Alec Arroyo

IST 707

Final Project

Music Genre Detection

Kaggle Dataset Source - <https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification>

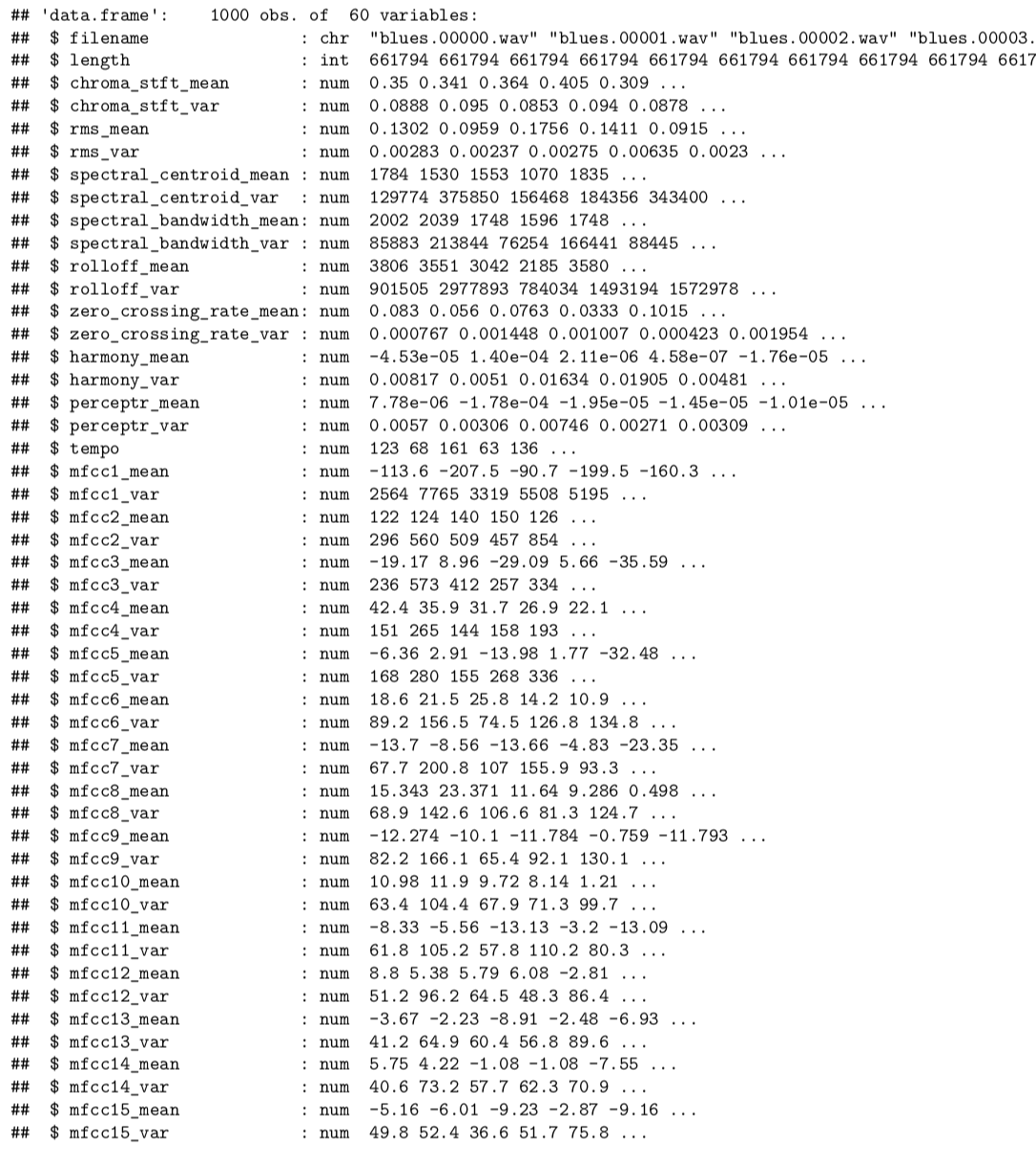
Introduction:

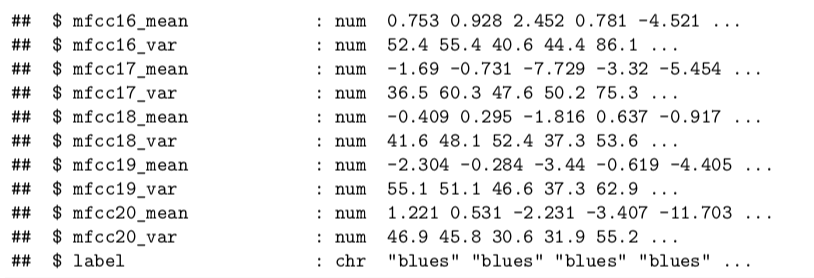
With many music applications, you notice all of them are sorted and organized by genre. Hundreds of thousands of playlists are created for millions of users based off of the types of music they like, for all formats of music. Apple music, Spotify, TIDAL, Soundcloud and many more music providers all use similar approaches when it comes to categorizing music to certain playlists. For my final project, I plan on using a dataset on Kaggle involving music data, and developing and training models to predict what genre of music a song is.

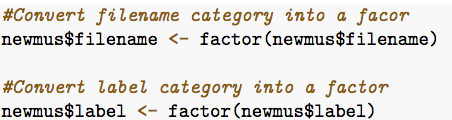
The data consists of 60 attributes that define different parameters of each audio file used such as harmony, roll-off, length of song, etc. The original source of the data is 30 second audio files each of different genres of music and multiple samples of each. They are categorized by the “filename” field that can tell you what type of audio file the record is. There is also a “label” field that categorizes what type of genre of music the file is. For example, for filename “blues.002.wav” the label that is defined for this record is “blues”. This attribute will be used to test and predict whether my models can correctly predict what type of genre of music a song is.

Data Preparation:

To start, the music data was loaded locally and a check was ran to remove nulls from the dataset using na.null(). When we inspect the structure of the data, we see that there are a couple of variables that we want to convert from a char data type into a factor data type to label them as categorical variables to apply on our models.

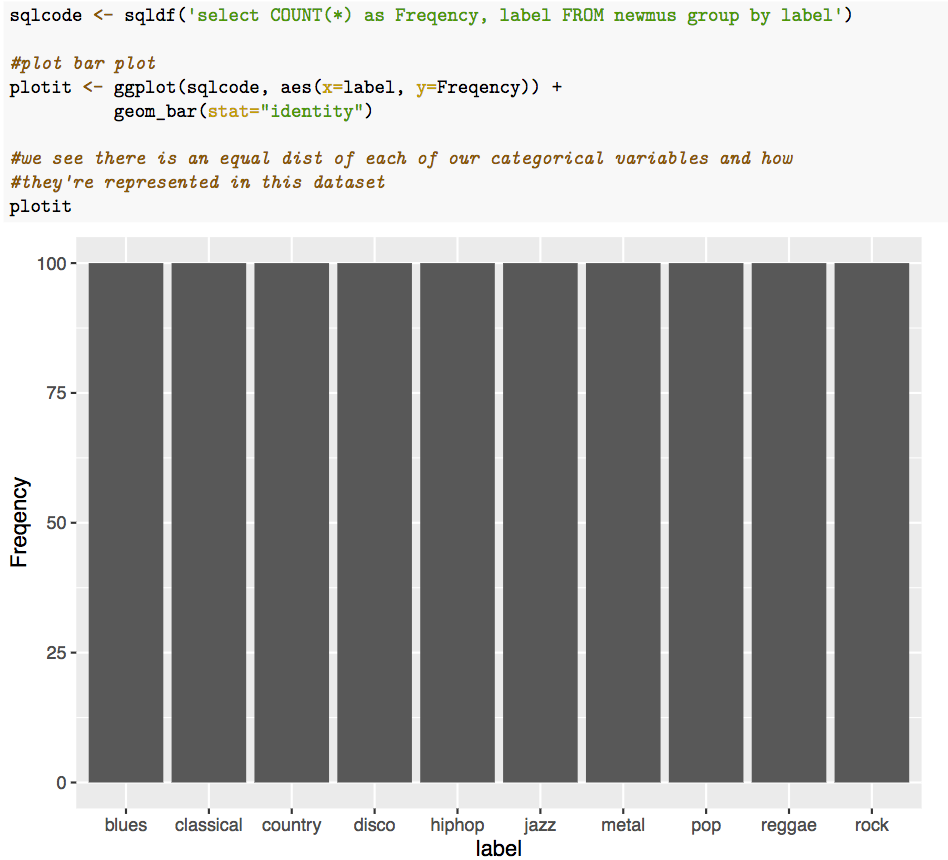




We first converted the filename label into a factor data type in order to run our models against the categorical variable that will determine what type of music genre the song is. In addition, we the filename field was converted into a factor data type as well for the same reason:  


Once we converted the correct fields to what they should be, a bar plot was created to see the distribution of each type of label of music for this dataset. In doing this, we can see if there is we see there is an equal distribution of each of our categorical variables and how

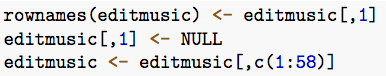
they're represented in the dataset. We used the sqldf() function to run a query on the dataset and gather a total count of each type of label, then plotted it to see the distribution.



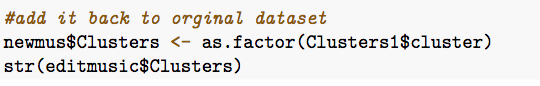
We can see that there is a great representation of each type of genre of music, where there are 100 songs for each genre. This is good since we want to run models with data that represent each type of music genre in a fair and equal way. The last thing that was needed to prepare the data, was to get rid of any nulls that are in the dataset. This is done because the models aren’t able to account for null information when creating and testing the data. The na.null() function was ran and it turned out there were no nulls in the dataset.

Analysis and models:

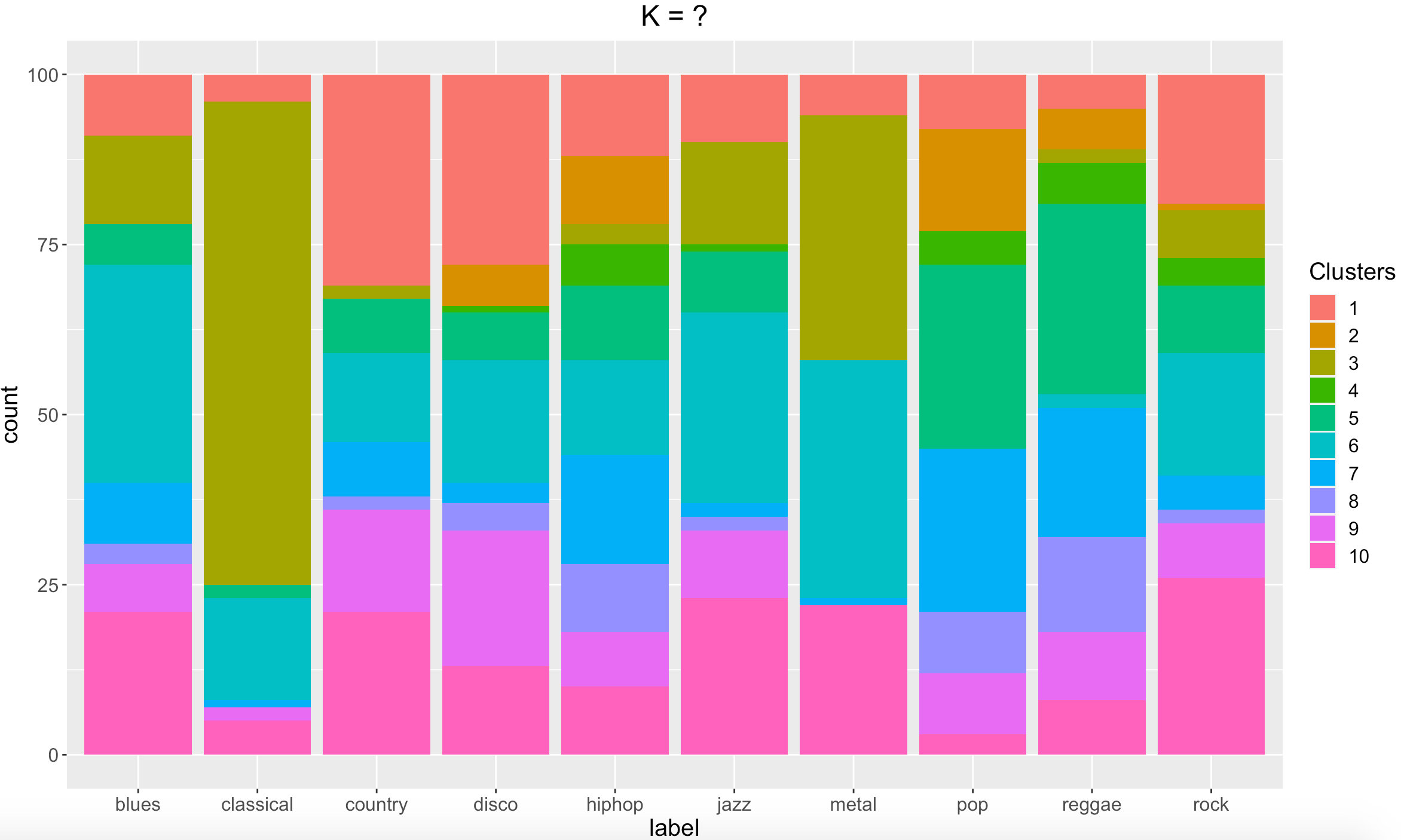
After the data was setup, the machine learning algorithms were applied on it. For the clustering algorithm, we wanted to remove the categorical field from the data to see what the clusters are predicting each song is classified as. We first created a new data frame the same as our original, and remove that “label” field.



Next, the data type for the “length” field needed to be changed to a data type of numeric in order to calculate the means for the clusters. We ran our kmeans() function that represents the clustering algorithm on our dataset and we set the amount of clusters equal to 10. The reason why the number of initial centroids to 10 was because we have 10 different genres of music represented in the dataset. In doing this, we can get a good representation of all the data by choosing a large number of centroids for the model. Once we ran our statement to get the clusters we added it back to the original dataset for comparisons.

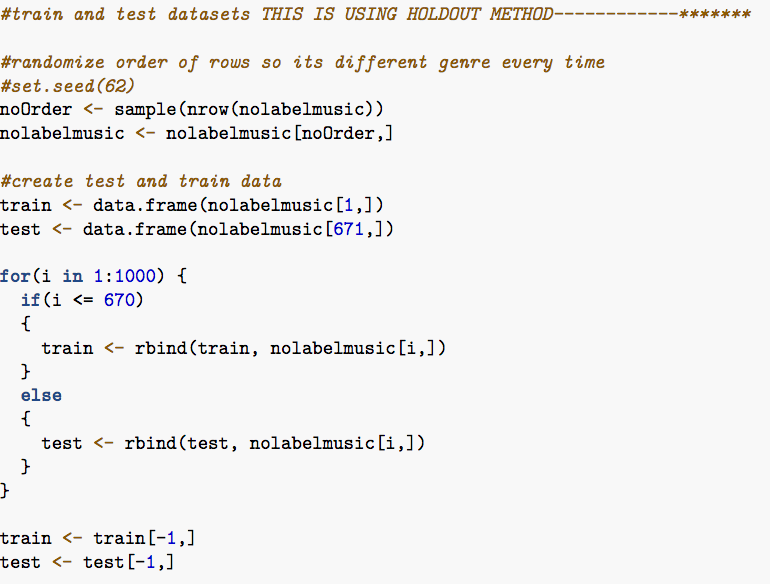


We then plot the clusters in a bar graph so we can see the clustering for each type of genre of music:

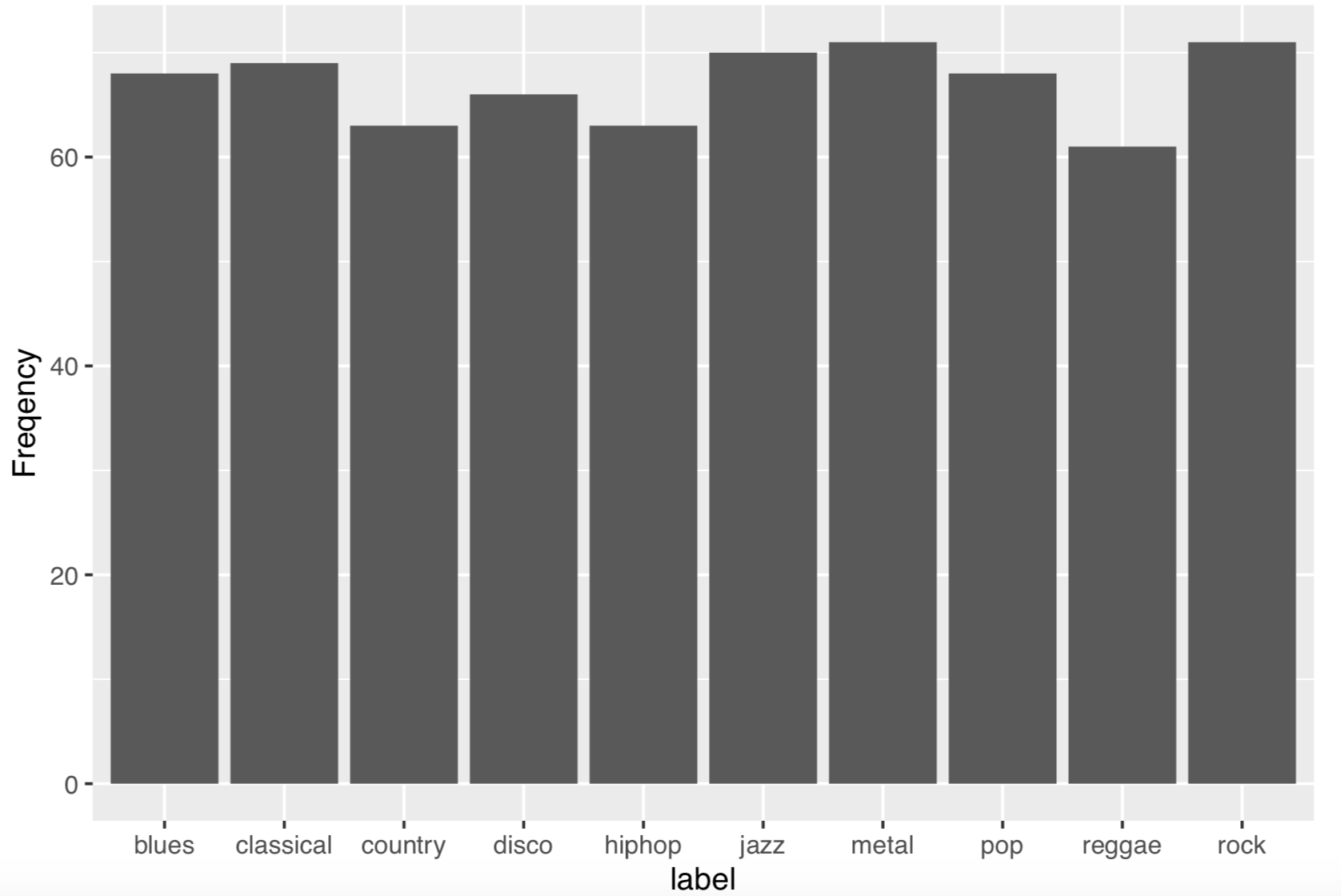


Here, we can see that the 3rd cluster seemed to have grouped records that were identified as classical music. In addition, the 5th cluster seemed to have populated the most for either pop or reggae. This could indicate that there may be similar values in numerous different fields that pop and reggae files share. From this same observation, we can also see that jazz and rock are highly clustered together from cluster 10. Some other notes are that the largest cluster for 7 was pop, for cluster 1 it was either country or disco, and for cluster 6 it was blues. Overall, it looks like it had some trouble determining the types of genre of music each file was besides classical music.

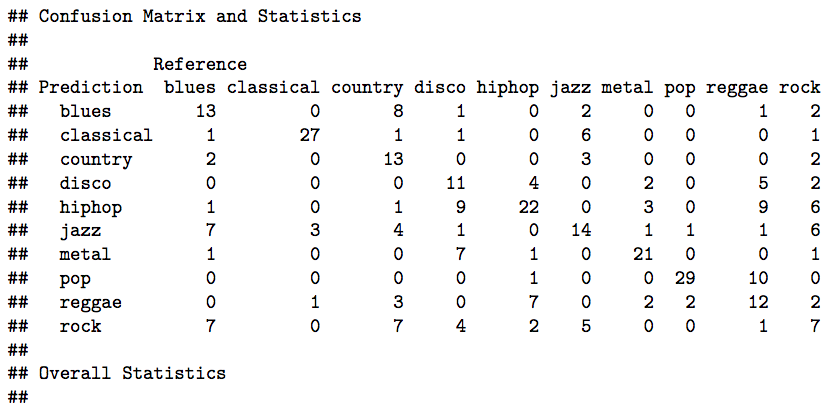
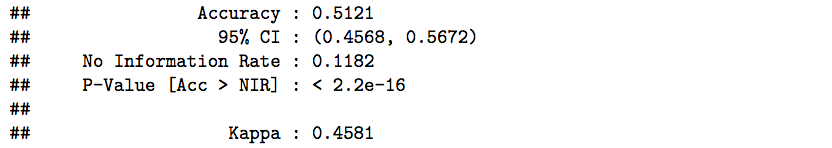
Since cluster modeling found it difficult to determine what genre of music each file is, some different models were applied to try to predict them. The next model that was tested was a decision tree model, where we would train and test the dataset to predict the label for the files. The original dataset was loaded again, and nulls were removed. Next was to create our training and testing datasets. For the decision tree model, the models were creating using the holdout method, where we randomly split the dataset into a training and testing datasets. Code was ran to randomize the dataset, and then split it 2/3 into the training dataset and 1/3 into the testing dataset:



Then another bar plot was ran on the training dataset to see what the representation of each genre of music was represented:



We can see that there is a good representation of each type of genre of music. This is very beneficial to us where our model can be used without any bias of one genre showing up way more than the others. The decision tree model was ran on the training dataset and predictions were tested against the trained tree model. After, a confusion matrix was created to see the accuracy of the model and how it performed on predicting the files:

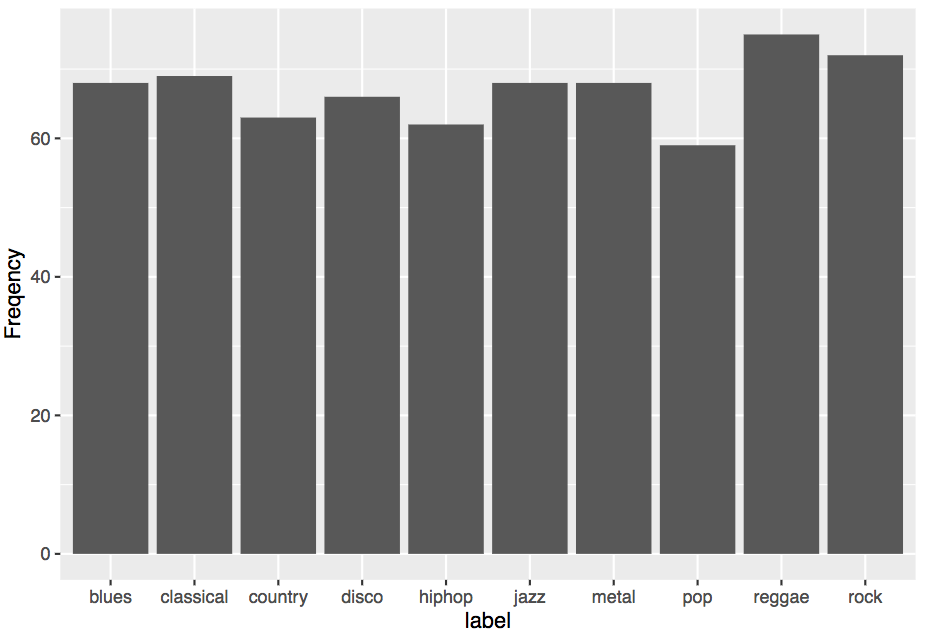
 

From the confusion matrix, we can see that the accuracy of the model is 51% which is relatively low in predicting the correct label for each file in the testing dataset. It looks like the music genres that were most successful in predicting were classical and metal music. The least successful music genres predicted were jazz and rock.

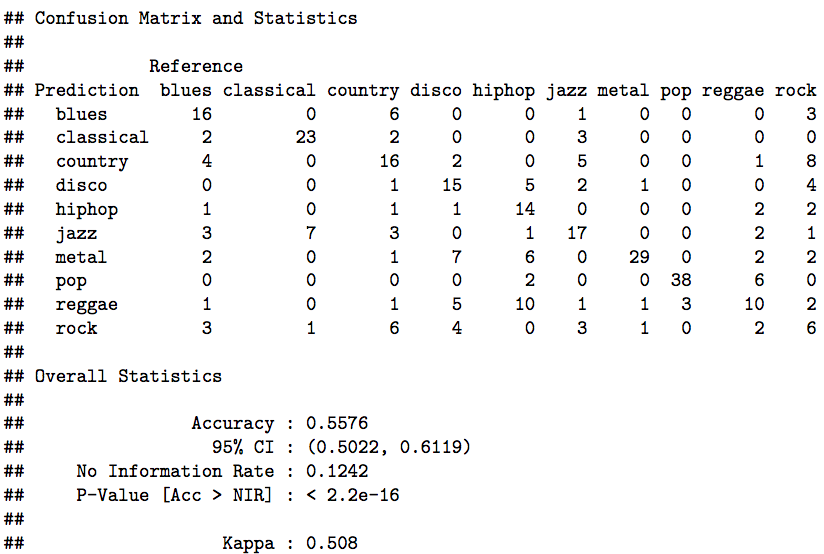
In an attempt to increase the accuracy of the decision tree model, a second model was created but with training and testing data created using cross validation techniques. Since this is a small dataset of 1000, we want to use something that will provide more variety in the training data to represent everything in the dataset. To start, we set the number of folds to 6 to gather a large sampling, and ran a for loop to run through the decision tree algorithm like so:



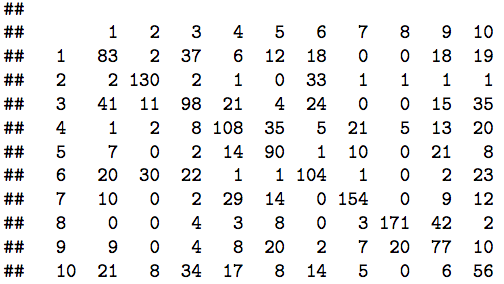
Another bar plot was then ran to see the representation of files for each type of genre of music.



It’s clear that there is a good representation of each type of music in the dataset that was used for cross validation. This will also show that there was no bias in the dataset, and the model had a fair representation of each category. A confusion matrix was created to see the accuracy of the model using:

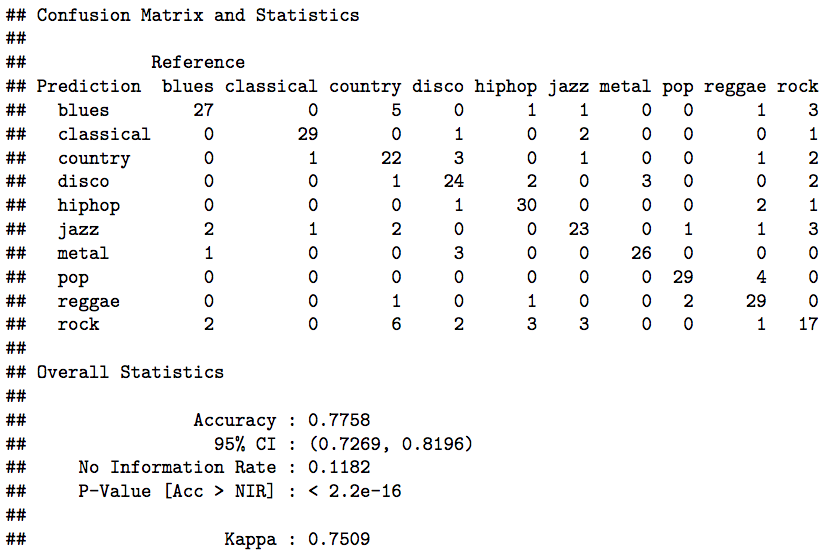


The decision tree model was more accurate when using cross validation at a percentage of 56%. Compared to the decision tree model predicting using the holdout method, it performed about 5% better in predicting the correct genre of music for each type of file. A table was then created to see the total number of iterations and the predictions for each.



It looks like the model did a good job predicting for numbers 2, 7, and 8, which represent classical, metal and pop. However, the model didn’t do a good job predicting for numbers 1, 9, and 10 which represent blue, reggae and rock. Like the holdout method, rock was the hardest genre to predict.

The final model that was ran was using a random forest model to predict the music genre for the testing dataset, and to see if the accuracy for the model was better than the other ones used. Using the train() function, the random forest model was applied onto the training dataset. Then predictions were then made based on that model on the testing data. After which, the confusion matrix was created to display the results:



Based on the matrix, the model was 78% accurate in predicting the genre of music for the files in the testing data. This very accurate and it looks like the genres of music predicted the most successfully were classical, hip hop, pop, and reggae. This least successful genres of music predicted correctly were country jazz and rock.

Conclusion:

The three models that were applied on the music dataset were clustering, decision trees, and random forests. Of the three models, the random forest model performed the best with a 78% accuracy. The two decision trees ran both using the holdout and cross validation techniques had accuracies of 51% and 56%. The clustering algorithm seemed to be not very accurate, where each cluster was almost even for every type of genre, and there weren’t any clusters that stood out for a specific genre of music besides classical. The decision tree algorithm had performed over 50%, and ran even higher when switching from holdout method to cross validation methods. The random forest model was significantly better, with the highest accuracy. One observation made was that the decision tree and random forest models did a good job predicting for music genres classical and pop. However, they also both struggled with predicting for music genre type rock. This could imply that the attributes used for this dataset weren’t helpful enough in validating a music file that is of genre type rock.

For future enhancements, getting a larger sample for the dataset would definitely give the models a better prediction. A sample size of 1,000 is not large enough to give accurate estimates of what defines each genre of music, especially since there are 10 different genres of music in the dataset. In order to increase the accuracy of the random forest model, a suggestion would be to use cross validation on the model to fill that gap of using a larger dataset. Overall, the models used did a valid job predicting the files music genre, and displayed a good representation of how prediction techniques can be applied on datasets using these machine learning models.