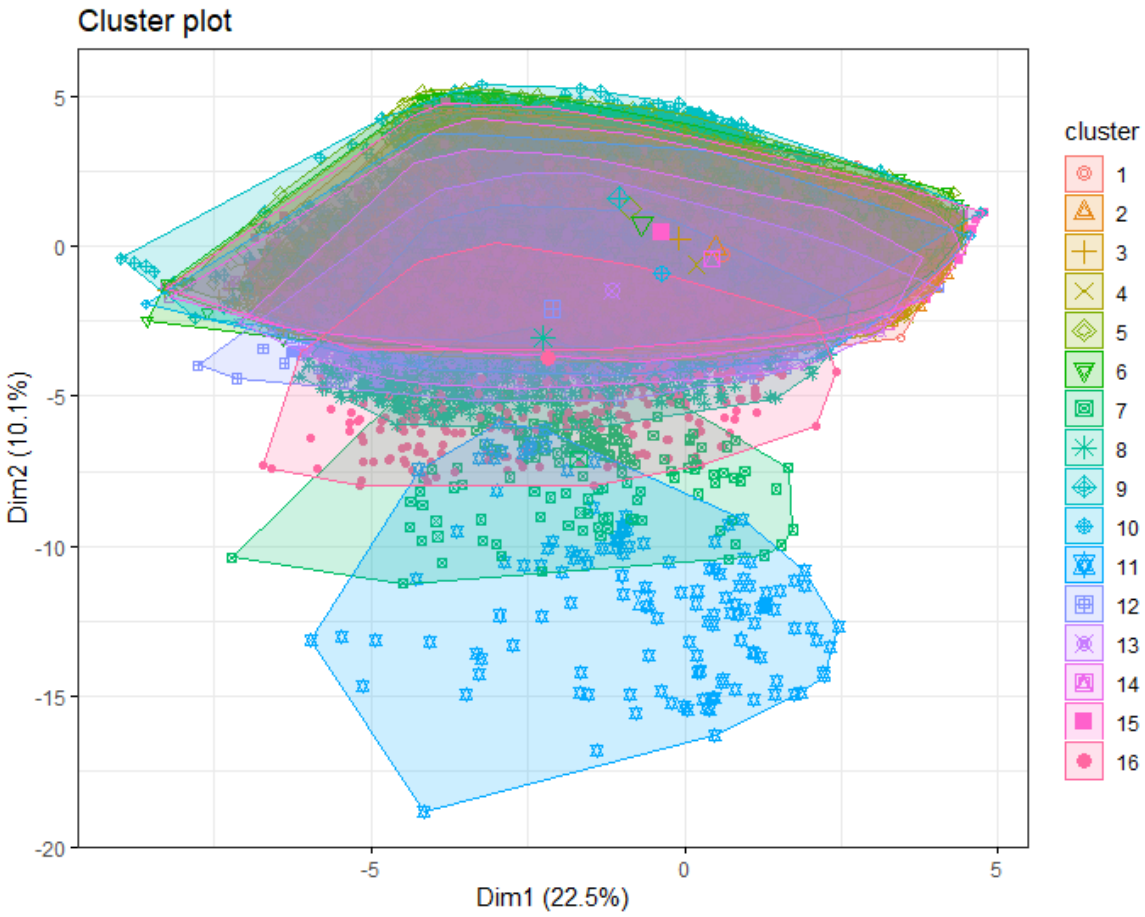
Unsupervised Learning Model - Clustering

In order to provide music recommendations, we used unsupervised learning to group music into clusters. If an user like a certain song in one cluster, we can offer a few songs from the same cluster to fulfill the goal.

We specifically focused on KMeans for clustering. In order to use KMeans to its fullest, we need to first find the number of clusters to use. To determine K, we used the elbow method below.



The idea of the elbow method is basically finding the tradeoff point where adding one more cluster doesn't drop the error in each cluster by much. In the graph above, we can argue that the first elbow exist where K=5. I eliminated the first elbow immediately due to two reasons. For one, there are over 580,000 music in the dataset. I don't see a point to cluster on average 100,000 music per cluster. The range is too broad and music recommendation becomes obsolete. Second, there is a good amount of decrease in error existing at K=7, making K=5 not as effective. Due to the above reasons, I chose the value of K to be 14 in the end as there is a clear elbow and further increase in K doesn't appear to decrease the error as much.




```
K-means clustering with 16 clusters of sizes 81847, 85035, 87649, 56033, 23778, 44445, 171, 1219, 19015, 26061, 137, 3130, 9983, 79187, 68487, 495

Cluster means:
   popularity duration_ms    explicit release_year danceability    energy      key   loudness      mode speechiness acousticness
1    32.22645    237956.37 0.049824673     1996.311    0.5750289 0.5996390 5.296749  -8.982342 0.6394370  0.07100538    0.3521589
2    30.89274    212994.12 0.054824484     1993.782    0.5789707 0.5866399 5.270547  -9.052985 0.6452755  0.07600668    0.3895979
3    25.98465    189881.20 0.050337140     1984.627    0.5729548 0.5254667 5.229153  -9.831471 0.6684845  0.08269725    0.4990758
4    30.27223    314566.68 0.036478504     1992.414    0.5422994 0.5680553 5.236878  -9.960248 0.6295040  0.07414656    0.3824094
5    21.87598    100475.99 0.029523089     1983.712    0.5759877 0.4587286 5.038439 -13.177438 0.7043486  0.33766294    0.5867755
6    20.89524    137895.85 0.047834402     1977.168    0.5690636 0.4690965 5.154101 -11.841954 0.7209135  0.14094184    0.5720546
7    16.89474   2548195.20 0.023391813     1974.789    0.4115433 0.4151494 5.754386 -12.508228 0.6842105  0.17604620    0.7721300
8    15.47416   1086207.40 0.002461034     1974.569    0.3959987 0.3522698 4.983593 -15.121600 0.6915505  0.19264405    0.6831410
9    21.45412     61243.17 0.014672627     1984.656    0.5825062 0.4537379 5.081515 -14.005755 0.6607415  0.43829740    0.5886677
10   26.05134    391295.80 0.022639193     1987.478    0.5132863 0.5373760 5.241587 -11.296392 0.6121024  0.08456393    0.4168347
11   14.08029   3907040.23 0.094890511     1996.511    0.4456204 0.5985854 5.474453 -10.614927 0.6569343  0.20041679    0.4574408
12   17.78371    734618.47 0.014376997     1974.677    0.3984338 0.3636414 5.027796 -15.010009 0.6440895  0.15250383    0.6544433
13   21.72363    517211.87 0.015526395     1982.263    0.4601697 0.4689608 5.188520 -13.092258 0.6374837  0.09768823    0.4970759
14   31.29010    269078.79 0.041509339     1994.849    0.5603105 0.5845625 5.261659  -9.310980 0.6416584  0.06869300    0.3677736
15   23.64437    165984.50 0.050476733     1979.965    0.5730739 0.4948389 5.159315 -10.476514 0.6939273  0.08231128    0.5527342
16   14.46667   1605051.69 0.014141414     1975.006    0.4615178 0.3559981 4.945455 -15.978358 0.7171717  0.38335172    0.7032123
   instrumentalness  liveness   valence    tempo time_signature
1        0.06295684 0.2003391 0.5431270 120.4146       3.922050
2        0.07602213 0.2015114 0.5696311 120.0022       3.902546
3        0.11043913 0.2034006 0.5896351 118.8444       3.872206
4        0.10887181 0.2130552 0.4899392 118.8183       3.903682
5        0.16002514 0.2932147 0.5695363 112.0623       3.733703
6        0.13370471 0.2089498 0.6227911 118.2603       3.822927
7        0.15140579 0.5635094 0.3769781 112.9921       3.836257
8        0.37843651 0.2467076 0.3131259 106.4945       3.803117
9        0.20554330 0.3305732 0.5528852 108.3284       3.645490
10       0.22178492 0.2298605 0.4511800 117.3719       3.881701
11       0.27924179 0.3512547 0.3059403 114.6459       3.729927
12       0.38423149 0.2548768 0.3240754 109.1514       3.798403
13       0.34942662 0.2535959 0.3765908 114.7095       3.847040
14       0.07472143 0.2010370 0.5162060 119.7930       3.915150
15       0.12383187 0.2017110 0.6159729 118.8272       3.858426
16       0.22330769 0.3378261 0.3441975 102.0657       3.769697
```

Note: in clustering, I changed “release_year” to just the year (without month,day) and numeric to show the transition in time as a scalar value.

Now we try to make sense of the clusters. One thing I noticed right away is that the release year seems to have little effect on the KMeans model. There is not a single cluster having extreme values in release year and the averages all falls in the middle 2 quadrants. This is the same case for “time_signature”, “tempo”, “mode”, and “key”. Perhaps if we check for feature importance we would find those 5 columns to have the least importance.

Cluster 11 has the least number of music in it. After analyzing the cluster, we realized that this is the cluster with the LEAST average popularity, LONGEST average duration, and HIGHEST explicitness. It also has the lowest valence and second highest energy. This is clearly a cluster for some specific type of music that only appeals to a small group of people and thus having the least number of music make sense in this scenario.

Cluster 7 has the second to least number of music. It has an extremely high value for liveness, ahead of others by a big margin. Also almost all of its features fall on the far ends of the spectrum. This would also explain why cluster 7 is such a small group.

Clusters 1, 2, and 3 have the most amount of music in them with cluster 3 having visibly higher instrumentalness. Cluster 9 has the most speechiness. Cluster 8 has almost no explicitness and really high instrumentalness and it makes me think of classical music. All clusters have their unique qualities thus the model is a great success.

Classification Model - Predicting Popularity

Predicting music popularity only requires the numerical columns so I dropped the categorical columns right from the start. For this question I tuned 3 different types of model thoroughly, the glm, LogitBoost, and treebag. Also since this model is going to be classification, I changed the existing popularity column with a scalar variable to a categorical column with over 41 popularity to be popular and less than or equal to 41 to be not popular. I chose the 41 cutoff as it is the cutoff for top 25%. One in four music are popular make sense to me.

After training and tuning each model and apply the model to the testing set, LogitBoost appears to be the clear winner with a testing accuracy of 86.75%.

```
> confusionMatrix(lbPred,y_test)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 54202  4455
         1  5118  8459

               Accuracy : 0.8675
                 95% CI : (0.865, 0.8699)
    No Information Rate : 0.8212
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5576

 Mcnemar's Test P-Value : 1.324e-11

            Sensitivity : 0.9137
            Specificity : 0.6550
         Pos Pred Value : 0.9240
         Neg Pred Value : 0.6230
             Prevalence : 0.8212
         Detection Rate : 0.7504
   Detection Prevalence : 0.8120
      Balanced Accuracy : 0.7844

       'Positive' Class : 0
```

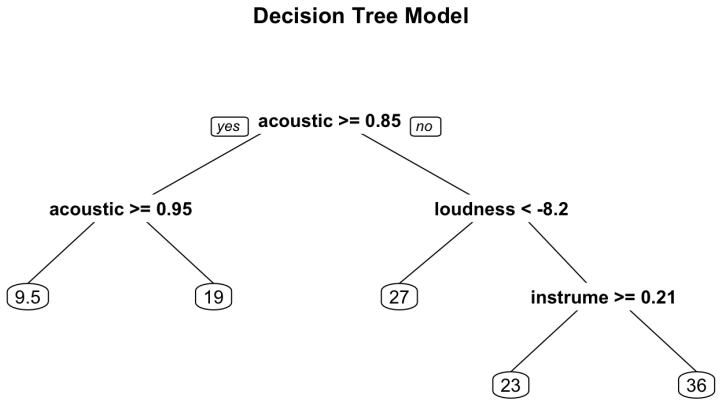
Regression Model - Predicting Score of Popularity to check for successful artist

For this task we only used the continuous audio features namely speechiness, energy, danceability, loudness, acousticness, liveness, tempo, valence and instrumentalness to predict a continuous target variable popularity to check what level of popularity would a song and an artist get based on the aforementioned audio features.

First I tried the multiple linear regression model by the adjusted r-squared value for the model was quite low i.e 0.2159.

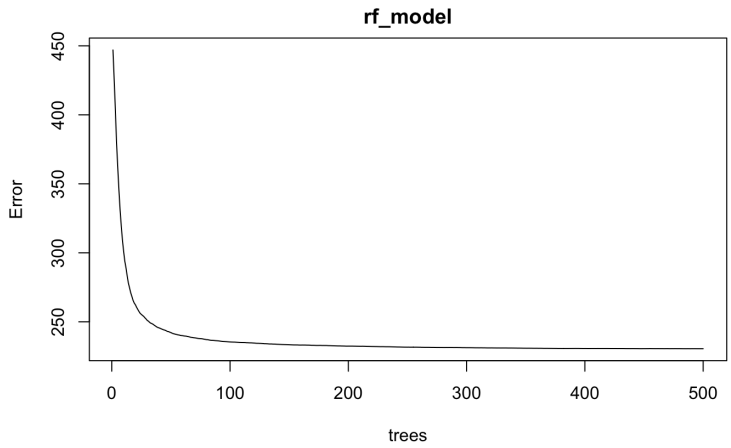
Next I tried the decision tree model for the task and got the following metrics:

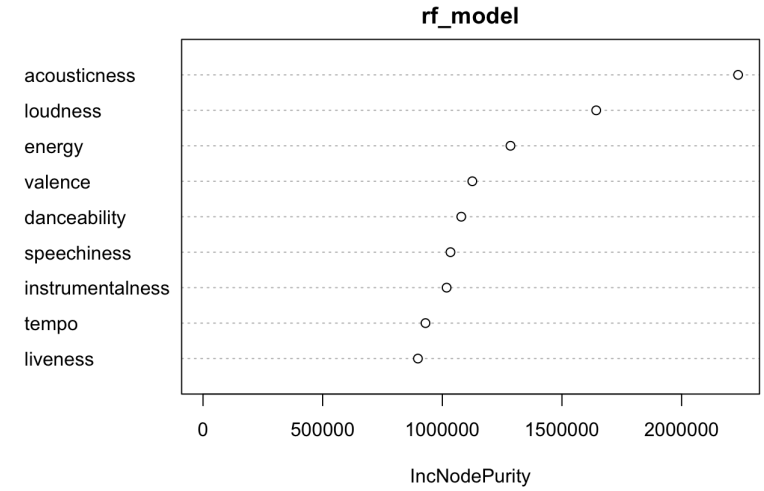
1. MSE on Test Dataset: 274.7365
2. Variance on Test Dataset: 339:3172
3. R-Square on Test Dataset: 0.1903257



As you can see from the evaluation, the decision tree did not perform well either. We tried using the random forest model for the task at hand. The evaluation metrics were as follows:

1. MSE on Test Dataset: 235.6079
2. Variance on Test Dataset: 339.3172
3. R-Square on Test Dataset: 0.3056412



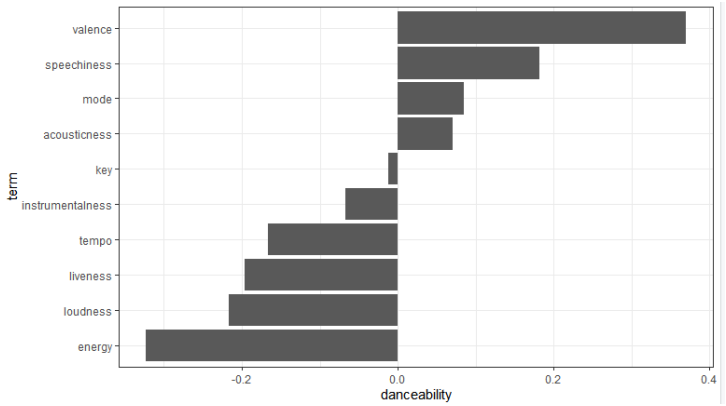


Regression - Predict Danceability based on different features

In this analysis, we predict the danceability of a song based on the different features such as energy, key, loudness, mode, speechiness, acousticness , instrumentalness, liveness , valence, tempo. A correlation matrix was plotted between the features to see if there was any relationship between the different features. The matrix we received was as follows:

term	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
danceability	NA	-0.323	-0.0127	-0.217	0.0844	0.182	0.0699	-0.0671	-0.197	0.370	-0.166
energy	-0.323	NA	0.0447	0.602	-0.0322	-0.148	-0.497	0.304	0.232	-0.0135	-0.0245
key	-0.0127	0.0447	NA	-0.00681	-0.248	-0.0306	-0.00494	0.0683	0.00275	0.0286	-0.00986
loudness	-0.217	0.602	-0.00681	NA	-0.00474	0.0480	-0.284	-0.188	0.167	0.0809	0.152
mode	0.0844	-0.0322	-0.248	-0.00474	NA	0.0503	-0.0145	-0.0164	0.00728	0.0222	-0.0122
speechiness	0.182	-0.148	-0.0306	0.0480	0.0503	NA	0.160	-0.387	0.0574	0.219	0.165
acousticness	0.0699	-0.497	-0.00494	-0.284	-0.0145	0.160	NA	-0.262	-0.107	0.0993	0.0556
instrumental~	-0.0671	0.304	0.0683	-0.188	-0.0164	-0.387	-0.262	NA	-0.0160	-0.257	-0.208
liveness	-0.197	0.232	0.00275	0.167	0.00728	0.0574	-0.107	-0.0160	NA	-0.0252	0.0283
valence	0.370	-0.0135	0.0286	0.0809	0.0222	0.219	0.0993	-0.257	-0.0252	NA	0.0584
tempo	-0.166	-0.0245	-0.00986	0.152	-0.0122	0.165	0.0556	-0.208	0.0283	0.0584	NA

This shows that the correlation relationship between the features is very less. We also plotted the correlation plot between the features while keeping the focus term as danceability. This plot is shown below:



The above plot shows that key, instrumentalness, acousticness and mode are close to zero. Therefore, we can remove them from our analysis. The data was then divided into test and train data. We applied different sets of model to analyze the data such as Multiple Linear Regression (MLR), MLR with scaling on specific features, Logistic Regression, Random Forest. All the models except for Random Forest, gave a R-squared value for the test data between 0.05 to 0.35. With random forest, we received a R-squred value of 0.6. Therefore we chose that as our model for this analysis. Below are the values we received for the R-square for each model:

Sr. No.	**Model Type**	**R-squared value**
1.	Multiple Linear Regression	0.3045
2.	Multiple Linear Regression (with scaling)	0.3045
3.	Logistic Regression	0.072
4.	Random Forest	0.593

Conclusion

Since we have 8 research questions, we will conclude each one separately.

1. To predict the danceability of a song based on different features, data was analyzed with different models such as Multiple Linear Regression (MLR), MLR with scaling on specific features, Logistic Regression, Random Forest. The best model that was from the Ranom Forest with R-squared value of 0.593, which is a reasonable number to predict the danceability. Therefore, we can conclude that we can predict the danceability of a song using the Random Forest model based on the different features like tempo, energy, valence, speechiness, loudness, liveness and mode.

2. After training the random Forest model, with the R-square value of 0.81, the model looks like good enough to predict the year, from different features, we can give input features and it can predict the year in which the song got release, as of now it's predicting decade

not exactly the year. But if we have more data with specific year, and some more relevant columns like genre, artist names, artist basic information, that will be significant features to predict an year.

3. After the KMeans model separating the dataset into 16 clusters, we get to finally answer the research question. When an user search for one song, we can recommend more song by going to the original song's cluster and pick out some random music from the same cluster. This is an unsupervised learning model since there are no target variables.

4. With the LogitBoost model, the final result had an accuracy of 86.75% and the z-value is infinitely close to 0. That proves the model is significant in determining whether the song is popular or not. For future research, we can try out other models such as SVM. The treebag model took my computer 2 hours to finish and so maybe a better setup can help speed up and performance dramatically.

5. From the Analysis section, we see that pop is famous genre among most of the year, even the pop was in it's peak in 60s, and after r&b comes into the picture that is also popular in some of the decade like 80s and 90s, we can say people before 80s listened pop genre, but after it got decreased and r&b got into the picture.

6. All findings are mentioned in the Data Analysis section with an interpretation followed by a plot.

7. To check which artists were famous, we used the popularity feature and categorized artists as popular if their popularity was more than 90. This analysis showed that out of 1,104,349 artists, only 51 artists had a popularity rating of more than 90.
8. To conclude from our research and trials for this research question, the random forest model did a better job among the three models but it appears to be that audio features alone are not good predictors of the popularity value (0 to 100). It is viable to classify if a song/artist will be popular or not, but to predict the popularity level of the song is a difficult task and it depends on a lot of other features apart from audio features.