2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 7-9, 2022

Playlist Classifier based on Song Features

Arunima Agarwal
Information & Communication Technology Department,
Manipal Institute of Technology, Manipal Academy of Higher Education,
Manipal, Karnataka, India
arunimaagl@gmail.com

Poornalatha G
Information & Communication Technology Department,
Manipal Institute of Technology, Manipal Academy of Higher Education,
Manipal, Karnataka, India
poornalatha.g@manipal.edu

***Abstract*—Playlist curation is a tedious task and with the rise of various online music streaming platforms, there has been a significant increase in the demand for automatic playlist curation. Existing playlist classifiers consider either user taste or genre-based song information, which excludes an individual song's audio features. This paper attempts to use each song's audio features to classify it into a playlist. The playlists chosen are both genre-based playlists (rock, pop etc.) and user defined abstract playlists (workout, summer etc.).**

We compare various data sources for extracting the audio features for each song in a handpicked subset of Spotify's million playlist dataset. These features are analyzed for their relevance to playlist classification and then pre-processed, i.e. cleaned, formatted, and organized, thereby making them ready-to-go for Machine Learning models. Various machine learning models like SVM, Decision Tree, Random Forest, XGBoost and KNN are implemented and compared. A simple dense neural network with 3 hidden layers is also implemented and used for comparison.

The implementation shows that trees based algorithms are most suitable for playlist classification problem. The highest median accuracy obtained was through implementing XGBoost algorithm and it was noted to be 93.9% with a standard deviation of 0.82%. Decision Tree and Random Forest algorithms were also observed to have similar accuracies.

Keywords—machine learning, KNN, SVM, Decision Tree, Random Forest, Dense Neural Network, classification

I. INTRODUCTION

Music listeners create playlists based on a variety of strategies and preferences. Playlists can consist of similar sounding songs, just one artist or even a mixture of multiple genres, era, and soundscapes. Playlist creation itself is motivated by several factors such as background music for studying, driving, introducing friends to new music etc. due to which they can be a set of random, seemingly unrelated music too.

Classifying music content is complicated as the categories are not often clearly distinguishable from each other. In this project we use a handpicked subset of Spotify's million playlist dataset [4] which spans genre based playlists, activity based playlists as well as an abstract playlist (summer). We compare various data sources for extracting audio features for each of the songs and finally use Spotify's API (Spotipy) which provides a range of song features and genre data which spans standard numerical audio features such as key, tempo etc., and categorical features like genre tags and artist names. It also includes extracted numerical features like danceability, speechiness etc.

These features are analysed for their relevance to playlist classification and then pre-processed, ie. cleaned, formatted and organized. New features are also added to reduce redundancy posed by highly correlated features. We use a variety and combination of machine learning and deep learning algorithms to find the most suitable algorithm for playlist classification.

The rest of the paper is organized as follows: the problem statement is given in section II, literature survey is discussed in section III, details of data collection is provided in section IV, pre-processing of data is given in section V, results are analysed in section VI followed by the conclusions in section VII and references at the end.

II. PROBLEM STATEMENT

Music industry is one of the ever growing industries in the entertainment sector, expanding with the availability of online music streaming. One of the most significant service provided by such platforms is the automatically generated playlists provided to the users. However, it is not a simple task as playlists usually overlap and classifying songs into playlists is not a direct choice even for humans. For instance, a rock song belongs to genre based overlapping subcategories like soft-rock, indie-rock etc. and can also be added to user taste based playlists as vague as 'favourites' or as specific as the artist name. Playlists can also be curated based on the activity such as 'study' or 'exercising' which have a more diverse yet seemingly random spread of songs.

Existing research for playlist classification were observed to primarily use a genre based dataset with audio features or spectrogram generated from 30second audio clips as training features. The former loses significance in scenarios where people create playlist aimed for an activity instead of using genre as the base. It is also important to note that a majority of songs change music after the first 30-40 seconds. Lyrics often start after this time interval. Using spectrogram generated from just 30second audio clip results in losing out a lot of this vital information.

Some papers were also found to use a set of user curated playlists along with user information like music taste and listening habits. However, these pose a problem when dealing with a new user with scarce information about their music preferences. Also, limiting the dataset to just user curated playlists discards genre based auto-generated playlists that the user might have 'liked'. This paper aims to utilize both kinds of playlists and focus on audio features as the classifying parameters to avoid skew against new user scenario.

2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT) | 978-1-6654-6855-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/GCAT55367.2022.9972120

There are indeed widely accepted rules in music theory that help human classification, such as chords and rhythmic structures, instrumental arrangement, etc. However, in general, musical content is complex and music genres are not well defined, making it a challenging machine learning problem.

This problem can be defined mathematically as – given a set, K, of unfinished playlists, and a set of unclassified songs, S, we aim to sort "s ϵ S" into the best playlist "k ϵ K", on the basis of audio features, f extracted for each s.

The audio features, f are analysed, cleaned and transformed from a set of audio and song features, F consisting of instrumentalness, speechiness, popularity, duration and other information about the track.

III. Literature Survey

In [1], the authors compare different SVMs with a basic deep neural network on a toy dataset of 13 playlists and find the neural network to have a significantly better accuracy. They considered 10 song features and a one hot vector for 2095 genre tags as the input. An accuracy of 82% with the neural network approach signals that building upon this with a larger number of song features and more intense neural networks might help achieve better results.

Convolutional RNNs on spectrograms derived from audio files, to curate playlists based on the genre as described in [2] interestingly combines two major neural network approaches in the model. The author's approach of using RNNs to extract music patterns like sequences, repetitions etc. directly from audio files and define the genre vector space based on the information gained is an interesting venture into automatizing genre classification. Comparing models trained upon pre- defined song features with such models can yield interesting results.

For audio feature extraction [3] has explored four mathematical models - one based on time domain while three are spectral. The paper further used SVMs, decision trees and nearest neighbour classifiers which are trained on a combination of the features extracted and the results are contrasted to find the most logical model. SVMs were found to outperform the other machine learning models.

The authors in [4] considered a mathematical approach to the problem of multiple genre classification by analyzing the audio files and determining genre borders in the track. The authors divided the track in musical segments and categorized into genres using a simple mathematical approach. Assigning multiple genres to a song helped to analyze modern songs better, and also assisted in multi-playlist classification. However, without a machine learning approach to detect genre borders, required significant refining.

In [5], the authors generated playlists based on song emotion, determined by lyric analysis and few Spotify song features. They considered four emotional categories - happy, sad, angry, and neutral - and compared a variety of models like SVM, decision tree and random forest, to achieve the highest average accuracy of 61%. The predictions could be improved using more refined NLP methods. However, Spotify provides a song feature, 'Valence' which represents the song connotation and emotion and can be used instead.

IV. Data Collection

The dataset consists of 12 handpicked playlists, each comprising of over 200 songs. The playlists are selected from Spotify's million playlist challenge dataset [4] and contain no information about the user or user taste. The playlists are selected ensuring diversity and maximum coverage of different styles. Each playlist is composed of number of tracks, albums, artists; total duration of playlist and a list of track details such as artist name, track name, album name, duration, and URI to track, artist and album data. The track URI can be used for extracting audio features, such as tempo, key, energy etc. provided by Spotipy.

Detailed explanation of the audio features provided by Spotipy is as follows -

- Danceability - It is mapped over a range of 0.0 to 1.0 from least danceable to extremely danceable song. The feature is derived from tempo, beat strength, and rhythm stability. Regularity of a song also affects its danceability.
- Energy – It is mapped over a range of 0.0 to 1.0, and signifies the intensity of a song. The faster and louder a song is, the higher would be its energy.
- Key - Integer values [0, 11] denote the estimated overall key of the track. E.g. 0 = C, 2 = D, and so on.
- Loudness - The average decibel value of the entire track. This is used to derive features like danceability, energy and valence.
- Mode - Indicates the type of scale from which the melodic content of a track is derived, i.e. the modality (major or minor) of a track. Major is represented by 1 and minor with a 0.
- Speechiness - This feature represents the presence of spoken words in a track. If there are more words in a track, its value would be higher.
- Acousticness - A confidence measure from 0.0 to 1.0 to indicate whether the track is acoustic or not.
- Instrumentalness - This is the opposite of speechiness feature and depicts whether a track contains no vocals while considering "ooh" and "aah" sounds as instrumental.
- Liveness - This feature shows the presence of background audience cheer/response in the track, predicting the probability that the track was performed live.
- Valence - This represents the connotation conveyed by a track. Higher valence is allotted to more positive (e.g. happy, cheerful, euphoric) tracks, while low valence tracks sound more negative (e.g. sad, depressed, angry).
- Tempo - The overall estimated tempo of a track is in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece, and derives directly from the average beat duration.
- Id, URI, Track HREF, Analysis URL - Unique identifier for the track

- Duration - The duration of the track in milliseconds.
- Time Signature - An estimate of number of beats per bar for overall track.

V. Data PreProcessing

The process flow of data pre-processing is shown in Fig. 1. Audio features for each track is combined with its popularity, album, artist, and playlist details. This compiled data is stored into a data frame, which is then analysed and pre-processed to get the data ready for the machine learning models. The first step taken is data cleaning to drop out irrelevant columns such as playlist collaboration, playlist modification timestamp, number of edits of playlist, playlist duration and number of tracks, artists and albums in the playlist. These properties are dynamic information about the playlist which might change on addition of a song to the playlist but have no effect on the classification process itself.

Next, we encode categorical data into integral values as machine learning models can only take numerical values as input. Since all the variables are numerical, we normalize them, except for loudness and duration, to make them easier to compare as well as process. The values for loudness are converted to a positive scale while duration is scaled to minutes before normalization.

Values for instrumentalness and speechiness are found to be skewed towards 0 (both have average below 0.1). The outliers for both are scaled down to provide a better spread of data and reduce the skew. Figure 2 shows the distribution of features after all data transformation is done.

After ensuring all variables are normalized, we plot the scatter plot matrix to analyse the correlation between variables and drop any highly correlated variables and/or define new ones as needed. In a scatter matrix, highly correlated variables have higher density along either diagonal.

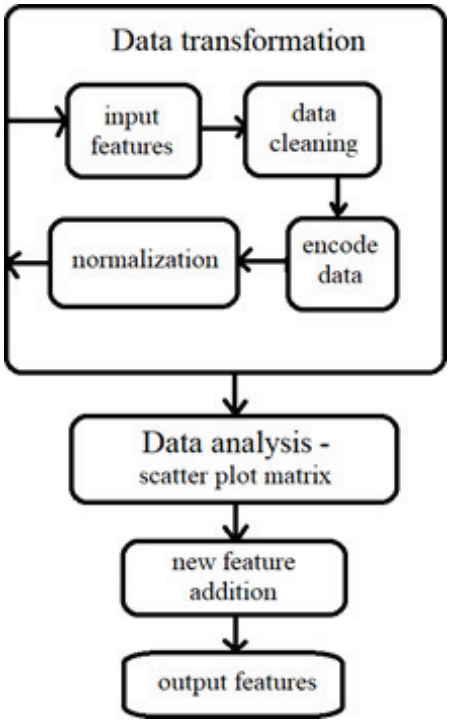

Fig. 1. Data pre-processing flow

	min	max	mean	std	sum
danceability	0.068200	0.986	0.586499	0.162642	1561.261200
energy	0.007370	0.998	0.706317	0.218164	1880.214970
loudness	0.000000	1.000	0.791217	0.115441	2106.219819
speechiness	0.045400	1.000	0.189155	0.194571	503.530600
acousticness	0.000001	0.994	0.199664	0.260623	531.504326
valence	0.038400	0.990	0.513704	0.238142	1367.480100
tempo	0.000000	1.000	0.475172	0.167184	1264.908086
liveness	0.020000	0.996	0.202055	0.161560	537.870300
instrumentalness	0.000000	1.000	0.350982	0.455180	934.315000
duration_ms	0.000000	1.000	0.373895	0.119872	995.307589

Fig. 2. Distribution of numerical features after data transformation

Figure 3 highlights such trends between danceability and valence, as well as loudness and energy, which show high positive correlation. Similar trend is also visible between loudness, energy and acousticness although with lower intensity. We also see danceability mildly correlated to acousticness and energy.

Highly correlated variables do not add new information to the model but rather slow down the processing. To overcome this, we define a new variable called interestingness as given in Equation 1.

$$\text{interestingness} = \text{loudness} + \text{tempo} + (\text{energy}*100) + (\text{danceability}*100) + (\text{acousticness}*100) \quad (1)$$

This feature describes how 'interesting' or 'not boring' a track is, by involving danceability, energy and other correlated features. Another set of highly correlated variables are the popularity indexes for the album, artist and track. We define a new variable popularity index which gives weight to track, artist and album in ratio 3:2:1. These weights have been determined keeping in mind that track popularity boosts up the album and artist popularity while album popularity contributes to artist popularity. These new variables are also normalized to maintain uniformity in data.

The dataset is then split into training and test sets with 2129 rows in the training set and 533 in the test set. A skewed training set will bias the model to predict a playlist more often than other and vice versa. Hence "train test split" function from sklearn module is used with appropriate hyper parameters to ensure that each playlist has equal weightage and neither of the sets are skewed.

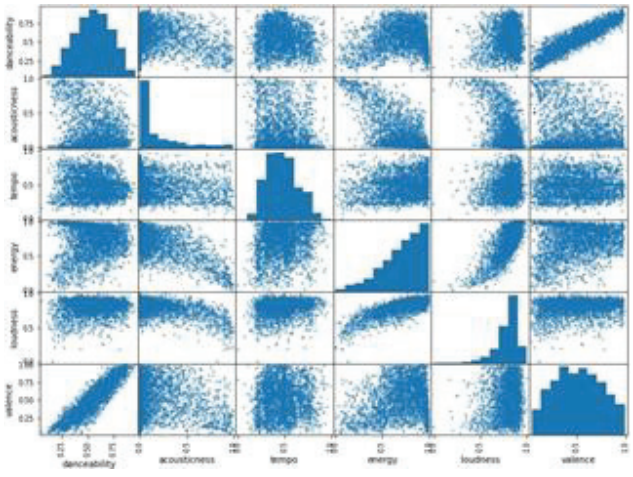

Fig. 3. Scatter plot of audio features

VI. Result analysis

Several machine learning models - Support Vector Classifier, Decision Tree, Random Forest, XGBoost, K-Nearest Neighbours - and a dense neural network model are applied to the dataset. The models are fine-tuned with a variety of parameters to determine the most suitable configuration for each and the results are noted as in Table I.

TABLE I. Model Accuracy

Model	Accuracy (%)
Support vector	89
Decision tree	94
Random forest	93
XGBoost	91
K-Nearest Neighbours	92
Dense Neural Network	86

It is evident that tree based algorithms are best suited for the problem statement. However even though these results seem conclusive, we apply K-fold validation to verify that the results are indeed consistent. K-fold is applied only for the Machine Learning algorithms thus excluding the neural network. As we can see in Table II, the models exhibit a low standard deviation and maintain a median accuracy near the values depicted in Table I.

Hence, we can conclude that the results are consistent throughout the dataset used and tree based algorithms gave the best results, with decision tree algorithm giving the highest accuracy of 94percent.

VII. Conclusion and Future Scope

The results obtained shows the complexity of this problem statement as well as how arbitrary human choices could be while compiling a playlist. This shows that while there are various avenues through which the prediction accuracy might be improved, these models used can already 'think' along the same lines as humans in terms of songs and audio features. It cannot be ignored that playlist classification task in itself does not have well defined boundaries. Playlists not only often overlap but are also highly subjected to bias of user opinions. Hence, classification accuracy of 94% should be considered to be acceptably precise.

TABLE II. MEDIAN AND STANDARD DEVIATION OF ACCURACY OF VARIOUS MODELS

Model	Median (%)	Standard Deviation (%)
Support vector	87.5	±1.5
Random forest	93.8	±1.3
XGBoost	93.6	±0.9
K-Nearest Neighbours	93.9	±0.82
Dense Neural Network	91.9	±1.3

It has also been observed how the same data set is treated differently by various algorithms and shows that specific models are appropriate for specific dataset. The six models have been compared using K-Fold cross validation and tree based algorithms have been found to show the best results for this problem statement. This implementation has been found to have a higher median accuracy and a smaller standard deviation than [1].

The dataset used in this project is twelve playlists, each with over 200 songs and no overlap among playlists, from Spotify's million playlist dataset [4]. For future work, a broader, more diverse set of playlists can be selected. The Classifier can be trained on smaller playlists and modified to curate playlists even from scratch. This can be utilized in a real time music player to generate infinite playlists and song suggestions in playlist recommendation and music player apps for new users who don't have a well-defined music taste profile yet.

Besides selecting audio features, song lyrics and instruments used can be included as important features as well. NLP techniques can be used on song lyrics to see if songs from same playlist have higher frequency using certain words. For instance, playlist "happy mood" will contain more words with positive connotation than a "sad" playlist. This paper lays foundation for all such future works relating to playlist classification as well as playlist generation.

References

[1] A. Awadelkarim and K. Coelho. (2018) Training a playlist curator based on user taste. [Online]. Available: http://cs229.stanford.edu/proj2018/report/22.pdf

[2] Adiyansjah, A. A. S. Gunawan, and D. Suhartono, Music recommender system based on genre using convolutional recurrent neural networks, Procedia Computer Science, vol. 157, pp. 99–109, 2019.

[3] A. Viloria, O. B. P. Lezama, and D. Cabrera, Segmentation process and spectral characteristics in the determination of musical genres, Procedia Computer Science, vol. 175, pp. 96–101, 2020.

[4] H. Nakamura, H. Huang and K. Kawagoe, "Detecting Musical Genre Borders for Multi-label Genre Classification," 2013 IEEE International Symposium on Multimedia, pp. 532-533, 2013.

[5] G. Subramaniam, J. Verma, N. Chandrasekhar, N. K. C. and K. George, "Generating Playlists on the Basis of Emotion," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 366-373, 2018.

[6] Spotify's million playlist dataset. [Online]. Available: https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge, 2020