**ETL Project – Spotify Top 100 YoY Kasey Geist**

**11/6/19**

**­**

**Project Overview:**

At the end of each year, Spotify compiles a playlist of the 100 songs streamed most often over the course of that year. I was curious to see the difference in the artist who appeared YoY and to see if there was any significant change, not only in the # of songs, but in the types of songs that they produced that were in this top 100.

In looking at 2017 and 2018 results (last full years of data), I wanted to look to see if an artist's increase/decrease in the number of songs on the list relates to a style in music change in "Danceability" and "Energy" defined below.

* Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
* Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

**ETL Process Summary:**

In order to Extract, Transform, and Load this data the following steps/processes were done.

* **Extract:** Searched on various dataset websites and ultimately acquired two csv files via Kaggle, which were previously pulled from Spotify’s API. They contained unique information from what I could find from the source (“Spotify”) such as the “Danceability” and “Energy” metrics detailed above. I pulled these datasets into a python notebook using the “read\_csv” function and then began to transform the data.
* **Transform**:
  + First to transform the data, I only selected the columns that were relevant ['id', 'name', 'artists', 'danceability', 'energy'] for each year, eliminating several columns.
  + Then, I cleaned the data by rounding and summarizing the data to total the number of songs and provide the average of “Danceability” and “Energy” by artist for each year.
  + Next, I renamed the columns to identify the year they were from so when combining the data would be cleaner and easier to understand.
  + I then sorted the new data frame to have the artist with the most songs on the list appear at the top (descending order).
  + Now having two clean data sets, I created formulas to compare YoY in a column and then put this information into a comparison data frame (“df\_agg\_summary”).
* **Load**: Since there was a relationship between the two sources of data (i.e. the artists names) I decided to utilize a SQL database. The final database is called “spotify\_db” and contains the three tables that were the data frames that were transformed in python: “spotify\_2017”, “spotify\_2018”, and “spotify\_2017\_2018”.