Predicting Song Popularity

Katherine Lin
Computer Science
Northwestern University
Evanston, IL 60201
katherinelin2016@u.northwestern.edu

Rudolf Newman
Computer Science
Northwestern University
Evanston, IL 60201
rudolfnewman2017@u.northwestern.edu

Motivation

Our project aims to generate an accurate predictor of a song's popularity. A reliable hit predictor could be invaluable to several key players in the music industry. Record companies wishing to discover new talent would be able to do so without hearing artists in person. Singers themselves would learn what kinds of songs are most likely to project them into stardom. Songwriters would know how to tailor songs to maximize popularity.

It is also interesting and worthwhile from a cultural standpoint to analyze the ways in which factors contributing to a song's popularity have varied over time. It is our secondary aim to gain insight into whether the nature of popular songs has changed over the decades.

Testing and Training

We used a subset of the Million Song Dataset (MSD), a collection of audio features and metadata of popular songs produced by The Echo Nest and The Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) at Columbia University. The subset included the following seven attributes taken directly from the MSD:

- Key (key the song is in, nominal)
- Loudness (in decibels, numeric)
- Mode (1 for major, 0 for minor, nominal)
- Tempo (in beats per minute, numeric)
- Year (release year, numeric)
- Time signature (number of beats per bar, nominal)
- Song hotttnesss (algorithmic popularity estimation by The Echo Nest, numeric)

These attributes were chosen based on their direct relevance to song categorization as well as their completeness in the existing dataset. Song hotttnesss, the attribute we wanted to predict, was given on a continuous scale from 0 to 1, with 1 being the most popular.

The dataset used consisted of 22,592 known instances. We split this by putting 80% of the dataset into a training set and the remaining 20% into a test set. This resulted in a training set consisting of 18,074 instances and a test set of 4,518 instances.

In Weka, we tried a number of different classifiers with their default settings. The number we tried was limited due to "song popularity" being a numeric attribute. Figure 1 displays the performance of the classifiers we tried. We defined a classifier's success as minimizing the root mean squared error of song popularity predictions, which have a value between 0 and 1, inclusive. We used ZeroR as a baseline for comparison. It returned a root mean squared error of 0.2475 for the training set and 0.2461 for the test set.

Some classifiers, especially the regression type, did not perform well at all. For K-nearest neighbors, even when we varied the values of k from 1 to 100, none of them performed significantly better than ZeroR. This was what we expected, given that our dataset might have been too high dimensional to work well with a straightforward implementation of nearest neighbors. In training, we used 10-fold cross validation.

We determined that the REPTree algorithm outperformed the others. It returned a root mean squared error of 0.1648 for the training set and 0.1511 for the test set using the default parameters.

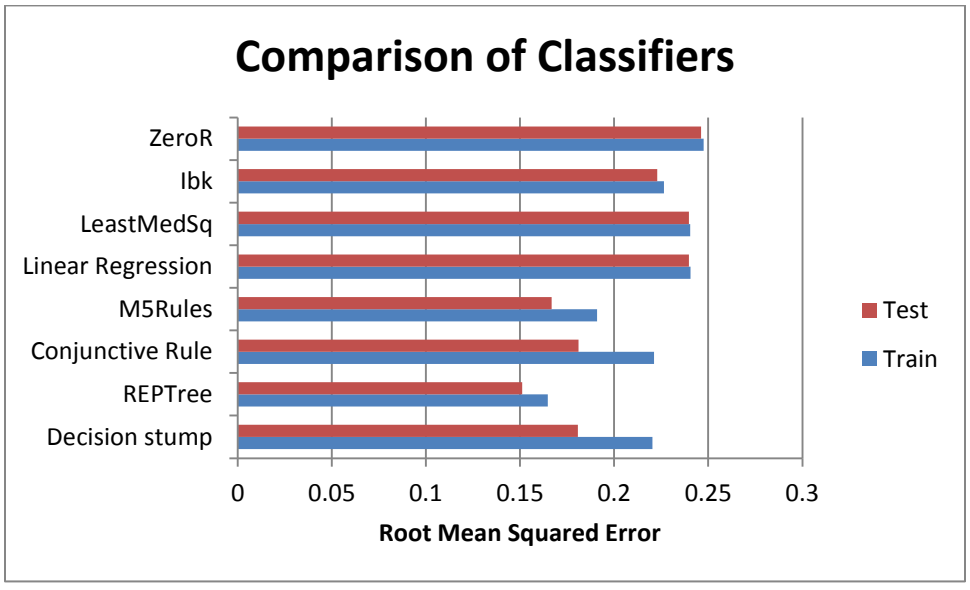

Figure 1: Performance of Different Classifiers

Solution

To predict song popularity, we created a decision tree using the REPTree algorithm in Weka. The REPTree algorithm uses regression tree logic and creates multiple trees over different iterations.[3] From these, it selects the best tree. It minimizes overfitting by using reduced error pruning. We tuned the classifier by playing around with parameters including maximum tree depth, the minimum total weight of instances in a leaf, minimum proportion of variance, and number of folds. The results are displayed in Figures 2, 3, and 4.

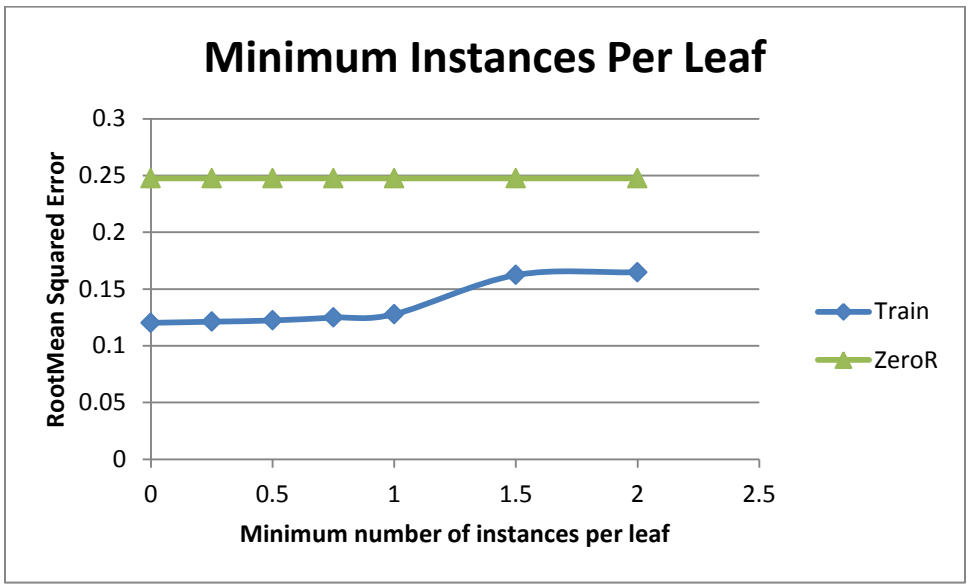

Figure 2: Minimum Number of Instances per Leaf and Error (# folds = 5; min proportion of variance = 0.001)

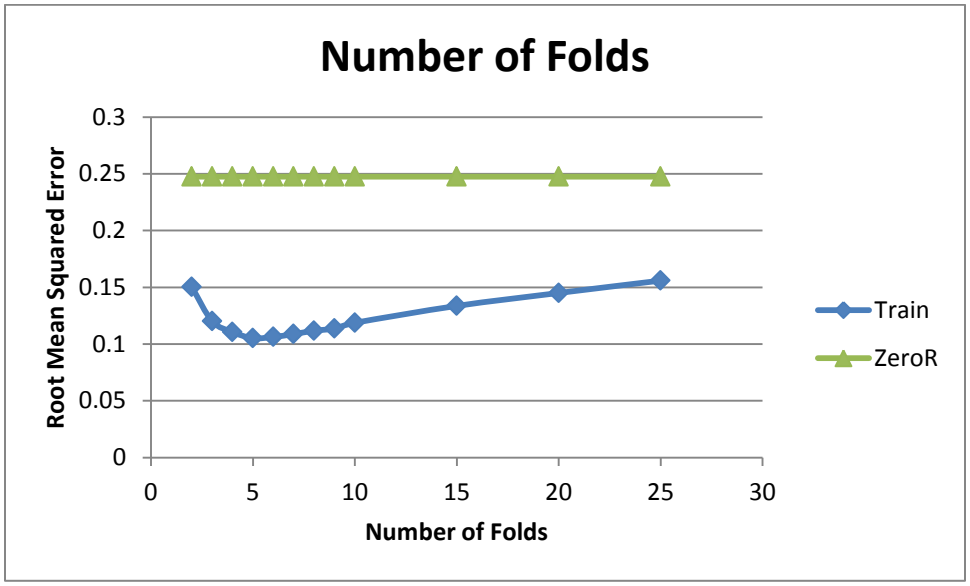

Figure 3: Number of Folds and Error (min instances = 0, min proportion of variance = 0.001)

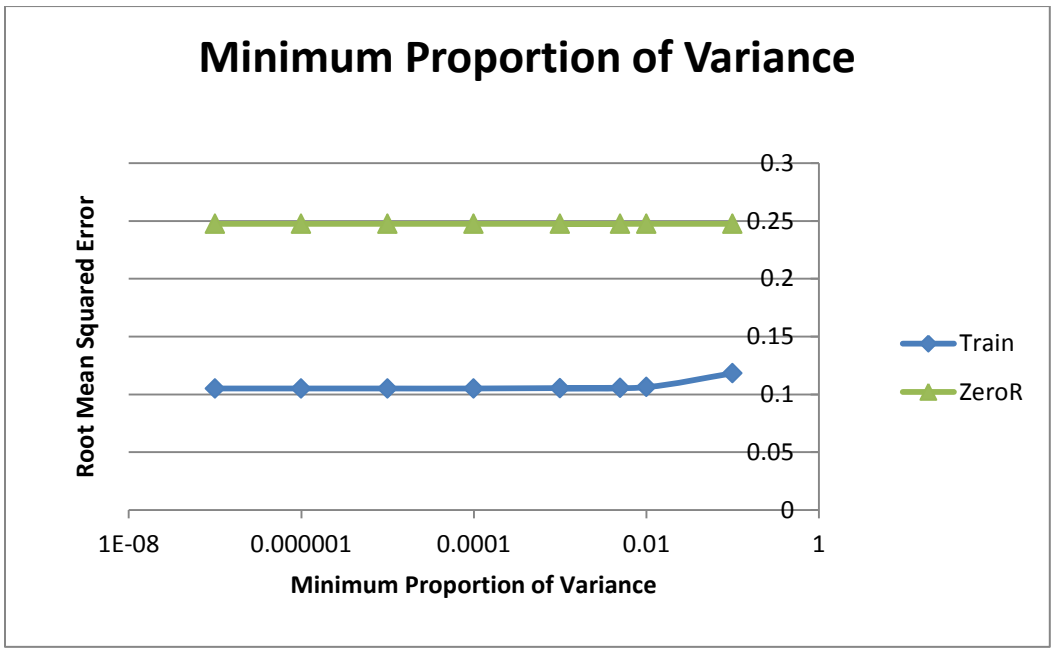

Figure 4: Minimum Proportion of Variance and Error (# folds = 5, min instances = 0)

The best combination of parameters had a minimum of zero instances per leaf, five folds for pruning, and a minimum proportion of variance of 0.00001. This resulted in a root mean squared error of 0.1051 in the training set, versus 0.2475 with ZeroR. When evaluating our test set with the model generated on the training set, we obtained a root mean squared error of 0.0693. This indicates that this model works well, performing even better on the test set than the training set, and that overfitting was not a problem. Figure 5 illustrates the performance of our model.

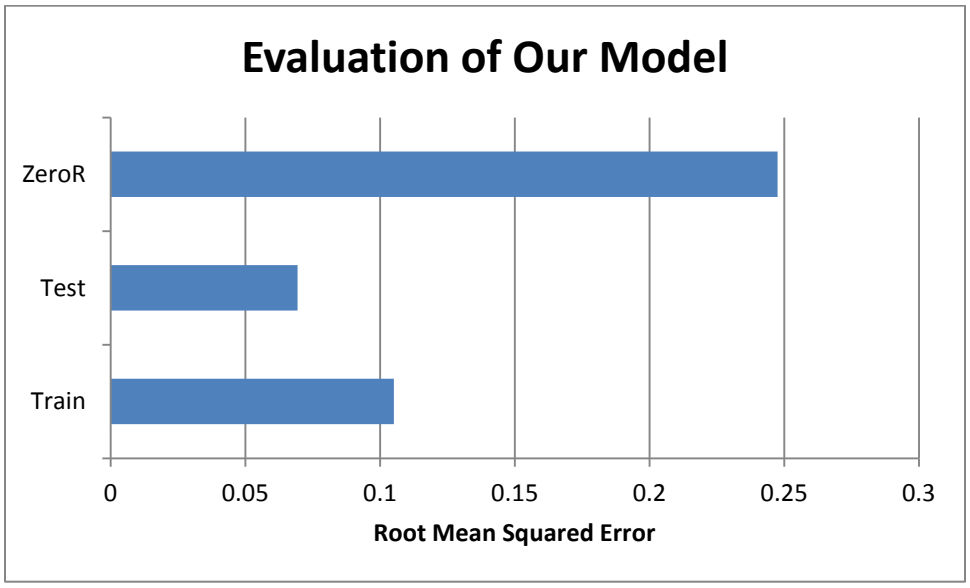

Figure 5: Evaluation of Our Model

Key Findings

Our decision tree model indicates that loudness is a major factor in determining whether a song will be popular or not. In particular, louder songs tend to be more popular than their quieter counterparts. Note that loudness here is given in decibels. Other important features include tempo. Faster songs tend to be more popular than slower ones. Mode also was important, with songs in major keys tending to be better received than those in minor keys.

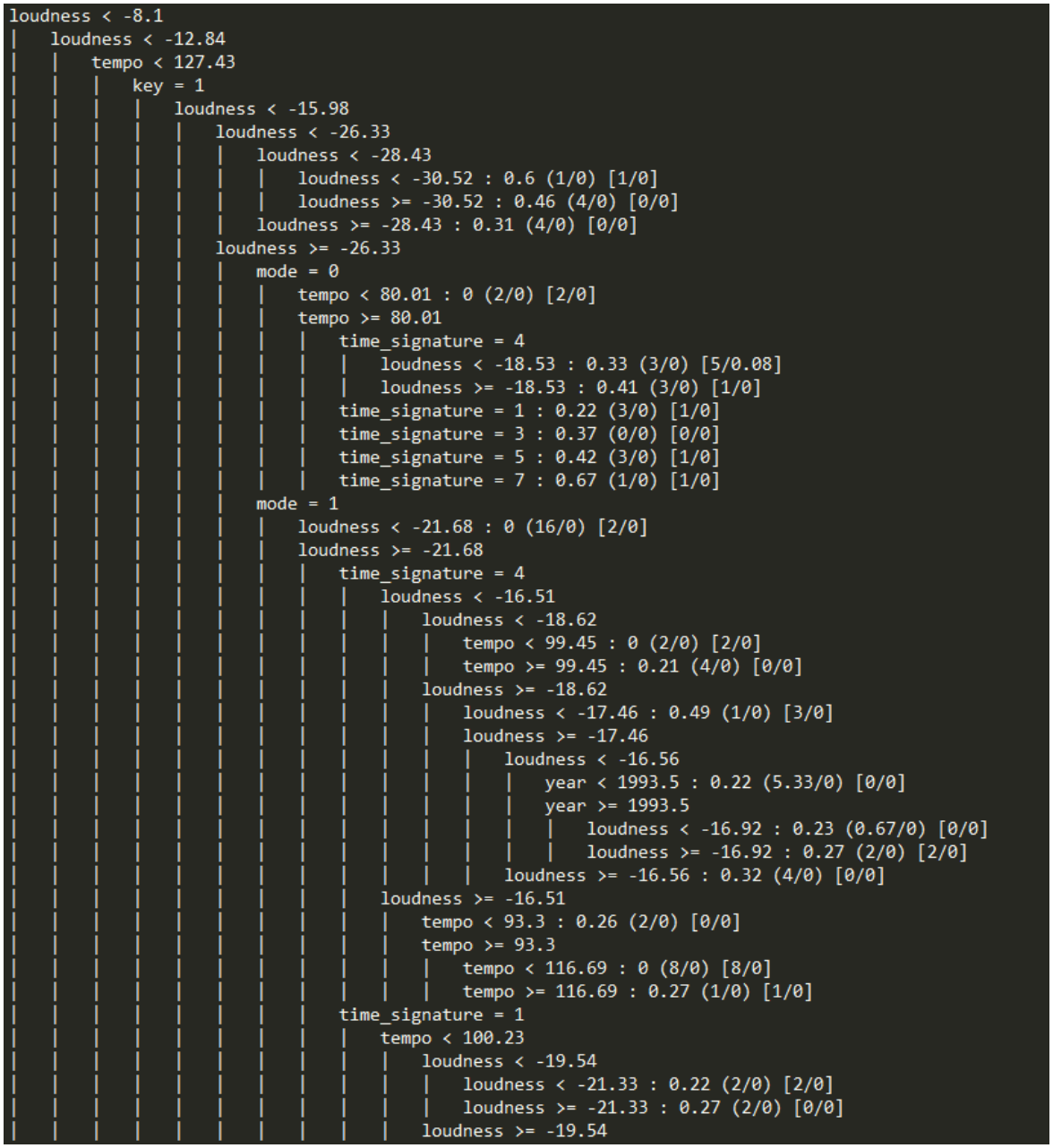

Figure 6: Snippet of Decision Tree (full tree can be found on website)

Secondary Results

Our tree suggests that in earlier years, quieter songs were more popular. Over time, louder songs became more popular. It is difficult to rigorously analyze this trend by examining our tree, but as a generalization, it is supported in many observed cases. In one instance, for example, songs before 1969 that are quieter (below -26.03 dB) are predicted to have a popularity of 0.61, whereas songs that are louder (above -26.03 dB) are predicted to have a popularity of 0 if the key is minor and 0.27 if the key is major.

Future Steps

To create a more accurate classifier, we need to address the limitations of our dataset. We initially wanted to include "danceability" and "energy" as attributes. Related work[1] has shown that danceability, especially, plays an important role in song popularity, especially in recent decades. The danceability and energy of each song, unfortunately, were not analyzed in MSD. Additionally, the vast majority of the data was for pop songs in the 2000s or later. For our secondary goal of examining changes in pop music over time, it would be good to get more data from other decades. It would be interesting to be able to directly extract attributes from the songs themselves. If we were able to do this, we could compile a larger dataset that includes any song we wish.

Discussion

Music analysis through systems such as Shazam and Spotify allows people in the pop music business to identify popular songs through big data rather than relying solely on experts[4]. This poses interesting questions regarding creativity, innovation, and artistry. While analytics may allow for better understanding of the tastes of the crowd, there is also concern that producing songs under these guidelines would affect the quality of music. There are limitations to what a dataset can express. In the end, music is human and can only be as predictable as we are.

Works Cited

1. Ni Yizhao, Raul Santos-Rodrigues, Matt Mcvicar, Tijl De Bie. Hit Song Science Once Again a Science? 4th International Workshop on Machine Learning and Music (2011).
2. Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.
3. Kalmegh, Sushilkumar. Analysis of WeKA Data Mining Algorithm REPTree, Simple Cart and Random Tree for Classification of Indian News. International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 2 (2015).
4. Thompson, Derek. The Shazam Effect. (December 2014) http://www.theatlantic.com/magazine/archive/2014/12/the-shazam-effect/382237/