CS539 Project Proposal

Alexander Beck

Nate Schneider

Dataset

For the final project, we intend to create a model to determine if a song will be “popular.” To accomplish this, we first found a dataset (or 4, but settled one this one) that contained the following information[1]:

- 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.
- 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.
- 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. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
- instrumentalness: Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
- 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.
- liveness: 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: The overall loudness of a track in decibels (dB). 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 typical range between -60 and 0 db.
- 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.

[1] Some of the descriptions for these were found here because the dataset we used had truncated the descriptions. The source of the datasets were both Spotify, so the column descriptions were nearly identical.

- 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.
- 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.
- time_signature: An estimated time signature. The time signature (meter) is a notational convention to specify how many beats
- 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).
- **track_popularity**: Song Popularity (0-100) where higher is better.

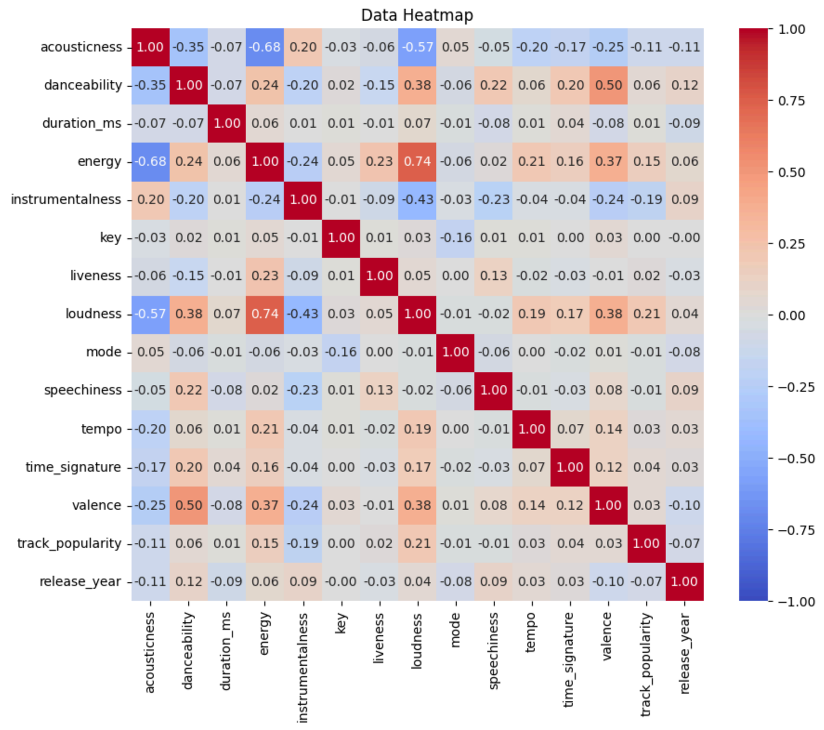
The above is all integers or floats, which make them easy to work with. The following is also in the dataset, though is not as straightforward for creating a model. Some of it may end up being used, such as genre_0 or explicit.

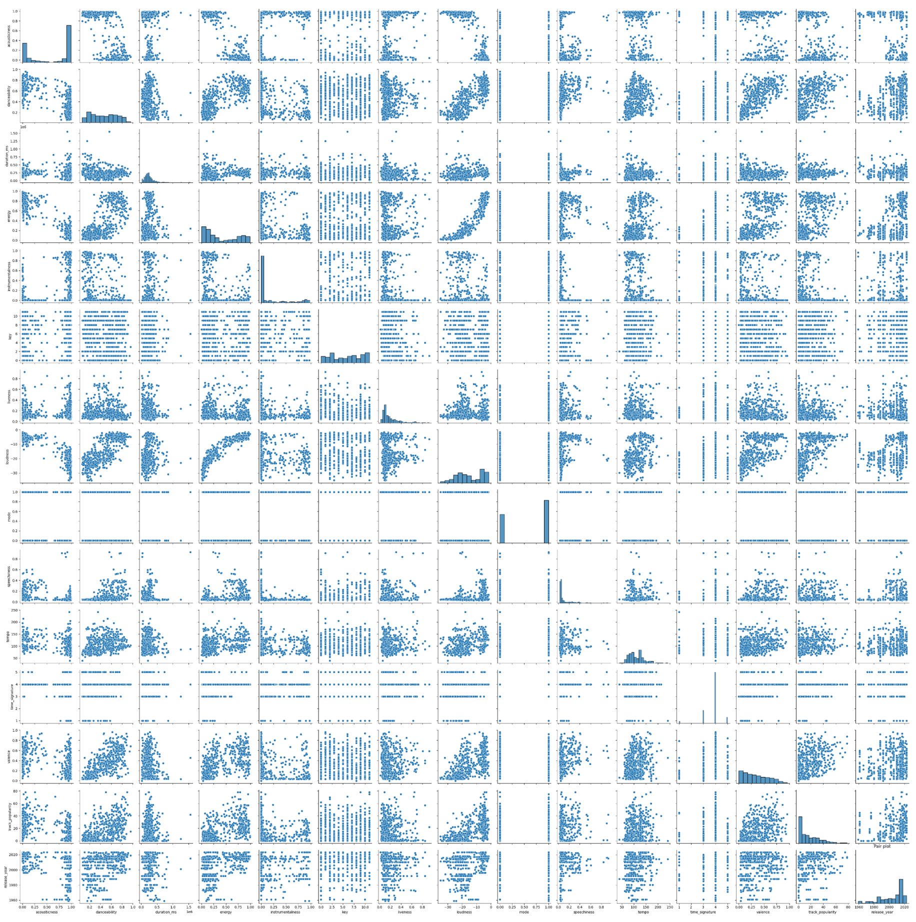
- album_id: The Spotify ID for the album.
- album_name: The name of the album. In case of an album takedown, the value may be an empty string.
- album_popularity: The popularity of the album. The value will be between 0 and 100, with 100 being the most popular.
- album_type: The type of the album. Allowed values: album, single, compilation
- artists: The artists who performed the track
- artist_0: The main artist.
- artist_1 - artist_4: The featuring artist(s). Multiple columns.
- artist_id: The Spotify ID for the main artist.
- duration_sec: Track length in seconds
- duration_ms: The duration of the track in milliseconds.
- label: The label associated with the album.
- release_date: The date the album was first released.
- total_tracks: The number of tracks in the album.
- track_id: The Spotify ID for the track.
- track_name: Name of the track
- track_number: The number of the track. If an album has several discs, the track number is the number on the specified disc.
- artist_genres: A list of the genres the artist is associated with. If not yet classified, the array is empty.
- artist_popularity: The popularity of the artist. The value will be between 0 and 100, with 100 being the most popular.
- followers: The total number of followers for the main artist.

- name: Name of the Artist
- genre_0: The main genre
- genre_1 - genre_4: The subgenre(s). Multiple columns.
- analysis_url: A URL to access the full audio analysis of this track. An access token is required to access this data.
- track_href: A link to the Web API endpoint providing full details of the track.
- type: The object type.
- uri: The Spotify URI for the track.
- explicit: Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown).
- release_year: The year of the album/single release
- release_month: The month of the album/single release
- rn: Arguably the MOST important column in the dataset. The description of it is “ignore”, and 100% of it is just 1’s.

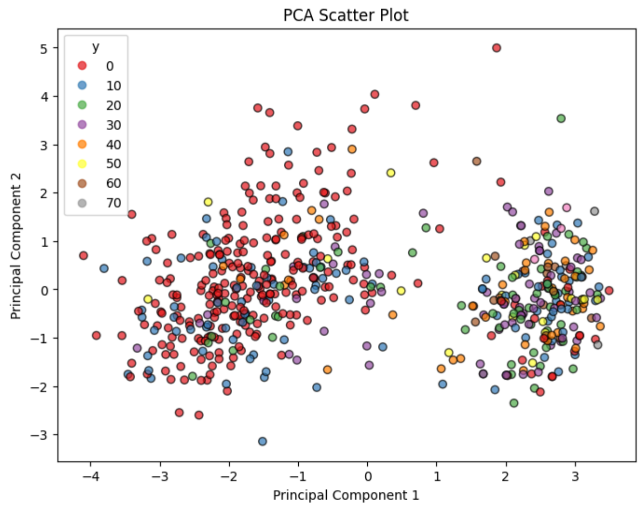
Data Exploration

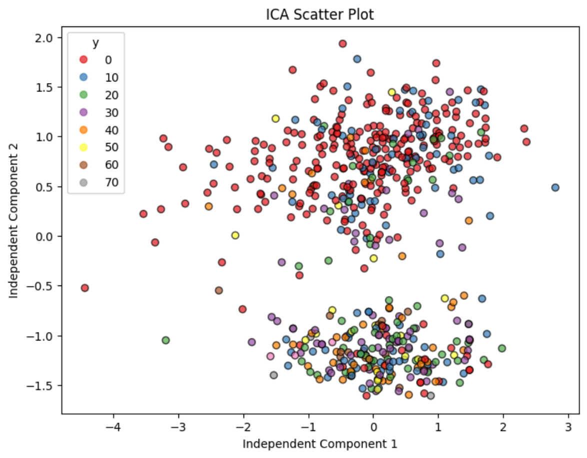
The exploration we did was very similar to the one done in class. We first created a heatmap of the data, then we plotted the data against itself (a pair plot). It is near impossible to see on this document, so I don’t recommend trying. The general shape of the plots, however, do reveal some information. For example, the loudness and energy plot have a very defined curve, which is supported by the positive correlation in the heatmap. Additionally, it shows certain data types, such as that key is not a continuous number (which is already known information, but is still shown).





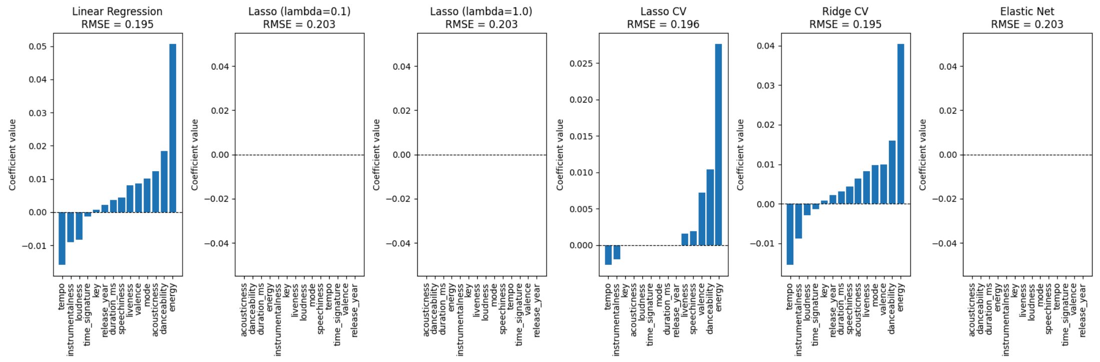
We also tried doing PCA and ICA, with the results below.





There are clearly two groups within the PCA and ICA, which is supported by a K-Means which has a peak silhouette score of 0.64 with 2 clusters.

Finally, we fit the data to some linear models. We defined the goal as track_popularity > 50, because just setting the goal as track_popularity was trying to make linear regression guess the number from 1-100, which is not its intended use case. The results are shown below.



The RMSE for Linear Regression is 0.195, which is a clear indicator that it is possible to predict the popularity of songs using this dataset.

Project Objective

Our goal is to find a way to separate data into partitions, run linear regression, and combine the model parameters to maximize the ability to detect if a song is “popular.”

Baseline Model

There are a few baseline models we intend to use on the data. We can fine-tune a Linear Regression, along with a Lasso and Ridge to see which gives the best performance. Additionally, using k-fold on the data would help to keep the model from being over-tuned. Above is what the Linear Regression looks like out-of-the-box

Proposed Adjustments

We intend to run Linear Regression on different sets of the model, and try various mathematical operations of combining the parameters in the end. We theorize that this will give a more generalized model since it will reduce overfitting. Some mathematical operations include using the mean, using the median, and even adding bits of randomness and seeing the results. The specific operations are not yet known, as only by doing them would we see which ones work the best. This is a variation of a k-fold cross validation. There are two options that we can modify the k-fold with: instead of progressively changing the model parameters, we do all of them individually and bulk combine them in the end, or we can do it progressively like k-folds is intended. Comparing the two methods might make for a better model.