# Machine Learning: Music Genre Classification

Devon Hunter

May 2, 2019

## Abstract

Styles of music with similar aural attributes such as key, tempo, loudness, and energy are often categorized by genre. This project attempts to classify songs by genre from 9 different genres using the tonal qualities of each song. To accomplish this, models like KNN, SVM, and random forest were trained using over 14,000 songs. With 9 possible classes, however, it is difficult for any model to achieve reasonably high accuracy. At best, accuracy of just under 50% is attainable.

## Problem Definition and Goals

The purpose of this project is to classify songs by genre using aural attributes. Accurate music genre classification is invaluable to streaming services like Spotify or Pandora and is primarily used for music recommendation systems or automated playlist development. Many genres and subgenres exist, but the scope of this project is limited to 9 popular genres: Blues, Classical, Country, Electronic, Folk, Hip-Hop, Jazz, Pop, and Rock. The goal is to use multi-class classification methods to predict the genre of songs as accurately as possible.

The dataset was obtained from Kaggle.com and the data was provided by Spotify. A link to the dataset is provided in the R notebook for this project. It includes 228,159 songs from 26 genres consisting of 18 features. The features were generated by Spotify and are as follows:

- Genre - The genre the song belongs to, as classified by Spotify
- Artist Name - The artist of the song
- Track Name - The name of the song

- Track ID - Spotify's internal identification number for the track
- Popularity - Measure of song popularity between 0 and 100
- Acousticness - Confidence measure of whether the song is acoustic
- Danceability - A measure of how suitable the song is for dancing
- Duration_ms - The length of the song in milliseconds
- Energy - A measure of song intensity and activity
- Instrumentalness - Confidence measure of whether a song is instrumental
- Key - The estimated key of the track
- Liveness - Probability that the track was recorded live
- Loudness - Overall track loudness in decibels
- Mode - A binary variable to indicate if a track is major or minor
- Speechiness - A measure of spoken words in a song
- Tempo - Estimated beats per minute
- Time_signature - The predicted number of beats in a measure
- Valence - A measure of musical positiveness (e.g. happiness, cheerfulness, etc.)

Before training the machine learning models, the data needs to be cleaned, downsampled, and preprocessed. The exact treatment is described in the "Data Exploration and Preprocessing" section. Once the dataset is properly prepared, various classifiers are trained and tuned. Finally, the models' out-of-sample performance is compared using accuracy and AUC as performance measures.

## Related Work

Accurate music genre classification is an ongoing challenge in the field of music information retrieval. In 2010, researchers trained a convolutional neural network to predict song genre using Mel Frequency Cepstral Coefficients (MFCC's) extracted from a dataset of 1,000 30-second song clips. The resulting model was similar to

image recognition models since the MFCC's were used as a visual representation of a song's frequencies at a given point in time. Their model performed well when classifying 4 or less genres, but only achieved around 35% accuracy with a 6-genre dataset [1].

Researchers at Stanford used non-visualized MFCC's from a dataset of 400 song clips to train a k-nearest neighbors model, a support vector machine, and a neural network. Based on the results of [1], they limited their experiment to a 4-genre dataset. To increase classification accuracy, they selected distinct genres - Classical, Jazz, Metal, and Pop. Their neural network classified songs from these genres with 96% accuracy [2].

Unlike [1] and [2], this project attempts to classify songs using a relatively high-level approach. The frequency data has been abstracted into attributes like key, mode, tempo, and valence. We also have additional attributes such as duration and time signature that will hopefully increase classification accuracy.

## Data Exploration and Preprocessing

The first step of this project was to remove unnecessary variables and some of the excess genres in the data. Artist name, track name, track ID, and popularity were removed, leaving 13 aural attributes not including the class variable. Considering the results of related work, it is safe to assume a 26-genre dataset would not yield good results. For this reason, obscure genres and subgenres were removed to leave only the 9 popular genres listed in the first section of this paper. Still, the dataset contained over 8,000 songs for each remaining genre, which made training the classifiers impractically slow. The dplyr package was used to downsample the data evenly by the class variable, leaving 1,733 observations per class. There were no missing values. With this step complete, data exploration can begin.

Multi-class classification problems present some unique challenges relating to data exploration. Correlation coefficients and scatter plots can't be used because the class variable is categorical. Instead, heatmaps and chi-square tests were used to explore the relationship between genre and the three nominal variables: key, mode, and time signature. The ggplot2 package was used to generate the visualizations depicted in figures 1, 2, and 3. Darker blue indicates more frequent values among instances.

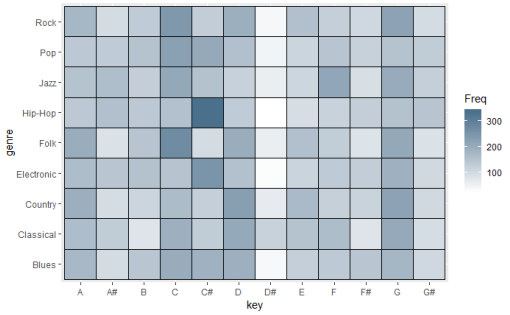


Figure 1: Key/Genre

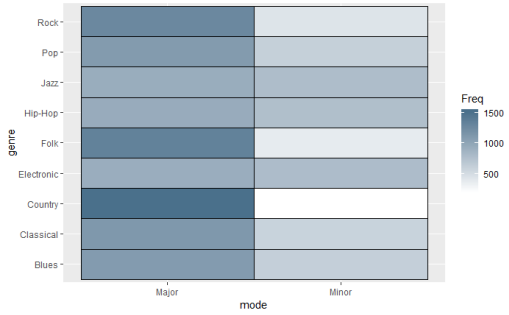


Figure 2: Mode/Genre

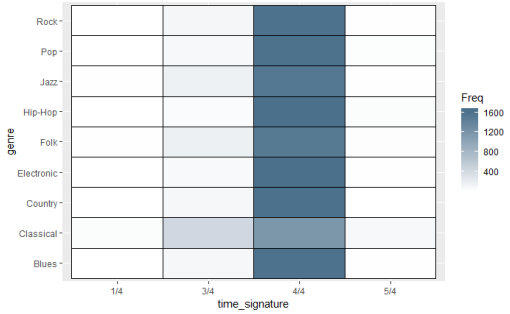


Figure 3: Time Signature/Genre

The heatmaps show that certain genres tend to prefer certain keys. Most genres use 4/4 time and major modes, but some use minor modes and odd time signatures.

Side-by-side box plots and ANOVA tests were used for the numeric variables, as shown in figures 4 through 13. Larger versions of the heatmaps and box plots can be viewed in the project's R notebook or the corresponding powerpoint.

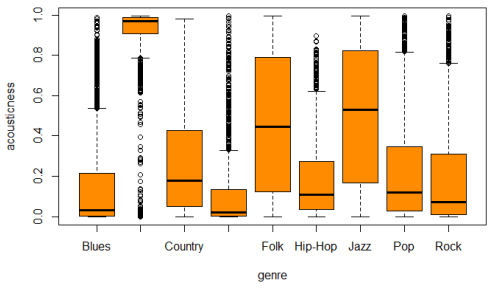


Figure 4: Tempo/Genre

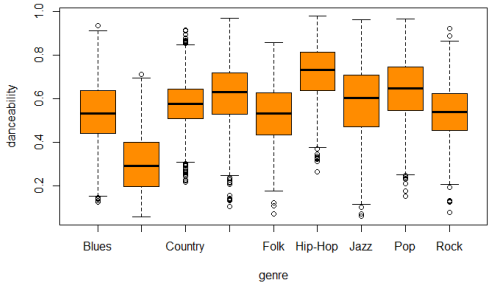


Figure 5: Valence/Genre

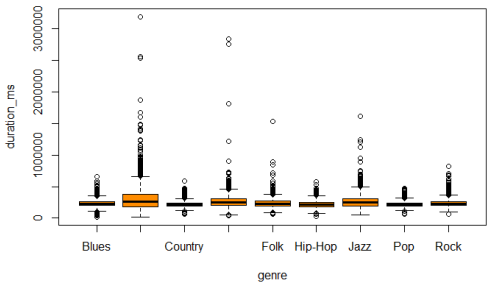


Figure 6: Duration/Genre

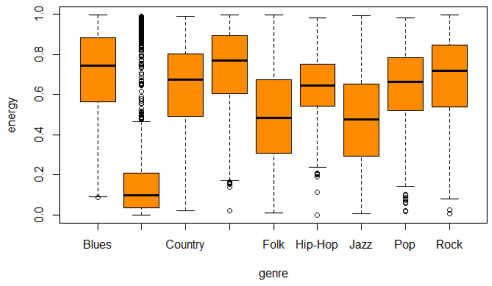


Figure 7: Energy/Genre

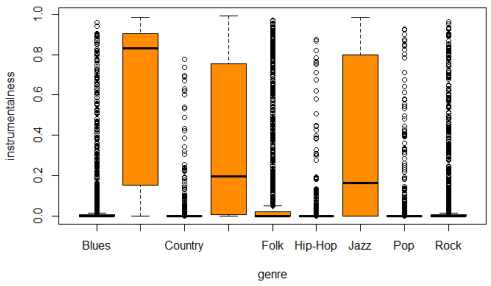


Figure 8: Instrumentalness/Genre

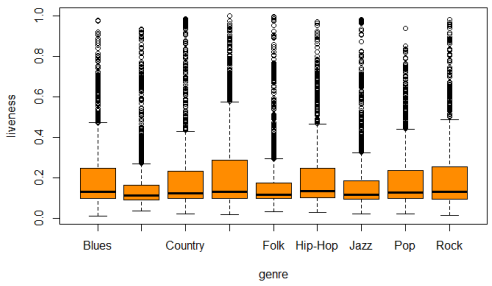


Figure 9: Liveness/Genre

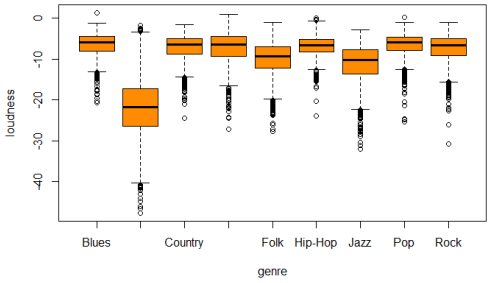


Figure 10: Loudness/Genre

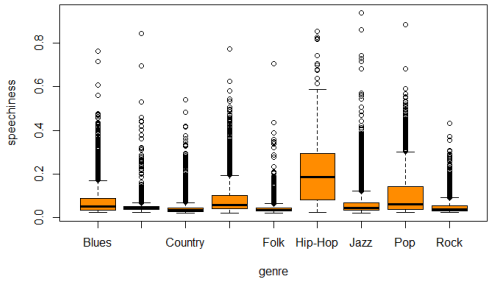


Figure 11: Speechiness/Genre

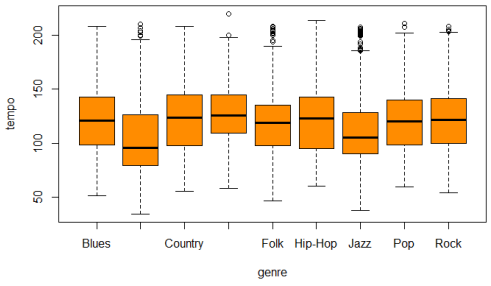


Figure 12: Tempo/Genre

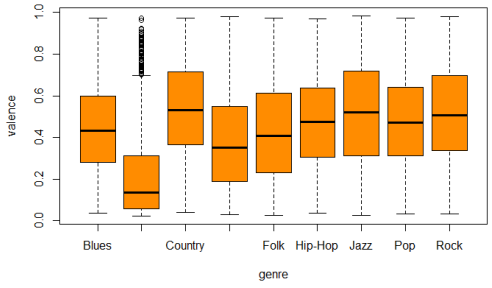


Figure 13: Valence/Genre

Most of the box plots show a noticeable between the means of each genre. The chi-square tests and ANOVA tests for each attribute show a p-value of almost 0, indicating that there is a relationship between each variable and the class variable. Each variable was used as-is and no feature engineering was necessary.

Next, the dataset was partitioned into train and test datasets using 90% of the data for training and the remainder for testing. The training and test datasets were duplicated and the factor variables were converted to numeric. The variables (except those already between 0 and 1) in the all-numeric datasets were normalized using min-max normalization. The all-numeric train and test datasets were used for the KNN, SVM, and neural network models, while the original train and test sets were used for the ensemble learners.

Additional preparation was needed before creating the neural network. First, it was necessary to create a validation dataset from the training data so the neural network can be tuned. Second, using neural networks for multi-class classification requires the class labels to be one-hot encoded; fortunately, the Keras package offers a convenient function to do this. The class labels were separated from the training and validation sets, then one-hot encoded using the to_categorical() function.

## Data Analysis and Results

The following models were used:

- KNN
- SVM
    - Linear Kernel
    - Radial Kernel
- Random Forest
- Gradient Boosted Tree
- Neural Network

Every model except the neural network was tuned using the caret package and 10-fold cross-validation. In most models, a custom tune grid was used. KNN was tuned using 8 values between 1 and the square root of the number of observations in the training set. The linear SVM cost parameter was tuned using 4 values between 0.5 and 5, though the values had almost no effect on the performance. Random

Forest was tuned using a tune grid of 6 values between 1 and the number of features in the dataset for mtry. GBM and radial SVM were the slowest classifiers to train, so caret was used to autotune the hyperparameters instead of using a custom tune grid. The neural network was tuned manually by trying different values for the mini-batch size, hidden layers, and neurons in the hidden layers. Even with a downsampled dataset, the classifiers took over two hours to train. After training, each model made predictions on the test dataset and the accuracy and AUC were stored. The results are shown below:

| Model | Accuracy | AUC |
|---|---|---|
| KNN | 42.3% | 0.64 |
| SVM Linear | 45.1% | 0.651 |
| SVM Radial | 46.7% | 0.651 |
| Random Forest | 47.5% | 0.668 |
| GBM | 49.4% | 0.657 |
| Neural Network | 45.9% | 0.653 |

GBM and random forest are the best-performing models. Although the performance is much better than the 'No Information Rate' of 0.1111, the accuracy and AUC are not very good. The classifiers are generally successful in predicting Classical, Electronic, Hip-Hop, and Jazz, but struggle to classify Pop and Rock songs, as evidenced by the following confusion matrix:

```
Confusion Matrix and Statistics

            Reference
Prediction   Blues Classical Country Electronic Folk Hip-Hop Jazz Pop Rock
  Blues         61         7       8         10   15      10    2   9   43
  Classical      0       153       1          2    5       0   15   0    0
  Country       24         1      98          3   31       5    3  34   43
  Electronic    14         4       3        109    8       6   27  11   14
  Folk          15         6      28          5   70       1   26  19   32
  Hip-Hop       23         1       9         13    4     117   12  51    5
  Jazz           5         1       3         23   24       4   77   1    8
  Pop           11         0      12          5    4      29    7  34    7
  Rock          20         0      11          3   12       1    4  14   21
```

Figure 14: Random Forest Confusion Matrix

The confusion matrix is from the random forest model, though the other models behave similarly. On average, the models misclassify around 85% of Rock songs and 75% of Pop songs. This could be because Pop and Rock artists tend to borrow heavily from other genres and lack distinction compared to genres like Classical, Electronic, and so on.

The models' performance could be improved by using a dataset with fewer genres, but the goal of this project was to maximize classification accuracy with the 9 popular genres listed in the first section using the given features. It's likely that additional features would increase accuracy. For instance, certain genres frequently use a distinguishing sequence of chords (e.g. the 12-bar blues progression). Instrumentation also plays a role in a song's genre - Electronic songs don't typically have a horns section, but Jazz songs might. However, given the features of the dataset, I don't think the classification accuracy of the models can improve much unless fewer genres are considered.

## Conclusion

Multi-genre classification is a difficult task. 3-genre or 4-genre datasets can achieve high accuracy, but are not as useful as datasets containing more genres. Additional genres result is greater classifier confusion, however. To achieve useful performance, we need a dataset with more features to accurately capture the intricacies of music genres and lessen confusion between music that is heavily influenced by other genres, like Rock or Pop.

The classifiers in this project did not perform well enough to be used confidently in any practical application, but the work presented here is a step towards accurately classifying song datasets with many genres. Still, the 9 genres included in the dataset are only a small fraction of the total number of music genres. Future research might involve subgenre classification (i.e. Garage rock, Punk rock, Glam rock, etc.) or extracting additional features from songs to increase genre descriptiveness.

## References

[1] T.L. Li, A. B. Chan, A. H. Chun, "Automatic musical pattern feature extraction using convolutional neural networks," in *Proc. Intl. MultiConference of Engineers and Computer Scientists (IMECS '10)*, March, pp. 546-550.

[2] M. Haggblade, Y. Hong, K. Kao, "Music genre classification," *Department of Computer Science Stanford University*, 2011.