Spotif-why

Predicting song popularity using Spotify music features

Kushagra Agarwal, Sorin Cho, Diana Lee, Aaron Wang

Introduction

Given the many song features provided by Spotify and the wide selection of songs on Spotify, we were curious if a song's popularity could be predicted based on its features. Although we expect the song's artist's popularity to be the largest contributing factor, we were interested in seeing if a hit song could be "manufactured" by meeting specific song feature criteria that maximizes popularity.

Data

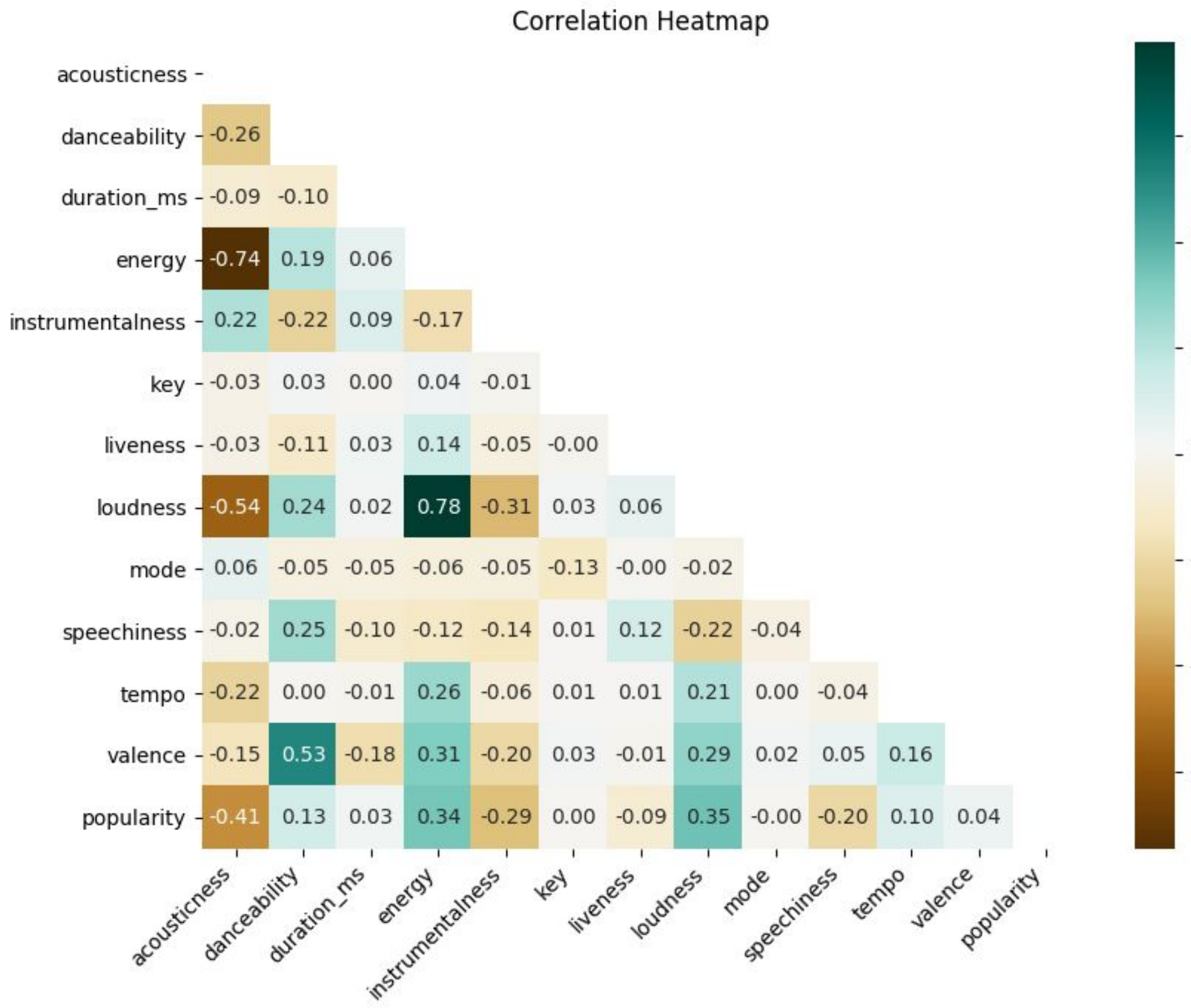
First, we obtained data from a Kaggle dataset which gave us songs from 1921-2020.. We also scraped songs from Spotify's U.S. weekly top 200 songs from 2017-present to get a sample of popular songs. For the scraped songs, the Spotify API was used to obtain song features. This gave us a total of 160,341 songs. We used the following song features:

- Valence
- Danceability
- Energy
- Loudness
- Tempo
- Duration
- Mode
- Speechiness
- Acousticness
- Instrumentalness
- Liveness
- Key
- Popularity

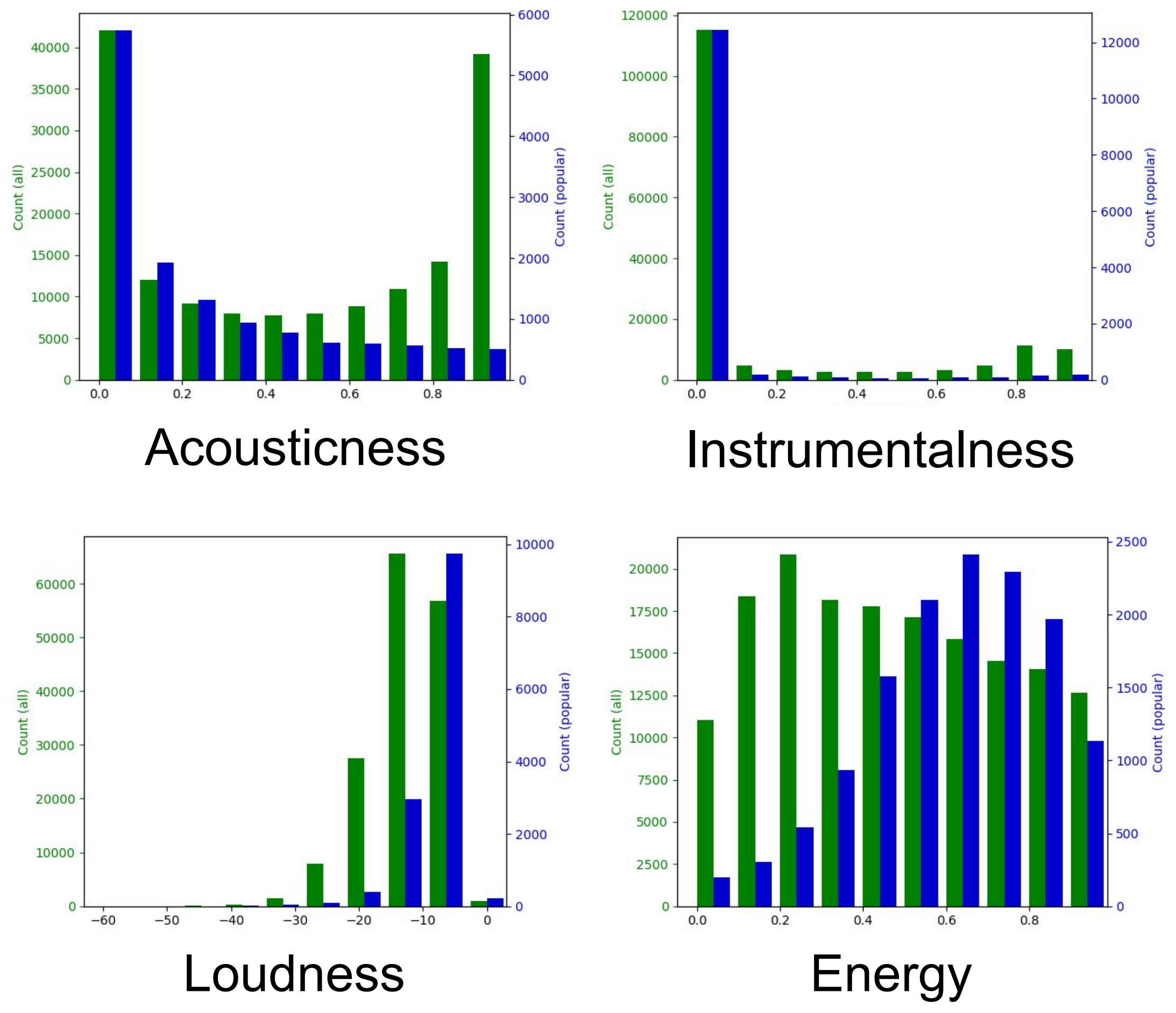
Methodology

1. Analyzed feature distributions and correlations between features to determine which features were the most impactful on popularity.
2. Data was divided into a 80-20 train-test split to minimize any bias and to evaluate the predictive ability of the models. The three models we attempted to use were a multiple linear regression, a support vector regression, and an artificial neural network.
3. Compared models on the basis of mean squared error since this was a metric consistent across the models. We also graphed predicted popularity vs expected popularity and residuals vs expected popularity to get a better understanding of each models' performance.

Preliminary Investigation on Features

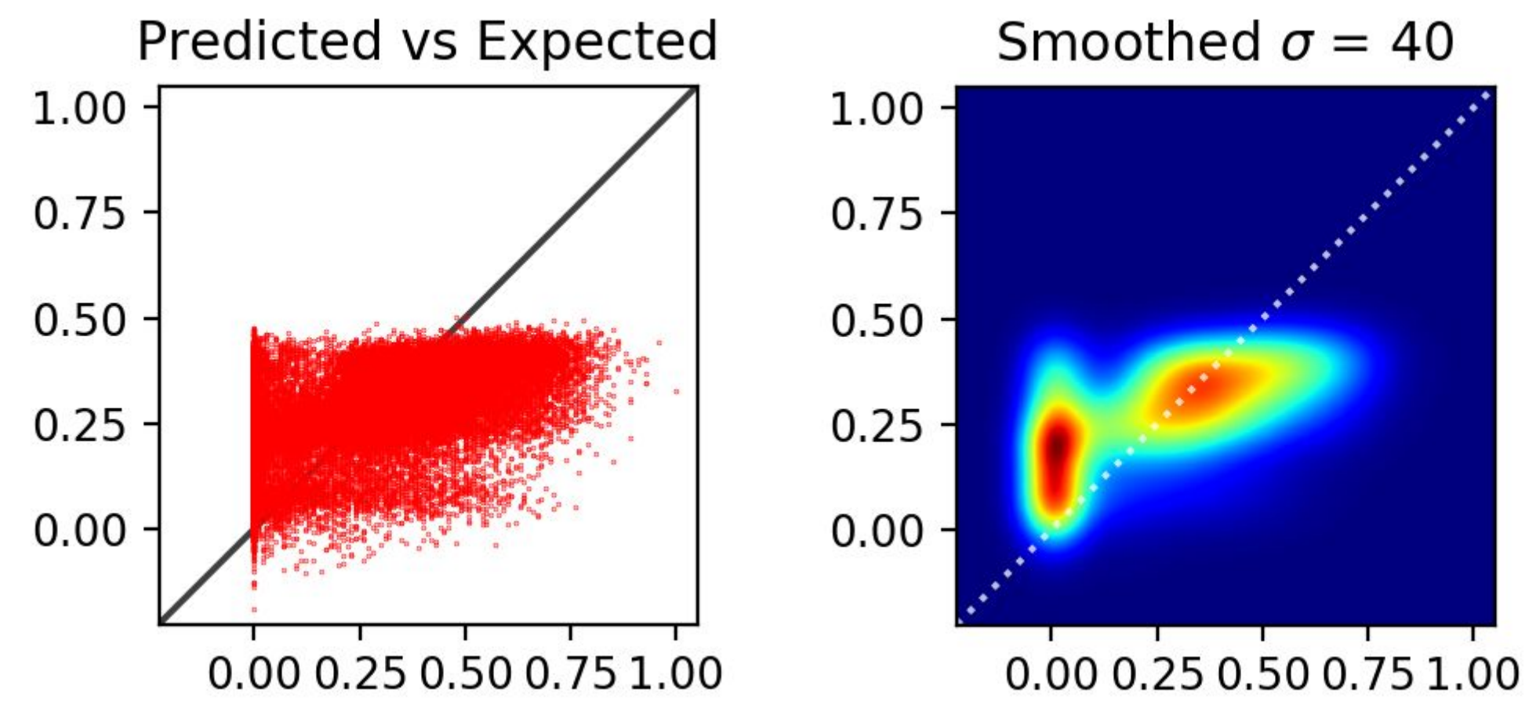


Song features and correlations with other features (popularity is the bottom row)

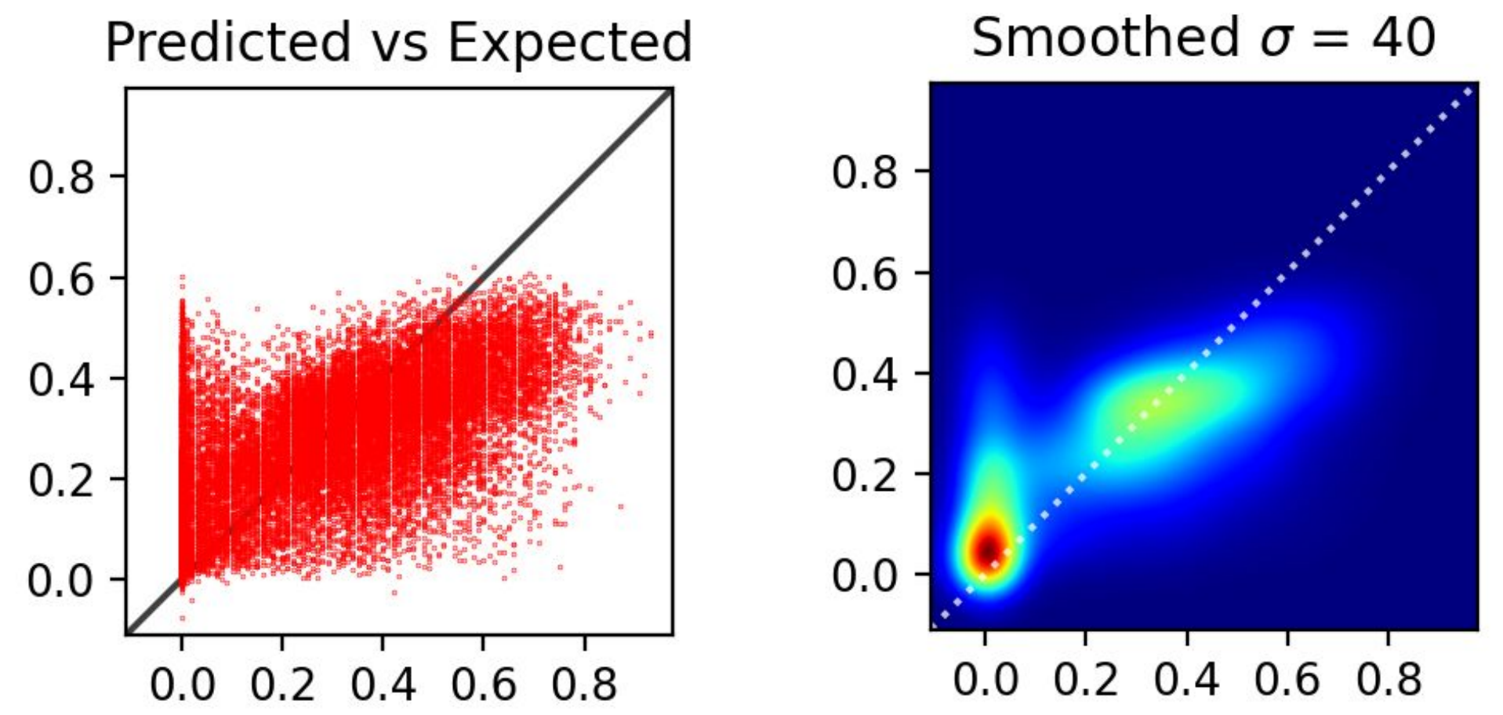


Sample of song feature distributions (green: all songs, blue: songs with popularity > 60)

Regression Results

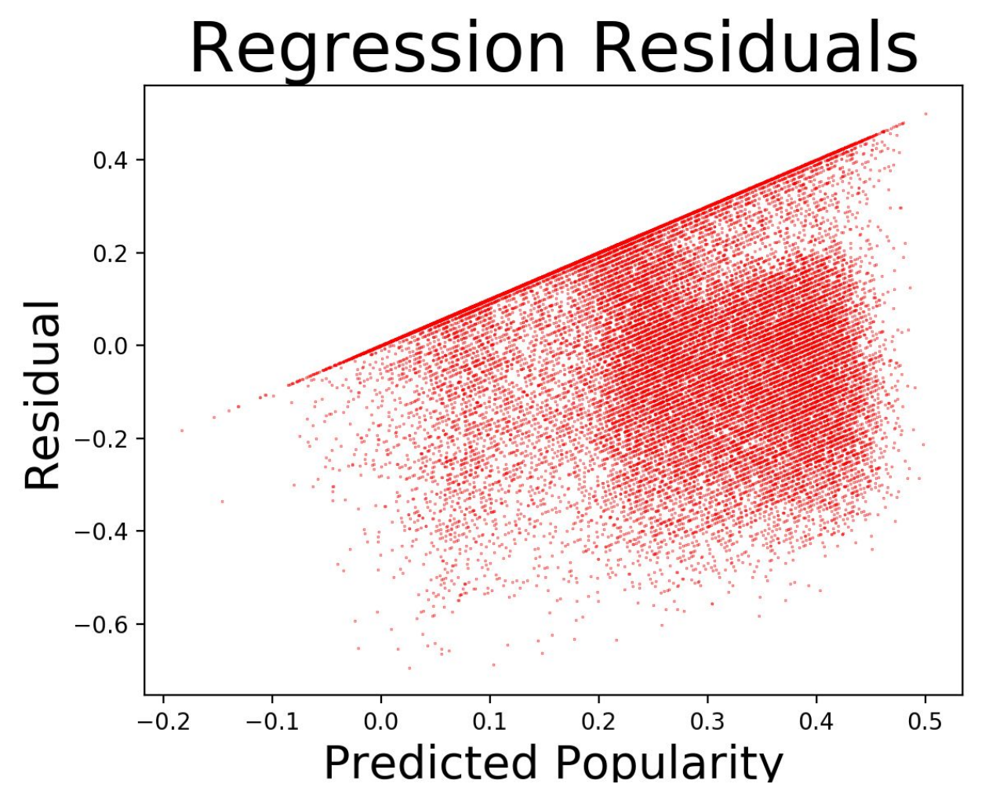


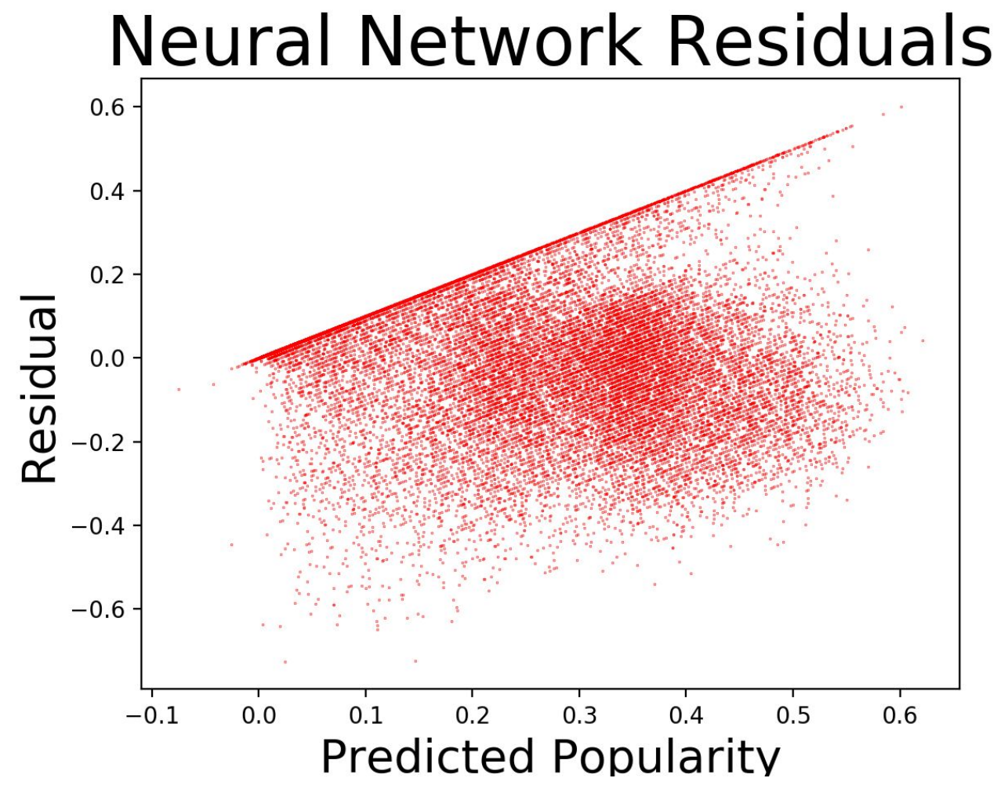
Neural Net Results



These graphs plot the model's predicted popularity against the popularity score Spotify assigned. 2D histogram binning and Gaussian blurring were applied to get a better sense of the distribution of the scatter plot since there are so many data points.

	Multiple Linear Regression	Neural Network
Training MSE	0.0351	0.0275
Testing MSE	0.0350	0.0276





Mostly random residual trends for both models indicate that the models we used were reasonable for the prediction task.

Conclusions

1. **Popularity can't be exactly predicted but can be approximated using musical features**
   Given our model's MSE's, accurately predicting popularity using our models is unlikely, but the models can predict within a moderate range.
2. **More complex models aren't the key to predicting musical popularity**
   The neural net is only slightly better at predicting popularity than linear regression. This, we believe, is evidence that we need another approach to modelling musical popularity, than just more complex models. Music features alone cannot predict a songs popularity.
3. **Acousticness, Instrumentalness, Loudness, and Energy are the most significant contributors to musical popularity.**
   The heatmap shows these 4 attributes have the highest magnitude in influencing popularity. There is also significant difference in the distribution of popular songs and all songs for these features as seen by the histograms.

Challenges

- Initial dataset of only popular songs gave very poor results (MSE: 0.0678).
- Larger dataset improved our results by almost halving the MSE, but we were unable to successfully use the SVM with the large dataset
- There are many instances where a song's single release is initially popular but has a large drop in popularity due to the song eventually being released on an album. Filtering these songs required manual work.

Moving Forward

1. **Grouping by genre or region**
   Calculating separate popularity scores region - wise, and training genre specific models might lead to better performance.
2. **Grouping by popularity**
   We found that the skew in data between less popular and more popular songs is hurting our performance. Training 2 models and a simple classifier that predicts if the song will be 'popular' or 'unpopular' to pass onto the right model, might lead to much better performance.