Music Genre Recognition

Bhanu Kaushik Kanamarlapudi

kanamar1@msu.edu

1. Introduction

One of the key problems in music information retrieval is music classification which has many potential applications. One such application is developing a classification system to provide end users with option to search for type of music they are interested in. Music genres are categories that have arisen through a complex interplay of cultures, artists, and market forces to characterize similarities between compositions and organize music collections. But the boundaries between the genres remain fuzzy, thus making music genre recognition a non-trivial task. Music genre classification forms a basic step for building a strong recommendation system.

The main objective of automating the music genre classification is to make the selection of songs quick and more manageable. To manually classify songs, one has to listen to a whole lot of songs and then assign the genres. This a very difficult process and time consuming too. Automating the classification of music can help finding valuable data such as current trends, popular genres, and artists. Determining the music genres is the first step for building an efficient recommendation system. The aim of this project is to try various techniques such as feature selection techniques, feature transformation techniques, etc. that can improve the performance and accuracy of machine learning models to classify songs into different genres based on its metadata.

The rest of the report is organized as follows. The next section provides details about the datasets used and the description of the datasets. The methodology section provides the details of the implementation flow and the methods used. The results and analysis section shows insights about the performance and accuracy of the classification models. And the report concludes by providing summary and conclusion of the experiments.

2. Datasets

The dataset used for this project is Free Music Archive (FMA) dataset, which is an open and easily accessible dataset and has various metadata files related to music. Out of these metadata files, I used only 3 different metadata files namely – track.csv, genres.csv, echonest.csv.

Track metadata (track.csv) is collected through API. It contains metadata related to per-track, per-album, and per-artist. Per-track metadata contains features such as bit_rate, composer, duration, favorites, genre_top, genres, genres_all, number of interests, number of listens, lyricist, publisher, tags,title for each track-id, etc. Per-album metadata contains comments, date_created, date_released, favorites, id, listens, producer, tags, title, tracks, type for each track-id, etc. Per-artist metadata contains features such as active_year_begin, active_year_end, bio, favorites, id, location, related projects, tags, websites for each track-id, etc. I just listed some of the features in track dataset though there are many more. In total, the Track dataset has 106,574 samples and 52 features.

Genre metadata dataset (genres.csv) contains number of tracks, parent genre, title of the genre and top-level information for each of 163 genres. Out of these 163 genres, only 16 are parent/root genres and rest of them are sub-genres. Echonest dataset (echonest.csv) contains 249 features extracted for around 13,129 tracks using the echonest API. It contains metadata of features such as album_date, album_name, artist_name, artist_location and audio_features like acousticness, danceability, energy, instrumentalness, liveliness, speechiness, tempo and valence and social_features such as artist_discovery, artist_familiarity, artist_hotttnesss, song_currency, song_hottness and it's corresponding rank related features like artist_discovery_rank, artist_familiarity_rank, artist_hotttness_rank, song_currency_rank and song_hotttnesss_rank.

Using these three metadata files, I clubbed them to make a single dataset having as many features as possible as input features to the classification models and try to predict the genre of the music tracks.

3. Methodology

Firstly, the data is collected from the fma-github. The data contains the metadata of tracks, genres, and data collected through echonest and other track features. The very first step after the data collection is to perform preprocessing. The track dataset contains information about song title, album, artist, and per-track genres; User data such as per-track/album/artist favorites, play counts; free-form text such as per-track/album/artist tags, album description and artist biography. The header and row index had to be modified to have the correct column names and track-ids respectively. Then, the genre dataset has been preprocessed by dropping irrelevant columns for the analysis. Columns such as genre-title and number of tracks in each genre are retained. Finally, the echonest dataset is loaded and preprocessed. Just like track dataset, the columns and row index of the echonest dataframe are modified.

In the final dataset created, I've retained echonest samples which have entries in the track and genre datasets, by making use of track dataset and genre dataset. I've converted the non-numeric based columns into numeric based columns instead of dropping as they may contain valuable information. I've also dropped rows which have missing values. Due to this dropping of data having missing values, the total types of genres present in the final dataset are 7 and the final dataset has 247 dimensions.

Once the preprocessing is completed and the dataset is ready, I've implemented feature selection techniques. Selection techniques like wrapper-based techniques, which uses a criterion function to select the most important features. Wrapper based methods such as Sequential forward selection and sequential floating forward selection have been implemented. I wanted to implement the Sequential backward selection and Sequential floating backward selection methods too but since the dataset has many features, the time taken for selecting the important features is quite huge. So, I've limited to implementation of only SFS and SFFS. I've also implemented a filter-based feature selection technique Variance Threshold, which doesn't take any criterion function to evaluate the most important features.

Next, I've tried performing dimensionality reduction techniques on the data set. I've implemented the techniques that were taught in class. The first reduction technique I've implemented is Principal Component Analysis, which primarily focuses on capturing the maximum variation of the dataset. The next reduction technique I've implemented is Linear Discriminant Analysis, which primarily focuses on capturing the feature subspace to maximize the separability of the classes. The main drawback of the PCA technique is it doesn't take the class information into account. As a part of this project, I compared the performance of the classification models that are trained and tested using these reduced PCA and LDA data.

Once the feature reduction is performed, the reduced data generated by PCA and LDA are stored to a data frame. The PCA and LDA data has been used as input to perform the below classification models. The classification algorithms implemented to train the classification model are K-Nearest Neighbor, multi-layer perceptron classifier, Logistic Regression, Support Vector Machines and Bayes Classifier. The performance of these classification models are measured using various metrics such as average error rate, variance of error rate, and misclassification errors using confusion matrix. Figure [1] shows the flow of the project.

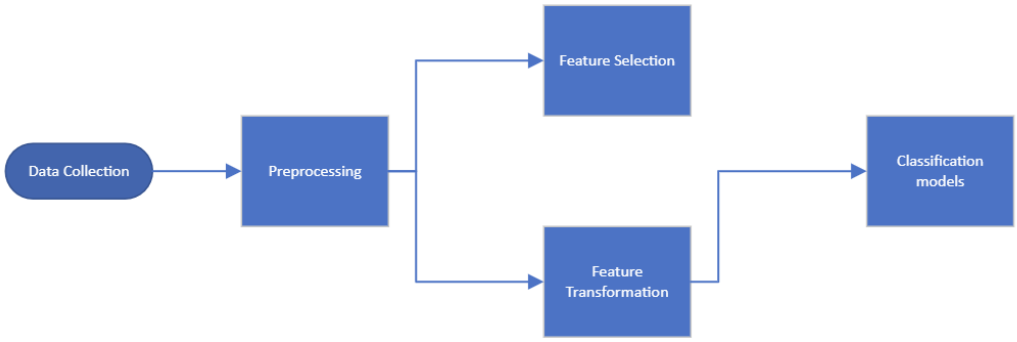

Figure [1]: Project Flow

4. Results

The techniques that were implemented on the preprocessed data are the feature selection techniques - Sequential Forward Selection (SFS) and Sequential Floating Forward selection (SFFS). Both the methods used a KNN classifier as the criterion function. The number of neighbors for the KNN classifier are set to 3. The input dataset consists of 247 features. The number of features to be selected are set to 3. The criterion function uses accuracy as the metric to select the most important features. Therefore, both SFS and SFFS methods selected top 3 important features in the following order – temporal_features-029, temporal_features-159, temporal_features-201. The below figure [2] shows the features selected in each iteration along with their accuracy scores.

```
Features selected using SFS for k = 3 are
{1: {'feature_idx': (182,), 'cv_scores': array([0.34146661]), 'avg_score': 0.34146661419
38869, 'feature_names': ('temporal_features-159',)}, 2: {'feature_idx': (52, 182), 'cv_s
cores': array([0.34369671]), 'avg_score': 0.34369670733307095, 'feature_names': ('tempor
al_features-029', 'temporal_features-159')}, 3: {'feature_idx': (52, 182, 224), 'cv_scor
es': array([0.34487734]), 'avg_score': 0.3448773448773449, 'feature_names': ('temporal_f
eatures-029', 'temporal_features-159', 'temporal_features-201')}}
Features selected using SFFS for k = 3 are
{1: {'feature_idx': (182,), 'cv_scores': array([0.34146661]), 'avg_score': 0.34146661419
38869, 'feature_names': ('temporal_features-159',)}, 2: {'feature_idx': (52, 182), 'cv_s
cores': array([0.34369671]), 'avg_score': 0.34369670733307095, 'feature_names': ('tempor
al_features-029', 'temporal_features-159')}, 3: {'feature_idx': (52, 182, 224), 'cv_scor
es': array([0.34487734]), 'avg_score': 0.3448773448773449, 'feature_names': ('temporal_f
eatures-029', 'temporal_features-159', 'temporal_features-201')}}
```

Figure [2] SFS and SFFS feature selection

The other feature selection technique implemented was Variance Threshold. This method removes all the features of the dataset whose variance is below a specified threshold. Figure [3] illustrates the number of features selected when the variance threshold has been varied between 0 to 100.

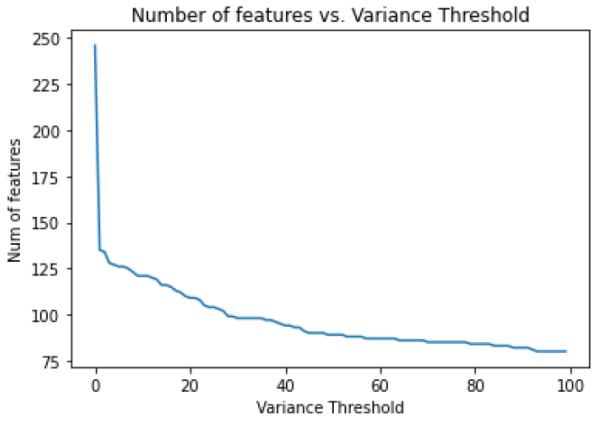

Figure [3] : Number of features vs Variance Threshold

The next techniques applied on the dataset are the feature transformation techniques. The PCA technique has been applied to reduce the data to 3-dimensions. Similarly, the LDA technique has been applied to reduce the data to 3-dimensions. Figure[4][a] illustrates the 3-d plot of data reduced using PCA. Figure[4][b] illustrates the 3-d plot of data reduced using LDA.

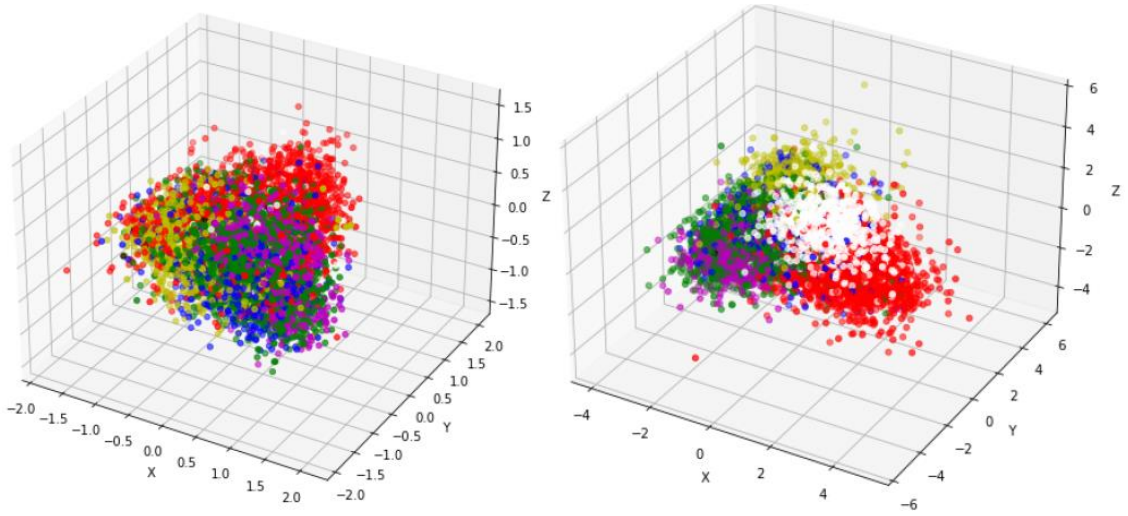

Figure [4] : [a] PCA 3-d plot, [b] LDA 3-d plot

After applying the feature transformation techniques, reduced data of both PCA and LDA has been used to perform the genre classification. The performance of the classification models are as follows. When PCA reduced data has been used for K-NN classifier, the model achieved minimum classification error when k=85. Figure [5] provides more insights about the performance of the model as value of k is varied.

```
Classification performed using PCA data
Minimum classification error occured when k = 85
Confusion Matrix when k = 85:
[[ 75  88  82  37   6   2   0]
 [ 37 256 111  41   4  20   0]
 [ 23  77 417  44  12   7   0]
 [ 35  37  34 104   1   0   0]
 [  9  21 144   2  16   0   0]
 [  3 104  22   2   0  24   0]
 [  1   1   1   5   1   0   0]]
Average Error rate: 54.783050810787415%
Variance of Error rate: 1.9369537013427987
```

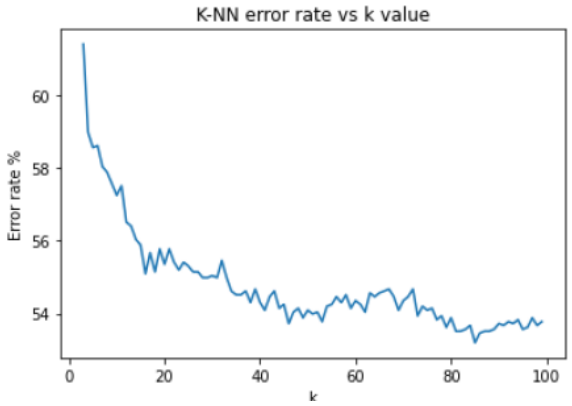

Figure [5]: K-NN using PCA reduced data

Similarly, when the LDA reduced data has been used, the K-NN model achieved minimum classification error when k=58. Figure [6] provides more insights about the performance of the model on LDA reduced data.

```
Classification performed using LDA data
Minimum classification error occured when k = 58
Confusion Matrix when k = 58:
[[ 27 120  84  43  13   3   0]
 [ 24 335  49  23  13  25   0]
 [ 14  26 505  11  23   1   0]
 [ 12  27  21 146   5   0   0]
 [  5   5  38   0 144   0   0]
 [  2 124   5   0   2  22   0]
 [  1   0   1   6   1   0   0]]
Average Error rate: 39.28018952629244%
Variance of Error rate: 2.334919798011917
```

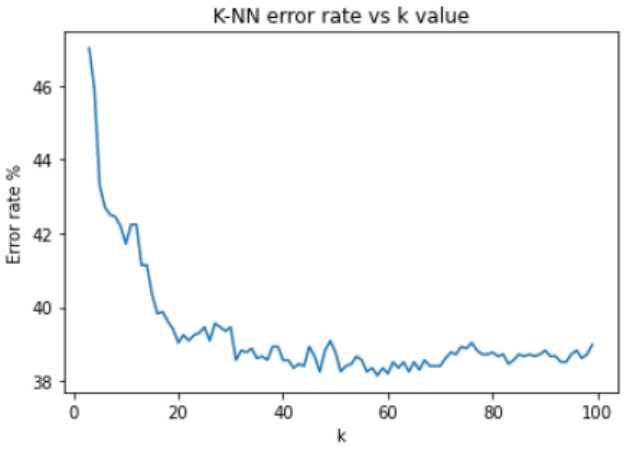

Figure [6]: K-NN using LDA reduced data

After K-NN model is tested, the next model implemented was Multi-Layer Perceptron. Figure [7] shows the performance of the model on PCA reduced data and LDA reduced data.

```
MLP Classifier using PCA data
Confusion Matrix:
[[ 34  39  37  22   2   1   0]
 [ 22 102  53  24   0  10   0]
 [  6  41 225  25   2   4   0]
 [ 16  13  18  60   2   0   0]
 [  4  12  81   0   0   0   0]
 [  2  55  17   2   0  16   0]
 [  1   2   1   1   0   0   0]]
Average error rate: 54.15048233354203%
Variance: 1.652903073406156
```

```
MLP Classifier using LDA data
Confusion Matrix:
[[ 15  57  48  23   7   1   0]
 [ 15 135  22  15   4   9   0]
 [  7  26 245  10  19   1   0]
 [  7  13  10  77   4   0   0]
 [  0   0  28   3  59   0   0]
 [  1  72   2   1   0  12   0]
 [  0   0   3   1   0   0   0]]
Average error rate: 39.406396871445324%
Variance: 3.9015807829532276
```

Figure[7]: Performance of MLP on reduced PCA and LDA data

Figure[8], provides insights of the performance of the logistic regression model on the reduced PCA and LDA data.

```
Logistic Regression using PCA data
Confusion Matrix:
[[ 28  51  42  26   0   1   0]
 [ 16 134  69  16   0  12   0]
 [ 10  36 209  12   0   5   0]
 [  8  24  15  47   0   1   0]
 [  6   8  95   0   1   1   0]
 [  1  41  18   0   0  11   0]
 [  0   4   2   2   0   0   0]]
Average error rate: 54.86409569074219%
Variance: 0.730000779283485
```

```
Logistic Regression using LDA data
Confusion Matrix:
[[ 11  60  42  13   6   0   0]
 [ 11 174  21  14   4   7   0]
 [  6  13 262  16  10   1   0]
 [ 12  12  10  74   2   0   0]
 [  1   4  19   0  64   1   0]
 [  1  63   1   0   0  13   0]
 [  0   0   4   0   0   0   0]]
Average error rate: 39.64713377481108%
Variance: 1.3807663702259716
```

Figure[8]: Performance of Logistic regression on reduced PCA and LDA data

Figure [9], gives the performance insights of the SVM model trained and tested on both PCA and LDA reduced data.

```
SVM Classifier using PCA data
Confusion Matrix:
[[ 30  43  42  21   0   0   0]
 [ 18 123  71  16   0   0   0]
 [ 12  41 246  19   2   0   0]
 [ 16  24  17  52   0   0   0]
 [  4  11  69   0   4   0   0]
 [  2  52  10   1   0   4   0]
 [  0   1   0   1   0   0   0]]
Average error rate: 53.9298830760006%
Variance: 2.868827472362577
```

```
SVM Classifier using LDA data
Confusion Matrix:
[[  7  71  44  22   4   0   0]
 [  4 173  20   9   3   0   0]
 [  4  26 267   9  18   0   0]
 [  8  17  12  63   1   0   0]
 [  0   8  25   1  54   0   0]
 [  0  70   4   2   1   0   0]
 [  0   0   3   2   0   0   0]]
Average error rate: 39.542576736885735%
Variance: 1.2627918338425101
```

Figure[9]: Performance of SVM on reduced PCA and LDA data

Figure [10], shows the performance of the Bayes classifier, which was implemented from scratch.

Predicted / Actual	Pop	Rock	Electronic	Folk	Instrumental	Hip-Hop	Punk
Pop	694	14	2	56	24	102	20
Rock	704	38	14	48	30	38	92
Electronic	318	32	32	78	32	324	88
Folk	288	6	0	78	16	8	0
Instrumental	16	6	0	6	0	12	0
Hip-Hop	116	0	0	0	4	134	10
Punk	246	30	2	0	0	6	46

Predicted / Actual	Pop	Rock	Electronic	Folk	Instrumental	Hip-Hop	Punk
Pop	418	103	68	115	66	77	65
Rock	418	167	20	90	31	33	205
Electronic	203	27	327	51	59	190	47
Folk	169	24	3	135	54	6	5
Instrumental	10	4	9	3	6	7	1
Hip-Hop	64	3	18	1	3	170	5
Punk	129	47	6	9	1	7	131

Figure[10]: Performance of Bayes Classifier on reduced PCA and LDA data

The table below consolidates the results of classifiers implemented for easy comparison

Classifier	Average Error Rate	Variance
KNN with PCA	54.78%	1.94
KNN with LDA	39.28%	2.33
MLP with PCA	54.15%	1.65
MLP with LDA	39.41%	3.90
Logistic Regression with PCA	54.86%	0.73
Logistic Regression with LDA	39.65%	1.38
SVM with PCA	53.93%	2.87
SVM with LDA	39.54%	1.26
Bayes Classifier with PCA	46%	-
Bayes Classifier with LDA	29%	-

Table [1]: Performance comparison of classifiers implemented

5. Analysis of Results

First, the feature selection techniques are applied on the preprocessed dataset, where rows with missing values are dropped. Thus, the total number of genres present in the final dataset are 7. Both the Sequential forward selection and Sequential floating forward selection methods selected the same set of important features in each iteration. The other feature selection that was tested on the dataset is the variance threshold technique. We can see from figure [3] that as the value of the threshold variance is increased, the number of features selected are decreasing. Basically, the features having less variance are discarded which has no or minimal impact when used in prediction models.

The feature transformation techniques have been used to reduce the data from high dimensions to lower dimension. From figure [4], we can infer that the 3-d plot of the data reduced by the LDA method looks more organized and classified when compared to 3-d plot of the data reduced by the PCA technique. This is because LDA is a supervised technique which considers class labels into account when reducing the data to lower dimension. However, the PCA is an unsupervised technique which doesn't consider the class labelled data. Once the data is reduced to lower dimension (3 dimensions in this case), the reduced data is used as input to classifiers to predict the genres.

The K-NN model performance was tested by varying the number of nearest neighbors in the range [3,100]. From figure [5] and figure [6], it can be inferred that data reduced using LDA transformation technique performed better in terms of average error rate when compared to the data reduced using the PCA technique. However, the variance of error rate is higher for LDA data when compared to PCA data. Also, it is observed that the minimum error of classification occurred when number of nearest neighbors is 58 where data reduced by LDA technique is used and the minimum error of classification is occurred when the number of nearest neighbors is 85, where data reduced by PCA technique is used.

The Kfold technique was used for Multilayer perceptron (MLP), Logistic Regression (LR) and Support vector machines (SVM) models to fine tune the hyper parameters of the models. The number of folds value is set to 10 and the data is shuffled. From the results of MLP, LR and SVM, it can be inferred that, like K-NN classifier model, the models performed better on LDA data compared to PCA data in terms of average of error rate. It is also observed that the average error and variance of error rate of the classifiers modelled using the PCA data are almost same for all the models. A similar inference holds true for the classifiers modelled using LDA data.

The final classification technique that was implemented was the Bayes classifier, which was implemented using the concepts learnt in course. The dataset is split into 80% train set and 20% for testing. The error of the classifier when trained and tested using the data reduced by PCA is around 46%. The error of the classifier when trained and tested using the data reduced by the LDA technique is around 29%. Compared to the model performance on PCA data, the model

performed better on LDA data. Figure [10] provides information about the classification performance of the Bayes model by using a confusion matrix. Majority of the data is predicted as Pop when PCA data is used. When the LDA data is used, the data is classified more correctly.

6. Summary and Conclusion

Three datasets have been used for this project. The datasets are preprocessed by reindexing the columns and rows and dropped the rows containing missing values. The final dataset has been generated by making use of the track and genre datasets to retain samples present in the echonest dataset. The final generated dataset contained 247 features. Feature selection techniques such as wrapper based like Sequential Forward Selection and Sequential Floating Forward Selection and filter based like variance threshold have been implemented. Both the SFS and SFFS techniques have selected same set of top 3 important features out of 247 features. Feature reduction techniques have been applied on the dataset to reduce the dimensionality. Techniques such as Principal Component Analysis and Linear Discriminant Analysis have been implemented. From their corresponding 3-d plots, the LDA technique created better low dimensional visualization compared to PCA technique. Various classification models have been implemented on these reduced datasets to compare the performance of the classifiers. Models performed better on data reduced by LDA technique than compared to data reduced by the PCA technique as LDA considers class labels while reducing the data. Out of all the models, the Bayes Classifier performed better on both the datasets. The performance of these classifier models can be improved by various methods such as by hyper tuning the parameters of the classifier models, increase the number of features into consideration(dimensions>3), etc.