Predicting Music Genre with Supervised Learning

Using Audio Features and Lyrics

Music genre classification is a challenging, but popular theme within the discipline of music information retrieval (MIR; Bahuleyan, 2018). Music genres can be ambiguous and often overlapping labels. For example, a hard rock song can be simultaneously classified as rock and metal, not to mention other labels such as classic rock, proto-metal, and, obviously, hard rock. Similarly, the genre "classical" can encompass music from vastly different time periods, like Renaissance and the early 20th century. Nonetheless, music genre classification is frequently studied and implemented, particularly in the context of music recommendation on platforms such as Soundcloud and Spotify (Bahuleyan, 2018; Elbir, Çam, Iyican, Öztürk, & Aydin, 2018)

Despite these challenges, I decided to try my luck at writing a music genre classifier in Python for my Special Topics in Digital Humanities project.

Motivation

In the beginning of the course, I was not exactly sure what I wanted to make my project on. I definitely wanted to do something related to Machine Learning (ML), and preferably something different than text processing. I have had a key interest in ML and Artificial Intelligence for a long time, but I lacked the motivation and skills to gain practical knowledge. After almost two semesters of the Digital Humanities minor, I decided that my coding skills were finally enough to try and attempt an ML project.

After coming up with a few ideas that I ultimately dropped, I had the idea of a project related to music. While I would definitely call myself an enthusiastic consumer of music, I do not know more about music theory than the bare basics. Nonetheless, I read up about music classification and ended up finding many tutorials related to genre classification. I thought about writing a classifier for non-Western music (for example, I found a nice dataset of classical Persian music), but it proved to be too challenging. I decided that writing a classifier based on Western music would be a good start.

During my literature search, I found several projects that classified genre using musical or audio features such as mode (major vs minor), key (e.g. C or A), and loudness, while others looked at the lyrics. Since I found a data for both audio features and lyrics, I decided to try and write a classifier based on both.

Goals

In general, my goal for this project was to create a classifier that will predict music genre based on two kinds of predictor variables: (1) audio features, and (2) lyrics. In terms of learning outcomes, I mainly had the following three goals:

1. Gain practical skills and experience in classical machine learning using Python,
2. Gain more experience in data wrangling,
3. Gain theoretical knowledge in music classification.

Method

Dataset

Preliminary data wrangling. In order to conduct my analysis, I needed to find a dataset that contained all three aspects I was interested in: genre, audio features, and lyrics. In order to do that, I needed to combine several separate datasets courtesy of the Million Song Dataset project (Bertin-Mahieux, Ellis, Whitman, & Lamere, 2011).

My first step was to download the MSD genre dataset which divided songs into the following genres: classic pop and rock, folk, dance and electronica, jazz and blues, soul and reggae, punk, metal, classical, pop, hip-hop. Next, I downloaded a few datasets (also from the MSD website) pertaining to musiXmatch, a service that provides lyrics. The lyrics were given in a Bag-of-Words format.

I combined the datasets so that, in the end, I had a .csv file that contained both audio features and the 5000 most popular lyrics. The lyrics are contained in the following format: if song X contains the word "the" 20 times and the word "scrumptious" 0 times, then it will have the number 20 in the column "the", and 0 in the column "scrumptious". For more detailed explanation of my workflow, please refer to the comments in my Python files.

The audio features used are loudness, tempo, time signature, key, mode, duration, average and covariance of timbre vectors. Two categorical predictors, time signature and music key, were further converted into binary dummy variables. This resulted in 48 overall audio feature predictors.

Next, I inspected my data to see the distribution of genres (see table 1). I previously manually removed all songs classified as "classic pop and rock"; in my opinion, this genre was the least well-defined and contained examples of songs that were too distinct from each other. After this deletion, I still had quite a high number of songs (9091); unfortunately, my data was not uniformly distributed, which could have a negative effect on my prediction. I first removed around 2500 folk songs so that it could have a similar count to the metal, soul and reggae, and punk. Next, I removed the genres with the least songs: classical, hip-hop, jazz and blues, and dance and electronica.

genre	count
folk	3869
metal	1251
soul and reggae	1055
punk	1021
pop	776
dance and electronica	588
jazz and blues	387
hip-hop	92
classical	52

Table 1 – distribution of songs per genre before data cleaning.

Results of data cleaning. After my data cleaning and transformation, I ended up with a dataset that contains 5472 songs and 5048 features. The news genre distribution can be seen in figure 1. The data still is not perfectly balanced, but it definitely looks better than before.

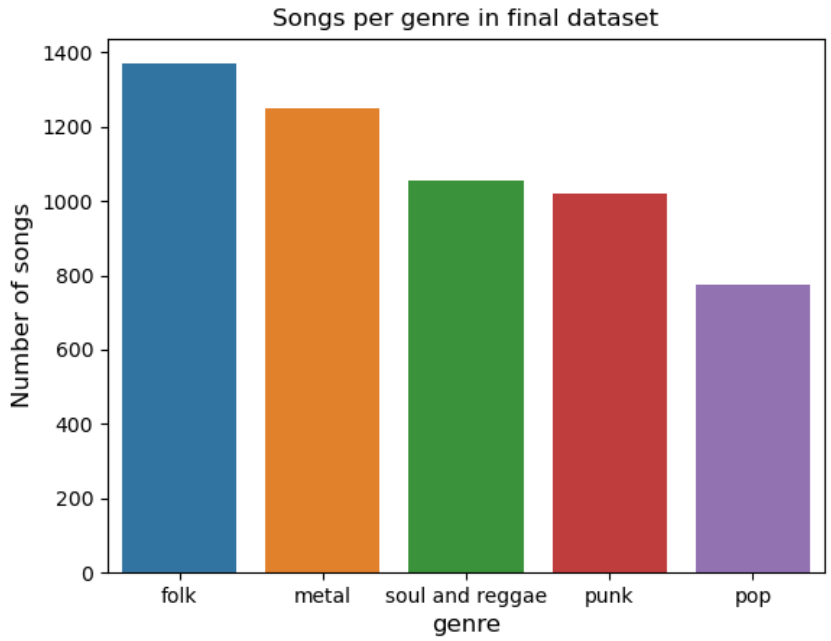

Figure 1 – Distribution of songs in the final dataset.

Results

For my classification, I used the Python library scikit-learn. This choice was motivated by the vast number of tutorials for this package, as well as its simplicity and accessibility. Before writing my code, I skimmed through a few introductory tutorials and examples of classification. Based on what I read, I decided to write a classifier based on three algorithms: Gaussian Naïve Bayes (GNB), Linear Support Vector Classification (LSVC), and K-Nearest Neighbours (KNN). All three are well-documented and suitable for multiclass classification.

Classification using all features

I ran the audio features and lyrics classification separately; however, in both cases, I ran the program on all predictors (48 for audio features, 5000 for lyrics). I initially scaled the values, but removed that part of the code after I realized it considerably slowed down my lyrics classification; for how this might have affected my results, please refer to the "discussion" section of this report. The data was separated into 80% train and 20% test.

Results for both audio features classification can be viewed in table 2. For both audio features and lyrics, the most accurate algorithm was Linear SVC, while K-Nearest Neighbours was the least accurate. Furthermore, in general, classification based on audio features was more accurate than lyrics classification.

	Audio Features	Lyrics
GNB	60,00%	58,99%
LSVC	71,60%	64,20%
KNN	47,03%	44,57%

Table 2 – accuracy scores for all three algorithms for both audio features and lyrics classification.

Classification using best features

Next, I ran code that classified genre based on only the best features. I first removed the features with the lowest variance, and then further reduced the predictors to only the best *n* features based on analysis of variance (ANOVA). For both audio and lyrics, I compared the accuracy scores for a few kinds of *n* (chosen based on the initial number of features), in order to find the approximate optimal number of features.

The results can be seen in table 3. For audio classification, the optimal number was 26 features. For lyrics, the optimal number was somewhere between 250 and 200 features. As in the previous classification, Linear SVC was generally the most accurate predictor, and K-Nearest Neighbours was the least accurate.

	Audio					Lyrics			
	26 features	25 features	20 features	15 features	10 features	284 features	250 features	200 features	100 features
GNB	57,99%	58,08%	56,71%	54,43%	53,42%	47,58%	50,05%	51,05%	50,68%
LSVC	70,41%	70,14%	68,49%	63,38%	59,82%	63,11%	63,65%	63,11%	61,55%
KNN	47,03%	46,94%	46,85%	45,57%	46,85%	44,20%	44,48%	44,11%	44,47%

Table 3 – accuracy scores for all three algorithms for both audio features and lyrics classification based on *n* best features.

Discussion

In the beginning of the Special Topics in Digital Humanities course, I only had a vague idea of what I wanted to do. All I knew was that I wanted to do something related to Machine Learning. Furthermore, I wanted to do something beyond text processing. I decided to create a multiclass music genre classifier using Python. My accuracy scores ranged from 40% to 70% based on the kind of features, number of features, and algorithm used.

I spent a lot of time on this project, but I believe there are still areas of improvement. For example, I am currently looking into finding out how accurately each genre was classified, or which exactly features were the best predictors.

Furthermore, I would love to do a similar project based on modern genres that are often more aesthetics-driven rather than based on distinct musical genres. Could we use the same techniques to accurately classify songs as vaporwave, witch house, avant-pop or future funk? This is definitely a question I would love to explore in the near future.

Did I meet my goals?

I think I definitely gained knowledge in data wrangling and classical ML. In this project, I learned how to combine datasets and extract relevant information, working with Pandas dataframes, clean data, scale values, run basic classification algorithms, and extract best features based on Chi Square, ANOVA F-tests, and variance. Obviously, I still have miles ahead of me if I ever want to do ML professionally, but I think I definitely learned some useful skills. Furthermore, I am now even more sure that I want to develop my skills in ML, and perhaps even try my luck at deep learning.

When it comes to the theoretical side of ML, I still definitely have a lot to learn. Even though I read up about the mechanisms behind the three algorithms I used, I definitely have to brush up on my maths skills. For example, I made the mistake of not scaling my features in the end, which might have had a negative effect on my accuracy (particularly for the k-NN algorithm).

However, I do not really think that I learned more about music information retrieval and genre classification. Even though I used these predictors in the final project, I have to admit that I do not really understand the theory behind the “avg_timbre” and “var_timbre” features. Most write-ups and papers I found that used the MSD either ignored those predictors altogether, or used terminology that was a bit too advanced for me (my knowledge of music theory is restricted to 5 years of piano and one introductory Music Cognition course). Music Information Retrieval is a complex, interdisciplinary field that requires advanced knowledge of maths, physics, and, of course, music theory. With my project, I only scratched the surface.

Conclusion

To conclude, for my Special Topics project, I wrote a classical ML classifier that predicts music genre based on audio features and lyrics. My results showed that Linear SVC was the most accurate algorithm; furthermore, the classification was generally more accurate when based on audio features rather than the frequency of lyrics.

I believe this project was an ideal conclusion to the Digital Humanities minor. Although coding was my main method of analysis, the data I worked with (namely music) was unequivocally linked to the arts and humanities. Furthermore, I had to be mindful to look for data that is open-source and reproducible. Finally, when developing my code, I worked with tutorials, examples, and blog entries written by other people involved in ML and music genre classification. This perfectly demonstrates the idea of collaborative scholarship, another key feature of DH research. I am definitely looking forward to developing my ML skills and working in other kinds of humanities-related projects: perhaps something to do with visual art, or maybe poetry?

References:

Bahuleyan, H. (2018, April 3). Music Genre Classification using Machine Learning Techniques. Retrieved June 30, 2020, from the ArXiv database.

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., & Lamere, P. (2011). The Million Song Dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011*. https://doi.org/10.7916/D8NZ8J07

Elbir, A., Çam, H. B., Iyican, M. E., Öztürk, B., & Aydin, N. (2018). Music Genre Classification and Recommendation by Using Machine Learning Techniques. In *Proceedings - 2018 Innovations in Intelligent Systems and Applications Conference, ASYU 2018*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ASYU.2018.8554016