STATS 503 Final Project

Members: Katherine Ahn, Haonan Feng, Diana Liang, Karen Wang

Introduction

Music genre is a fluid categorization of expected experiences. The songs that belong in each genre and whether new genres or subgenres should be created as a response to new cultures are hotly debated. Defining these arbitrary borders becomes more complicated as the world becomes more interconnected and the basis of experts widens. Yet many proclaim devotion to certain genres and pass judgement on others. Clearly these borders do exist even if they are elusive. Such a complicated and computation intensive task, though not impossible for humans, may best be suited by computer models.

Project Statement

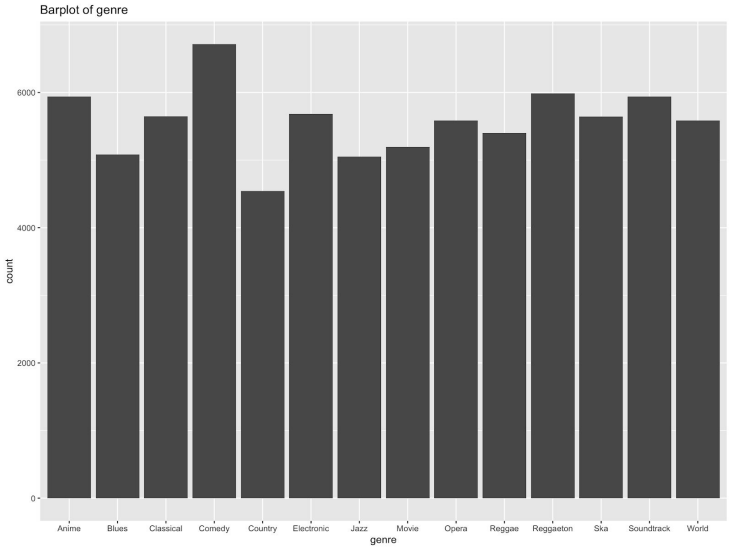
Our goal is to fit a model to determine music genre from a multitude of predictors. The data, from Kaggle, contains 18 different variables that detail the 232,725 observed tracks as described and computed by Spotify. The response variable, naturally, is the categorical variable 'genre'. The 14 predictor variables range from continuous to categorical: 'popularity', 'acousticness', 'danceability', 'duration', 'energy', 'instrumentalness', 'key', 'liveness', 'loudness', 'mode', 'speechiness', 'tempo', 'time signature', and 'valence'.

Since so many types of models exist, we will compare a multitude of models to find the one with the best performance: The methods to be explored are: random forests, adaboosting, logistic regression, MLP, LDA, QDA, and KNN.

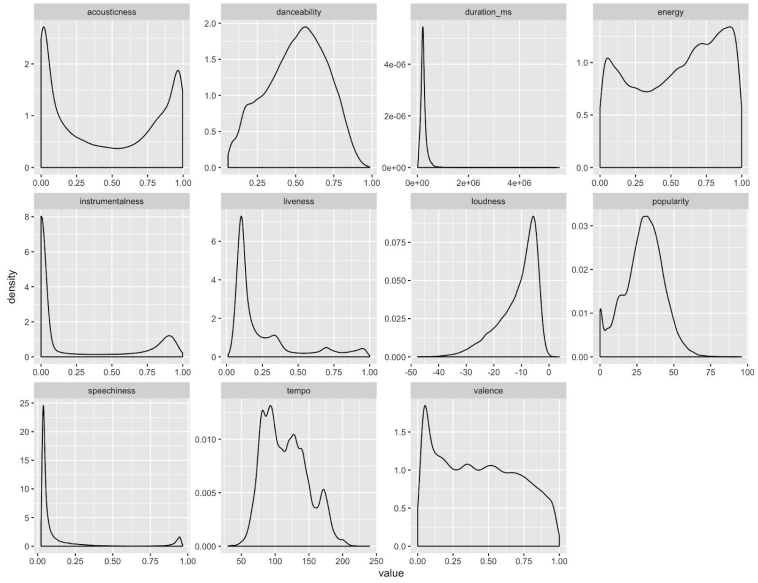
Data

For the exploratory data analysis, we wanted to have similar population size among the different genres. Thus, we removed the genres of significantly lower frequency from the rest of the genres and merged the genres that were differentiated only by spelling. The resulting frequencies of each genre are shown in the first plot below.

Barplot of genres:

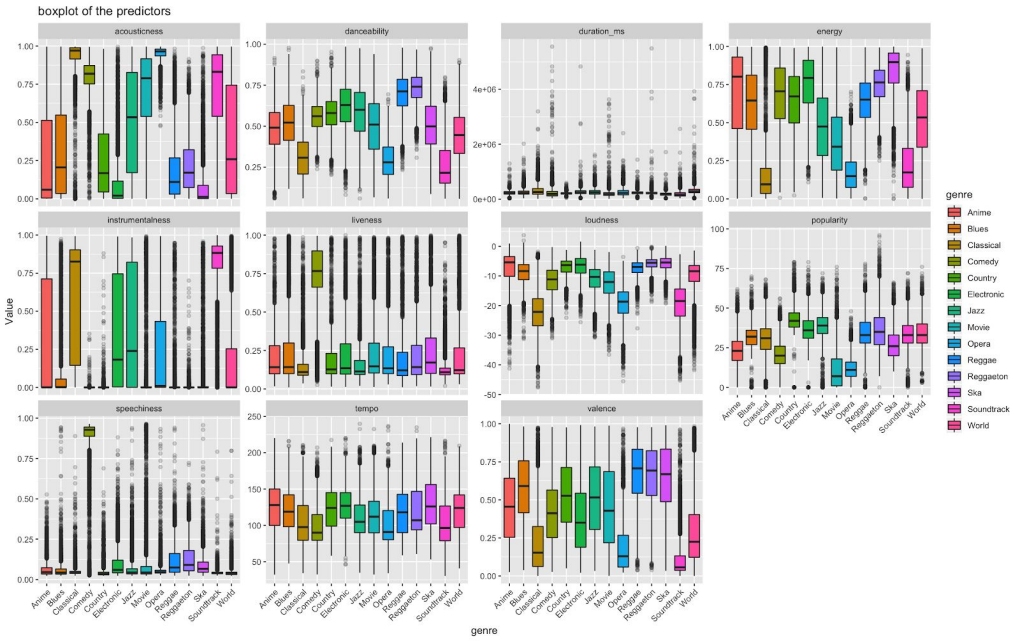


Distributions for all the quantitative predictor variables:



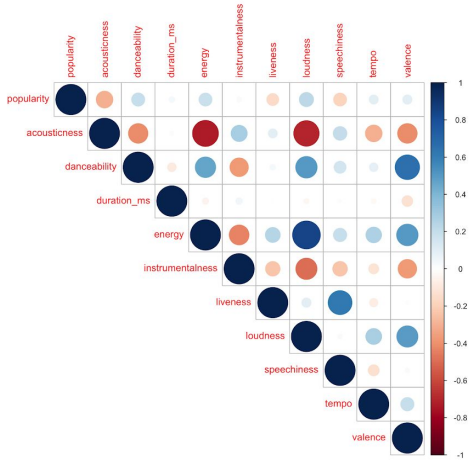
Many of the predictors have skewed and multimodal distributions, hopefully providing crucial information to accurately classify between the genres.

Box plot of the quantitative predictors:



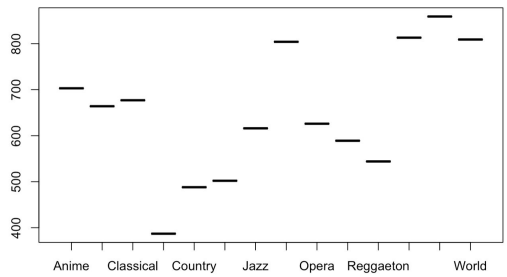
By looking at the above boxplots, we anticipate most of the predictors except ‘duration_ms’ and ‘tempo’ would be useful in classifying genres. ‘Liveness’ and ‘speechiness’ are expected to play a big role in classifying comedy.

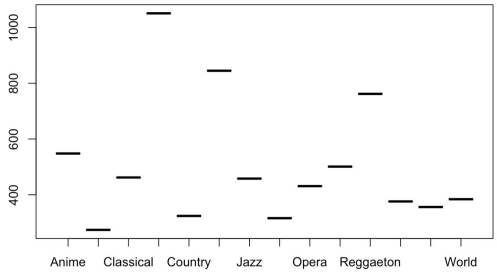
Correlation plot of the entire dataset, disregarding the genres:



We also wanted to see the correlation between our independent variables. We see that 'energy' and 'accousticness' are negatively highly correlated; inversely, 'energy' and 'loudness' are positively correlated, which corroborates common knowledge. 'Duration', on the other hand, has the least correlation with the other variables.

Frequency plots on key C and C#:





This is an example of the frequency on different keys. Certain genres might have a favor in leaning towards a particular key.

Model Fitting (Methodology)

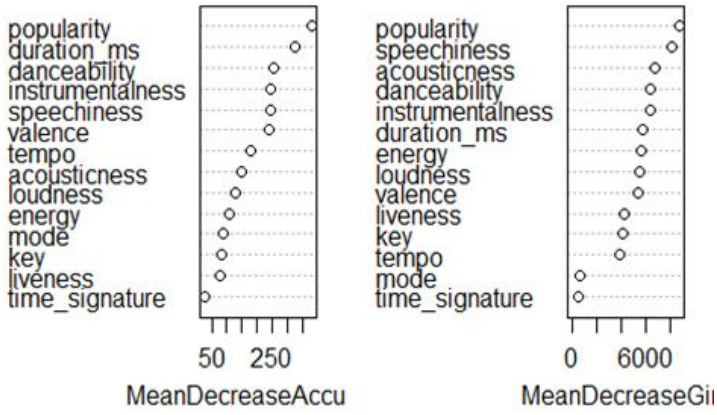
A. Random Forest

We used the *randomForests* package to fit a random forest to the data. Since classification trees can take in different types of data, the entirety of the dataset was used in model fitting.

To improve performance we compared the training error for a multitude of parameters and chose those with the lowest training error. The main focus was on controlling model complexity, so the parameters of interest were the number of trees and the number of predictor variables considered for each node.

The best model came from 1000 trees and 3 predictor variables, which provided a test error of around 31.5%. The variable importance plot below shows that the most important

variables in predicting genre are: popularity, speechiness, danceability, instrumentalness, and duration.

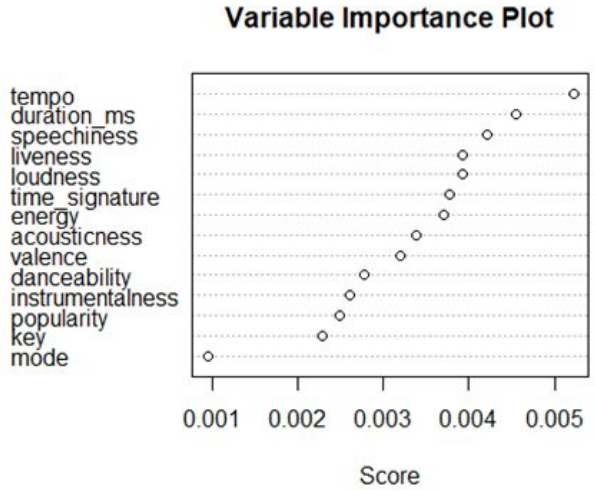


B. Adaboost

Since the usual methods of Adaboosting only work for 2 classes, we used the maboost package for multiclass classification with adaboost. Once again the entire train dataset was used because boosting is based on classification trees.

The parameters of interest for this model were iterations and a weight parameter. While a multitude of parameter combinations were possible, each model usually took more than 3 hours to complete, limiting the number of combinations that were able to be tested for the smallest training error. Since most of the combinations tested had very similar training errors, the model with the smallest training error was not chosen due to intense computational costs.

The model chosen had 3000 iterations and a weight parameter of 0.5 with a test error of 42.8%. The variable importance plot below shows that the most important variables in predicting genre are: tempo, duration, speechiness, liveness, and loudness. Compared to the random forest model, the model performance was lower and the ranking of variable importance quite different.



C. Logistic Regression (LR)

We applied one-hot encoding for the categorical data and used 10-fold cross validation to choose the optimal inverse regulation strength (C) from {0.001,0.01,0.1,1,10}.

```
Using C =  0.001 CV error:  0.5301648513333513
Using C =  0.01 CV error:  0.554086535818428
Using C =  0.1 CV error:  0.5606378562438102
Using C =  1 CV error:  0.5619202348703095
Using C =  10 CV error:  0.5619203366919147
```

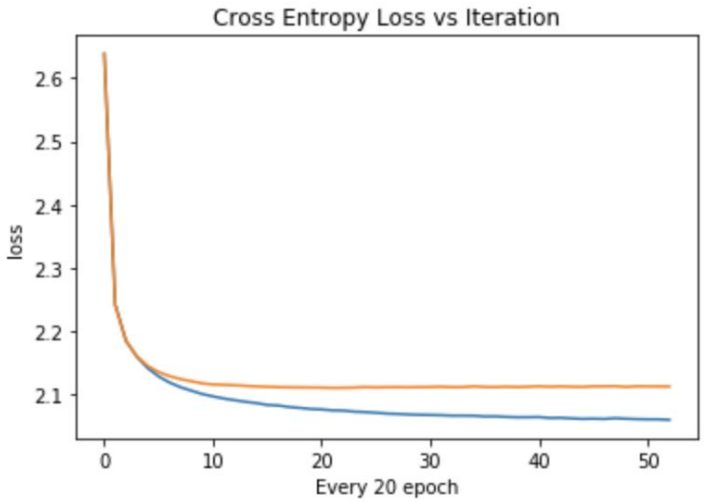
Based on the CV errors, we choose the logistic regression model with C = 1 for easier interpretation.

D. Multi-Layer Perceptron (MLP)

We applied one-hot encoding again and used three variations of MLP's, each with different hidden layers. A validation set is randomly splitted from the training set with a proportion of 0.25 for estimating the test error for the three neural networks.

1. Input -> Hidden 35 -> LeakyRelu -> BatchNorm -> Output -> Softmax
2. Input -> Hidden 60 -> LeakyRelu -> BatchNorm -> Output -> Softmax
3. Input -> Hidden 35 -> LeakyRelu -> BatchNorm -> Hidden 25 -> LeakyRelu -> Output -> Softmax

In each of the neural networks, we used Adam optimizer, set the learning rate at 0.01, and set the loss function to be cross-entropy loss. From the validation results, the second was chosen (with a validation error = 0.3598). We also plot the training loss (in blue) and validation loss (in orange) during training this network, to show that the loss achieves convergence.



E. Discriminant Analysis

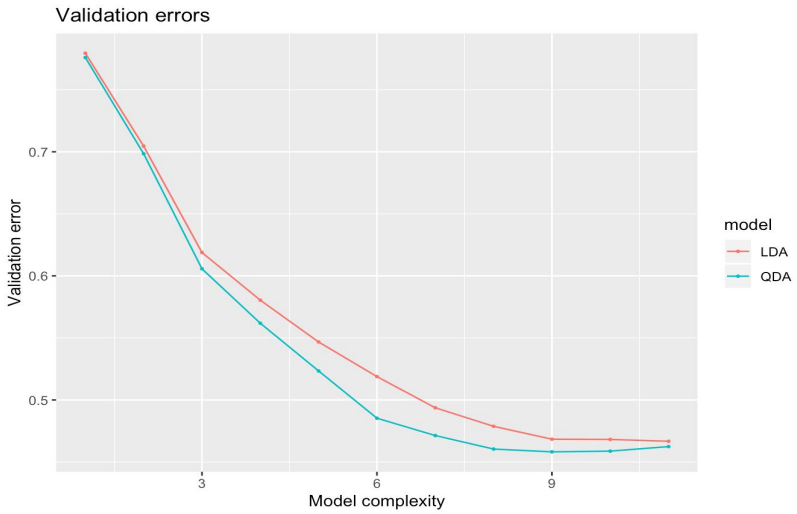
Discriminant Analysis assumes the distribution of the predictors given a class follows Gaussian distribution, so only the quantitative predictors are taken into account. The validation approach was implemented to find the optimal model complexity for LDA and QDA. The train and validation set were first divided 2:8. Then forward selection was used to find the best combination of predictors as well as to reduce computational cost. `duration_ms` predictor is

divided by 1000 to convert it to the `second` scale. And a log transformation was made after the scaling to make the density more normal.

The LDA model with all predictors has the lowest validation error of 0.467, but a simpler model with 9 predictors had similar performance. Since it's simpler, LDA with 9 predictors was preferred with: acousticness, popularity, danceability, speechiness, instrumentalness, valence, energy, duration_ms, and loudness.

QDA with 9 predictors has the lowest validation error of 0.458, but QDA with 8 predictors has similar performance (0.460). As the simpler model is preferred, QDA with 8 predictors was selected with: acousticness, popularity, danceability, duration_ms, energy, valence, speechiness and loudness.

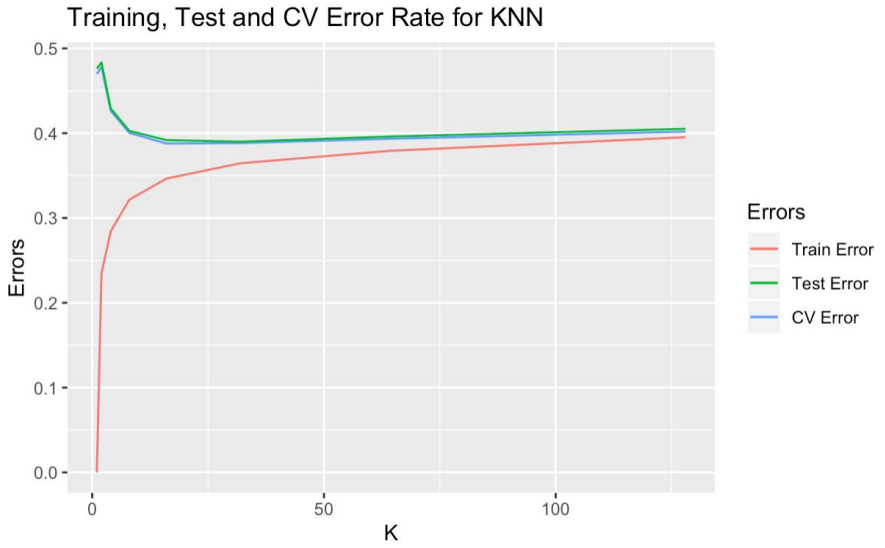
Lastly, as LDA and QDA are parametric classifiers, it did not take a lot of time to fit each model. However, the models are showing relatively high misclassification error as the predictors generally do not follow a normal distribution.



F. KNN

We used the KNN function from the 'Class' package. For the first model, we didn't take the categorical variables into account (key, mode, time_signature). Due to computational costs, we

used a 5 fold CV and a K values of 1, 2, 4, 8, 16, 32, 64, 128. The model with the lowest CV error



was with K = 32.

We then wanted to improve the model by adding the categorical variables. We converted the categorical variables into numerical so the variables could be scaled and added into the entire data set. By using the same number of folds and same values of K, the model did not vary much, and the CV error increased by about 0.025.

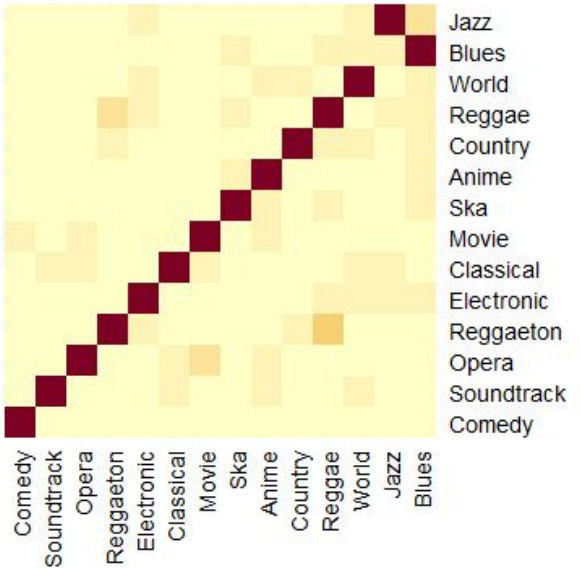
G. Model Comparisons

Model Type	**CV/Validation Error**	**Test Error**
Random Forest	0.316	0.315
Adaboost	0.425	0.428
Logistic Regression	0.438	0.436
MLP	0.360	0.363
LDA	0.468	0.468
QDA	0.460	0.464
KNN	0.390 0.415 (with categorical)	0.388 0.413 (with categorical)

Evaluation

We choose random forest as our final model, as it has the lowest test error. We expected random forest to have good performance since it's nonparametric and robust to outliers. The variable importance plot says that popularity, speechiness, danceability, instrumentalness, and duration are the most important

predictors in predicting genre. On the other hand, categorical variables (key, mode, time_signature) are the least significant.



The heatmap visualizes how each genre is classified using random forest. The horizontal axis indicates the truth genre and the vertical axis indicates the prediction. Our final model classified most of the genre correctly; however, there are several genres that were greatly misclassified: in particular, between jazz and blues; between reggae and reggaeton; and between movie and opera. This implies that these six genres are difficult to discriminate using the 14 predictors of our dataset.

Each model has a different computation cost when training the data. The time it took to fit a model is one of the measures of the computation cost. The computation cost of the models we attempted are as follows: Adaboost, MLP > RF > LR >> KNN > LDA/QDA. Although RF has a quite high computational cost, it is worth it for the better performance.

Conclusion

There are some limitations of the dataset. For one, we have 14 variables (or 31 variables if we do one-hot encoding), but we also have 14 classes to predict, making classification harder than expected. In addition, some other significant variables, such as timbre, might also be correlated to the genre but were not included as predictor variables. With consideration of the above difficulties, our models did a good job on this multi-classification task. The highest test accuracy was 0.7, which is quite good since most of the misclassifications are caused by the aforementioned problems.

In this project, we mainly compared the testing results from each of the methods we used. For future improvement, instead of taking out the genres, we could classify those genres with smaller amounts into an ‘other’ category or find more songs that would fit into those genres. This would increase the number of observations and genres and increase the scope of our models. We could also improve the models by removing highly correlated predictors and include predictors that would be better at differentiating the difficult genres mentioned above, such as jazz and blues. This would greatly decrease the number of misclassifications and the test error as a result. It would also be interesting to see if getting rid of the categorical variables would improve the best model, as both the variable importance plots and the high performance of the KNN without categoricals suggest that they aren’t as important in genre classification.

Our experiment implies that random forest has the best performance for this specific dataset. In conclusion, our experiment could serve as a ‘reference’ to quickly assign genres to new music on music storages like Spotify or to attempt a more global understanding of music from difficult cultures. Since prediction models are problem-specific and data-specific, we cannot make any inferences outside the scope of our data or comment on how genres should be defined. For that no model would be perfect.