Genre classification of Music files

Puneet Girdhar
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080
Email: pxg151330@utdallas.edu

Dhruvkumar Patel
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080
Email: drp150030@utdallas.edu

***Abstract*—Music classification is an interesting problem with many applications. Music streaming services like Spotify, Apple Music and Google Play offer music catalog categorized into different genres and moods. It isn't clear however whether this process of music categorization is automated or manual. This feature is also not very commonly available for offline music files. The reason is that music genres are hard to describe systematically due to their subjective nature. In this project, we have applied audio signal processing and machine learning techniques to classify music files in different genres. We received comparable results to recent work in music genre classification task.**

Keywords- Music Genre Classification, MIR, Music Information Retrieval

I. Introduction

For our project, we have used GTZAN genre collection dataset, which is available as part of Marsyas audio processing framework. G. Tzanetakis and P. Cook curated this dataset in their work [1] on music genre classification. Even though they did not intend for this dataset to be standard dataset for work on music genre classification task [3], its easy availability on web has made it the primary dataset used by many [2] [3] [5]. This dataset consists of 1000 audio tracks, each of 30 seconds duration. These tracks are spanned over ten different genres namely: Classical, Blues, Country, Hip Hop, Disco, Jazz, Metal, Popular, Reggae, and Rock.

We investigate various machine learning algorithms on GTZAN dataset in this project, including K-nearest neighbours(KNN), multiclass SVM, and neural networks. For feature extraction task, we have relied on timbre-based features, namely Mel Frequency Cepstral Coefficients(MFCCs), rhythmic features as well as spectral features. This has been recommended by previous work on this problem [6]. There are many other audio features that can be used like power spectrum of the signal, FFT of audio signal, but it has been found that MFCCs represent given music file's timbre attributes, which is the most important aspect for genre classification problem. We have verified this by getting comparable accuracy values on given dataset as original work [1].

II. Feature Extraction

First step to any machine learning algorithm is to extract features i.e. to identify the components of audio signal that are good for classification. We are using Librosa, an opensource framework for music analysis in python for feature extraction. Analysis of some features is described below. In these descriptions, we have included images of feature vectors for each genre. These images help us in learning that how each feature is important in classyfing different genres.

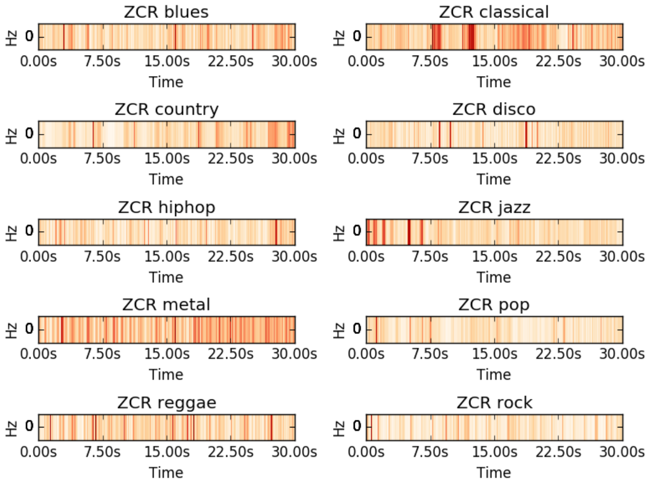
A. Zero Crossing Rate

Zero-crossings is the number of zero crossings of the signal in the time domain. It reflects the noisiness of the signal. Periodic sounds tend to have a small value of zero crossings while noisy sounds usually have a high value. The 'zero-crossing rate' is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back. [4]

ZCR is formally defined as:

$$zcr = \frac{1}{T-1} \sum_{t=1}^{T-1} f(s_t s_{t-1} < 0) \tag{1}$$

where f(t) is 1 if t is true.
Files from different genres from GTZAN dataset represented as zero crossing rate vectors are represented in given figure.



We can clearly see that different genres have different zero crossing rate distributions.

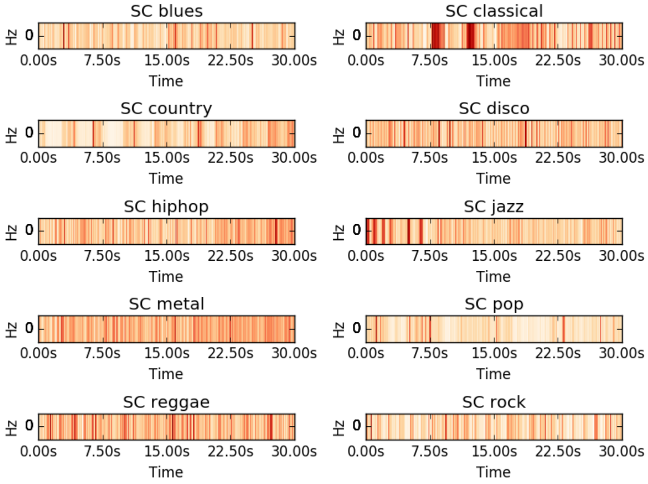
B. Spectral Centroid

The spectral centroid is a measure used in digital signal processing to characterise a spectrum. It indicates where the "center of mass" of the spectrum is. Perceptually, it has a robust connection with the impression of ”brightness” of a sound. [4]

It is calculated as the weighted mean of the frequencies present in the signal, determined using a Fourier transform, with their magnitudes as the weights:

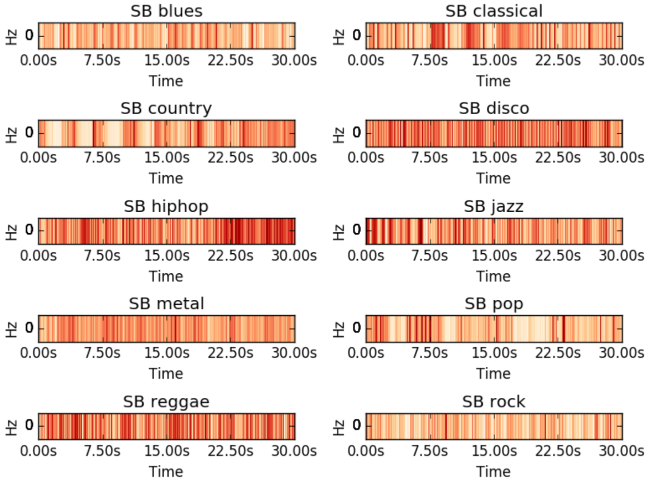
$$Centroid = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)} \tag{2}$$

Spectral Centroid distribution for different genre files is given in following figure.



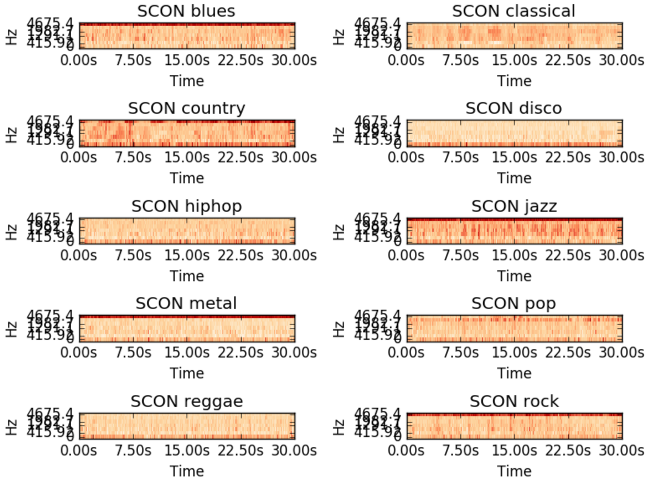
C. Spectral Bandwidth

Spectral Bandwidth is the Wavelength interval in which a radiated spectral quantity is not less than half its maximum value. Spectral Bandwidth distribution for different genre files is given in following figure.



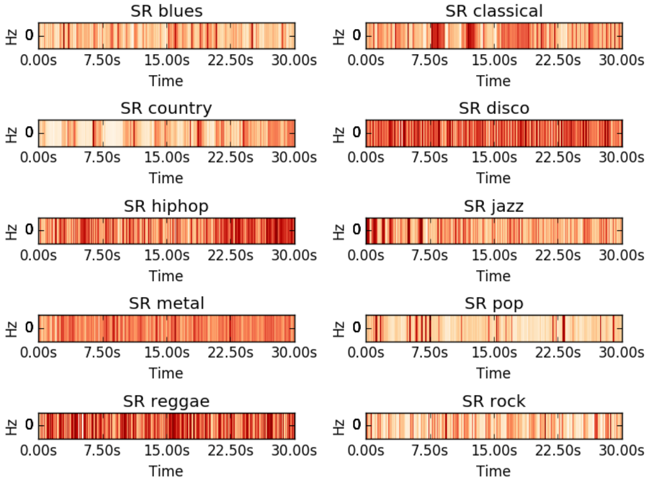
D. Spectral Contrast

Spectral contrast is defined as the decibel difference between peaks and valleys in the spectrum [7]. Spectral contrast distribution for different genre files is given in following figure.



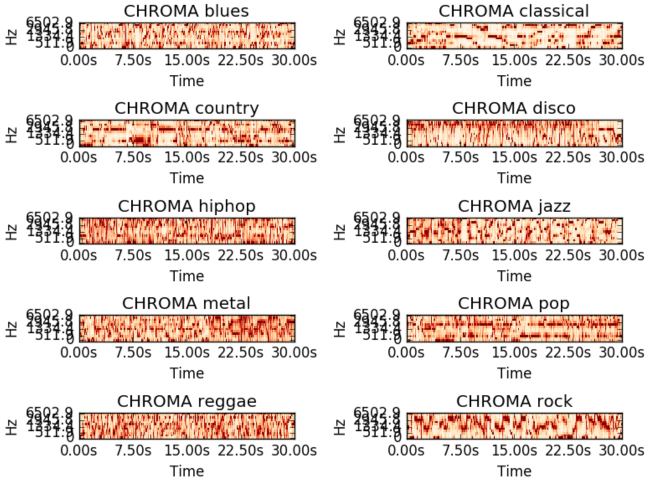
E. Spectral Rolloff

Spectral Rolloff is a measure of the amount of the right-skewedness of the power spectrum. The spectral rolloff point is the fraction of bins in the power spectrum at which 85% of the power is at lower frequencies [8]. Spectral Rolloff distribution for different genre files is given in following figure.



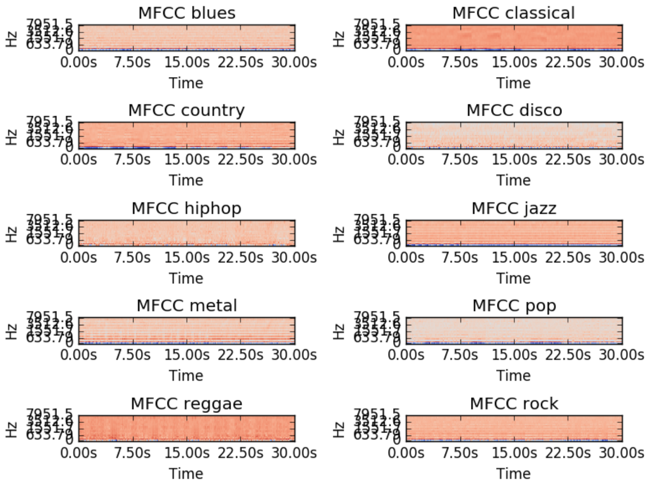
F. Chroma Vector

Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave. The term chroma closely relates to the twelve different pitch classes. Harmonic pitch class profiles (HPCP) is a group of features that a computer program extracts from an audio signal, based on a pitch class profilea descriptor proposed in the context of a chord recognition system. Often, the twelve pitch spelling attributes are also referred to as chroma and the HPCP features are closely related to what is called chroma features or chromagrams [4]. Chroma Vector distribution for different genre files is given in following figure. From these image representations, this feature looks more promising than others for genre classification.



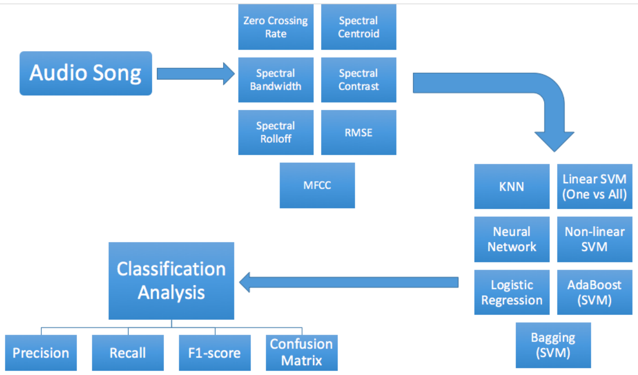
G. MFCC

MFCC, Mel-Frequency Cepstral Co-efficients, a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency [4]. MFCC vector distribution for different genre files is given in following figure.



III. Data preprocessing pipeline

We wrote a python script which reads audio files of 100 songs per genre, extract all features and saves in a sqlite database. Once feature vectors are stored, multiple models can be trained using combination of feature vectors. We further introduced custom matrix representation for each song which represents music feature mean vectors and covariances of those features as 1-D vector, effectively modelling features as Multi-variate Guassian distribution. Lastly, we applied different supervised learning algorithms using the reduced mean vector and covariance matrix as the features for each song to train on.



Given a piece of music, each section of song(20 msec) is transformed into a vector with N dimensions and hence complete song can be transfomed into a feature matrix of N rows and M columns where M = number of frames.

IV. Classification

Once the feature vectors were obtained, we trained classifiers on diffrent set of feature vectors. We divided the dataset in 57% - 10% - 33% ratio for training, validation and testing respectively. Feature selection is done manually by hit and trial method and parameter learning is learnt through minimizing error in cross validation set. Following were the different classifiers used:

1) Multi Label K Nearest neighbour
2) One-Vs-All approach with SVM
3) Neural Network
4) SVM
5) Logistic Regression
6) SGD Classifer
7) AdaBoost
8) Bagging Classifier

A. Multi Label K Nearest Neighbour

Intuitively, Music belongs to one genre should share same audio features. If we consider audio featuers as N dimensional vector then music belongs to one genre should be close to each other. This is similar to the idea behind multi-Label KNN algorithm in which it utilizes audio feature information of a music's k-nearest neighbour to infer its genres. We used KL divergence measure (described below) for calculating similarity between two songs.

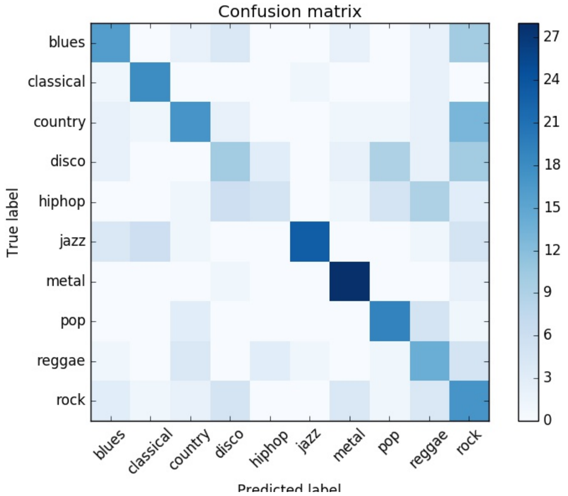
Consider $p(x)$ and $q(x)$ to be two song multivariate guassian distributions obtained from feature extraction. Then we have the following:

$$2KL(p||q) = \log\left(\frac{\sum_q}{\sum_p}\right) + Tr\left(\sum\nolimits_q^{-1}\sum\nolimits_p\right)$$
$$+(\mu_p - \mu_q)^T \sum\nolimits_q^{-1}(\mu_p - \mu_q) - d$$

Since, KL divergence is not symmetric but the distance should be symmetric, Hence we used following similarity measure:

$$D_{KL} = KL\left(p||q\right) + KL\left(q||p\right)$$

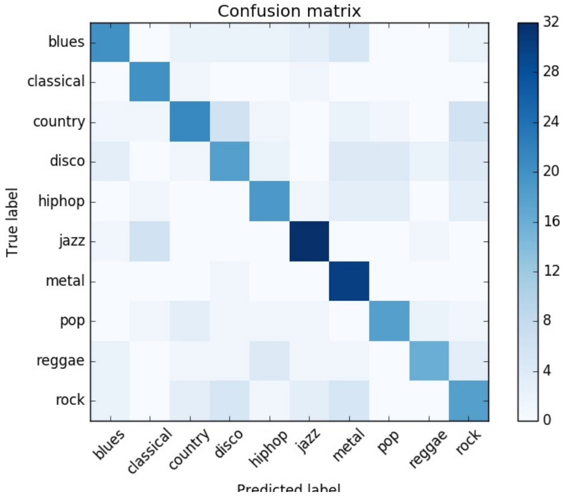
Here we experimented with multiple values of K and found out that K=11 yields the best output.



B. One vs All approach with SVM

Also known as OVR (One vs Rest) multi class strategy. For each classifier, the class is fitted against all other classes. In addition to its computationally efficency only n_classes classifiers are needed, one advantage of this approach is interpretability.

We experimented with different cost penalties (1e-5 to 1e+5), loss functions (hinge and squared hinge) and penalty (L1 and L2) and observed that combination of L2 regularized hinge loss with C=100 yields the best result with guassian kernel.



C. Neural Network

In order to train the neural network data set was first normalized. Normalization implies that all feature values from dataset should have zero mean and unit standard deviation.

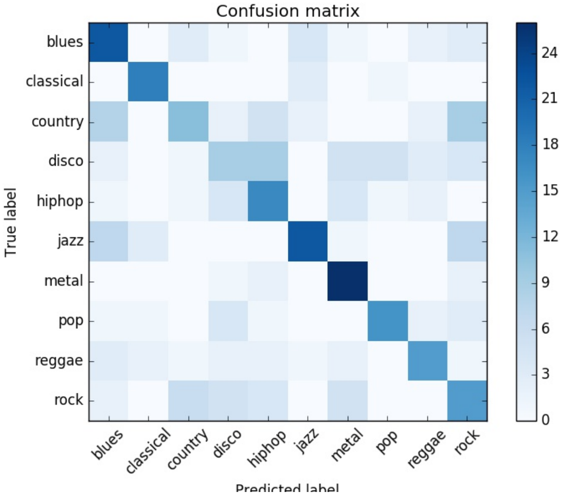
$$X_n = \frac{X_n - \mu}{\sigma}$$

where μ is mean feature value and σ is standard deviation.

We trained neural network with 10% of training instances as validation set which is the used to tune model parameters. We built a model with two hidden layer (10, 5), input layer size (256) and output layer size(10). We chose activation function as sigmoid function which gives us smooth non-linear function and cost function with regularization.

We also attempted to utilize PCA to reduce the dimension of input data but didn't improve our accuracy much.

Confusion matrix for ANN is given below.

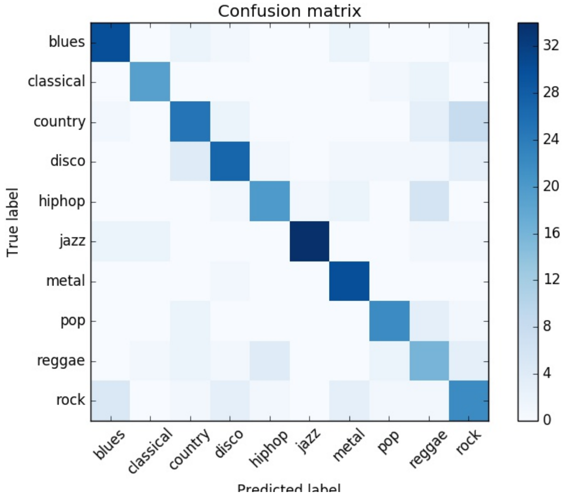


D. SVM

Same as with Neural network, we first normalized the data and then feed it into SVM. SVMs are based on two properties: margin maximization (which allows for a good generalization of the classifier) and nonlinear transformation of the feature space with kernels(as data set is more easily seperable in high dimensional feature space).

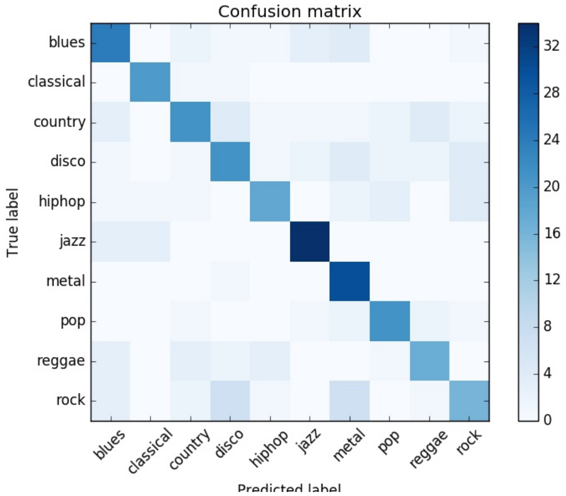
We used Python scikit package to implement Non-linear SVM algorithm. It implements Öne-against-oneäpproach (Knerr et. al 1990) for multiclass classification. if n_class is the number of classes, then $n\_class * (n\_class - 1)/2$ classifiers are constructed and each one trains data from two classes.

Confusion matrix for SVM is given below.



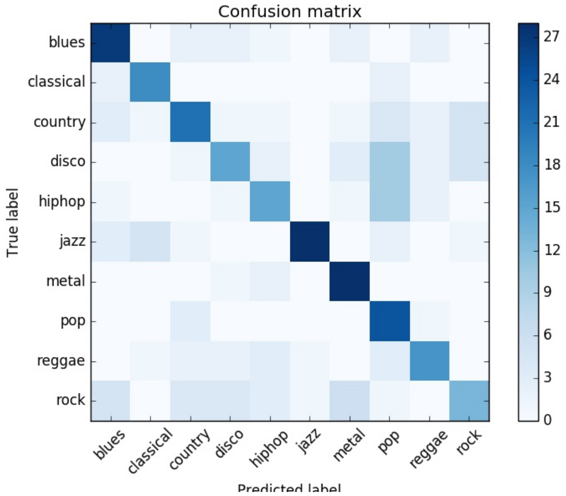
E. Logistics Regression

It is again implemented using One-vs-rest scheme. We used regularized logistics regression with lbfgs solver. LBFGS solver works with L2 regularization. where best hyperparameter is selected by cross-validator StratifiedKFold mechanism. Confusion matrix for Logistic Regression is given below.



F. SGD classifier

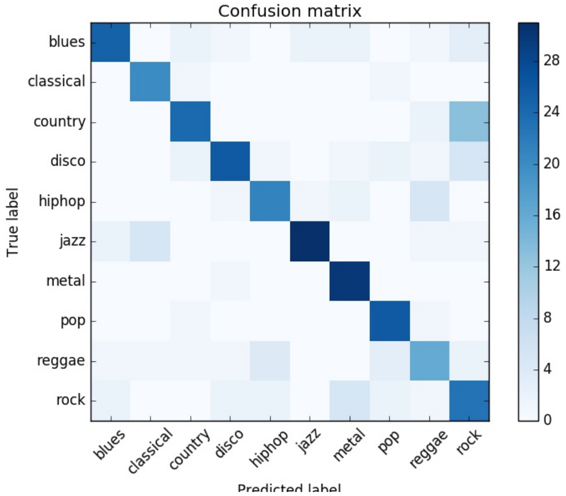
This estimator implements regularized linear models with stochastic gradient descent(SGD) learning. As with other classifiers, we first normalized the data with zero mean and unit variance and trained this model with "hinge" loss and L2 regularization. Optimal values were found by cross examining with minimizing error on K Fold random set.



G. AdaBoost

We used Adaboost algorithm to fit a sequence of weak learners on repeatedly modified versions of data. We experimented it with number of esimtaros and $learning_rate$ parameter. $n_estimator$ controls the number of iterations on models and $learning_rate$ controls the contribution of weak learners in final combination.

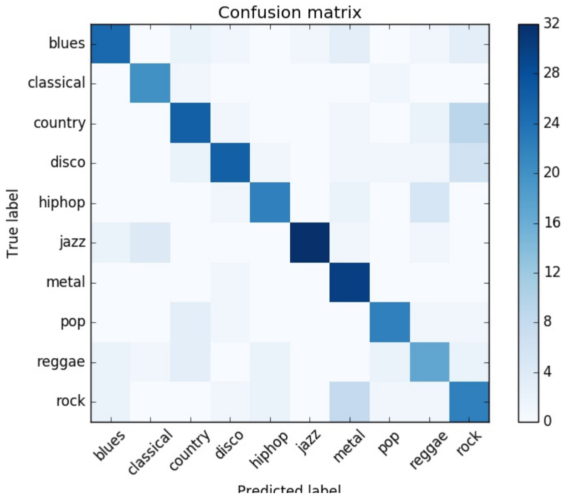
Here we experimented it with SGDClassifier as base estimator for adaboost and observed better results that original estimator. Confusion matrix for AdaBoost is given below.



H. Bagging Classifier

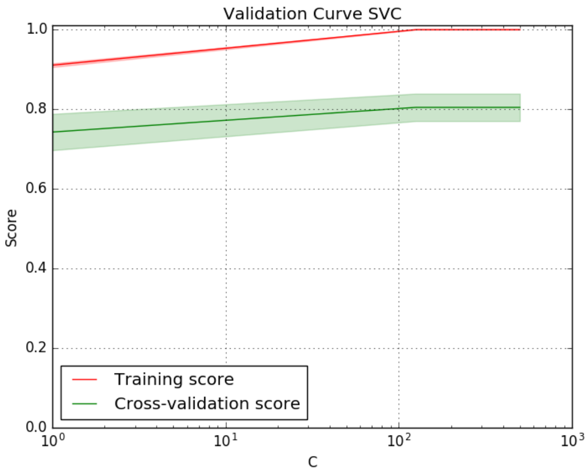
We have also experimented on Bagging classifiers which fits base classifier each on random subsets of the original dataset and then aggregate the individual predictions(either by voting or by averaging) to form a final prediction. Such a meta-estimator is very useful in reducing the variance of base estimator by introducing randomization into its construction procedure and then making ensemble out of it.

Here we experimented it with SGDClassifier as base estimator in bagging classifier with varying number of model estimators. Confusion matrix for Bagging classifier is given below.



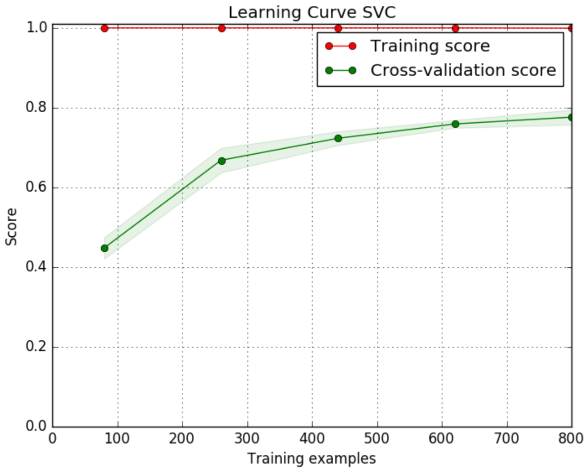
V. Parameter Learning

Hyperparameters of our classifiers are learnt based on the performance on validation set. Here we are showing parameter learning technique to Non-linear SVM classifier. We can clearly see in following figure that C=100 is optimal value for SVM parameter:



VI. Overfitting Analysis

To validate our SVM model that it does not suffer from bias/variance error, we applied learning curve analysis, which shows the training score is much greater than the validation score. Adding more training samples will most likely increase generalization. This proves that our model does not suffer form any bias vairance errors.

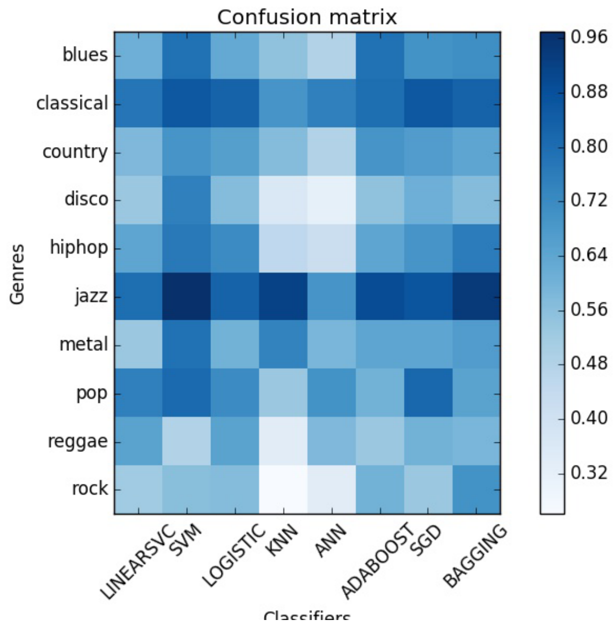


VII. Classification Metrics

Classification accuracy varied between the different machine learning techniques and genres. We got best accuracy with SVM classifier closed to 74% accuracy. Reggae and Rock were difficult ones to classify whereas jazz achieved highest classification score. Here are some confusion matrix results for different classifiers.

A. Precision

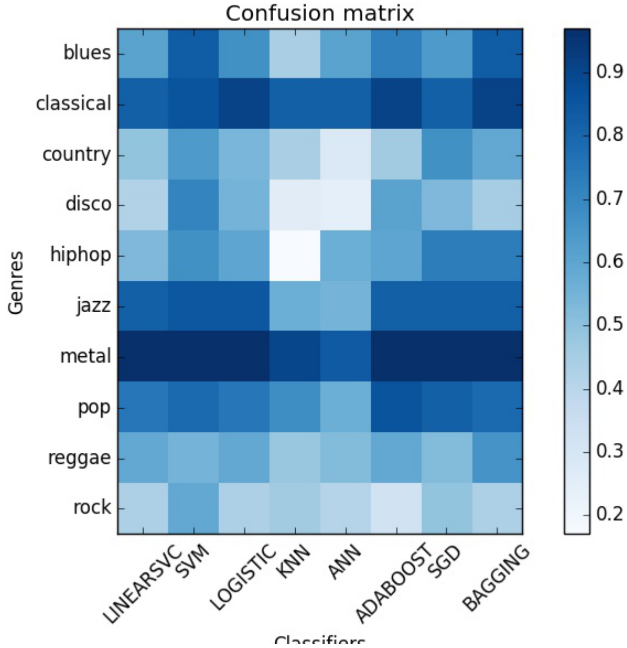
The precision is the ratio $\frac{tp}{tp+fp}$ where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. Below is the confusion matrix for precision values for different classifiers against different genres:



We can clearly see that Jazz genre is having highest precision for almost all classifiers. Classical genre seems second best and consistent precision values for all classifiers. Rock seems to have low precision consistenly, because it is often confused as Metal, as can be seen from true label axis of Rock genre in Confusion matrices of diffrent classifiers.

B. Recall

The recall is the ratio $\frac{tp}{tp+fn}$ where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. Below is the confusion matrix for Recall values for different classifiers against different genres:



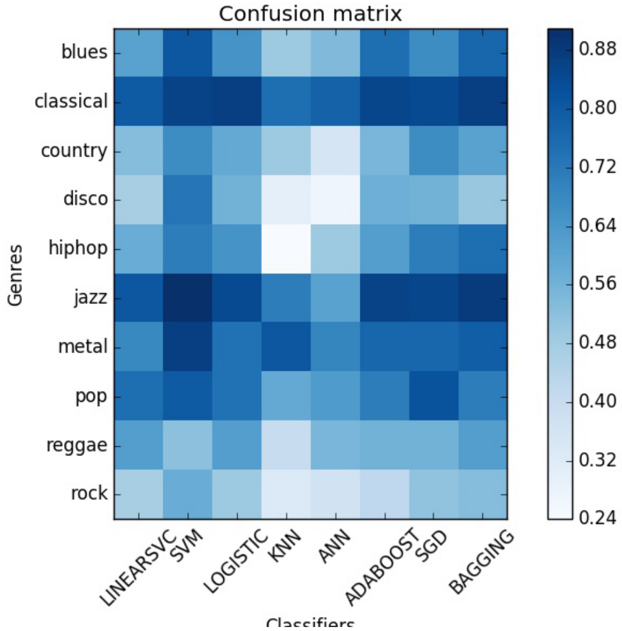
Metal genre is having the highest recall among all classifiers. We can clearly see in confusion matrix images of different classifiers that almost all true labels of Metal have been classified as Metal. That’s why the recall is high for Metal.

C. F1 Score

F1 score is harmonic mean of precision and recall values.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

Below is the confusion matrix for F1 values for different classifiers against different genres:



VIII. Conclusion

Here is accuracy report on test dataset for various classifiers.

CLASSIFIER	ACCURACY
SVM	0.74
One vs All SVM	0.63
SGD	0.66
ADABOOST	0.71
BAGGING	0.68
ARTIFICIAL NEURAL NETWORK (ANN)	0.51
LOGISTIC	0.70
KNN	0.53

TABLE I: My caption

According to our analysis, music files consist of many important features, especially for genre classification task. We were able to get better accuracy results by using mutiple features together. It is also observed that selection of classification algorithm affects the classification accuracy of different genres. In general, SVM outperformed other algorithms, but then again Neural Network was not used to its full potential. Some genres have better precision values, while other have better recall values and some genres have poor results for both. This means that, Jazz and Classical were overall easier to classify than other genres. In future work, we can apply neural network and especially deep learning architecture to this problem which can probably model this problem better. Also if we can acquire more domain knowledge about genre classification, we can apply Bayesian Network techniques and construct problem-specific Bayesian model for this problem.

References

[1] G. Tzanetakis and P. Cook, *Musical genre classification of audio signals*. IEEE Transactions on Audio and Speech Processing, 2002.

[2] Michael Haggblade, Yang Hong and Kenny Kao, *Music Genre Classification*. http://cs229.stanford.edu/proj2011/HaggbladeHongKao-MusicGenreClassification.pdf.

[3] Bob L. Sturm, *An Analysis of the GTZAN Music Genre Dataset*. MIRUM 2012.

[4] http://www.wikipedia.com

[5] Ioannis Panagakis, Emmanouil Benetos, and Constantine Kotropoulos, *Music genre classification: a multilinear approach*. ISMIR 2008.

[6] Fu, A., Lu, G., Ting, K.M., Zhang, D. *A Survey of Audio-Based Music Classification and Annotation*. IEEE transactions on multimedia 2010.

[7] Jun Yang, Fa-Long Luo, Arye Nehorai. *Spectral contrast enhancement: Algorithms and comparisons*. Speech Communication - Special issue on speech processing for hearing aids, Vol 39, 2003.

[8] http://jaudio.sourceforge.net/jaudio10/features/spectralrolloffpoint.html