How to Make a Hit Song

Using Regression to Predict Song Popularity

Timothy Regis

12/9/2020

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(boot)
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

Keywords

Generalized Linear Regression, Musical Attributes, Spotify Music Catalog, Song Popularity

Abstract

Making a hit song can often seem like an impossible task to the average person as the components of a final composition can be extremely complex. In an attempt to uncover these secrets, this paper applies a generalized regression model to a dataset of Spotify's music catalog, and analyzes the effects of different

musical characteristics on the song's popularity. The findings of this analysis suggest that the danceability, valence, and liveness of a song are all strong predictors of the song's popularity, with danceability showing a significant positive correlation and the latter two suggesting negative returns to popularity. While this cannot be considered a 'golden rule' for songmaking, these results can help interested musicians incorporate new factors into their music to better appeal to wider audiences.

Introduction

The musical industry of today's world is incredibly complex. Differing tastes have given rise to a seemingly endless pool of genres to choose from. This has made it increasingly difficult to uncover just what exactly goes into making a hit song that resonates with the greatest number of listeners. At times, it seems like only a small and exclusive group holds the secret to this as only a select few appear to top the charts on a regular basis. It can easily be assumed that this is a pure fame game, that is, the most famous and flashy artists garner the largest listening bases and thus will take the lead in a popularity contest. However, this assumption ignores both what got the artist to this point, and why people are still listening.

Whether it be the heaviest of heavy metal, the lightest pieces of classical music, or anything in between, a song can take off in popularity for a number of different reasons. To gain a more thorough and logical understanding of this concept, Spotify has computed numerical values for a wide variety of musical factors that can be studied and compared to a song's popularity. This includes things like a song's; valence, energy, danceability, and many more specific traits. All of these characteristics result in songs that either gain massive popularity or succumb to total irrelevancy, never to be streamed again.

In this paper, this exact process was carried out. Using a generalized linear regression model, I conducted an analysis on the effects of a song's musical traits on its overall popularity to be able to provide a rudimentary framework for creating a 'hit' song. To accomplish this, I have made use of a large dataset containing data on over 170,000 songs on Spotify's catalog. The regression model was then applied to this dataset using its popularity measure as the dependent variable and a variety of characteristics as independent predictor variables.

The results of this analysis suggest that a song's popularity is strongly influenced by its danceability, with more danceable songs showing much higher popularity. Furthermore, a song's valence and liveness are both strong predictors of its popularity as well, however, these qualities appear to be negatively correlated with the song's popularity.

An impoortant area not covered by this analysis is the use of advertising space across Spotify's platforms. On all versions, Spotify picks a few artists or specific songs to advertise on a user's homepage. This makes it difficult to find a definitive answer as these advertisments can pull in very large groups of listeners and sway the popularity in the artist's favor. Moreover, based on a user's listening history, Spotify will often re-suggest these songs or others made by the same or similar artists providing the idea that a song's popularity may not be a case of linear growth due to a suggestion bias. While this makes it more difficult to uncover the truth behind what makes a song most popular, the results that have been generated provide a good initial idea of the most important factors in a song's overall popularity. As such, these findings can be taken advantage of by future artists that hope to give their music a slight edge over their competition.

To begin, Section 2 provides an overview of the dataset used for the analysis and the according modifications that were required. Section 3 lays out the development of the regression model used for this paper along with a brief explanation on the chosen model. In Section 4, the results of this model analysis are displayed along with their interpretations and a validation process is performed on the model. Section 5 features a discussion of these results and the conclusion that these results bring us to. Additionally, in this section, the weaknesses of this analysis are discussed as well as the areas for improvements and future work. Finally, Section 6 is an Appendix section featuring the code used for this analysis, and Section 7 provides references for all sources used for this paper.

2 Data Discussion

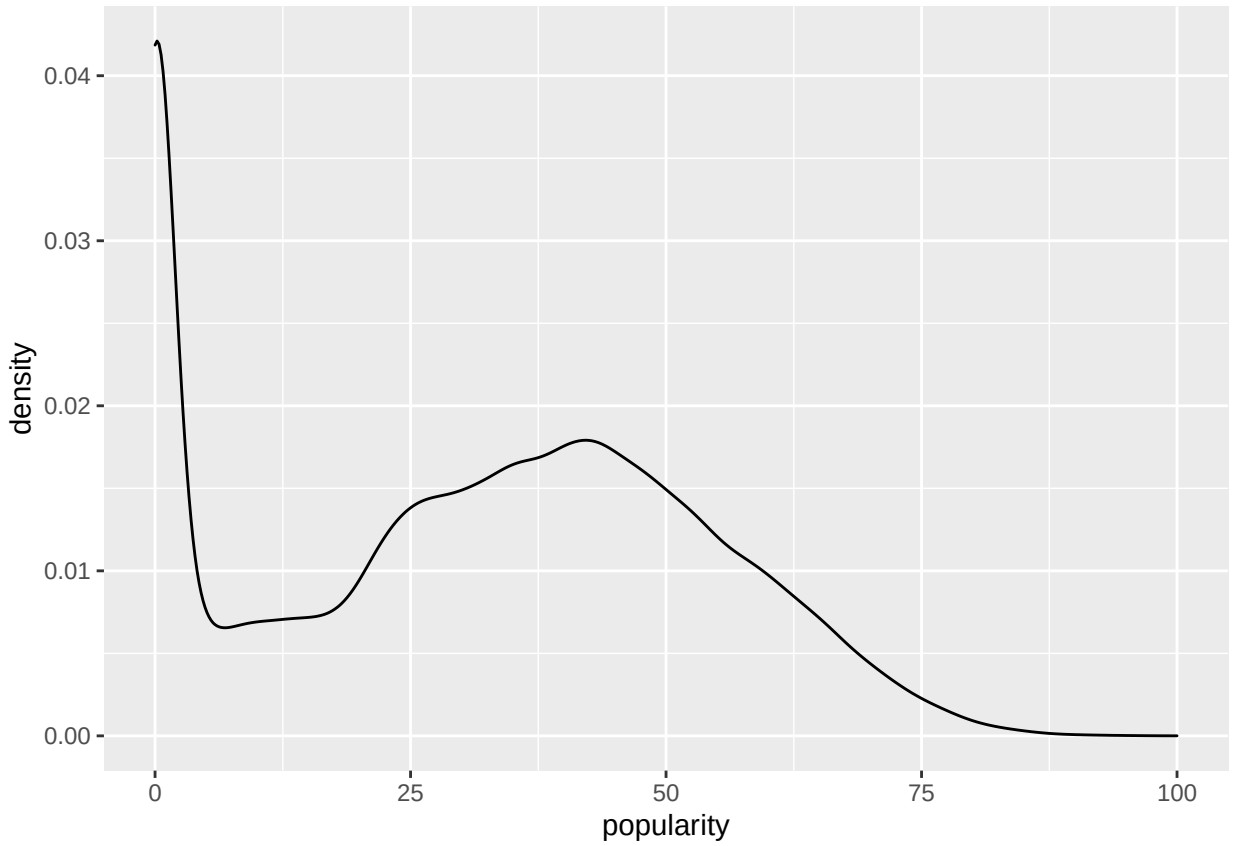
For the purposes of this analysis, one extremely large dataset containing information on over 170,000 songs in Spotify's catalog was used. This data was gathered using Spotify's Web API software and was set up and organized by a user on Kaggle.

```
rawdata <- read_csv("spotifydata.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   valence = col_double(),
##   year = col_double(),
##   acousticness = col_double(),
##   artists = col_character(),
##   danceability = col_double(),
##   duration_ms = col_double(),
##   energy = col_double(),
##   explicit = col_double(),
##   id = col_character(),
##   instrumentalness = col_double(),
##   key = col_double(),
##   liveness = col_double(),
##   loudness = col_double(),
##   mode = col_double(),
##   name = col_character(),
##   popularity = col_double(),
##   release_date = col_character(),
##   speechiness = col_double(),
##   tempo = col_double()
## )
```

```
data <- rawdata %>%
  dplyr::select(artists,
                name,
                popularity,
                danceability,
                valence,
                liveness)
```

```
data %>%
  ggplot(aes(x = popularity)) +
  geom_density()
```



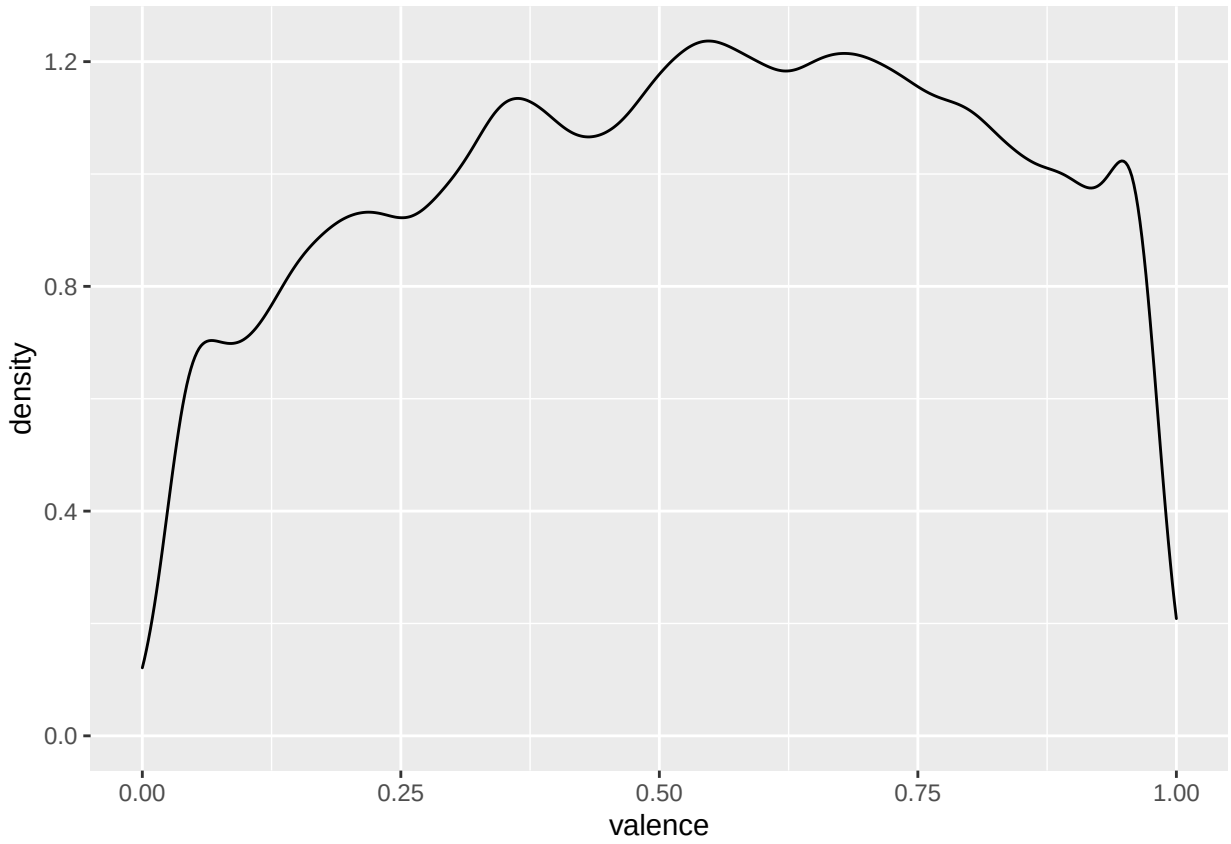

```
mean(data$popularity)
```

```
## [1] 31.43179
```

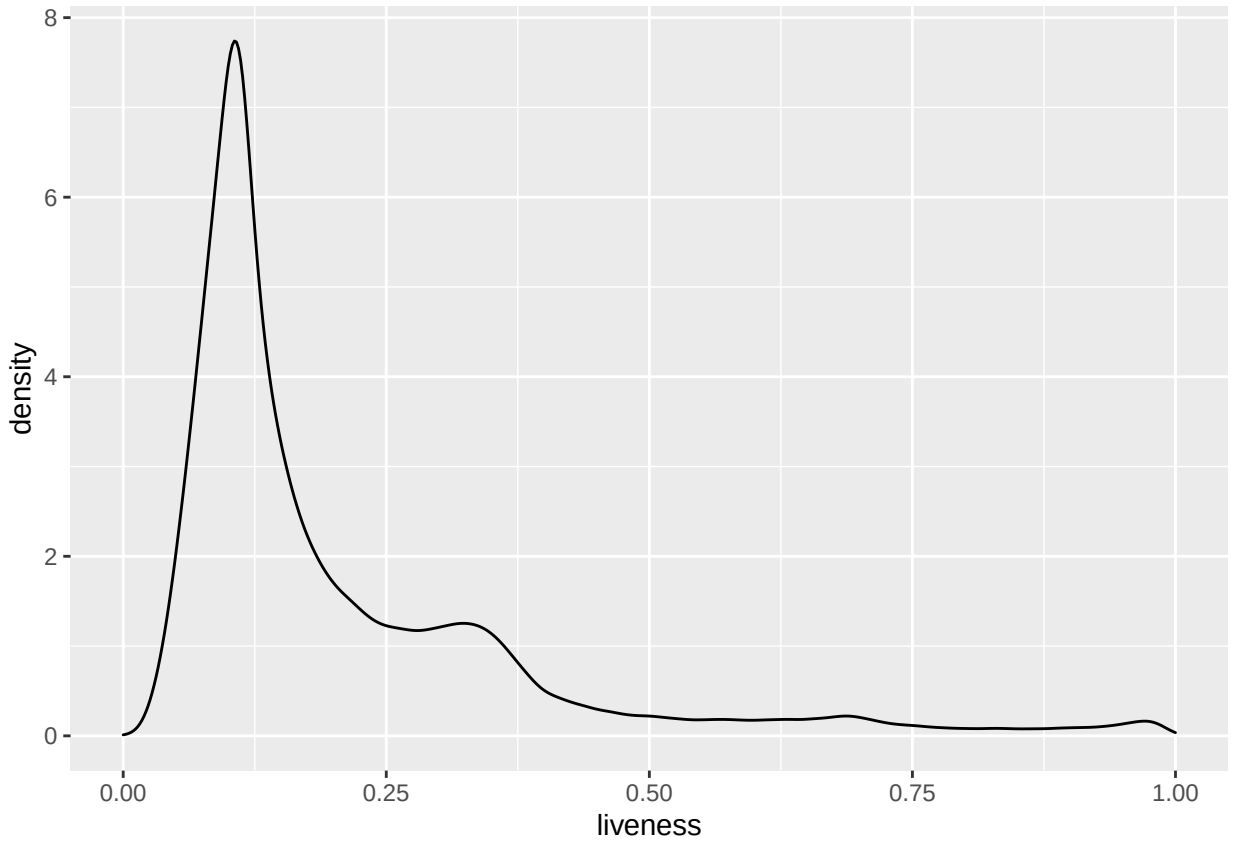
```
sd(data$popularity)
```

```
## [1] 21.82662
```

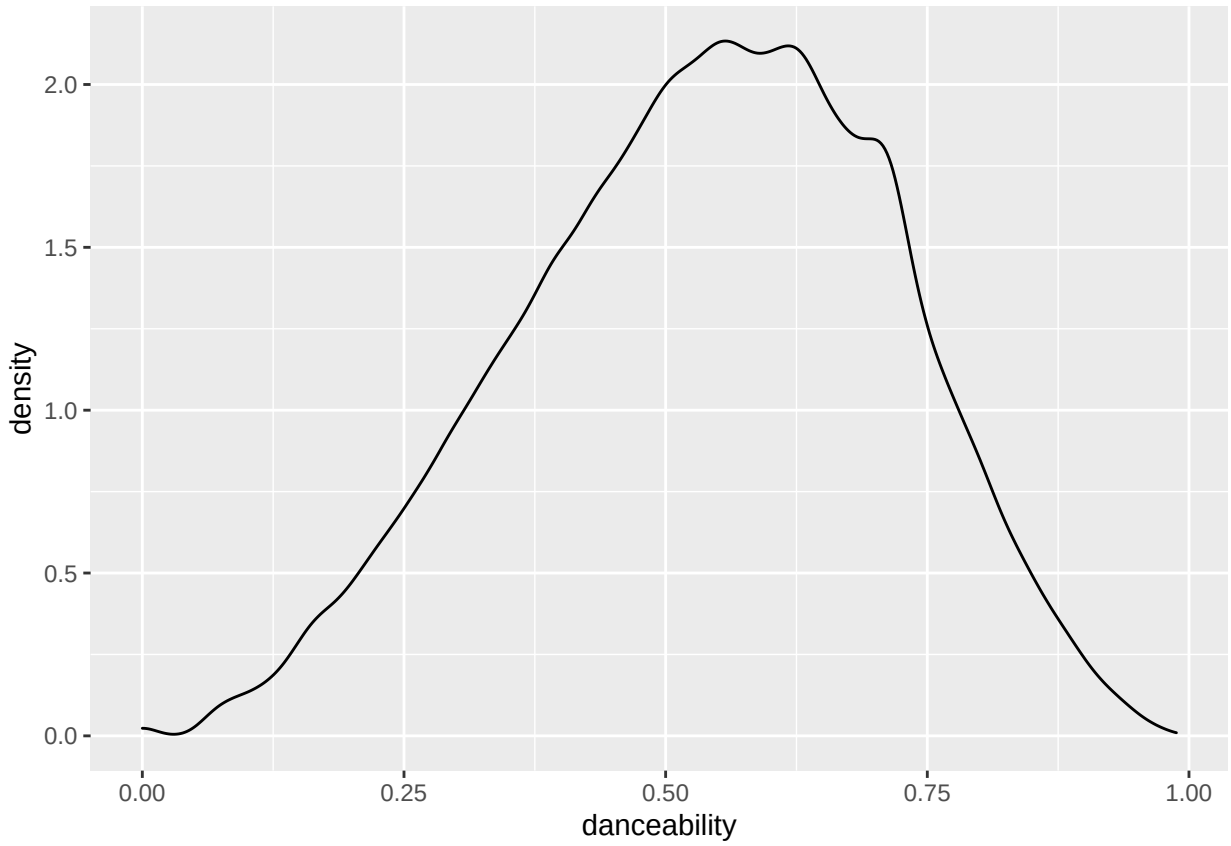
```
data %>%
  ggplot(aes(x = valence)) +
  geom_density()
```




```
data %>%
  ggplot(aes(x = liveness)) +
  geom_density()
```




```
data %>%
  ggplot(aes(x = danceability)) +
  geom_density()
```



3 Model Development

```
model <- glm(popularity ~ danceability + liveness + valence, data = data, family = "poisson")
```

4 Model Results

```
summary(model)
```

```
##
## Call:
## glm(formula = popularity ~ danceability + liveness + valence,
##     family = "poisson", data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -10.8466   -3.9529    0.3904    2.6839   10.3172
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.093135   0.001603 1929.37   <2e-16 ***
## danceability  1.059647   0.002952  358.99   <2e-16 ***
## liveness     -0.209137   0.002632  -79.45   <2e-16 ***
```

```
## valence      -0.350287   0.001967 -178.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3423309  on 170652  degrees of freedom
## Residual deviance: 3278897  on 170649  degrees of freedom
## AIC: 4028634
##
## Number of Fisher Scoring iterations: 5
```

4.1 Model Validation

5 Discussion

Limitations

Future Work

6 Appendix

7 References