multiple-linear-regression.R

ethan-school

2021-12-17

library(leaps)  
library(MLmetrics)

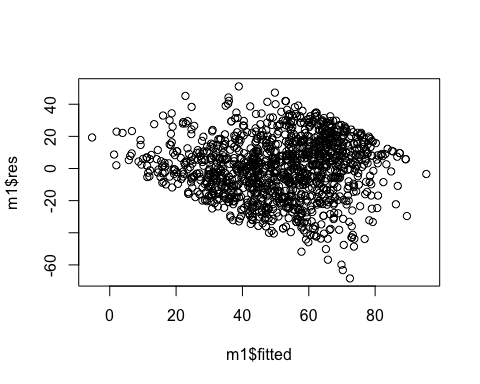
##   
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':  
##   
## Recall

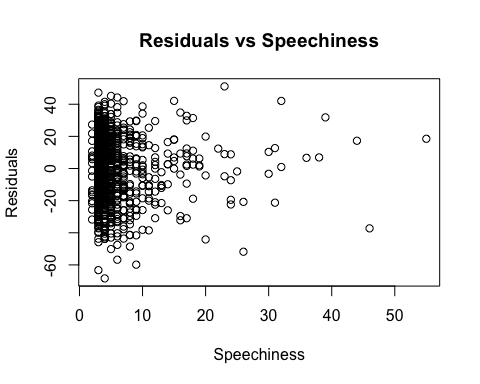
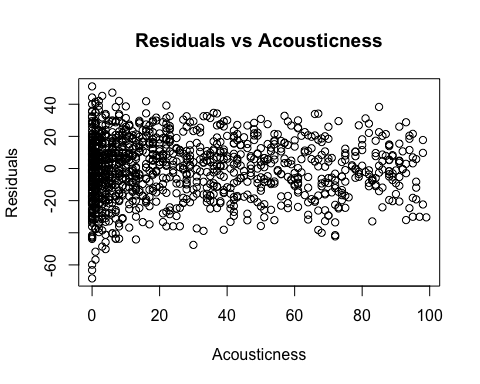
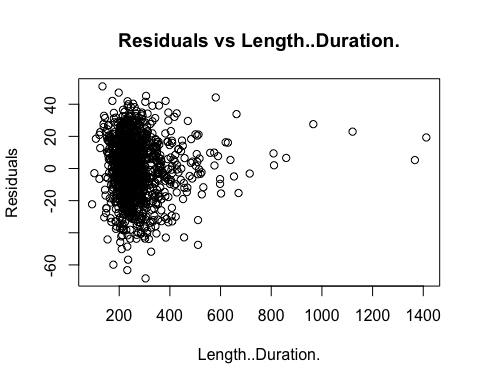
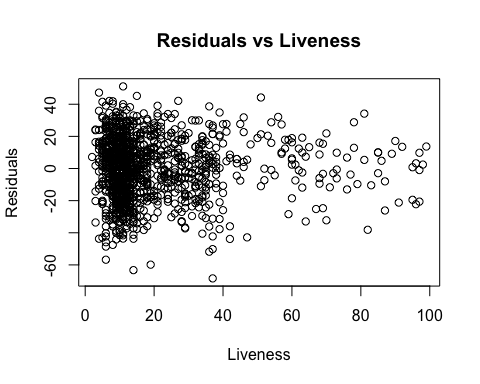
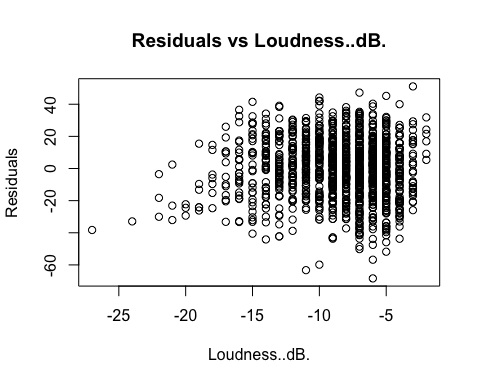
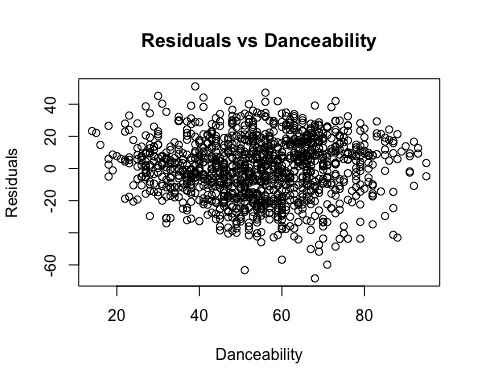
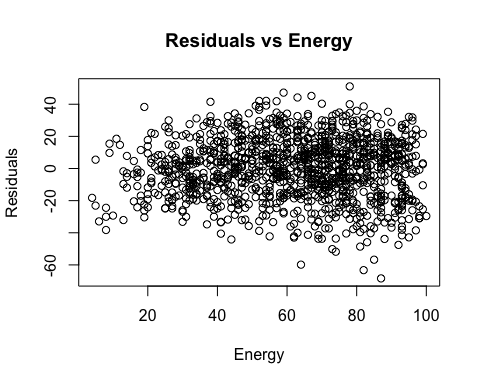
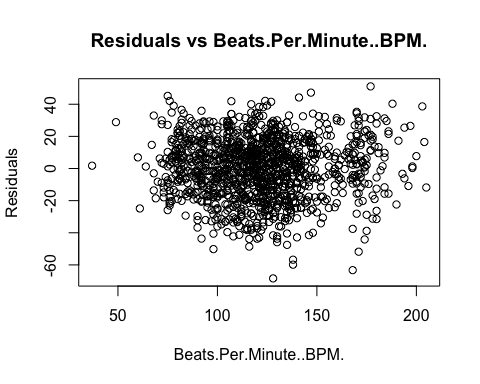
setwd("~/Documents/Fall-2021/datamining/spotify-project")  
spotify\_clean <- read.csv("csv/spotify-2000-clean.csv")  
spotify\_df <- data.frame(spotify\_clean)  
spotify\_tr\_df <- spotify\_df[1:1200,]  
spotify\_test\_df <- spotify\_df[1201:nrow(spotify\_df),]  
  
# remove irrelevant (categorical) attributes  
spotify\_tr\_df\_valence <-  
 subset(spotify\_tr\_df,  
 select = -c(Title, Artist, Year, Popularity, TopGenre))  
spotify\_test\_df <-  
 subset(spotify\_test\_df,  
 select = -c(Title, Artist, Year, Popularity, TopGenre))  
  
# multiple linear regression, numeric only - predict valence  
# need to verify LINE assumptions and also do cross validation to confirm  
# also do variable selection  
  
m1 <- lm(Valence ~ ., data = spotify\_tr\_df\_valence)  
print(summary(m1))

##   
## Call:  
## lm(formula = Valence ~ ., data = spotify\_tr\_df\_valence)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -68.401 -12.379 1.315 13.211 51.104   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -51.225004 5.668271 -9.037 < 2e-16 \*\*\*  
## Beats.Per.Minute..BPM. 0.059881 0.020241 2.958 0.00315 \*\*   
## Energy 0.679823 0.044886 15.146 < 2e-16 \*\*\*  
## Danceability 0.780893 0.037361 20.901 < 2e-16 \*\*\*  
## Loudness..dB. -2.109250 0.218613 -9.648 < 2e-16 \*\*\*  
## Liveness 0.038047 0.032718 1.163 0.24511   
## Length..Duration. -0.030230 0.005532 -5.464 5.65e-08 \*\*\*  
## Acousticness 0.049343 0.026290 1.877 0.06077 .   
## Speechiness -0.289322 0.117681 -2.459 0.01409 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.7 on 1191 degrees of freedom  
## Multiple R-squared: 0.451, Adjusted R-squared: 0.4473   
## F-statistic: 122.3 on 8 and 1191 DF, p-value: < 2.2e-16

plot(m1$res ~ m1$fitted)



# plot residuals vs Xi1 ... Xik  
names\_clean <- setdiff(names(spotify\_tr\_df\_valence), c("Valence"))  
for (colname in names\_clean) {  
 plot(  
 m1$res ~ spotify\_tr\_df\_valence[, colname],  
 main = paste("Residuals vs", colname),  
 ylab = "Residuals",  
 xlab = colname  
 )  
}



X <-  
 spotify\_tr\_df\_valence[,!names(spotify\_tr\_df\_valence) %in% c("Valence")]  
y <- spotify\_tr\_df\_valence[, "Valence"]  
out <-  
 summary(regsubsets(X, y, nbest = 2, nvmax = ncol(X))) # 'nbest' is the number of top models shown for subset size  
tab1 <-  
 cbind(out$which, out$rsq, out$adjr2, out$cp) # organize the results into a table  
print(tab1)

## (Intercept) Beats.Per.Minute..BPM. Energy Danceability Loudness..dB. Liveness  
## 1 1 0 0 1 0 0  
## 1 1 0 1 0 0 0  
## 2 1 0 1 1 0 0  
## 2 1 0 0 1 0 0  
## 3 1 0 1 1 1 0  
## 3 1 0 1 1 0 0  
## 4 1 0 1 1 1 0  
## 4 1 1 1 1 1 0  
## 5 1 1 1 1 1 0  
## 5 1 0 1 1 1 0  
## 6 1 1 1 1 1 0  
## 6 1 1 1 1 1 0  
## 7 1 1 1 1 1 0  
## 7 1 1 1 1 1 1  
## 8 1 1 1 1 1 1  
## Length..Duration. Acousticness Speechiness   
## 1 0 0 0 0.2832387 0.2826404 358.973045  
## 1 0 0 0 0.1557210 0.1550163 635.615446  
## 2 0 0 0 0.3910831 0.3900657 127.010699  
## 2 0 1 0 0.3186145 0.3174761 284.227151  
## 3 0 0 0 0.4277238 0.4262883 49.520848  
## 3 1 0 0 0.3993454 0.3978388 111.086103  
## 4 1 0 0 0.4429438 0.4410792 18.501816  
## 4 0 0 0 0.4314915 0.4295885 43.347002  
## 5 1 0 0 0.4461100 0.4437905 13.633064  
## 5 1 0 1 0.4449903 0.4426661 16.062213  
## 6 1 0 1 0.4485946 0.4458214 10.242878  
## 6 1 1 0 0.4477831 0.4450058 12.003226  
## 7 1 1 1 0.4503879 0.4471603 8.352282  
## 7 1 0 1 0.4493874 0.4461540 10.522784  
## 8 1 1 1 0.4510113 0.4473237 9.000000

step(m1)

## Start: AIC=7037.79  
## Valence ~ Beats.Per.Minute..BPM. + Energy + Danceability + Loudness..dB. +   
## Liveness + Length..Duration. + Acousticness + Speechiness  
##   
## Df Sum of Sq RSS AIC  
## - Liveness 1 473 417083 7037.2  
## <none> 416610 7037.8  
## - Acousticness 1 1232 417842 7039.3  
## - Speechiness 1 2114 418724 7041.9  
## - Beats.Per.Minute..BPM. 1 3062 419671 7044.6  
## - Length..Duration. 1 10445 427055 7065.5  
## - Loudness..dB. 1 32563 449172 7126.1  
## - Energy 1 80240 496849 7247.2  
## - Danceability 1 152812 569422 7410.8  
##   
## Step: AIC=7037.16  
## Valence ~ Beats.Per.Minute..BPM. + Energy + Danceability + Loudness..dB. +   
## Length..Duration. + Acousticness + Speechiness  
##   
## Df Sum of Sq RSS AIC  
## <none> 417083 7037.2  
## - Acousticness 1 1361 418444 7039.1  
## - Speechiness 1 1977 419059 7040.8  
## - Beats.Per.Minute..BPM. 1 2974 420056 7043.7  
## - Length..Duration. 1 10537 427620 7065.1  
## - Loudness..dB. 1 33037 450119 7126.6  
## - Energy 1 85050 502133 7257.9  
## - Danceability 1 153273 570356 7410.7

##   
## Call:  
## lm(formula = Valence ~ Beats.Per.Minute..BPM. + Energy + Danceability +   
## Loudness..dB. + Length..Duration. + Acousticness + Speechiness,   
## data = spotify\_tr\_df\_valence)  
##   
## Coefficients:  
## (Intercept) Beats.Per.Minute..BPM. Energy   
## -50.83329 0.05897 0.68900   
## Danceability Loudness..dB. Length..Duration.   
## 0.77511 -2.12191 -0.03036   
## Acousticness Speechiness   
## 0.05170 -0.27894

pred1 <- predict.lm(m1, newdata = spotify\_test\_df, type = "response")  
  
spotify\_table\_preds <- cbind(spotify\_test\_df,pred1)  
  
print(MAPE(spotify\_table\_preds$pred1,spotify\_table\_preds$Valence))

## [1] 0.4734709