Song Genre Classification

Devanshi Mittal[1], Jiaming Lu[2], Changchang Ding[3]

Abstract
The goal of this project is to be able to classify the genre of a song. According to Spotify Statistics taken in December 2018, over 20,000 tracks are uploaded to Spotify alone. That's 1 million tracks every 6 weeks on a single streaming service. Hence, it becomes imperative to predict the Genre of a song. The genre of the song is one in many features that is used in building recommendation systems. A person who has heard a song of a particular genre is more likely to listen to another song of the same genre. This project aims to provide the genre by studying the relation of different features like dance-ability and acoustic-ness.
Through this project, we want to also understand the most optimal machine learning algorithms that can help us predict the genre of the song with the given dataset of features.

Keywords
Genre — Classify — Machine Learning Algorithms

[1] *Data Science, School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*
[2] *Statistics, School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*
[3] *Computer Science, School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*

Contents

Introduction

Spotify, one of the leading streaming services, has a large catalog of music (35+ million songs) and the best playlist recommendations among other features that make it great for most music listeners. Spotify, using its music-intelligence division, the Echo Nest, analyses music based on its digital signatures for a number of factors, including tempo, acoustic-ness, energy, danceability, strength of the beat and emotional tone. With the large number of features, a song has, and the large catalog of music, it has become necessary to do choose the pertinent features to predict the genre. [1]

Genre classification is important because it is one of the many features that are required to build a good recommendation system.

1. Background

Over the past few years, streaming services with huge catalogs have become the primary means through which most people listen to their favorite music. But at the same time, the sheer amount of music on offer can mean users might be a bit overwhelmed when trying to look for newer music that suits their tastes.

For this reason, streaming services have looked into means of categorizing music to allow for personalized recommendations. One method involves direct analysis of the raw audio information in a given song, scoring the raw data on a variety of metrics. On the other hand, Spotify data alchemist also tried to use a complex machine learning algorithm to evaluate subjective psychoacoustic attributes of songs, analyze and categorize upwards of 60 million songs on a molecular level. There is a massive map of genres at Spotify umbrella genres like pop and country, as well as smaller niches like Thai hip-hop, German metal, and discofox. The site now lists more than 1700 genres and is constantly growing, with analysis of listening data over the years and every week to help identification of new musical ideas and directions.

MSD provides a few global song-level features such as title, artist, tempo, loudness as well as some segment features providing MFCC values which can be viewed as time series. More song-level audio statistics are provided on Spotify.

For the MFCC data, DNNs have recently become widely

used in audio analysis, especially deep convolutional neural networks (CNNs). For the million song dataset, Fully Convolutional Networks (FCNs) are used in [2] and convolutional recurrent neural networks (CRNNs) are used in [3].

For this project, our group will be examining data compiled by a research group known as The Echo Nest. Our goal is to look through this dataset and classify song genres - all without listening to a single one ourselves. In doing so, we will clean our data, do some exploratory data visualization, and use feature reduction towards the goal of feeding our data through some simple machine learning algorithms, such as decision trees and logistic regression.

2. Models and Methodology

2.1 Data

We have merged Top MAGD dataset[4] with the million song dataset[5] to tag all tracks with genres, resulting in 406427 songs classified into 13 top genres. Then we tried to fetch more song level audio statistics as new features (acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence and tempo) by searching the song on Spotify, resulting in 323854 songs successfully found.

Table 1. Genres Distribution

Genre	Number	Percentage
Pop Rock	193496	59.74%
Rap	15967	4.93%
Blues	5639	1.74%
RnB	10737	3.32%
Folk	5016	1.55%
Country	9763	3.01%
Jazz	14345	4.43%
New Age	3162	0.98%
Latin	15179	4.69%
Electronic	28355	8.76%
International	12008	3.71%
Vocal	4696	1.45%
Reggae	5491	1.70%

The dataset we built so far is imbalanced. We split the dataset into training and testing sets by 3 different ways: 1000 fixed training data for each genre, 2000 fixed for each genre and 80 to 20 split of training and testing for each genre, of which the first and second are balanced data, and the last one is imbalanced.

There are two levels of data for a song track, the song level data such as title, artist, acousticness, danceablility and so on, and also segments level data that are much about the detailed audio feature of each segments of the particular song track, including pitches, timbres (both have 12 features) and the level and position of the loudness of this segments. The number of segments also vary in different tracks. We just use the central 64 segments of each track and ignore the tracks that have less than 64 segments. A balanced dataset can also be built by 'duplicating' some tracks via shifting the selected segments.

2.2 Models

2.2.1 Random Forest Classifier

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and to control over-fitting.

To implement the model, we referred to: Random Forest Classifier

2.2.2 Logistic Regression

Logistic Regression is a technique used for prediction of categorical variables. Here we have a multi-class classification problem and with the features given we are trying to predict the song genre. For penalization we use an l2 method called newton-sg here and use weights since we have an imbalanced dataset.

To implement the model and tweak the hyperparameters we used:

Logistic Regression from sklearn

2.2.3 xgBoost Classifier

Boosting in general is a technique where the weak learners are trained and converted to strong learners. xgBoost is a very famous machine learning technique that uses boosted trees and very powerful in terms of computation power.

With the highly unbalanced data, the positive and negative weights were balanced with the scale-pos-weight parameter.

2.2.4 Neural Network

We tried 3 different neural network models: a simple 2 hidden-layer deep neural network, FCN suggested by [2] and CRNN suggested by [3].

For the 64 segments of each track, each segment contains 12 pitches, 12 timbres, max dB value, time of max dB and dB value of the start of this segment, resulting in 64×27 features.

The configurations of these models:

1. DNN: 1024 units for the first hidden layer and 512 for the second.

2. FCN consists of 4 convolutional layers and 4 max-pooling layers to reduce the size of feature maps from 64×27 to 1×1, then feeding to the output layers. Table 2 shows the detailed configuration.

3. CRNN uses a 2-layer RNN with gated recurrent units (GRU) to summarise temporal patterns on the top of two-dimensional 4-layer CNNs. [3] Table 3 shows the detailed configuration.

2.3 Resampling Techniques used to Handle Imbalanced Data

1. Weighted

Table 2. FCN Configuration

Layer	Output	Number of Params
Conv2D MaxPooling (2,1)	64× 27× 32 32× 27× 32	320
Conv2D MaxPooling (2,1)	32× 27× 64 16× 27× 64	18496
Conv2D MaxPooling (2,2)	16× 27× 128 8× 13× 128	73856
Conv2D MaxPooling (2,2)	8× 13× 256 4× 6× 256	295168
Conv2D MaxPooling (4,4)	4× 6× 512 1× 1× 512	1180160
SoftMax	13	13325

Table 3. CRNN Configuration

Layer	Output	Number of Params
Conv2D MaxPooling (2,2)	64 × 27 × 64 32 × 13 × 64	640
Conv2D MaxPooling (2,2)	32 × 13 × 128 16 × 6 × 128	73856
Conv2D MaxPooling (2,2)	16 × 6 × 256 8 × 3 × 256	295168
Conv2D MaxPooling (2,2)	8× 3× 512 4× 1× 512	1180160
GRU	512	1574400
GRU	512	1574400
SoftMax	13	6669

Sklearn has the parameter class-weight = 'balanced' which basically means applying class weights to loss functions.

2. Over Sampling

   The Over Sampling Technique soughts to add copies of instances from the under-represented class.

3. Under Sampling

   The Under Sampling Technique deletes instances from the over-represented classes. We performed under sampling by taking 2000 samples in each genre in the train set.

4. SMOTE: Synthetic Minority Over-sampling Technique

   SMOTE is an over-sampling method that works by creating synthetic samples from the minor class instead of creating copies. It selects similar instances using a distance measure perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances.

3. Experiments and Results

3.1 Dataset

Firstly, we collected the raw data from two main dataset: Million Song Dataset and MSD Allmusic Genre Dataset. They only provides track_id, segment-level feature and genre. Since we want more song-level features, we use the Spotify Web API to access its resource. In our python code, an HTTP request is sent and attached with the authorization. Then a JSON object is sent back with the audio information. Finally, we retain the song-level features and reconstruct the dataset such that each track_id has song-level features added in.

3.2 EDA and Feature Engineering

The correlation between the different features were assessed using the correlation matrix. The correlation matrix attempts to show how the variables are correlated. A correlation of greater than 0.7 means that the variables are correlated. We observed that there exists some correlation between features loudness and energy.

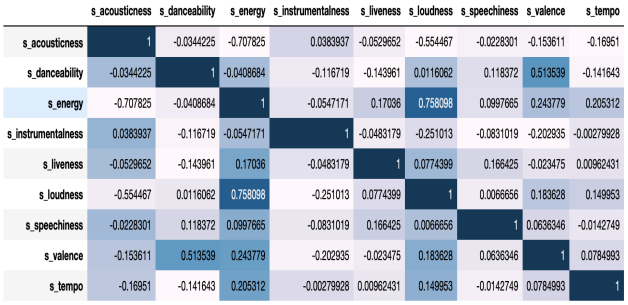

Figure 1. Correlation Matrix

Acousticness is highly right-skewed, danceability is mound-shaped and energy is left-skewed.

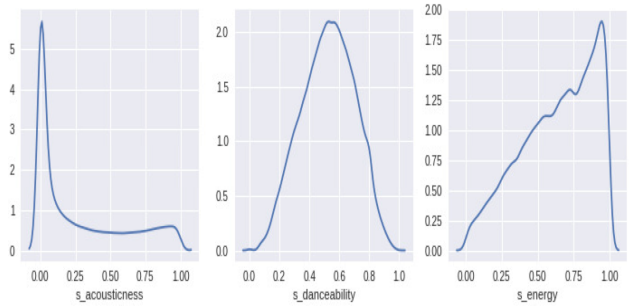

Figure 2. Density Plot

Instrumentalness and livenss are highly right-skewed, liveness also fluctuates around the tail. loudness is left-skewed.

Speechiness is extremely right-skewed and centered around one value, valence is mound-shaped and relatively symmetric, tempo is bimodal distributed.

To take into effect all variables and even the correlation, we performed Principal Component Analysis. It attempts to explain the variation in the data by creating a lesser number of components as compared to the true number of components.

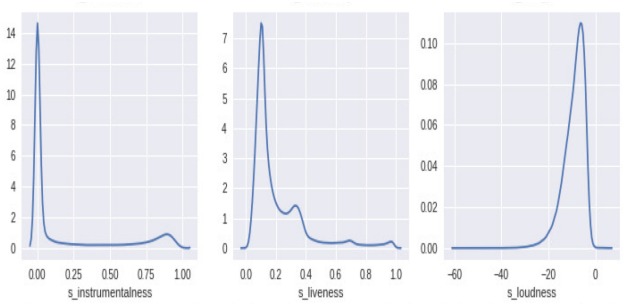

Figure 3. Density Plot

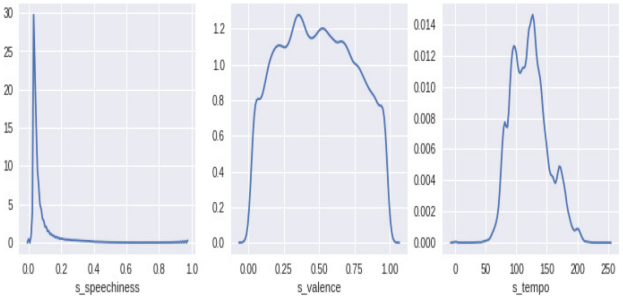

Figure 4. Density Plot

The following results were obtained on performing Principal Component Analysis on the 9 features of our data:

Figure 5 shows the variation explained by each feature:

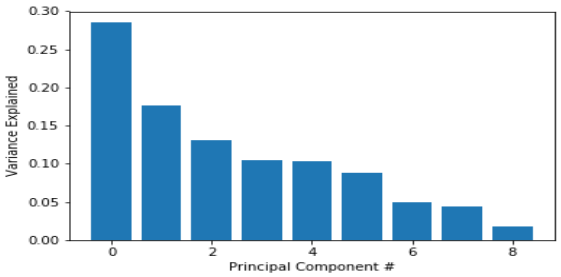

Figure 5. Variance Explained by Each Feature

After performing PCA on the scaled features, we observe that 90 percent of the variation can be explained by 6 components, which can be explained by Figure 6.

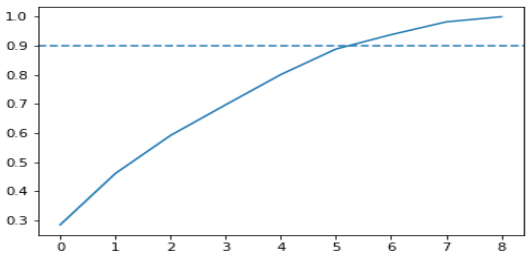

Figure 6. Optimum Number of Components

3.3 Models

Since the data is highly unbalanced and we are looking at multi-class classification that spans over 13 genres, the performance metric considered here is the F-1 score which is the weighted average of precision and recall.

3.3.1 Random Forest

Table 4. Random Forest F-1 Scores

PCA	Weighted	Under Sampling	Over Sampling
0.54	0.58	0.03	0.47

3.3.2 Logistic Regression

Table 5. Logistic Regression F-1 Scores

PCA	Weighted	Under Sampling	Over Sampling
0.44	0.47	0.38	0.47

3.3.3 xgBoost Classifier

Table 6. XGBoost F-1 Scores

PCA	Unweighted	Under Sampling	Over Sampling
0.55	0.59	0.53	0.47

Table 7. xgBoost with 80:20 train-test split - Best Accuracy

	precision	recall	f1-score	support
Blues	0.37	0.07	0.12	1128
Country	0.34	0.03	0.05	1953
Electronic	0.61	0.43	0.51	5671
Folk	0.33	0.02	0.04	1004
International	0.32	0.02	0.04	2402
Jazz	0.44	0.29	0.35	2869
Latin	0.41	0.11	0.17	3036
New Age	0.32	0.09	0.15	633
Pop Rock	0.69	0.93	0.79	38700
Rap	0.59	0.65	0.62	3194
Reggae	0.48	0.29	0.36	1099
RnB	0.35	0.04	0.08	2148
Vocal	0.35	0.08	0.13	940
micro avg	0.66	0.66	0.66	64777
macro avg	0.43	0.24	0.26	64777
weighted avg	0.59	0.66	0.59	64777

3.3.4 Neural Network

We use ReLU as the activation function for CNN and DNN layers and softmax for the output layer. We use adam algorithm for the optimizer. For the imbalanced data, we tried under/over-sampling and weighted loss function.

For under-sampling data, DNN, FCN and CRNN have almost the same performance, with weighted average precision around 0.7 and weighted average recall around 0.3. The

CRNN has the best weighted average precision and FCN has the best weighted average recall and the f1-score.

For over-sampling data, the results are similar, except for FCN which has the best f1-score of 0.53.

For imbalanced data with weighted loss function, DNN and FCN are getting much better results.

Over all, the best result is from FCN with weighted loss function and FCN is the best for each balancing method.

Table 8. Neural Network Performance

	precision	recall	f1-score
DNN under-sampling	0.68	0.22	0.28
FCN under-sampling	0.68	0.44	0.51
CRNN under-sampling	0.70	0.28	0.33
DNN over-sampling	0.63	0.28	0.34
FCN over-sampling	0.57	0.47	0.51
CRNN over-sampling	0.64	0.38	0.44
DNN weighted	0.52	0.63	0.54
FCN weighted	0.62	0.66	0.60
CRNN weighted	0.66	0.35	0.41

4. Summary and Conclusions

We tried different models to predict the genre of songs: random forest, logistic regression and xgBoost to work on the song level features and three different neural network models to work on segment level features. Among all these models, xgBoost and FCN perform the best. But the segment level features do not give significant benefit compared with the song level features.

For the imbalanced data, we tried three different method to deal with it: weighted loss function, under-sampling and over-sampling. Among all these models, weighted loss function performs the best and under-sampling performs the worst.

In the future, we would like to build a model that permits one song to have more than one label and combine the information from low-level, high-level and cultural features to get better results.

References

[1] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[2] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks, 2016.

[3] Keunwoo Choi, George Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification, 2016.

[4] Million song dataset benchmarks. http://www.ifs.tuwien.ac.at/mir/msd/. Accessed: 2019-03-01.

[5] Million song dataset. https://labrosa.ee.columbia.edu/millionsong/. Accessed: 2019-02-20.