



Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

Song recommendations using Spotify API

Shane Keane, Dylan Kierans, Alexander Sanfilippo
17329836, 21346261, 21343904

CS7CS4 Machine Learning Group Project

Contents

1	Dataset and Features [1 page]	2
1.1	Data Collection	2
1.2	Training Data	2
2	Methods [1 page]	3
2.1	Some baseline models	3
2.2	kNN	3
2.3	Logistic regression based models	3
2.4	MLP	3
3	Experiments/Results/Discussion [2-3 pages]	3
4	Summary [100-200 words]	3
5	Contributions	3
6	Github Link	3

Introduction [half page]

Streaming platforms are as popular as ever. While the film/TV and podcast streaming sector has seen an increase in competition in recent years, the same can not be said for music platforms. Spotify is the home of music streaming. We take a slight turn away from user-specific recommender systems which the platform is well known for, and instead look at automating the classification of songs into genre-specific playlists. This would be a useful tool for a growing resource such as Spotify. The music platform was founded in 2008, and now contains over 70 million songs. That is an average of about 14,000 songs per day since launch, with Spotify reporting earlier this year that they now 'publish' over 60,000 songs per year .

That's a lot of songs to tag, and its likely Spotify already have advanced deep learning software analysing every part of the platform and user interaction. We hope to imitate a possible genre classification model the multi-billion company may use.

1 Dataset and Features [1 page]

1.1 Data Collection

Spotify provide a free API for developers [<https://developer.spotify.com/documentation/web-api/>], which we use for data collection in order to compare songs . Spotify stores a whole range of metadata which is accessible via its API such as information about artists, users, podcasts and can be used to return search queries for 3rd party applications. For our project we are just interesting in playlist and song information. In particular, gathering the audio feature attributes of every song in specific playlists.

The project sets out to use these audio features as the input parameters (including feature engineering if necessary) to create an effective multi-class classification model. Features of interest include normalized parameters acousticness, danceability, energy, instrumentality, liveness, speechiness, valence. Other features which aren't normalized but may be of interest include continuous duration, tempo, and discrete key, mode, time signature

1.2 Training Data

We chose 8 genre-specific spotify playlists each with over 200 songs, to train our models. The genres we chose to work with are:

- Jazz
- Country/folk
- Classical
- Hip Hop
- Rock
- Kpop
- Heavy Metal

- Dance

Of course music genre isn't entirely objective, so it is expected that there will be noise in training data, and it won't be possible to create a 100% accurate model. In any case, we had a total of 2300 songs and were left with 2276 after removing duplicates. For each of these songs we gathered all audio features from the API. We then added an additional column to each dataset specifying which playlist number we had got the song from. We now had 2276 input vectors, and a set of 8 possible target values to train the model on.

In order to test models, we perform a 5-fold train/test data split.

Comment on not just evaluating model, but also evaluating how explanatory the audio features are

2 Methods [1 page]

Our goal was to create an accurate multi-class classification model for a finite number of genres. For the sake of simplicity, and in order to gain an understanding of the audio features from Spotify, we work on just taking the most likely classification from the model predictions. We understand that more complex measures of accuracy can be considered by taking a more dynamic approach to accepting multiple-classes within a certain threshold. In an industry application of such a model this would be a useful consideration.

However, for the purpose of this project we focused on running a wide variety of models in order to find the one which best captured the behaviour of the data.

List all our models from most basic – > most complex.

Comment on pipelining method for normalizing data

2.1 Some baseline models

2.2 kNN

2.3 Logistic regression based models

2.4 MLP

3 Experiments/Results/Discussion [2-3 pages]

4 Summary [100-200 words]

5 Contributions

6 Github Link