Predicting Music Genres from Audio, Metadata and Text Features

Jie Ren*

Faculty of Engineering and IT, University of Melbourne, Melbourne, Australia

*Corresponding author: renxk@nwnu.edu.cn

Abstract. This paper focuses on identify music genres by using machine learning techniques and predictive models to categorize music without specific criteria based on audio, metadata, and text aspects.

Keywords: Naive Bayes, KNN, Decision Tree, Random Forest, Machine Learning Prediction.

1. Introduction

A key element of the music information retrieval system is the genre classification of music. It has gained importance due to the rise of digital music and the increasing number of people using it.[1]. We will go into greater detail about the experiments by displaying the experimental data as well as the images and tables that were generated throughout the process. Finally, we will present the experimental findings and the inferences that were made at the conclusion of this paper.

2. Related Work

Numerous factors have been considered in previous work on music genre classification. Such as, the music information retrieval system is used to automatically classify music types.[2] Some people use multiple auto regressive feature model to solve this problem in music genre classification.[3]

3. Feature Selection

3.1. Data set

The data set used for this paper was from a subset of the given data set [4][5]. From the data set we can observe that the data types of different attributes are different, whether nominal, ordinal, discrete or continuous. The structure details of several files are displayed in Table.1 below.

Table 1. Data Structure

Data	Features	
Training data set	Features	7678 rows × 156 columns
	Labels	7678 rows × 1 columns
Local test set	Features	450 rows × 156 columns
	Labels	450 rows × 1 columns
Test set	Features	428 rows × 156 columns

3.2. Data Pre-processing

For the purpose of improve the accuracy of classification results, we need to preprocess the original data. The original data representation included a large sum of attributes. These data include textual data, Nominal data, continuous data, and discrete data.

After viewing the data, the attribute "trackID" has nothing to do with music classification and does not affect the accuracy of the model. Because "trackID" has no practical effect on our data analysis, we delete this line of data.

The TF_IDF method is used for textual data in the dataset and the Normalization method for continuous data and discrete data in the dataset.

3.3. TF_IDF

Term Frequency (TF) is the number of times a word appears in a text. Generally speaking, a word appears many times in the article, so this word must have a great role, but if we practice ourselves, we will surely see that most TF you have counted out are words like this: "a", "an", "the" are similar to such words. Such words, without a doubt, do not aid our analysis or statistics. For the reason listed above, we need to remove useless word using some corpus of stopped words.

The value of a word in relation to its commonness is determined by the Inverse Document Frequency (IDF). If a word is rare, but it appears frequently in an article, then it is considered to be a key word. The importance of a word is computed by taking into account the frequency of its appearance.

(1) Calculate TF

N_w : the number of times a word appears in an article,N_t: the total number of words in the article

$$TF = \frac{N_w}{N_t} \tag{1}$$

(2) Calculate IDF

N_d: the total number of documents in the corpus,N_c: the number of documents containing the word

$$IDF = log\ \frac{N_d}{N_c+1} \tag{2}$$

(3) Calculate TF_IDF

$$TF_IDF = TF * IDF \tag{3}$$

TF-IDF algorithm is able to get quick and realistic results, but only using "word frequency" does not reflect the importance of a word very well. In addition, the word position information also affects the accuracy of the results, so the algorithm is not a very perfect solution to our problem of classifying music genres.

3.4. Normalization

Normalization of data scales the data so that it falls into a small specific range. Normalization removes unit restrictions from the data in the processing of comparative metrics so that metrics of different orders of magnitude can be weighted.

(4) Normalization formula

$$y = (-m\ in)\ /\ (-m\ in) \tag{4}$$

Normalization changes the number to a decimal between (0,1), so that the difference in magnitude of the data is not ignored or mistakenly processed.

4. Methodology

Since the problem to be studied is a classification problem, classification algorithms should be considered first in many machine learning models, such as Naive Bayes, KNN, Decision Tree and Random Forest. The accuracies of different models are listed as follows: for NB is 0.67, for KNN is 0.19, for Decision Tree is 0.41, for Random Forest is 0.45. After comparing the performance of these four models, we choose the best and worst models to focus on in this article. They are Naive Bayes and KNN models respectively.

4.1. Naive Bayes

The Naive Bayes method can solve the common problem in real life: given a conditional probability, how to get the probability of two events swapped, that is, how to get P(B|A)given P(A|B),P(A),P(B).

Calculate Naive Bayes

(5) $P(A|B)$: The probability of event A occurring, given that event B has occurred

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (5)$$

4.2. KNN

KNN is an instance-based learning where the function is only approximated locally and all computations are deferred until the function is evaluated [6].

The idea of the KNN algorithm is to train the model based on the data and known labels then compare the features of the test data with the features corresponding to the training focus and the highest frequency of occurrence in the classification corresponds to the category K.

5. Conclusions

5.1. Results

The main work of this paper is as follows:1. Use Pandas library to read the function. After reading the data, observe the characteristics and types of the data. 2. Conduct corresponding preprocessing for different types of data to facilitate subsequent analysis. Delete data that is not associated with the analysis. TF_IDF method is used to process text data so that text data can be analyzed scientifically. The Normalization method is used for continuous data and discrete data, according to their larger size differences. 3. Machine learning model was used to analyze the data and calculate accuracy. 4. Analyze on the Training data set and compare the accuracy according to the given labels data set. Further do the same analysis on the local Test set, and finally apply it on the test set to get the predicted result. 5. Collate the format of the data and output it as a.CSV file.

5.2. TF_IDF parameters

There are a lot of stop words that appear frequently but have no practical significance. Since they have no meaning, so we need to eliminate all the stop words.

There are also two parameters in TF_IDF. 'min' and 'max', respectively. 'min' means that if the occurrence of a word is less than N%, the word is not counted. Because the frequency of occurrence is too low, the randomness is too great, it's hard to judge. 'max' means that if the occurrence of a word is more than M%, the word is not counted. Because words that appear too often are not very important, they don't help us. The values of 'Max' and 'min' can be floating point or integer.

5.3. Different Naive Bayes model

From the inside of Bayes, this experiment calls Naive Bayes, BernoulliNB and GaussianNB models to test the data. The accuracies of different Bayes models are listed as follows: for NB is 0.67, for BernoulliNB is 0.57, for GaussianNB is 0.43. The BernoulliNB model requires data of type 0,1. The data in this dataset is mostly between 0 and 1, so BernoulliNB's model does not perform well on this data set. The GaussianNB model has a better effect on the data consistent with the Gaussian distribution. The data does not conform to the Gaussian distribution, so the GaussianNB model does not perform well on this data set. Figure 1 below shows whether the variable conforms to the Gaussian distribution.

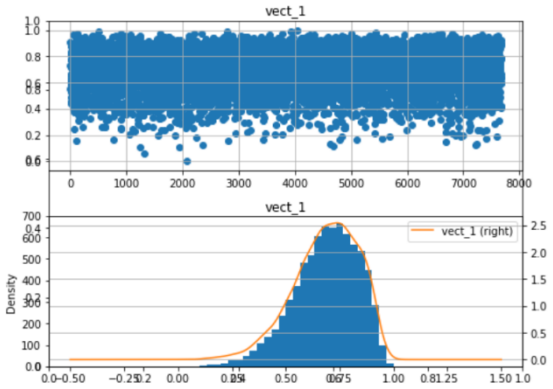


Figure 1. whether the variable conforms to the Gaussian

5.4. Analysis

For Naive Bayes, this model is simple because the Naive Bayes classifier is a simple probabilistic classifier applying bayes' theorem based on independent hypotheses. Bayesian models require data to be independent of each other, and it is difficult to make all attributes independent of each other in a data set. So Naive Bayes is not the best model either.

KNN has no visible pre-training phase, but once the program is launched and the data set is loaded into memory, it can begin to classify without training, hence its performance on the data set is relatively poor. The selection of K value in the program is likely to lead to poor performance of the program, which will have a great impact on the final result. An obvious shortcoming of KNN is that it is not suitable for processing data sets with large dimensions, so the final effect is not good.

5.5. Summary

To summarize, this article uses a machine learning prediction model to predict the type of music based on data such as audio, metadata and text features. From the experimental results, the overall accuracy of Naive Bayes is better than KNN. Also, it is very necessary to observe the data and make targeted pretreatment of the data, which can make our later analysis work more effective and convenient.

References

[1] Tao, L., Ogihara, M. and Qi, L., 2003. A Comparative Study on Content-Based Music Genre Classification. SIGIR Forum, pp.282-289.

[2] Tao, Li, and Mitsunori Ogihara. "Music Genre Classification With Taxonomy". Institute Of Electrical And Electronics Engineers Inc., 2020, pp. 197-200, Accessed 15 Oct 2005.

[3] Anders, Meng et al. "Temporal Feature Integration For Music Genre Classification". I E E E Transactions On Audio, Speech And Language Processing, 2007, Accessed 15 Oct 2020.

[4] T. Bertin-Mahieux, D. P.W. Ellis, B. Whitman, and P. Lamere. The million song dataset. In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), 2011.

[5] A. Schindler and A. Rauber. Capturing the temporal domain in Echonest Features for improved classification effectiveness. In Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval (AMR), 2012.

[6] Piryonesi S. Madeh; El-Diraby Tamer E. (2020-06-01). "Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems". Journal of Transportation Engineering, Part B: Pavements. 146 (2): 04020022. doi:10.1061/JPEODX.0000175.