ISyE 6416 – Basic Statistical Methods - Spring 2016
Music Genre Classification
Final Report

Team Member Names: Chen Feng, Mina Georgieva and Tony Yaacoub

Project Title: Music Genre Classification

1. Introduction

1.1 Background

Classifying the genre of a song, although a subjective task by itself, comes quite easily for the human ear. Within seconds of hearing a new song one can easily recognize the timbre, distinct instruments, beat, chord progression, lyrics, and genre of the song. For machines on the other hand this is quite a complex and daunting task as the whole "human" experience of listening to a song is transformed into a vector of features about the song. Historically, machines haven't been able to reliably detect many of these musical characteristics that humans recognize in music. Currently, machine learning algorithms haven't been able to surpass the 70% testing accuracy.

1.2 Goal and motivation

The aim of this project is to improve upon the accuracy of genre classification. We are considering a 10-genre classification problem with the following categories: classic pop and rock; classical; dance and electronics; fold; hip-hop; jazz and blues; metal, pop; punk; soul and reggae. The features we will use for classification are timbre, tempo, loudness information, time signature, key and mode.

YouTube, Spotify and similar websites lie behind the motivation for this project. Streaming or broadcasting websites rely on metadata to organize their musical content for easier search and access by the users. A metadata is simply information about the song – album name, artist name, song name, year of publication, genre, etc. While most of the information can easily be extracted from the title of the song, the genre is one of the important features that cannot be easily determined. A lot of the online musical content though lacks this important piece of information. Some websites like Spotify use manual (human) classification of the songs on their website. With the explosion of the musical content online categorizing songs manually can soon become unrealistic. Automatic genre classification would make this process much easier and faster, and it would also improve the quality of the music recommendations. Finally, it will allow for local artists to reach to a greater audience on the web.

2. Methods

2.1 Data

Our study is based on the Million Song Dataset, which is a freely-available collection of audio features and metadata for one million contemporary popular music tracks. The data was contributed by The Echo Nest, a music intelligence and data platform for developers and media companies, and sponsored by the National Science Foundation. For the scope of this project, a subset of the data set is considered – only 59,600 songs.

The data set contains the following information:

- Song ID, title and contributing artist
- Genre - 10 categories: classic pop and rock, folk, dance and electronica, jazz and blues, soul and reggae, punk, metal, classical, pop, hip-hop
- Loudness (numerical, from -40 to 0)
- Tempo (numerical, from 0 to 255)
- Time signature (categorical, from 0 to 7)
- Key (categorical, from 0 to 11)
- Mode (binary, 0 or 1)
- Duration (numerical)
- Average and variance of timbre vectors (numerical) – 12 variables for each (24 in total)

We consider all of the above variables except for song ID, title and contributing artist as our prediction variables for developing the model.

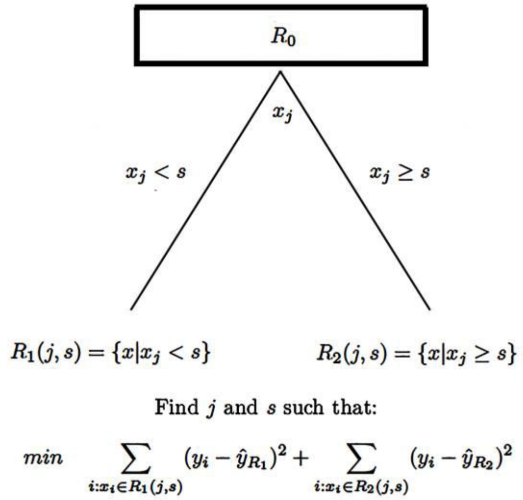
One of the main problems of this dataset is its unbalancedness and skewdness. The 'classic pop and rock' class, for example, is represented by 23,895 tracks, while the 'hip-hop' one has 434 tracks. For the genres, we rely solely on musicbrainz tags, but they could be wrong or incomplete since they were applied by humans and are usually very descriptive. These problems could account for a low testing accuracy, which needs paying extra attention to when we do the statistical analysis.

2.2 Statistical Analysis

2.2.1 Random Forest

We first turned to supervised learning methods, and selected random forest for building our first model. Random Forest is a method for classification and regression introduced by Breiman and Cutler. Firstly, the data is randomly divided into separate training and testing datasets without any overlapping – the training set is based on 80% of the original data set, and the testing set is based on the rest. Then, an ensemble of trees is grown from a bootstrapped sample of the training data. The method is based upon building a forest from multiple decision trees by averaging the predictions of all trees. The tree is grown by top-down binary partitioning – at each node (parent) we use greedy approach to split the node into no more than two children. We choose that “split” values that minimizes the error (see figure 1). We repeat for every node, until we reach a node with a very few observations and we stop. At the end we average all the trees (bagging).

Figure 1: Random forest - splitting a node


R_0
x_j
$x_j < s$
$x_j \geq s$
$R_1(j,s) = \{x|x_j < s\}$
$R_2(j,s) = \{x|x_j \geq s\}$
Find j and s such that:
$min \sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$

The advantages of random forest are that it's an accurate and fast learning model, works great with large datasets, and is resistant to overtraining. Random forest also achieves randomness as it is re-run multiple times and is training on different training datasets (through bootstrapping).

For our model, we decided to build 500 forests and average the predictions. Also, for our variable space, we decided to include a choice of $\sqrt{p}$ variables, where p is the number of predictors – 30 in our case.

2.2.2. K-means

For our second model, we used K-means clustering technique. K-means is an unsupervised learning technique, where only the features are used to predict the classes of data (i.e. only the 59600*30 matrix X of the features is used and not the genre y vector.) However, since we already know that we have 10 genres, we will use $k = 10$ as the number of clusters. In our work, we use the MATLAB built-in function 'kmeans' with 100 replicates. The built-in function chooses the output based on the replicate with the least total sum of within-cluster sums of point-to-centroid Euclidean distances, which is equivalent to finding a local minimum. Note that k-means does not necessarily converge to a global minimum so the best option is to search through various replicates, since at each replicate the initial centroids are chosen randomly. Furthermore, to eliminate the difference in variability between the predictors, we scale the predictors to be between 0 and 1.

To evaluate the performance of k-means, we run the algorithm on the whole data set. Since the algorithm only returns which point of the 59600 points belongs to which of the 10 clusters, and since the clusters cannot be distinguished as to which genre they map, we use a matrix P with the entries $P_{i,j}$ representing the proportion of the points from genre j that are identified in cluster i for $i, j = 1, \ldots, 10$. If the algorithm is working perfectly then we would observe that at each column, we would have an entry of 1 and the rest of zeros. The same applies to each row. If this is the case, then one can automatically map the cluster i where $P_{i,j} = 1$ to genre j.

2.2.3. Multinomial Logistic Regression

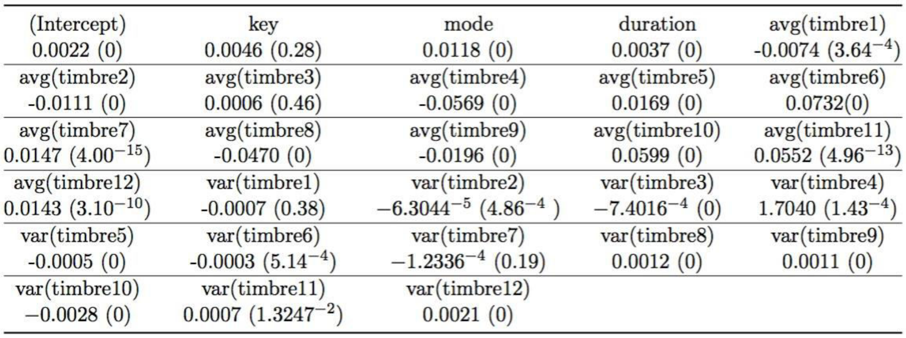
For our third model, we used multinomial logistic regression. We modeled the log odds of the outcomes as a linear combination of the predictor variables. We used classic pop and rock as our reference level for the model, and the remaining 9 categories were compared to that base level. We used the multinom function from the nnet package in R to estimate the model coefficients. Since p-values were not provided by the package, we used Wald tests (z-tests in this case) to calculate them.

For model illustration purposes, we can consider the genre folk as an example:

$$log\left(\frac{P(genre = fold)}{P(genre = classic\ pop\ and\ rock)}\right) = 0.0022\ +\ 0.0046key\ +\ 0.0118mode\ +\ 0.0037duration -$$
$$0.0074avg(timbre1) -\ 0.0111avg(timbre2) +\ 0.0006avg(timbre3) -\ 0.0569avg(timbre4) +$$
$$0.0169avg(timbre5) +\ 0.0732avg(timbre6) +\ 0.0147avg(timbre7) -\ 0.0470avg(timbre8) -$$
$$0.0196avg(timbre9) +\ 0.0599avg(timbre10) +\ 0.0552avg(timbre11) +\ 0.0143avg(timbre12) -$$
$$0.0007var(timbre1) -\ 6.3044var(timbre2) -\ 7.40164var(timbre3) +\ 1.7040var(timbre4) -$$
$$0.0005var(timbre5) -\ 0.0003var(timbre6) -\ 1.23364var(timbre7)\ +\ 0.0012var(timbre8) +$$
$$0.0011var(timbre9) -\ 0.0028var(timbre10)\ +\ 0.0007var(timbre11)\ +\ 0.0021var(timbre12)$$

The p-values for all the above coefficients are listed in Table 1. The only variables that were not significant at α level of 0.05 were key and some of the timbre vectors. The y variables can be interpreted as the log ratios of the probabilities of choosing one category (folk in this case) over the base level (classic pop and rock). The exponentiated regression coefficients are the relative risk ratios for a unit change in the predictor variable. For instance, one-unit increase in the key variable is associated with a of 0.0046 increase in the log odds of folk versus classic pop and rock.

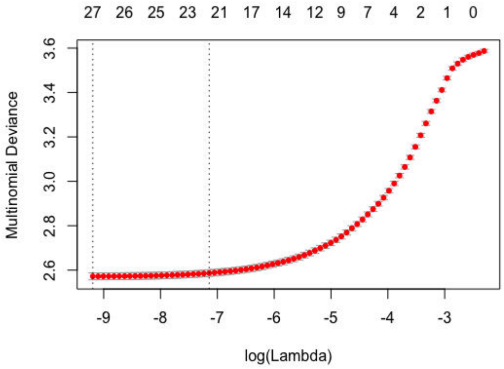
Table 1: Coefficients and p-values of the coefficients of log odds of folk vs. classic pop and rock



(Intercept)	key	mode	duration	avg(timbre1)
0.0022 (0)	0.0046 (0.28)	0.0118 (0)	0.0037 (0)	-0.0074 (3.64^{-4})
avg(timbre2)	avg(timbre3)	avg(timbre4)	avg(timbre5)	avg(timbre6)
-0.0111 (0)	0.0006 (0.46)	-0.0569 (0)	0.0169 (0)	0.0732(0)
avg(timbre7)	avg(timbre8)	avg(timbre9)	avg(timbre10)	avg(timbre11)
0.0147 (4.00^{-15})	-0.0470 (0)	-0.0196 (0)	0.0599 (0)	0.0552 (4.96^{-13})
avg(timbre12)	var(timbre1)	var(timbre2)	var(timbre3)	var(timbre4)
0.0143 (3.10^{-10})	-0.0007 (0.38)	-6.3044^{-5} (4.86^{-4})	-7.4016^{-4} (0)	1.7040 (1.43^{-4})
var(timbre5)	var(timbre6)	var(timbre7)	var(timbre8)	var(timbre9)
-0.0005 (0)	-0.0003 (5.14^{-4})	-1.2336^{-4} (0.19)	0.0012 (0)	0.0011 (0)
var(timbre10)	var(timbre11)	var(timbre12)		
−0.0028 (0)	0.0007 (1.3247^{-2})	0.0021 (0)		

We also performed two variable selection procedures – stepwise regression and LASSO. Stepwise regression eliminated two of the timbre features. LASSO was performed through the cv.glmnet function in R. LASSO aims to find the optimal λ that minimizes the cross-validated error, which is 1.019^{-4}. LASSO kept all variables. The multinomial deviance against $log(\lambda)$ plot is shown below in figure 2:

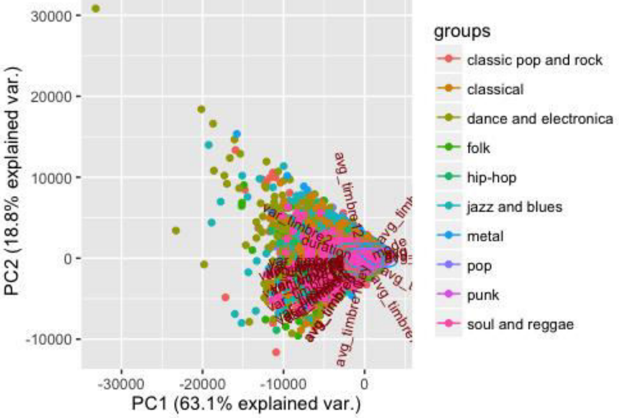
Figure 2: LASSO for multinomial logistic regression



2.2.4. LDA and QDA

Next, we performed Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). Both LDA and QDA are supervised methods that try to identify attributes that account for the highest variance among classes. Prior to applying LDA and QDA, we used a Principal Component Analysis (PCA). PCA identifies combinations of features (principal components) that account for the biggest variance in the data. We use PCA to regularize the problem and avoid overfitting. We then used those principal components, or directions, to project the original data points onto the first two components (figure 3). Even though there are some visible cluster patterns, most of the data points are clustered around the origin. We then proceeded with LDA and QDA on our new data set. Note that we used leave-one-out cross validation to estimate the average performance of both LDA and QDA.

Figure 3: PCA's 2 principal components projection



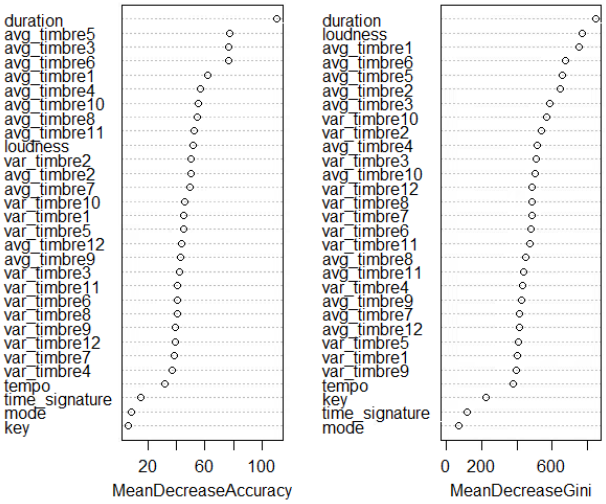
3. Results

3.1 Random Forest

We first proceeded with random forest model on all 30 variables. The model was built on 200 trees, and at every node we allowed a choice of 5 variables for making the split decision. The out-of-bag error was 42.92% (equivalent to 57.08% success rate). The out-of-bag error was based on whatever was left out of the training data set after bootstrapped sampling. The successful classification for our testing dataset was 57.69% which surpassed most current literature’s achieved results.

As we can see from figure 2 below, the mode, key, tempo and time signature of the song were insignificant. The most important variables were duration and average timbre vectors. Based on intuition, we expected that tempo would be one of the significant variables, but it turned out to be far from important.

Figure 4: Importance of variables for random forest



Although the overall success rate was around 57%, some categories achieved a very low success rate, while others very high. Classical and metal were the two most easily recognizable genres with success rate of 83% and 76% respectively. Hip-hop and classic pop and rock on the other hand were confirmed to be the least differentiable categories with corresponding success rates of only 9% and 29%. Hip-hop was mostly confused with soul and reggae, and classic pop and rock was confused with folk and soul and reggae.

Table 2: Confusion matrix for random forest

	classic pop and rock	**classical**	**dance and electronica**	**folk**	**hip-hop**	**jazz and blues**	**metal**	**pop**	**punk**	**soul and reggae**	**error**
classic pop and rock	563	22	125	362	1	139	74	173	186	288	71%
classical	23	1246	65	56	1	58	20	18	7	15	17%
dance and electronica	103	69	1079	101	2	168	56	70	82	210	44%
folk	233	50	35	1089	0	192	12	128	61	116	43%
hip-hop	12	3	69	15	32	5	5	21	9	171	91%
jazz and blues	125	128	107	196	0	1116	24	39	37	129	41%
metal	70	11	45	33	0	25	1282	11	192	17	24%
pop	175	3	81	169	0	37	18	588	56	157	54%
punk	153	13	78	72	3	33	162	57	1183	173	39%
soul and reggae	184	4	174	137	2	86	2	134	28	1146	40%

3.2 K-means

The performance of k-means is evaluated with the use of a matrix P, with the entries $P_{i,j}$ representing the proportion of the points from genre j that are identified in cluster i for $i, j = 1, \dots, 10$. The matrix P, obtained after running 100 replicates and choosing the best model, is summarized below.

Table 3: Entries of Matrix P

	Classic pop and rock	Classical	Dance and electronica	Folk	Hip-hop	Jazz and blues	Metal	Pop	Punk	Soul and Reggae
1	0.2893	0.3794	0.3973	0.4365	0.3490	0.4257	0.3719	0.3562	0.3320	0.4663
2	0.0781	0.0220	0.0186	0.0155	0.0468	0.0252	0.0667	0.0666	0.0728	0.0069
3	0.0972	0.0997	0.1366	0.0727	0.1007	0.0619	0.0490	0.0467	0.0775	0.0812
4	0.3145	0.1656	0.1274	0.2480	0.1871	0.2334	0.3001	0.3090	0.2158	0.1857
5	0.0041	0.0087	0.0180	0.0052	0.0080	0.0065	0.0027	0.0024	0.0060	0.0076
6	0.1434	0.0951	0.0728	0.0525	0.1105	0.0694	0.0586	0.0680	0.0959	0.0399
7	0.0037	0.0517	0.0439	0.0321	0.0529	0.0382	0.0264	0.0235	0.0286	0.0379
8	0.0069	0.0349	0.0303	0.0311	0.0267	0.0290	0.0096	0.0147	0.0120	0.0378
9	0.0093	0.0445	0.0568	0.0488	0.0457	0.0508	0.0799	0.0728	0.0881	0.0647
10	0.0536	0.0985	0.0982	0.0577	0.0725	0.0599	0.0351	0.0403	0.0715	0.0719

From the table above, one can notice that most of the songs in each genre are identified in clusters 1 and 4, making it difficult to distinguish as to which cluster belongs to which genre. For instance, 46.63% of the Soul and Reggae songs are identified in cluster 1, but also 42.57% of the Jazz and Blues songs are also identified in cluster 1. Thus, one can see that k-means does not perform well in distinguishing between the genres.

3.3 Multinomial Logistic Regression

The performance of the model is evaluated by the following test error:

$$TE = \frac{1}{n}\sum_{i=0}^{n_1} I\left(Y_i \neq \hat{f}(x_i)\right),$$

where n_1 is the number of testing data points, $I\left(Y_i \neq \hat{f}(x_i)\right) = 1$ if $Y_i \neq \hat{f}(x_i)$, and , $I\left(Y_i \neq \hat{f}(x_i)\right) = 0$ otherwise. To further assess the robustness of each method, we repeated the above computation 10 times, i.e. we did 10 loops to get the average performance of each model.

Unfortunately, the results were not terribly promising. The average test error for each method is presented in Table 2. Some of the reasons for such performance might be: inappropriate assumption of linear relationship between log odds ratios and the variables; nonlinear relationships which cannot be captured by logistic regression or incorrect assumptions of independence. For instance, the assumption of variable independence can be compromised due to the timbres always following certain patterns in the pursuit of harmony and rhythm. Therefore, it is quite plausible that they correlate with each other. Finally, the dataset itself is very imbalanced as we mentioned before. Altogether, these can result in the low classification accuracy.

Table 4: Test errors for logistic regression

Logistic regression	Stepwise regression in logistic regression	LASSO in logistic regression
0.5003356	0.4930369	0.6098154

3.4 LDA and QDA

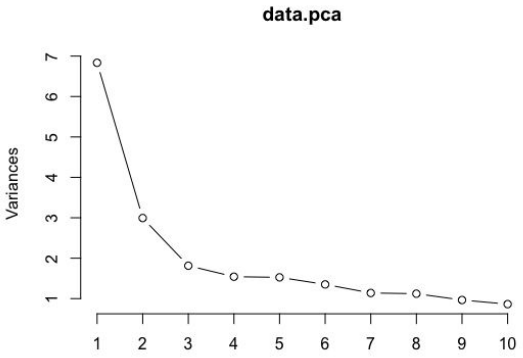
The computation time for LDA and QDA was smaller than that of the logistic regression. Therefore, we run 100 loops to get the average test errors. The corresponding results for LDA and QDA on the original dataset and on new dataset obtained from PCA are shown in Table 4. PCA did not improve the classification performance. Figure 5 shows a plot of the variances (y-axis) associated with the first 10 PCs (x-axis). In our case with 27 PCs, we chose 2 PCs to preserve for further analysis. However, our dataset is quite unbalanced. Although the first two PCs explain most of the variability, the total variability is as high as 81.9%, which led to poor classification performance.

The overall performances of LDA and QDA were not satisfying. The normal distribution assumptions for LDA and QDA are too strong to guarantee classification accuracy. Interestingly, LDA performed better than QDA with respect to the original dataset according to our simulation. This suggests that the nonlinear decision boundary does not necessarily gives better classification results.

Table 5: Test errors for discriminant analysis

LDA	QDA	LDA with PCA	QDA with PCA
0.4530839	0.5088893	0.6019295	0.5911913

Figure 5: Variance versus PC



4. Discussion

We observed how different classifiers performed on categorizing music genre based on the specific song features. Although random forest performed the best, it reached a success rate of only 58%. Adding more features such as multi-word context to provide a more complete picture of the song's genre might improve the accuracy of music classification. Also, a more balanced and representative data might lead to better results. One last possibility we haven't considered in this project is looking at the top two choices of genre instead of just one. We still have to keep in mind that boundaries separating music genres are often blurry and subjective. Even humans cannot achieve a perfect accuracy rate. One last challenge that couldn't be tackled in this project was signal processing techniques used to capture the musical features of songs the way humans can hear and recognize them. More advanced signal processing methods will help for a more reliable feature detection and selection.