

Final 210 Project

Sean Villoresi and Ellie Kang

Introduction

Music is an essential part of culture, creativity, and history. Specific songs and types of music can have great significance to groups of people and individuals alike. In America today, the music industry is both highly regarded and hotly debated. One successful song can launch an artist to the top of the charts, etching them into modern history. The importance of music and potential significance of a single song motivates the question - what makes a song successful?

Wanting to explore this question in our project, we found a dataset¹ on Kaggle with data on Spotify streams, Youtube views, and various song characteristics. There are 28 columns, and 20,719 observations. The data was collected on February 7th, 2023 by extracting the data from Youtube and Spotify. Our goal is to determine and develop the best model for predicting the success of a song based on the number of streams. We chose to use streams (as opposed to Youtube views) as our outcome variable because of inconsistencies within the data when it comes to music videos. Some music videos were not from the artist's channel (unofficial), and we wanted to test this variable as a predictor.

Variables: We will be using streams as the model's outcome variable. We chose the following variables as potential predictors based on their relevance to the listening experience of a song (as opposed to a more descriptive variable such as Description) Stream: number of streams of the song on Spotify Energy: a measure from 0.0 to 1.0 representing a perceptual measure (dynamic range, loudness, timbre, onset rate, general entropy) of intensity and activity. Key: the key the track is in measured in integers representing pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C /D , 2 = D. If no key was detected, the value is -1. Loudness: the overall loudness of a track in decibels (dB) Speechiness: a measure from 0.0 to 1.0 representing the presence of spoken words in a track. Acousticness: a from 0.0 to 1.0 of whether the track is acoustic. Instrumentalness: a measure from 0.0 to 1.0 that predicts whether a track contains no vocals. Liveness: a measure from 0.0 to 1.0 that detects the presence of an audience in the recording. Valence: a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tempo: the overall estimated tempo of a track in beats per minute (BPM). Duration_ms: the duration of the track in milliseconds. Official_video: boolean value that indicates if the video found is the official video of the song. We felt that the dataset had a sufficient number of both quantitative and categorical variables to test predictability. Thus, we

chose not to create additional predictors. However, in our data cleaning process, we removed any observations with missing values for streams, danceability, and licensed. After removing missing values for these variables, there were no remaining observations with missing data for relevant variables as listed above.

Read in the data

```
library(tidyverse)
library(tidymodels)
library(broom)
library(leaps)
library(MASS)
library(caret)
library(glmnet)
library(Stat2Data)
library(nnet)
library(lme4)
music <- read_csv("data/Spotify_Youtube.csv")
```

Cleaning the data

```
music$Uri=NULL
music$Url_youtube=NULL
music$Url_spotify=NULL
music$Description=NULL

music <- music[complete.cases(music$Stream), ]
music <- music[complete.cases(music$Danceability), ]
music <- music[complete.cases(music$Licensed), ]
sum(is.na(music$Speechiness))
```

```
[1] 0
```

Variable Selection

```
indices <- sample(1:19691, size = 15000 * 0.8, replace = F)
train.data <- music %>%
  slice(indices)
test.data <- music %>%
  slice(-indices)

lm_none <- lm(Stream ~ 1, data = train.data)

lm_all <- lm(Stream ~ Danceability + Energy + factor(Key) + Loudness + Speechiness +
  Acousticness + Instrumentalness + Liveness + Valence + Tempo +
  Duration_ms + official_video, data = train.data)

stepAIC(lm_all,
  scope = list(lower = lm_none, upper = lm_all),
  data = music, direction = "both")
```

Start: AIC=463064.5

```
Stream ~ Danceability + Energy + factor(Key) + Loudness + Speechiness +
  Acousticness + Instrumentalness + Liveness + Valence + Tempo +
  Duration_ms + official_video
```

	Df	Sum of Sq	RSS	AIC
- factor(Key)	11	7.3673e+17	6.8683e+20	463055
- Tempo	1	3.9964e+16	6.8613e+20	463063
<none>			6.8609e+20	463065
- Duration_ms	1	1.4205e+17	6.8623e+20	463065
- Liveness	1	5.1234e+17	6.8660e+20	463072
- Instrumentalness	1	5.3515e+17	6.8663e+20	463072
- Danceability	1	5.4897e+17	6.8664e+20	463072
- Speechiness	1	6.2022e+17	6.8671e+20	463073
- Valence	1	1.0679e+18	6.8716e+20	463081
- Energy	1	3.8301e+18	6.8992e+20	463129
- Acousticness	1	4.0180e+18	6.9011e+20	463133
- Loudness	1	4.1576e+18	6.9025e+20	463135
- official_video	1	5.1292e+18	6.9122e+20	463152

Step: AIC=463055.4

```
Stream ~ Danceability + Energy + Loudness + Speechiness + Acousticness +
  Instrumentalness + Liveness + Valence + Tempo + Duration_ms +
```

official_video

	Df	Sum of Sq	RSS	AIC
- Tempo	1	3.7938e+16	6.8686e+20	463054
<none>			6.8683e+20	463055
- Duration_ms	1	1.6029e+17	6.8699e+20	463056
- Liveness	1	5.3073e+17	6.8736e+20	463063
- Speechiness	1	5.4438e+17	6.8737e+20	463063
- Instrumentalness	1	5.5095e+17	6.8738e+20	463063
- Danceability	1	5.9768e+17	6.8742e+20	463064
+ factor(Key)	11	7.3673e+17	6.8609e+20	463065
- Valence	1	1.1059e+18	6.8793e+20	463073
- Energy	1	3.8384e+18	6.9067e+20	463120
- Acousticness	1	4.1152e+18	6.9094e+20	463125
- Loudness	1	4.1931e+18	6.9102e+20	463126
- official_video	1	5.1552e+18	6.9198e+20	463143

Step: AIC=463054.1

Stream ~ Danceability + Energy + Loudness + Speechiness + Acousticness +
Instrumentalness + Liveness + Valence + Duration_ms + official_video

	Df	Sum of Sq	RSS	AIC
<none>			6.8686e+20	463054
- Duration_ms	1	1.5716e+17	6.8702e+20	463055
+ Tempo	1	3.7938e+16	6.8683e+20	463055
- Liveness	1	5.2309e+17	6.8739e+20	463061
- Instrumentalness	1	5.4712e+17	6.8741e+20	463062
- Speechiness	1	5.7178e+17	6.8744e+20	463062
+ factor(Key)	11	7.3470e+17	6.8613e+20	463063
- Danceability	1	6.6805e+17	6.8753e+20	463064
- Valence	1	1.1545e+18	6.8802e+20	463072
- Energy	1	3.8372e+18	6.9070e+20	463119
- Acousticness	1	4.0804e+18	6.9095e+20	463123
- Loudness	1	4.1569e+18	6.9102e+20	463124
- official_video	1	5.1477e+18	6.9201e+20	463142

Call:

```
lm(formula = Stream ~ Danceability + Energy + Loudness + Speechiness +  
    Acousticness + Instrumentalness + Liveness + Valence + Duration_ms +  
    official_video, data = train.data)
```

Coefficients:

(Intercept)	Danceability	Energy	Loudness
2.903e+08	5.655e+07	-1.550e+08	7.151e+06
Speechiness	Acousticness	Instrumentalness	Liveness
-6.684e+07	-8.904e+07	-4.353e+07	-4.100e+07
Valence	Duration_ms	official_videoTRUE	
-4.925e+07	-2.934e+01	5.046e+07	

```
y <- music$Stream
x <- model.matrix(Stream ~ Danceability + Energy + factor(Key) + Loudness + Speechiness +
  Acousticness + Instrumentalness + Liveness + Valence + Tempo +
  Duration_ms + official_video,
  data = music)

m_lasso_cv <- cv.glmnet(x, y, alpha = 1)

best_lambda <- m_lasso_cv$lambda.min
m_best <- glmnet(x, y, alpha = 1, lambda = best_lambda)
m_best$beta
```

23 x 1 sparse Matrix of class "dgCMatrix"

```
s0
(Intercept)      .
Danceability     5.425383e+07
Energy           -1.225667e+08
factor(Key)1     1.186910e+07
factor(Key)2     -7.844484e+06
factor(Key)3     -1.761978e+06
factor(Key)4      .
factor(Key)5      .
factor(Key)6     1.613351e+06
factor(Key)7     -9.784489e+06
factor(Key)8     2.302353e+06
factor(Key)9     -1.506024e+07
factor(Key)10    -1.676901e+06
factor(Key)11     5.852992e+06
Loudness         6.532253e+06
Speechiness      -6.401535e+07
Acousticness     -7.858671e+07
Instrumentalness -3.777677e+07
```

```
Liveness          -4.111619e+07
Valence           -4.954162e+07
Tempo             -3.435953e+04
Duration_ms       -2.149974e+01
official_videoTRUE 4.901070e+07
```

```
## CHANGE LATER AFTER WE ACTUALLY CHOOSE MODELS FROM ABOVE< CURRENTLY A PLACE HOLDER.
pretransform_model <- lm(Stream ~ Danceability + Energy + factor(Key) + Loudness
  + Speechiness +
  Acousticness + Instrumentalness + Liveness + Valence + Tempo +
  Duration_ms + official_video, data = music)
```

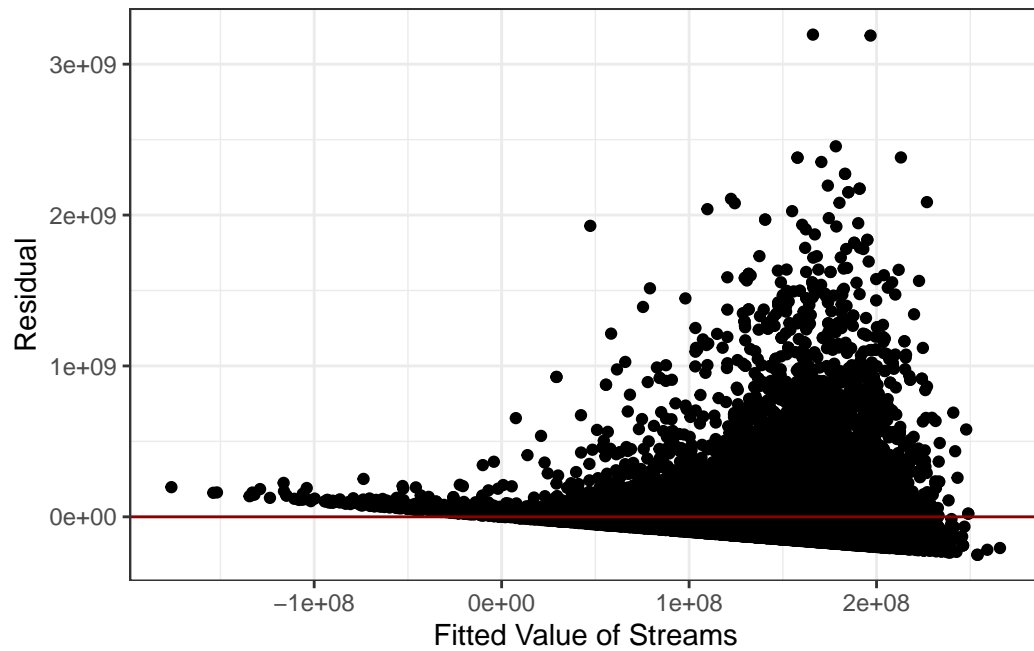
Linearity Assumptions and Checks for Transformations

```
ptmodel_aug <- augment(pretransform_model)

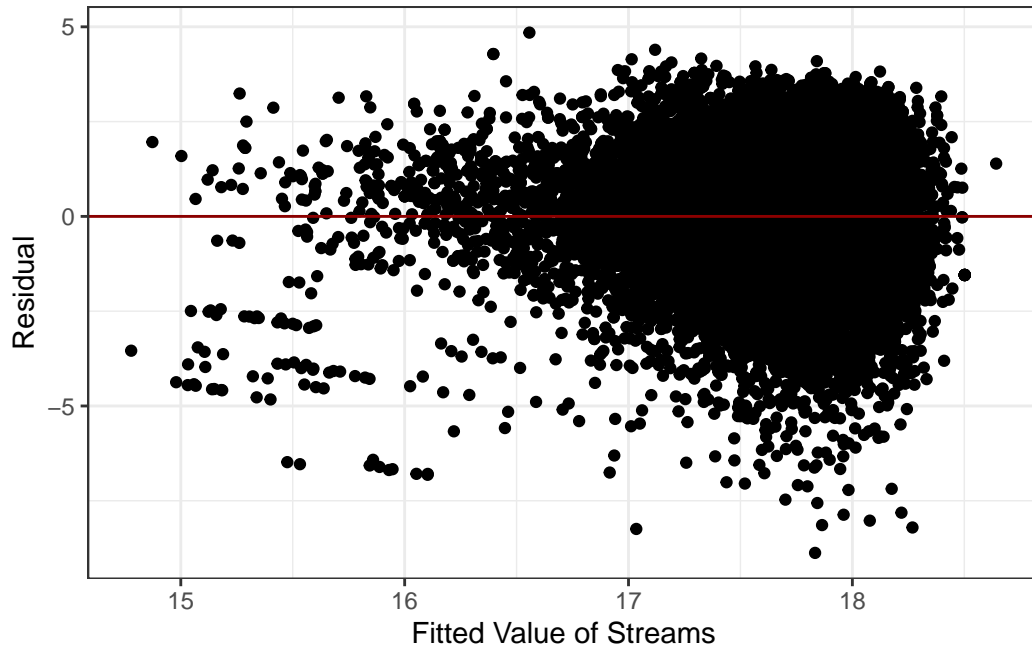
transform_model <- lm(log(Stream) ~ Danceability + Energy + factor(Key) + Loudness +
  Speechiness + Acousticness + Instrumentalness + Liveness
  + Valence + Tempo + Duration_ms + official_video,
  data = music)

tmodel_aug <- augment(transform_model)

ggplot(ptmodel_aug, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept=0, color = "darkred") +
  labs(x = "Fitted Value of Streams", y = "Residual") +
  theme_bw()
```

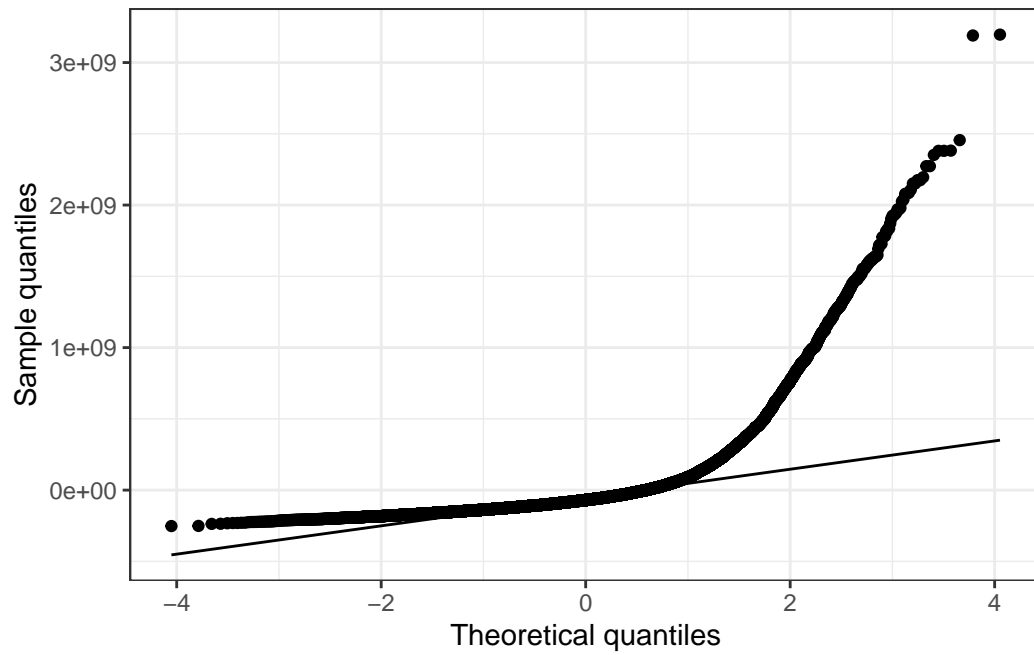


```
ggplot(tmodel_aug, aes(x = .fitted, y=.resid)) +  
  geom_point() +  
  geom_hline(yintercept=0, color = "darkred") +  
  labs(x = "Fitted Value of Streams", y = "Residual") +  
  theme_bw()
```



Looking at the visualizations above, we can see that the transformed model gives us a much better spread on the residual split around our red line then our untransformed model. As such, the residuals appear roughly symmetrical along the horizontal axis for our transformed plot, so we feel it safe to assume approximate linearity.

```
ggplot(ptmodel_aug, aes(sample = .resid)) +  
  stat_qq() +  
  stat_qq_line() +  
  theme_bw() +  
  labs(x = "Theoretical quantiles",  
       y = "Sample quantiles")
```

```
ggplot(tmodel_aug, aes(sample = .resid)) +  
  stat_qq() +  
  stat_qq_line() +  
  theme_bw() +  
  labs(x = "Theoretical quantiles",  
       y = "Sample quantiles")
```

