Maschinelles Lernen für Physiker

The Genre Factor

Henry Krämerkämper
henry.kraemerkaemper@tu-dortmund.de

July 27, 2023

TU Dortmund – Fakultät Physik

Contents

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

2.1 Sourcing the Data

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 at this stage is shown in table 1. We do not keep all of these attributes; since the architecture of our neural network does not feature text embedding, the attributes `Track`, `Album`, `Title`, `Channel` and `Description` are dropped. The features `Uri` and `Url_youtube` are most likely random and do not contain useful information for our models to learn, therefore these are not used as well.

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

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

2.2 Preprocessing

Subsequently, the dataset is cleaned; missing or erroneous values in cardinal or ordinal features are substituted by a value derived by a k-nearest-neighbours-approach. We implement this by using the `SimpleImputer` from the `Scikit-Learn` package [9]. The cardinal features are the scaled to a range of $[-1, 1]$ and transformed to follow a normal distribution to improve numerical stability and the convergence speed as well as preventing the 'Exploding/Vaishing-Gradient' problem. The aforementioned wikidata query results in 397 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 1.

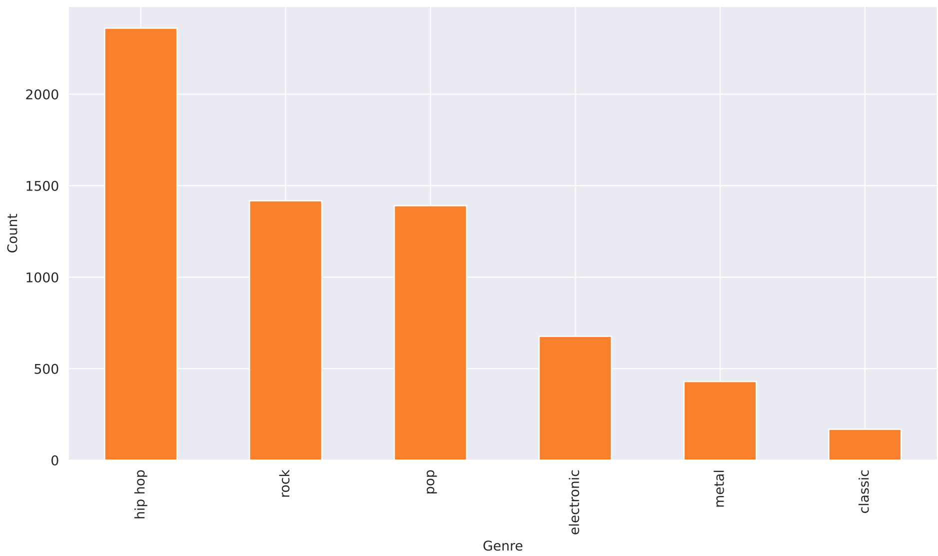

Figure 1: Class imbalance of the resulting dataset.

Upon application of the preprocessing steps, the dataset is reduced to 6446 entries.

3 The dense Neural Network Model

3.1 Architecture and Implementation

3.2 Performance and Results

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.