

# Spotify Data Analysis

GR5291: Advanced Data Analysis

COLUMBIA  
UNIVERSITY



Group 27

Ray Fan (mf3312)

Yunan Xu (yx2585)

Chengming Xie (cx2234)

Jocelyn Yu (yy3038)

Hongshan Lin (hl3353)

Mengxin Li (ml4395)

Jiadong Wu (jw3856)

Haoyu Zhang (hz2659)

JiaruiYang (jy3036)

# Introduction

In this project, we used data from Spotify to establish multiple models through regression analysis, classification analysis and clustering methods to find the relationship between features and certain characteristics.

In the prediction part, we established OLS, WLS, KNN, decision tree models as regression approaches; Random Forest, XGBoost, Logistic regression as classification approaches. We will explore which method is the most accurate and have the best performance for our analysis.

In the clustering part, we established PCA and K-means cluster methods to group the songs by genres. And we will find out which genre would be the most influential to the popularity of a song.

## Project Objective

This report has two main objectives. On one hand, it aims to predict popularity rating by regression and classification models, and select the best model for prediction. On the other hand, it aims to differentiate genres by clustering.

## Data Source & Description

### Section 1: Data Introduction

Our data is from Kaggle [1]. According to the data introduction, the data was collected from Spotify Web API, also known as Spotify for Developers. Every track on Spotify has an Spotify ID. By inputting uniquely identified tracks from Spotify, two programs [2, 3] on Spotify for Developers automatically obtain basic information of the tracks, such as *artists* and *released date*, and generate various audio features.

The only variable that was generated by neither of the programs is *year*. We assumed that it is because the quality of *release date* is not good enough, where plenty of the values are missing months or dates. However, all of them have years. Even though Ay, the author of the data set on Kaggle, did not specify, we think that *year* is simply derived from the *release*

*date*. The low-quality of *release date* is also the reason why this variable is eliminated in data processing.

According to the feature description, “popularity is calculated by algorithm and is based, in most parts, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past” [3]. It indicates that *popularity* and *year* is probably highly correlated. This correlation is explored in EDA in Appendix, and each model deals with it specifically if needed.

## Data Processing

Four variables, *id*, *name*, *artists*, *release date*, are eliminated since they are either identifiers that should not be included in the models (*id*, *name*, *artists*), or of low quality (*release date*). There are no *NA* values and we eliminate duplicates in advance. Tracks prior to 1960 are also eliminated because of the mixed distribution problem. It will be further discussed in EDA in Appendix.

According to the feature description [3], *instrumentalness* greater than 0.5 indicates that it is an instrumental track, and a vocal track otherwise. Similarly, *liveness* greater than 0.8 indicates a live track, and not a live track otherwise. Hence, these two variables are converted from continuous to categorical variables.

After data processing, there are 15 variables and 142,552 unique tracks in the dataset.

## Methods of Analysis

### Section 2: Popularity Rate Prediction: Regression

#### Section 2.1: Ordinary Least Squares:

##### Section 2.1.1: Collinearity:

We want to check if there exists substantial collinearity between any regressors. If strong collinearity exists, then it would create redundancy in our model as they would be the same thing presented in a different manner, and we would need to decide whether or not we should

eliminate any regressors or use other regression methods such as ridge regression to avoid that problem.

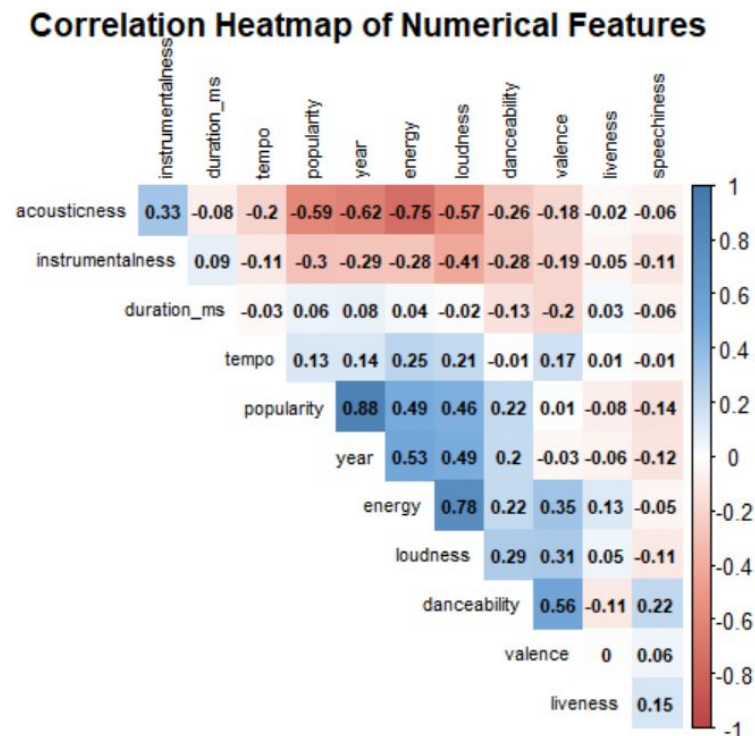


Figure 2.1.1 Correlation Heatmap

Correlation heatmap is an especially useful tool to examine marginal associations between pairs of variables. A brief glance through the above plot, energy and loudness are highly correlated and might cause multicollinearity problems during the model building process, so we need to take a look at the scatterplot of those two variables (figure 1).

To make a better-informed decision, we decide to use Variance Inflation Factor(VIF) as a criterion for high collinearity. VIF measures how much the variance of the estimated regression coefficient is inflated by collinearity. Generally, a VIF greater than 2 indicates significant collinearity. Below is a table showing the VIFs of the different variables.

acousticness	danceability	duration_ms	energy
2.3	1.9	1.1	4.1
explicit	instrumentalness	key	liveness
1.4	1.2	1.0	1.0
loudness	mode	speechiness	tempo
3.1	1.0	1.3	1.1
valence	year		
2.0	1.5		

Figure 2.1.2 VIF

Looking through the VIF table, the score of energy is 4.1 and the score of loudness is 3.1, which should be considered as significant collinearity. By nature, higher energy always

indicates higher loudness. Therefore, keeping those two variables would create redundancy in our model, and we decided to drop the energy variable because the information we lose by dropping the energy variable is not significant according to the r-squared score. After dropping the energy variable, none of the other variables have strong collinearity.

acousticness	danceability	duration_ms	explicit
1.6	1.7	1.1	1.4
instrumentalness	key	liveness	loudness
1.2	1.0	1.0	1.9
mode	speechiness	tempo	valence
1.0	1.3	1.1	1.8
year			
1.5			

*Figure 2.1.3 VIF Adjusted*

### **Section 2.1.2: Full Model:**

According to the summary(Figure 2.1.2) of the full model, we can see that about 55% of the variance in popularity is explained by the full model. Note that X variables such as acousticness ,duration\_ms and tempo are considered non-significant since they have p-values larger than 0.05.

The summary of the full model:

```

Call:
lm(formula = popularity ~ . - energy, data = train)

Residuals:
    Min       1Q   Median       3Q      Max
-63.3   -7.3   -1.6    5.8   54.1

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.2e+03    4.6e+00  -254.6  <2e-16 ***
acousticness   -2.8e-01    1.3e-01   -2.2    0.03  *
danceability    4.7e+00    2.5e-01   18.9  <2e-16 ***
duration_ms   -2.7e-07    3.2e-07   -0.8    0.40
explicit1      1.5e+00    1.3e-01   11.9  <2e-16 ***
instrumentalness1 -1.6e+00    1.2e-01  -13.3  <2e-16 ***
key           -1.9e-02    9.4e-03   -2.1    0.04  *
liveness1     -2.7e+00    2.2e-01  -12.1  <2e-16 ***
loudness       9.3e-02    8.8e-03   10.6  <2e-16 ***
mode1         -5.1e-01    7.4e-02   -7.0    3e-12 ***
speechiness    -8.0e+00    4.0e-01  -19.9  <2e-16 ***
tempo          1.2e-03    1.1e-03    1.1    0.28
valence        -1.8e+00    1.7e-01  -10.5  <2e-16 ***
year           6.1e-01    2.3e-03   265.7  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10 on 97262 degrees of freedom
Multiple R-squared:  0.55,    Adjusted R-squared:  0.55
F-statistic: 9.1e+03 on 13 and 97262 DF,  p-value: <2e-16

```

*Figure 2.1.2 Summary of the Full Model*

### **Section 2.1.3: Model Selection Strategy:**

Our goal is to create a multivariate regression model that best characterizes the relationship between popularity ratings and explanatory variables. Thus, we used model selection methods to help us determine the best combination of the variables. We used BIC, adjusted r-squared, and Cp score as matrices to select explanatory variables (figure 2).

A good model turns to have Higher Adjusted r-squared, lower BIC, and Cp score  $\approx$  number of explanatory variables in the model + 1. From this plot, all three methods indicate that the reduced model should include 9 explanatory variables: danceability, explicit, instrumentalness, liveness, loudness, mode, speechiness, valence, and year.

Summary of the reduced model:

```

Call:
lm(formula = popularity ~ danceability + explicit + instrumentalness +
    liveness + loudness + mode + speechiness + valence + year,
    data = train)

Residuals:
    Min       1Q   Median       3Q      Max
   -63.2    -7.3    -1.6     5.8    54.1

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.2e+03   4.5e+00  -258.5  <2e-16 ***
danceability    4.7e+00   2.4e-01   19.2  <2e-16 ***
explicit1      1.6e+00   1.3e-01   12.0  <2e-16 ***
instrumentalness1 -1.6e+00  1.2e-01  -13.4  <2e-16 ***
liveness1     -2.7e+00   2.2e-01  -12.2  <2e-16 ***
loudness       1.0e-01   7.9e-03   13.0  <2e-16 ***
mode1         -5.0e-01   7.3e-02   -6.8   9e-12 ***
speechiness    -8.0e+00   4.0e-01  -20.0  <2e-16 ***
valence       -1.7e+00   1.6e-01  -10.3  <2e-16 ***
year           6.1e-01   2.3e-03  269.6  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10 on 97266 degrees of freedom
Multiple R-squared:  0.55,    Adjusted R-squared:  0.55
F-statistic: 1.3e+04 on 9 and 97266 DF,  p-value: <2e-16

```

*Figure 2.1.3 Summary of Reduced Model*

In the reduced model, 55% variance of popularity is explained by explanatory variables. And the adjusted r-squared is 0.55, which is the same as the adjusted R-squared in the full model.

## **Section 2.2: Weighted Least Squares:**

Looking at the reduced model above, the r squared value is relatively low, so we decided to take a look at weighted least squares models. Notice that most of the dependent variable variation is captured by the variable Year, since songs from recent years turn out to be more popular than songs in the past. And overall, the variance of popularity decreases as the year increases. Therefore, we choose weight by the inverse of the variance of residuals of the full model by year.

Summary of the weighted least squares model:



```

Call:
lm(formula = popularity ~ danceability + explicit + instrumentalness +
    liveness + loudness + mode + speechiness + valence + year,
    data = train, weights = 1/sg2)

Weighted Residuals:
    Min       1Q   Median       3Q      Max
-3.13  -0.77  -0.17   0.60   5.07

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.2e+03    4.2e+00  -292.4  <2e-16 ***
danceability    5.1e+00    2.3e-01   22.5  <2e-16 ***
explicit1      1.2e+00    1.1e-01   11.3  <2e-16 ***
instrumentalness1 -1.4e+00    1.2e-01  -12.0  <2e-16 ***
liveness1      -2.4e+00    2.2e-01  -11.2  <2e-16 ***
loudness       5.5e-02    7.5e-03    7.3   2e-13 ***
mode1         -5.2e-01    6.7e-02   -7.8   8e-15 ***
speechiness    -6.8e+00    3.6e-01  -18.8  <2e-16 ***
valence        -1.8e+00    1.5e-01  -12.0  <2e-16 ***
year           6.4e-01    2.1e-03   305.0  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.1 on 97266 degrees of freedom
Multiple R-squared:  0.6,    Adjusted R-squared:  0.6
F-statistic: 1.6e+04 on 9 and 97266 DF,  p-value: <2e-16

```

*Figure 2.2 Summary of the Weighted Model*

The summary of the weighted model shows that compared with the unweighted model, the adjusted r-squared increases from 0.55 to 0.6, by 5 percent, which is a relatively beneficial result.

### **Section 2.3: KNN Method:**

The k-nearest neighbors (KNN) algorithm is an easy-to-implement supervised machine learning algorithm that can be used to predict the popularity ratings. We first choose k equals to 8 by the lowest RMSE and then train the knn model to do the prediction (figure 3).

### **Section 2.4: Decision Tree Method:**

A decision tree is a tree based algorithm used to solve regression and classification problems. A decision tree uses a tree structure to specify sequences of decisions and consequences. The idea is simple: it breaks down a dataset into smaller subsets. Tree-based methods tend to perform well on unprocessed data (i.e., without normalizing, centering, scaling features). Hence in this project, we build a regression tree to predict the popularity based on the rest of the features.

We first built the tree with all numeric features, including year, and achieved RMSE 9.54 test RMSE. The picture below shows how the tree split and the decision made. It shows that year



contributes most: older year, lower popularity, and vice versa. Although the year dominates here, if we remove the year, RMSE drops significantly. Hence we decided to retain the year.

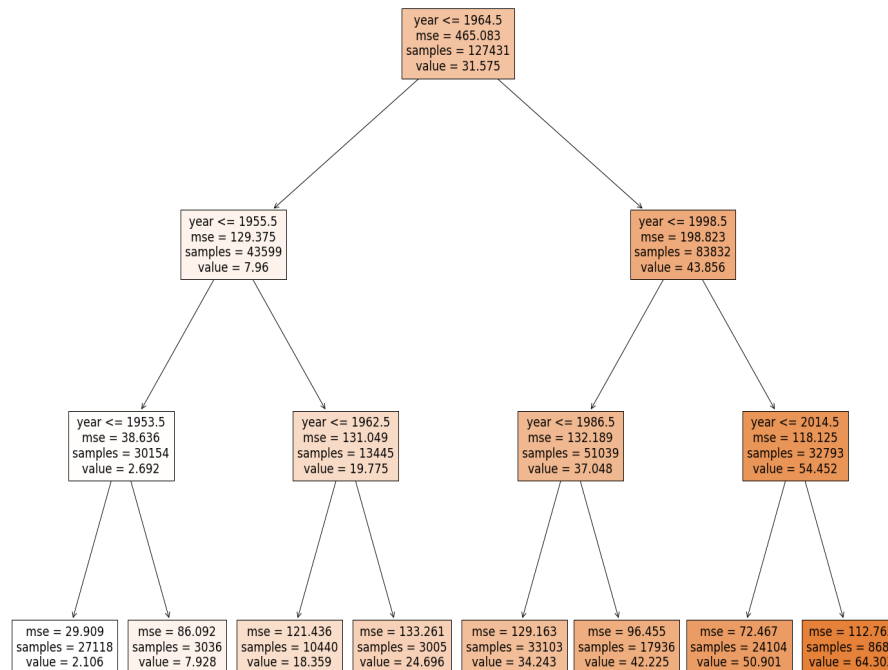


Figure 2.4 Decision Tree Result

## Section 2.5: Regression Results and Model Comparison:

	<i>Train</i>	<i>Test</i>
<i>Decision (Regression) Tree</i>	9.48	9.54
<i>OLS</i>	10.230	10.274
<i>WLS</i>	10.247	10.293
<i>KNN</i>	11.121	12.641

Table 2.5 Comparison Between Models

In terms of test RMSE, Decision Tree is the best method. the OLS and WLS perform better than KNN. And from the model perspective, WLS has 5% higher adjusted r-squared than OLS.

## Section 3: Popularity Rate Prediction - Classification

After predicting popularity rate using regression and tree-based models, we reframe our objective as a classification problem. Under the classification criterion, we aim to predict whether a track would be popular or not, given the music and other characteristics of that specific track.

We split our continuous variable, *popularity*, into 2 and 5 categories, which are then followed by predictive modelling respectively.

### Section 3.1. Binary Classification

#### Section 3.1.1 Random Forest

Regarding the binary classification problem, we split our target variable into two categories in which class label == 0 represents low popularity class including *popularity* less than 50 while class label == 1 represents high popularity class including *popularity* larger than 50.

We first build a random forest model using features kept during the feature engineering stage, and the model accuracy score reaches 0.81 while the AUC reaches 0.85 (Figure 3.0, Left) which is decent enough. Based on the feature importance plot (Figure 3.0, Right), the *Year* feature is the most influential feature in the prediction of popularity class and the *Loudness* feature is the second most important feature in our predictive modeling.

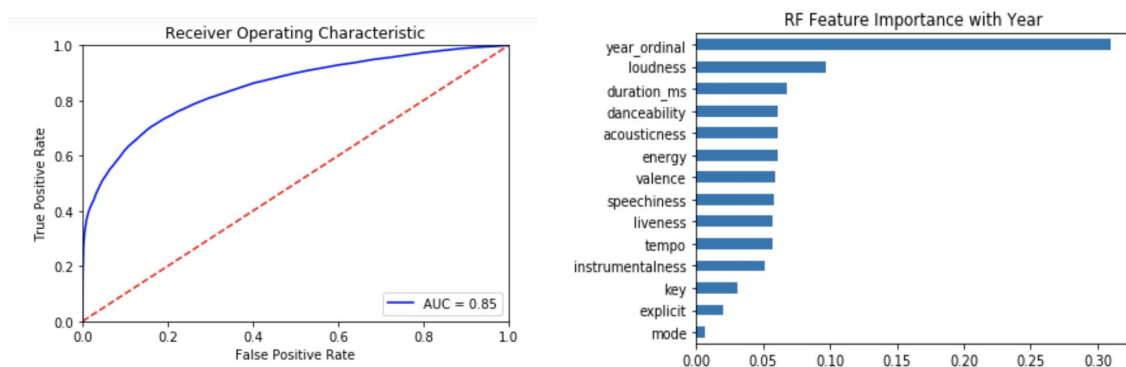


Figure 3.0 Random Forest (Binary)

Two partial dependence plots (Figure 3.1) of the two most important features are drawn to derive an insight into the relationship of each variable to how likely the track is popular. The partial dependence plot below roughly shows an inverse relationship between the *acousticness* and probability of track being popular and a positive relationship between *Year* and the probability of track being popular.

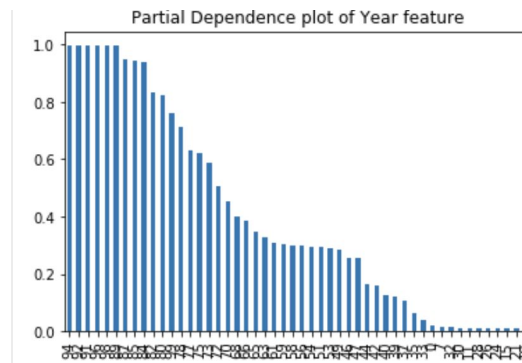


Figure 3.1 Pdp Plot of two Features

(X-axis in the left is roughly decreasing while right is roughly increasing)

## **Section 3.2. Multi-class Classification**

### **Section 3.2.1 Random Forest**

We then split our target variable, *popularity*, into 4 categories with equal intervals of 25. However as a result, the distribution of transformed target variables is highly balanced (Figure 3.2, Left). Thus we fixed the weight of each training data instance in the model objective function by the relative size of the class which it belongs to. The resulting random forest model reaches an accuracy of 0.71. However, the classification report (Figure 3.2, Right) tells us that the model still has a weak predictive power in the imbalance class (i.e. class 3 with highest popularity rate) and f1-score is extremely low as 0.06.

			precision	recall	f1-score	support
1	67441	0	0.67	0.44	0.53	5540
2	34168	1	0.70	0.88	0.78	22223
0	16569	2	0.74	0.52	0.61	11245
3	1478	3	0.48	0.03	0.06	479

Figure 3.2 Random Forest (Multi-class)

### **Section 3.2.2 XGBoost Classifier**

To achieve better accuracy in the highly imbalanced class and control overfitting, we built a XGBoost classifier and alleviated the imbalanced class problem by introducing sample

weights for each training data instance in the softmax loss function. Although the overall accuracy stays at 0.72 which seemingly is not improving a lot, the f1-score of class 3 in the classification report (Figure 3.3) is significantly larger than that of the random forest model (i.e.  $0.21 > 0.06$ ). Besides, the feature importance plot of XGBoost (Figure 3.4) somewhat gives us more information about *explicit* variable which is ranked as the second most important feature while the random forest model does not observe that relationship.

	precision	recall	f1-score	support
0	0.52	0.67	0.59	5540
1	0.75	0.71	0.73	22223
2	0.66	0.63	0.64	11245
3	0.26	0.18	0.21	479

Figure 3.3 XGBoost classification report (Multi-class)



Figure 3.4 Feature Importance Plot of XGBoost

### **Section 3.2.3 Multinomial Logistic Regression**

For the next classification method, we applied Logistic Regression. Same as the EDA part above, we dropped categorical variables ID, Name, Released Date and Artist. Moreover, we dropped data that was released before 1960. The popularity of these data are 0 that will lead to skewed distribution of popularity.

Since popularity spreads over 0 to 100. Same as the XGBoost method above, we separate popularity into different levels. The final step is to separate data into training and test data sets based on proportion.

Firstly, we separate popularity into 2 levels. For popularity less than 50, we classify popularity as 0, and for popularity greater than 50, we classify popularity as 1. The bar-plot of popularity after separation is shown as below (Figure 3.5).

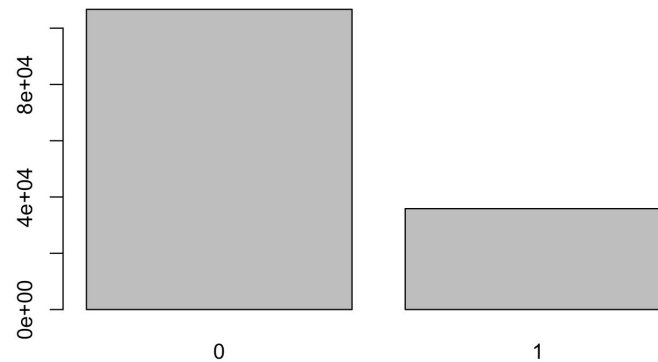


Figure 3.5 Data in Two Levels.

Based on exploratory data analysis, there is a multicollinearity problem in our data. In order to fix this problem, we applied logistic regression with L-1 penalty to process variable selection. Moreover, we also applied 10 fold cross validation to increase the accuracy of the model. The minimum lambda is 0.000514375. After we applied the minimum lambda into the logistic model, the model as well as the selected variables is shown below (Figure 3.7).

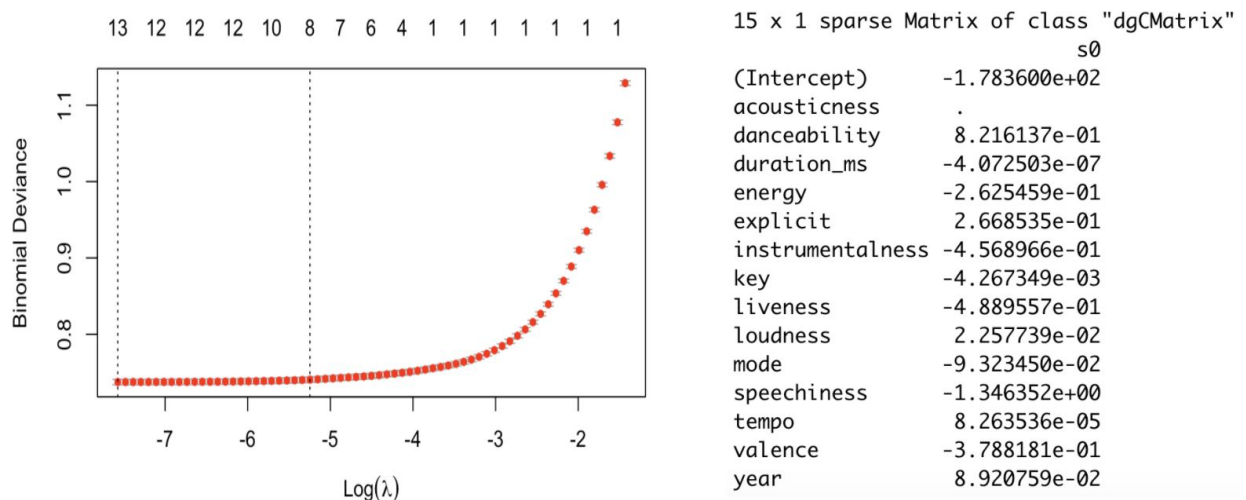
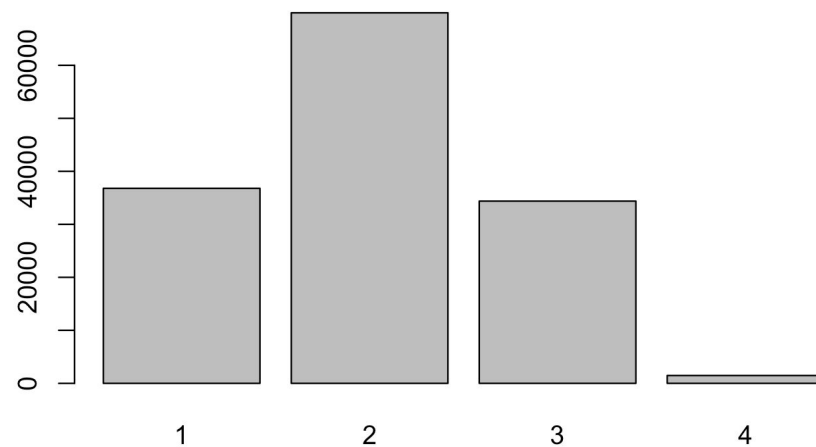


Figure 3.6 Choosing Best Lambda

Figure 3.7 Result of 2-level Logistic Model

In this model, the accuracy rate is 0.8424763 and the misclassification rate is 0.1575237.

For the next part, we separate popularity into 4 levels; popularity between 0-25 are classified as level 1; popularity between 26-50 are classified as level 2; popularity between 51-75 are classified as level 3; and for popularity between 76-100, we are classified as level 4. The bar-plot of popularity after classification is shown as below (Figure 3.7).



*Figure 3.7 Data in Four Levels*

The bar-plot above indicates that level 4 has less data compared to other levels. Same as the procedure we did in the last part, we applied multiple logistic regression with L-1 penalty and 10 fold cross validation. Since the data are separated into 4 levels, multiple logistic regression will classify each level based on simple logistic regression. Hence, we will obtain 4 different variable selections and the output is shown as below (Figure 3.8).

<pre> \$`1` 15 x 1 sparse Matrix of class "dgCMatix"       s0 acousticness 3.214712e+02 danceability 2.549094e-01 duration_ms -6.575294e-01 energy 2.340178e-07 explicit 3.730224e-01 instrumentalness -2.204051e-01 key 4.156323e-01 liveness 1.139290e-03 loudness 7.851744e-01 mode -4.286129e-02 speechiness 1.554504e-01 tempo 1.961726e+00 valence -1.130152e-04 year 3.055616e-01 year -1.752038e-01 </pre>	<pre> \$`2` 15 x 1 sparse Matrix of class "dgCMatix"       s0 acousticness 5.360868e+01 danceability -4.244304e-02 duration_ms -3.795024e-01 energy 4.019969e-07 explicit 1.532935e-01 instrumentalness -1.559627e-01 key 2.093268e-01 liveness 1.152531e-03 loudness 2.510874e-01 mode -1.352624e-02 speechiness 4.864729e-02 tempo 6.102592e-01 valence . year 1.881572e-01 year -3.861685e-02 </pre>
<pre> \$`3` 15 x 1 sparse Matrix of class "dgCMatix"       s0 acousticness -1.006570e+02 danceability 4.244304e-02 duration_ms 3.795024e-01 energy -2.340178e-07 explicit -1.532935e-01 instrumentalness 1.559627e-01 key -2.093268e-01 liveness -4.692912e-03 loudness -2.510874e-01 mode -4.286129e-02 speechiness -4.864729e-02 tempo -6.102592e-01 valence 8.461091e-05 year -1.881572e-01 year 3.861685e-02 </pre>	<pre> \$`4` 15 x 1 sparse Matrix of class "dgCMatix"       s0 acousticness -2.744229e+02 danceability -7.829025e-02 duration_ms 2.108108e+00 energy -2.659960e-06 explicit -8.756527e-01 instrumentalness 1.667175e-01 key -9.142503e-01 liveness -1.139290e-03 loudness -6.099093e-01 mode 7.807772e-02 speechiness -2.091928e-01 tempo -1.126111e+00 valence . year -2.802507e-01 year 1.238652e-01 </pre>

*Figure 3.8 Result of 4-level Logistic Model*

Minimum lambda is 0.0001313332. The accuracy rate is 0.7292178 and misclassification rate is 0.270782. Based on two logistic regression models above, we can see the accuracy rate increases with the decrease of the number of levels. One reason might be that there are only few data that have been classified as level 4 and these data didn't show up in the test data set. Missing data classified as level 4 in the test data set will directly lead to the increase of misclassification rate. The confusion matrix below shows the classification output (Figure 3.9).



	1	2	3
1	5664	1672	18
2	1271	10907	1767
3	117	2586	4219
4	1	31	257

Figure 3.9 Confusion Matrix of four-level Logistic Model

### Section 3.3 Conclusion and Classification Models' Comparison

- **Binary**

Model	Test Accuracy
Multinomial Logistic Regression	<b>0.84</b>
Random Forest	0.81

- **Four-class**

Model	Test Accuracy
Multinomial Logistic Regression	<b>0.729</b>
Random Forest	0.71
XGBoost	0.72

Figure 3.5 Model Comparison (Test accuracy)

The two tree-based classification models find that *Year* is the most important feature in predicting popularity classes while *Acousticness* and *Explicit* are another two important features.

As for the selection of the best model, in the binary classification problem, we can observe that multinomial logistic regression achieves a higher accuracy score than random forest. In the 4-class classification problem however, multinomial logistic regression reaches the highest overall accuracy of 0.729 but XGBoost is much better in the prediction of the highly imbalanced class (i.e. the class 3, *popularity* > 75). The overall accuracy in the highly imbalanced classification problems may no longer be a proper metric to measure the model performance. In conclusion, we choose multinomial logistic regression as our best model in the binary classification problem and XGBoost as our best model in the 4-class classification problem.

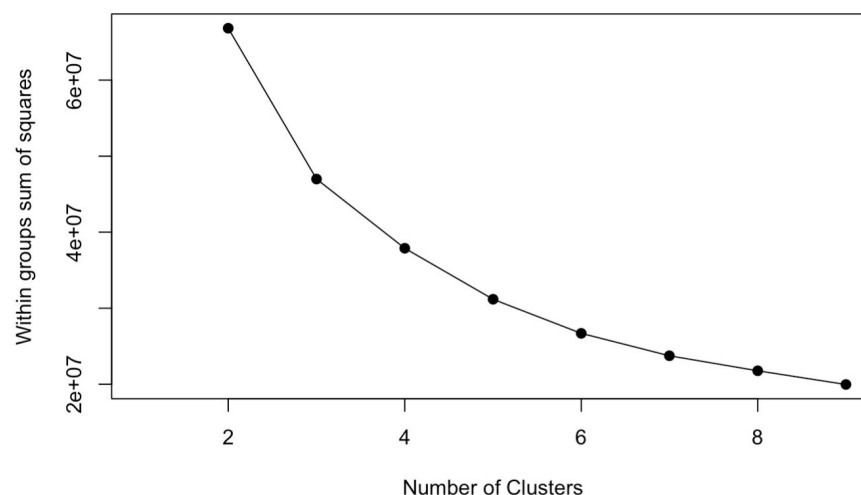
## Section 4: Genres Clustering

In section 4, Principal Component Analysis(PCA) and k-means clusters are applied to classify the genre of Spotify songs. By doing that, each cluster will have different emphases on the audio characteristics. Particularly, with clear classification, it can be determined that how does each audio characteristic influence the popularity of a song,

Firstly, due to the usage of PCA and K-means clustering method, only the numerical variables are applied to cluster the music genres. These numerical variables are: Acousticness, Danceability, Energy, Instrumentalness, Liveness, Loudness, Popularity, Tempo, Valence. The only excluded numerical variable is Speechiness since it is not contained in audio characteristics and irrelevant to music genre classification.

### Section 4.1: K-means Clustering

Clustering methods deal with finding similarities and structure in a collection of data points. In order to identify the characteristics one song might have, K-means clustering method is used. It is a distance-based method consisting of creating clusters so that the total intra-cluster variation is minimized. In k-means clustering, each cluster is represented by its center which corresponds to the mean of points assigned to the cluster. Before applying k-means clustering, it is crucial to identify the number of clusters represented by k. Therefore, the Elbow Method is used to identify k by plotting the explained variation as a function of the number of clusters. Graph 4.1 shows the result.



*Graph 4.1 Elbow Method*

From the graph, it is obtained that the curve becomes smooth after  $k=3$ , therefore,  $k$  is determined as 3 for k-means clustering. That is, the original songs are categorized into three genres.

## **Section 4.2: PCA**

Due to the complexity of the dataset, it is necessary to perform Principal Component Analysis(PCA), a dimensionality reduction technique that allows for optimized visualization of high dimensional data by projecting key variables (components) that contribute to the highest variance, to reduce the dimension and better illustrate the spread of the songs. By checking the importance of components in Table 4.2, it is found that a high dimensional space can be projected on to the first two principal components. That means PC1 and PC2 will be used since they have explained more than 98% of the data.

Importance of components:

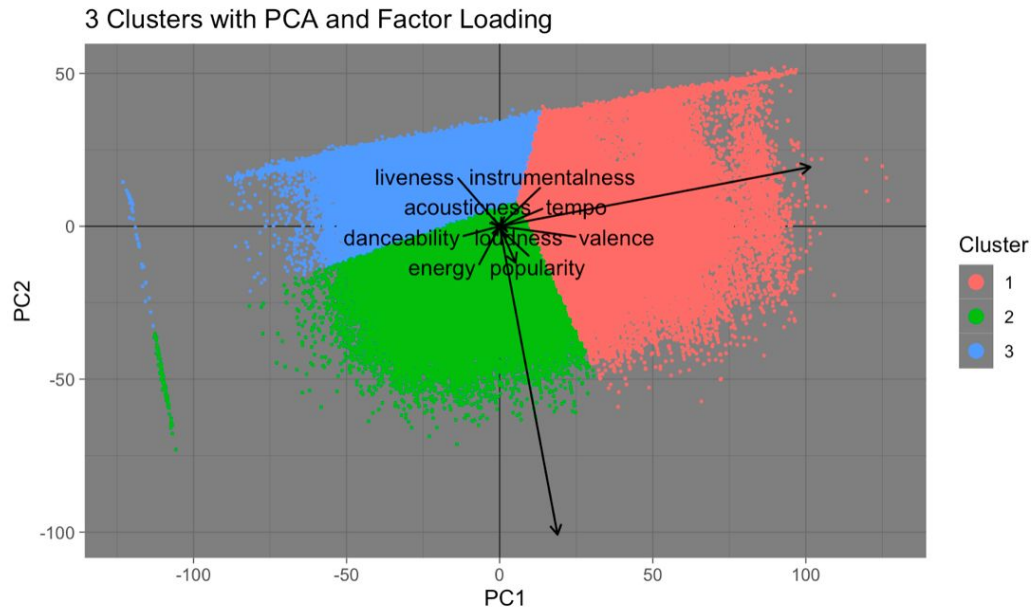
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Standard deviation	31.0275	21.3404	4.9084	0.30763	0.27930	0.25400	0.17841	0.13314	0.10243
Proportion of Variance	0.6674	0.3157	0.0167	0.00007	0.00005	0.00004	0.00002	0.00001	0.00001
Cumulative Proportion	0.6674	0.9831	0.9998	0.99986	0.99991	0.99996	0.99998	0.99999	1.00000

*Table 4.2 Importance of Components*

Afterwards, K-means clustering results will be performed on PC1 and PC2 to better illustrate the audio characteristics of three genres.

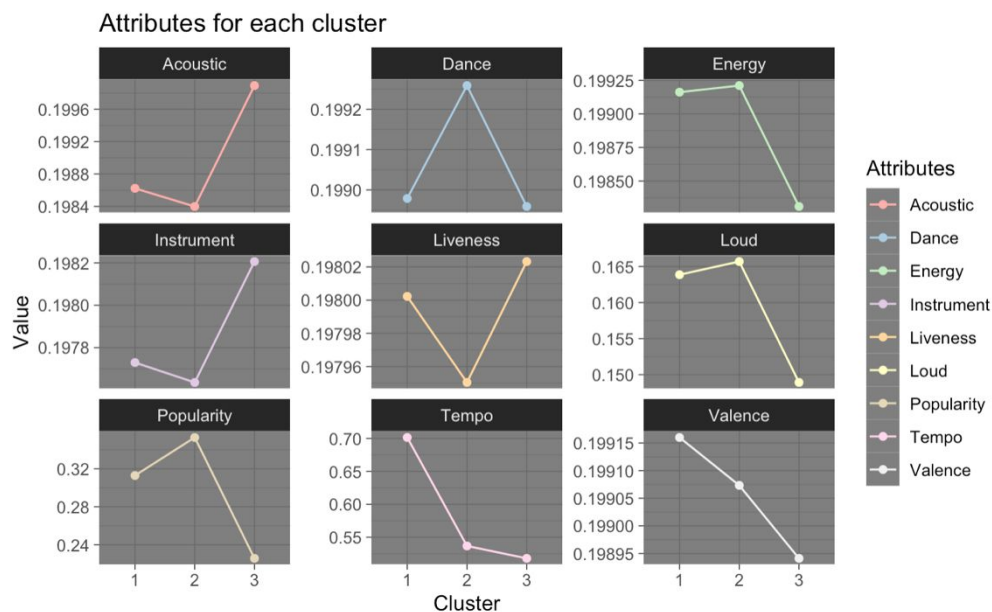
## **Section 4.3: Cluster Attributes**

In this section, a Biplot is generated to explore and visualize the attributes in each cluster. The plot is shown in the graph 4.3.1 below.



Graph 4.3.1 3 Clusters with PCA

Different from the popularity prediction objective, popularity is included as a key feature to indicate other features corresponding to it in this process. From the graph 4.3.1, it can be roughly observed that there are several features like loudness, energy and danceability have the same direction with popularity. However, to better analyze the relationship between popularity and each audio attribute, a graph of each audio feature's mean for three clusters is visualized below.

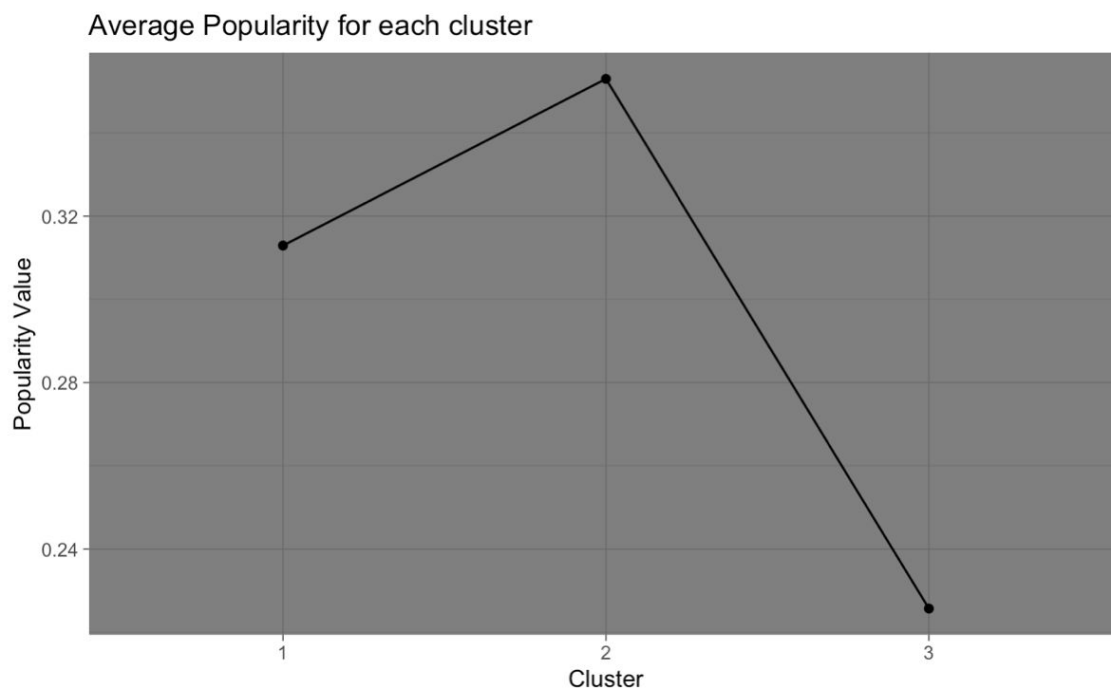


Graph 4.3.2 Attributes for Each Cluster

A conclusion of the main characteristics in the 3 clusters can be drawn by combining these two graphs:

1. Cluster 1 : Tempo, Valence
2. Cluster 2 : Popularity, Danceability, Loudness, Energy
3. Cluster 3 : Acousticness, Liveness, Instrumentalness

To emphasize the cluster result on popularity, the average popularity for each cluster is zoomed up in Graph 4.3.3.



*Graph 4.3.3 Average Popularity for Each Cluster*

#### **Section 4.4: Conclusion of Clustering**

From the result above, it is concluded that Genre 1 with medium rate of popularity has characteristics of valence and tempo, and some energy, such as the song “Good Feeling - Bingo Players Remix” by Fio Rida.

Genre 2 with highest popularity tends to be songs with strong dancing power and high energy and loudness in it, such as the song “Sour Candy” that is made by Lady Gaga and Blackpink.

Genre 3 with lowest popularity, it is more likely to be the livehouse music with more instrumental and acoustic melody on it; the represented song is “Can’t Stop” by Red Chili

Pepper. It is reasonable because live music performances usually happen in acoustic or full-band format.

Based on the audio characteristics of each cluster, songs with higher energy, loudness, and with strong dancing power tends to be more popular.

## Conclusion

In the regression part, we found out that Decision Tree performs the best among all methods with test RMSE equals to 9.54. In the classification part, multinomial Logistic regression performs the best in both binary and multi-level classification methods with test accuracy 0.84 and 0.729 respectively. One interesting result is how the release year of a song plays significant influence in all of our models. That being said, people are intended to listen to songs from the recent period more than songs from the past. That makes sense because more people began getting involved in music during this era and prefer to enjoy top trending music that is released in recent years.

Moreover, in the clustering part, it is noticeable that danceability, energy and loudness are the most influential features of songs regarding popularity. In other words, people nowadays prefer more about songs that have higher danceability, energy and loudness. That also makes sense because R&B/Hip-hop music, which consists of a stylized rhythmic music that commonly accompanies rapping, rhythmic and rhyming speech, became increasingly popular during the past ten years. Due to the rise of streaming platforms such as Spotify, also known as the “Age of Streaming”, R&B/Hip-hop music has an impressive lead over other kinds of music and has spread all over the world. According to Nielsen's 2017 U.S. music year-end report, for the first time ever, R&B/Hip-hop became the most-dominant genre in the US, and has been growing over time.

## Limitation

In addition to the observations and analysis, there are still many limitations we need to consider. The first one is the mixed distribution. Since we dropped songs before 1960, there could be a mixed distribution problem.

In the highest popularity group, the data is unbalanced. It may bias the prediction model towards the more common class. And may further overfit majority data and lead to lower accuracy. Thus, we need more data or we can try subsampling in future study.

In the clustering part, the elbow method only measures a global clustering characteristic, so the k found is not accurate enough. Also, due to the possible existence of the dependency within the data, elbow method could lead to an inaccurate result. A more sophisticated method is required to provide a statistical procedure to formalize the elbow in order to estimate the optimal number of clusters.

# Appendix

## Data Description

### Short Description:

Primary:

- - id (Id of track generated by Spotify)

Numerical:

- - acousticness (Ranges from 0 to 1)
- - danceability (Ranges from 0 to 1)
- - energy (Ranges from 0 to 1)
- - duration\_ms (Integer typically ranging from 200k to 300k)
- - instrumentalness (Ranges from 0 to 1)
- - valence (Ranges from 0 to 1)
- - popularity (Ranges from 0 to 100)
- - tempo (Float typically ranging from 50 to 150)
- - liveness (Ranges from 0 to 1)
- - loudness (Float typically ranging from -60 to 0)
- - speechiness (Ranges from 0 to 1)
- - year (Ranges from 1921 to 2020)

Dummy:

- - mode (0 = Minor, 1 = Major)

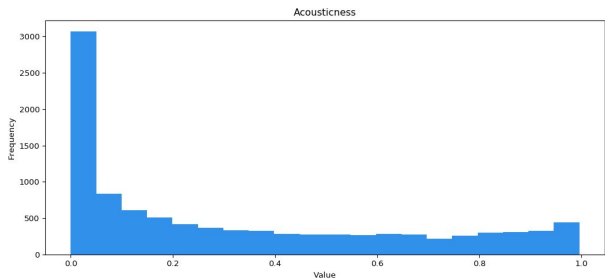


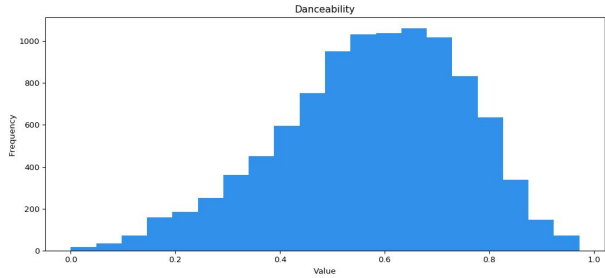
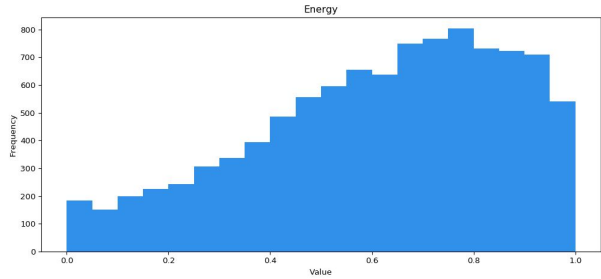
- - explicit (0 = No explicit content, 1 = Explicit content)

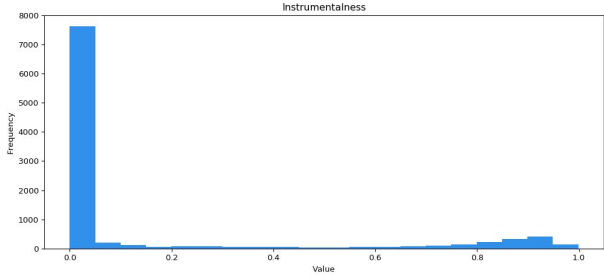
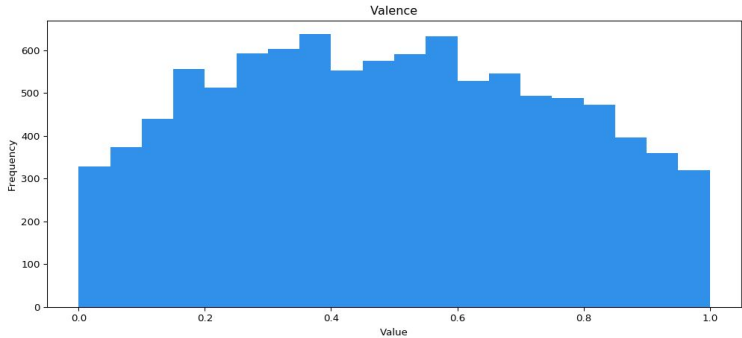
Categorical:

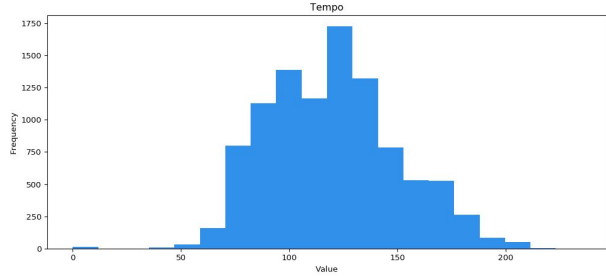
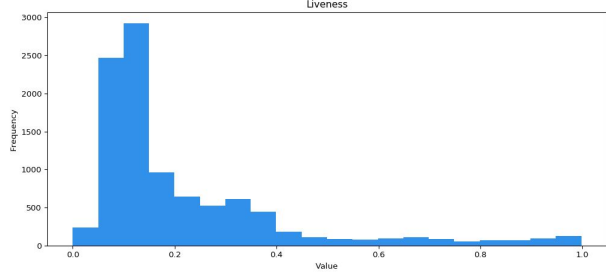
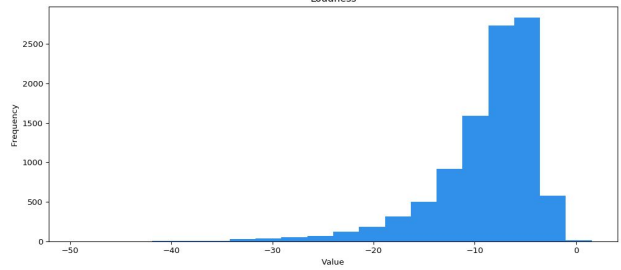
- - key (All keys on octave encoded as values ranging from 0 to 11, starting on C as 0, C# as 1 and so on...)
- - artists (List of artists mentioned)
- - release\_date (Date of release mostly in yyyy-mm-dd format, however precision of date may vary)
- - name (Name of the song)

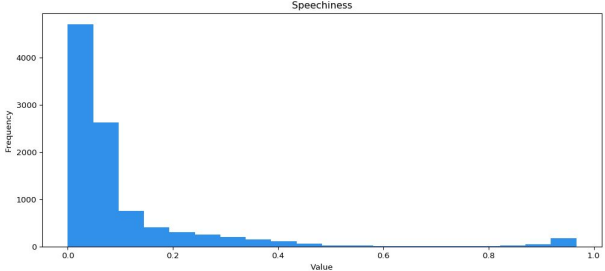
### Detailed Description:

Variable Name	Description
id	The Spotify ID for the track.
acousticness	<p>A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic. The distribution of values for this feature look like this:</p> 
danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is

	<p>most danceable. The distribution of values for this feature look like this:</p>  <p>A histogram titled 'Danceability' showing the frequency distribution of danceability values. The x-axis is labeled 'Value' and ranges from 0.0 to 1.0 with increments of 0.2. The y-axis is labeled 'Frequency' and ranges from 0 to 1000 with increments of 200. The distribution is unimodal and slightly right-skewed, peaking at a frequency of approximately 1000 for values between 0.6 and 0.7.</p>
energy	<p>Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. The distribution of values for this feature look like this:</p>  <p>A histogram titled 'Energy' showing the frequency distribution of energy values. The x-axis is labeled 'Value' and ranges from 0.0 to 1.0 with increments of 0.2. The y-axis is labeled 'Frequency' and ranges from 0 to 800 with increments of 100. The distribution is unimodal and slightly left-skewed, peaking at a frequency of approximately 800 for values between 0.7 and 0.8.</p>
duration_ms	The duration of the track in milliseconds.
instrumentalness	<p>Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the</p>

	<p>value approaches 1.0. The distribution of values for this feature look like this:</p> 
valence	<p>A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry). The distribution of values for this feature look like this:</p> 
popularity	<p>The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are.</p> <p>Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity. Note that the popularity value may lag actual popularity by a few days: the value is not updated in real time.</p>
tempo	<p>The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration. The distribution of values</p>

	<p>for this feature look like this:</p> 
liveness	<p>Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live. The distribution of values for this feature look like this:</p> 
loudness	<p>The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db. The distribution of values for this feature look</p>  <p>like this:</p>
speechiness	<p>Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that</p>

	<p>are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks. The distribution of values for this feature look like this:</p> 
year*	The released year of the track.
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.
explicit	Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown).
key	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C $\sharp$ / D $\flat$ , 2 = D, and so on. If no key was detected, the value is -1.
artists	The artists who performed the track.
release_date*	Date of release mostly in yyyy-mm-dd format, however precision of date may vary.
name	The name of the track.

## EDA Summary

From EDA, we are able to identify the unbalanced situation in our dataset (especially for *explicit*, and *popularity* over years). Moreover, we are able to find the correlation among all

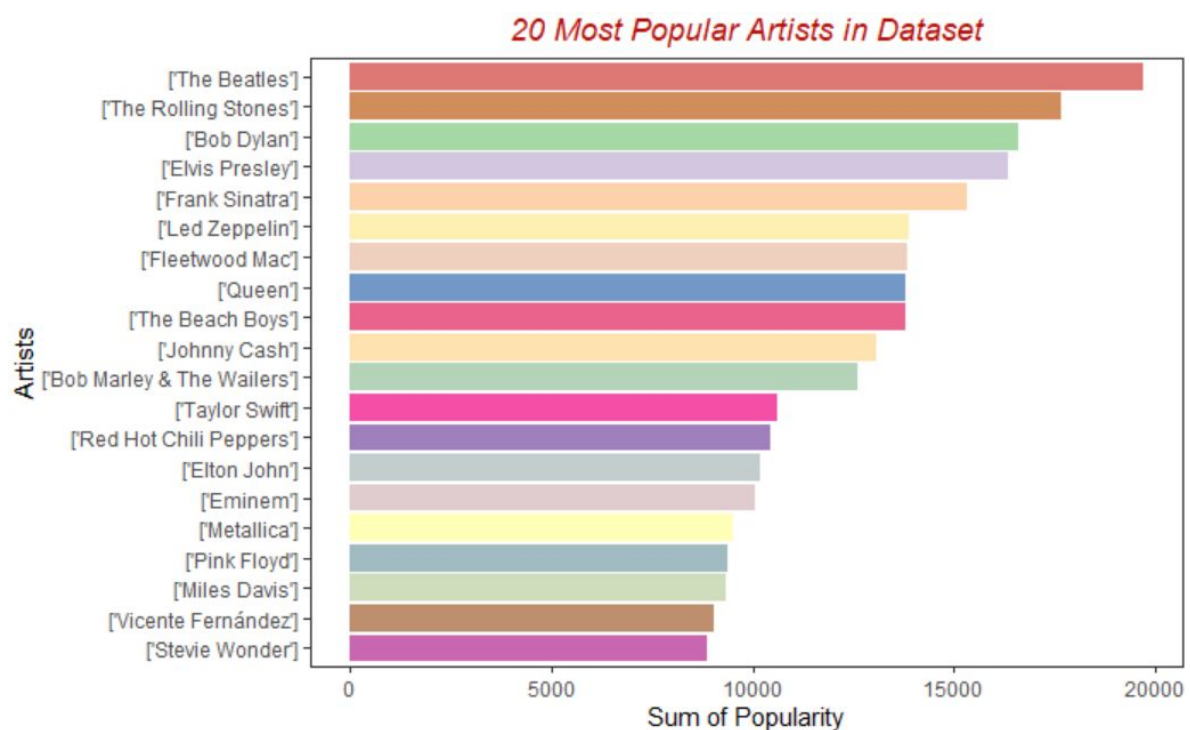
15 variables from our correlation heatmap. We've also observed that audio characteristics are changing over years (especially for *acousticness* and *energy*).

For our target variable - popularity, The distribution is far away from normal. There are a huge number of songs that their popularity = 0. After dropping all the zero records, it becomes an approximately normal distribution.

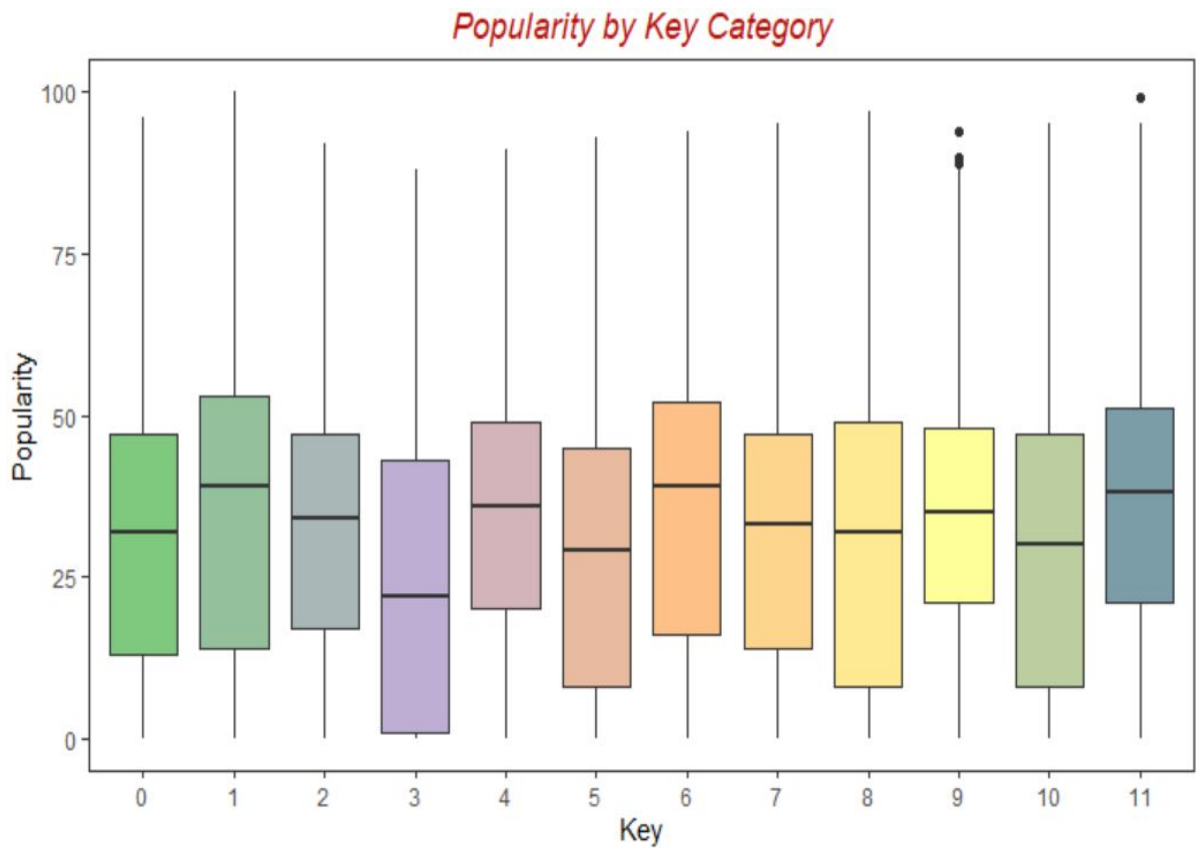
Most of the zero records clustered before 1960, the reason might be: Spotify was founded in 2006, its target users are relatively young who rarely listen to songs that before 1960s.

Thus, We decide to use the data after the 1960s.

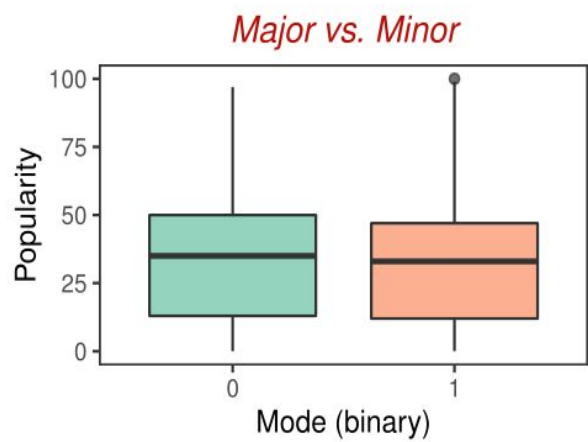
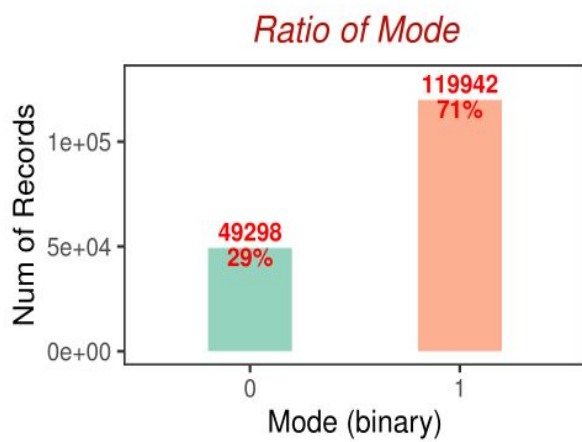
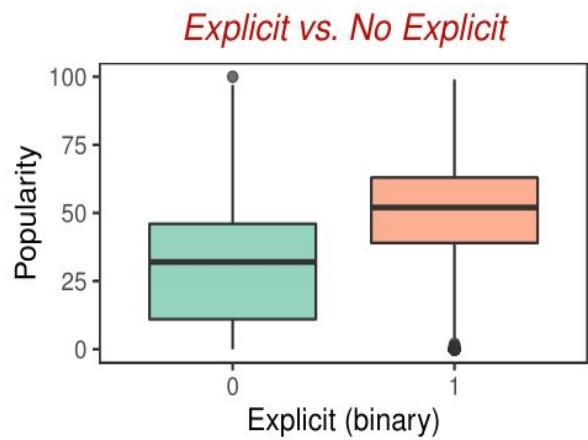
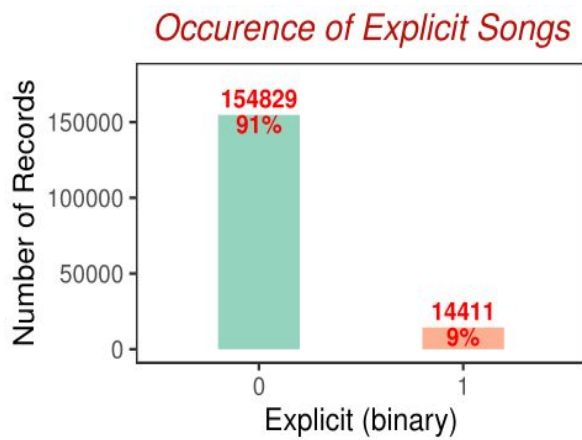
## Data Visualization on EDA



EDA Plot 1

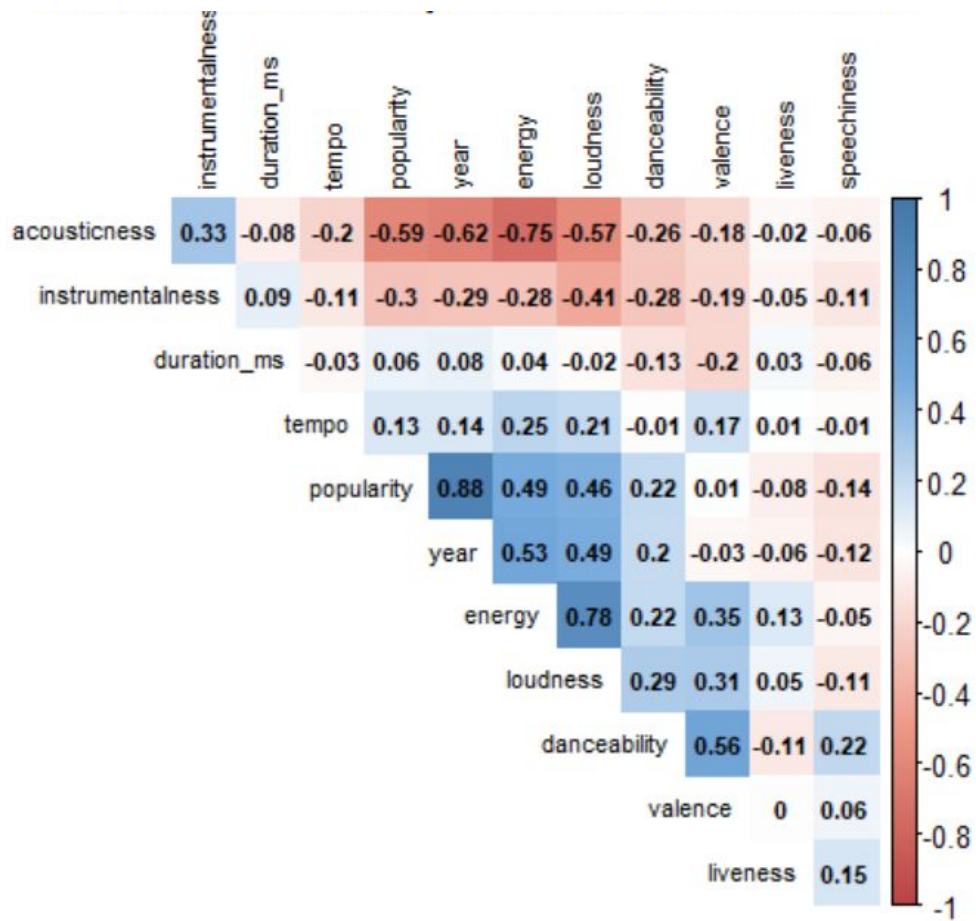


EDA Plot 2



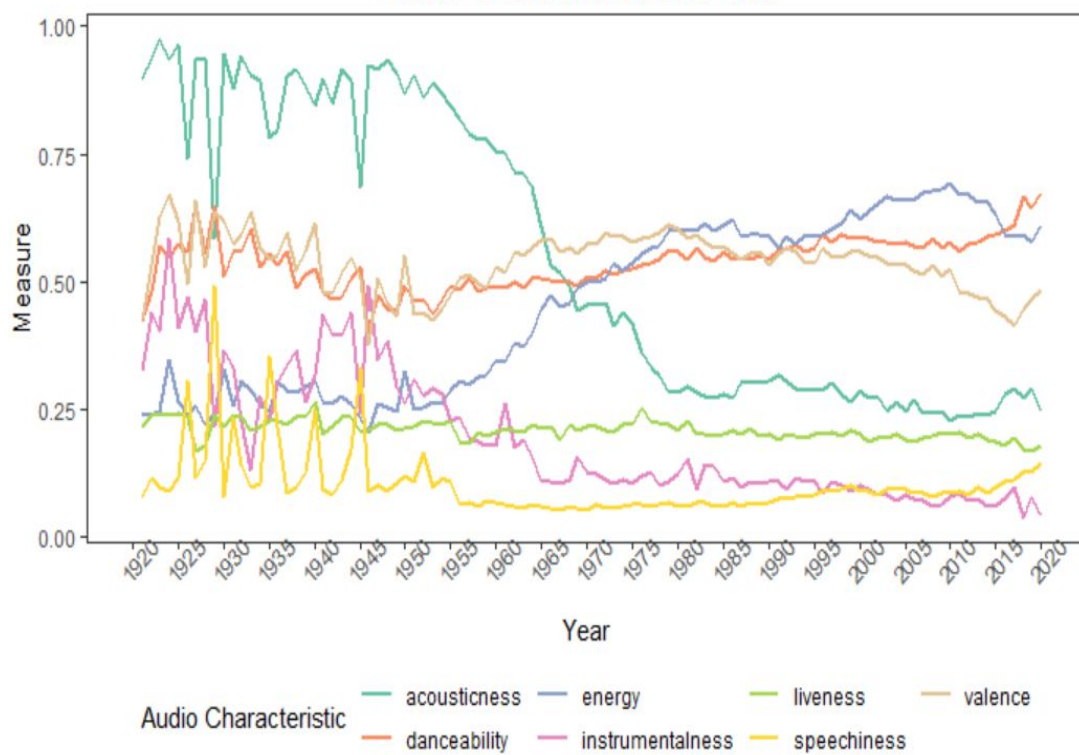


EDA Plot 3

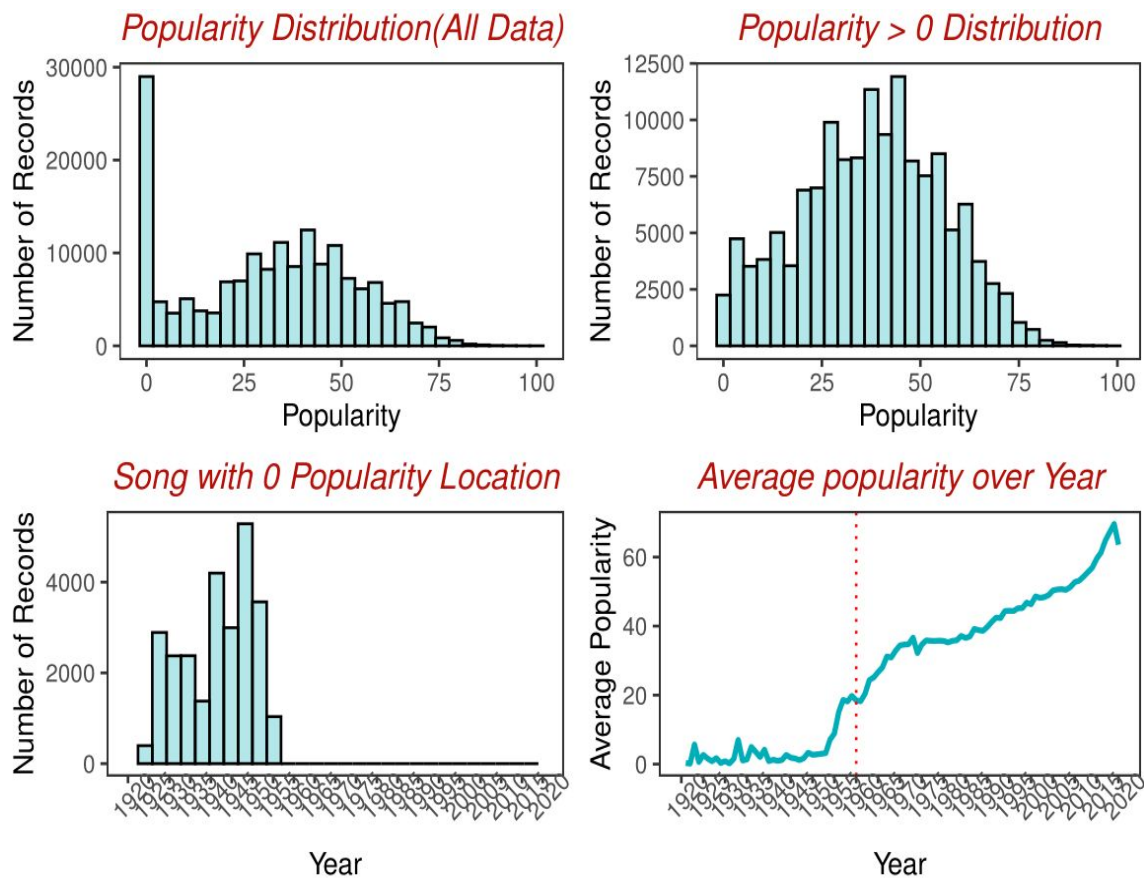


EDA Plot 4

### Audio Characteristic over Year

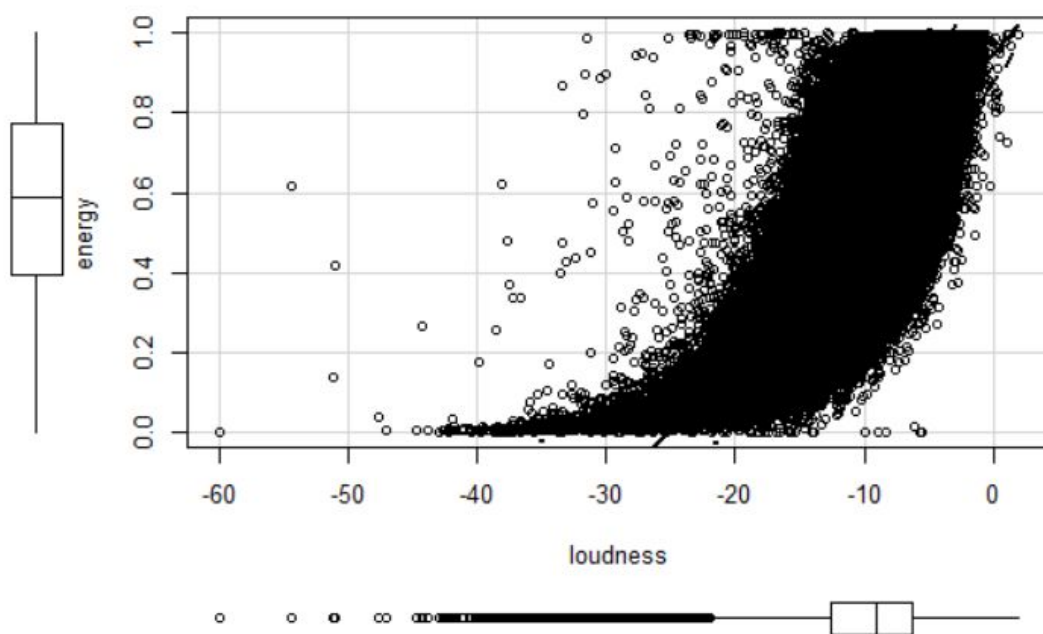


EDA Plot 5

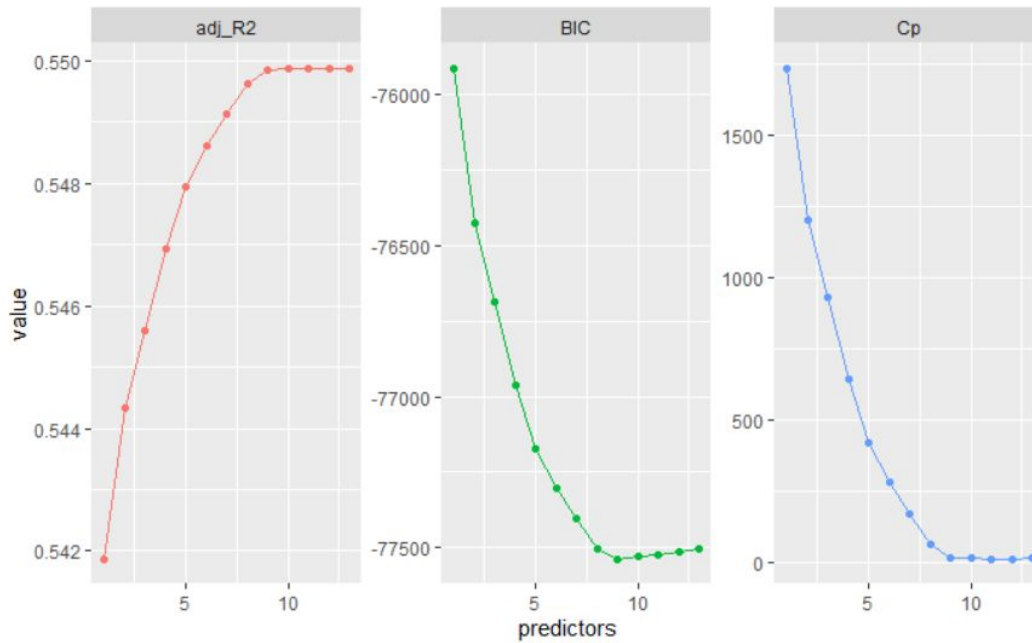


EDA Plot 6

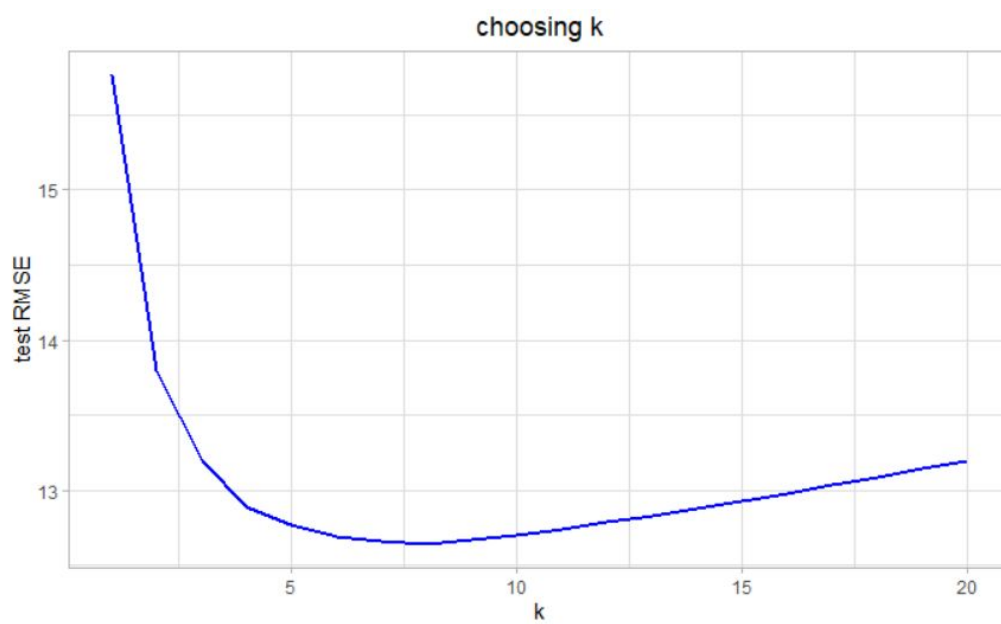
**Figure 1:** Scatterplot between energy and loudness



**Figure 2:** BIC, adjusted r-squared, and Cp score



**Figure 3:** Choosing k for Knn model



# References

1. Spotify Dataset 1921-2020, 160k+ Tracks, Kaggle, available from:

*<https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks>*

2. Get a track, Spotify for Developers, available from:

*<https://developer.spotify.com/documentation/web-api/reference/tracks/get-track/>*

3. Get Audio Features for a Track, Spotify for Developers, available from:

*<https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/>*

4. Nielsen's 2017 U.S. music year-end report, available from:

*<https://www.nielsen.com/us/en/insights/report/2018/2017-music-us-year-end-report/>*

5. Hip hop music, Wikipedia, available from:

*[https://en.wikipedia.org/wiki/Hip\\_hop\\_music](https://en.wikipedia.org/wiki/Hip_hop_music)*