Music Genre Classification Engine

CS-412 - Spring 2020
Final Project

Aditya Sawant

Background and Motivation

- Traditional Machine Learning applications have always been around computer vision, text/language processing and other domains where the data can be handled and manipulated easily.

- Which is why, audio classification has not had as much attention as the other applications.

- This is also in part because understanding the domain is pretty tricky as one would have to develop knowhow in digital signal processing, audio engineering and in some cases maybe even acoustics and music theory.

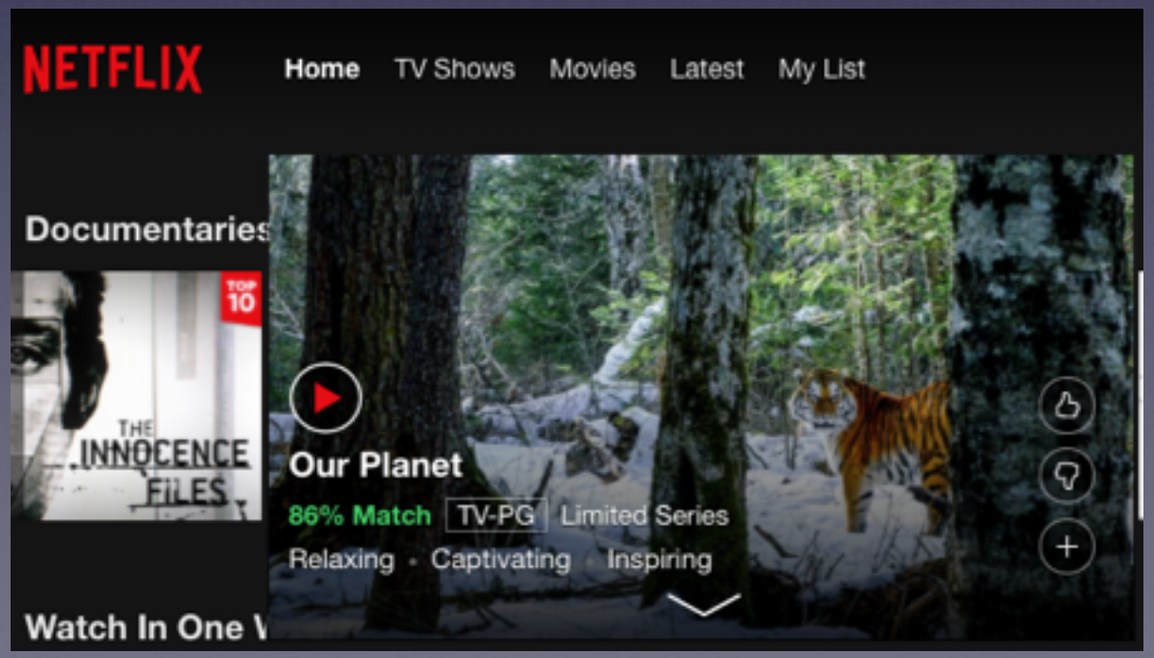
- On the bright side, there have been recent developments in python libraries such as Librosa and SoundFile which can help extract features from audio samples.

Objective

- Like most art forms, music can also exhibit an overlap between genres.
- To determine this overlap, we can make use of features from an audio sample to form a feature vector.
- Based on multi-class learning techniques, we can determine if feature vectors can belong to regions of multiple genres.
- And based on the proximity, we can help determine the probability of how ‘close’ a feature vector is to a genre or a group of genres.

Have we seen multi-class learning somewhere?

Yes! It is currently used by Netflix while viewing the details of a particular movie/ show.



Dataset Selection

- For this project, the following dataset has been chosen from Kaggle - https://www.kaggle.com/insiyeah/musicfeatures#data.csv

- The dataset contains information extracted from 1000 audio tracks each 30 seconds long. There are 10 genres, each represented by 100 tracks.

- The genres we will be working with are: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae & rock.

- There are about 29 attributes for each audio track.

Model Selection

- What I wanted to accomplish at first was that, the model should given an output for how similar a given audio sample is for each genre.
- For example : 70% Country & 30% Rock or 60% Pop & 40% Hip-Hop.
- To accomplish this result, I first thought of using mixture models. But there a re a few limitations which would not guarantee this kind of result.
- Hence, I decided to look into multi-class learning techniques. From these, I’ve decided to move forward with using softmax regression.

Learning Pipeline - Training

- My proposition is that out of the 100 tracks available for each genre, we can us about 40-70 tracks for building the genre vectors.
- From the 29 available attributes 20 attributes collectively make up this feature known as MFC. Since these features have more to do with the ‘shape’ of an audio sample, I’m thinking of not using these features or scaling them down to 10 cepstrums as these might have more influence in building the vector rather than features such as:
  - BPM(beats per minute) - describes the tempo of a song.
  - Spectral bandwidth - which determines the centroids along an audio waveform to better abstract its shape.
  - Zero-crossing rate - which determines how many times the signal moves across 0 on the y-axis (which denotes intensity).

Learning Pipeline - Testing

- Based on the model built using training data, we would test the model against the remaining 60-30 samples for each genre.
- Additionally, to make this more full-proof, we can have the user input a given audio sample, extract the relevant features from the model and then have the model predict what genre labels can be applied to the given audio track.

Technical Aspects

- While Scikit has provisions for using softmax with logistic regression, I would like to try using a similar offering from Tensorflow.
- If time permits, I would also try to implement this idea using Ludwig, which is a no-code command line wrapper for Tensorflow developed by Uber to experiment with rapid prototyping in a Machine Learning lifecycle.

Referred Links

https://musicinformationretrieval.com/

https://www.youtube.com/watch?v=nNDqbUhtIRg

https://towardsdatascience.com/multi-label-image-classification-in-tensorflow-2-0-7d4cf8a4bc72

https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

https://scikit-learn.org/stable/modules/multiclass.html

https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax

https://www.tensorflow.org/api_docs/python/tf/nn/softmax

https://uber.github.io/ludwig/examples/

Thank you

^_^