Classify Song Genre from Audio Data

Aditya Panchal[1] Advait Thakkar[2]
[1,2]A. D. Patel Institute of Technology, India

Abstract— This system was focused on creating a classifying application based on music genre. The first and basic step to achieve this task is to have a data set with the clear differential genre based on which we will make the decisions, followed by normalization of data, which is performed by use of python libraries like Librosa and PyAudio. There are also built-in modules for some basic audio functionalities in these libraries or by performing component analysis which is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables on that data. Finally, compare the trained decision tree to logistic regression for classifying the genre of the given song.

Keywords: Music Genre Classification, Machine Learning, Pattern Classification, Feature Selection, Data Manipulation, Importing and Cleaning Data, NumPy, Pandas, Python3, Audio Signal Processing, Music, Signal Classification, Statistical Analysis, Feature Extraction, Information Retrieval

I. Introduction

In the last few years, streaming services with diverse and huge catalogs have become the primary means for people to listen to music that gratifies them. In spite of that fact, the colossal amount of music on offer can mean users might be a bit swamped when trying to look for new and distant music that fits in their fashion. MP3 is one of the most important and widespread formats over which a high volume of digital music is heard. Thereby, the researchers have shown great interest in developing music information retrieval techniques that would be beneficial for the listeners, musicologist and Internet music search engines to find a particular choice of music from a plethora of options. Automatic classification of music helps to bifurcate into categories such as mood, artist or genre and consequently these are widely studied the topic in music information retrieval since it is an effective procedure to categorize and structure the junk of music files available on the Internet.

Classification and distinguishing of music are done by characteristics shared by different styles of music which is described as a music genre. Rhythm, harmony, instrumentation, and melody of the music are certain characteristics that help bifurcating different genre of music [1]. Feature extraction and classification are the two main steps for the classification process of music based on genre. Initially audio signal information is obtained and then from the extracted features classify them to a different genre.

To categorize music to allow personalized recommendation we will use methods that will involve direct analysis of audio information in a given song and score the data onto different matrics. Our goal is to look through this dataset and classify songs into different genres based on their characteristics all without listening to a single one ourselves. To obtain the result we will clean our data, do some exploratory data visualization, use machine learning algorithms such as decision trees and logistic regression.

A. Dataset:

Data Set used here is a cluster of 1 million songs from different artists counted as 44,745 with different genre expertise and user-supplied tags for songs from the MusicBrainz website, comprising around 2,300 unique social tags and picked a set of 10 tags that seemed to represent musical genres with frequencies which follow a power law-like distribution. The musical genre is a notoriously subjective concept, but we tried to follow a genre set with a somewhat balanced distribution of songs per genre. For a few genres whose tags contained a lower number of songs for data, we added songs from a few possible tag names, which include minor spelling variations.

II. Workflow

A. Data Gathering:

To commence with, we will load the metadata of tracks and the metrics compiled by the Million Song Database [2]. Feature selection is one of the first and important steps while performing any machine learning task we have a dataset that contains musical feature of distinct tracks such as acoustics, energy, instrumentals, danceability, liveness, speeches and tempo on a scale of negative one to a positive one. We will obtain two files one that denotes tabular data and other with stores the result of queries. For further processing here we are going to use python's pandas [3] library which is a powerful Python data analysis toolkit. It provides quick and expressive data structures designed to make working with "relational" or "labeled" data instinctively easy. It serves as the fundamental block for performing real-world data analysis in Python. It's well suited for many different kinds of data as Tabular, Ordered/unordered, arbitrary matrix or any form of observational/statistical data sets. And as we have tabular data for our classification pandas is a top-notch choice.

B. Examine Relationships between Variables:

Avoidance of variables with strong correlation is necessary to avoid redundancy and hence the model will be kept simple and improve interpretability. Also with large dataset use of fewer variables can boost the computational speed. Packages such as pandas will help to find the correlated features in the data. If any strong correlation between any of the features is not found then the need to remove the feature from data is eliminated.

C. Normalization of Dataset:

Normalization of data serves as a useful procedure to simplify our models and use the least possible features to achieve unparalleled results. Since the features listed in the consideration are lacking any typical correlation we can make use of a common approach to minimize the number of features called principal component analysis (PCA). There is also a feasible option to explain the variance between genres by scanty features from a dataset. PCA rotates the data along the axis of highest variance, thus helping us assess the relative contribution of each feature of our data towards the variance between classes. Despite that PCA considers the absolute variance of a feature to rotate the data, a feature with a greater carbine of values will degrade the neutrality of the algorithm relative to the other features. To prevent this, we must first normalize our data. There are several methods to do this, but the general way is through standardization, such that all features have a mean = 0 and standard deviation = 1 (the resultant is a z-score).

D. Principal Component Analysis of Scaled Data:

After preprocessing the data, PCA is used to reduce the dimensionality of the data. scree-plots and cumulative explained ratio plots are used to find the number of components for the later part of the analysis. Sorted data of components against variance in decreasing order is plotted by Screen-plot and its even useful in differentiating which components are sufficient to segregate variance form the data, During this an ‘elbow’ (a sudden drop from one data point to the next) in the plot is generally used to decide on a proper cutoff. Python library NumPy [4] is used for this, It is the basal package for scientific computing. It contains things like powerful N-dimensional array object, broadcasting functions, and various tools integrating C/C++, Fortran code and even useful linear algebra, random number capabilities and Fourier transform.

Other than its basic use, it can also be used as a generic data container of multiple dimensions. It can even define arbitrary data-types. This allows NumPy with flawless and fast integration of a wide variety of databases.

E. Further Visualization of Principal Component Analysis:

Using this method, complications to find the number of intrinsic dimensions are greater and odds to achieve target becomes slim. Alternatively, we can use the cumulative explained variance plot for determination of the number of features required. After we successfully determine an apt number of components, we can perform Principal Component Analysis using the same components, which in turn helps in reducing the dimensionality of the data.

F. Train a Decision Tree to Classify the Genre:

Classification is a two-step method, namely the learning step and prediction step. The learning phase involves the development of a model based on the training data, whereas, in the prediction step, the model helps to predict the output for data. To achieve classification, we have used Decision Tree since it is one of the most popular and unchallenging classification algorithms. Also, this algorithm is capable of both classifications as well as regressing related problems.

G. Compare Logistic Regression to our Decision Tree Model:

We start with applying logistic regression which makes use of the logistic function. The logistic function calculates the odds of a data point belonging to a given class. After all the models are developed, we compare them on a few performance metrics, such as false positive (how many points are classified accurately) and false-negative rate.

H. Balancing the Data for Greater Performance:

We can obtain the value of accurate classification in each class inversely to the occurrence of data points for each class. A correct classification for a particular genre is not important rather a correct classification of another genre is more vital (and vice versa), we need to know the differences in the sample size of our data points when we weigh our class here. Also, the relative importance of each class is not significant here. Once we are completed with this we have balanced our dataset but during this procedure, we have removed a plethora of data points that might be pivotal to train our models. We need to test that the data improves model bias towards a particular genre classification while keeping the performance of overall classification.

I. Balancing the Dataset Improves Model Bias?:

Now the dataset is balanced, but during the procedure, a lot of data points are removed which might serve as an important portion in training our model. A test is performed to check if balancing the data enhance model bias towards the "Rock" classification while still pertaining overall classification performance at it’s best.

The size of the dataset is already reduced and no further dimensionality reduction is required. Practically, dimensionality reduction can be considered more rigorously when dealing with an extensive amount of datasets or in cases where computation time becomes inhibitory long.

J. Use of Cross-Validation to Evaluate the Models

Voila! Balancing the data removes the bias towards the more extensive class. To test the actual performance of the model, the cross-validation is applied. This step compares models in a more meticulously. As the data is split into train and test sets can impact model performance, CV ventures to cleave the data multiple ways and test the model on each of the portions. There are various CV methods with their pros and cons, but K-fold CV suits the requirements in this case. It separates the data into k different, commensurating subsets. Then, iteratively each subset is used as a test set while using the other subsets of the data as train sets. In the end, aggregation of the results of each fold is done for a final model performance score.

III. Conclusion

The overall analysis and classification of songs by the use of decision tree simplify process and filters the songs over genres in a more accurate way as it forces the consideration of all viable outcomes of a decision and traces every possible path to a conclusion. It creates a perfect analysis of each branch and identifies decision nodes that need further analysis. The major outcome of this can be used for song suggestions for users in online music players where they can be suggested songs based on the genre of the song they heard previously with that application.

References

[1] W. H. Tsai and D. F. Bao, “Clustering Music Recordings Based on Genres”, Journal of Information Science and Engineering, vol. 26, (2010).

[2] https://www.kaggle.com/c/mlp2016-7-msd-genre/overview

[3] https://pandas.pydata.org/pandas-docs/stable/pandas.pdf

[4] https://docs.scipy.org/doc/numpy-1.11.0/numpy-ref-1.11.0.pdf

[5] https://www.datacamp.com/projects/

[6] http://haralick.org/ML/CLASSIFICATION_OF_MUSICAL_GENRE_A_MACHINE_LEARNING_APPROACH.pdf

[7] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5990848/