Project Description

Sayan Samanta
DATA 1030 - Hands On Machine Learning
Github

September 30, 2019

1 Introduction

The dataset under consideration is the **The Billboard 200 acoustic data** which encompasses the entire chart from 1963-2019, along with the EchoNest acoustic features of as many songs as available. All the features of the songs were obtained using Spotify's python API.

2 Dataset Description

The dataset has 2 tables:

1. albums - The raw dataset contains 574,000 rows which include all the albums that in the Billboard 200 starting from 1/5/1963 up to 1/19/2019. The features include: date of the record, artist name, album name, rank (at that week). no. of track in album, total duration (in miliseconds)

   Note. that if the song is not available for streaming on Spotify, then number of tracks and length of the album are represented as NaN

2. acoustic features - This contains Spotify EchoNest acoustic data for tracks appearing on Billboard 200 albums from 1/5/1963 to 1/19/2019. Each row contain the following features: An unique track ID on Spotify, Name of the track, album name, artist, acousticness, danceability of the track, duration of the track (in miliseconds), energy, intrumentalness, key, liveness, loudness (in dB), mode, speechiness, tempo, time signature, valence, album id., release date.

After removing NaN entries, the combined dataset (with each album in the first dataset cross-reference to the album ID on the second database) contains 5335557 entries (songs which stayed on the billboard for consecutive weeks, are in duplicates) with 25 features (after preprocessing).

The Spotify python API is free for public use (under MIT License). Hence the data can be supplemented with other data (such as songs that did not appear on the Billboard 200) for comparison purposes and further analysis. The problem described here is however based on the usage of this dataset alone.

The current dataset was curated for a piece on the data science analytics website Components. However that piece dealt with the album length alone, there could be plethora of other questions and analysis that can be answered/performed with this data.

3 Objectives

3.1 Target Variable

For the puporse of this project, we will however limit ourselves to predicting the 'era' of a song. In this context, we define 'era' as the decade in which the song appeared on the billboard top 200. Hence it is a **classification** problem.

3.2 Importance of the classifier

In our opinion, an 'era' predicting classifier will be instrumental in providing a variety of insights into the music industry. These include (but are not limited to)

1. The major characteristic of a 'chart-topping' track, which is an indirect indicator of the 'trend-setter' at a particular time-frame and how the charateristic (or the general listeners taste in music) evolved over time.

2. Measuring this trend over broad time-frames might indicate some of periodic/fickle/other patterns that exists in the listening habits over the years.

3. Comparing adjacent time-frames might give us insight as to major tracks/albums which caused a disruption in the listening choices of people.

4. The rise and fall of genres (subject to further availability of the 'genre' data)

A personal curiosity but perhaps not accurately answerable question (hence not promised as a deliverable) might be to prove if the top songs of today would appear in the charts of a previous era or vice versa. Another interesting challenge would be to correlate people's listening choices with the politico-socio-economic condition of the period in which the song was released and how it affected the nature of the songs itself. Such problems are subject to availability of more datapoints about songs that did not appear in the billboard top 200.

4 Preprocessing of the database

The following are the broad categories of pre-processing applied to the datasets:

1. The date values (date released and dates on the chart) were each parsed to python datetime object and split to 3 columns - day,month, year. No scaling is done as yet on these features.

2. The song_id and album_id are kept in the spotify unique id format. To be scaled upon further discussion.

3. Acousticness, danceability, energy, instrumentalness, liveness, mode, speechiness, valence features are already scaled between $[0, 1]$. No further pre-processing was required.

4. Rank, track duration, key, tempo, time signature, loudness, album duration, album length, were all scaled using the MinMax scaler as there are bounds to all of the features.