Audio Feature Extraction and Music Genre Classification

ID: 5525549 Date: 06/01/2024

1. Introduction

Genre has been a frequently discussed term in the world of music. As artists begin their career, they may already have in mind which type of music they would like to make; record labels, managers or producers may prefer to focus on a specific genre and work with musicians with similar interests; most importantly, listeners may search for new music to listen to according to genres they have found themselves enjoying before, and streaming platforms may use genre for personalised recommendation.

Genres are distinguished by music style, and classifications can be arbitrary. Sometimes artists claim to belong to a certain genre themselves, sometimes respected critics and reviewers in the industry put labels on them, and the fan community have their views. Different parties may not reach a consensus on which genre an artist belong to, for example, the boundaries between Rock, Alternative and Indie could be quite obscure. Moreover, an artist's personal style may vary throughout their career: one typically pop musician may change their style to produce an entire rock-sounding album or collaborate with a rapper to end up with just one hip-hop track among ten other pop songs. Therefore, it is not always reliable to classify one track to a genre simply according to the label put on the artist or the album. Can we put aside our human knowledge and let the computer classify a track based on its audio features alone?

2. Dataset and Tools

Two datasets from Kaggle are used, *Spotify Tracks Dataset* and the *GTZAN Dataset*.

Spotify Tracks Dataset consists of a CSV file containing 114,000 entries, obtained with the Spotify API's *Get Track* and *Get Track's Audio Features*. Each entry has 20 fields, six are not relevant to this task: the track's unique Spotify ID, track name, artist (performer), album (the track belongs to), whether the track has explicit lyrics, and popularity (rated between 0-100 based on recent number of plays). What we care about are the 13 audio features and the genre a track belongs to (target classifier), The audio features are specific to Spotify and can be summarized from their Web API documentation as below:

1) **acousticness** [float]: level of confidence (0.0-1.0) that the track is acoustic.
2) **danceability** [float]: how suitable (0.0-1.0) a track is for dancing based on its musical elements like tempo, rhythm stability, beat strength, and overall regularity.
3) **duration_ms** [integer]: duration of the track represented in milliseconds.
4) **energy** [float]: measure of intensity and activity (0.0-1.0) calculated based on its dynamic range, perceived loudness, timbre, onset rate, and general entropy.
5) **instrumentalness** [float]: level of confidence (0.0-1.0) that the track is instrumental (contains no vocal content), values above 0.5 are intended to represent instrumental tracks.
6) **key** [integer]: the key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.

7) **liveness** [float]: level of confidence (0.0-1.0) that the track was recorded live based on whether an audience can be detected, values above 0.8 are considered to provide strong likelihood.
8) **loudness** [float]: the overall loudness of a track in decibels (dB), values are averaged across an entire track and typically range between -60 and 0 db.
9) **mode** [integer]: the modality (1 for major and 0 for minor) of a track.
10) **speechiness** [float]: level of the presence of spoken words in a track.
11) **tempo** [float]: the average tempo of a track in beats per minute (BPM).
12) **time_signature** [integer]: how many beats (3-7) there are in each bar (or measure).
13) **valence** [float]: the musical positiveness (0.0-1.0) conveyed by the track.

The dataset does not have any missing values. There are 114 genres in total, each with 1000 tracks, seems rather balanced, but there are duplicate track ids which is supposed to be unique. Despite that the dataset contains 114,000 entries, there are just 89,741 unique track ids, that is to say, 89,741 unique tracks. This is due to overlaps between genres, "Song 2" by Blur, for example, appears in the genres Rock, Alternative and… well, Alternative Rock! This is not the worst yet, a problem with such streaming platforms is that if a same song appears on different albums (compilations) it is given different track ids, though the audio file and its audio features should be the same across all occurrences. "The Foundations of Decay" by My Chemical Romance happen to appear with five separate track ids: the original single album, and all four other compilations, *Rock - Best of 2022*, *Rock Brandneu*, *Tek It - New Noise* and *Top of the Rock*.

The *GTZAN Dataset* is a collection of 10 genres each represented by 100 30-second audio files. The files do not have any information about the song name or artist (named in the format of genre plus a sequence of numbers), and the features generated with librosa are rawer and less comprehensible to human.

The tools involved include Weka (to apply popular classifying methods conveniently) and Python (using packages like NumPy, pandas, matplotlib and seaborn for data preprocessing, and plotting and extracting features from audio files using the open source music information retrieval library librosa.).

3. Preprocess

Before we start, we need to first preprocess the *Spotify Tracks Dataset*. All duplicated entries are removed based on the thirteen features. Although it is reasonable to categorize one song as various genres in reality, we do not wish to confuse the classifier, since it would be predicting only one label for each track. There are 67,530 tracks left, and their distribution among genres become highly unbalanced - numbers ranging from 66 to 992. We also drop the six irrelevant fields (id, name, artist, etc.), leaving only the features and the genre label. Finally, since the *GTZAN Dataset* has ten genres with 100 tracks each, we also randomly choose 100 samples from each of the ten genres, so that the results using these two different datasets can be somewhat comparable. Unfortunately, after removing the duplicates, only 71 reggae tracks remain in the *Spotify Tracks Dataset*, so we decide to include only the nine other genres in further steps.

4. Working with Spotify Audio Features

We look at each of the audio feature separately. First the numeric features: from the boxplot we can see that among nine features, duration, instrumentalness, liveness, loudness and speechiness all have quite a lot of outliers. This could be explained, that 75% of the tracks are shorter than 5 minutes, but the longest reaches an hour and a half (those above half an hour may not follow the one-song-per-track format). For instrumentalness, liveness and loudness, it seems that most tracks fall in the negation category, that is to say, many of the tracks are non-instrumental (contain spoken words), non-live (studio recorded), and not so quiet (in order to be heard). The rest may vary in terms of degree and level. The same with speechiness, as most tracks do have music content apart from human vocal. Acousticness, danceability, energy, tempo have very few outliers, suggesting more well-behaved and concentrated distributions.

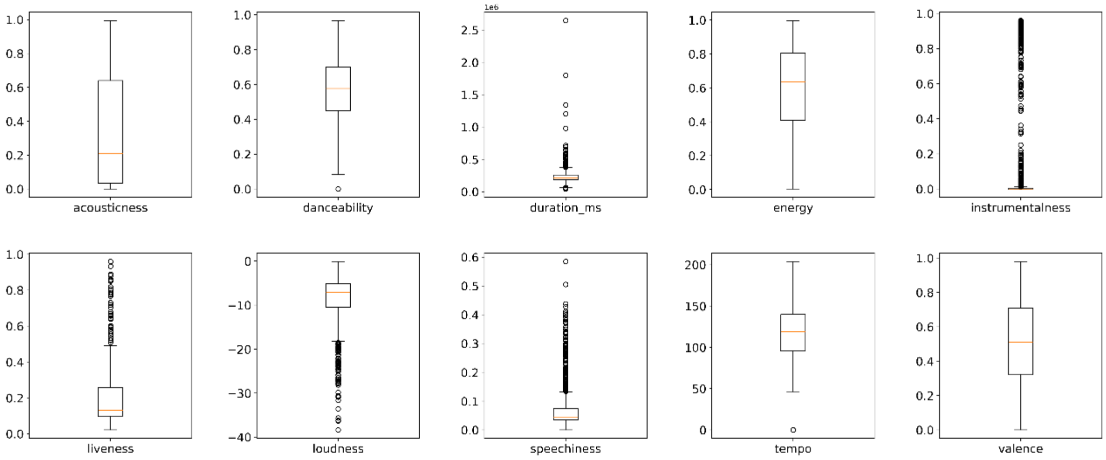

Fig 1. acousticness, danceability, duration_ms, energy, instrumentalness, liveness, loudness, speechiness, tempo, valence

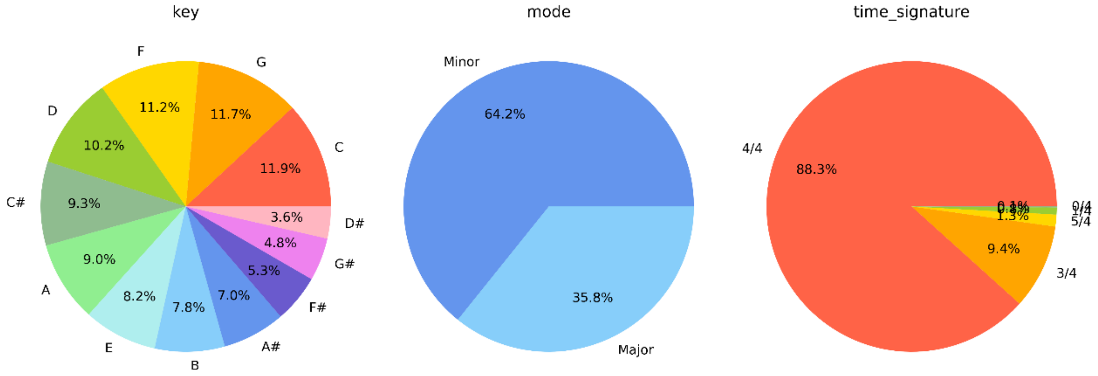

Fig 2. key, mode & time signature

Then we have the three non-numeric features. Keys C, G, F and D are most frequently used. Major keys are generally more popular compared to minor ones, with one exception, C#, coming right after the first four. The distribution of keys is relatively balanced with a gradual decrease in percentage, with D# being the least used key with only 3.6% tracks. Around two thirds of the tracks are derived from a major scale, and the rest follows the minor scale. As for time signature, it can be seen that most (up to 88.3%) of the tracks are composed using “4/4” time signature, followed by “3/4” that takes up 9.4%, while the rest go to “5/4” and “1/4”. Other cases, though theoretically possible, do not appear in this dataset.

Note that mode (scale pattern) and key are different concepts, in other words, if a track is composed in a major key, it does not necessarily mean its scale pattern must also be a major one. A Chi-Square Test for independence is performed on key and mode, and a significant relationship is found between the features with X^2(11, N=900) = 76.235, p = 7.851e-12. The Chi-Square test statistic is:

$$X^2 = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i} \quad (1)$$

Here O_i represents the observed value, while E_i represents the expected value, the degree of freedom is $(12-1) \times (2-1) = 11$ since we have twelve distinct keys and two modes in total, the sample size is 900, and the null hypothesis is the two features are not related. We have a p-value that is much smaller than 0.01, which allows us to say with confidence that they are related.

Similarly, we can tell that key and time_signature, mode and time_signature are likely to be independent from each other, since their calculated p-values are 0.439 and 0.377 respectively.

For the numeric features we simply calculate Pearson’s correlation coefficient between each pair:

$$corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E(XY)-E(X)E(Y)}{\sqrt{E(X^2)-E(X)^2} \cdot \sqrt{E(Y^2)-E(Y)^2}} \quad (2)$$

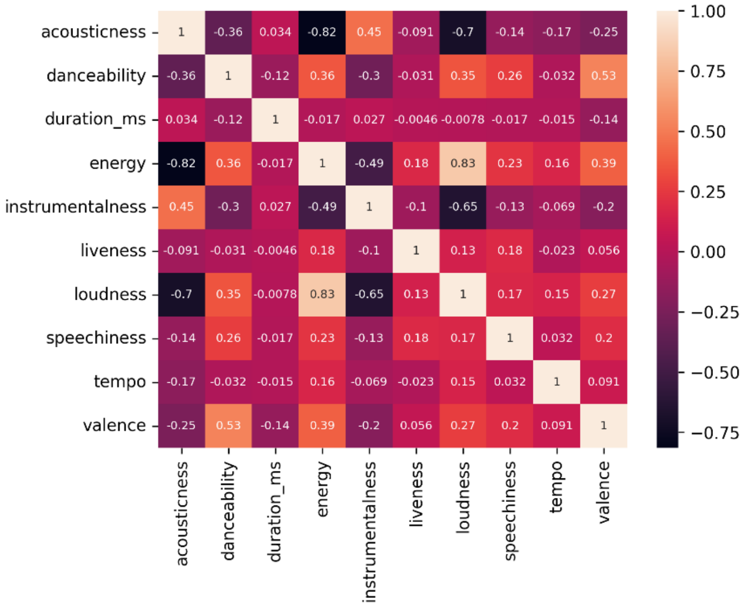

Fig 3. correlation matrix for all numeric features

The correlation matrix is drawn as a heatmap, from which we can infer that there is a very strong, positive correlation between energy and loudness. Meanwhile, strong negative correlation occurs between energy and acousticness. These correlations are quite intuitive, for noisy electronic music tend to sound energetic. We can propose other possible relationships: acoustic music is often not that loud. Instrumental music is normally not very loud or energetic; electric sounds are less likely to be pure instrumental. Energetic and loud music is suitable to dance to; music that conveys positive emotions tend to be louder, more energetic and "danceable"; sad music might be more common in acoustic and instrumental tracks, etc. Tempo and duration both have weak correlation with the other features, so the length of tracks and their speed/pace do not exhibit much difference in their distribution as other features vary.

We then come to the most vital problem: which of the features can/should be used for classification? We perform the Chi-Square Test between genre and key, mode, time_signature respectively, the p-values are 7.583e-5, 4.191e-8 and 3.271e-6, so all three features might be relevant to predicting the genre.

As for the numeric features, we draw boxplots divide by different genres. The example below shows the feature "energy", from which one may gain an intuitive view that classical music has much lower energy levels (the median being around 0.1) compared to metal (the median almost reaching 0.9), so energy can be useful in telling the two genres apart. Due to space limitations, other plots are not shown here, but we can similarly decide how useful other features are. The potentially least influential ones are duration_ms, liveness and speechiness, which we may consider not to include in the classifier.

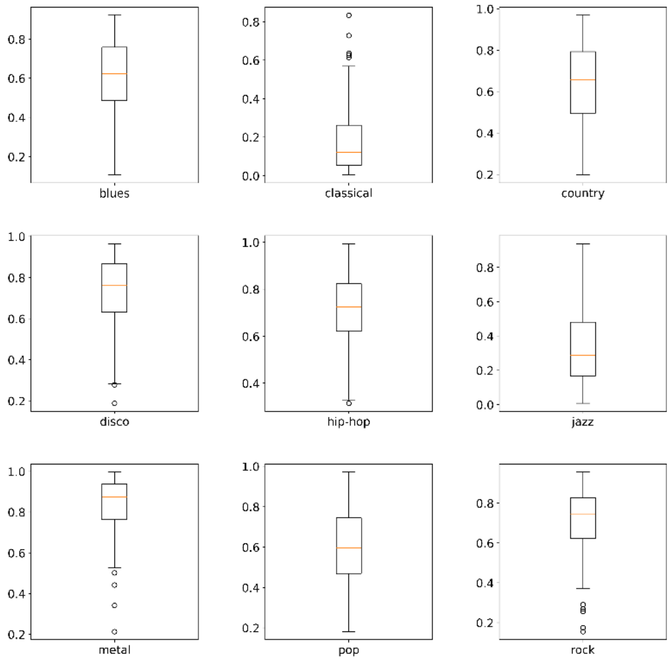

Fig 4. Feature "energy" plotted against nine genres

5. Classifying the genres

We convert mode and time_signature to nominal (they are represented as integers in original dataset) in Weka but leave the keys as numeric, for pitches can be compared and put in sequential order.

With 10-folds cross validation, the best accuracy we can achieve, unfortunately, is 47% using the logistic regression model, when duration, key, liveness and tempo are excluded from the input. With Naïve Bayes the best outcome is 43.667%, with SVM it is 43.778 %, for J48 Decision Tree 40.778%, and KNN yields its best result 38.222%, given N = 11. We look at the confusion matrix for the best model:

classified->	blues	classical	country	disco	hip-hop	jazz	metal	pop	rock
Blues	**14**	3	14	17	12	11	18	8	3
Classical	0	**82**	1	0	1	8	0	7	1
Country	7	1	**54**	8	6	5	8	7	4
Disco	6	0	13	**60**	8	5	2	1	5
hip-hop	6	0	6	10	**56**	3	3	13	3
Jazz	2	24	6	11	0	**52**	1	4	0
metal	3	0	11	3	2	0	**72**	2	7
pop	6	2	20	8	21	9	5	**24**	5
rock	3	0	21	20	10	8	25	4	**9**

Table 1. Confusion matrix

	blues	classical	country	disco	hip-hop	jazz	metal	pop	rock
TP	0.140	0.820	0.540	0.600	0.560	0.520	0.720	0.240	0.090
FP	0.041	0.038	0.115	0.096	0.075	0.061	0.078	0.058	0.035
Precision	0.298	0.732	0.370	0.438	0.483	0.515	0.537	0.343	0.243

Table 2. Detailed accuracy by genre

We see the model is the most confident identifying classical or metal music (relatively high truth positive rate compared to other genres). This may be the same to human ears, as they are both very characteristic and easy to identify upon hearing. However, the precision of metal is worse than classical, for the model often mistake rock music for metal. The model is extremely poor at telling how should blues, pop or rock sound, prediction for these types of music are rather randomly scattered among all genres.

6. Librosa Audio Feature Extraction

We then look at the *GTZAN Dataset*. Librosa's load() function takes an audio file as a floating-point time series, whose output can be used in feature extraction functions, producing a series of spectral and rhythm features which are quite obscure and much less intuitive to armatures, so we skip the explanation. Eleven of the features are used, and for outputs those returned in the form of one or two-dimensional arrays, we use the mean and variance to represent them, eventually we have 23 numeric features. File jazz.00054 is corrupted, and the genre "reggae" is removed to parallel with the previous dataset, so we have 899 entries left. The best accuracy is 66.0734%, also achieved using the logistic regression model.

classified->	blues	classical	country	disco	hip-hop	jazz	metal	pop	rock
Blues	**70**	1	4	5	2	7	3	0	8
Classical	0	**95**	1	0	0	3	0	0	1
Country	10	2	**52**	7	0	11	0	5	13
Disco	8	0	4	**56**	11	3	2	4	12
hip-hop	3	0	2	6	**67**	0	8	11	3
Jazz	6	9	10	2	1	**65**	0	4	2
metal	2	0	1	2	4	0	**84**	0	7
pop	0	1	7	3	9	3	0	**70**	7
rock	9	0	16	16	4	4	9	7	**35**

Table 3. Confusion matrix

	blues	classical	country	disco	hip-hop	jazz	metal	pop	rock
TP	0.700	0.950	0.520	0.560	0.670	0.657	0.840	0.700	0.350
FP	0.048	0.016	0.056	0.051	0.039	0.039	0.028	0.039	0.066
Precision	0.648	0.880	0.536	0.577	0.684	0.677	0.792	0.693	0.398

Table 4. Detailed accuracy by genre

We see this model is very good at identifying classical and metal music too, while being worst at deciding when a track belongs to rock. These characteristics are similar to the previous model, so we may conclude that some genres are just more difficult to identify than others in their nature.

7. Conclusion

We found that while Spotify music features are more interpretable to humans and can be used to discover some interesting patterns between characteristics of different genres, the features generated using Librosa are more effective when it comes to predicting the genre. One flaw is that despite the attempt to align the two datasets to contain the same number of tracks and the same distribution among nine genres, they are still not truly parallel, for we cannot get any other information from the *GTZAN Dataset*, like track name, artist and album. Otherwise, it would be a better approach to search for those songs directly using Spotify API and combine them together rather than trying to align it with a different dataset.

If we are to expand the task, there is the *Free Music Archive(FMA) Dataset* [1], which has similar format as *GTZAN* (30-second-long audio files grouped by genres) but is much larger (106,574 tracks in total).

As for methodology, the audio features generated by Librosa are often arrays and matrices, but since we cannot deal with multidimensional attributes in Weka we only use the mean and variance of them, and a lot information is lost during the process. More advanced Machine Learning or Deep Learning methods should make better use of such features, such as the ones propose by Shetty et al [3].

8. Reference

[1] Defferrard, M., Benzi, K., Vandergheynst, P., & Bresson, X. FMA: A Dataset for Music Analysis. In *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[2] McFee, B., Raffel, C., Liang D., Ellis, D., McVicar, M., Battenberg, E. & Nieto, O. Librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python in Science Conference*, 2015.

[3] C. Shetty, S. K. Debnath, Adithya, M. J. Falleiro, S. H and R. Srikanteswara. Advancing Music Genre Identification Through Deep Learning Techniques. *International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*. Erode, India. 2023.

[3] Spotify Web API Documentation. *https://developer.spotify.com/documentation/web-api*