EECS 349 Final Project
Justin Chen, Kenneth Gomez, Leanna Hue
Doug Downey

# Culture Clock Final Report

Our task is, given a song and its numerical attributes (such as danceability, acousticness, liveness, and others), determine if it'll be considered a "popular" song. We believe this task to hold value, as if it were to be implemented or questioned in the real world, you may want to know if your song will be commercially successful or not. Gathering information about the songs that are netting the most money nowadays could provide a wide range of potential uses, ranging from using it to make our own success through creating "better" songs, to understanding what's musically and culturally valued in our current society.

To build our dataset, we first had to create a collection of songs because there currently isn't a way to pull random songs from the Spotify API. In order to accomplish this, we scraped the website songfacts.com, which has a database of every song released sorted by year. So now we have a list of songs and their respective artist from 2008 - 2018, with approximately 2000 songs per year for a total of around 20,000 songs.

We are then able to work with the Spotify API to gather the statistics of each of these songs. The Spotify API, given a song, can provide a wide range of attributes about the song itself. For the purposes of this project, we are concerned with the following: duration, acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence, tempo, and popularity. Spotify can provide numeric values for each of these attributes for each song, and with this, we plan to determine what traits are common in the most popular song and understand what makes a song popular.

To test various models and determine what is the best model to represent and classify our data, we loaded our data into Weka and used 10-fold cross validation to test the following models: Decision Tree, K Nearest Neighbor, Naive Bayes, and Logistic Regression. For each attribute, we modified the values and studied the distribution for the normal attribute values, the square of the values, and the natural log of the values. We wanted to choose the modification of the attribute values that would be most descriptive and have a higher impact on the resulting classification. As a result, for our final tests, we chose values where the distribution was close to normal or even.  The images, as well as dissections into what each graph represents how we split up the data, are provided in the site.

Ultimately, in regards to our results, we found that there is no significant relationship between attributes and the popularity of a song. As shown in our image below (or in figure 38 on the website), unfortunately, none of these actually proved to be strong predictors. When you look at the distributions of all the attributes versus the popularity classifications, none of the attributes have a significant split. In each of the attributes, there's an even amount of each kind of song across the range of values. This implies that there really isn't a strong relationship between the attributes we selected and the popularity of the song.  Nonetheless, it was still interesting to look at how some combinations and transformation of the attributes tested could perform compared to others. Additionally, it was interesting to look at how the different machine learning algorithms performed amongst the various combinations and transformations despite them all utilizing the same dataset, ultimately creating a nice display of some algorithms' benefits and downsides.

| Algorithm | Accuracy using raw attribute values | Accuracy using transformed attribute values |
|---|---|---|
| Decision Trees | 39.26 | 25.69 |
| KNN (k=5) | 37.02 | 28.38 |
| Naive Bayes | 35.23 | 30.48 |
| Logistic Regression | 42.23 | 32.25 |
| ZeroR | 29.22 | 25.69 |

Besides simply transforming the individual attributes, we also tried kernel methods to further distinguish bad songs from good songs. The kernel classifier we found on Weka implemented kernel functions with logistic regression, however it only worked for binary attributes. We split the data in half, the bottom half representing unpopular songs and the upper half representing popular songs. We had to reduce the size of our dataset to 10,000 because Weka would crash if we put anymore. The results of this algorithm with 10-fold CV was 65.83%. This seemed really promising at first, but ZeroR was 63%, which means the model wasn't accomplishing much. So, our conclusion remains the same.

For future considerations, we believe there to be a fair number of factors that could play into what makes songs popular that we were unable to track with just Spotify's API. While we were able to gather some factors like danceability or tempo, there are a lot of musically external factors, such as how much effort was put into advertising the song, artist popularity, musical trends that may even change from year to year, lyrical content, and relational ties to other forms of medias (such as movies or events), that could change the outcome of popularity. Additionally, it was not entirely clear how some of the attributes used within Spotify's API were calculated. Some details and even general trends were provided, however, we lack any explanation of how they were achieved. If we are to overcome these, we may come up with a more numerical and reliable conclusion; however, until that time comes, we just have to note that music is a constantly changing area of our world that adapts just as quickly as our culture does.