Music Genre Classification

Caitlyn Yee, Dalia Jeiroudi, Justin Wolkowicz, Maclean Witmer, Zach Maguire, Joey Dronkers

Abstract

Recognizing the genre of a song from an audio clip is a problem with narrow yet useful benefits to solve. For instance, an application in music may want to use a model that automatically tags a song with a genre to reduce required user input. This paper focuses on various methodologies (e.g. similarity-based learning, deep learning) applied to a database of three-second song files. Each file is classified based on a subset of audio features determined through relative importance and we contrast six different learning models trained to predict the ten genres classified by the dataset. An overall accuracy of .61 among the multilayer perceptron and support vector machine models was the highest accuracy achieved among all approaches.

Introduction

Large databases of music are easily accessible to end-users, which has made classifying similar types of songs with a specific index, or genre, more important. This can be done by examining different musical features like harmonics, tempo in beats per minute, spectral features, and others. Thus, the aim of this project is to train several models on the same dataset and determine which model achieves the highest accuracy. Given audio files rich in information, an additional goal of this project is feature extraction in order to determine the attributes of audio of greatest impact. The models tested were K-Nearest Neighbors, logistic regression, random forest, multilayer perceptron, Naive Bayes, and support vector machine. These models were trained and tested on the GTZAN genre collection dataset, a popular collection of audio files belonging to ten different genre classes. These are hip-hop, classical, rock, pop, disco, country, metal, jazz, reggae, and rock. The dataset consists of 1000 songs evenly distributed across the 10 genres. To increase data samples used for training and testing, each of these songs was cut into ten three-second-clips resulting in a dataset composed of 10,000 three-second song samples. An issue with the extended dataset was that clips of the same song tended to bundle together in our similarity based models. This was rectified by ensuring clips from the same song could not be found in both training and testing sets.

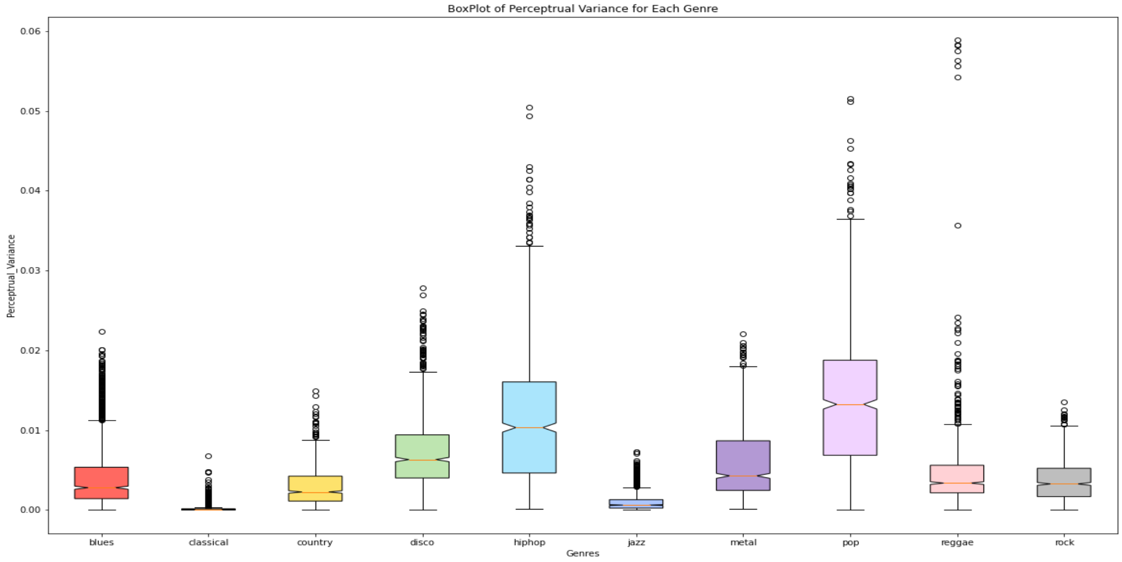
Previous Approaches

The practicality of music genre classification has generated lots of interest in the topic among researchers. The work of Bahuleyan explores the performance of various feature-engineered classifiers including logistic regression and support vector machines, while also contrasting their results to a spectrogram-based convolutional neural network [1]. A more common approach, as explored in the work of Haggblade et al., was to apply classifier models to the Mel frequency cepstral coefficients (MFCC) extracted from the audio files, which in essence would describe the overall shape of the sound [3]. Our approach to the classification problem is to combine components of both papers through applying a wide range of features extracted from the audio (including MFCCs) to various classifier models ranging from k-nearest neighbor, support vector machines, and multilayer perceptrons.

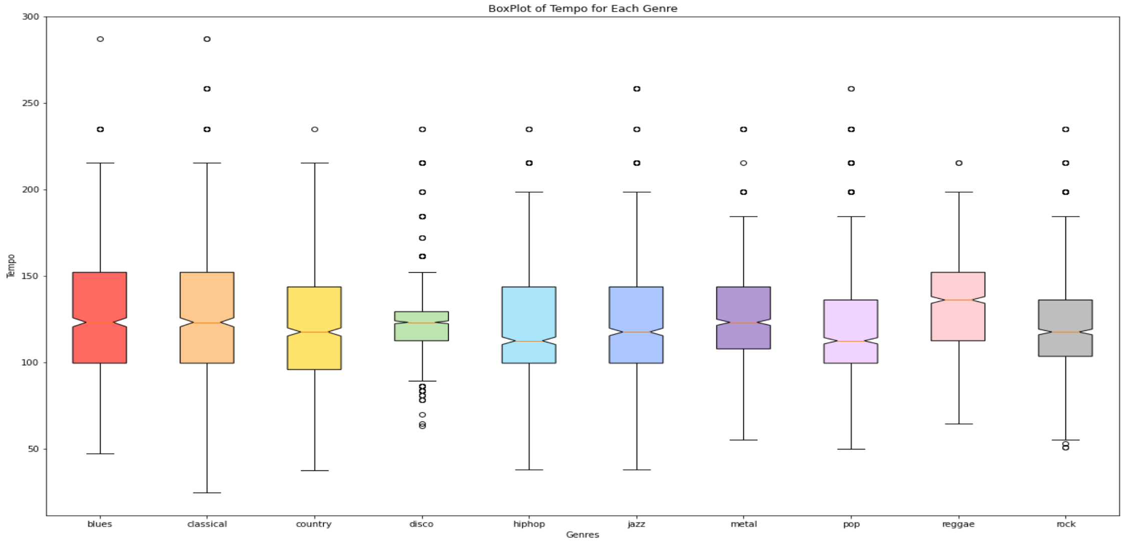
Exploring the Data

We initially extracted 58 features for each song in the dataset. For each musical feature, we found its value at each frame in its sound file, taking the mean of the frames as one feature and the variance of the frame as another. Once we accumulated all these features, we noticed that having this many could lead to overfitting. To restrain overfitting, we only used the most important features which had the most impact on the classification, or the ones that distinguish genres the best.

To illustrate the difference between an 'important' and 'unimportant' feature, we graphed boxplots of one of the most important features, perceptrual variance, and then graphed a less significant feature, tempo.



In this first boxplot of the variable perceptr_var, we can see how a machine learning model would be able to distinguish different genres easily. Pop and hip hop have a much higher mean than the other genres, while classical and jazz not only have a low mean but also very little standard deviation. For instance, this means that if a song has close to 0 perceptrual variance, it is much more likely to be classified as classical or jazz than it is hip hop or pop. This graph does not illustrate exactly what a model would predict as this is just one feature and there are many outliers for each genre, but it illustrates what makes perceptr_var an 'important' feature.



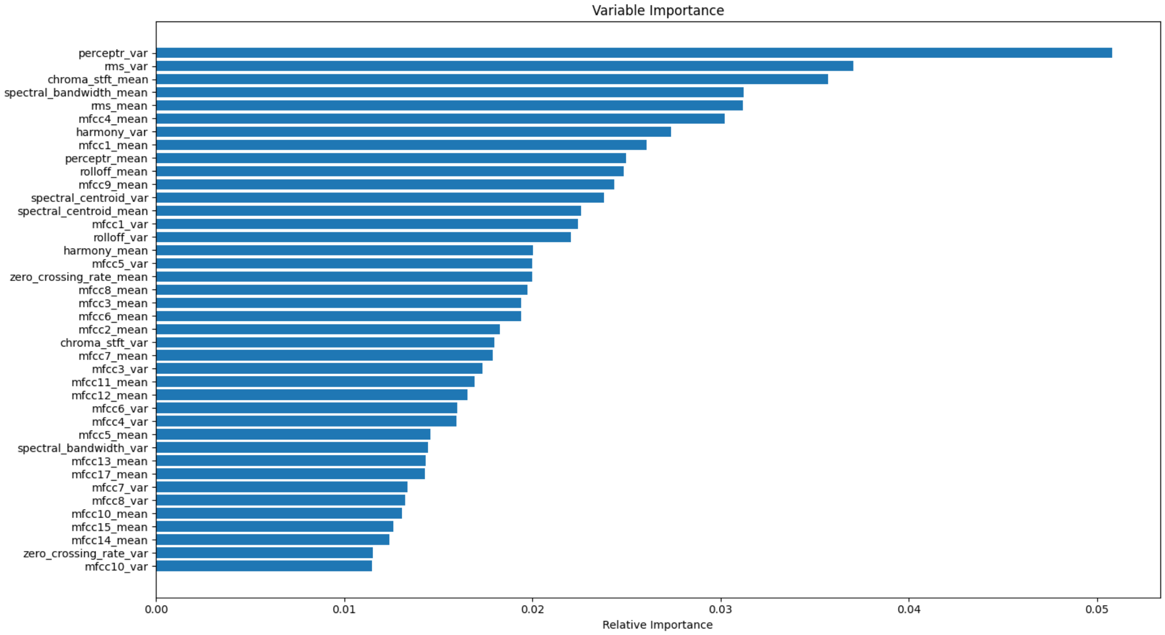
On the other hand, the boxplot for tempo shows little differentiation in tempo amongst all of the genres. Nearly every genre has the same plot, indicating that although we have an additional feature it is not useful for determining genre. Moreover, some of the features we deemed important in determining genre classification are as follows:

Chroma Frequencies	Root Mean Square (RMS)	Perceptual	Mel Frequency Cepstral Coefficients (MFCC)
Representation for music audio where the spectrum analyzes the pitch. Represents distinct chromas of an octave, creating a scale.	Root mean square of the energy in the audio.	Understanding shock waves (percussive sounds) as they represent the rhythm and emotion of the song.	A set of coefficients that define the shape and distinctions of the sound (timbre).

Methods

The data was tested on K-Nearest Neighbors, logistic regression, random forest, multilayer perceptron, Naive Bayes, and support vector machine by creating a pipeline. Given the number of models being tested, nested cross-validation would be very computationally complex. Therefore, in choosing optimal models through hyperparameter tuning, we opted to utilize an un-nested grid search using the training data. Because this was the case, we decided to only use a train/test split for our pipeline with no data reserved for a validation set. Given that the

GTZAN dataset contains only 1000 distinct songs of 30 second length, we decided to use an expanded dataset with the same songs where each was split into 3 second segments. In order to maintain the validity of the train and test split, we had to ensure that no song's segments would be split between the two sets. This was done by first applying the split to the 30 second dataset, whereupon the associated 3 second segments were sorted to their respective set. After the GTZAN data was split 70/30 for train/test, the pipeline standardized each split individually. From there, it regularized the data by stripping the features of less importance. The significance of the features pertaining to the class label was determined by using a random forest classifier on the training set, which then ranked the data features in order of importance. This determination aligned with our predictions for feature importance which were based on the above box plots.



The choice of relevant features was an effort in dimensionality reduction, as the number of features in the original dataset proved to be excessive. Using the dataset as constructed would lead to overfitting, likely requiring more complex models to fit the many columns of data. So, we opted to only utilize the 15 most relevant features as ranked by the classifier. After the data was processed with the aforementioned items, the pipeline then trained all the models and measured their respective accuracies on the test set.

The training process involved executing cross-validation using grid searches on each of the distinct learning models. Upon its completion, the best models for each respective learning method were fitted with the training data, whereupon they were further analyzed on the test set. Due to differing approaches, each learning method required distinct grids for their respective hyperparameters.

K-Nearest Neighbor (KNN) is a supervised learning algorithm that can be used for both regression and classification. In this case, it is a non-parametric classifier used to make proximity-based predictions about the grouping of a data point. The performance of KNN is dependent on a distance metric of choice (Euclidean, Manhattan, etc.) and the choice of *k*, the number of similar examples grouped to make a class prediction. Therefore the KNN models cross-validated between choices of *k* and various distance metrics.

Logistic regression is a method used to predict a binary outcome based on a weighted output of the logistic function. Given that the dataset contains more than two classes, a One-vs-Rest approach is used for Logistic regression, where each class was assigned a regression model against the rest of the classes. In learning, logistic regression utilizes gradient descent to converge to weights giving minimal in-sample error, so we chose to assess performance of models of varying regularization constants, *C*.

Random forest is a feature extraction method that utilizes decision trees to estimate classifications using an ensemble approach – a group outperforms an individual. Therefore, it was important to consider the number of trees, the depth of each tree, and the decision criterion for each tree. The random forest models were cross-validated on a grid of hyperparameters associated with these key factors.

Multilayer perceptrons (MLP) is a type of neural network characterized by multiple layers of input nodes connected as a directed graph between the input and output layers. The shape of such layers was an important hyperparameter, as it directly affected the feature extraction and classification of the model. The learning rate and regularization coefficient were hyperparameters to be considered since MLP training involves backpropagation, which is the process of computing gradients of in-sample error for gradient descent.

Naive Bayes is a probabilistic classifier that assumes independence among features. Namely, one feature in a class does not affect another feature in the class. This classifier chooses the class that has the highest probability given the data: max(P(class | data)). Given the relatively simple nature of the model, it was implemented under its default parameters.

Support vector machines (SVM) map data to a high dimensional space and then find the optimal boundary to divide the data into two classes. The optimal boundary is found by maximizing the distance between the hyperplane and the nearest data point. Similar to logistic regression, it requires an extension to be applicable for multiclass classification. In this case multiclass SVM also implements the One-vs-Rest approach. The choice of hyperparameters were the same as logistic regression, where the models were cross-validated over a range of regularization constants, *C*.

To evaluate whether the models' performances were satisfactory, we needed a baseline to evaluate all six models. Given that there are 10 different classes for which each song can be classified, a random baseline of 0.1 did not provide a good basis of performance. The Bahulean paper's feature based models showed accuracy of 57% for the SVM model and 54% for random forest [1]. However, the Chowdhry article showed an accuracy of 82% with the CNN model, but using feature based models such as KNN and SVM, with an accuracy between 55-60% [2]. The GTZAN dataset is very evenly distributed meaning that accuracy and F1-score will provide almost identical results on model performance. From this, the baseline to evaluate the models was set at 60% accuracy and F1-score alike.

Results and Discussion

Cross-Validation Results

KNN	MLP	NB	SVM	LR	RF
k = 29, metric = manhattan	hidden_layer_sizes = (100,), solver = sgd	default	C = 10, gamma = 0.1	C = 10	criterion = entropy, max_depth = 16, estimators = 500

Upon cross-validation, the above models were determined to be optimal and were chosen to be applied to the test set.

Model Mean Scores

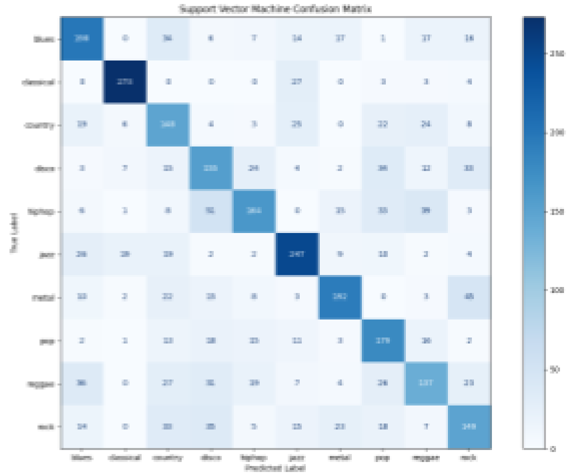
	Precision	Recall	F1-Score	Accuracy
KNN	0.60	0.59	0.58	0.59
MLP	0.62	0.61	0.61	0.61
NB	0.42	0.41	0.39	0.41
SVM	0.62	0.61	0.61	0.61
LR	0.51	0.52	0.52	0.52
RF	0.59	0.60	0.59	0.60

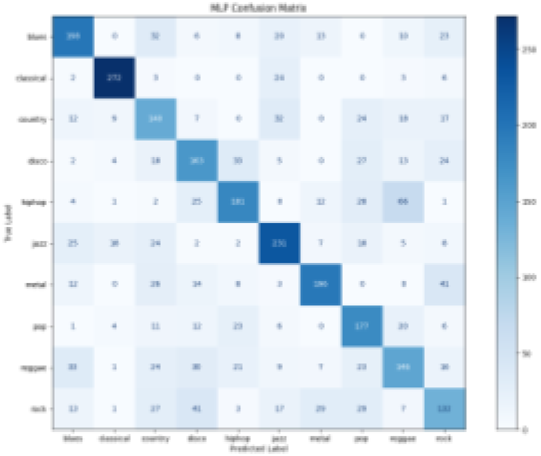
The averaged metrics of the models over the test set indicate that an approach of a neural-network (MLP) and a Support Vector Machine (SVM) achieve the best results in terms of accuracy and F1-score, which accounts for precision and recall. Average classification scores of Logistic Regression (LR) and Naive Bayes (NB) indicate that these approaches struggled to identify boundaries that could properly distinguish classes.

Model F1-Scores

	KNN	MLP	NB	SVM	LR	RF
Blues	0.63	0.65	0.35	0.63	0.56	0.66
Classical	0.84	0.88	0.70	0.88	0.84	0.84
Country	0.47	0.49	0.31	0.51	0.30	0.49
Disco	0.56	0.55	0.40	0.51	0.46	0.54
Hiphop	0.48	0.60	0.32	0.58	0.51	0.53
Jazz	0.66	0.67	0.33	0.71	0.58	0.68
Metal	0.67	0.70	0.60	0.68	0.69	0.68
Pop	0.66	0.60	0.51	0.61	0.53	0.63
Reggae	0.48	0.49	0.26	0.48	0.40	0.49
Rock	0.37	0.46	0.16	0.51	0.25	0.32

Over the individual classes, the model F1-scores varied. For example, although the mean precision between SVM and MLP were identical, the SVM model was able to classify rock and jazz slightly better than the MLP model. Therefore, the best model is genre-dependent with particular models being more accurate in distinguishing between particular genres. In this case, SVM distinguishes best between classical, country, jazz, and rock, and worst between blues and disco. For those particular genres, better performance is seen with MLP, KNN, and RF.





From the SVM and MLP's associated confusion matrices, it is difficult to assess a better overall model, as they both have their tradeoffs between various genres. Moreover, we see that a similarity-based approach (KNN) performs comparatively to the top models (SVM, MLP), but struggles significantly with particular genres such as hip hop and rock. As for naive Bayes, the model performs poorly throughout, achieving the worst results in nearly every category. One of the regression approaches achieves some of the best results (SVM) and outperforms the other (LR) in almost every class. For feature extraction approaches, the neural network (MLP) and random forest (RF) achieve comparable results, with the only distinction being the latter's poor performance on hip hop and rock.

As expected, a deep learning approach provided some of the best results, achieving relatively high F1-scores. This approach offered the best method of feature extraction, as neural networks transform high-dimensional data into easily bounded and classified shapes. As for our SVM model, the relatively high F1-scores were unexpected, but could be explained by the learning method's ability to identify optimal boundaries, especially in high-dimensional space. A similarity approach seemed intuitive, but the KNN model indicated that, with its decreased variance associated with its choice of hyperparameter (k=29), it was too simple to pick up the intricacies of some of the more disputed genres. The poor performance of Naive Bayes is likely due to its reliance on the independence of the features, which was not true in the context of this problem.

Conclusion

As shown by the results, the best learning methods as they pertain to genre classification are a multi-layered neural network (MLP) and support vector machines (SVM). Their success relative to the other learning methods tested in this paper can be attributed to their strong ability to extract features and identify boundaries within high-dimensional data. In the future, out-of-sample performance could be improved by utilizing a more modern dataset containing a larger sample of music. In doing so, the models would be trained on a sample that is more representative of the population we were trying to classify. However, the results in this paper indicate that in pursuing a more practical model for musical genre classification, a choice of a multi-layered neural network or support vector machines lead to higher performance.

References

[1] University of Waterloo. 2018. “Music Genre Classification using Machine Learning Techniques.” 12. https://arxiv.org/abs/1804.01149

[2] Chowdhry, Arsh. 2021. “Music Genre Classification Using CNN.” Clairvoyant. https://www.clairvoyant.ai/blog/music-genre-classification-using-cnn.

[3] Haggblade, Hong, and Kao. “Music Genre Classification.” http://cs229.stanford.edu/proj2011/HaggbladeHongKao-MusicGenreClassification.pdf