Modeling Song Popularity with Spotify

Abstract

In this research paper, we investigate the relationship between different audio features, such as danceability, loudness, and tempo, and artist features of a song with the song's popularity on Spotify and its probability of ending up in *Billboard Top 100*. We use model selection (forward, backward, forward-backward selection) to determine which factors are most important in determining the song's popularity rate. We also apply a logistic regression model and compare it with a classification tree to evaluate our model's accuracy. Through the use of different models, we were also able to evaluate different trends in our dataset and identify the optimal values for different audio features.

Background

In 2020, the music industry generated $23.1 billion, and streaming made up 56% of the total revenue. Because of the increasing profit generated by the music industry and growing streaming services, artists and record labels have looked to maximize their chances of making their songs popular. The research question we're addressing here is: what are the most important factors that make a hit song and given these features, what is the probability that a song will end up in the *Billboard Hot 100*?

Data Collection and Cleaning

The data cleaning took the vast majority of the work. We realized data cleaning is essential to identify and remove errors and duplicates. This improves the quality of the training data for future analytics and helps increase our accuracy.

We started by looking at the dataset we found in TidyTuesday. The billboard dataset had records of weekly *Billboard Hot 100* going back to the 1950s. The songs features dataset included all the songs on the billboard dataset that are also on Spotify. It also included a host of song features: *danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature, spotify_track_popularity, and spotify_track_explicit*.

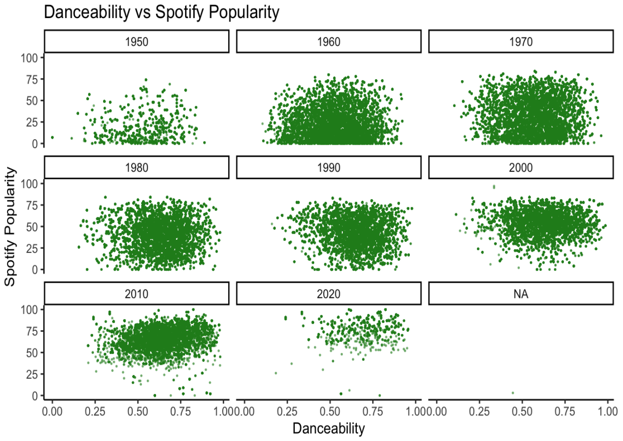
The Songs Feature dataset included a lot of NAs for many fields, which we confirmed were because of the lack of information about the songs in Spotify. We omitted those entries and moved on to create a songs features list of songs that didn't make it to the *Billboard Hot 100* list. We found an existing playlist on Spotify that had up to 5295 songs.

In Appendix 1, we talk further about utilizing Spotify API to extract song features for this data.

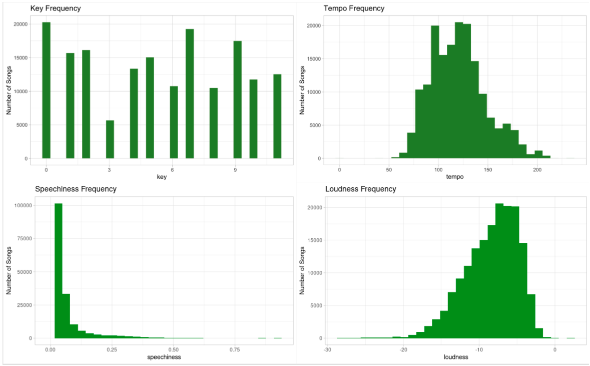
After binding the two datasets to make datasets of all the songs, we then moved on to create two smaller datasets for training and testing purposes. This required us to split the dataset into two randomly chosen training and test datasets.

Data Exploration

After collecting and cleaning the data, we then moved on to data exploration by looking into trends in our dataset.



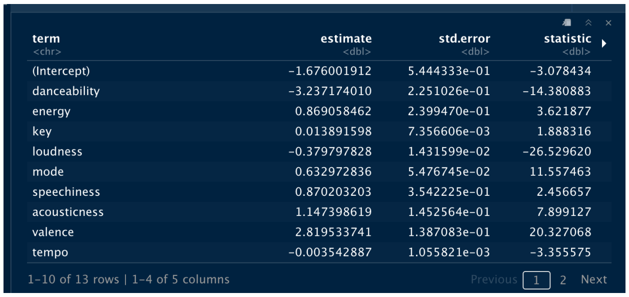
This demonstrates how music trends change over years from 1950 to 2020, specifically with the *danceability* factor. The y-axis is "Spotify Popularity", which according to Spotify, is based on "the total number of plays the track has had and how recent those plays are" (Pecker). The value will be between 0 to 100, with 100 being the most popular. The 2020 graph shows how higher danceable songs have higher *Spotify popularity,* possibly because of platforms such as TikTok and Instagram. These days, danceable songs mean more people listen and dance to it, and are thus more popular.



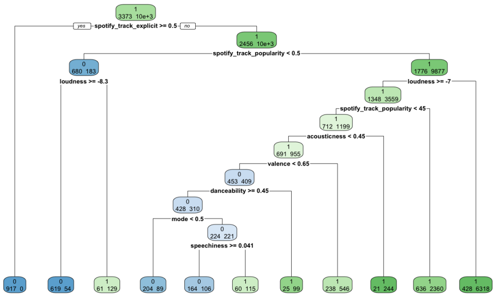
We were also interested in the distribution of hit songs based on other features. The graphs show the most common features of *tempo, speechiness, loudness*, and *key* of songs that ended up in *Billboard*, which is the y-axis. We can see that for *tempo*, there were two peaks within this range at about 100 bpm (beats per minute) and 135 bpm. We also discovered that most of the songs that end up in Billboard have less speechiness, while most songs have greater loudness.

Audio Feature	Mean	Description
Tempo	120.28	beats per minute (BPM)
Loudness	8.405	overall loudness of a track in decibels (dB)
Speechiness	0.0686	presence of spoken words in a track
Key	5.282	estimated overall key of the track (ex: 0 = C, 1 = C♯/D♭, 2 = D)

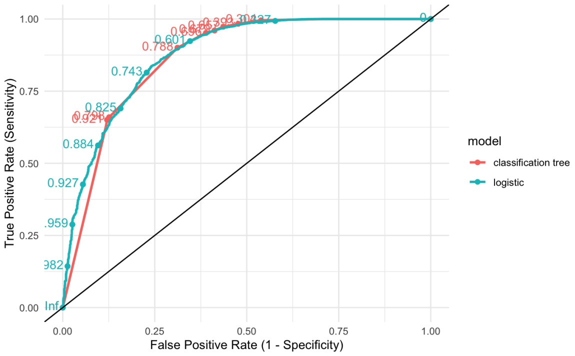
For the songs that made Billboard's Top 100, we looked into the mean values for the audio features we detected previously using and the results were fairly reasonable. *Tempo* was at about 120.28 bpm, *loudness* was at 8.405, *speechiness* was at 0.0686, and the estimated overall *key* was 5. This shows most songs on Billboard are fast paced, loud, and are in major key.


```
term                 estimate      std.error      statistic
<chr>                   <dbl>          <dbl>          <dbl>
(Intercept)      -1.676001912   5.444333e-01      -3.078434
danceability     -3.237174010   2.251026e-01     -14.380883
energy            0.869058462   2.399470e-01       3.621877
key               0.013891598   7.356606e-03       1.888316
loudness         -0.379797828   1.431599e-02     -26.529620
mode              0.632972836   5.476745e-02      11.557463
speechiness       0.870203203   3.542225e-01       2.456657
acousticness      1.147398619   1.452564e-01       7.899127
valence           2.819533741   1.387083e-01      20.327068
tempo            -0.003542887   1.055821e-03      -3.355575
1-10 of 13 rows | 1-4 of 5 columns              Previous  1  2  Next
```

We also conducted a variable selection process using forward selection, backward elimination, and forward-backward selection. We used our train dataset to achieve this. All 3 models eliminate *instrumentalness* and *liveness*, and only include the following audio features: *loudness, Spotify track popularity, Spotify track explicit, valence, danceability, mode, acoustiness, tempo, speechiness, key, time signature, energy*. Using these variables, we built a logistic regression model. The estimated coefficients of the logistic regression model show that there is a positive correlation between loudness and the outcome (overall loudness of a track are in decibels (dB) that typically range between -60 and 0 db). All of the p-values are less than 0.05, which implies that the given audio features are statistically significant.



We compared the logistic regression model with a classification tree. The decision tree reveals to us the variables deemed important by the classification algorithm. It shows that the most important variables in determining whether a song is going to end up on the billboard are *spotify_track_explicit* and *spotify_track_popularity*. We can infer that most of the songs on the *Billboard* are popular on Spotify and they are not explicit. All other observations are distributed among other leaves, which use *loudness*, *acousticness, valence, danceability, mode, and speechiness* to predict whether a song ended up in *Billboard Hot 100* or not.



We trained our two models, classification tree and logistic regression model, with our training dataset and then we created the ROC curves using the test dataset. The ROC curve shown on the left shows that the logistic regression model is a better model for predicting the likelihood of a song ending up. Our model shows that it correctly predicts the songs ending up in *Billboard* 96% of the time.

Discussion

In summation, we were able to produce a ranked list of 12 audio features which are the most important predictors of what makes a song popular. Using these features, we can see that the logistic regression model is the best model for predicting the likelihood of a song ending up in the *Billboard Hot 100*. It is important to note, however, that the logistic regression model does not reach 96.7% true positive rate until 43.2% false positive rate as we can see in the ROC curve. This is likely because of the limitations we discuss next.

The biggest critique of our model is that it fails to use an artist's popularity as a predictor of whether a song will end up in the *Billboard Hot 100*. An artist's popularity is potentially an essential predictor because more famous artists often have more famous songs. For instance, if an amateur musician puts out a song and a more established musician puts out a song with the exact same audio features, our model will wrongly predict that both songs have an equal likelihood of ending up in the *Billboard Hot 100*. We can develop a measure of how popular an artist is by looking at how many streams their songs have had in the past or how many of their songs have ended up in the *Billboard Hot 100* in the past. This will help us differentiate amateur musicians from more popular musicians.

Secondly, as we saw in the Data Exploration section, trends in what makes a song popular change over time but our models fail to account for these changes. At the moment, our model grants equal weights to old and new songs and therefore assumes that audio features of old songs are equally good predictors of what makes a song popular as of newer songs. In order to fix this, we can weigh newer songs more heavily in the data set so that our model accounts for recent trends in music tastes. Another way to do this would be to have more new songs in the data set as compared to old songs so that the prediction is driven more heavily by recent trends.

References

Ketola, Susanna. “Biggest Playlist Ever.” Spotify, 2017. open.spotify.com/playlist/4rnleEAOdmFAbRcNCgZMpY.

Lallier, Oscar. “The Longest Playlist Ever.” Spotify, 2017. open.spotify.com/playlist/6yPiKpy7evrwvZodByKvM9.

Peker, Philip. “Predicting Popularity on Spotify - When Data Needs Culture More than Culture Needs Data.” *Medium*, Towards Data Science, 17 Nov. 2021, towardsdatascience.com/predicting-popularity-on-spotify-when-data-needs-culture-more-than-culture-needs-data-2ed3661f75f1#:~

Published by Statista Research Department.“Recorded Music Industry - Global Revenue 2020.” *Statista*, 14 July 2021, www.statista.com/statistics/272305/global-revenue-of-the-music-industry/

Rfordatascience. “Tidytuesday/Data/2021/2021-09-14 at Master · Rfordatascience/Tidytuesday.” GitHub, github.com/rfordatascience/tidytuesday/tree/master/data/2021/2021-09-14

Appendix

Appendix 1

The Spotify API and all the libraries associated with it will only let us extract the first 100 songs from any playlist. We used Python's library "Spotipy" to extract all songs from the playlist.

These songs are songs by popular artists and had songs that actually made it to the *Billboard Hot 100*. After extracting these songs, we had to filter out all songs that were on the *Billboard* so we could work with songs that didn't make the cut. Then, we add all the songs features, and we bind the list with the dataset we created with Billboard and Songs_Features dataset.

Before binding the two datasets, however, we had to make sure to label the songs that didn't make it to the *Billboard* as 0 and the songs that made it to the *Billboard* as 1. We added a new variable called in_billboard with the respective values inserted.

Appendix 2

Figure 1: Danceability vs. Frequency

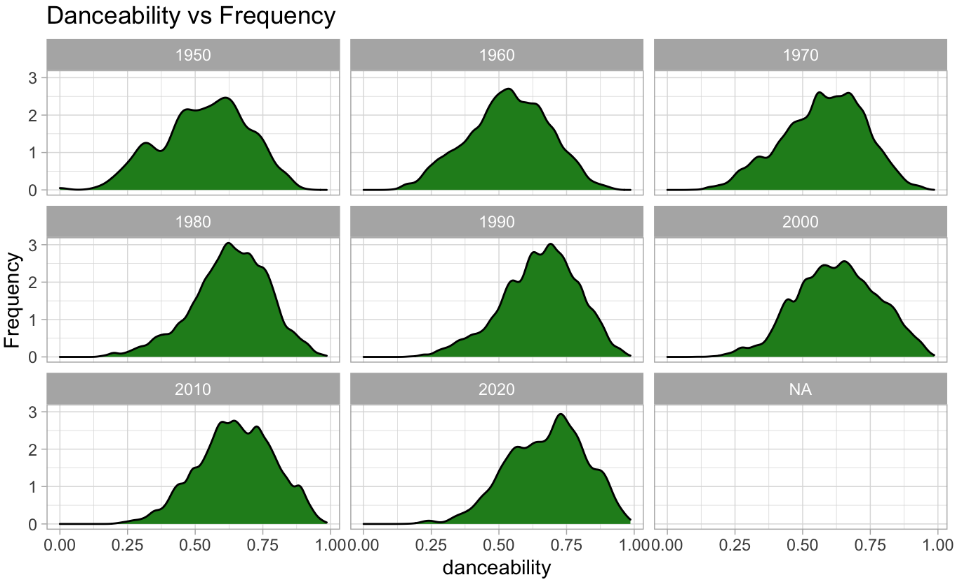
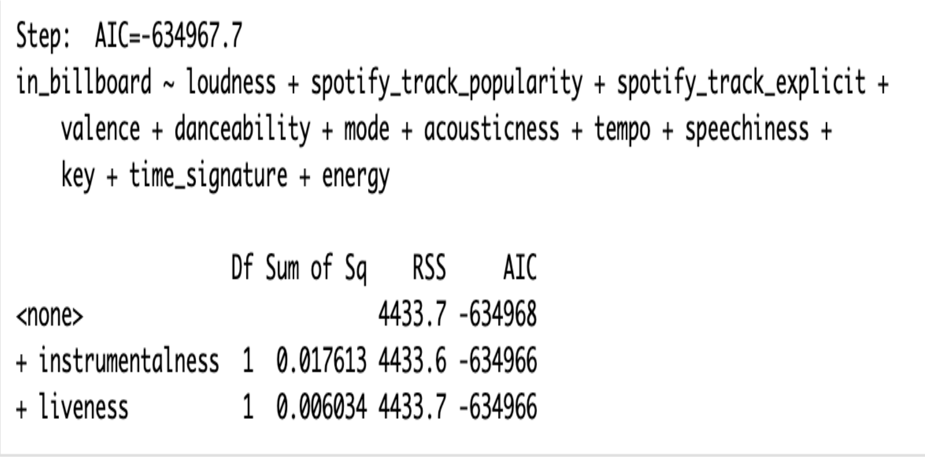


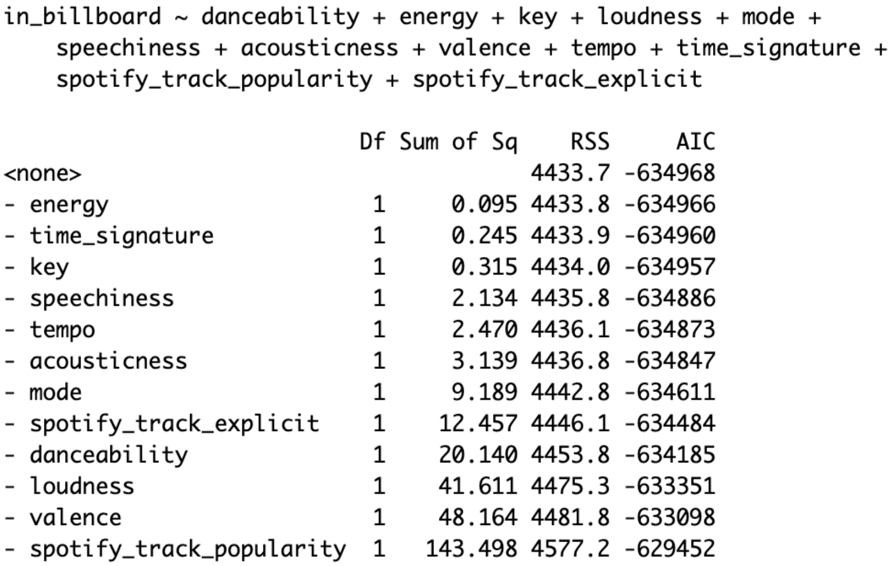
Figure 2: Forward Selection



```
Step:  AIC=-634967.7
in_billboard ~ loudness + spotify_track_popularity + spotify_track_explicit +
    valence + danceability + mode + acousticness + tempo + speechiness +
    key + time_signature + energy

                  Df Sum of Sq    RSS     AIC
<none>                         4433.7 -634968
+ instrumentalness  1  0.017613 4433.6 -634966
+ liveness          1  0.006034 4433.7 -634966
```

Figure 3: Backward Selection



```
in_billboard ~ danceability + energy + key + loudness + mode +
    speechiness + acousticness + valence + tempo + time_signature +
    spotify_track_popularity + spotify_track_explicit

                           Df Sum of Sq    RSS     AIC
<none>                                  4433.7 -634968
- energy                    1     0.095 4433.8 -634966
- time_signature            1     0.245 4433.9 -634960
- key                       1     0.315 4434.0 -634957
- speechiness               1     2.134 4435.8 -634886
- tempo                     1     2.470 4436.1 -634873
- acousticness              1     3.139 4436.8 -634847
- mode                      1     9.189 4442.8 -634611
- spotify_track_explicit    1    12.457 4446.1 -634484
- danceability              1    20.140 4453.8 -634185
- loudness                  1    41.611 4475.3 -633351
- valence                   1    48.164 4481.8 -633098
- spotify_track_popularity  1   143.498 4577.2 -629452
```

Figure 4: Forward-Backward Selection

```
Step:  AIC=-634967.7
in_billboard ~ loudness + spotify_track_popularity + spotify_track_explicit +
    valence + danceability + mode + acousticness + tempo + speechiness +
    key + time_signature + energy

                           Df Sum of Sq    RSS     AIC
<none>                                  4433.7 -634968
+ instrumentalness          1     0.018 4433.6 -634966
- energy                    1     0.095 4433.8 -634966
+ liveness                  1     0.006 4433.7 -634966
- time_signature            1     0.245 4433.9 -634960
- key                       1     0.315 4434.0 -634957
- speechiness               1     2.134 4435.8 -634886
- tempo                     1     2.470 4436.1 -634873
- acousticness              1     3.139 4436.8 -634847
- mode                      1     9.189 4442.8 -634611
- spotify_track_explicit    1    12.457 4446.1 -634484
- danceability              1    20.140 4453.8 -634185
- loudness                  1    41.611 4475.3 -633351
- valence                   1    48.164 4481.8 -633098
- spotify_track_popularity  1   143.498 4577.2 -629452
```