Introduction

The Billboard Hot 100 is the music industry standard record chart in the United States and has been a reliable source of music popularity rankings since its inception. There is a lot of appeal, both culturally and financially, for artists and record labels to enter and maintain a spot on the Billboard Hot 100. Artists want their music to be heard and label heads want to invest in popular artists. For this reason, many artists will optimize their music to maximize their chances of success of staying on the charts. We are interested in what audio features can influence chart longevity and how impactful those audio features are. Spotify, the world's largest music streaming service, provides a measure of numerical and categorical audio features for every song in its catalog. In this research, we aim to use those audio features to predict how long a song stays on the Billboard Hot 100.

Our paper seeks to answer the question : Given that a song has made it onto the Billboard's Hot 100 rankings chart, what is the best regression model that predicts how long a song continuously stays on the Billboard chart and how confident are we in prediction capabilities of a model that primarily uses audio features along with limited information from the chart to determine it's longevity?

Specifications

The response variable **weeks** is a performance metric of songs in the billboard charts which we also refer to as longevity. Relevant explanatory variables used in our models are : **year** , **entry_position**, **art_prev_h100_ct**, **st_duration_ms**, **danceability**, **energy**, **key**, **loudness**, **speechiness**, **acousticness**, **instrumentalness**, **liveness**, **valence**, **tempo** , **entry_month**, **st_explicit**, **mode** , **genre**, **has_feat** which are mostly audio features, along with chart entry variables. (Refer to **Apendix A** for full data description). The response **weeks** is quantitative and we assume it has a linear relationship with our relevant explanatory variables. While our EDA did reveal weaker correlations that we desire, there was no non linear relationships apparent, so we chose to proceed with linear regression for our final model.

We imagine our model, if it proves to have significant predictive quality will be beneficial primarily to musical artist's and those involved in their music production and marketing. Our predictive model is most likely of best use in aiding marketing decisions for artists/music labels releasing and/or promoting new songs.

Final Model

Before starting our modeling, we considered how changes in music trends/taste over time which influence many aspects of music popularity may affect our modeling results. From our EDA, the distribution of **weeks** drastically changes over the decades, but we see much closer distributions from 2008 to the present. Therefore we decided to focus solely on songs entering the billboard chart starting from the year 2008 in our regression analysis. This is done under the assumption that including songs before 2008 would not reflect more recent longevity of songs in the charts which would negatively affect our model's predictive performance. We further split the data into a testing and training set with songs released before 2015 inclusive in the training set and songs released after 2015 and beyond in the testing set. This allows us to see how well our model can perform with unseen data and in the years outside of the model's training domain.

From the results of our exploratory data analysis we observed that the distribution of our response variable **weeks** is right skewed. The Box-Cox family of power transformations $X \to X^p$ can be used to improve symmetry in skewed distributions, and power transformations down the power ladder will correct our positive skew and preserve the order of data (Fox 2016, 60). Taking $p = 0$, We applied a natural log transformation on **weeks** before fitting our models which reduced the skew and can be reverted to actual values with the exponential function. The log transformation allowed us to normalize the response variable hoping to satisfy our model's underlying assumptions of normality and constant variance. For relationships between our

quantitative variables, we found few significant correlations between weeks and our explanatory variables as follows:

art__prev__100__ct	**speechiness**	**acousticness**	**yr__chart__ct**	**valence**
−0.2	−0.12	−0.10	−0.11	0.11

We also observed the most significant qualitative variables were **genre**, **st__explicit**, and **key** but also with minimal differences in distributions of chart longevity **weeks**.

The initial model we fitted with the 9 most correlated/significant variables had an R^2 value of 0.1854 which wasn't very good, but that was not surprising with the low correlations we have with most of the variables. It was very challenging to find improvements for the model but the next steps we considered were including more variables along with interactions. With the addition of more variables, our models became very complex and with many regressors that don't really improve model quality.

We were interested in the best subset model of the large scope of regressors with interactions but with $p \geq 20$ predictors, total subsets 2^p would be computationally intensive. So we decided to use forward selection, which only checks for at most $\sum_{k=0}^{p-1}(p-k)$ models to find the best sequence of predictors (James et al 2013, 208). This would also limit the scope of interactions to consider since any main effect excluded removes it's interactions as well. We used both AIC and BIC which are appropriate criterion for selection between models with different p variables that address complexity concerns and got the following final two models:

Model	Score	Adj R^2	# of regressors	Cross validated RMSE
AIC Model	9493.924	0.2444	39	1.149951
BIC Model	9550.964	0.2228	10	1.160277

We observed that both models from our selection results did not improve our initial R^2 by much, both have $R^2 < 0.25$, and BIC discarded 29 regressors compared to the AIC model. A lower score is sought for both criterion, but for prediction purposes, our goal is to choose a model with the lowest test error.

Our test data should not be used in our model fitting and selection process but we can directly estimate test error using a cross-validation approach which replicates a training and test split of our data. We chose to perform leave one out cross validation which is equivalent to a k-fold cross-validation with folds equal to the number of our observations. For linear regression, leave one out cross validation mean squared error can be analytically calculated using our model hat matrix as $\frac{1}{n}\sum_i^n(\frac{y_i-\hat{y_i}}{1-h_i})^2$ (James et al. 2013, 180). Using roots of the analytical estimate for test error, the model from BIC selection resulted in a lower LOO-CV-RMSE which makes it much more preferential for our prediction goal.

Aside from a good estimated test error, we are also concerned with model complexity which tends to increase model variance. Although the AIC model resulted in a slightly lower CV-RMSE, it remains more complex with nearly 4 times # of regressors compared to the BIC model. We decided to use the model selected by BIC criteria as our best model because it nicely penalizes model complexity more than other criterions we considered. Our final and least complex model using BIC criteria is summarized as follows:

Final Model	Terms
BIC Model	**genre**, **art__prev__100__ct**, **danceability**, **acousticness**, **st__duration__ms**, **st__duration__ms:acousticness**

The BIC model retained significant variables as we saw from our EDA results and retained one significant interaction between **acousticness** and **st__duration__ms**. Unfortunately our selected best model has a low R^2 only explaining about 22.28% of variation in the training data. This is once again not surprising as we have seen very weak correlations between the variables in our data and the response **weeks**.

We visualized our model's predictive performance to understand how it's low R^2 affected performance on the training data of songs from $2008 - 2015$ as well as generalizability which was evaluated using the test data for songs released after 2015 as follows :

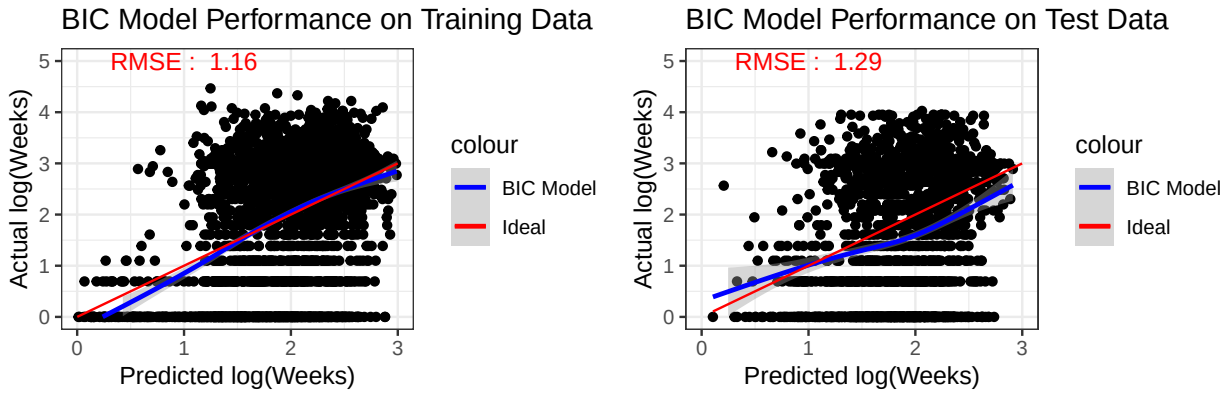

Figure 1: (a) Training fitted log(weeks) against actual log(weeks) (b) Test predicted log(weeks) against actual log(weeks). Both plots of model performance shows there is a very weak goodness of fit with many points over/under estimated severely and far from the ideal regression line.

Discussion and Final Conclusions

The main takeaway from our final model is that due to a lack of goodness of fit we are unable to obtain a predictive model with acceptable predictive performance. This result indicates that the problems with our model stem mostly from the explanatory variables we used during the modeling process. However, we can also report that using a model mainly composed of audio features, does provide some measure of capability to predict the number of weeks a song is on the chart. Although, it is not in scope of our goal for finding the best model for prediction, it is also possible to further explore the significance of audio features such as **genre**, **acousticness**, **danceability**, and **st_duration_ms**. However, given that there are also issues with the underlying model assumptions which further reflect the lack of good fit, we cant fully trust test statistics in our model. An alternative approach is using bootstrap of estimated coefficients which makes no assumptions about the underlying data generating distribution.

A main limitation to our model is that the underlying assumptions in using linear regression are not satisfied, one of which is the independence assumption that is a problem by nature of our data. The number of weeks a song is on the billboard chart could be dependent on the release of another song and how long that song is on the charts. Additionally, our model only looks at the audio features of songs and does not deal with the historical contexts nor the actual lyrical composition of the song, which has been shown to influence the popularity of a song. By being bound to only easily quantifiable data, our model is limited. Furthermore, our data only deals with songs that have made it into the Billboard Hot 100 chart, so if someone wants to use our model for prediction, it has to be under the assumption that the new song will be on the chart for at least one week, which is not likely to occur for many songs.

With such a low Adj R^2 value and the poor performance of the model as seen in **Figure 1**, the model should not be used in any serious capacity. This project does provide evidence that it might be worthwhile to include audio features in future studies on what affects the number of weeks a song remains on the Billboard Hot 100 Chart. While our model's non-audio features were limited mainly to information on a song's release date and recording artists, further regression analyses could look to incorporate data on a song's lyrical composition and the amount of advertising the song received. They could look into such factors as the number of repeated verses in a song and the complexity of the words in the song, providing a measure of how catchy the song is. In conclusion, we are not confident in the prediction capabilities of our final model which primarily uses audio features to determine a songs longevity on the Billboard Hot 100 Charts.

Additional Work

Other Modeling Approaches

As part of our efforts to improve the predictive model, we also considered the Ridge and LASSO regularization methods which have the benefit of reducing model test error at the cost of additional bias. We were interested specifically the variable selection aspect of LASSO which shrinks coefficients of insignificant regressors to zero by applying the following penalty on least square coefficient estimates $\hat{\theta}$ for linear regression:

$$\hat{\theta}_{LASSO} = argmin_\theta \frac{1}{n}\|\mathbf{Y} - \mathbf{X}\theta\|_2^2 + \lambda \sum_{j=1}^{d} |\theta_i|$$

The expansive model fitted with many possible interactions is very complex and we observed many coefficients of varying magnitudes. Generally, Lasso assumes many coefficients are truly zero, and it performs better where some of the predictors have large coefficients, and the remaining predictors have very small coefficients (James et al. 2013, 223). Both those aspects are present in our expanded model, but mostly the model complexity is our concern so we opt for lasso regression to produce a simpler model that we prefer.

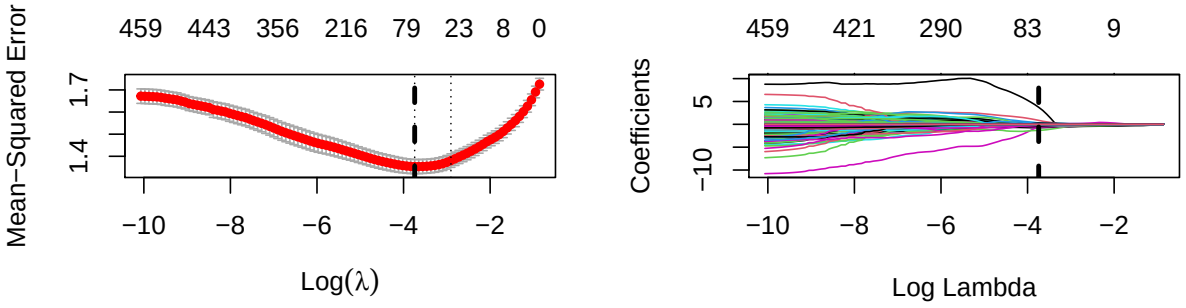

Figure 2 : (a) 10-fold CV to choose best lambda for lasso regression minimizing MSE. (b) The shrinkage of coefficients in lasso regression; Many of the larger coefficients shrunk to zero well before reaching best lambda value. Broken lines indicate best lambda value as Log(λ).

Model	R^2	# of regressors	RMSE
AIC Model	0.2444	39	1.149951
BIC Model	0.2228	10	1.160277
LASSO Model	0.2632	80	1.130655

For the LASSO model, # of regressors counts total nonzero coefficient terms. So the model obtained with LASSO performed significant variable selection by shrinking many coefficients to zero as we wanted. But comparing important metrics, we see that the AIC/BIC models obtained with a simpler forward selection have a very similar estimate of the test error with much lower # of regressors. So Lasso regression kept many coefficients which accounts for it having a higher R^2 but did not really improve test error by a lot which is our metric of concern for predictions. Consequently we prioritized conducting model selection using the simpler variable selection approaches with AIC and BIC.

Final Model Diagnostics

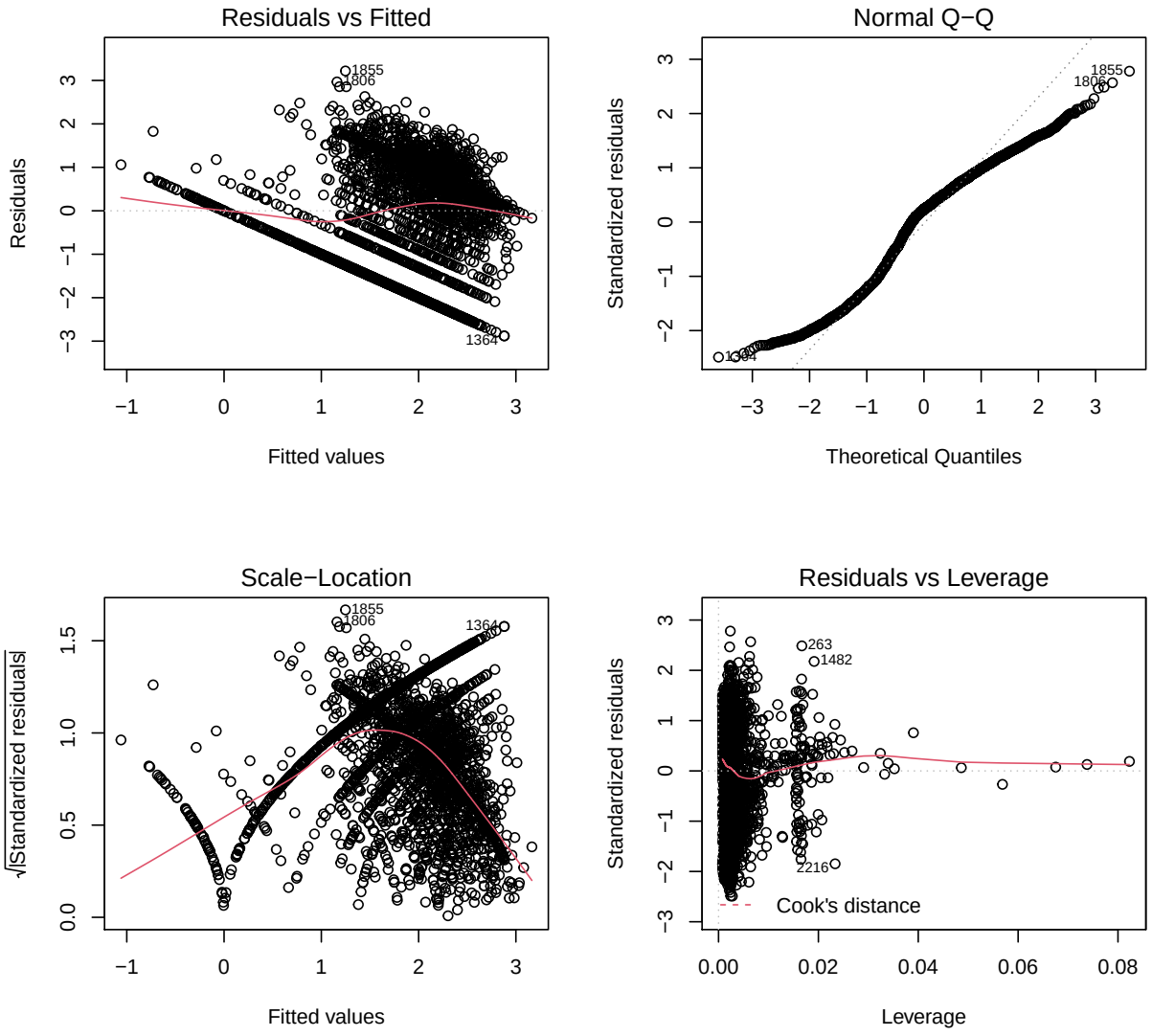

Figure 3 : Diagnostics plots for BIC model with log transformation on **weeks**. (a) Top Left: The 'red-line' is close to the mean value of 0 but the appearance of line-like patterns on the left side is a cause for concern about the linearity assumption. (b) Top Right:The Q-Q plot with log transformation slightly improved normality but presence of skew on both tails persists, and many points are not on the line. (c) Bottom Left: The Scale-Location plot shows we do not satisfy homoscedasticity assumption. (d) Bottom Right: The Residuals vs Leverage plot indicates presence of some high leverage but non-influential points in our regression.

References

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. An Introduction to Statistical Learning with Applications in R. 6th ed. Springer.

Fox, John. 2016. Applied Regression Analysis & Generalized Linear Models. 3rd ed. Sage.

Miller, Sean. "Billboard Hot Weekly Charts - Dataset by Kcmillersean." Data.world, 9 June 2018, data.world/kcmillersean/billboard-hot-100-1958-2017.

Appendixes

Appendix A: Overview of Data

The data used is a pre-processed combination of two datasets. The first original dataset is *Billboard Hot Weekly Charts Data* between 8/2/1958 and 12/28/2019. Aggregate information for each unique song was filtered to obtain relevant information such as genres, total number of weeks in the charts, entry_position, entry_month, and year chart count at time of song's entry. The second dataset is the Spotify audio features for each corresponding unique song obtained from the Spotify API. Both original datasets were obtained from the website **data.world** available for public use and credited to the author Sean Miller. The final datasets has a total of 28474 records/rows each representing a unique song that entered the Billboard Hot 100 charts from 1958 to 2019 and with columns of information about the song's appearance in the rankings and Spotify audio features for the song.

The following are the variables of interest from the dataset relevant to the the research question outlined along with their descriptions.

Response Variable	Description
weeks	The number of consecutive weeks a song was on the chart since it's release.

Quant Variables	Description
year	The year a song made its first appearance on the chart.
yr_chart_ct	Total year count of songs that entered the chart at the time of songs first appearance on the chart.
entry_position	The position of a song in its first appearance on the chart.
art_prev_h100	Total previous songs by the same artist that appeared on the chart.
st_duration_ms	The duration of the song's spotify track in milliseconds.
danceability	A danceability rating between 0-1 with 1 being most danceable.
energy	Float from 0 to 1 measuring the perceptual intensity of the song.
key	An estimated measure of a track's "key" represented by integers 0-11 mapped to pitches using the standard pitch class notation.
loudness	A measure of a track's loudness in decibels.
speechiness	A measure of the presence of spoken word on a track from 0-1 with 1 having the most spoken word.
acousticness	A 0-1 measure of how acoustic (no electrical amplification) a track is with 1 indicating the most acoustic.
instrumentalness	A measure of how instrumental (no spoken word) a track is from 0-1 with 1 being the most instrumental.

Quant Variables	Description
liveness	A measure of audience presence in a track from 0-1 with 1 indicating a higher probability of a live performance.
valence	A measure of the "musical positiveness" of a track from 0-1 with 1 indicating the highest presence of cheerful, happy, and euphoric music.
tempo	A measure of the track's estimated beats per minute (BPM).
time_signature	An estimated overall time signature of a track in minutes.

Catg Variables	Description
entry_month	The month in which a song made its first appearance on the chart.
st_explicit	A logical vector that is 1 if the song is explicit and 0 if else.
mode	Modality of the track, major is represented by 1 and minor is 0.
genre	The primary genre of the song.
has_feat	A logical vector that is 1 if the song had another artist featured and 0 if else

Appendix B: Exploratory Data Analysis Summary

From summary of the data we had around 4902 rows of mostly older songs with missing values. Missing values in the data being were audio features obtained from spotify. After filtering our data for songs only from 2007 missing values were much lower and pattern was reduced. Due to their nature we cant really use an imputing approach to replace audio features so we deemed it fine to remove all rows with missing values.

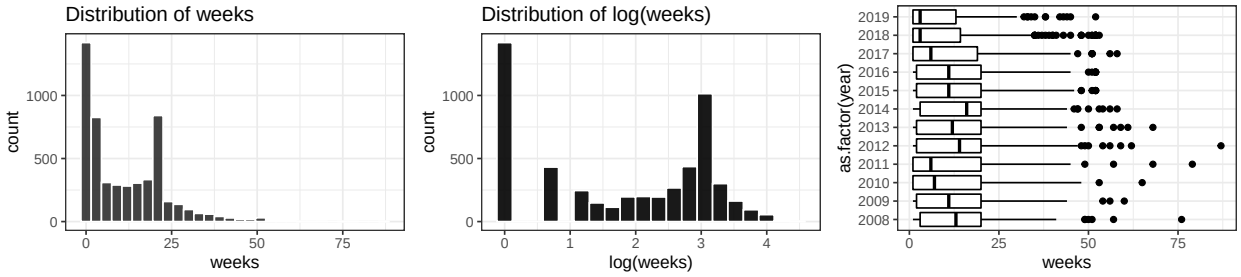

Figure 1: (a) Distribution of response variable **weeks** is skewed to the right as most higher week values are not very common. Interesting double modes at around 1 weeks and around 23 weeks. (b) Distribution of response variable **weeks** is less skewed after log transformation, but bimodality remains. (c) Comparing distribution of weeks over range years we use in our modeling.

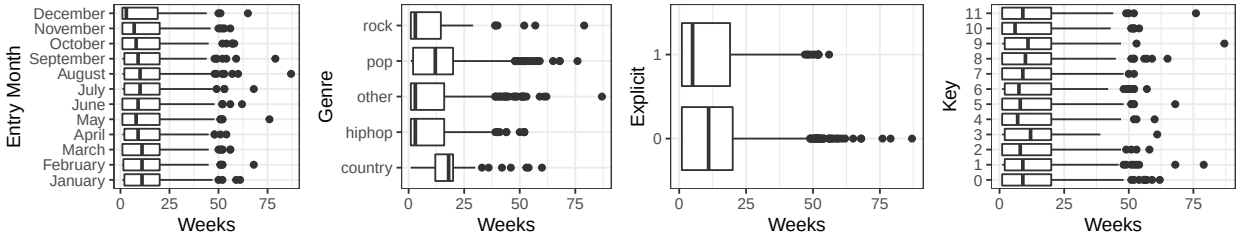

Figure 2 : Relationship between **weeks** and significant qualitative variables. **Genre** seems likely to be more influential in our models compared to the other variables, followed by **st_explicit** and **key**.

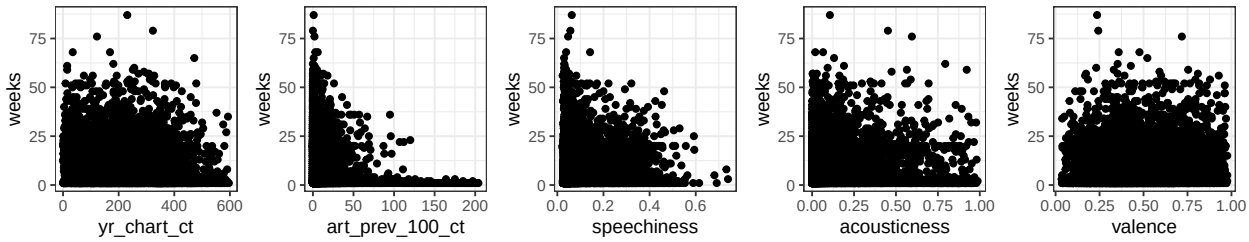

Figure 4 : Bivariate plots for relationship between **weeks** and most correlated explanatory variables.

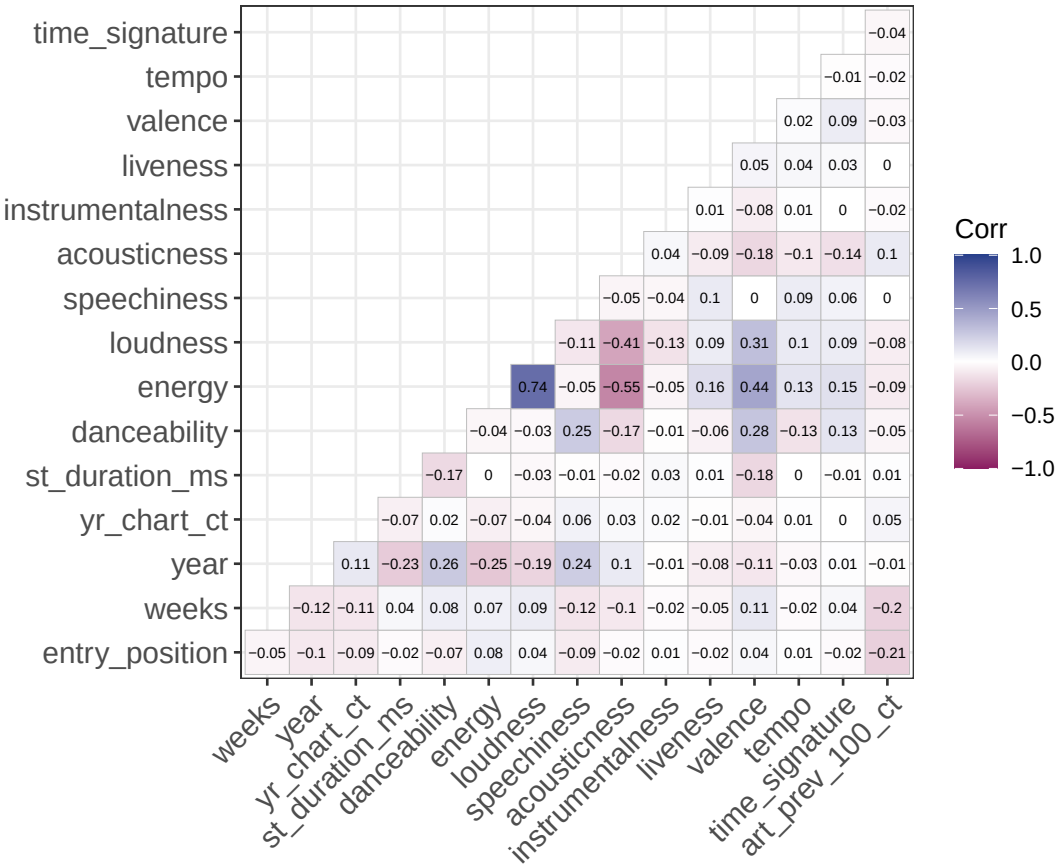

Figure 3 : Correlation among quantitative variables of interest and with response variable **Weeks**. **Weeks** has highest correlation with **art_prev_100_ct**, **yr_chart_ct** and **speechiness**, **valence**, **acousticness** and **loudness** among the audio features, but all the correlations are quite low as well. We also observe considerable correlations between some audio features, notably the trio; **energy**, **loudness**, and **acousticness**.

Appendix C: Code

```
knitr::opts_chunk$set(echo = TRUE, eval=FALSE, warning = FALSE, message = FALSE)
library(tidyverse)
library(caret)
library(leaps)
library(gridExtra)
library(cowplot)
library(corrplot)
library(reshape2)
library(ggplot2)
library(car)
library(glmnet)
library(GGally)
library(olsrr)
library(ggcorrplot)
library(knitr)
```

```
#Loading preprocessed dataset
data  <- read.csv(file = "hot-100/hot_merged_full.csv",
                            header = TRUE,
                            sep = ",",
                            dec = ".")

data <- filter(data, year > 2007)
```

```
### EXPLORATORY DATA ANALYSIS ####
#check where most missing data are across years in the data set
rows_missing <- data[!complete.cases(data), ] %>% ggplot() +
  geom_bar(aes(x=year),width = 1, color = 'white', fill = 'gray18') + theme_bw() +
  labs(title = "Songs with missing values by year")

#most of the missing data is audio features so their unavailability for older songs makes sense.
#Removing rows with missing audio features not a big impact on our response variable either.


#plot distribution of response
weeks_full <- ggplot(data['weeks'], aes(x=weeks)) +
  geom_histogram(bins = 20, alpha = 0.9, color = "white", fill = "gray18", binwidth = 3) +
  theme_bw() +
  labs(title = "Distribution of weeks")


#plot distribution of response with transformation
weeks_log <- ggplot((data %>%
                              dplyr::select(weeks)), aes(x = log(weeks))) +
  geom_histogram(alpha = 0.9, color = "white", fill = "black", bins = 20) +
  theme_bw() +
  labs(title = "Distribution of log(weeks)")

#distribution of weeks by years
weeks_years <-ggplot(data=data, aes(x=as.factor(year), y=weeks))  +
  geom_boxplot(color='black') + theme_bw() + coord_flip()
```

```r
#Removing rows with missing values
data <- data %>% na.omit()

#Fixing data type for some columns
data <- data %>%
  mutate(st_explicit = as.factor(data$st_explicit)) %>%
  mutate(mode = as.factor(data$mode)) %>%
  mutate(has_feat = as.factor(data$has_feat))

#dropping irrelevant columns as per proposal
data <- data %>%
  dplyr::select(-c('songid','entry_weekid','performer',
            'st_popularity','pop','rock','country','hiphop','r.b','dance',
            'feat_artist'))


## Plots for relationships with categorical data
bxplt1 <- ggplot(data=data) +
  geom_boxplot(aes(x=factor(entry_month, levels = month.name), y=weeks)) +
  theme_bw() +
  coord_flip() +
  labs(x = "Entry Month", y = "Weeks");

bxplt2 <-ggplot(data=data) +
  geom_boxplot(aes(x=genre, y=weeks)) +
  theme_bw() +
  coord_flip() +
  labs(x = "Genre", y = "Weeks");

bxplt3 <-ggplot(data=data) +
  geom_boxplot(aes(x=has_feat, y=weeks)) +
  theme_bw() +
  coord_flip() +
  labs(x = "has_feat", y = "Weeks");

bxplt4 <-ggplot(data=data) +
  geom_boxplot(aes(x=st_explicit, y=weeks)) +
  theme_bw() +
  coord_flip() +
  labs(x = "Explicit", y = "Weeks");

bxplt5 <-ggplot(data=data) +
  geom_boxplot(aes(x=mode, y=weeks)) +
  theme_bw() +
  coord_flip() +
  labs(x = "Mode", y = "Weeks");

bxplt6 <-ggplot(data=data) +
  geom_boxplot(aes(x=as.factor(key), y=weeks)) +
  theme_bw() +
  coord_flip() +
  labs(x = "Key", y = "Weeks");
```

```r
#correlation between quantitative independent variables and response
corr_plt <- data %>% dplyr::select(-c('entry_month','genre',
                                      'has_feat', 'st_explicit',
                                      'mode','key', 'peak')) %>%
  cor() %>%
  ggcorrplot(type = 'lower',
             ggtheme = ggplot2::theme_bw,
             colors = c("maroon4", "white", "royalblue4"),
             lab = TRUE, lab_size = 2)

#bivariate plots between response and most correlated variables
qsc_plt1 <- ggplot(data=data, aes(x=yr_chart_ct, y=weeks)) +
  geom_point(color='black') + theme_bw()
qsc_plt2 <-ggplot(data=data, aes(x=art_prev_100_ct, y=weeks))  +
  geom_point(color='black') + theme_bw()
qsc_plt3 <-ggplot(data=data, aes(x=speechiness, y=weeks))  +
  geom_point(color='black') + theme_bw()
qsc_plt4 <-ggplot(data=data, aes(x=acousticness, y=weeks))  +
  geom_point(color='black') + theme_bw()
qsc_plt5 <-ggplot(data=data, aes(x=valence, y=weeks))  +
  geom_point(color='black') + theme_bw()
qsc_plt6 <-ggplot(data=data, aes(x=as.factor(year), y=weeks))  +
  geom_boxplot(color='black') + theme_bw() + coord_flip()
```

```r
## Train test split
training <- filter(data, year <= 2015)
test <- filter(data, year > 2015)
```

```r
## INITIAL Model
init.model <- lm(data = data, formula = log(weeks) ~ art_prev_100_ct +
                   yr_chart_ct + valence + acousticness + danceability +
                   speechiness + genre + key + st_explicit)



### FINAL MODEL from Forward selection with AIC and BIC
#With log transformation
biggest1 = formula(lm(log(weeks) ~ (yr_chart_ct + entry_month +
                                      genre + st_explicit + st_duration_ms +
                                      danceability + energy + key + loudness +
                                      mode + speechiness + acousticness+
                                      instrumentalness + liveness +
                                      valence + tempo + time_signature +
                                      has_feat + art_prev_100_ct)^2, data = training))

min.model1 = lm(log(weeks) ~ 1, data = training)

fwd.model.AIC1 = step(min.model1,
                      direction = "forward",
                      trace = 0,
                      scope = biggest1, k = 2)

fwd.model.BIC1 = step(min.model1,
                      direction = "forward",
```

```
                      trace = 0,
                      scope = biggest1,
                      k = log(nrow(training)))



#Without log transformation
biggest2 = formula(lm(weeks ~ (yr_chart_ct + entry_month + genre +
                                 st_explicit + st_duration_ms +
                                 danceability + energy + key + loudness +
                                 mode + speechiness + acousticness+
                                 instrumentalness + liveness + valence +
                                 tempo + time_signature +
                                 has_feat + art_prev_100_ct)^2, data = training))

min.model2 = lm(weeks ~ 1, data = training)

fwd.model.AIC2 = step(min.model2,
                      direction = "forward",
                      trace = 0,
                      scope = biggest2, k = 2)

fwd.model.BIC2 = step(min.model2,
                      direction = "forward",
                      trace = 0,
                      scope = biggest2,
                      k = log(nrow(training)))
```

```
#' @title rmse_cv_loo
#' @description calculates leave one out cross validation error
#' @param model model object
#' @return root mean squared error
rmse_cv_loo <- function(model) {
  cv_mse = mean((resid(model) / (1-hatvalues(model)))^2)
  return(sqrt(cv_mse))
}

fwd.model.AIC1 %>% rmse_cv_loo()
fwd.model.BIC1 %>% rmse_cv_loo()
```

```
### FINAL MODEL WITH LASSO APPROACH
modelX <- lm(data=training, formula=biggest1) %>% model.matrix()
lasso.cv <- cv.glmnet(y = log(training$weeks),
                      x = modelX,
                      alpha = 1,
                      family = "gaussian")

best_lambda <- lasso.cv$lambda.min

lasso.model2_tr <- glmnet(y = log(training$weeks),
                      x = modelX,
                      alpha = 1)

lasso.model2 <- glmnet(y = log(training$weeks),
```

```
                  x = modelX,
                  lambda = best_lambda,
                  alpha = 1)

#lasso model predictions on training data
lasso_predictions <- lasso.model2 %>% predict(modelX) %>% as.vector()

#lasso performance metrics
Lasso.RMSE = RMSE(lasso_predictions, log(training$weeks))
Lasso.R2 = R2(lasso_predictions, log(training$weeks))
```

```
#get # of regressors in lasso model
coef(lasso.model2) %>% as.matrix() %>%
  as.data.frame() %>%
  filter(s0 > 0 | s0 < 0) %>%
  nrow()
```

```
#FINAL MODEL PERFORMANCE PLOTS

BIC_train <- fwd.model.BIC1$fitted.values
BIC_pred <- predict.lm(fwd.model.BIC1, newdata = test)


rmse_train_pred <- round(sqrt(mean((log(training$weeks) - BIC_train)^2)), digits = 2)
rmse_test_pred <- round(sqrt(mean((log(test$weeks) - BIC_pred)^2)), digits = 2)

#Training perfomance plot
bic_plt_train <- ggplot(training, aes(x=BIC_train, y = log(weeks))) +
  geom_point(color='black') +
  geom_smooth(aes(color = 'BIC Model'),
              stat = 'smooth', method = 'gam', formula = y ~ s(x, bs = "cs")) +
  geom_line(aes(x=seq(min(BIC_train),max(BIC_train),
                      length.out = length(BIC_train)),
                y=seq(min(BIC_train),max(BIC_train),
                      length.out = length(BIC_train)), color = 'Ideal')) +
  scale_color_manual(values = c('blue', 'red')) +
  theme(legend.position = "none") +
  theme_bw() +
  labs(title = "BIC Model Performance on Training Data",
       x = "Predicted log(Weeks)", y = 'Actual log(Weeks)') +
  annotate(geom = "text", x=1, y=5,
           label=paste("RMSE : ",rmse_train_pred), color = "red") +
  xlim(0,3) + ylim(0,5)

#Test perfomance plot
bic_plt_test <- ggplot(test, aes(x=BIC_pred,
                                 y = log(weeks))) + geom_point(color='black') +
  geom_smooth(aes(color = 'BIC Model'),
              stat = 'smooth', method = 'gam', formula = y ~ s(x, bs = "cs")) +
  geom_line(aes(x=seq(min(BIC_pred),max(BIC_pred),
                      length.out = length(BIC_pred)),
                y=seq(min(BIC_pred),max(BIC_pred),
                      length.out = length(BIC_pred)), color = 'Ideal')) +
  scale_color_manual(values = c('blue', 'red')) +
```

```
  theme(legend.position = "none") +
  theme_bw() +
  labs(title = "BIC Model Performance on Test Data",
       x = "Predicted log(Weeks)", y = 'Actual log(Weeks)') +
  annotate(geom = "text", x=1, y=5,
           label=paste("RMSE : ",rmse_test_pred), color = "red") +
  xlim(0,3) + ylim(0,5)
```

```
#Show final model performance plots
grid.arrange(bic_plt_train, bic_plt_test, ncol=2)
```

```
# Plot Lasso results
par(mfrow=c(1,2))


plot(lasso.cv)
lines(c(log(best_lambda), log(best_lambda)),
      c(-1000, 1000), lty = "dashed", lwd = 3)

plot(lasso.model2_tr, xvar = "lambda")
lines(c(log(best_lambda), log(best_lambda)),
      c(-1000, 1000), lty = "dashed", lwd = 3)
```

```
#Show final model diagnostics plots
par(mfrow=c(2,2))
plot(fwd.model.BIC1)
```

```
#show response variable EDA plots
grid.arrange(weeks_full, weeks_log, weeks_years, ncol = 3)
```

```
#show response-qualitative relationships plots
grid.arrange(bxplt1, bxplt2, bxplt4, bxplt6, nrow = 1, ncol = 4)
```

```
#show response-quantitative relashionship plots
grid.arrange(qsc_plt1, qsc_plt2, qsc_plt3, qsc_plt4, qsc_plt5, nrow=1, ncol=5)
```

```
#show quantitative variables correlations plot
corr_plt
```