Maschinelles Lernen für Physiker

The Genre Factor

Henry Krämerkämper
henry.kraemerkaemper@tu-dortmund.de

July 26, 2023

TU Dortmund – Fakultät Physik

Contents

1 Introduction **1**

2 The Utilized Dataset **1**

3 Implementation and Results of a Dense Neural Network **3**

4 Alternative Approaches to the Problem **3**

5 Discussion and Insights **3**

References **3**

1 Introduction

The task of classifying the genre of a song is common in the digital music industry. Most services offering music listening present some information about each song, which often includes the genre. Some services might even use the information to suggest other songs to listen to, which requires accurate information about the genre (or the genres) that a song belongs to. Retrieving this information is not easy, since there are no clear definitions of a genres attributes. Additionally, most songs do not belong to only one genre. The genre itself might change over time as well, which further complicates the problem. While the classification task might be technically solvable by humans, it remains a non-trivial endeavor due to its inherent complexity. Given the immense size of most music libraries, a manual approach to classification becomes highly impractical, necessitating alternative, more efficient solutions.

With these factors in mind, the task is evidently predisposed to a solution via a machine learning approach. As such, this strategy has become prevalent in addressing this problem, with a plethora of diverse methods having been explored to date (see, for example [1]). In this study, we attempt to classify music genres using a dense neural network. For this, we use a dataset sourced from the website Kaggle [2] containing songs and their attributes taken from the services YouTube [3] and Spotify [4]. We compare the neural network with two other, less sophisticated machine learning techniques, namely support vector machines [5] and the k-nearest-neighbours-approach [6], to establish a baseline. We aim to find out whether employing more complex and labour-intensive techniques result in an improvement in the face of the limited information contained in the dataset.

The report is structured as follows; first, the utilized dataset and the applied preprocessing is described in detail. Subsequently, the architecture of the dense neural network is laid out and the results are presented. These findings are then compared to the results of the alternative approaches. Finally, we draw a conclusion based on our analysis.

2 The Utilized Dataset

The dataset used in this project [7] contains 26 attributes about 18862 songs from 2079 unique artists. However, the genre of the song is not included in the dataset; we query Wikidata for the corresponding genre of each song, using the python package `pywikibot` [8]. An example entry of the resulting dataset is shown in table 1. Afterwards, the dataset is cleaned; missing or erroneous values in cardinal or ordinal features are replaced by a value calculated by a

k-nearest-neighbours-approach. We implement this by using the `SimpleImputer` from the `Scikit-Learn` package [9]. The architecture of our neural network does not feature text embedding, therefore attributes like `artist` or `album` are dropped.

Feature	Example	Type	Preprocessing strategy
Artist	Gorillaz	nominal	Use Label Encoder
Url_spotify	`https://open.spotify...`	nominal	Dropped
Track	Feel Good Inc.	nominal	Dropped
Album	Demon Days	nominal	Dropped
Album_type	album	nominal	Use Label Encoder
Uri	spotify:track:0d28khcov6AiegS...	nominal	Dropped
Danceability	0.818	cardinal	Scale and Transform
Energy	0.705	cardinal	Scale and Transform
Key	6.0	cardinal	Scale and Transform
Loudness	-6.679	cardinal	Scale and Transform
Speechiness	0.177	cardinal	Scale and Transform
Acousticness	0.00836	cardinal	Scale and Transform
Instrumentalness	0.00233	cardinal	Scale and Transform
Liveness	0.613	cardinal	Scale and Transform
Valence	0.772	cardinal	Scale and Transform
Tempo	138.559	cardinal	Scale and Transform
Duration_ms	222640.0	cardinal	Scale and Transform
Url_youtube	`https://www.youtube...`	nominal	Dropped
Title	Gorillaz - Feel Good Inc. (Official...	nominal	Dropped
Channel	Gorillaz	nominal	Dropped
Views	693555221.0	cardinal	Scale and Transform
Likes	6220896.0	cardinal	Scale and Transform
Comments	169907.0	cardinal	Scale and Transform
Description	Official HD Video for Gorillaz'...	nominal	Dropped
Licensed	True	nominal	Use Label Encoder
official_video	True	nominal	Use Label Encoder
Stream	1040234854.0	cardinal	Scale and Transform
Genre	Hip Hop	nominal	Target

Table 1: The attributes contained in the dataset, shown for an example song.

The cardinal features are the scaled to a range of $[-1, 1]$ and transformed to follow a normal distribution. The query results in ??? different genres. As these are higly specific, these categories are consolidated into 26 broader genres to achieve a more streamlined dataset with more samples per class. For example, the genre 'latin' contains 'salsa', 'bossa nova', 'samba' among others. Given the constraints of our datasets size, we only keep the top 6 genres with

the most songs to ensure a sufficient sample size. The remaining genres are hip hop, rock, pop, electronic, metal and classic. The datasets class imbalance can be seen in figure **??**.

3 Implementation and Results of a Dense Neural Network

4 Alternative Approaches to the Problem

5 Discussion and Insights

References

[1] Zhouyu Fu et al. "A survey of audio-based music classification and annotation". English. In: *IEEE Transactions on Multimedia* 13.2 (2011), pp. 303–319. ISSN: 1520-9210. DOI: 10.1109/TMM.2010.2098858.

[2] D. Sculley. *Kaggle: Level up with the largest AI and ML community*. URL: https://www.kaggle.com/ (visited on 07/26/2023).

[3] *YouTube*. URL: https://www.youtube.com/ (visited on 07/26/2023).

[4] *Spotify*. URL: https://open.spotify.com/ (visited on 07/26/2023).

[5] Corinna Cortes and Vladimir Naumovich Vapnik. "Support-Vector Networks". In: *Machine Learning* 20 (1995), pp. 273–297.

[6] T. Cover and P. Hart. "Nearest neighbor pattern classification". In: *IEEE Transactions on Information Theory* 13.1 (1967), pp. 21–27. DOI: 10.1109/TIT.1967.1053964.

[7] Salvatore Rastelli, Marco Guarisco, and Marco Sallustio. *Spotify and Youtube: Statistics for the Top 10 songs of various spotify artists and their youtube-video.* URL: https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube (visited on 07/26/2023).

[8] *Pywikibot: Python MediaWiki Bot Framework*. URL: https://pypi.org/project/pywikibot/ (visited on 07/26/2023).

[9] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.