MIS 545 Data Mining Description

Predicting a Song's Genre

by

Kirsten Fure

August 2019

Problem Description

With the popularity of digital music, there has been great interest in browsing, searching and organizing large music collections. Users become overwhelmed with the volume of music and desire recommendations for music they might like. In order to provide these predictions to users, first one must categorize music into types/genres and then into sub-genres. Many songs span several genres and consist of a combination of sub-genres. In order to categorize music in such a detailed way, it would be useful to have a computer program which could predict a song's genre. The goal of this project is determine if we can predict a song's top-level genre based upon a set of attributes and musical features contained in the database. Questions I will explore include:

- What elements of a song are similar among genres?
- Is it possible to predict a song's genre given the data within this database?
- How accurate can we be in our predictions?

Executive Summary of Results and Findings

Categorizing songs into genres is a difficult task. Often two people listening to a song can't agree on a song's genre, and there is a lot of crossover in music. Some pop songs will have elements of Jazz or Folk, vice versa. Using a computer program to predict a song's genre is no less complex. It requires a decent number of musical attributes and records, but does result in a prediction better than baseline. The decision tree proved to perform better than Naive Bayes in this project.

Project Dataset Description

For this project, I used the FMA (Free Music Archive) Database. It includes audio information for 106,574 tracks from 16,341 artists and 14,854 albums. It consists of 9 relational databases, and has a hierarchical structure of assigned genres that expands to 161 sub-genres. There are 17 different top-level genres. The database contains the full-length audio and related information to the track, artist and the album. The dataset also includes detailed features about the music that have been extracted using a program called libROSA. This python-based program has been run on each audio track and the database includes numerous musical features such as: spectral, rhythm, manipulation, inversion, bandwidth, flatness, roll-off frequency, tonnetz (tonal centroid), and many other special features such as chromagrams. These features are represented

as continuous attributes.
The data consists of the following csv files:

1. Tracks.csv
2. Features.csv
3. Echonest.csv
4. Genres.csv
5. Raw_albums.csv
6. Raw_artists.csv
7. Raw_echonest.csv
8. Raw_genres.csv
9. Raw_tracks.csv

The Tracks table contains 106,574 rows and 53 columns and contains information about the song, artist, title, bitrate, etc..

The Features table contains 106,574 rows 519 columns, which mostly consists of continuous values of musical features relating to: Chroma Energy Normalized" (CENS), Constant-Q chromagram, chromagram from a waveform, Mel-frequency cepstral coefficients (MFCCs), root-mean-square (RMS), spectral centroid, spectral bandwidth, spectral contrast, Spectral flatness (or tonality coefficient), roll-off frequency, tonal centroid features (tonnetz), etc.

The Echonest table contains interesting data, but only for a subset of the songs. I did extensive analysis including this data, but found that the attributes from the Features table were actually giving better accuracy ratings. Due to the fact that the Echonest table did not include all the songs and was not increasing accuracy, I decided not to use this table in my final analysis.

Data Preprocessing / Transformation

In order to make the data more manageable, I selected 3 genres to work with. I created a subset of the original data by selecting Folk, Classical and Hip-Hop music and the columns that were relevant to this project. I only selected songs that had a top-level genre and no missing data. There were 1230 Classical songs, 2803 Folk songs and 3552 Hip-Hop songs in my subset.

In the features data, I also created a subset of columns from 519 variables to 12. I merged these 2 tables based upon Track_ID. I chose the following continuous independent variables:

- Track bitrate

- Chroma Energy Normalized (chroma-CENS)
- Constant-Q chromagram (chroma-cqt),
- Chromagram from a waveform (chroma-stft)
- Mel-frequency cepstral coefficients (MFCCs),
- Root-mean-square (RMSE),
- Spectral centroid,
- Spectral bandwidth,
- Spectral contrast,
- zero-crossing rate (zcr),
- roll-off frequency,
- tonal centroid features (tonnetz)

Table	**Field Name**	**Variable Type**	**Data Format**	**Description**
Tracks.csv	Track_ID	IV	int	Unique identifier for each song. Used to merge tables.
Tracks.csv	Track_bitrate	IV	int	Number of bits per unit of time
Tracks.csv	Track_genre_top	DV	factor	Music genre category
Features.csv	Chroma_cens	IV	num	Chroma Energy Normalized
Features.csv	Chroma_cqt	IV	num	Constant-Q chromagram
Features.csv	Chroma_stft	IV	num	Chromagram from a waveform or power spectrogram
Features.csv	Mfcc	IV	num	Mel-frequency cepstral coefficients
Features.csv	Rmse	IV	num	Root-mean-square
Features.csv	Spectral_bandwidth	IV	num	Spectral bandwidth
Features.csv	Spectral_contrast	IV	num	Spectral contrast
Features.csv	Spectral_centroid	IV	num	Spectral centroid
Features.csv	Spectral_rolloff	IV	num	Spectral rolloff for each frame
Features.csv	Tonnetz	IV	num	tonal centroid features
Features.csv	zcr	IV	num	zero-crossing rate

Here is the structure and summary of the subset data:

```
> str(Song)
'data.frame':	7585 obs. of  26 variables:
 $ X.1               : int  1 2 3 4 5 6 7 8 9 10 ...
 $ track_ID          : int  2 3 5 134 139 140 141 142 188 189 ...
 $ X                 : int  3 4 5 12 17 18 19 20 62 63 ...
 $ album_listens     : int  6073 6073 6073 6073 1304 1300 1304 845 12333 894 ...
 $ album_title       : chr  "AWOL - A Way Of Life" "AWOL - A Way Of Life" "AWOL - A Way Of Life" "AWOL - 
A Way Of Life" ...
 $ artist_name       : chr  "AWOL" "AWOL" "AWOL" "AWOL" ...
 $ set_split         : chr  "training" "training" "training" "training" ...
 $ set_subset        : chr  "small" "medium" "small" "medium" ...
 $ track_bitrate     : int  256000 256000 256000 256000 128000 128000 128000 128000 256000 256000 ...
 $ track_genre_top   : Factor w/ 3 levels "Classical","Folk",..: 3 3 3 3 2 2 2 2 2 2 ...
 $ track_genres      : chr  "[21]" "[21]" "[21]" "[21]" ...
 $ track_genres_all  : chr  "[21]" "[21]" "[21]" "[21]" ...
 $ track_interest    : int  4656 1470 1933 1126 702 1593 839 1223 3137 730 ...
 $ track_listens     : int  1293 514 1151 943 582 1299 725 848 1253 101 ...
 $ track_title       : chr  "Food" "Electric Ave" "This World" "Street Music" ...
 $ chroma_cens       : num  7.1807 1.889 0.5276 0.9184 -0.0209 ...
 $ chroma_cqt        : num  4.518 -0.855 -0.605 -0.829 -0.436 ...
 $ chroma_stft       : num  -1.006 -0.952 -0.795 -0.972 -0.278 ...
 $ mfcc              : num  3.86 4.3 2.62 1.56 5.04 ...
 $ rmse              : num  2.49986 -0.64396 0.00178 2.13323 1.45388 ...
 $ spectral_bandwidth: num  3.874 2.383 0.895 0.373 0.402 ...
 $ spectral_centroid : num  2.41 3.52 1.32 2.21 1.64 ...
 $ spectral_contrast : num  2.27 3.21 1.48 1.83 2.81 ...
 $ spectral_rolloff  : num  0.841 2.379 -0.239 0.488 0.961 ...
 $ tonnetz           : num  2.303 2.004 10.772 3.829 0.437 ...
 $ zcr               : num  5.76 2.82 6.81 4.73 1.16 ...
```

```
> summary(Song)
      X.1          track_ID            X          album_listens      album_title       
 Min.   :   1   Min.   :     2   Min.   :     3   Min.   :     -1   Length:7585       
 1st Qu.:1897   1st Qu.: 33717   1st Qu.: 21567   1st Qu.:   3704   Class :character  
 Median :3793   Median : 69826   Median : 47477   Median :   7611   Mode  :character  
 Mean   :3793   Mean   : 73441   Mean   : 49136   Mean   :  29036                     
 3rd Qu.:5689   3rd Qu.:110928   3rd Qu.: 73139   3rd Qu.:  18286                     
 Max.   :7585   Max.   :155306   Max.   :106563   Max.   :1193803                     
 artist_name         set_split          set_subset        track_bitrate      track_genre_top
 Length:7585        Length:7585        Length:7585        Min.   :    -1   Classical:1230  
 Class :character   Class :character   Class :character   1st Qu.:192000   Folk     :2803  
 Mode  :character   Mode  :character   Mode  :character   Median :256000   Hip-Hop  :3552  
                                                          Mean   :256883                   
                                                          3rd Qu.:320000                   
                                                          Max.   :324221                   
 track_genres       track_genres_all   track_interest    track_listens     track_title       
 Length:7585        Length:7585        Min.   :     25   Min.   :    18   Length:7585       
 Class :character   Class :character   1st Qu.:    542   1st Qu.:   274   Class :character  
 Mode  :character   Mode  :character   Median :   1106   Median :   675   Mode  :character  
                                       Mean   :   3296   Mean   :  2040                     
                                       3rd Qu.:   2559   3rd Qu.:  1657                     
                                       Max.   :1038669   Max.   :433992                     
  chroma_cens         chroma_cqt         chroma_stft          mfcc              rmse         
 Min.   :-1.7136   Min.   :-1.7778   Min.   :-1.8081   Min.   : -1.7314   Min.   : -1.6853  
 1st Qu.:-0.7451   1st Qu.:-1.0641   1st Qu.:-1.1624   1st Qu.:  0.2013   1st Qu.: -0.2160  
 Median :-0.3230   Median :-0.7130   Median :-0.8606   Median :  1.4552   Median :  0.5264  
 Mean   : 0.0006   Mean   :-0.3690   Mean   :-0.4130   Mean   :  3.2162   Mean   :  1.4832  
 3rd Qu.: 0.2826   3rd Qu.:-0.1613   3rd Qu.:-0.1395   3rd Qu.:  3.8507   3rd Qu.:  1.8563  
 Max.   :55.7586   Max.   :37.4453   Max.   :31.6291   Max.   :986.1246   Max.   :135.5881  
 spectral_bandwidth spectral_centroid  spectral_contrast spectral_rolloff     tonnetz        
 Min.   : -1.720    Min.   :  -1.897   Min.   :-0.9696   Min.   : -1.8927   Min.   : -1.2269  
 1st Qu.:  0.065    1st Qu.:   1.977   1st Qu.: 1.2803   1st Qu.:  0.0229   1st Qu.:  0.1297  
 Median :  1.592    Median :   6.794   Median : 1.9215   Median :  2.4889   Median :  0.4973  
 Mean   :  6.787    Mean   :  26.121   Mean   : 1.9417   Mean   : 16.8848   Mean   :  1.2062  
 3rd Qu.:  6.070    3rd Qu.:  21.763   3rd Qu.: 2.5524   3rd Qu.: 12.1243   3rd Qu.:  1.1537  
 Max.   :304.935    Max.   :2439.780   Max.   :18.5083   Max.   :974.8627   Max.   :111.9395  
      zcr          
 Min.   :  -1.773  
 1st Qu.:   4.690  
 Median :  11.431  
 Mean   :  26.807  
 3rd Qu.:  29.739  
 Max.   :1187.837  
```

Feature Selection - Descriptive Analysis

I used linear regression for feature selection, converting my dependent variable to numeric. I used StepAIC both ways (forward and backward) in order to get the best results. The final model recommended 10 of the 12 independent variables, making it difficult to eliminate many attributes.

```
> steps$anova
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
track_genre_top ~ track_bitrate + chroma_cens + chroma_cqt +
    chroma_stft + mfcc + rmse + spectral_bandwidth + spectral_centroid +
    spectral_contrast + spectral_rolloff + tonnetz + zcr

Final Model:
track_genre_top ~ track_bitrate + chroma_cens + chroma_cqt +
    chroma_stft + rmse + spectral_bandwidth + spectral_centroid +
    spectral_rolloff + tonnetz + zcr


                 Step Df  Deviance Resid. Df Resid. Dev       AIC
1                                       7572   2412.111 -8663.908
2 - spectral_contrast  1 0.1859761      7573   2412.297 -8665.323
3              - mfcc  1 0.4388505      7574   2412.736 -8665.943
```

I also used the conditional random forest algorithm and the varimp() function to determine variable importance according to this method. I ran it with 50 trees max and with 500 and the results gave similar orders of variable importance.

```
> cfs = cforest(track_genre_top ~ . , data = S, control=cforest_unbiased(mtry=2, ntree=50))
> varimp(cfs)
     track_bitrate        chroma_cens         chroma_cqt        chroma_stft               mfcc
       0.040759798        0.014456008        0.007814328        0.027943623        0.012986888
              rmse spectral_bandwidth  spectral_centroid  spectral_contrast   spectral_rolloff
       0.024430680        0.049199836        0.062526847        0.003763833        0.122026545
           tonnetz                zcr
       0.002253692        0.012142676
```

When I change ntry to a higher value (like 12), I get slightly different results in variable importance. Spectral bandwidth is more important and centroid less.

```
> cfs = cforest(track_genre_top ~ . , data = S, control=cforest_unbiased(mtry=12, ntree=50))
> varimp(cfs)
     track_bitrate        chroma_cens         chroma_cqt        chroma_stft               mfcc
      0.0432819778       0.0088355428       0.0079398065       0.0290433536       0.0106986743
              rmse spectral_bandwidth  spectral_centroid  spectral_contrast   spectral_rolloff
      0.0137513436       0.0766606951       0.0139949839       0.0079899678       0.0790899319
           tonnetz                zcr
      0.0006234325       0.0115084199
```

Linear Regression and Random Forest give slightly different results. The random forest indicates that tonnetz is the least important, followed by chroma_cqt, spectral_contrast, , chroma_cens, mfcc, and zcr. They both seem to agree that spectral_contrast is lower in importance and possibly mfcc too.

Associated Rule Analysis

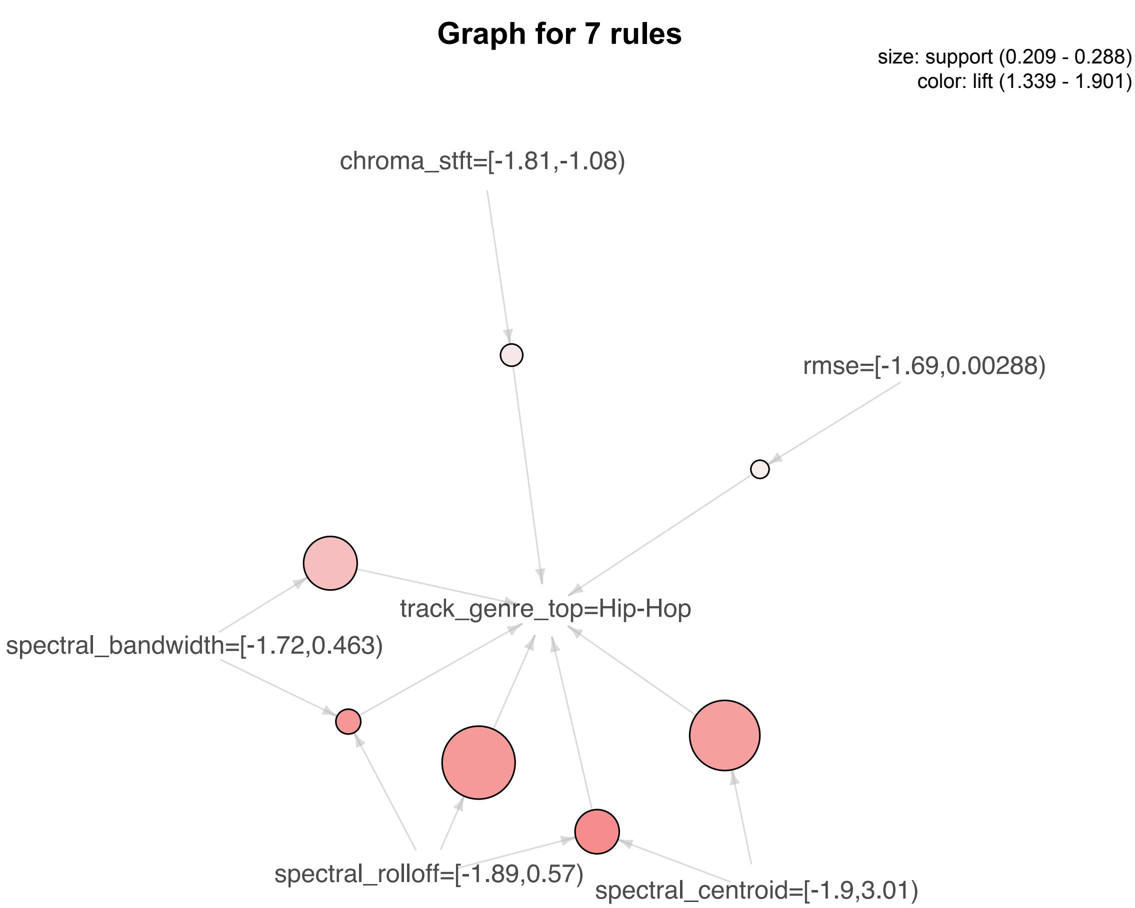
In order to use associated rule analysis and the apriori function, I needed to convert the continuous independent variables into categorical. I used the discretize function to do this.

The overall top 7 rules (sorted by support, confidence or lift) all indicated Hip Hop as the genre. Spectral_rolloff, Spectral_centroid, Spectral_Bandwidth are important indicators for Hip Hop. Also chroma_stft and rmse make an impact.

For the Folk genre, the top variables were chroma_cqt, chroma_stft, spectral_rolloff, and zcr. Spectral_bandwidth and spectral_centroid also make an impact.

For the Classical genre, spectral_bandwidth, spectral_centroid, spectral_rolloff and zcr are the biggest indicators.

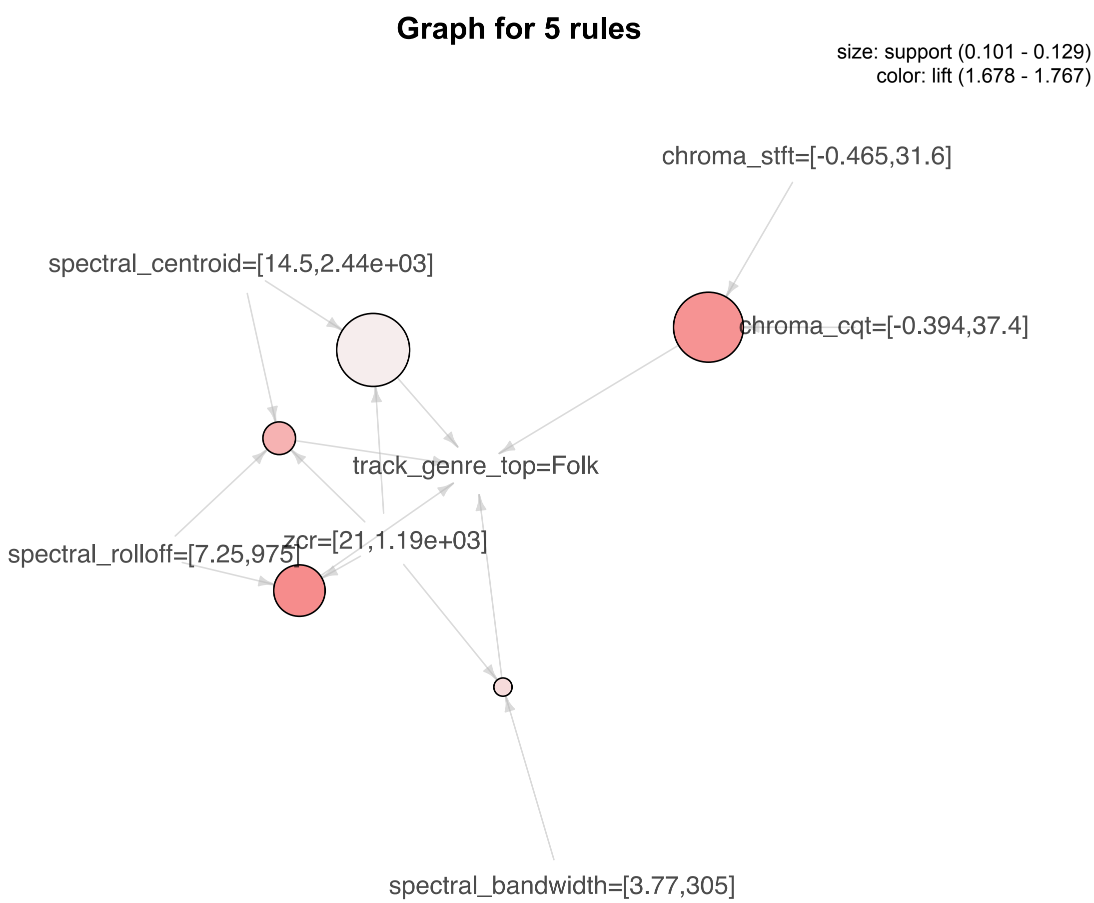
According to the association rules, chroma_cens and tonnetz do not make a big impact to any genre indication. Also, chroma_stft and rmse are less important.



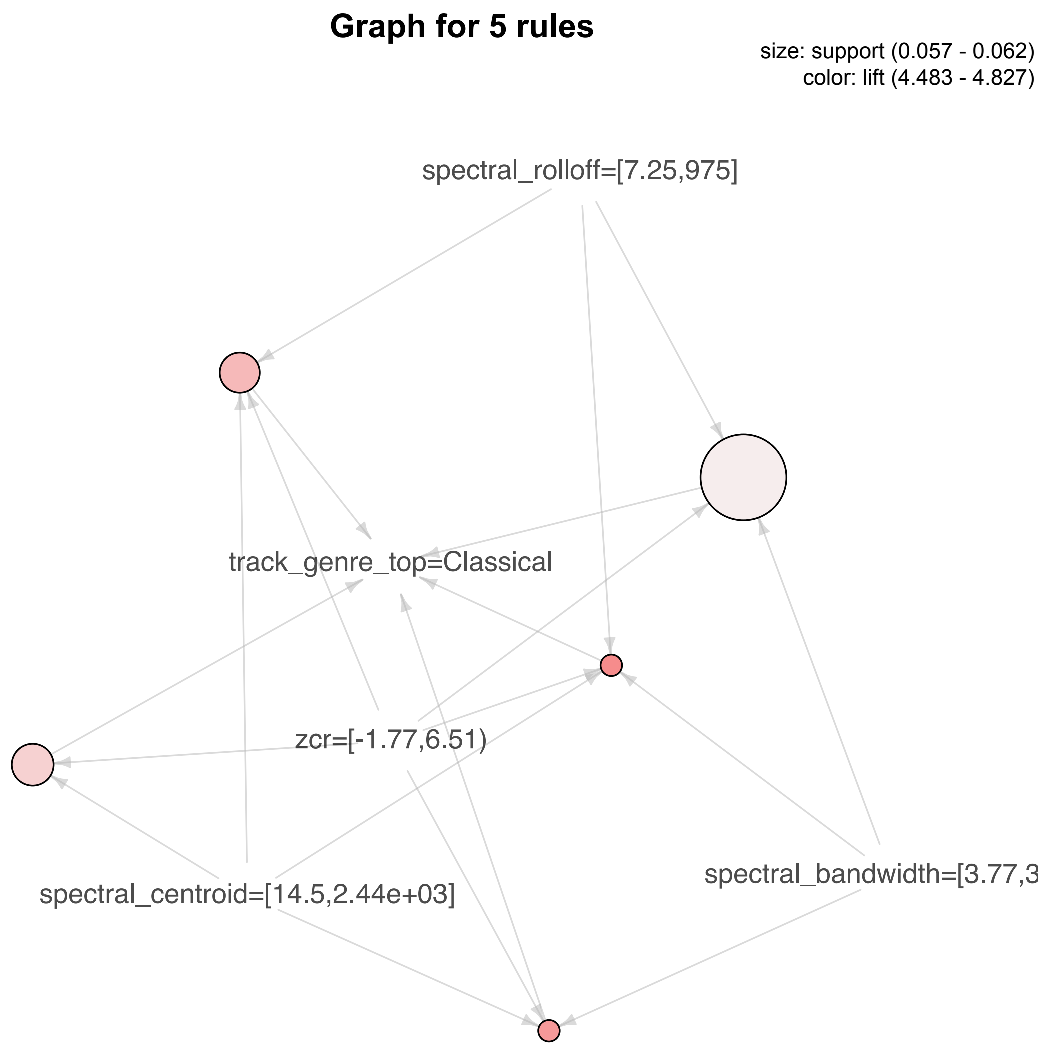
Top 7 overall rules:

```
> rules <- sort(rules, decreasing = TRUE, by = "support")
> inspect(rules[1:7])
    lhs                                    rhs                         support confidence     lift count
[1] {spectral_rolloff=[-1.89,0.57)}     => {track_genre_top=Hip-Hop} 0.2882004  0.8647152 1.846527  2186
[2] {spectral_centroid=[-1.9,3.01)}     => {track_genre_top=Hip-Hop} 0.2841134  0.8524525 1.820341  2155
[3] {spectral_bandwidth=[-1.72,0.463)} => {track_genre_top=Hip-Hop} 0.2601187  0.7804589 1.666605  1973
[4] {spectral_centroid=[-1.9,3.01),
     spectral_rolloff=[-1.89,0.57)}     => {track_genre_top=Hip-Hop} 0.2465392  0.8900524 1.900633  1870
[5] {spectral_bandwidth=[-1.72,0.463),
     spectral_rolloff=[-1.89,0.57)}     => {track_genre_top=Hip-Hop} 0.2185893  0.8671548 1.851737  1658
[6] {chroma_stft=[-1.81,-1.08)}         => {track_genre_top=Hip-Hop} 0.2147660  0.6443829 1.376026  1629
[7] {rmse=[-1.69,0.00288)}              => {track_genre_top=Hip-Hop} 0.2089651  0.6269778 1.338859  1585
```

Top 5 rules for Folk music genre:



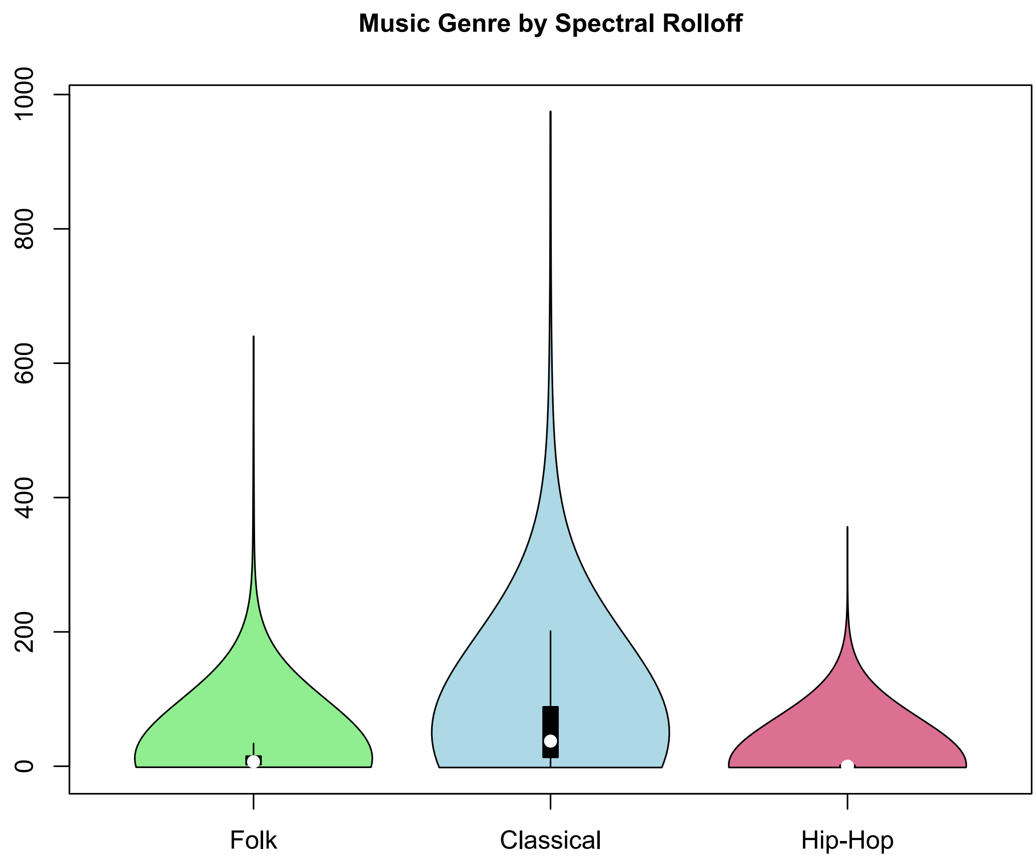
Top 5 rules for Classical music:

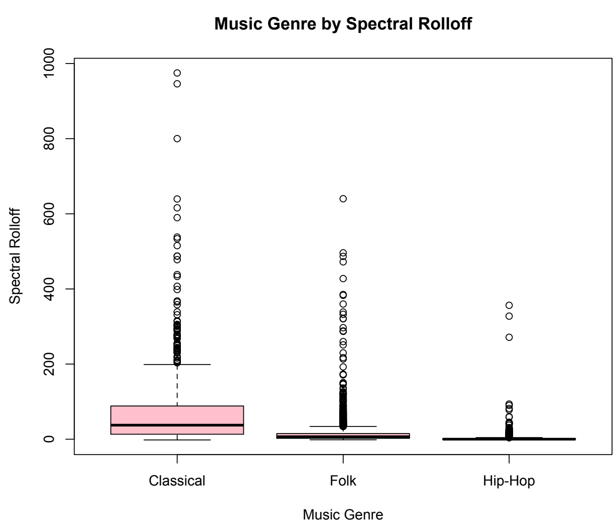


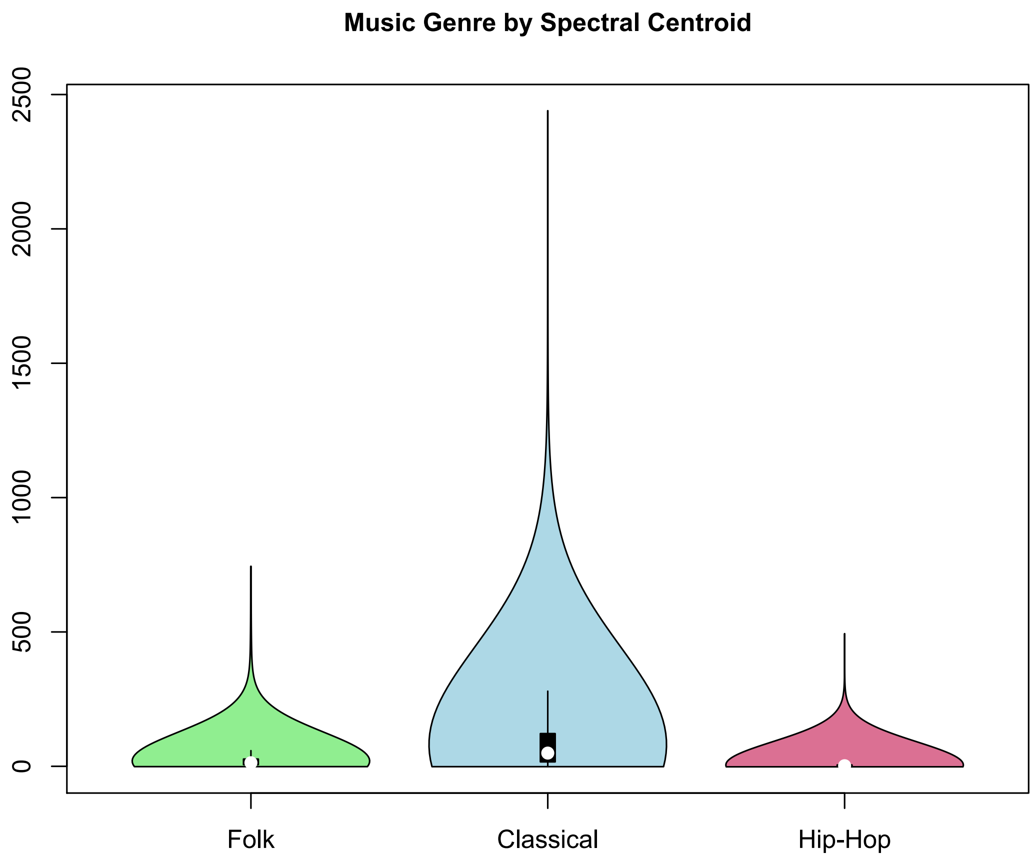
Feature Selection Exploratory Analysis

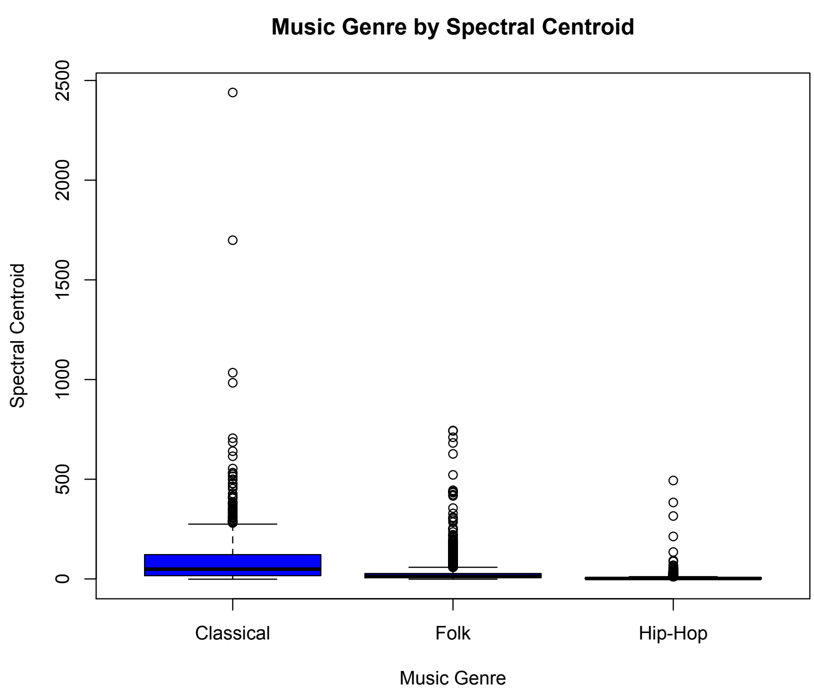
To analyze the data, I looked at relationships between genre and the individual attributes. No single attribute showed a strong exclusive relationship.

Here are some graphs showing how these values associate with the genres:





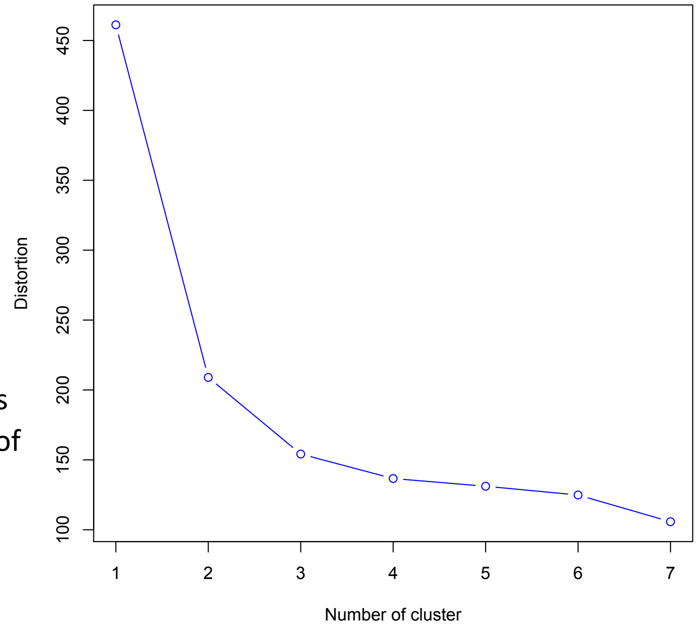

Music Genre by Spectral Centroid
0
500
1000
1500
2000
2500
Folk
Classical
Hip-Hop


Music Genre by Spectral Centroid
Spectral Centroid
0
500
1000
1500
2000
2500
Classical
Folk
Hip-Hop
Music Genre

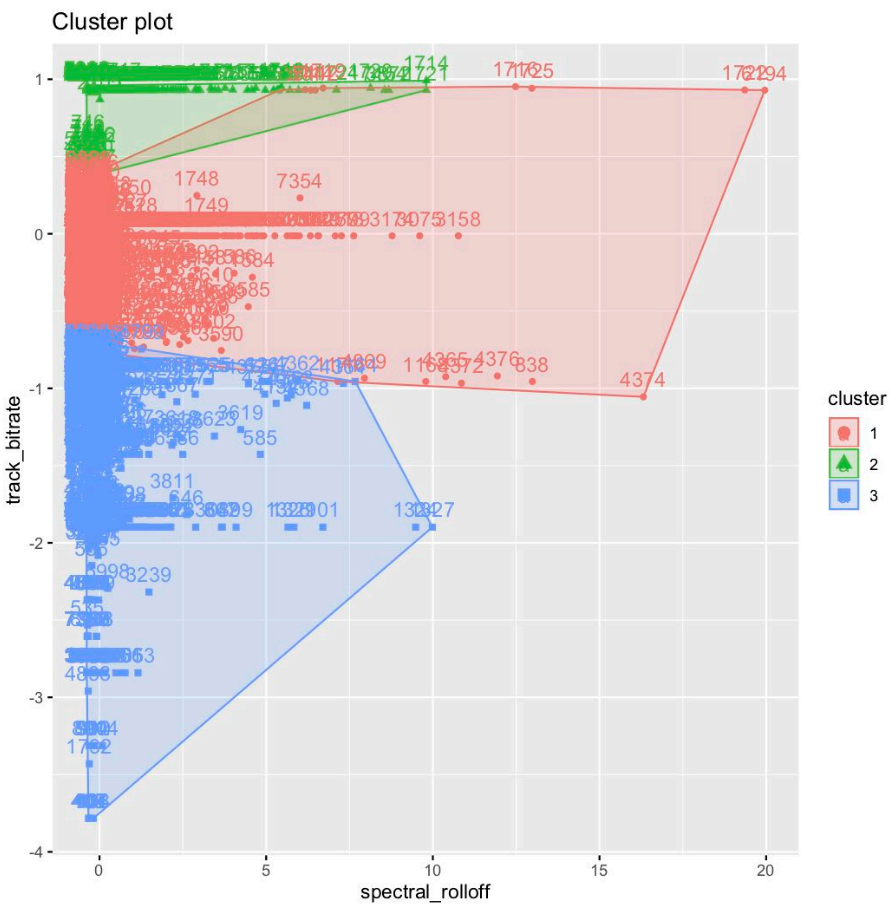
Descriptive Analysis with Clustering Analysis

I ran K-means clustering on all the continuous independent variables (normalized) to see if there are interesting relationships. Because my subset data has 3 genres and the elbow curve supports that value, I chose K=3 for the number of clusters.

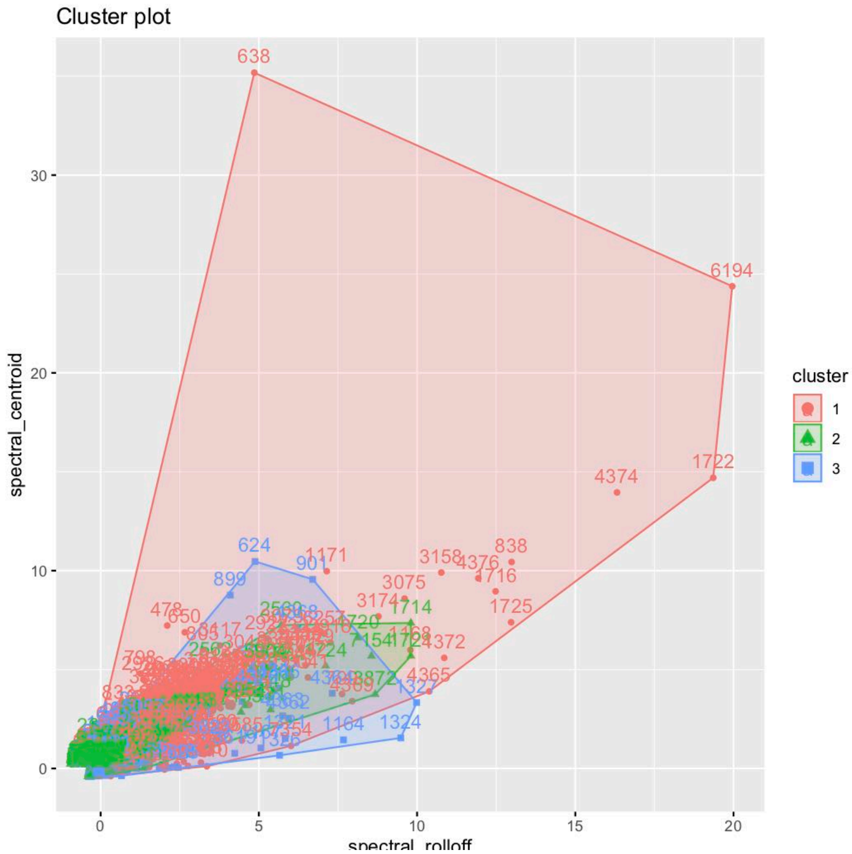
I used the fviz_cluster() function to visualize the clusters for all variables and was able to change the x and y coordinates to different variables in my dataset. Most of the clusters were nested, showing that K-means is not the best algorithm for this dataset. Hierarchical clustering may give better results.

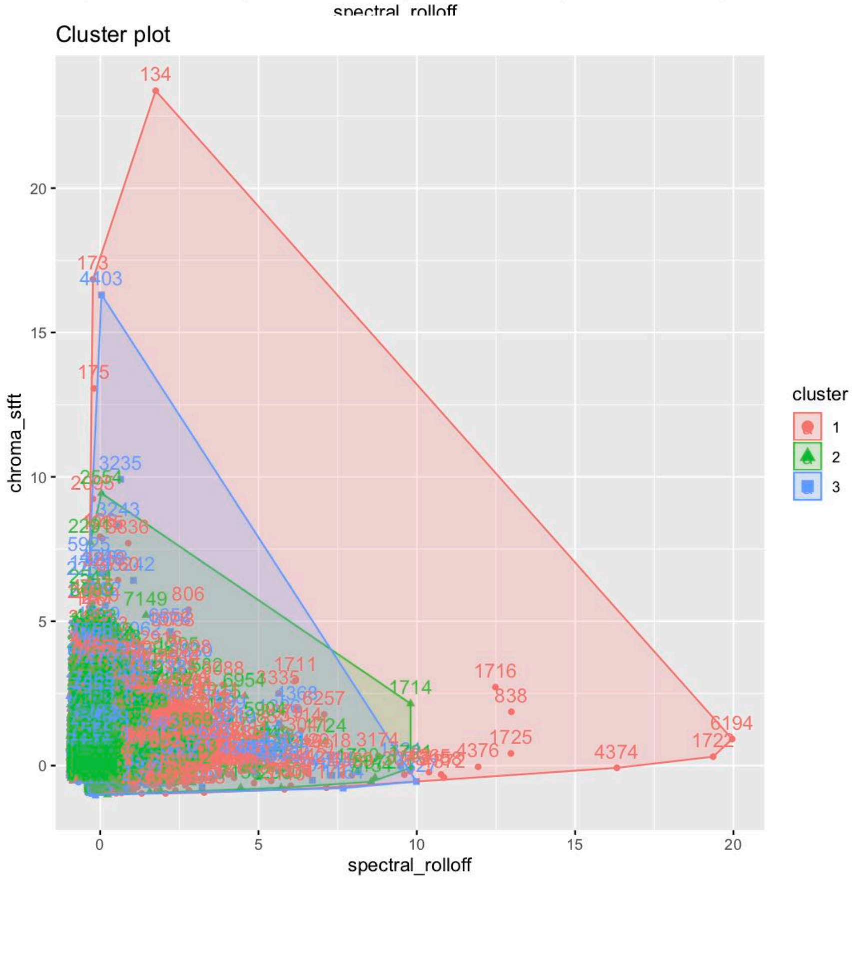


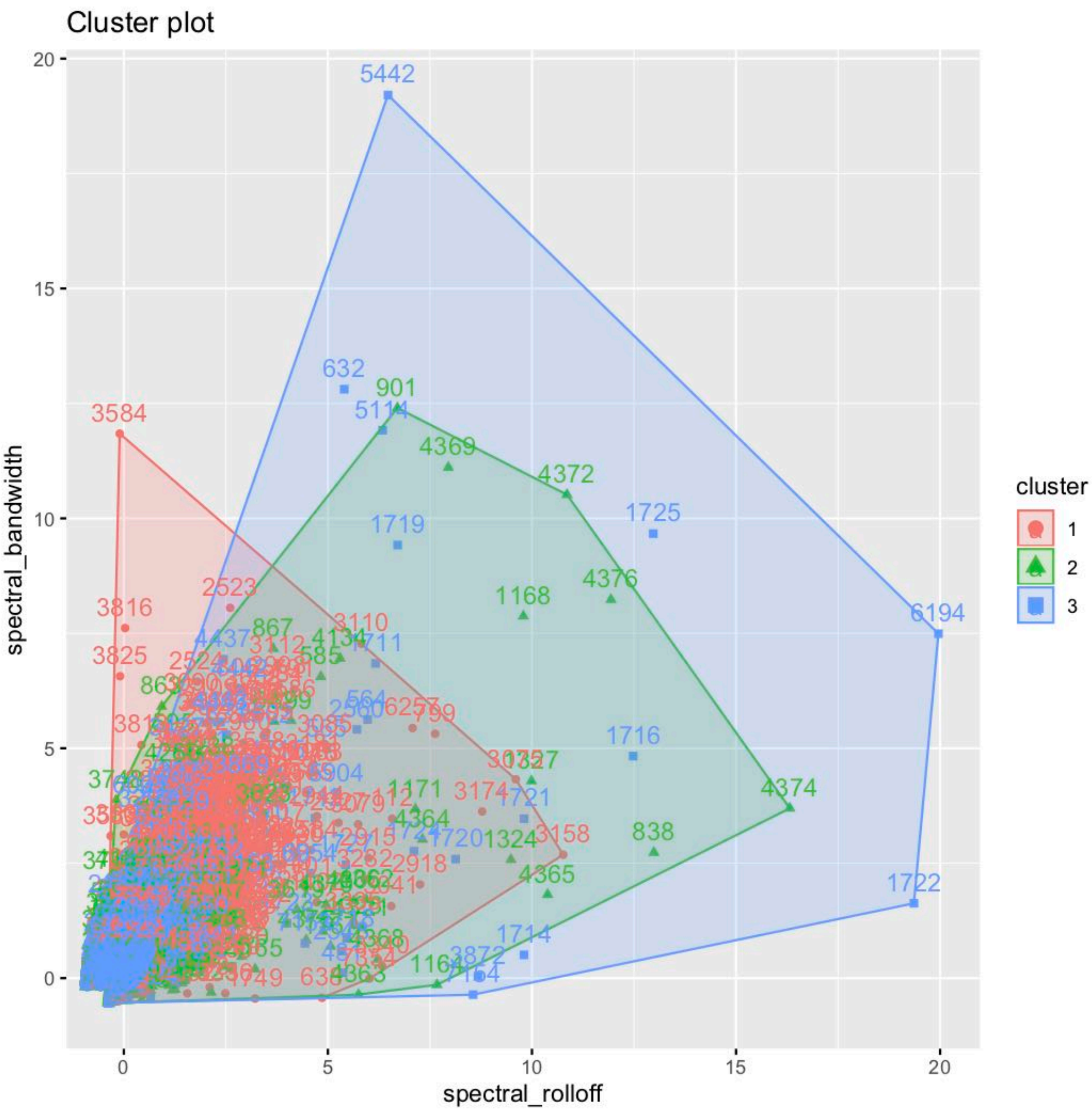
Spectral Rolloff and Bitrate had the best separated clusters.



Here are just a few of some examples of other clustering combinations, showing nesting:





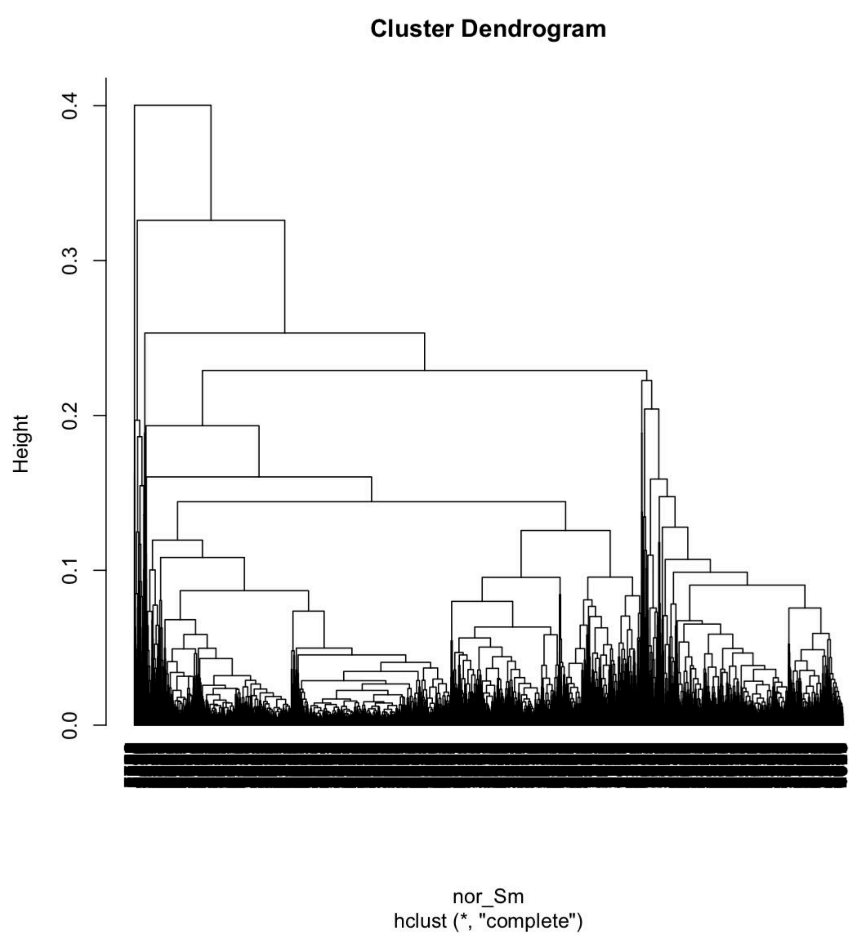

Cluster plot
spectral_bandwidth
spectral_rolloff
0
5
10
15
20
cluster
1
2
3

Descriptive Analysis with Hierarchical Clustering

I ran hclust() to see if hierarchical clustering could give more insight.

```
> nor_Sm = daisy(nor_Sm, metric = "gower")
> hc = hclust(nor_Sm)
> hc

Call:
hclust(d = nor_Sm)

Cluster method   : complete
Number of objects: 7585

> summary(hc)
            Length Class  Mode
merge       15168  -none- numeric
height       7584  -none- numeric
order        7585  -none- numeric
labels          0  -none- NULL
method          1  -none- character
call            2  -none- call
dist.method     0  -none- NULL
```

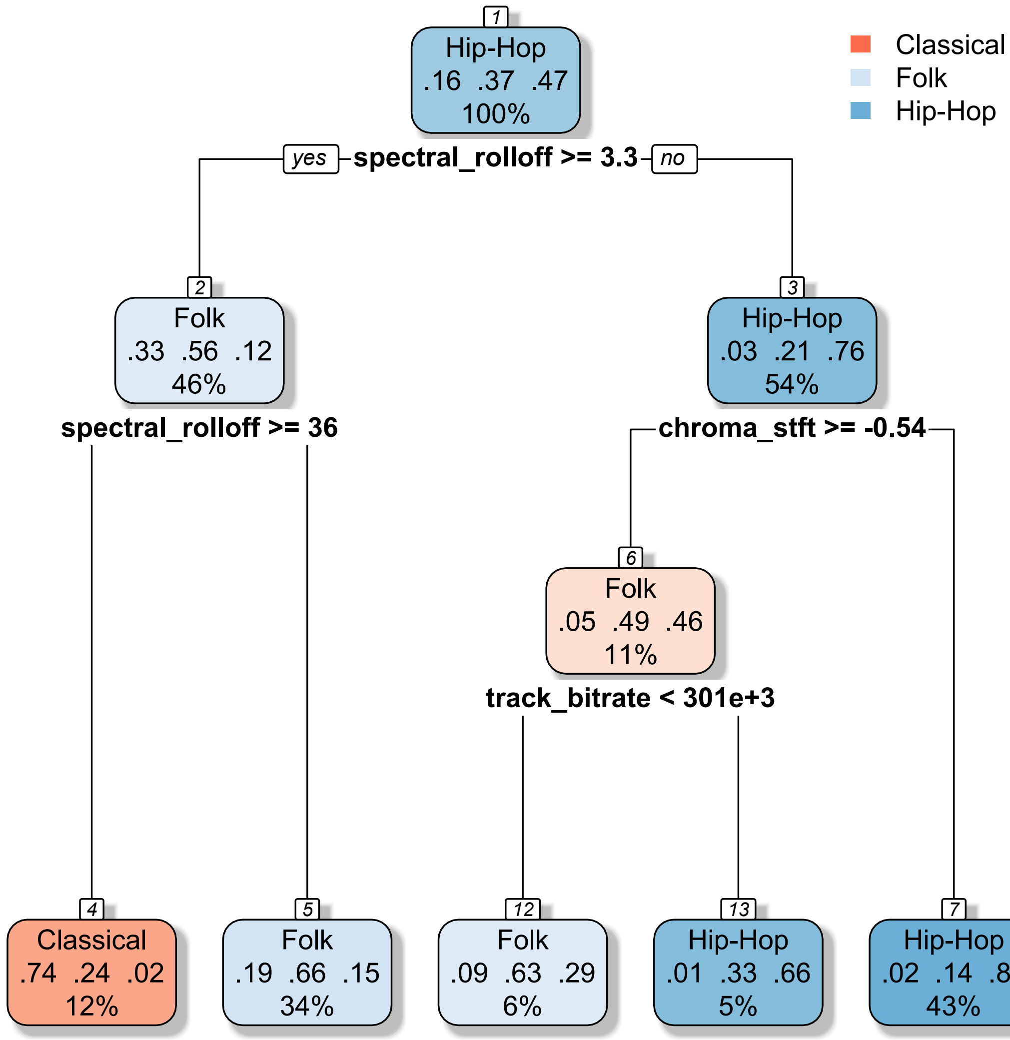
The resulting dendogram is complex and shows the nesting nature of the attributes.



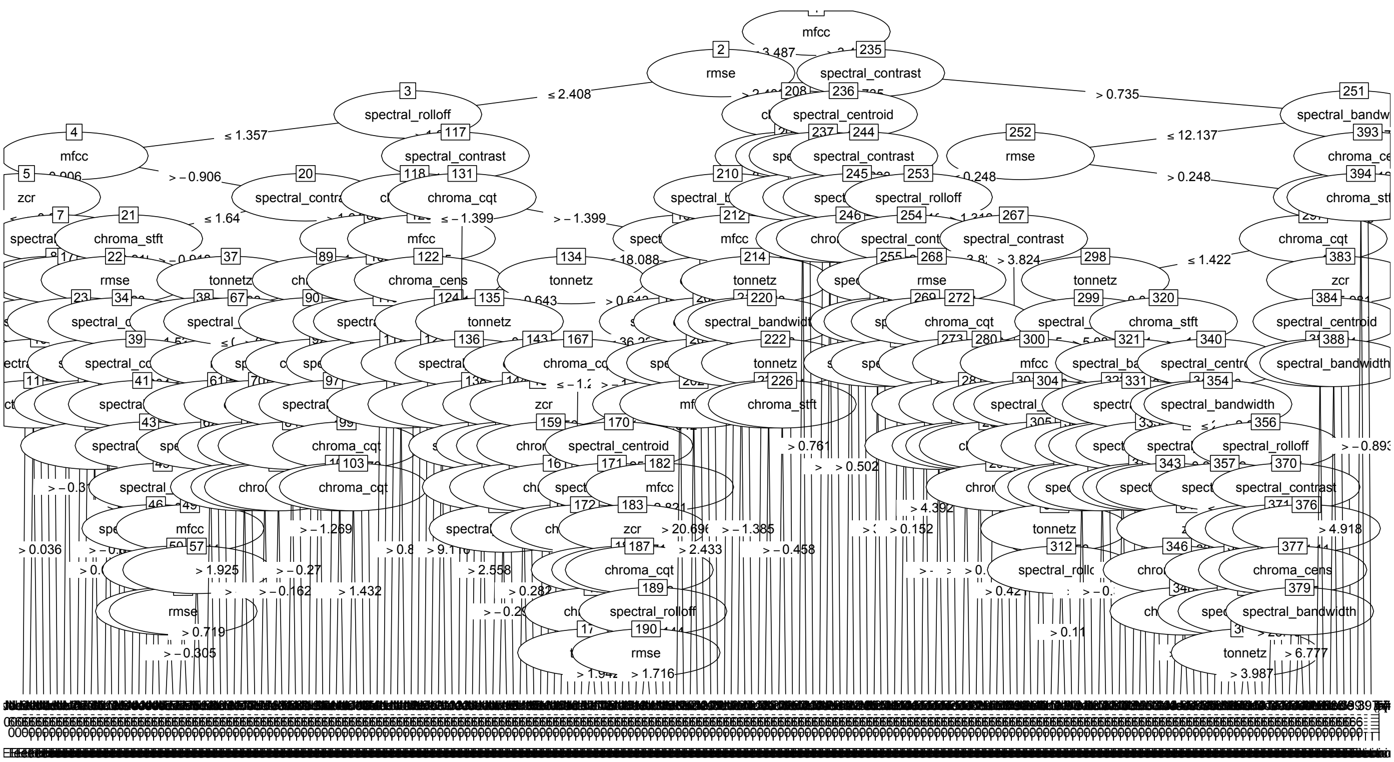
Predictive Analysis with Decision Tree

I created training data with an 80% random subset of data using all independent variables, and created testing data as the other 20% of records. I ran the decision tree on the training data and it created 57 sub-trees total.

```
Evaluation on training data (6068 cases):

	    Decision Tree
	  ----------------
	  Size      Errors

	   171  889(14.7%)   <<


	   (a)   (b)   (c)    <-classified as
	  ----  ----  ----
	   813   138    32    (a): class Classical
	   167  1745   318    (b): class Folk
	    13   221  2621    (c): class Hip-Hop


	Attribute usage:

	100.00%	spectral_rolloff
	 72.30%	chroma_stft
	 70.14%	mfcc
	 68.72%	track_bitrate
	 60.07%	rmse
	 49.16%	spectral_centroid
	 40.97%	chroma_cens
	 33.97%	zcr
	 23.62%	spectral_contrast
	 23.42%	chroma_cqt
	 12.28%	tonnetz
	  5.49%	spectral_bandwidth


Time: 0.1 secs
```



This high-level chart of the decision tree shows how the algorithm split the data at the top sub-trees, and shows that spectral rolloff, chroma stft, and track bitrate had the biggest impact.



The entire decision tree consists of 57 sub-trees and is complex to visualize.

Evaluation of the Decision Tree Model

I ran the prediction with the testing dataset, and compared whether the predicted genre matched the actual genre.

The decision tree accuracy was **74%** and here is the resulting confusion matrix showing the testing data.

```
> sum(dt_evaluation$correct) / nrow(dt_evaluation)
[1] 0.7376401
> table(dt_evaluation$track_genre_top, dt_evaluation$dt_pred)

            Classical Folk Hip-Hop
  Classical       162   73      12
  Folk             65  377     131
  Hip-Hop           8  109     580
```

The Recall/True Positive Rate (True Positive for Class / Actual for Class):

Classical	66%	(162/247 = 66%)
Folk	66%	(377/573 = 66%)
Hip-Hop	83%	(580/697 = 83%)

Average Recall – 71.5%

The Precision (True Positive for Class / Predicted for Class):

Classical	69%	(162/235 = 69%)
Folk	67%	(377/559 = 67%)
Hip-Hop	80%	(580/723 = 80%)

Average Precision – 72%

These evaluation metrics for precision and recall show that the decision tree model is doing a fairly good job of predicting the genre. The Recall and Precision are balanced, so it has similar performance in predicting correctly and false predictions.

There are different ways to calculate the F1 score: macro, weighted and micro. Since we did not discuss the differences for multi-class situations in our class, I will use the simplest macro average calculation. F1 can be misleading for multi-class situations and favor the lower numbers.

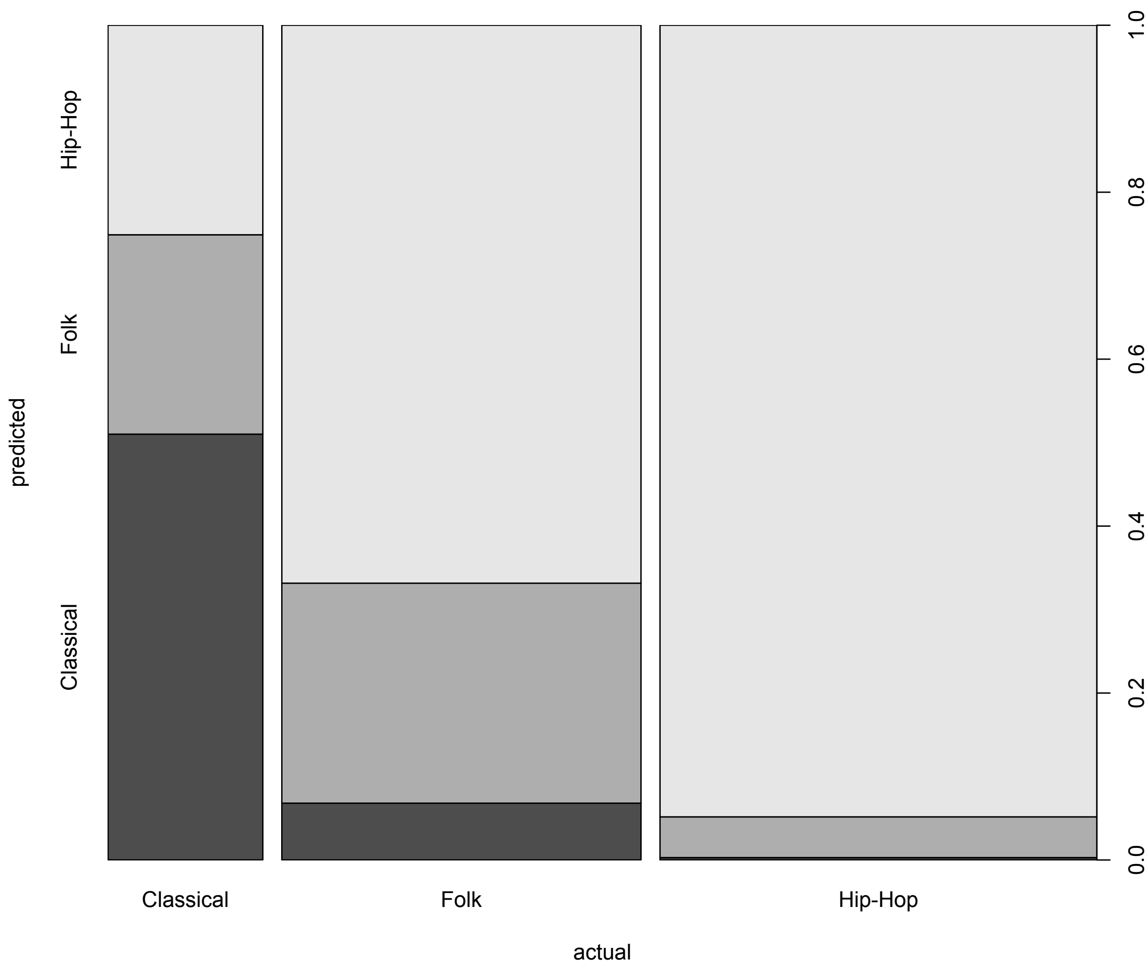
F1 Score [2 × (precision × recall)/(precision + recall)]:

Classical	67%
Folk	67%
Hip-Hop	82%

Overall Macro Average F1 Score – 72%

Predictive Analysis with Naïve Bayes

I ran the Naïve Bayes algorithm on the training data with all independent variables and then ran prediction with the testing dataset.



This chart of the Naïve Bayes model shows that it was very successful in predicting Hip-Hop (hence the large light grey area), and fairly good at predicting Classical (hence the larger dark grey area. But, it struggled to predict Folk correctly; we would want the medium grey area to be the largest, but unfortunately it was incorrectly predicting Hip-Hop more often.

Evaluation Analysis of the Naïve Bayes Model

Comparing predicted genre with the actual genre gave an accuracy rating of **62%,** so the decision tree had better results.

```
> table(results)
           predicted
actual      Classical Folk Hip-Hop
  Classical       126   59      62
  Folk             39  151     383
  Hip-Hop           2   34     661
```

The Recall/True Positive Rate (True Positive for Class / Actual for Class):

Classical 51%
Folk 26%
Hip-Hop 94%
Average Recall – 57%

The Precision (True Positive for Class / Predicted for Class):

Classical 75%
Folk 62%
Hip-Hop 60%
Average Precision – 66%

These evaluation metrics for precision and recall show that the Naïve Bayes algorithm is usually doing better at predicting than pure guessing, but had a large number of wrong predictions for Folk. It predicted Hip-Hop many times when it was a Folk song. It did a very good job at recognizing Hip-Hop songs when they were actually Hip-Hop; it just seemed to think many Folk songs were also Hip-Hop.

There are different ways to calculate the F1 score: macro, weighted and micro. Since we did not discuss the differences for multi-class situations in our class, I will use the simplest macro average calculation. F1 can be misleading for multi-class situations and favor the lower numbers.

F1 Score [2 × (precision × recall)/(precision + recall)]:

Classical 60%
Folk 37%
Hip-Hop 73%
Overall Macro Average F1 Score – 57%

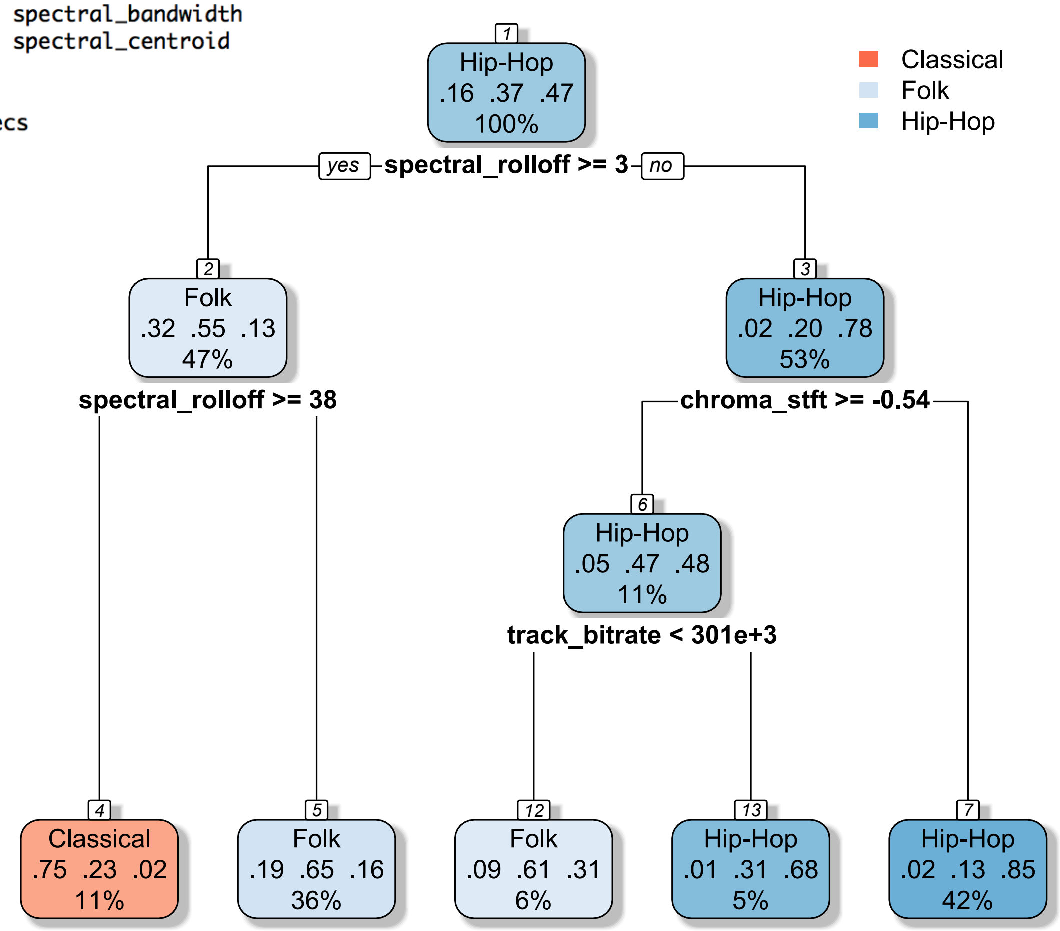
Using R to calculate statistics of the Naïve Bayes Model:

```
> cm = confusionMatrix(results$predicted, results$actual)
> cm
Confusion Matrix and Statistics

           Reference
Prediction  Classical Folk Hip-Hop
  Classical       134   48       7
  Folk             59  160      28
  Hip-Hop          71  350     660

Overall Statistics

               Accuracy : 0.6289
                 95% CI : (0.604, 0.6532)
    No Information Rate : 0.4581
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3731

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Classical Class: Folk Class: Hip-Hop
Sensitivity                   0.50758      0.2867         0.9496
Specificity                   0.95611      0.9093         0.4878
Pos Pred Value                0.70899      0.6478         0.6105
Neg Pred Value                0.90211      0.6866         0.9197
Prevalence                    0.17403      0.3678         0.4581
Detection Rate                0.08833      0.1055         0.4351
Detection Prevalence          0.12459      0.1628         0.7126
Balanced Accuracy             0.73184      0.5980         0.7187
>
```

Fine-Tuning Attributes

After combining all the analysis, I chose 5 top attributes (spectral_rolloff, track_bitrate, chromas_stft, spectral_bandwidth and spectral_centroid) and re-ran the Decision Tree receiving the same 74% accuracy and very similar RTree of the top attributes.

```
Evaluation on training data (6068 cases):

	    Decision Tree
	  ----------------
	  Size      Errors

	   113 1101(18.1%)   <<


	   (a)   (b)   (c)    <-classified as
	  ----  ----  ----
	   650   296    46    (a): class Classical
	   118  1705   402    (b): class Folk
	    26   213  2612    (c): class Hip-Hop


	Attribute usage:

	100.00%  track_bitrate
	100.00%  spectral_rolloff
	 69.87%  chroma_stft
	 50.64%  spectral_bandwidth
	 24.67%  spectral_centroid


Time: 0.0 secs
```



Statistics of Decision Tree with top 5 attributes:

```
> cm = confusionMatrix( dt_evaluation$dt_pred , dt_evaluation$track_genre_top)
> cm
Confusion Matrix and Statistics

           Reference
Prediction  Classical Folk Hip-Hop
  Classical       136   54      10
  Folk             89  386      84
  Hip-Hop          13  138     607

Overall Statistics

               Accuracy : 0.7442
                 95% CI : (0.7215, 0.766)
    No Information Rate : 0.4621
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5793

 Mcnemar's Test P-Value : 6.239e-05

Statistics by Class:

                     Class: Classical Class: Folk Class: Hip-Hop
Sensitivity                   0.57143      0.6678         0.8659
Specificity                   0.94996      0.8158         0.8150
Pos Pred Value                0.68000      0.6905         0.8008
Neg Pred Value                0.92255      0.7996         0.8762
Prevalence                    0.15689      0.3810         0.4621
Detection Rate                0.08965      0.2544         0.4001
Detection Prevalence          0.13184      0.3685         0.4997
Balanced Accuracy             0.76069      0.7418         0.8404
```

I also re-ran the Naïve Bayes model with the same top 5 attributes chosen and received similar accuracy at 62% and similar statistics.

```
> Balance.model = naiveBayes(track_genre_top ~ . , data = train)
> Balance.model

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
Classical      Folk   Hip-Hop 
0.1634806 0.3666777 0.4698418 

Conditional probabilities:
           track_bitrate
Y               [,1]     [,2]
  Classical 228177.0 63079.78
  Folk      240855.8 64359.56
  Hip-Hop   280127.6 64623.12

           chroma_stft
Y                  [,1]      [,2]
  Classical  0.12478538 1.5917108
  Folk      -0.05996469 1.5595827
  Hip-Hop   -0.90117312 0.5895883

           spectral_bandwidth
Y                [,1]      [,2]
  Classical 23.114598 26.104833
  Folk       6.733016 12.188977
  Hip-Hop    1.216702  6.786786

           spectral_centroid
Y                [,1]      [,2]
  Classical 89.836514 138.50083
  Folk      25.612783  49.03236
  Hip-Hop    5.088629  14.62509

           spectral_rolloff
Y               [,1]     [,2]
  Classical 65.53946 89.44267
  Folk      15.24591 36.02310
  Hip-Hop    1.52557 10.27318
```

Statistics for Naïve Bayes with top 5 attributes:

```
> sum(results$correct) / nrow(results)
[1] 0.619644
> cm = confusionMatrix(results$predicted, results$actual)
> cm
Confusion Matrix and Statistics

           Reference
Prediction  Classical Folk Hip-Hop
  Classical        83   21       2
  Folk             98  180      22
  Hip-Hop          57  377     677

Overall Statistics

               Accuracy : 0.6196
                 95% CI : (0.5947, 0.6442)
    No Information Rate : 0.4621
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3388

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Classical Class: Folk Class: Hip-Hop
Sensitivity                   0.34874      0.3114         0.9658
Specificity                   0.98202      0.8722         0.4681
Pos Pred Value                0.78302      0.6000         0.6094
Neg Pred Value                0.89015      0.6730         0.9409
Prevalence                    0.15689      0.3810         0.4621
Detection Rate                0.05471      0.1187         0.4463
Detection Prevalence          0.06987      0.1978         0.7324
Balanced Accuracy             0.66538      0.5918         0.7170
```

Summary Conclusions

It is possible to predict music genre based upon attributes, but we don't get 100% accuracy. The Decision Tree performed the best at 74% accuracy, and the Naïve Bayes method had 62% accuracy

Each method used indicated different attributes as most important, however after combining all the results and choosing 5 top attributes, I was able to match similar performance in the two models as with 12 attributes.

In the future, I would recommend applying more hierarchical clustering methods to obtain more insight into the data, because it seems the attributes and relationships are nested. I would also like to compare the results from the Decision Tree and Naïve Bayes with Neural Network and SVM algorithms. Then I recommend taking the best performing model and run it on the full dataset with all the genres.