**Hypothesis**:

What features of a song lead to a higher popularity score (85 and above or the top 15%) on Spotify?

Which features of song are highly correlated to one another and may have an effect on others?

**Narrative**:

For this analysis, I wanted to understand the aspects of songs such as the beats per minute, the length of the song, the amount of lyrics, or loudness that lead to its popularity in our country. Hence, I chose to use a dataset including the top songs from 2010 to 2019 from Spotify. When people are listening to songs, they usually will not consider these aspects when deciding if they like a song or if it is popular. Therefore, I wanted to see which of these hidden aspects have an effect on how popular a song is at the time.

**Data Source:**

<https://www.kaggle.com/leonardopena/top-spotify-songs-from-20102019-by-year>

Song Number – Song’s number in set

Title – Song’s title

Artist – Song’s artist

Top Genre – The genre of the track

Year – Song’s year in the Billboard

Bpm – (Beats Per Minute) The tempo of the song

Nrgy – (Energy) The energy of a song – the higher the value, the more energetic song

Dnce – (Danceability) The higher the value, the easier it is to dance to this song

dB – (Loudness) The higher the value, the louder the song

Live – (Liveness) The higher the value, the more likely the song is a live recording

Val – (Valence) The higher the value, the more positive mood for the song

Dur – (Length) The duration of the song

Acous – (Acousticness) The higher the value, the more acoustic the song is

Spch – (Speechiness) The higher the value, the more spoken word the song contains

Pop – (Popularity) The higher the value, the more popular the song is

**Step-by-step Instructions:**

**Graph Analysis**

**Step 1:** Load data into a dataframe

**Step 2:** check the dimension of the table

**Step 3:** Look at the data

**Step 4:** what type of variables are in the table

**Step 5:** import visualization packages

-set up the figure size

-make subplots

-Specify the features of interest

-Draw histograms

**Step 6:** Pearson Ranking

-set up the figure size

-import the package for visualization of the correlation

-extract the numpy arrays from the data frame

-Instantiate the visualizer with the Covariance ranking algorithm

-fit the data to the visualizer

-Transform the data

-Draw/show/poof the data

**Feature Reduction / Dimensionality**

**Step 7:** Log-transformation

-log-transformation method using numpy

-check the data for popularity

**Step 8:** check the distribution using histogram

-set up the figure size

**Step 9:** Feature Selection with Univariate Statistical Tests

-Create feature and target variable for Popularity problem

- feature extraction with the SelectKBest import

**Model Evaluation and Selection**

**Step 10:** Create training and testing data

-import ConfusionMatrix from yellowbrick classifier and train\_test\_split from sklearn

-split the data for validation

-instantiate model and visualizer

**Step 11:** Confusion Matrix

-create random forest classifier model

-fit, score, and display the visualizer

**Step 12:** Display number of samples in each set, training and validation

-Display number of songs in the top 15% of each set

**Step 13:** Classification Report and ROCAUC

-instantiate the classification report visualizer with the random forest classifier

-fit the training data to the visualizer

-evaluate the model on the test data

-instantiate the ROC and AUC visualizer with the random forest classifier

-fit the training data to the visualizer

-evaluate the model on the test data