Generating music using AI. Physically using music files. We convert music into text data, then that becomes the data powering the whole network. Generates a text based data, then back to music.

<https://towardsdatascience.com/making-music-with-machine-learning-908ff1b57636>

<https://colab.research.google.com/notebooks/magenta/hello_magenta/hello_magenta.ipynb#scrollTo=DQLjca9SwOiI>

Goal: Viable idea & topics we learn we can apply

Target date: Next Tuesday

Spotify’s playlist recommender, compose personal data sets, get data from onemillionsong dataset.

Differentiate Genres – cluster analysis

Neural nets course renamed – Natural Language Processing. Changed 578, may have been 583 which didn’t exist until now.

Naïve bayes, support vector machine, DT, **Random Forest, kNN**

Sunday, January 24, 2021 meeting

Get yearly top 50 genres, predicting song trend. Break Spotify data down by genre. Stop for last 3-4 years, and target the songs.

**Proposal:**

Predict song trends by genre?

Predict what type of songs will be trendy for the following year per genre?

Using the trending change in those values, our output would follow the curve for the next year.

**data.csv** This file contains 170,653 songs from 1921-2020.

**data\_by\_year.csv** This data file contains the track data grouped by the year of release of each track, and allows time-series operations to be performed. Each row represents a single year, each column represents an audio feature. 100 rows/years.

**data\_by\_genres.csv** This data file the audio features of each genre. The rows represent different genres, and the columns represent different audio features. Contains 2,974 rows (including header), which means we have 2,973 distinct genres.

First, we should group the data by genre. Classify the genre of each track by similarity measure of their features matching the genre’s features. That might significantly reduce the amount of tracks we have per genre to predict off of. Assuming the tracks are distributed evenly, 170,653 tracks/2,973 genres would be about 54 tracks per genre. That wouldn’t even cover 1 song per year, since we have 100 years’ worth of data. May not be able to properly analyze genre trends over the years, unless we categorize the “genre” of each year based on the features in **data\_by\_year.csv**. That could somewhat predict genre trends over the years.

Instead of predicting what types of songs will be trendy based on previous years, we can predict *popularity* rating of tracks per genre by picking 10 tracks with highest popularity rating out of those 54 tracks per genre, find the most important features that correlate or cause a song to be popular (PCA || DT), then train our model with 30 more tracks to fine tune our results, and test on the remaining 14 tracks.

* Sub DM task A: Explore Spotify API
  + Objective is to find more tracks grouped by genre to add to our dataset for more accurate prediction of popularity. Must have popularity feature.

<https://docs.google.com/document/d/1PpFQOPqSElNmj8CcG4xnLKwtD1ApugU23Kz_9lfjQRU/edit>

Timeline:

√ 1/31 Write up Proposal

√ 2/9 Proposal Due

√ 2/16 Receive feedback on proposal

√ 2/19 Finish all of Task 1

3/2 Final Project Deliverable Discussion

3/7 Finish task 2, start task 3

3/14 Start on Final Paper, PP, & video

3/19 Final Project Due

Crystal = Purple

Tavis = Red

Both = Green

**Analysis Approach**

DM Task 1: Pre-processing (general or detailed)

-   Sub DM task A: Collect the Data

o  Scrape Billboard’s historical top genre charts off website for popular song list

X Create ‘unpopular’ songs list per year per genre – Spotify API or billboards that aren’t on our list

O Aggregate Year to songs

If any of the numerical values are missing, drop the row, unless it’s something we can estimate such as explicit flag

Extracting songs from Spotify’s API using the scraped data from Billboards charts. Feed Spotify API song name; returns song ID; use song id in get-audio-features endpoint to get same data schema as original dataset.

-   Sub DM task B: Data Pre-Processing

o  Convert categorical features (such as “mood”) into dummy variables.

o   Flag tracks from the top 100 that were not actually released in the same year for accurate comparison.

   o   Use PCA or Forward/Backward Selection to determine which features are most important to determining popularity of the song per genre

  o   Clean up encoding / address any missing data values

o   Using song ID, merge album information which contains release date

  o   Add higher weights to tracks in higher positions (1 = highest position)

DM Task 2: Exploratory phase e.g., clustering

-   Sub DM task A: Explore Spotify track data

o   Detect outliers

o   Detect correlated features

o  Analyze merits in features against the selected features the model outputs (e.g. duration would most likely get removed; using release date instead of date)

* Sub DM task B:  Use kNN to cluster buckets of top 100 tracks
  + I.e. Top 20; top 21-40; 41-60; etc.

DM Task 3: Supervised Learning : e.g., classification:

* Use Linear Regression, Decision Tree, & Naive Bayes to build different models that would evaluate the most important features to predict songs that would rank highly in that genre for the following year.  Training set will be all years prior to and including 2019.  Test set would be songs in 2020; and songs that were released in 2020 how they’d fair in 2021.
* Classification Analysis would use all of the data to determine the popularity of a track.  Output: boolean (popular or not popular)
* Predictive Analysis - Linear regression would only look at top 100 tracks.  Output: single song vector.

**Plan for Evaluation – Analysis of Results/ Discussion**

* Using the top 100 popular tracks per genre in combination with the additional “non-popular” tracks of the year, we will test our classifier’s ability to recognize potential popular tracks.  Popularity is defined as a track that made it to the top 100 genre chart.
* We will compare the results of our models via the statistics of our model’s performance, including accuracy, error rate, f-1 score, etc. to determine which performed the best prediction.
* Discuss ideal output vs actual results.