Scott Magnusson and Kyle Halloran
Data Mining and Knowledge Discovery
May 31, 2019

Predicting Song Success

Introduction

For our final project, we created a model that predicts song success. Specifically, we used song lyrics to improve upon past predictive models which used many features but failed to include lyrics. Our models greatly improve upon past techniques, likely because of the addition of lyric characteristics. In addition to testing predictive models, we also examined the correlation of specific musical characteristics with song success.

Data

We collected data from a variety of sources in order to create the full dataset. Our primary source of data was the Million Song Database, which contains audio features and metadata for a million contemporary music tracks. Because the dataset was too large the analyze efficiently, we used a randomly selected subset of 5,000 songs that were released between 1970 and 2010. We also scraped Genius.com for the lyrics attached to each of the songs in the database. We were able to successfully pull lyrics for around 80% of the dataset. For the sake of our analysis, we decided to drop the 20% of songs that we could not obtain lyrics for.

The Million Song Database includes a variable that measure song popularity ("song hotness"), but almost half of the observations for this variable were left empty. So in order to classify a song as "popular" or "not popular", we scraped the weekly Billboard Top 100 database from 1970 to 2012. We then created a dummy variable in our dataset which denoted whether the song ever appeared on the Billboard Top 100. A third of the songs from our original dataset appeared on the Billboard list at some point after the song's release.

Methods

To extract insights from our dataset, we used various machine learning algorithms as tools for our prediction analysis and data exploration. After cleaning the dataset, we engineered two features, 'lyric_predict' and 'sentiment'. The former is a prediction based on an XGBoost model that estimates the likelihood that a song made the Billboard Top 100 given a vectorized version of its lyrics. The training and testing datasets were split to be respectively 40% and 50% of the full dataset size. We had to balance a smaller dataset size with increased probability of overfitting from the lyric prediction feature.

To build our lyrics model, we cross-compared results using different word vectorizers, including the TF-IDF Vectorizer and Count Vectorizer, under the XGBoost model with parameter tuning on n_grams and min_df. On top of this, we played around with different preprocessing techniques such as word lemmatization, stemming, stop words, punctuation, and number formatting, but in the end found that only removing stop words increased the performance of our model. We also computed the sentiment of the lyrics, which we found during our data exploration, and added it to our feature list

to improve the prediction accuracy. We ultimately determined that using the TF-IDF Vectorizer and XGBoost model produced the most accurate predictions.

Of the 22 features that were chosen, two of them were engineered and the rest were handpicked from the Million Song Dataset. We wanted to create a model that was not based on an artist's previous song history so that a new artist and an artist with previous history would be judged more equally. Our criteria for feature selection included removing features with a high portion of missing values such as genre and data that had been computed by other algorithms such as artist hotness and familiarity. We followed a similar method to that of Nasreldin (2018), but included the extra removal of familiarity and artist hotness.

After selecting our 22 features, we then examined the various model performances of XGBoost, Logistic Regression, K-Nearest-Neighbor, Decision Tree, and Random Forest Classifier. Using parameter tuning, the best performances were squeezed out of our models and then scored on an AUC measurement with 5 fold cross validation to examine how our predictions compared to those of Nasreldin's team along with F-1, precision, and recall scores.

Results

The results from the prediction model are varied and point towards overfitting of the training dataset. Our cross-validation scores were consistently high, reaching into **89.6%** AUC scores for our best performing tuned XGBoost model, yet we saw a relatively low F-1 score of **26.8%**. We believe this is due to a small training dataset size in relation to features used, while also including the 'lyric_predict' feature. We are unable to cross-compare our F-1 results against those published by Nasreldin as their report only includes AUC scores from cross-validation on training sets.

In terms of overall model performance, we discovered that a tuned XGBoost model provided the best overall scores, while the K-nearest Neighbors model performed the worst. The Decision Tree model performed marginally better in F-1 scores, but the XGBoost model had better precision. In comparison to our dummy classifier, which performed at an F-1 of **19.3%**, our models were successful.

AUC Cross Validation Results

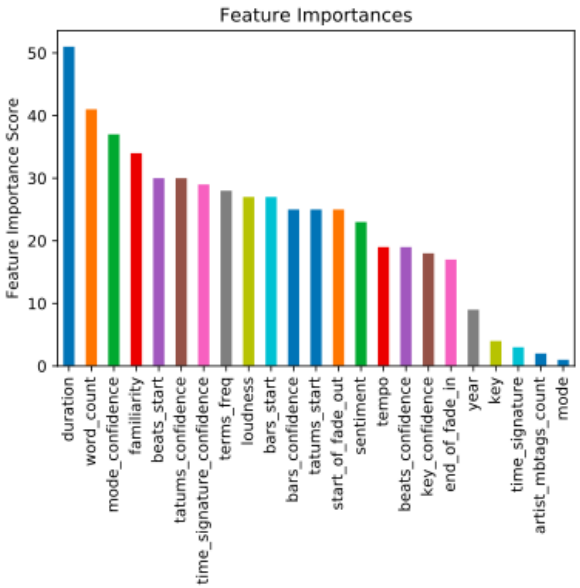
Nasreldin's Results

	Model
Score	
0.632412	XGB
0.617116	Logistic Regression
0.611623	Random Forest
0.542687	KNN
0.520234	Decision Tree
0.512634	Support Vector Machines

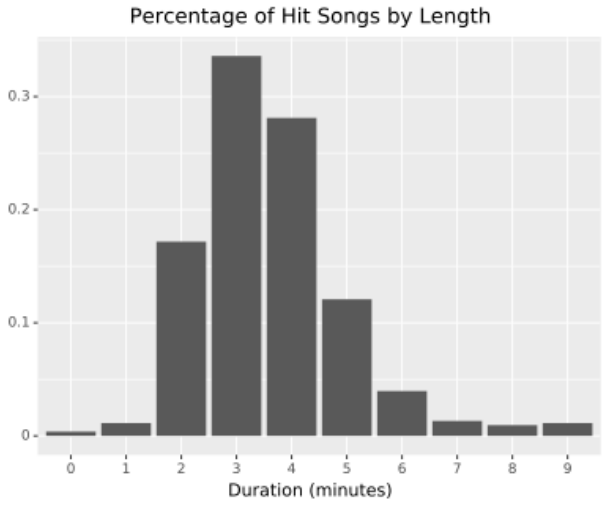
Our Results

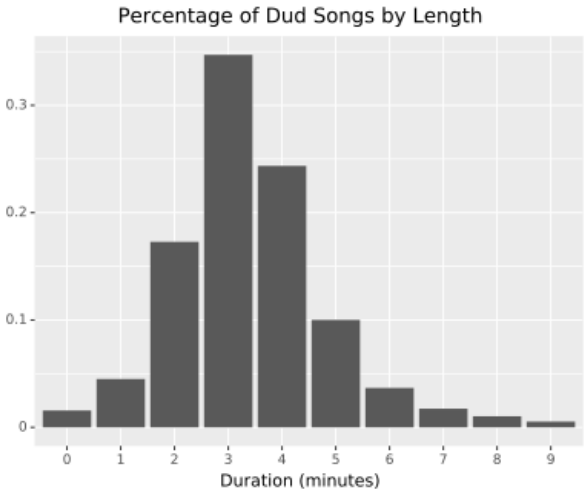
Score	**Model**
0.897034	Logistic Regression
0.896165	XGBoost
0.872875	Random Forest
0.856571	Decission Tree
0.569948	KNN

Along with building our predictive model, we also explored a few of the most important factors in predicting whether a song will be popular. After running the data through the XGBoost classifier, we were able determine the importance of each individual feature by using their respective F-scores. The chart below graphs each of the variables in order of importance to the model:



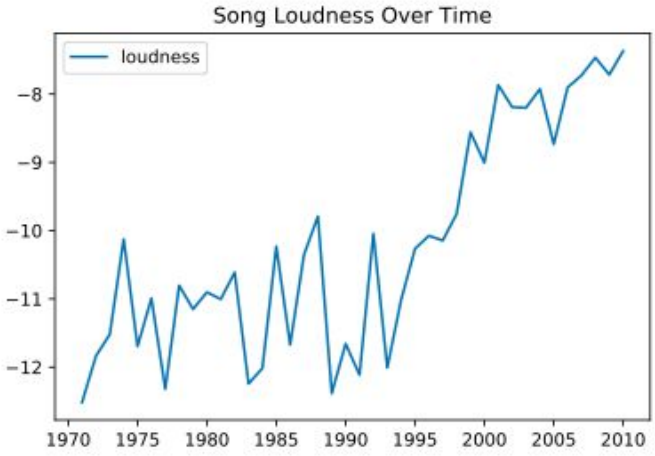
Duration is ranked as the most important variable in this chart. While we could not determine whether duration is the *cause* of a song's popularity, the data show that song duration is highly correlated with song popularity - hit songs are on average ten seconds longer than non-hit songs. While the average song which made the Billboard Top 100 was **4.1 minutes** long, the average song that did not make the Billboard Top 100 was **3.9 minutes** long. In the charts below, you can see the difference in the distribution of song duration for hit versus non-hit ("dud") songs.





Another important characteristic which seemed to differentiate hit versus dud songs was lyric sentiment. Our hypothesis was that songs with more positive lyrics would be more popular, and the data confirmed this hypothesis. Lyric sentiment is measured on a scale of -1.0 to 1.0, with the lower numbers denoting a more negative sentiment and higher numbers denoting a more positive sentiment. The lyrics of songs that made the Billboard Top 100 had a sentiment score of **0.31** on average, whereas the lyrics of songs that did not make the Top 100 had an average sentiment score of **0.19**. This result seems to make sense, as we expect people to be more attracted to happy or energetic songs and less inclined to listen to depressing or angry songs.

Loudness is another interesting characteristic which appears to be correlated with hit songs; hit songs are on average louder than non-hit songs. This correlation may explain why there appears to be an increase in loudness over time, as more artists follow the lead of louder hit songs:



Overall, we believe there is a lot of area for improvement in predicting song popularity. While each of the features we discussed may individually contribute to musical success, there are complicated interactions between all of these features which produce the end result. Furthermore, although lyric characteristics appear to influence song popularity, lyrics are likely more influential for certain genres and cannot be presumed to be a strong influencer of song success for all types of songs.

Works Cited

Nasreldin, Mohamed, et al. “Song Popularity Predictor.” *Medium*, Towards Data Science, 5 May 2018, towardsdatascience.com/song-popularity-predictor-1ef69735e380.