Max Bauer

Lucas Mentch

STAT 1361

04/16/2024

Final Data Science Project - Technical Report

Introduction/EDA

In the music industry, understanding what drives the popularity of songs is crucial. SonicWave wants to understand what factors are important in determining this, as well as predicting popularity. As a Data Science Consultant for SonicWave, I was handed the task of analyzing data on rock, jazz, and pop songs to predict their popularity utilizing a dataset containing popularity metrics and song attributes. The dataset contains nineteen variables: id, album_name, track_name, popularity, duration_ms, explicit, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature, track_genre. Our target variable, popularity, is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by an algorithm and is based, for the most part, on the total number of plays the track has had and how recent those plays are.

Looking at the data, there were no missing values, however, I noticed some oddities that may affect the accuracy in determining the popularity. The first concern was that very popular holiday songs had a popularity ranking of zero, in particular, Christmas songs. Almost all, two hundred ninety-seven songs that had Christmas in the album or track name received a score of zero, at most a score of two. For example, some classics that received this ranking were: “Baby, it’s Cold Outside”, “Frosty the Snowman”, and “Rudolph The Red-Nosed Reindeer”. This anomaly could be due to the way popularity was calculated, which considered how recent the plays were. Another oddity I noticed, was that some popular artists earned a popularity ranking of zero. One example of this is Taylor Swift. Her getting a zero for any of her songs is hard to believe due to her huge fanbase. Ultimately, I decided to do nothing about these oddities and concluded it was poor data collection. The only changes I made to the data were deleting the album and track name, and coding the track genre category. It was split into jazz, rock, and pop, so I made the column of track genres 0, 1, and 2, respectively.

When looking at the summary statistics for our target variable popularity, the data was positively skewed, as the mean, 27.38, was higher than the median, 1.00. I decided to try a square root transformation because our data contained mostly zeros. The skewness improved from .5749 to .4121, but the mean, 3.49, was still higher than the median, 1.00. I continued to investigate the difference between the untransformed and transformed target variables. I plotted each variable against the target variables, and it wasn't a huge difference, so I continued to test the transformed variable along with the untransformed one. As described in the next section, upon looking at the linear models I decided to use the square root transformation.

Methods Overview/Details

Before looking at the linear models, I split the training and testing dataset into two sets. The training set was split into an X_train set, with every variable besides our target variable, and a Y_train/Y_train_sqrt set, with the variable popularity untransformed/transformed. The testing set was split identically but named X_test and Y_train/Y_train_sqrt.

Then, I looked at the linear models of the target variables, with all predictors included. Both models had the same eight significant predictors: duration_ms, danceability, mode, speechiness, acousticness, valence, tempo, and track_genre. The model with the untransformed target variable had an R-squared of .2185, with an adjusted R-squared of .2043, and an F-statistic of 15.36 on 15 and 824 degrees of freedom. On the other hand, the model with the transformed target variable improved very slightly, with an R-squared of .222, an adjusted R-squared of .2078, and an F-statistic of 15.67 on 15 and 824 degrees of freedom. Even though it was very small, it was still an improvement, so I decided to continue using the transformed popularity data.

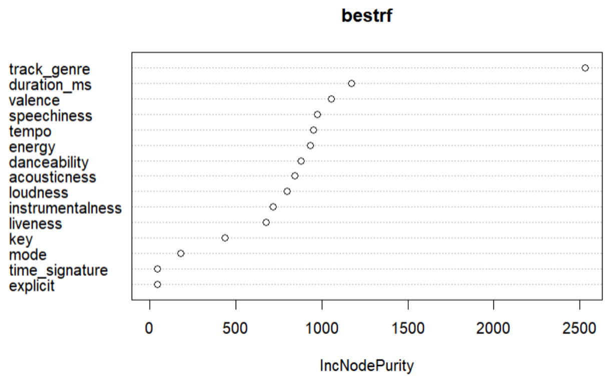
I then made a model only containing the significant predictors and compared their MSEs. I decided to use MSE to help determine the best model because our goal is to predict. When compared to the model with all predictors, the MSE of the model only containing significant predictors increased from 12.547 to 12.648. After this, I tried some automated methods.

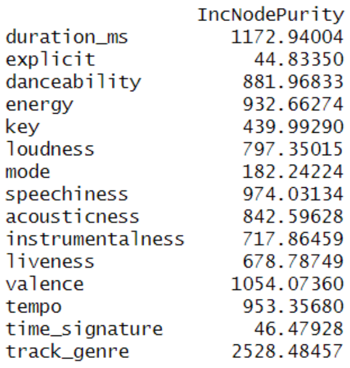
Firstly, I tried ridge regression and lasso, with the tuning parameter chosen by cross-validation. The ridge regression had a slightly improved MSE of 12.399, and the lasso's MSE also slightly improved to 12.167. I then constructed a decision tree model using the training dataset(s) and estimated the MSE on the test set(s). I got an MSE of 12.558, and also tried pruning, which slightly improved the MSE to 12.385.

Next, I discovered a huge improvement in MSE when trying random bagging and random forest. The bagging technique improved the validation set MSE to 9.538. Following this, I identified the optimal 'mtry' for the random forest model which was fourteen. This reduced the MSE to 9.498. I also tried boosting, which also had an improvement, but not as greatly as bagging and random forest, with an MSE of 11.501.

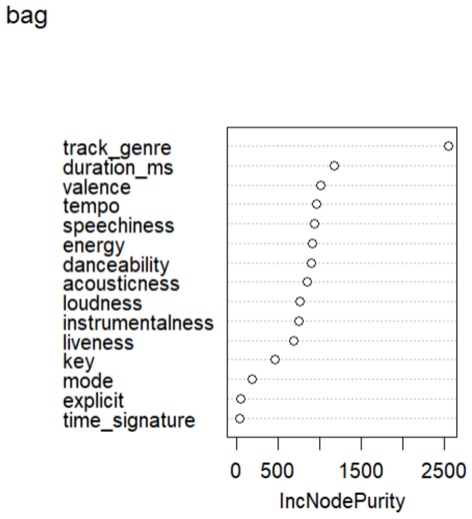
Summary of Results

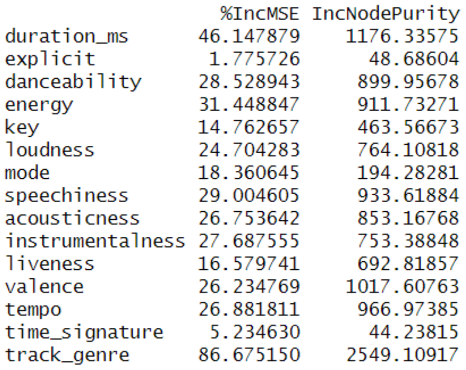
In terms of MSE, the best models were bagging and random forest, with the random forest being the best. Based on the increase in node purity, the variable track_genre was by far the most significant for both models, with a value of 2528.48 for random forest and 2549.11 for bagging. There were only two minor differences between the rankings of significant variables, the first being time_signature and explicit being swapped for the least significant variable. The second was bagging gave slightly more significance to tempo rather than speechiness, unlike random forest. Below is a graph and table of the most important variables for both models:





```
                 IncNodePurity
duration_ms         1172.94004
explicit              44.83350
danceability         881.96833
energy               932.66274
key                  439.99290
loudness             797.35015
mode                 182.24224
speechiness          974.03134
acousticness         842.59628
instrumentalness     717.86459
liveness             678.78749
valence             1054.07360
tempo                953.35680
time_signature        46.47928
track_genre         2528.48457
```





```
                   %IncMSE IncNodePurity
duration_ms      46.147879    1176.33575
explicit          1.775726      48.68604
danceability     28.528943     899.95678
energy           31.448847     911.73271
key              14.762657     463.56673
loudness         24.704283     764.10818
mode             18.360645     194.28281
speechiness      29.004605     933.61884
acousticness     26.753642     853.16768
instrumentalness 27.687555     753.38848
liveness         16.579741     692.81857
valence          26.234769    1017.60763
tempo            26.881811     966.97385
time_signature    5.234630      44.23815
track_genre      86.675150    2549.10917
```

Conclusion/Takeaways

Predicting song popularity across jazz, rock, and pop genres requires an understanding of various factors. Ensemble techniques like bagging and random forest outperformed traditional models. However, data anomalies, particularly regarding holiday songs and popular artists, need to be addressed. Future research should focus on refining data collection and exploring advanced modeling approaches to better capture the complex dynamics driving song popularity in the music industry.