ANLY 501 Project 2 Report

Min Xiao, Tianyu Yang, Cheng Zhong
Georgetown University
Nov 5th 2017

Introduction

In this round of analysis, we focused on analyzing the audio features and their interconnections with classification. The analysis consists of two major components, exploratory analysis and predictive analysis. We first looked into the distributions of the variables. Then, by using association rule mining and clustering analysis, we further explore the inter and intra connections of the variables. Next, we used statistical tests to analyzes the homogeneity of variable distributions of different classes. In the predictive part, we examined the predictive power of audio features over music category, meanwhile compare different classification methods. At the end, we found that (1) there are potential subcategories for a give category of music, (2) audio features has rather strong predictive power on music genre, and (3) except Naïve Bayes, all other classification methods yield satisfactory outcomes.

Exploratory Analysis

- ## Basic Statistical Analysis

In this section, we selected *category* and all audio feature attributes except *key* to conduct our first round of analysis. *Key* attribute is dropped since it is highly correlated to *mode*, an indication of major or minor key, which summarizes *Key*. The basic statistics of the audio feature attributes offer some insights on the distribution of the variable. *Acousticness*(slight right skewness)*, Danceability*(slight left skewness)*, Energy*(slight left skewness)*, Liveness*(slight right skewness)*, Tempo*(slight left skewness)*, Time Signature*(slight left skewness) *and Valence*(slight left skewness), whose means and medians are rather close, have less skewed distribution than *Instrumentalnessl*(extreme right skewness) and *speechiness*(right skewness). Meanwhile, since Acousticness, Danceability, Energy, Instrumentalness, Liveness, Speechiness and Valence are normalized by Spotify, the standard deviations reflect the Kurtosis of the distributions of the above variables. Lower(higher) standard deviation implies that more(less) values are around the mean, and thus the distribution will likely to have a thinner(fatter) tail. In addition, the category modes show that ‘party’ and ‘workout’ are the most common categories in our dataset. Other categories all appear in between 2000 to 5000 times.

Basic Statistics

	Mean	Median	Standard Deviation
Acousticness	0.308	0.282	0.289
Danceability	0.583	0.595	0.148

Energy	0.630	0.645	0.219
Instrumentalness	0.140	0.008	0.255
Liveness	0.212	0.206	0.170
Loudness	-8.435	-7.979	4.577
Mode	0.728		
Speechiness	0.123	0.084	0.165
Tempo	119.211	119.577	25.309
Time Signature	3.913	4.000	0.360
Valence	0.481	0.493	0.218

Category	Modes
Blues	3237
Classical	3148
Comedy	4306
Country	3480
Dinner	3901
Folk & Americana	4444
Funk	3445
Gaming	3780
Indie	3678
Jazz	3357
Kids	3264
Latin	3310
Metal	3566
Punk	3661
R&B	3227
Reggae	3717
Romance	2950
Sleep	3545
Soul	3766
Travel	3722
Trending	3036

chill	3653
electronic	3263
hip-hop	3280
mood	3158
party	37630
pop	2994
rock	3480
workout	35925

- ***Data Cleaning Insight***

In the second round of data cleaning, we identified outliers in all the quantitative variables using the boxplot and labeled them as outliers instead of dropping. In the previous project, we cleaned the dataset by dropping missing and bad values, and decomposing json strings. Since the percentage of bad values is extremely small, ignoring them will not significantly affect our analysis. In the case of missing values, we found that the observations either have all recorded audio features or have none of the features recorded. Therefore, observations that have missing audio features have absolutely no use in our analysis and were dropped. After missing and invalid values were cleaned, we move on to identifying outliers in all quantitative audio feature variables (acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, tempo, time signature and valence). Before any further analysis, we were only able to spot the outliers based on the variables' own distributions. Therefore, we used the boxplot method to find values that are above(below) 75(25) percent quantile +(-) 1.5*interquartile range in and created outlier labels for each of the aforementioned variables. Next, we labeled observations as general outlier if they are outliers in at least one of the audio feature attributes. Since a general outlier may still contain inlying attributes' values essential for further analysis, we decided to keep the outliers until deemed useless in following process.

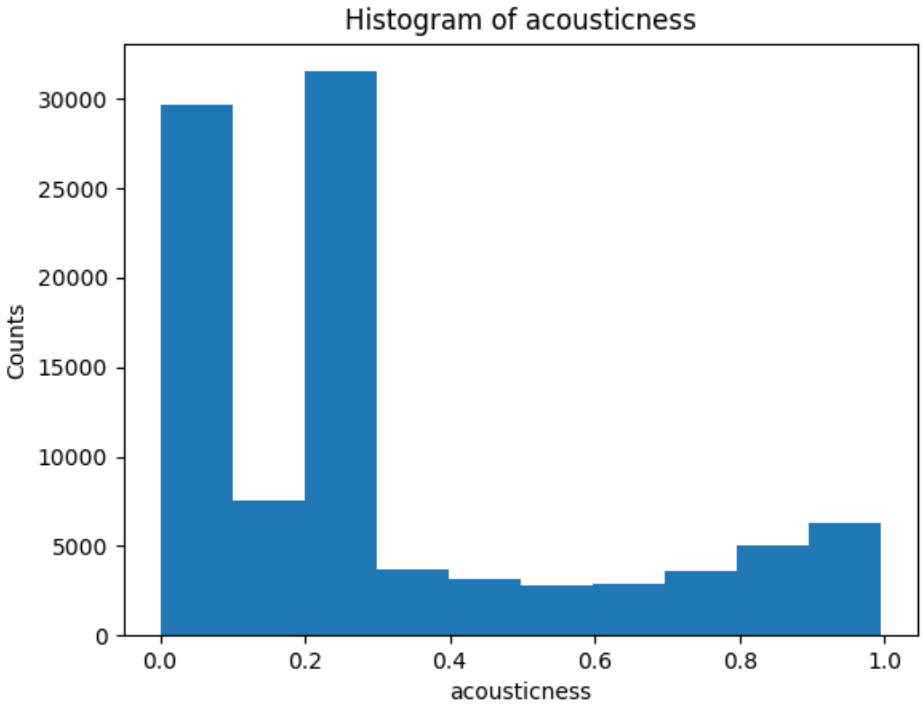
- ***Binning***

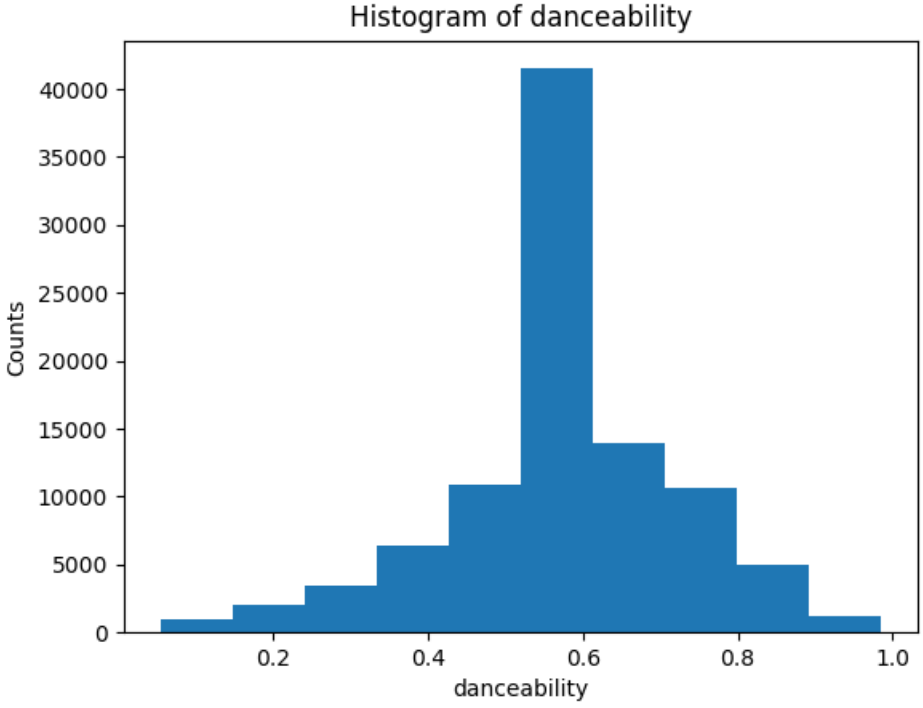
To conduct association rule mining, we used three equal-width bins (low, medium, high) to summarize each quantitative variable. Equal-width bins preserve the original distributions of variables and using three bins.

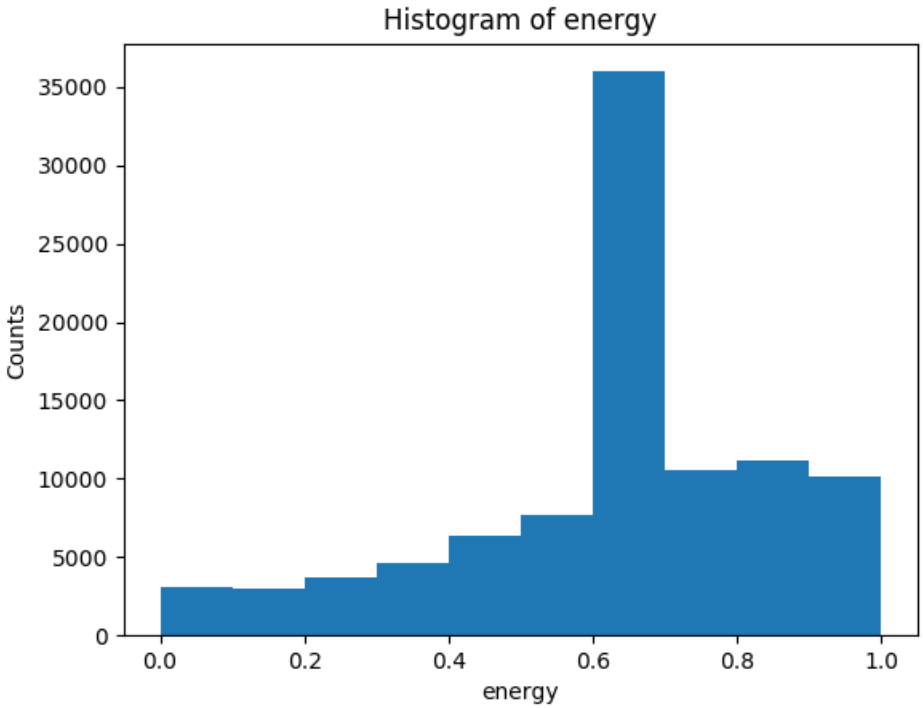
- ***Variable Distributions and Correlations***

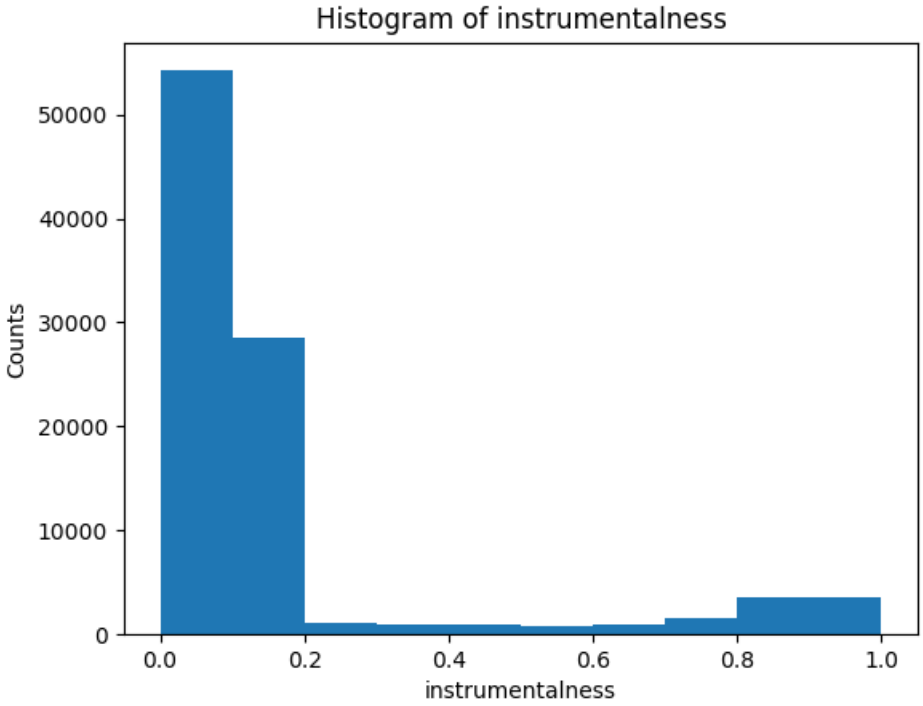
From the histograms, we observe that the distributions of acousticness, instrumentalness, speechiness, and liveness have obvious skewness to the right.

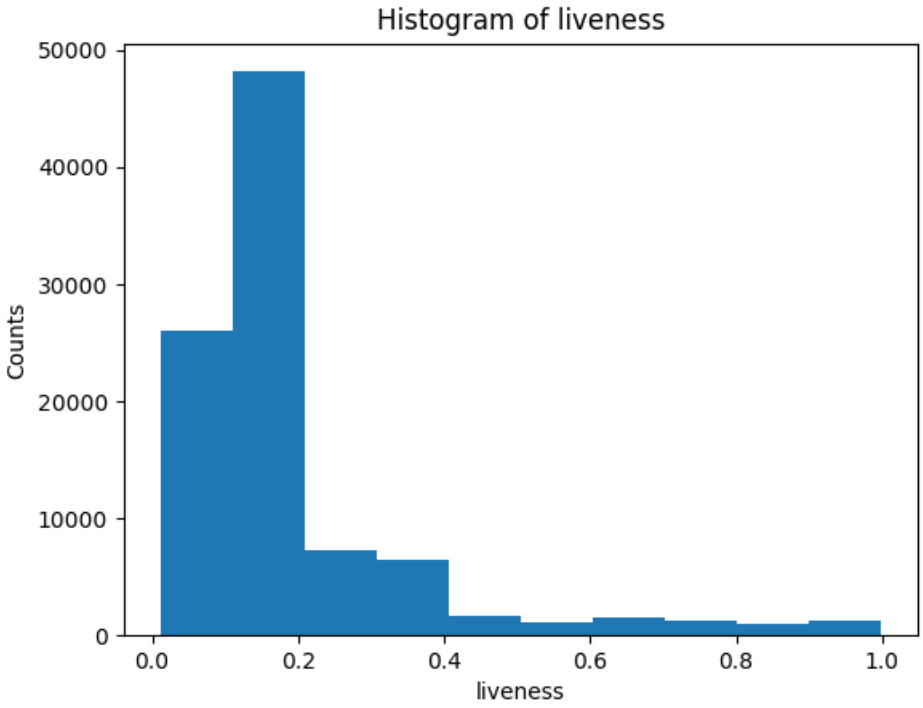
Meanwhile, the distributions of energy and loudness have obvious skewness to the right. Danceability, tempo and valence have more normal-like distributions but with higher sharpness in the middle. Although time signature is a nature quantitative variable, it only takes on three possible values in our dataset.

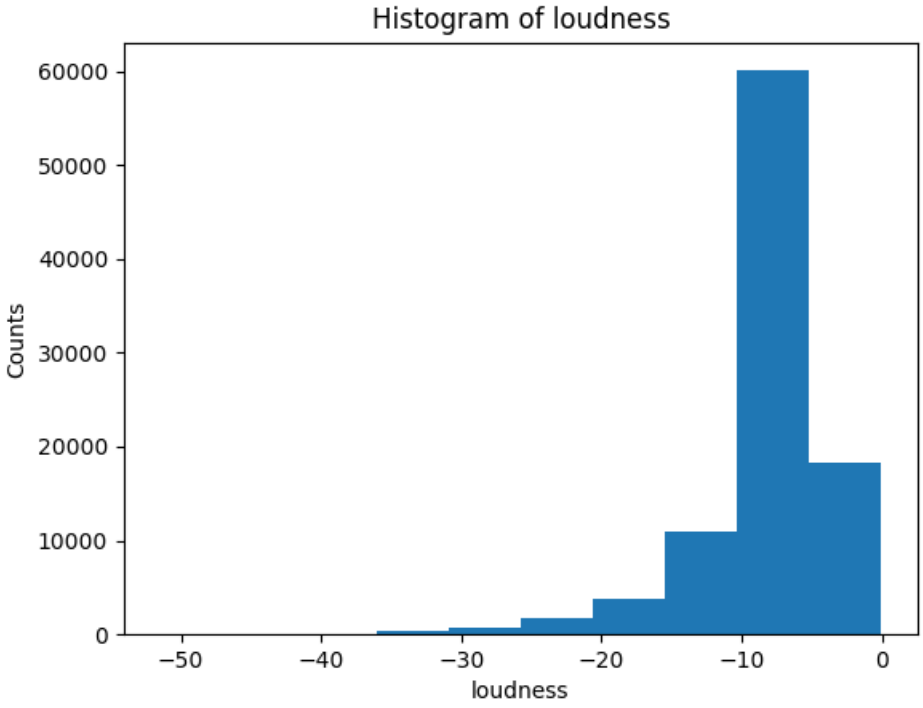


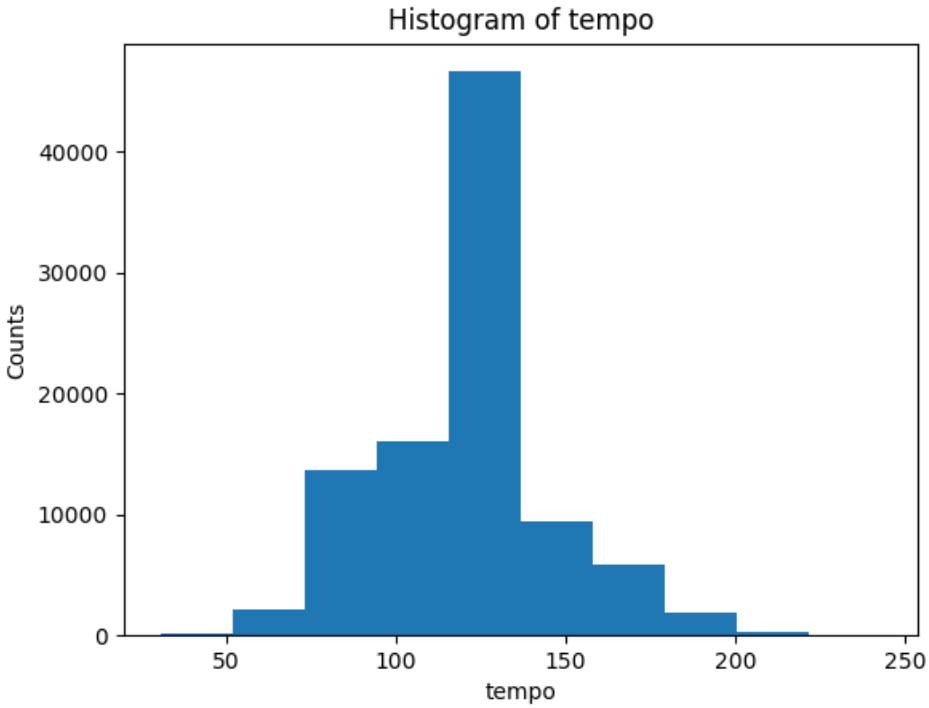


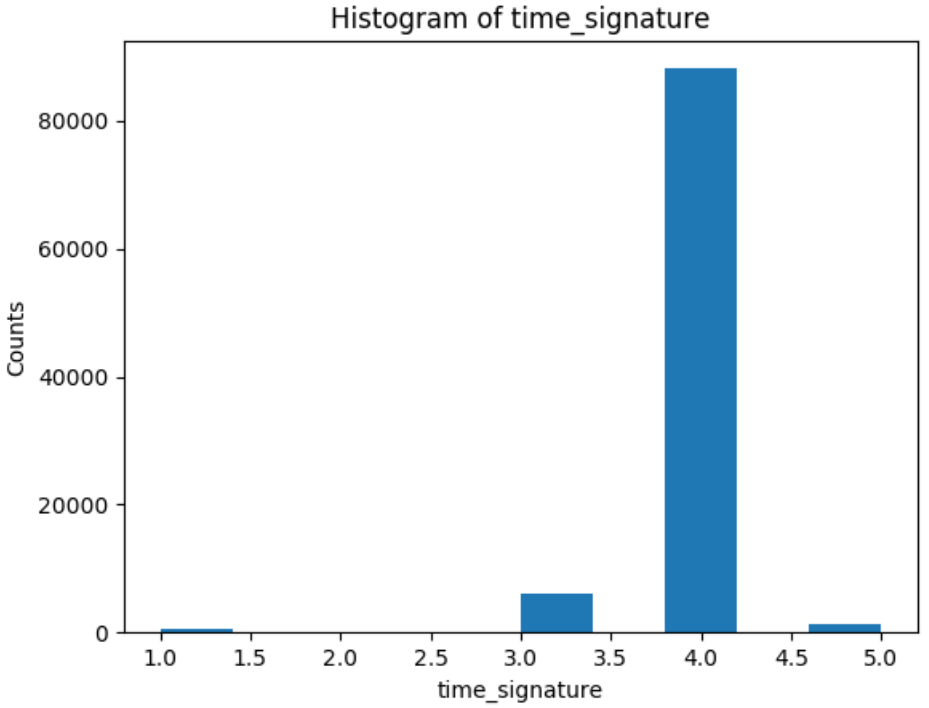

Histogram of energy
Counts
energy
0
5000
10000
15000
20000
25000
30000
35000
0.0
0.2
0.4
0.6
0.8
1.0

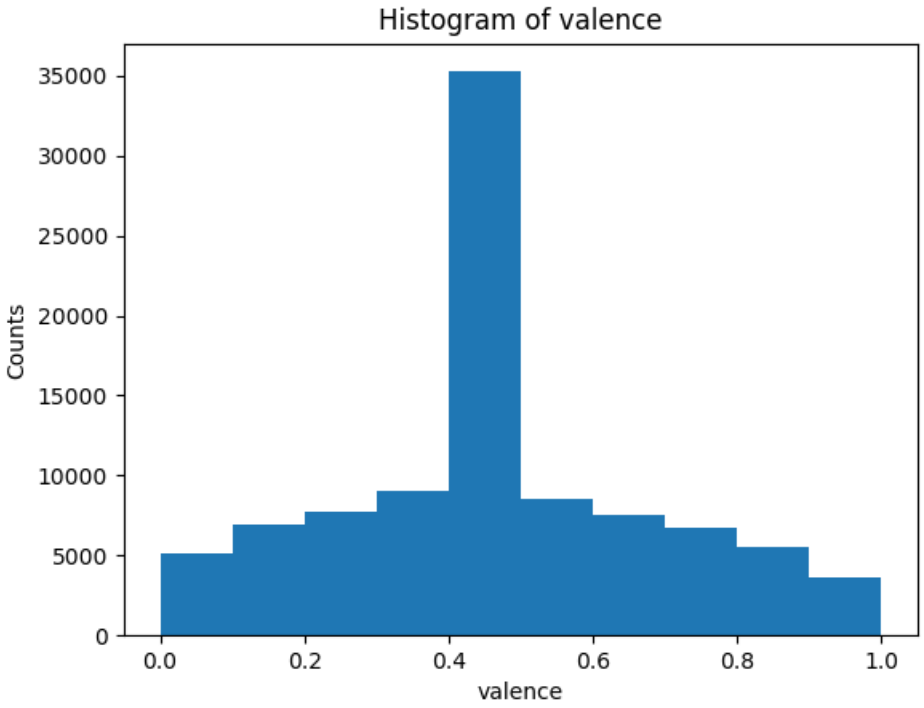

Histogram of instrumentalness
Counts
instrumentalness
0
10000
20000
30000
40000
50000
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of liveness
Counts
liveness
0
10000
20000
30000
40000
50000
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of loudness
Counts
loudness
0
10000
20000
30000
40000
50000
60000
−50
−40
−30
−20
−10
0

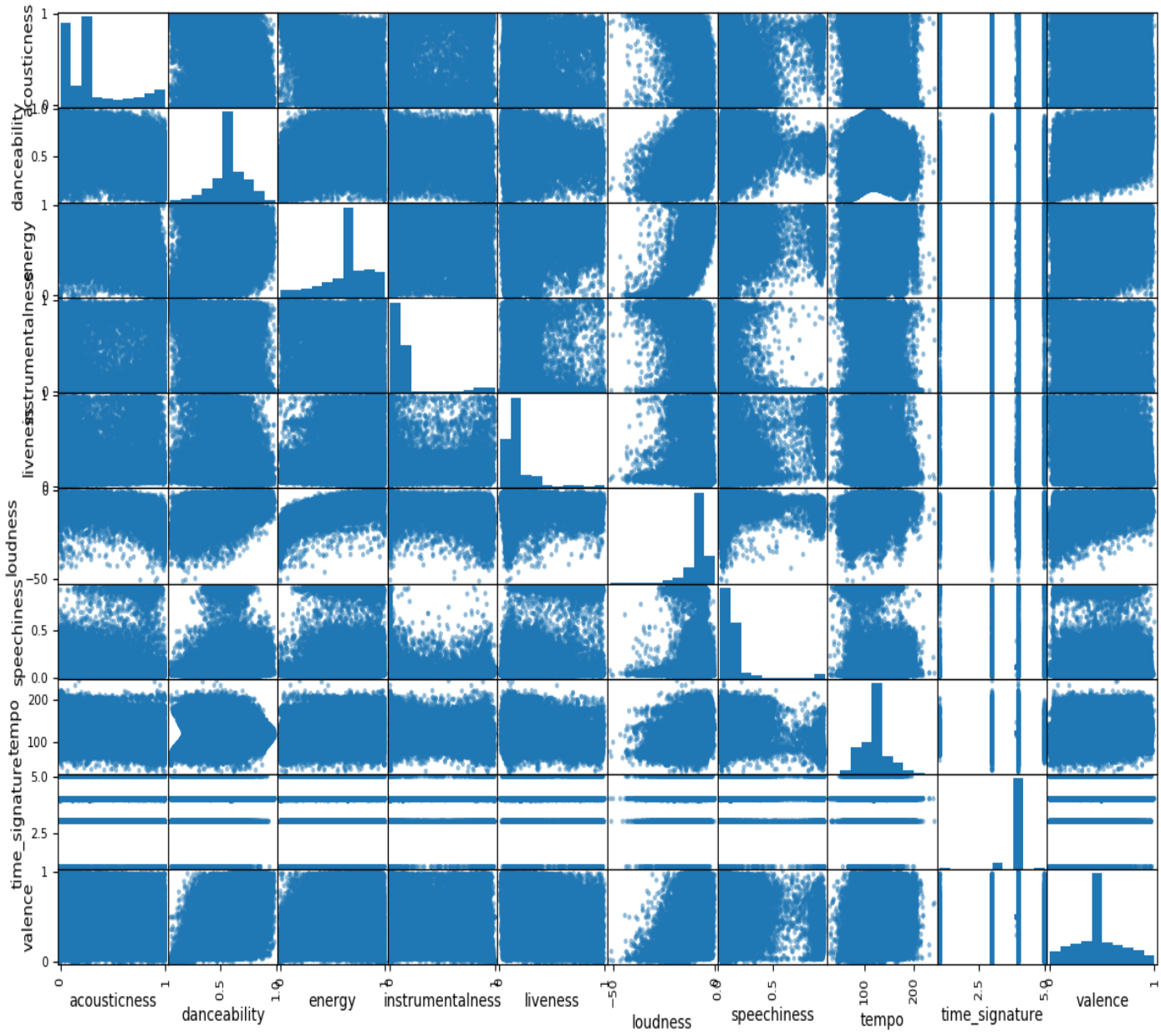

Histogram of tempo
Counts
40000
30000
20000
10000
0
50
100
150
200
250
tempo


Histogram of time_signature
Counts
80000
60000
40000
20000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
time_signature



The correlation matrix shows the relationship among all quantitative variables. Most of the entries are in between -0.3 to 0.3, which indicates rather weak linear relationship. However, there are a few cases with significant relationship. Acousticness has rather strong negative relationship with energy and loudness. Danceability has a rather strong positive relationship with valence. Energy is strongly correlated to loudness. The aforementioned relationships require cautions on assumptions of statistical tests that we conduct further down the road. The pairwise scatterplot strengthens our conclusion from the correlation matrix. The plots of the strongly correlated variables have better defined shapes. For example, the plot of danceability against valence shows an elliptical shape.

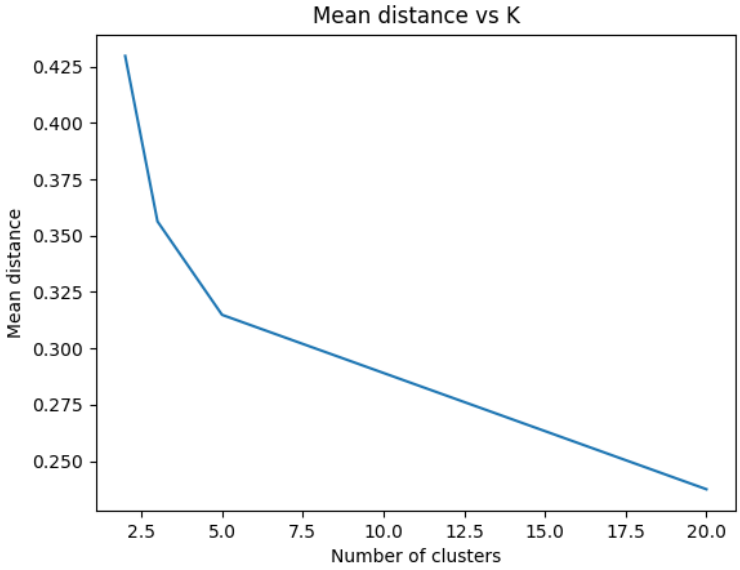
	acousticness	danceability	energy	instrumentalness	liveness	loudness	speechiness	tempo	time_signature	valence
acousticness	1.00	-0.24	-0.70	0.24	0.09	-0.64	0.21	-0.24	-0.21	-0.25
danceability	-0.24	1.00	0.18	-0.24	-0.09	0.32	0.06	-0.06	0.18	0.52
energy	-0.70	0.18	1.00	-0.33	0.22	0.77	0.14	0.22	0.16	0.31
instrumentalness	0.24	-0.24	-0.33	1.00	-0.11	-0.49	-0.15	-0.09	-0.08	-0.28
liveness	0.09	-0.09	0.22	-0.11	1.00	0.02	0.56	-0.06	-0.09	-0.04
loudness	-0.64	0.32	0.77	-0.49	0.02	1.00	-0.04	0.22	0.20	0.33
speechiness	0.21	0.06	0.14	-0.15	0.56	-0.04	1.00	-0.12	-0.13	-0.02
tempo	-0.24	-0.06	0.22	-0.09	-0.06	0.22	-0.12	1.00	0.04	0.09
time_signature	-0.21	0.18	0.16	-0.08	-0.09	0.20	-0.13	0.04	1.00	0.14
valence	-0.25	0.52	0.31	-0.28	-0.04	0.33	-0.02	0.09	0.14	1.00



- ***Cluster Analysis***

In the cluster analysis, we aim to find out are there sub-categories living in the given categories, using “pop” as an illustration. In order to find the clustering within “pop”, we performed clustering analysis using k means clustering, hierarchical clustering, and DBSCAN. By the k-means distance plot and the silhouette coefficients, we

observed a negative relationship between clustering quality and the number of cluster.



Overall Quality of Clustering

Based on the silhouette scores, hierarchical clustering with 2 final clusters produces clusters with best quality. However, overall, k-means clustering has a higher average silhouette score and a better fit. The highest silhouette is around 0.33, indicating that there may not be any subcategory.

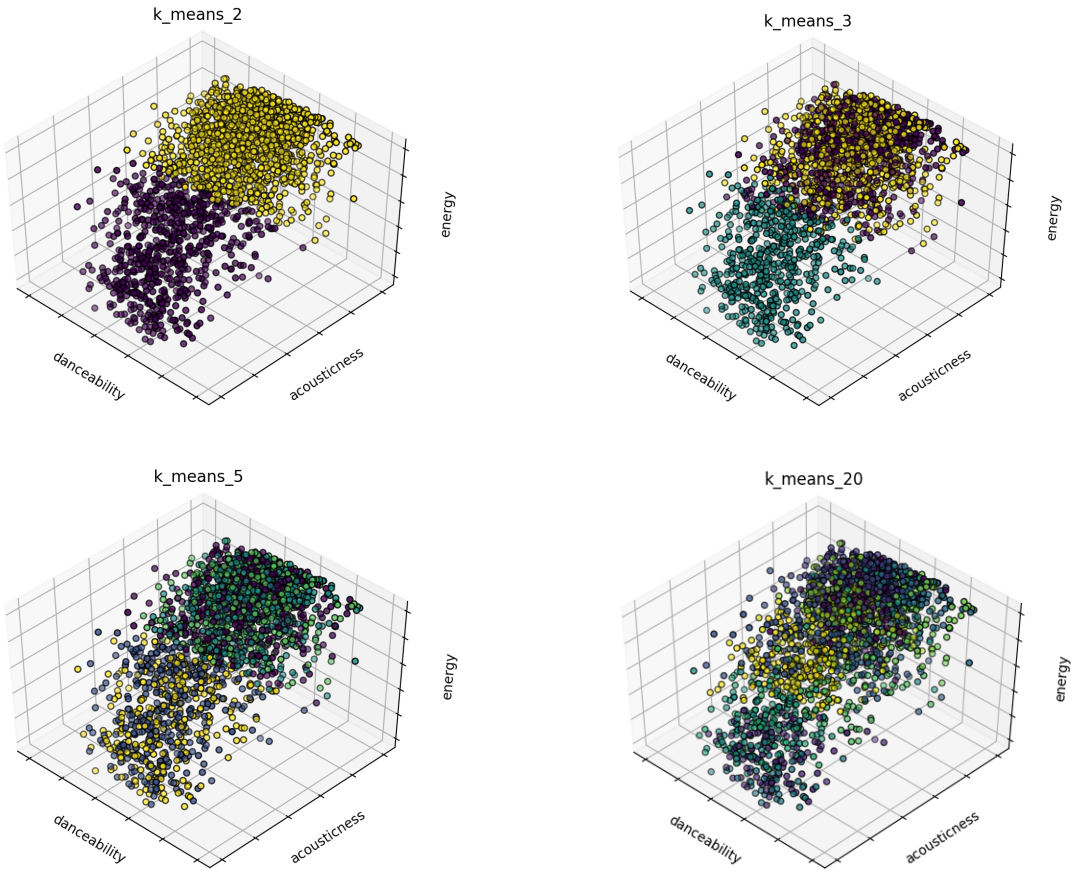
Methods	Kmeans k=2	Kmeans k=3	Kmeans k=5	Kmeans k=20	Average
Silhouette Score	0.326	0.318	0.228	0.172	0.261

Methods	Hierarchical n=2	Hierarchical n=3	Hierarchical n=5	Hierarchical n=20	Average
Silhouette Score	0.335	0.285	0.190	0.119	0.23225

Methods	DBSCAN (n=1)
Silhouette Score	0.288

Kmeans

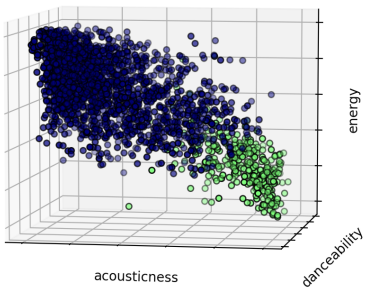
From the following cluster plot, we can see that when k = 2, there is a clear distinction among two clusters. When the number of cluster increases, the shape and boundary of each cluster get blurred, showing suboptimal quality.



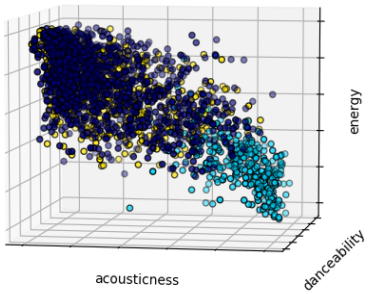
Hierarchical Clustering

The hierarchical clustering plots show the similar result to k-means. When n = 2, the two clusters have clear boundaries. When n = 3, there are mixed up among clusters. When n = 5 and 20, all the points are mixed up and there is no clear distinction among clusters.

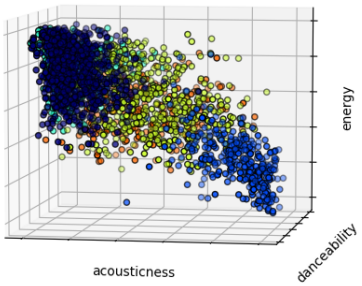
Hierachichal Clustering n= 2



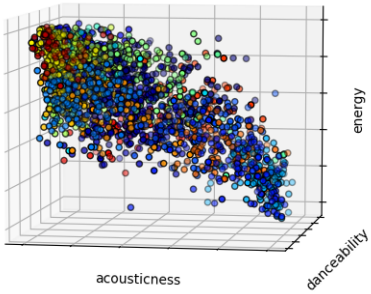
Hierachichal Clustering n= 3



Hierachichal Clustering n= 5

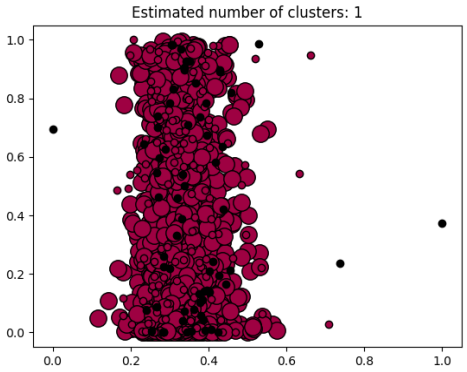


Hierachichal Clustering n= 20



DBSCAN

Using dbscan, we did not obtain any meaningful clustering result.



Overall, the data("pop") has weak subcategories. From the 3-D clustering plots, we observe two potential subcategories formed by songs with higher energy, danceability, and acousticness, and their counterparties.

Association Rule

We further explored the correlations among variables by association rule mining with three levels of support and confidence. Itemsets are formed by binned variables. To explore more than pairwise relationship, we only focused on itemsets with three or more elements.

Support, Confidence = 0.8

	items	support		ordered_statistics
0	(highloudness)	0.946512	0	[(), (highloudness), 0.9465123293084979, 1.0)]
1	(hightime_signature)	0.931609	1	[(), (hightime_signature), 0.9316089981591839...
2	(lowinstrumentalness)	0.874481	2	[(), (lowinstrumentalness), 0.874481295435399...
3	(lowliveness)	0.875917	3	[(), (lowliveness), 0.875916508065271, 1.0)]
4	(lowspeechiness)	0.944224	4	[(), (lowspeechiness), 0.9442243091739208, 1.0)]
5	(hightime_signature, highloudness)	0.894200	5	[(highloudness), (hightime_signature), 0.9447...
6	(highloudness, lowinstrumentalness)	0.859360	6	[(highloudness), (lowinstrumentalness), 0.907...
7	(highloudness, lowliveness)	0.830302	7	[(highloudness), (lowliveness), 0.87722228326...
8	(highloudness, lowspeechiness)	0.895500	8	[(highloudness), (lowspeechiness), 0.94610482...
9	(hightime_signature, lowinstrumentalness)	0.825112	9	[(hightime_signature), (lowinstrumentalness),...
10	(hightime_signature, lowliveness)	0.827566	10	[(hightime_signature), (lowliveness), 0.88831...
11	(hightime_signature, lowspeechiness)	0.892037	11	[(hightime_signature), (lowspeechiness), 0.95...
12	(lowinstrumentalness, lowspeechiness)	0.820432	12	[(lowinstrumentalness), (lowspeechiness), 0.9...
13	(lowliveness, lowspeechiness)	0.857009	13	[(lowliveness), (lowspeechiness), 0.978414191...
14	(hightime_signature, highloudness, lowinstrume...	0.814629	14	[(hightime_signature, highloudness), (lowinst...
15	(hightime_signature, highloudness, lowspeechin...	0.857363	15	[(hightime_signature, highloudness), (lowspee...
16	(highloudness, lowinstrumentalness, lowspeechi...	0.809866	16	[(highloudness, lowinstrumentalness), (lowspe...
17	(highloudness, lowliveness, lowspeechiness)	0.812039	17	[(highloudness, lowliveness), (lowspeechiness...
18	(hightime_signature, lowliveness, lowspeechiness)	0.810583	18	[(hightime_signature, lowliveness), (lowspeec...

Support, Confidence = 0.75

	items	support		ordered_statistics
0	(highloudness)	0.946512	0	[(), (highloudness), 0.9465123293084979, 1.0)]
1	(hightime_signature)	0.931609	1	[(), (hightime_signature), 0.9316089981591839...
2	(lowinstrumentalness)	0.874481	2	[(), (lowinstrumentalness), 0.874481295435399...
3	(lowliveness)	0.875917	3	[(), (lowliveness), 0.875916508065271, 1.0)]
4	(lowspeechiness)	0.944224	4	[(), (lowspeechiness), 0.9442243091739208, 1.0)]
5	(hightime_signature, highloudness)	0.894200	5	[(highloudness), (hightime_signature), 0.9447...
6	(highloudness, lowinstrumentalness)	0.859360	6	[(highloudness), (lowinstrumentalness), 0.907...
7	(lowliveness, highloudness)	0.830302	7	[(highloudness), (lowliveness), 0.87722228326...
8	(highloudness, lowspeechiness)	0.895500	8	[(highloudness), (lowspeechiness), 0.94610482...
9	(hightime_signature, lowinstrumentalness)	0.825112	9	[(hightime_signature), (lowinstrumentalness),...
10	(lowliveness, hightime_signature)	0.827566	10	[(hightime_signature), (lowliveness), 0.88831...
11	(hightime_signature, lowspeechiness)	0.892037	11	[(hightime_signature), (lowspeechiness), 0.95...
12	(lowliveness, lowinstrumentalness)	0.763793	12	[(lowinstrumentalness), (lowliveness), 0.8734...
13	(lowspeechiness, lowinstrumentalness)	0.820432	13	[(lowinstrumentalness), (lowspeechiness), 0.9...
14	(lowliveness, lowspeechiness)	0.857009	14	[(lowliveness), (lowspeechiness), 0.978414191...
15	(hightime_signature, highloudness, lowinstrume...	0.814629	15	[(hightime_signature, highloudness), (lowinst...
16	(lowliveness, hightime_signature, highloudness)	0.795181	16	[(hightime_signature, highloudness), (lowlive...
17	(hightime_signature, highloudness, lowspeechin...	0.857363	17	[(hightime_signature, highloudness), (lowspee...
18	(lowliveness, highloudness, lowinstrumentalness)	0.753601	18	[(highloudness, lowinstrumentalness), (lowliv...
19	(highloudness, lowspeechiness, lowinstrumental...	0.809866	19	[(highloudness, lowinstrumentalness), (lowspe...
20	(lowliveness, highloudness, lowspeechiness)	0.812039	20	[(lowliveness, highloudness), (lowspeechiness...
21	(hightime_signature, lowspeechiness, lowinstru...	0.787037	21	[(hightime_signature, lowinstrumentalness), (...
22	(lowliveness, hightime_signature, lowspeechiness)	0.810583	22	[(lowliveness, hightime_signature), (lowspeec...
23	(hightime_signature, highloudness, lowspeechin...	0.779144	23	[(hightime_signature, highloudness, lowinstru...
24	(lowliveness, hightime_signature, highloudness...	0.778592	24	[(lowliveness, hightime_signature, highloudne...

Support, Confidence = 0.7

	items	support
0	(highloudness)	0.946512
1	(hightime_signature)	0.931609
2	(lowacousticness)	0.728173
3	(lowinstrumentalness)	0.874481
4	(lowliveness)	0.875917
5	(lowspeechiness)	0.944224
6	(mediumtempo)	0.728038
7	(hightime_signature, highloudness)	0.894200
8	(lowacousticness, highloudness)	0.724169
9	(highloudness, lowinstrumentalness)	0.859360
10	(lowliveness, highloudness)	0.830302
11	(highloudness, lowspeechiness)	0.895500
12	(highloudness, mediumtempo)	0.705875
13	(hightime_signature, lowacousticness)	0.710430
14	(hightime_signature, lowinstrumentalness)	0.825112
15	(lowliveness, hightime_signature)	0.827566
16	(hightime_signature, lowspeechiness)	0.892037
17	(lowacousticness, lowspeechiness)	0.712198
18	(lowliveness, lowinstrumentalness)	0.763793
19	(lowspeechiness, lowinstrumentalness)	0.820432
20	(lowliveness, lowspeechiness)	0.857009
21	(mediumtempo, lowspeechiness)	0.705750
22	(hightime_signature, highloudness, lowacoustic...	0.707425
23	(hightime_signature, highloudness, lowinstrume...	0.814629
24	(lowliveness, hightime_signature, highloudness)	0.795181
25	(hightime_signature, highloudness, lowspeechin...	0.857363
26	(lowacousticness, highloudness, lowspeechiness)	0.708506
27	(lowliveness, highloudness, lowinstrumentalness)	0.753601
28	(highloudness, lowspeechiness, lowinstrumental...	0.809866
29	(lowliveness, highloudness, lowspeechiness)	0.812039
30	(lowliveness, hightime_signature, lowinstrumen...	0.732697
31	(hightime_signature, lowspeechiness, lowinstru...	0.787037
32	(lowliveness, hightime_signature, lowspeechiness)	0.810583
33	(lowliveness, lowspeechiness, lowinstrumentaln...	0.746186
34	(lowliveness, hightime_signature, highloudness...	0.725209
35	(hightime_signature, highloudness, lowspeechin...	0.779144
36	(lowliveness, hightime_signature, highloudness...	0.778592
37	(lowliveness, highloudness, lowspeechiness, lo...	0.736524
38	(lowliveness, hightime_signature, lowspeechine...	0.716889
39	(highloudness, lowinstrumentalness, lowlivenes...	0.709702

	ordered_statistics
0	[(), (highloudness), 0.9465123293084979, 1.0)]
1	[(), (hightime_signature), 0.9316089981591839...
2	[(), (lowacousticness), 0.7281728079207097, 1...
3	[(), (lowinstrumentalness), 0.874481295435399...
4	[(), (lowliveness), 0.875916508065271, 1.0)]
5	[(), (lowspeechiness), 0.9442243091739208, 1.0)]
6	[(), (mediumtempo), 0.7280376067309392, 1.0)]
7	[(highloudness), (hightime_signature), 0.9447...
8	[(highloudness), (lowacousticness), 0.7650917...
9	[(highloudness), (lowinstrumentalness), 0.907...
10	[(highloudness), (lowliveness), 0.87722228326...
11	[(highloudness), (lowspeechiness), 0.94610482...
12	[(highloudness), (mediumtempo), 0.74576420173...
13	[(hightime_signature), (lowacousticness), 0.7...
14	[(hightime_signature), (lowinstrumentalness),...
15	[(hightime_signature), (lowliveness), 0.88831...
16	[(hightime_signature), (lowspeechiness), 0.95...
17	[(lowacousticness), (lowspeechiness), 0.97806...
18	[(lowinstrumentalness), (lowliveness), 0.8734...
19	[(lowinstrumentalness), (lowspeechiness), 0.9...
20	[(lowliveness), (lowspeechiness), 0.978414191...
21	[(lowspeechiness), (mediumtempo), 0.747439145...
22	[(hightime_signature, highloudness), (lowacou...
23	[(hightime_signature, highloudness), (lowinst...
24	[(hightime_signature, highloudness), (lowlive...
25	[(hightime_signature, highloudness), (lowspee...
26	[(lowacousticness, highloudness), (lowspeechi...
27	[(highloudness, lowinstrumentalness), (lowliv...
28	[(highloudness, lowinstrumentalness), (lowspe...
29	[(lowliveness, highloudness), (lowspeechiness...
30	[(hightime_signature, lowinstrumentalness), (...
31	[(hightime_signature, lowinstrumentalness), (...
32	[(lowliveness, hightime_signature), (lowspeec...
33	[(lowliveness, lowinstrumentalness), (lowspee...
34	[(hightime_signature, highloudness, lowinstru...
35	[(hightime_signature, highloudness, lowinstru...
36	[(lowliveness, hightime_signature, highloudne...
37	[(lowliveness, highloudness, lowinstrumentaln...
38	[(lowliveness, hightime_signature, lowinstrum...
39	[(lowliveness, hightime_signature, highloudne...

After examining the itemsets according to three confidence levels, we observe that high time signature and high loudness imply low instrumentalness and lowspeechiness. High loudness and low instrumentalness imply low speechiness. High Loudness and low liveness imply low speechiness. High time signature and low liveness imply low speechiness.

The result is not surprising since the high loudness is the most frequent element with support of 94.6 percent. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude), which relates to the timbre of the music. Nowadays, the majority of the music use similar compositions of instruments that generate high loudness. Also, we have the high time signature for the majority of the songs. Time signature (meter) is a notational convention to specify how many beats are in each bar (or measure), which relates to the rhythmic structure of music. High frequency implies that modern music uses similar rhythmic structure. Besides, low speechiness(stands for the spoken words in a track) also has support over 0.94, indicating that most of the music are not lyrics oriented(rap music). For frequent itemsets contain two attributes, (low speechiness,high loudness) and (high time signature, high loudness) are extremely frequent, since both of them contains the highest support single attribute. Yet, (low acousticness) is a frequent itemset that has

0.728 support, exceeding our expectation on the proportion of music composed by electronic instruments.

Hypothesis and Classification

Hypothesis1:
H0: there is no significant difference in mean of feature *energy* values between pop songs and non-pop songs.

- **T-test:**

T test is used to test the difference between means when two samples are from independently normal distributions. We use the t statistic to evaluate whether two means are identical. A p value will be calculated. If p value is lower than the threshold, we can reject the Null hypothesis. Here we set the threshold as 0.05.

Motivation:
In the real world, there are no connection between pop songs and non pop songs. A reasonable assumption is that pop songs data and non-pop songs data are from independent normal distribution. To compare means of energy for two independent normal datasets, t test is a common choice.

Experiment:
1. select records with Parentcat as pop group
2. Rest rercords belong to non-pop group
3. apply t test on energy values of two groups

Result:
The statistical test result is

T statistic	P value
2.4435	0.0145

Since p value is smaller 0.05, we thus can reject the null hypothesis. There is significant difference between means of feature energy values between pop songs and non-pop songs.

Hypothesis 2:
H0: There is linear relationship between energy, loudness and acousticness, i.e. Acousticness=a*energy + b*loudness+ c

- **Linear Regression:**

Linear regression is a linear model to fit the independent variable X and dependent variable Y. It assumes there is a linear relationship between X and Y, residuals should be approximately normally distributed and homoscedasticity. Techniques to estimate the coefficients like OLS , max likelihood are commonly used.

Motivation:
In our dataset, there seems to be some connections between energy, loudness and acousticness. A linear model is used to fit the relationship.

Experiments:

1. Independent variable: energy, loudness.
2. Dependent variable: acousticness.
3. OLS is used to estimate the coefficients.

Results:

Dep. Variable:	acousticness	R-squared:	0.510
Model:	OLS	Adj. R-squared:	0.510
Method:	Least Squares	F-statistic:	5.001e+04
Date:	Sat, 04 Nov 2017	Prob (F-statistic):	0.00
Time:	22:58:01	Log-Likelihood:	17156.
No. Observations:	96153	AIC:	-3.431e+04
Df Residuals:	96150	BIC:	-3.428e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.5910	0.005	127.839	0.000	0.582	0.600
energy	-0.6620	0.005	-141.145	0.000	-0.671	-0.653
loudness	-0.0159	0.000	-70.807	0.000	-0.016	-0.015

Omnibus:	11110.391	Durbin-Watson:	1.166
Prob(Omnibus):	0.000	Jarque-Bera (JB):	30240.403
Skew:	0.647	Prob(JB):	0.00
Kurtosis:	5.424	Cond. No.	96.3

p values for coefficients are 0, which means coefficients are statistically significant. The R square is 0.732, which means this fitting is acceptable. Thus, there is a linear relationship between acousticness, energy and loudness.
From the table, this can be represented as

Acounsticness=-0.662*energy-0.0159*loudness+0.591

Hypothesis 3:
Different classification methods have similar performance over datasets when using sound features to predict playlist categories.

Motivation:
In a large pool of song tracks, creating playlists or classify song tracks into genres manually is tedious and inefficient. Machine learning techniques will make it possible to classify song tracks automatically and accurately. However, the real-world problem is complicated. Some methods may suitable while others not. To understand which methods are appropriate, we conducted a series of experiments.
Summary:

- **Decision Tree**
  Decision tree is a non-parametric classifier with the hierarchical tree structure. It is also a series of exclusive rule settings to separate datasets into several classes. Initially, we have a full dataset, and then repeatedly split the dataset into smaller subsets. Every split is based on the result of the last.
  To train a decision tree model, we need set rules to split nodes, to stop splitting, to prune an over large tree. Impurity is the key factor in splitting.
- **K-Nearest neighbor**

KNN is a non-parametric lazy learning algorithm and no assumption of the dataset is required. Unlike other eager learning classifiers, KNN performs classification by deciding its class based on the similarity to K nearest neighbors. Generally, we will find K nearest neighbors of an instance (for example, based on Euclidean distance) and assign the majority of its neighbors' class to the object.

- **Naïve Bayes**

Naive Bayes methods are a set of supervised learning algorithms based on Bayes' theory. It is suitable for high dimensional data. It assumes input features are independent. Class labels are assigned according to the probability P(A|C)P(C). Here, A is feature values and C is class label. Thus it is equivalent to choosing class C that maximizes P(A|C)P(C).

- **SVM**

SVM is a supervised learning method that constructs hyperplanes to separate datasets. It can use different kernel such linear, gamma, sigmoid to build the classifier. Intuitively, SVM build its classification boundary by maximizing the margin. For a linear kernel, the margin is the width that the boundary could be increased by beofe hitting a datapoint.

- **Random Forest:**

Random forest is an ensemble learning method for classification. To construct a random forest, a series of trees are constructed and give class labels independently. The output of the random forest is the majority vote of outputs of individual trees. Common methods to compute output of random forest include mean or mode. In random forest, more advanced techniques can be combined. However, the complexity of the model makes it harder to interpret.

In the next section, we will test every method above on our dataset.

Experiment:

In the dataset, songs can be assigned to different classes, so we will use binary classifiers to predict if a song track can be classified as a certain category. This means the whole dataset will be split using the Parentcat feature. All the basic classification methods will be applied to each genre dataset so that performance of all methods on every dataset can be compared. For each method, we use cross validation when training, calculate confusion matrix as well as ROC curve on test datasets. After presenting and discussing results for each method, there is a comparison of all methods at the end. We take party dataset as an example to illustrate all classification techniques and thus only display the result for party category. All other confusion matrix and accuracy reports can be found in Classification.txt.

1) ***Decision tree***:

Model setting:

To control the depth of the tree, we set max depth of trees as 4. When training, cross-validation is used. After that, we use another dataset to test the model and get the confusion matrix as well as ROC curve.

Input Features:

duration_ms, popularity, acousticness, dancebility, energy, instrumentalness, key, liveness, loudness, mode,speechesness, tempo, time signature, valence

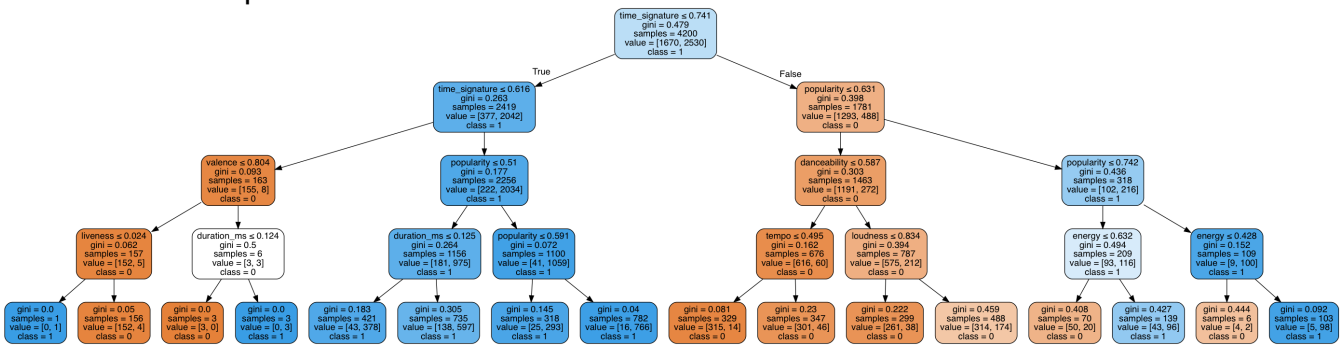
Class:

parentCat. 1 represents the case belong to the specific genre, 0 represents the case doesn't belong to the specific genre.

Result:

Tree Presentation:

Since there are 29 datasets after splitting based on playlist categories (genres), 29 trees have been created. Here we will only present the result for the category Party(zoom to have a closer look), all other results can be found in the file folder DecisionTreeGraph.



Decision tree model is very intuitive and easy to interpret. In the tree of Party tracks, time signature is chosen to be a root node, so this feature will give most information gain when splitting. From the tree, lower time signature is more likely for party music. Moreover, song tracks with popularity less than 0.631 are not in party playlist most of time. Thus, this model builds exclusive classification rules to separate data. For example, when time signature<0.616 & popularity<0.51(Normalized data), the label party can be assigned to the case.

Confusion Matrix on the test dataset

	Predicted Negative	Predicted Positive
True Negative	605	127
True Positive	119	949

Test data reports

	Precision	recall	F1-score	support
0	0.84	0.83	0.83	732
1	0.88	0.89	0.89	1068
avg	0.86	0.86	0.86	1800

2) K nearest neighbor

Model setting:

In k nearest neighbor, we set K=5(default).

Input features:

duration_ms, popularity, acousticness, dancebility, energy, instrumentalness, key, liveness, loudness, mode,speechesness, tempo, time signature, valence
Class:
parentCat: 1 represents the case belong to the specific genre, 0 represents the case doesn't belong to the specific genre.

Result:
Confusion Matrix on the test dataset

	Predicted Negative	Predicted Positive
True Negative	565	167
True Positive	110	958

Test data reports

	Precision	recall	F1-score	support
0	0.84	0.77	0.80	732
1	0.85	0.90	0.87	1068
avg	0.85	0.84	0.84	1800

3) Naive Bayes
Model setting: Assume all feature are independent.
Input features:
duration_ms, popularity, acousticness, dancebility, energy, instrumentalness, key, liveness, loudness, mode,speechesness, tempo, time signature, valence
Class:
parentCat. 1 represents the case belong to the specific genre, 0 represents the case doesn't belong to the specific genre.
Result:
Confusion Matrix on the test dataset

	Predicted Negative	Predicted Positive
True Negative	573	159
True Positive	140	928

Test data reports

	Precision	recall	F1-score	support
0	0.80	0.78	0.79	732
1	0.85	0.87	0.86	1068
avg	0.83	0.83	0.83	1800

4) SVM
Model setting: Linear kernel will be used.
Input features:
duration_ms, popularity, acousticness, dancebility, energy, instrumentalness, key, liveness, loudness, mode,speechesness, tempo, time signature, valence
Class:

parentCat. 1 represents the case belong to the specific genre, 0 represents the case doesn't belong to the specific genre.

Result:

Confusion Matrix on the test dataset

	Predicted Negative	Predicted Positive
True Negative	397	335
True Positive	95	973

Test data reports

	Precision	recall	F1-score	support
0	0.81	0.54	0.65	732
1	0.74	0.91	0.82	1068
avg	0.77	0.76	0.75	1800

5) Random Forest

Input features:

duration_ms, popularity, acousticness, dancebility, energy, instrumentalness, key, liveness, loudness, mode,speechesness, tempo, time signature, valence

Class:

parentCat. 1 represents the case belong to the specific genre, 0 represents the case doesn't belong to the specific genre.

Result:

Confusion Matrix on the test dataset

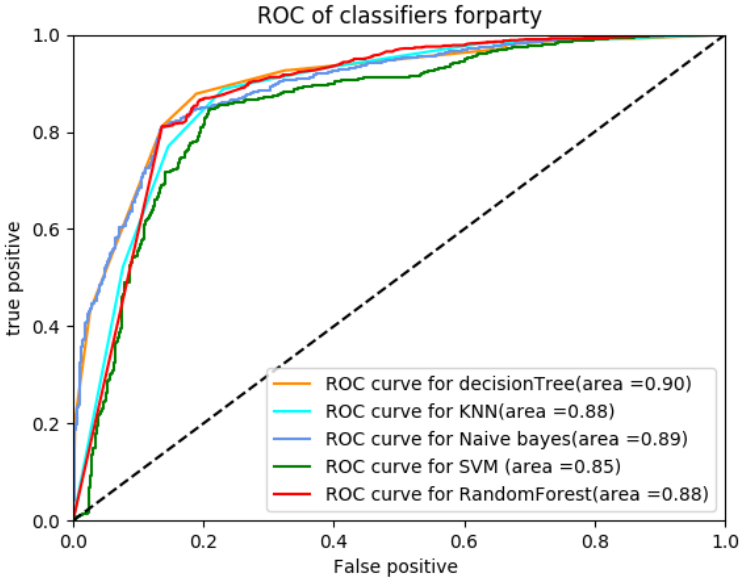
	Predicted Negative	Predicted Positive
True Negative	631	101
True Positive	193	875

Test data reports

	Precision	recall	F1-score	support
0	0.77	0.86	0.81	732
1	0.90	0.82	0.86	1068
avg	0.84	0.84	0.84	1800

ROC curve comparison and performance discussion:

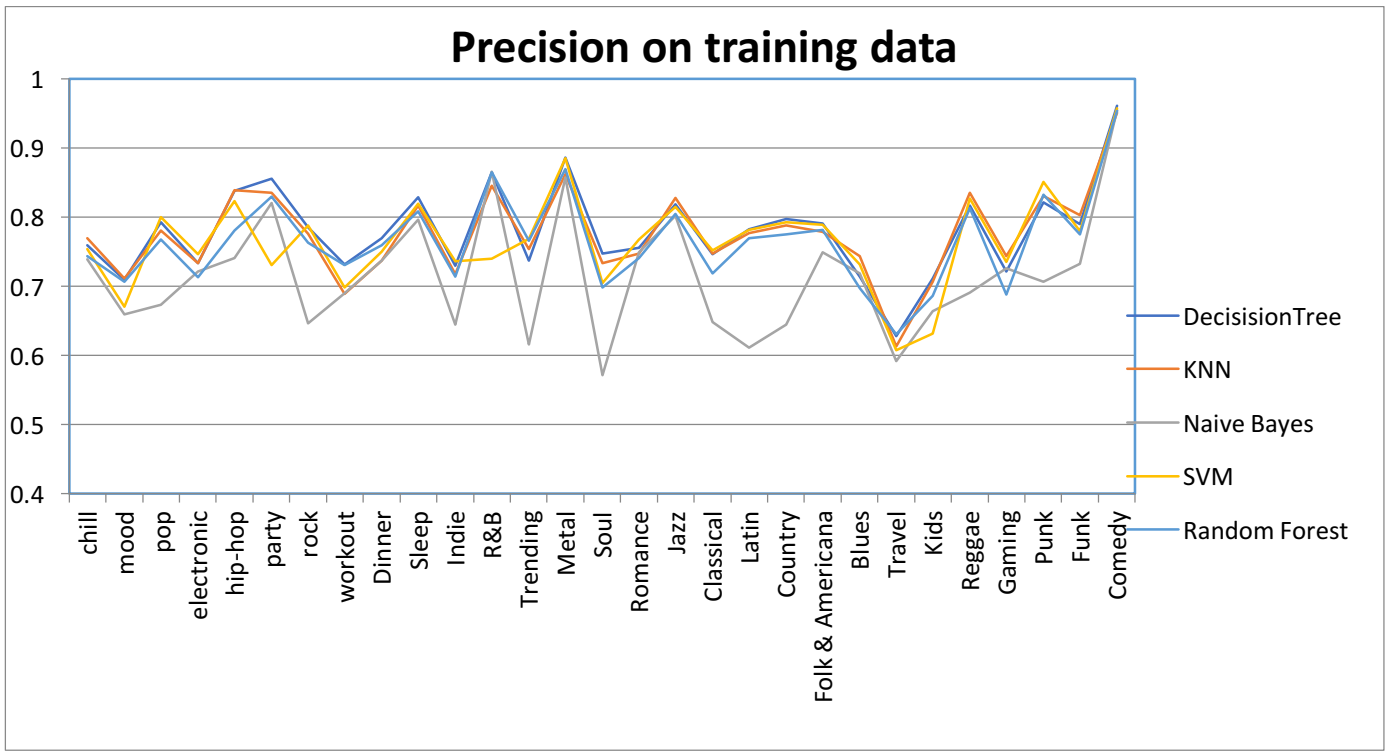
ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The larger the area under curve, the better the classifier is. Here we compare the performance of 5 classifiers on the party dataset using ROC curve. All other ROC curve plots can be found in the file folder ROCCurve.

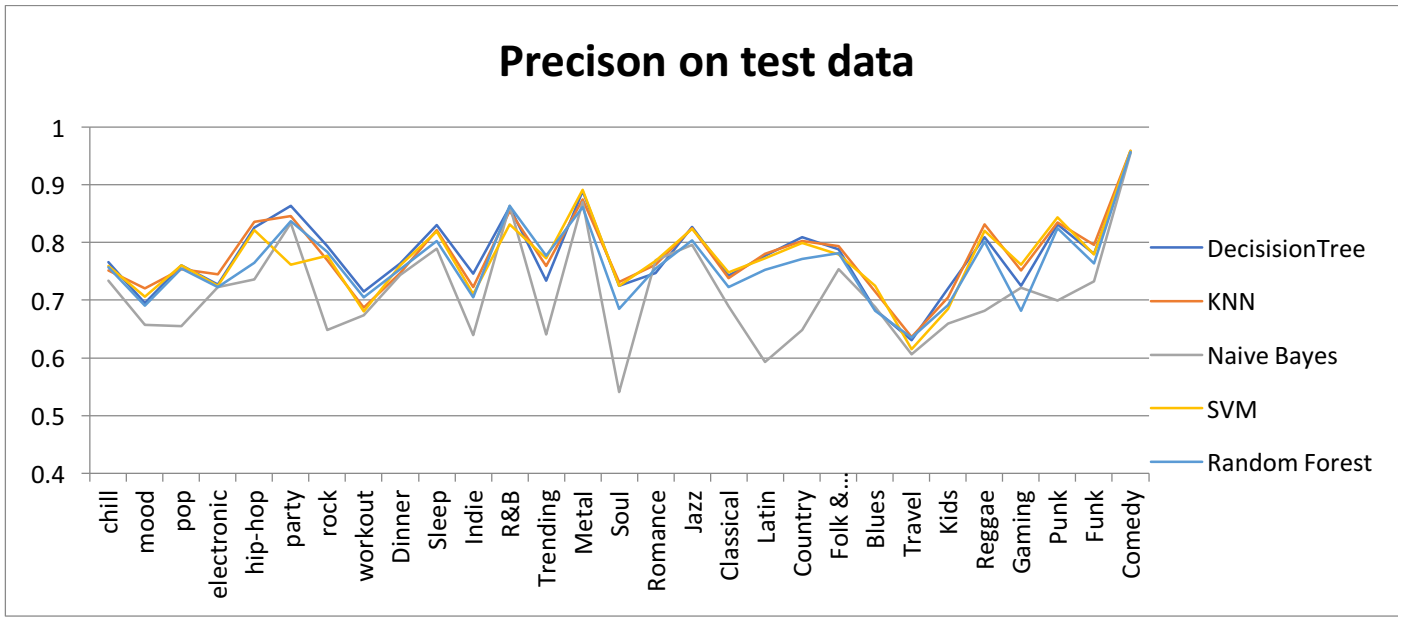


From the graph above, all the classification methods have good performance on the dataset, with the area under curve from 0.85 to 0.9. From the plots and tables above, it is evident that decision tree worked the best out of all of the algorithms that we implemented. However, all of the algorithms were still fairly successful classifying the genres considered. What is unexpected is that SVM has less satisfactory accuracy. Generally, SVM is a powerful classifier. The reason that it is inferior to other methods might because linear kernel is not a good hyperplane for this dataset.

Overall comparison:

In order to have a closer look at classification results, we take the party dataset as an example above. Next, we will give a summary of 5 classification methods on all genres datasets. To evaluate the performance for every classifier, precision on both training and test datasets are displayed.





In training and test data, classification methods have different performance in different genres. Some genres have high accuracy while others not. This may come from the characteristics of each genres. For example, Comedy music has highest accuracy. Some features such as speechiness significantly distinguish with other genres, making it is easy to be identified. This is consistent with common sense.

Generally, naïve bayes performes badly compared with others. Based on Bayesian theorem, it requires features are independent. However, the previous analysis shows there are closely related features in our dataset. This undermines the power of Naïve bayes. Other classification methods are all fairly successful for the classification of genres.

In summary, we conduct a series of binary classification using 5 methods over 29 datasets. Full results are in attached file folders. The classification results are relatively strong considering datasets have balanced classes. All classification methods work well except that Naïve bayes are less accurate.
Accuracy might be further improved if some advanced techniques are introduced. This will be the future work.

Conclusion

We have confirmed the predictive power of audio features on music genre throughout our analysis and selected suitable classification techniques. Our next step will focus on incorporating music ranking with audio features. We aim to come up insights about music trending with respect to artist, music genre, and audio features.