Musical Genre Classification by a Machine

IST 707 - Appplied Machine Learning

Kevin Harmer

kdharmer@syr.edu

May 11th, 2021

Abstract

Taste in music is unique to all listeners. With the hundreds of thousands of musical pieces developed over the last few centuries, however, it can be difficult to find a group of similar songs that any one person may like. With the development and advancement of machine learning algorithms, the difficulty of finding similar music is much easier to accomplish. With musical tracts broken into data, different algorithms can be used to tract similarities in songs that may appeal to listeners. With genre classification as a starting point, this paper addresses the difficulty of musical taste predictions. With the help of machine learning, genre prediction, and later musical taste prediction, is within reach.

1 Real World Problem

The music industry is a rapidly evolving environment. New artists, songs, and albums are developed daily. With practically an infinite amount of songs, it can be very hard to find new music without personal recommendations or listening to each song one by one. This project will develop an algorithm to classify songs in a fashion that relates similar music for the purpose of its listeners.

One way to address this issue is to classify songs into different genres. Genres categorize large amounts of songs and artists into a specific style of music, which include fields like Rock, Pop, Rap, Country and more. Understanding one's favorite genre has generally been a good starting point for understanding personal music interests.

There are other entities which lead to one's core interest in a song. Some deal with the melodic influence (how the music sounds), others preferences are related to the rhythmic influence (like tempo or beat), while others simply rely on the set of lyrics being sung. These ways, among others, require much larger amounts of data and larger algorithm computation strength. Due to this project's timeline only lasting a month, it will only deal with genre classification of songs and artists.

2 The Data

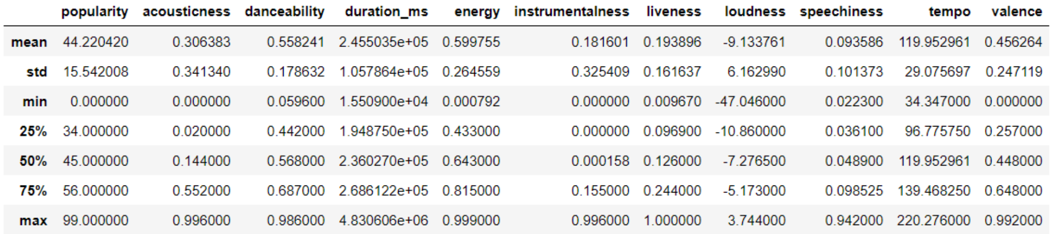
Data Set: https://www.kaggle.com/datasets/vicsuperman/prediction-of-music-genre

- instance_id: unique integer value for song's ID
- artist_name: name of artist that performs song
- track_name: name of the song
- popularity: how popular the song is (scale of 0 to 99)
- acousticness: how acoustic the song is (scale of 0 to 1)
- danceability: how danceable the song is (scale of 0 to 1)
- duration_ms: how long the song is in ms
- energy: energy feeling for the song
- instrumentalness: how instrumental the song is (on scale from 0 to 1)
- key: musical key the song is in
- liveness: how lively the song is (scale of 0 to 1)
- loudness: how loud the song is (in decibals)
- mode: major or minor key
- speechiness: proportion of speech (scale of 0 to 1)
- tempo: speed of the song (beats per minute)
- obtained_date: date when analyzed and added to data set
- valence: the positve or negative feeling assocaited with song (scale of 0 to 1)
- music_genre: genre of song

The data set has 18 columns over 50005 observations. Upon initial examination of the data set, it is not fully cleaned. There are 5 NA's and 4 duplicates. These NA values were found to be entire rows which also caused the duplicate reading. Removing the NA's resulted in 50000 observations, which were evenly distributed into each genre. Therefore, each one of the 10 genres saw 5000 different songs in this data set.

Once NAs were removed, summary statistics were observed, resulting in two notable issues. First, the tempo column was non-numeric. After looking deeper into the column, there were several entries with a question mark ("?"). These values were converted to the average tempo for the observation's artist (or the total average if it is the only observation for the artist). Second, there were nearly 5000 entries with time duration of -1 ms. The column was filtered similarly to the tempo column; the average tract time for the artist replaced the -1 ms (or data set average for artists with only one observation).

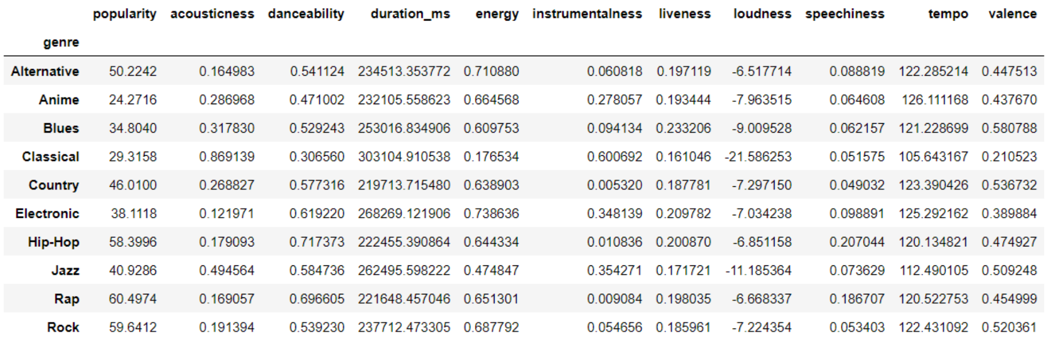
Focusing on the numeric variables, the summary statistics by genre are shown in the outputted python table below.



	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
mean	44.220420	0.306383	0.558241	2.455035e+05	0.599755	0.181601	0.193896	-9.133761	0.093586	119.952961	0.456264
std	15.542008	0.341340	0.178632	1.057864e+05	0.264559	0.325409	0.161637	6.162990	0.101373	29.075697	0.247119
min	0.000000	0.000000	0.059600	1.550900e+04	0.000792	0.000000	0.009670	-47.046000	0.022300	34.347000	0.000000
25%	34.000000	0.020000	0.442000	1.948750e+05	0.433000	0.000000	0.096900	-10.860000	0.036100	96.775750	0.257000
50%	45.000000	0.144000	0.568000	2.360270e+05	0.643000	0.000158	0.126000	-7.276500	0.048900	119.952961	0.448000
75%	56.000000	0.552000	0.687000	2.686122e+05	0.815000	0.155000	0.244000	-5.173000	0.098525	139.468250	0.648000
max	99.000000	0.996000	0.986000	4.830606e+06	0.999000	0.996000	1.000000	3.744000	0.942000	220.276000	0.992000

Figure 1: Summary Statistics of Prediction Variables.

This summary statistics only really show information about the individual variables. More information is likely contained in the variables by different genres. The genre average for each variable is shown in the outputted python table below:



	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
genre											
Alternative	50.2242	0.164983	0.541124	234513.353772	0.710880	0.060818	0.197119	-6.517714	0.088819	122.285214	0.447513
Anime	24.2716	0.286968	0.471002	232105.558623	0.664568	0.278057	0.193444	-7.963515	0.064608	126.111168	0.437670
Blues	34.8040	0.317830	0.529243	253016.834906	0.609753	0.094134	0.233206	-9.009528	0.062157	121.228699	0.580788
Classical	29.3158	0.869139	0.306560	303104.910538	0.176534	0.600692	0.161046	-21.586253	0.051575	105.643167	0.210523
Country	46.0100	0.268827	0.577316	219713.715480	0.638903	0.005320	0.187781	-7.297150	0.049032	123.390426	0.536732
Electronic	38.1118	0.121971	0.619220	268269.121906	0.738636	0.348139	0.209782	-7.034238	0.098891	125.292162	0.389884
Hip-Hop	58.3996	0.179093	0.717373	222455.390864	0.644334	0.010836	0.200870	-6.851158	0.207044	120.134821	0.474927
Jazz	40.9286	0.494564	0.584736	262495.598222	0.474847	0.354271	0.171721	-11.185364	0.073629	112.490105	0.509248
Rap	60.4974	0.169057	0.696605	221648.457046	0.651301	0.009084	0.198035	-6.668337	0.186707	120.522753	0.454999
Rock	59.6412	0.191394	0.539230	237712.473305	0.687792	0.054656	0.185961	-7.224354	0.053403	122.431092	0.520361

Figure 2: Average of Predicting Variables by Genre

The biggest differences in the data appear to be in the popularity and danceability columns. Just to take a look at these variables before moving into the data mining portion of the music algorithms, boxplots have been developed to showcase the differences seen by the descriptive statistics (shown below).

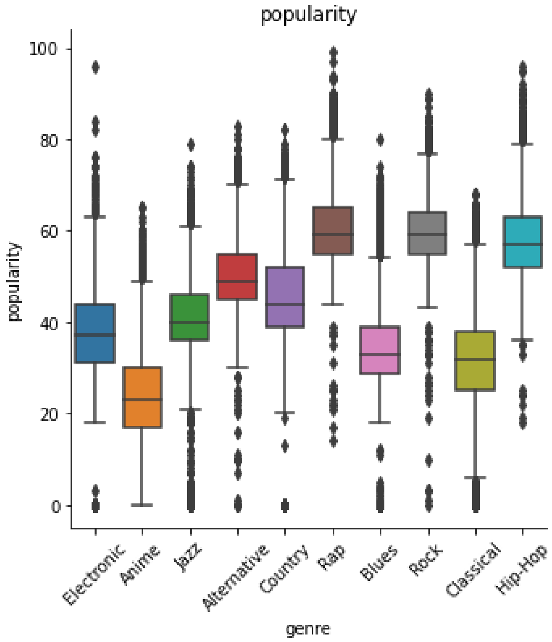


Figure 3: Popularity Boxplot by Genre

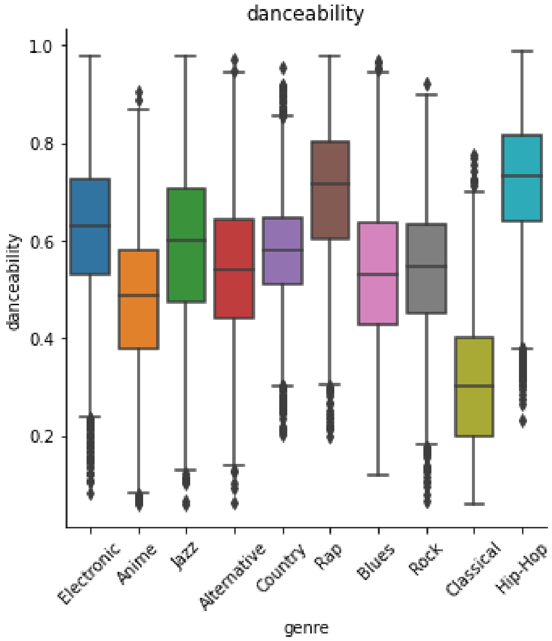


Figure 4: Danceability Boxplot by Genre

3 Data Mining Focus

To get valuable insight about prospective predictions, classification algorithms are the best option. K-Means Clustering will also be used to address the natural relationships between the song data. Theoretically, genres would a be good cluster diagnosis, which will be compared to the algorithm.

For clustering algorithms, the data set will be split into training and testing portions then compiled using support vector machines, decision tree classification and multinomial naïve bayes classification. SVM will be used due to the splitting nature of data vectors. The boundaries of numeric variables, in theory, would be properly showcased in an SVM model. Decision Tree Classification is used in a similar fashion; the close genres can be split using a line of a specific variable. Lastly, naïve bayes is a good model considering the data size, especially while focusing on a multinomial distribution with different genres.

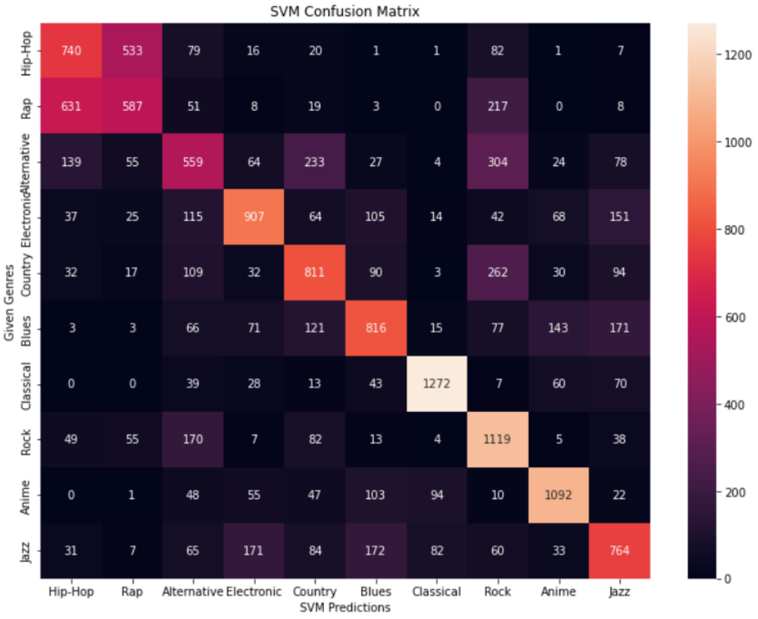
4 Machine Learning Algorithms

In preparation for the algorithms, the data was split into training and testing data sets. 70% of the data was incorporated into the training set and 30% into the testing set.

4.1 Support Vector Machines

Before getting into the actual SVM model, it is important to put the data through a vector transformation so that each value is on the same scale. Furthermore, principle component analysis was performed on the data set to break down the scaled vectors into more distinct normalized variables. Model tuning yielded a 0.534 cross-validation accuracy and fit a regularization parameter of 10.0 to a given linear model with gamma = 5.

After tuning, the model produced a training accuracy of 0.648 and a testing accuracy of 0.578. This accuracy is significant considering a random selection would only yield a 0.10 accuracy. The distribution of predictions is shown in the confusion matrix below. Furthermore, there is some over-fitting, but not too much where the model needs to be reworked. A linear kernel needed to be used for efficient performance; quadratic kernels in large data sets significantly increase training time for svm models.



	Hip-Hop	Rap	Alternative	Electronic	Country	Blues	Classical	Rock	Anime	Jazz
Hip-Hop	740	533	79	16	20	1	1	82	1	7
Rap	631	587	51	8	19	3	0	217	0	8
Alternative	139	55	559	64	233	27	4	304	24	78
Electronic	37	25	115	907	64	105	14	42	68	151
Country	32	17	109	32	811	90	3	262	30	94
Blues	3	3	66	71	121	816	15	77	143	171
Classical	0	0	39	28	13	43	1272	7	60	70
Rock	49	55	170	7	82	13	4	1119	5	38
Anime	0	1	48	55	47	103	94	10	1092	22
Jazz	31	7	65	171	84	172	82	60	33	764

Figure 5: Confusion Matrix for Support Vector Machines

After viewing the confusion matrix, the most notable amount of error is located in the Rap vs. Hip-Hop classification. It makes sense; these genres are similar to each other. Aside from these two, the model's over-prediction of rock, and model's under-prediction of alternative, the data was predicted correctly relatively consistently. There is a clear consistent prediction in each of the genres, especially classical music.

4.2 Decision Tree Classification

Tuning a decision tree classification model considered the depth, leaf split and the minimum samples per leaf, along with the criterion. After training across several parameters, the model found the best model to be trained with a 'gini' criterion, max depth of 10 (one for each genre), min samples per leaf of 2, and the minimum split samples to be 2. Other parameters can be used in future versions. The model tuning found a cross-validation accuracy of 0.455.

After fitting the data, the decision tree produced a training accuracy of 0.595 and a testing accuracy of 0.526. Again, not perfect considering the high amount of missed predictions, but definitely better than random prediction. The results were close to svm, but did not quite reach the accuracies found for the svm model. The distribution of predictions is shown in the confusion matrix below.

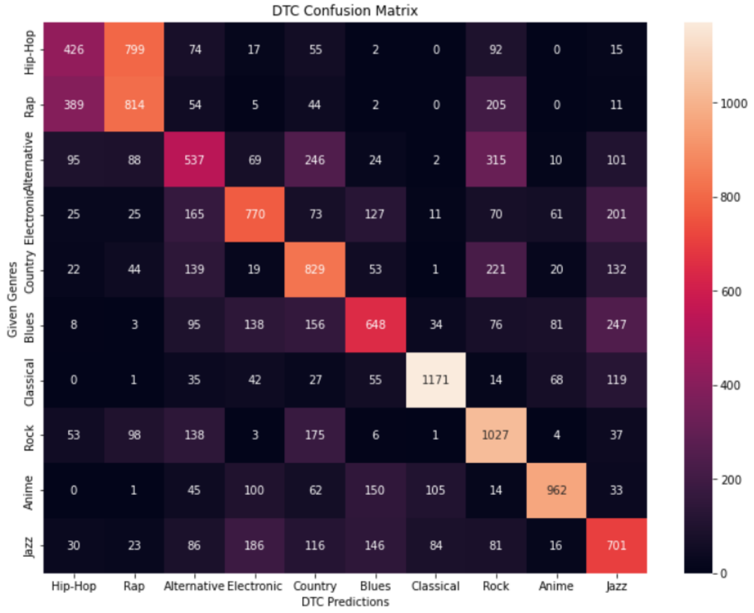
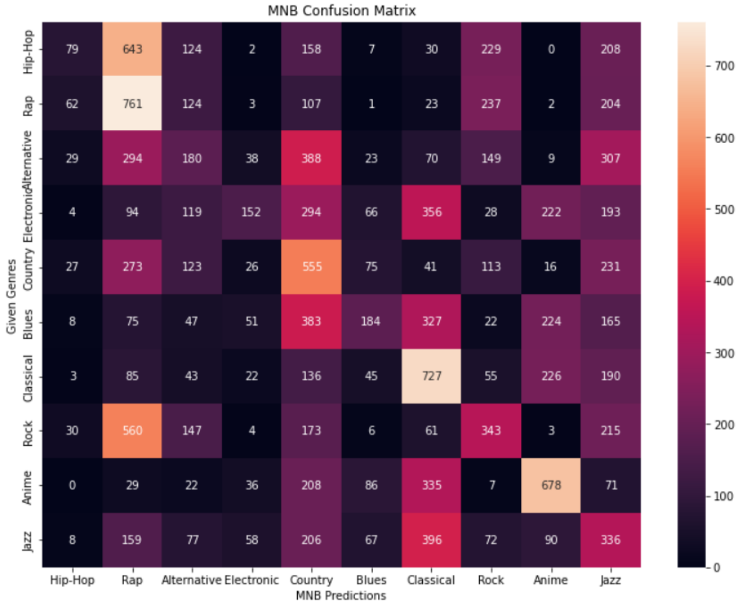


Figure 6: Confusion Matrix for Decision Tree Classification

Looking at the confusion matrix, the error distribution was relatively similar to the svm model. Classical was good, rock was over-predicted, and alternative was under-predicted. The rap and hip-hop error, additionally, was much worse, showing an extremely large amount of predicted rap songs which were actually hip-hop.

4.3 Multinomial Naïve Bayes Classification

The Naïve Bayes Classifier was conducted similarly to the last two models. Before moving into the actual model, loudness had to be removed from the data set due to Naïve Bayes inability to digest negative values. After brief tuning, alpha was set to be 0.001 with a cross-validation accuracy of 0.262. After training model, the Naïve Bayes method produced a training accuracy of 0.263 and a testing accuracy of 0.266. Consequently, this model is not a great predictor of genre given the numerical statistics. The actual distribution is shown in the confusion matrix below.



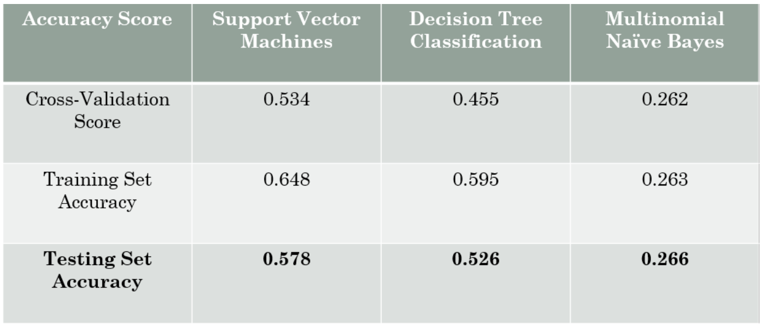
	Hip-Hop	Rap	Alternative	Electronic	Country	Blues	Classical	Rock	Anime	Jazz
Hip-Hop	79	643	124	2	158	7	30	229	0	208
Rap	62	761	124	3	107	1	23	237	2	204
Alternative	29	294	180	38	388	23	70	149	9	307
Electronic	4	94	119	152	294	66	356	28	222	193
Country	27	273	123	26	555	75	41	113	16	231
Blues	8	75	47	51	383	184	327	22	224	165
Classical	3	85	43	22	136	45	727	55	226	190
Rock	30	560	147	4	173	6	61	343	3	215
Anime	0	29	22	36	208	86	335	7	678	71
Jazz	8	159	77	58	206	67	396	72	90	336

Figure 7: Confusion Matrix for Multinomial Naïve Bayes Classification

The Naïve Bayes Confusion Matrix showcased the large amounts of error shown by the accuracy. Hip-hop, electronic, and blues saw very little predictions, while other genres, like rap, country, and classical saw many more predictions (which saw a large percentage of incorrect predictions). Otherwise, jazz saw a very high of incorrect predictions which is in line with the lower accuracies.

4.4 Overall Model Results

The accuracies, as discussed in the above sections, are shown in the table below.



Accuracy Score	Support Vector Machines	Decision Tree Classification	Multinomial Naïve Bayes
Cross-Validation Score	0.534	0.455	0.262
Training Set Accuracy	0.648	0.595	0.263
Testing Set Accuracy	**0.578**	**0.526**	**0.266**

Figure 8: Model Accuracies

Comparing the models, SVM proved to be the strongest method in genre classification, with a top training score 0.578. Decision Tree Classification also showed decent results, with a testing accuracy of 0.526. The difference in these two was likely due to the high similarities between different genres (ex. rap and hip-hop). Because it relies on distinct differences, the decision tree classifier showed larger amounts of error when it came to extremely similar genres.

The Naïve Bayes method ultimately failed genre prediction. This is likely due to the possible co-dependence between some variables (ex. danceability and energy), thus breaking the Naïve independence assumption. The other two models, on the other hand, did were successful in predicting a good percentage of genres given the input information.

4.5 Clustering Analysis: K-Means

To take advantage of numerical data and unsupervised machine learning, a K-Means clustering algorithm is used to analyze what the data is naturally gathered together. In theory, the development of genres were designed to categorize similar portions of music. Machine learning should similarly develop genre categories.

Consequently, a K-Means algorithm, with 10 clusters, was fit to the entire numerical data set. After using the model to make predictions on the same dataset, the results produced were very asymmetric. Clusters were generated to contain 5 observations in one group then 16281 observations in another. These were very different from the 5000 genre observations per genre. To confirm that it was not the amount of clusters (possible sub genres), the model was tested with 6-14 clusters, with similar results. As a result, it is confirmed that genre may not be the best classification for music quality, which may be part of the reasoning for the previous model's error.

5 Conclusion and Future Explorations

Overall, these models do contribute to the music genre prediction. Despite the nature of the clusters and the weaker Naïve Bayes model, SVM and Decision Tree Classification showed promising results. With more specific data (ex. audio files) or more computing power dedicated to training a non-linear svm model, results would likely be much stronger, as seen in some of the top musical streaming apps.

Yet even with stronger models, the accuracy of predictions is still limited. Genres are the result of human opinion without consideration of specific sound components. Factors such as artist and album likely play a much bigger role in genre classification, which may be incorporated in the future using association rule mining. Furthermore, incorporation of the categorical variables would also improve the models' accuracies.

Aside from model improvement, there are other considerations that could be related to these models. By developing a ”musical taste,” like a set of factors which can define a song, can be placed into these models and receive a predicted genre. This basis, when expanded into a more specific prediction process (achieved by more data and much more computing power), is the process of music recommendations by the today's top music streaming apps.