Final Project

Sam Webster, Yaxi Lei, Alissa Caltagirone

11/12/2018

setwd("/Users/samanthawebster/Desktop/")  
songDF <- read.csv("featuresdf.csv")  
songDF2 <- read.csv("data.csv")  
songDF2$key <- as.factor(songDF2$key)  
songDF2$mode <- as.factor(songDF2$mode)  
songDF2$time\_signature <- as.factor(songDF2$time\_signature)  
songDF2$target <- as.factor(songDF2$target)

set.seed(1861)  
top100names <- songDF$name  
songDF2$top100 <- as.factor(ifelse(is.element(songDF2$song\_title,top100names) == TRUE,1,0))  
songDF2$top100artist <- as.factor(ifelse(is.element(songDF2$artist,songDF$artists) == TRUE,1,0))  
songDF$ranking <- (1:100)

songDF$mode <- as.factor(songDF$mode)  
songDF$time\_signature <- as.factor(songDF$time\_signature)  
songDF$key <- as.factor(songDF$key)  
summary(songDF[,-c(1,2)]) #min, max, and mean

## artists danceability energy key   
## Ed Sheeran : 4 Min. :0.2580 Min. :0.3460 1 :14   
## The Chainsmokers: 4 1st Qu.:0.6350 1st Qu.:0.5565 11 :13   
## Drake : 3 Median :0.7140 Median :0.6675 8 :12   
## Martin Garrix : 3 Mean :0.6968 Mean :0.6607 0 :10   
## Bruno Mars : 2 3rd Qu.:0.7702 3rd Qu.:0.7875 5 : 9   
## Calvin Harris : 2 Max. :0.9270 Max. :0.9320 6 : 9   
## (Other) :82 (Other):33   
## loudness mode speechiness acousticness   
## Min. :-11.462 0:42 Min. :0.02320 Min. :0.000259   
## 1st Qu.: -6.595 1:58 1st Qu.:0.04312 1st Qu.:0.039100   
## Median : -5.437 Median :0.06265 Median :0.106500   
## Mean : -5.653 Mean :0.10397 Mean :0.166306   
## 3rd Qu.: -4.327 3rd Qu.:0.12300 3rd Qu.:0.231250   
## Max. : -2.396 Max. :0.43100 Max. :0.695000   
##   
## instrumentalness liveness valence tempo   
## Min. :0.000e+00 Min. :0.04240 Min. :0.0862 Min. : 75.02   
## 1st Qu.:0.000e+00 1st Qu.:0.09828 1st Qu.:0.3755 1st Qu.: 99.91   
## Median :0.000e+00 Median :0.12500 Median :0.5025 Median :112.47   
## Mean :4.796e-03 Mean :0.15061 Mean :0.5170 Mean :119.20   
## 3rd Qu.:1.335e-05 3rd Qu.:0.17925 3rd Qu.:0.6790 3rd Qu.:137.17   
## Max. :2.100e-01 Max. :0.44000 Max. :0.9660 Max. :199.86   
##   
## duration\_ms time\_signature ranking   
## Min. :165387 3: 1 Min. : 1.00   
## 1st Qu.:198490 4:99 1st Qu.: 25.75   
## Median :214106 Median : 50.50   
## Mean :218387 Mean : 50.50   
## 3rd Qu.:230543 3rd Qu.: 75.25   
## Max. :343150 Max. :100.00   
##

round(sapply(songDF[,-c(1,2)], sd), 4) #standard deviation

## artists danceability energy key   
## 22.6058 0.1251 0.1392 3.7315   
## loudness mode speechiness acousticness   
## 1.8021 0.4960 0.0951 0.1667   
## instrumentalness liveness valence tempo   
## 0.0260 0.0790 0.2164 27.9529   
## duration\_ms time\_signature ranking   
## 32851.0777 0.1000 29.0115

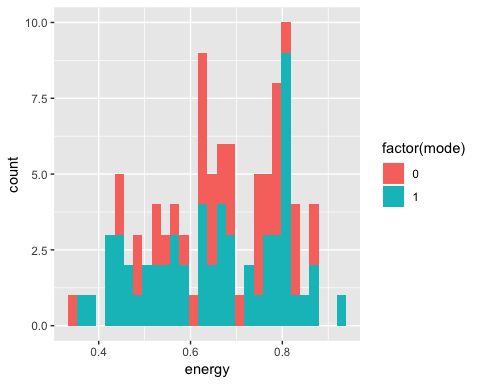
We delete the first two rows because they are id and names, which does not make sense in this analysis.

## Plots

## (1)

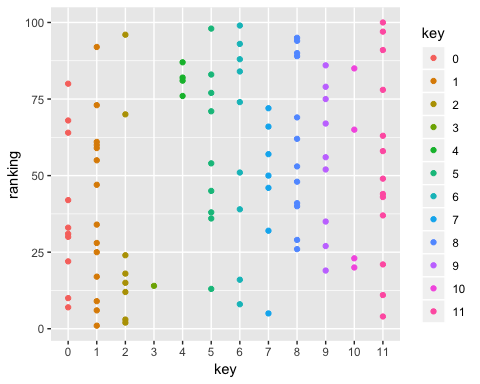
library(ggplot2)  
ggplot(songDF, aes(energy)) + geom\_histogram(aes(fill = factor(mode)))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

 This plot shows that in the top 100 songs, there are up to ten songs with the energy around 0.8. There are more songs with major modality than minor modality. At 0.8 energy, there are 9 songs have major modality while there is only one song has minor modality.

## (2)

ggplot(songDF, aes(x = key, y = ranking, color = key)) + geom\_point()

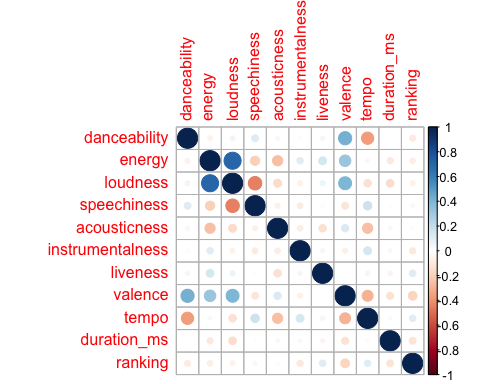
 These two histogram plots just show the distribution of the tempo and the valence in the top 100 songs seperately. There are more songs with lower tempo and medium valence.

## (3)

library("corrplot")

## corrplot 0.84 loaded

songDFnew <- songDF[,-c(1,2,3,6,8,16)]  
corrplot(cor(songDFnew))



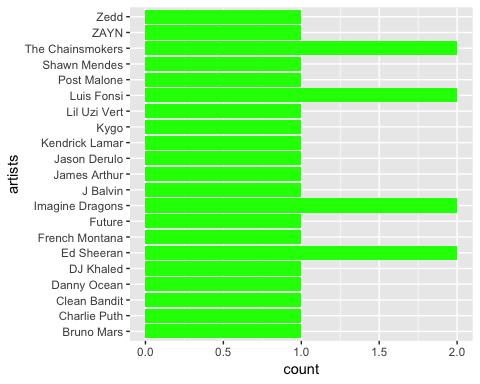
cor(songDFnew$ranking, songDFnew)

## danceability energy loudness speechiness acousticness  
## [1,] -0.08687914 -0.07363648 -0.05735295 0.03778292 0.01654014  
## instrumentalness liveness valence tempo duration\_ms ranking  
## [1,] -0.094071 0.1048057 -0.184857 0.1001237 -0.1143937 1

This is a correlation plot that indicates the correlation between variables among the 100 songs.

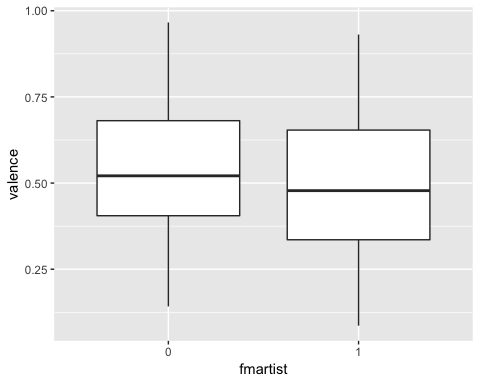
## (4)

songDF$artists <- as.factor(songDF$artists)  
ggplot(songDF[1:25,], aes(x = artists)) + geom\_bar(fill = "green") + coord\_flip()

 We sampled the top 25 songs and did a count on how many times a specific artist showing up. This shows that there are 4 artists show twice in the top 25 songs, which means artists may have an influence on the success of a song.

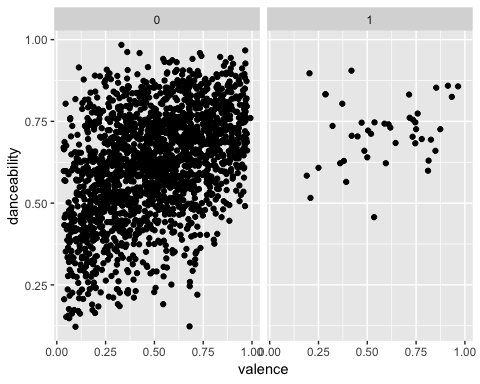
## (5)

library("plyr")  
counts <- count(songDF, var = "artists")  
names <- counts[c(which(counts$freq > 1)),1]  
songDF$fmartist <- as.factor(ifelse(is.element(songDF$artists,names) == TRUE,1,0))  
ggplot(songDF, aes(x = fmartist, y = valence)) + geom\_boxplot()

 Based on what we found in the previous plot, we created a new column which marks all rows with artists that appears more than once in the top 100 songs as “famous artist”(1) and otherwise 0. We did a boxplot based on that with the danceability and the artist. This shows the average danceability is lower for the “famous artists” than others.

## (6)

ggplot(songDF2, aes(x = valence, y = danceability)) + geom\_point() + facet\_wrap(songDF2$top100)



## Analysis

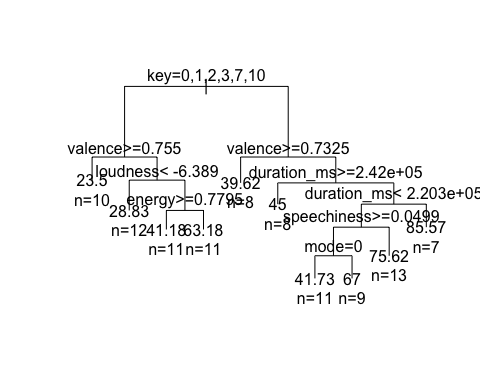
## Regression for predicting ranking

options(scipen = 15)  
mod1 <- lm(ranking ~ danceability + energy + loudness + I(mode) + speechiness + acousticness + instrumentalness + liveness + valence + tempo + duration\_ms, data = songDF)  
summary(mod1)

##   
## Call:  
## lm(formula = ranking ~ danceability + energy + loudness + I(mode) +   
## speechiness + acousticness + instrumentalness + liveness +   
## valence + tempo + duration\_ms, data = songDF)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -56.364 -24.034 1.721 25.668 49.017   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 78.41614648 47.60009754 1.647 0.103  
## danceability 7.21935207 28.80775982 0.251 0.803  
## energy -6.74724832 33.26010538 -0.203 0.840  
## loudness 0.77574809 2.80547773 0.277 0.783  
## I(mode)1 3.49914699 6.09287196 0.574 0.567  
## speechiness 2.06972871 37.31694359 0.055 0.956  
## acousticness 11.91704489 19.66807277 0.606 0.546  
## instrumentalness -122.41405575 120.08832097 -1.019 0.311  
## liveness 37.61724700 38.30596472 0.982 0.329  
## valence -28.96546803 17.52453691 -1.653 0.102  
## tempo 0.07709040 0.12344849 0.624 0.534  
## duration\_ms -0.00012648 0.00009273 -1.364 0.176  
##   
## Residual standard error: 29.42 on 88 degrees of freedom  
## Multiple R-squared: 0.08617, Adjusted R-squared: -0.02806   
## F-statistic: 0.7543 on 11 and 88 DF, p-value: 0.6837

## Tree for predicting ranking

require("rpart")  
ranktree <- rpart(ranking ~ danceability + energy + key + mode + loudness + speechiness + acousticness + instrumentalness + liveness + valence + tempo + duration\_ms, data = songDF)  
par(xpd = TRUE)   
plot(ranktree)  
text(ranktree, use.n = TRUE, pretty = FALSE)



library(randomForest)  
rf <- randomForest(ranking ~ danceability + energy + key + mode + loudness + speechiness + acousticness + instrumentalness + liveness + valence + tempo + duration\_ms, data = songDF)

## mse

library(Metrics)  
pred1 <- predict(ranktree, songDF)  
pred2 <- predict(mod1, songDF)  
pred3 <- predict(rf, songDF)  
mse(songDF$ranking, pred1)

## [1] 463.8229

mse(songDF$ranking, pred2)

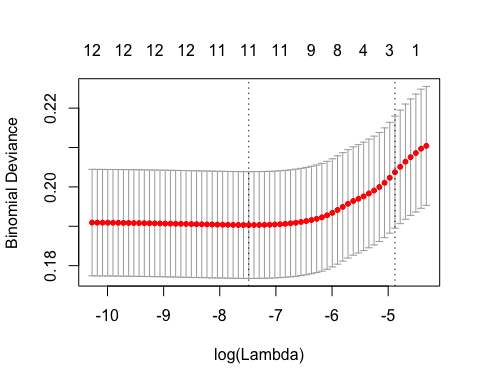
## [1] 761.4504

mse(songDF$ranking, pred3)

## [1] 161.462

## Lasso

options(scipen = 15)  
set.seed(23)  
library("useful")  
library("glmnet")  
MyFormula <- as.formula(top100 ~ .-X -key -target -song\_title -artist -top100artist)  
Xvar <- build.x(MyFormula, songDF2)  
Yvar <- build.y(MyFormula, songDF2)  
Lasso <- cv.glmnet(Xvar, Yvar, alpha = 1, family = "binomial")  
plot(Lasso)



coef(Lasso, s = "lambda.min")

## 16 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -2.2813352293594  
## (Intercept) .   
## acousticness .   
## danceability 3.0602455675787  
## duration\_ms -0.0000009753598  
## energy -3.1559586828271  
## instrumentalness -3.4579975045556  
## liveness -3.7234118692181  
## loudness 0.2635114527489  
## mode1 -0.0835273216959  
## speechiness 0.3990771605074  
## tempo -0.0057943786297  
## time\_signature3 .   
## time\_signature4 1.7302986757695  
## time\_signature5 .   
## valence 0.4329867408381

Lasso$glmnet.fit$dev.ratio[which(Lasso$glmnet.fit$lambda == Lasso$lambda.min)]

## [1] 0.1388902

coef(Lasso, s = "lambda.1se")

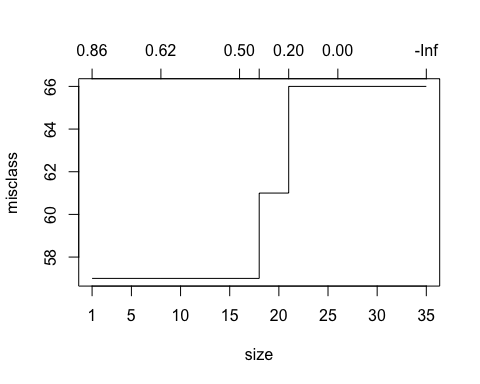
## 16 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -4.8144745  
## (Intercept) .   
## acousticness .   
## danceability 1.6900316  
## duration\_ms .   
## energy .   
## instrumentalness -0.3648323  
## liveness -0.1424253  
## loudness .   
## mode1 .   
## speechiness .   
## tempo .   
## time\_signature3 .   
## time\_signature4 .   
## time\_signature5 .   
## valence .

Lasso$glmnet.fit$dev.ratio[which(Lasso$glmnet.fit$lambda == Lasso$lambda.1se)]

## [1] 0.03689777

## tree

set.seed(1861)  
require("tree")  
tree <- tree(top100 ~ . -X -song\_title -target -artist -top100artist, data = songDF2)  
cv = cv.tree(tree,FUN=prune.misclass)  
plot(cv)



prune = prune.misclass(tree,best=3)  
plot(prune);text(prune, pretty = FALSE)

