Will My Song Be Spotify Famous?

Final Project for UCLA STATS 418

Daehyun Kim, Wesley Lau, & Austin Haynesworth

Introduction

What makes a song popular is a complex question. Undoubtedly, external social factors such as the level of fame of an artist or a song's social significance can impact its popularity. Internal musical attributes of a song, however, also play a critical role to its commercial success.

Having the ability to pre-determine the popularity of a song based on its musical attributes could be useful to singers, songwriters, and labels, and could help them decide the direction of their music.

How do we break a song down into its "musical attributes"? Currently, Spotify parsing software is able to break down a song into 13 musical attributes such as tempo, danceability, and time signature. Some metrics, such as length in milliseconds or loudness in decibels are easily quantified numerically. Others, like "acousticness" and "danceability" are determined through the parsing software and given a value between 0 and 1. Our dataset contains 19000 songs that have gone through this parsing process.

Our goal was to see how musical features themselves can predict a song's popularity. We did this by using the 13 audio features as our predictor variables. We used Spotify's "Song Popularity" variable as our response variable. Song popularity is a numeric value between 0-100 determined based on the number of plays and recentness of plays a song has received. For ease of use, the popularity in our model has been binned into 5 categories, 1-5, with being the most popular. Several predictive models of various types (Linear SVC, Random Forest, K Nearest Neighbors, etc.) were tested and the most accurate one was selected.

Lastly, we built a Flask App with HTML user interface such that a user can manually input the audio features of their own original song (i.e. a song that does not exist in our 19000 song data set) and predict its popularity. The UI has two components. First, a user can look up familiar songs using a search feature that connects to a Spotify API; the API will then return the 13 musical attributes of the known song. This allows the user to become familiar with the attributes. Using this information, the user can then input their estimates for an unknown or original song for each of the 13 attributes. These values are then run through our model and the predicted popularity of the song is returned to the user.

Overview of Data Set

Dataset: https://www.kaggle.com/edalrami/19000-spotify-songs

Our dataset is taken from kaggle, and represents audio features for around 19,000 songs in the Spotify database. The following table lists all the model variables and a short description. As we mentioned, these features are parsed by Spotify and can be accessed for any song in the Spotify database via their API.

As not all the feature names are intuitive for non-musicians, we performed some scaling to the variables in the flask app UI as well as some data transformations to the model variables which will be explained in the next section.

Feature	Explanation
duration_ms	The duration of the track in milliseconds.
key	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.
audio_mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
time_signature	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
danceability	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	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.

instrumentalness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
loudness	A categorical variable representing the loudness of a track. Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values that range between -40 and 0 db are binned into levels 1-4 with 4 being the loudest value.
speechiness	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music.Values below 0.33 most likely represent music and other non-speech-like tracks.
audio_valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.
song_popularity	Song ratings of spotify audience.

Exploratory Data Analysis and Data Transformations

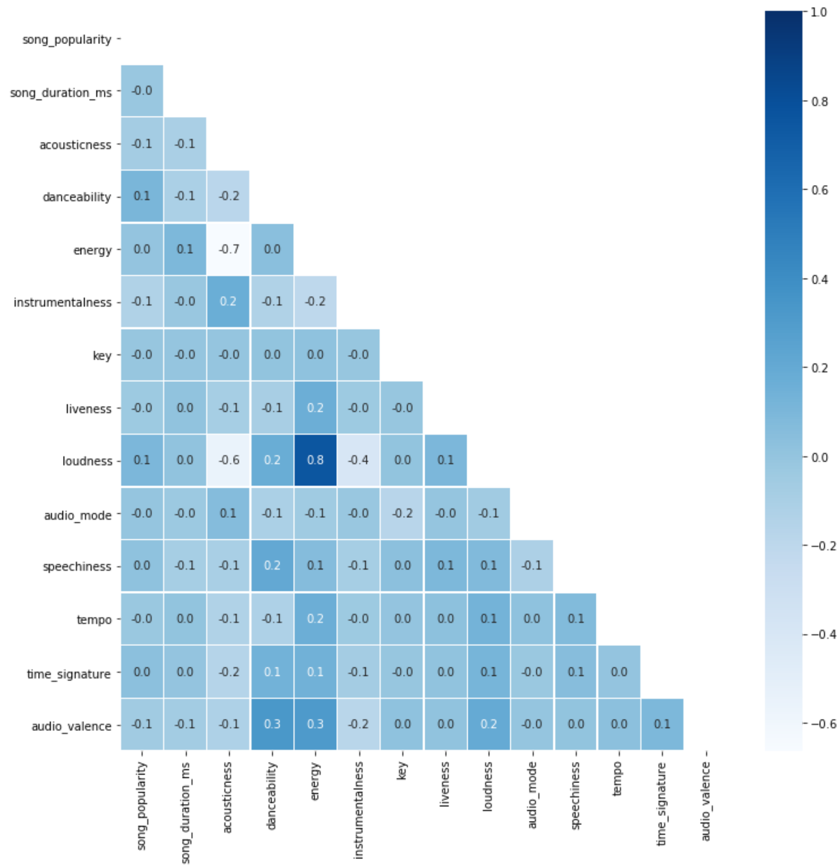
The top 5 most popular songs on spotify as of Nov 22, 2018 are shown below. One factor that the spotify popularity score is based on is the number of plays each track gets, and the recentness of those plays. Our 13 predictor variables do not include a release date or another measure of recentness of a song. Even though that can arguably help our model better predict the popularity of each track in the spotify database, it would not be helpful in the case where a user wants to predict how popular a new song would be that has never before been released.

Top 5 Songs

	Song Title	Song Popularity
1	Happier	100
2	I Love It (& Lil Pump)	99
3	Eastside (with Halsey & Khalid)	98
4	In My Feelings	98
5	Taki Taki (with Selena Gomez, Ozuna & Cardi B)	98

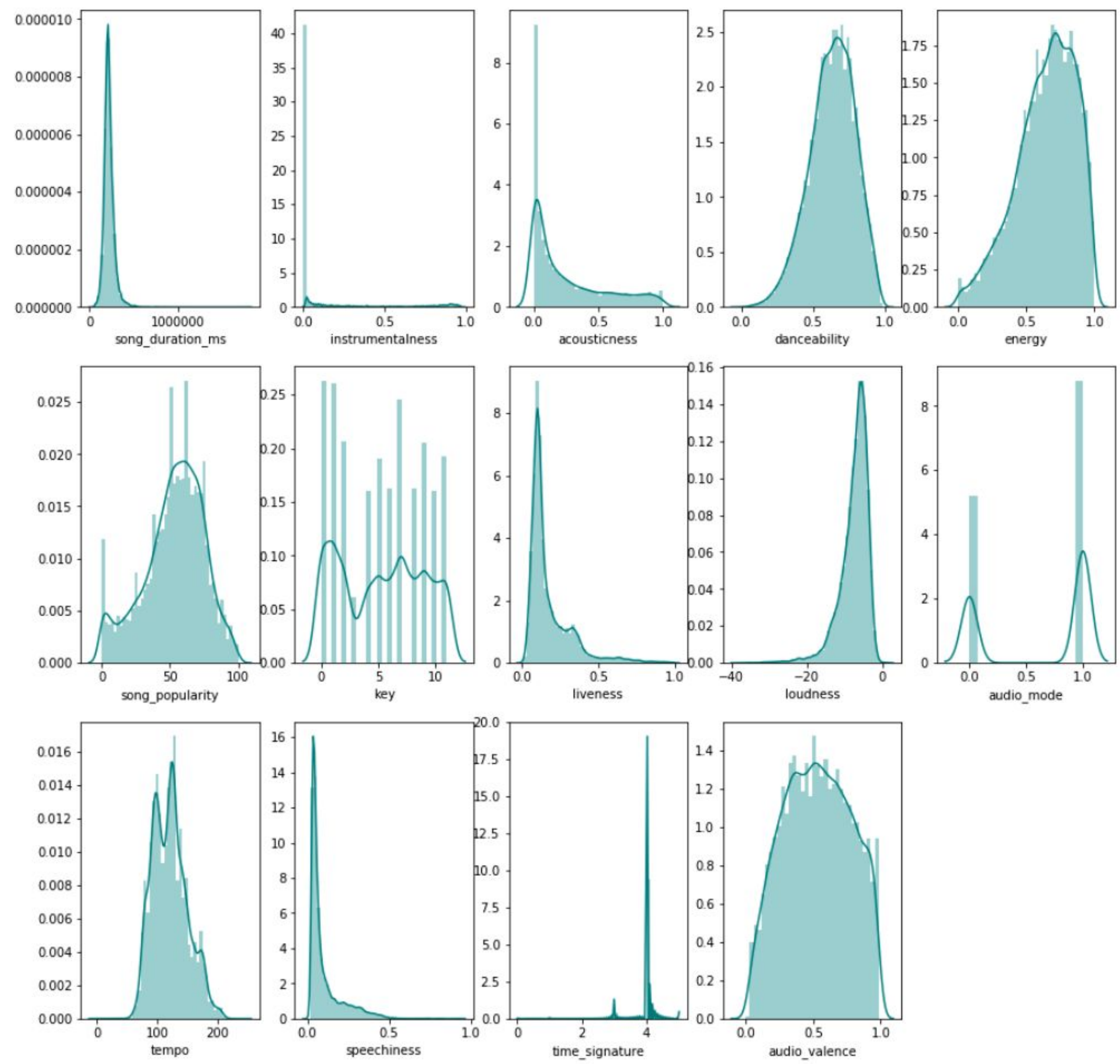
Correlation Chart for Predictor Variables

We found a high correlation between loudness and energy equal to +0.8. The second highest correlation is between energy and acousticness which is -0.7. When we compare the correlation between song_popularity and all other features, nothing stands out as being strongly correlated to our dependent variable.



Distributions of Variables

Not all of our variables turned out to be normally distributed, which makes sense given their definitions. For example: key and audio mode are discrete variables. Time signature is mostly represented by a few values.



The only variables that were transformed from the original data we received was loudness and song popularity. We found that most of the songs only ranged from -40dB to 0dB in loudness, so we ended up binning the loudness into qualitative bins from 1-4, with 4 representing the loudest value. Finally, we transformed the dependent variable, song popularity, into levels of 1-5 instead being in a 100 point scale.

Besides our transformations for modeling purposes, we also scaled some of the varaibles to exist from a 0-10 range instead of a 0-1 range. This allowed for a easy to understand UI interface with our flask app, where users can rate their own song on a 10 point scale.

Model Selection and Performance

For our model, our response variable was popularity, binned into 5 categories, with 5 being most popular. Our predictor variables were the 13 musical attributes, created from Spotifies parsing software.

Our goal in model selection was to find the most accurate model for predicting popularity. First data was split into train and test sets, with a 75:25 ratio. A grid search was run on the train data set using Linear SVC, K Nearest Neighbors, Perceptron, Logistic Regression, Multi-layer Perceptron (MLP) neural networks, and random forest models with various parameters. A total of 27 models were created via the grid search, and the model with highest accuracy was chosen.

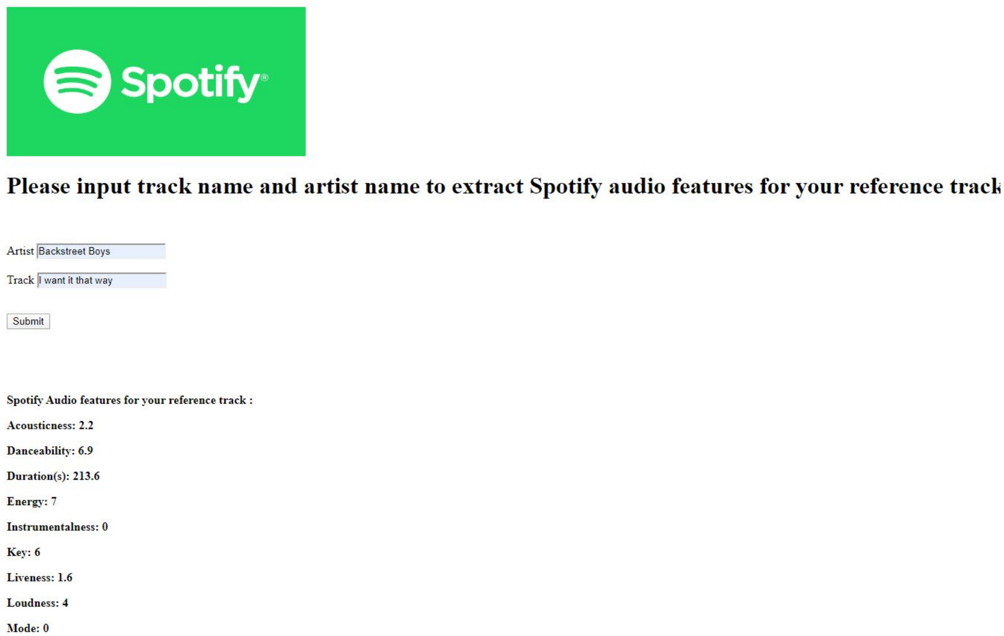
A random forest model was found to have the best accuracy. Accuracy, precision, recall, and F1 score were also found for this model.

Random forest model performance	
Accuracy	0.569
Precision	0.653
Recall	0.506
F1	0.518

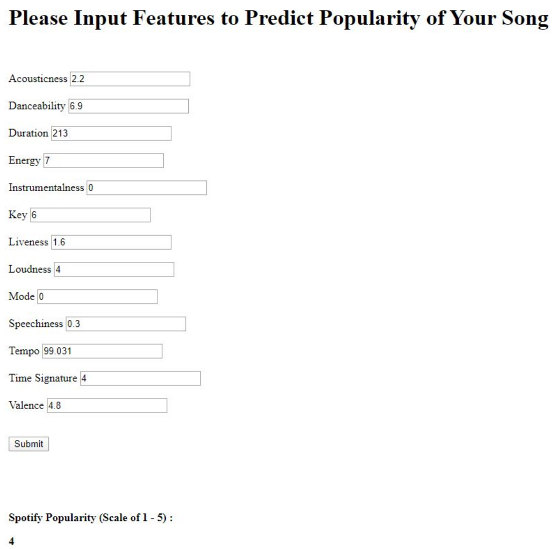
This selected model will be used to predict the popularity of original songs, with musical attributes input by the user.

Flask App Production

Machine Learning models are no use if there is no application utilizing the insights gained from the models. For this purpose, we developed a simple Flask App with HTML user interface where a user can manually input the audio features of their own original song (i.e. a song that does not exist in Spotify) and predict its popularity of Scale 1-5 (5 being most popular, 1 being unpopular). The app has two pages.



The first page lets a user search familiar songs from Spotify API and extract the 13 audio features of the known song. This helps the user become familiar with the attributes since most users would not have much understanding of what these numerical features are for their own song. Using this information and the feature descriptions on the same page, the user can probably refer to well-known songs that are similar to their original song to come up with that song's numeric values for the audio features.



Then, on the second page, the user can input their estimates for an unknown or original song for each of the 13 attributes. These values are then run through our model in the back-end and the predicted popularity of the song is returned to the user.

This app along with the general insights from our model can provide useful business cases for musical composers who can manipulate these musical attributes when they produce their original songs. Using the app, musical composers can analyze and test their own songs' audio features to make their songs as popular as possible on Spotify and many other music purchase/streaming platforms.

Reflections & Future Directions

Ultimately, our goal was to use ML to predict the Spotify popularity of a song based on its Spotify musical attributes.

Reflecting on our model, despite using a dataset of 19000 songs and testing nearly 30 models, our chosen model only has a 57% accuracy. This hints that factors beyond musical attributes (such as a songs thematic material or an artist's name recognition) may also play a large role in a songs popularity.

That said, musical attributes clearly also play a role, and our Flask app can be used to help singers, songwriters, and others in the music industry predict their song popularity.

While the app in its current form is useful, there are areas for improvement. Currently, users must estimate the values of numerical audio features, using their understanding of these numerical values based on reference songs. It would be more useful if the user could instead upload audio files, and the song would be parsed for these audio features, like Spotify currently does. Unfortunately, Spotify's parsing software is proprietary and not open source. For future directions, we would like to improve input methods for users, such that the numerical values they input for audio features are more precise, hence leading to a more precise popularity prediction.