TO CHOOSE A BEST CLASSIFIER MODEL FOR SPOTIFY DATA TO CATEGORIZE WHETHER THE SONG IS LIKED OR DISLIKED.

INTRODUCTION

Given a dataset of 2017 songs with attributes from Spotify's API. Each song is labeled "1" meaning liked it and "0" for songs didn't like. The aim is to check if a classifier could be built so that it could predict whether or not song is liked.

Each row represents a song. There are 16 columns. 13 of which are song attributes, one column for song name, one for artist, and a column called "target" which is the label for the song.

Here are the 13 track attributes: acousticness, danceability, duration (ms), energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, timesignature, and valence.

Link for dataset: https://www.kaggle.com/geompack/spotifyclassification

Audio Features Object

Sr no	Key	Type	Value description
1.	Duration_ms	Int	The duration of the track in milliseconds.
2.	Key	Int	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.
3.	Mode	Int	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
4.	Time_signature	Int	An estimated overall time signature of a track. The time signature (aka meter) is a notational convention to specify how many beats are in each bar (or measure).
5.	Acousticness	Float	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
6.	Danceability	Float	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
7.	Energy	Float	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude

			scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
8.	Instrumentalness	Float	Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
9.	Liveness	Float	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
10.	Loudness	Float	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
11.	Speechiness	Float	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
12.	Valence	Float	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
13.	Tempo	Float	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

DATA-ANALYSIS

```
##Importing data
library(readr)
data <- read_csv("C:/Users/k/Desktop/project/songdata.csv")


##Data Pre-Processing
data1=data[,c(-1,-16,-17)]
names(data1)

##  [1] "acousticness"     "danceability"     "duration_ms"      "energy"
##  [5] "instrumentalness" "key"              "liveness"         "loudness"
##  [9] "mode"             "speechiness"      "tempo"            "time_signature"
## [13] "valence"          "target"

is.null("data1")

## [1] FALSE

str(data1)

## tibble [2,017 x 14] (S3: tbl_df/tbl/data.frame)
##  $ acousticness    : num [1:2017] 0.0102 0.199 0.0344 0.604 0.18 0.00479 0.0145 0
.0202 0.0481 0.00208 ...
##  $ danceability    : num [1:2017] 0.833 0.743 0.838 0.494 0.678 0.804 0.739 0.266
 0.603 0.836 ...
##  $ duration_ms     : num [1:2017] 204600 326933 185707 199413 392893 ...
##  $ energy          : num [1:2017] 0.434 0.359 0.412 0.338 0.561 0.56 0.472 0.348
0.944 0.603 ...
##  $ instrumentalness: num [1:2017] 2.19e-02 6.11e-03 2.34e-04 5.10e-01 5.12e-01 0.
00 7.27e-06 6.64e-01 0.00 0.00 ...
##  $ key             : num [1:2017] 2 1 2 5 5 8 1 10 11 7 ...
##  $ liveness        : num [1:2017] 0.165 0.137 0.159 0.0922 0.439 0.164 0.207 0.16
 0.342 0.571 ...
##  $ loudness        : num [1:2017] -8.79 -10.4 -7.15 -15.24 -11.65 ...
##  $ mode            : num [1:2017] 1 1 1 1 0 1 1 0 0 1 ...
##  $ speechiness     : num [1:2017] 0.431 0.0794 0.289 0.0261 0.0694 0.185 0.156 0.
0371 0.347 0.237 ...
##  $ tempo           : num [1:2017] 150.1 160.1 75 86.5 174 ...
##  $ time_signature  : num [1:2017] 4 4 4 4 4 4 4 4 4 4 ...
##  $ valence         : num [1:2017] 0.286 0.588 0.173 0.23 0.904 0.264 0.308 0.393
```

```
0.398 0.386 ...
## $ target          : num [1:2017] 1 1 1 1 1 1 1 1 1 1 ...
```

```r
head(data1,10)
```

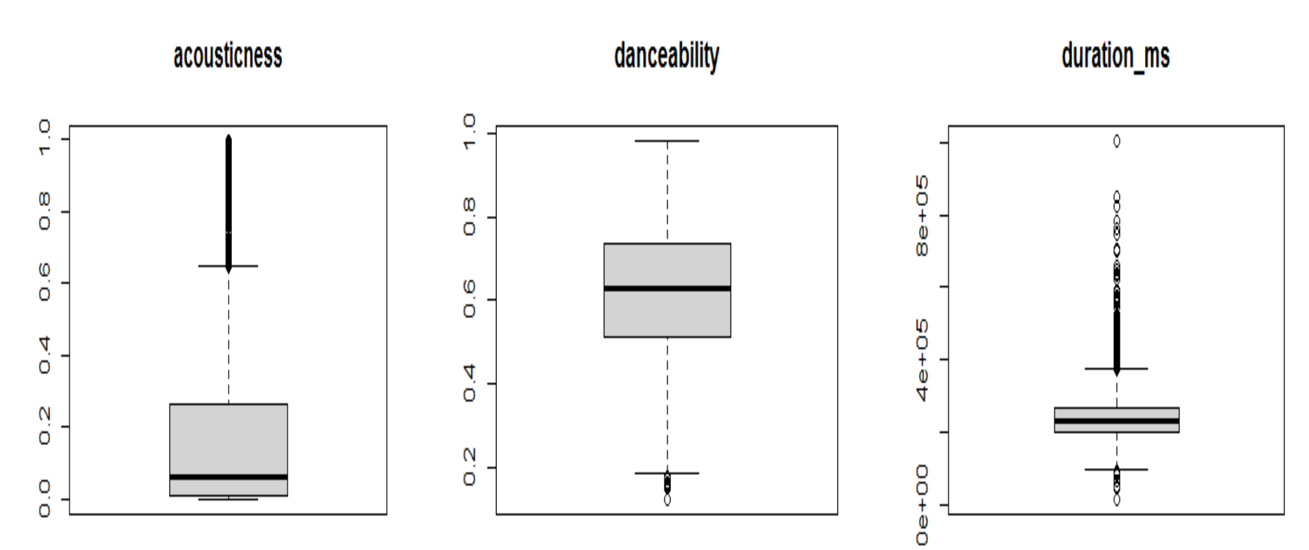
```
## # A tibble: 10 x 14
##    acousticness danceability duration_ms energy instrumentalness   key liveness
##           <dbl>        <dbl>       <dbl>  <dbl>            <dbl> <dbl>    <dbl>
##  1      0.0102         0.833      204600  0.434       0.0219         2   0.165
##  2      0.199          0.743      326933  0.359       0.00611        1   0.137
##  3      0.0344         0.838      185707  0.412       0.000234       2   0.159
##  4      0.604          0.494      199413  0.338       0.51           5   0.0922
##  5      0.18           0.678      392893  0.561       0.512          5   0.439
##  6      0.00479        0.804      251333  0.56        0              8   0.164
##  7      0.0145         0.739      241400  0.472       0.00000727     1   0.207
##  8      0.0202         0.266      349667  0.348       0.664         10   0.16
##  9      0.0481         0.603      202853  0.944       0             11   0.342
## 10      0.00208        0.836      226840  0.603       0              7   0.571
## # ... with 7 more variables: loudness <dbl>, mode <dbl>, speechiness <dbl>,
## #   tempo <dbl>, time_signature <dbl>, valence <dbl>, target <dbl>
```

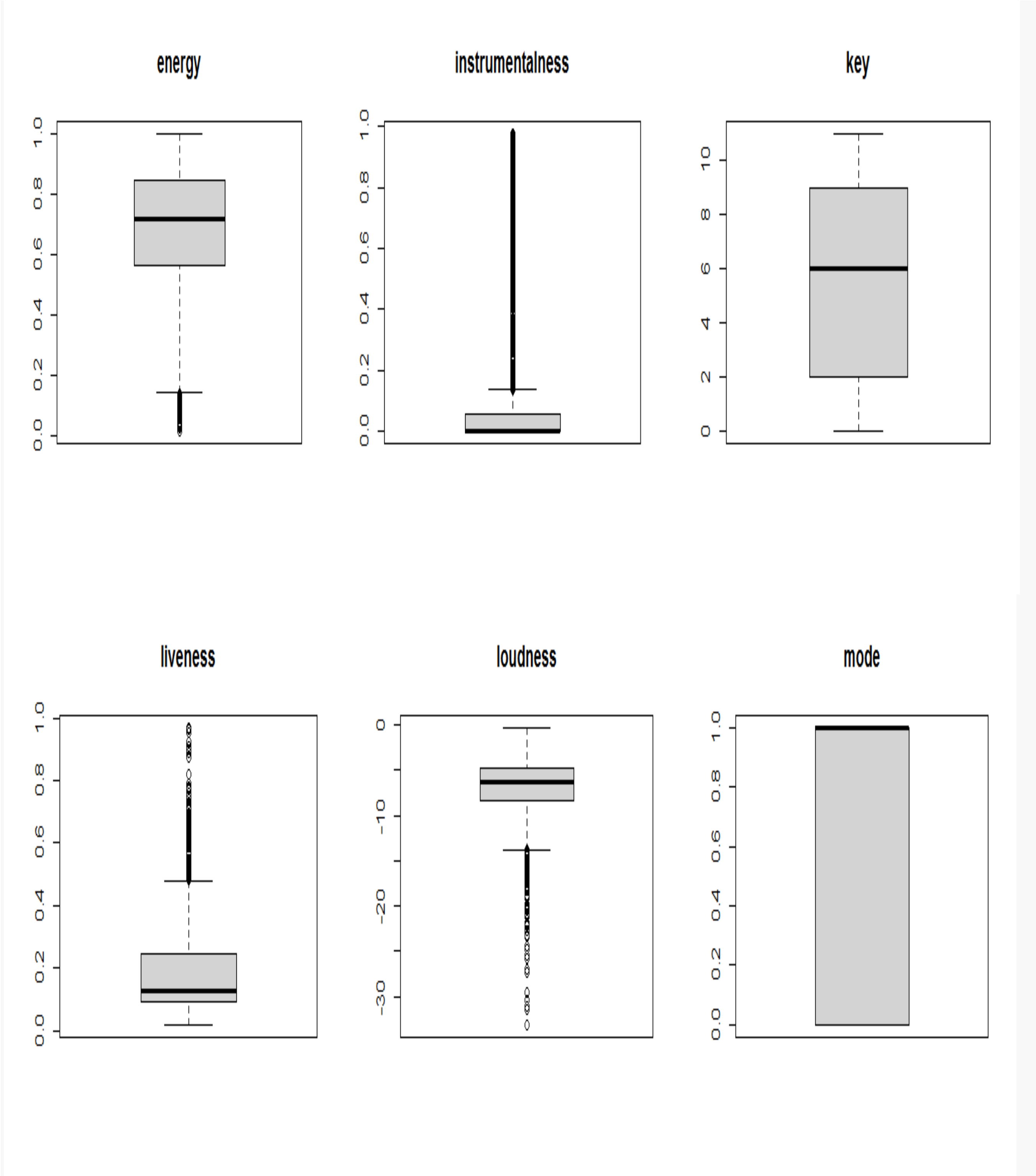
```r
summary(data1)
```

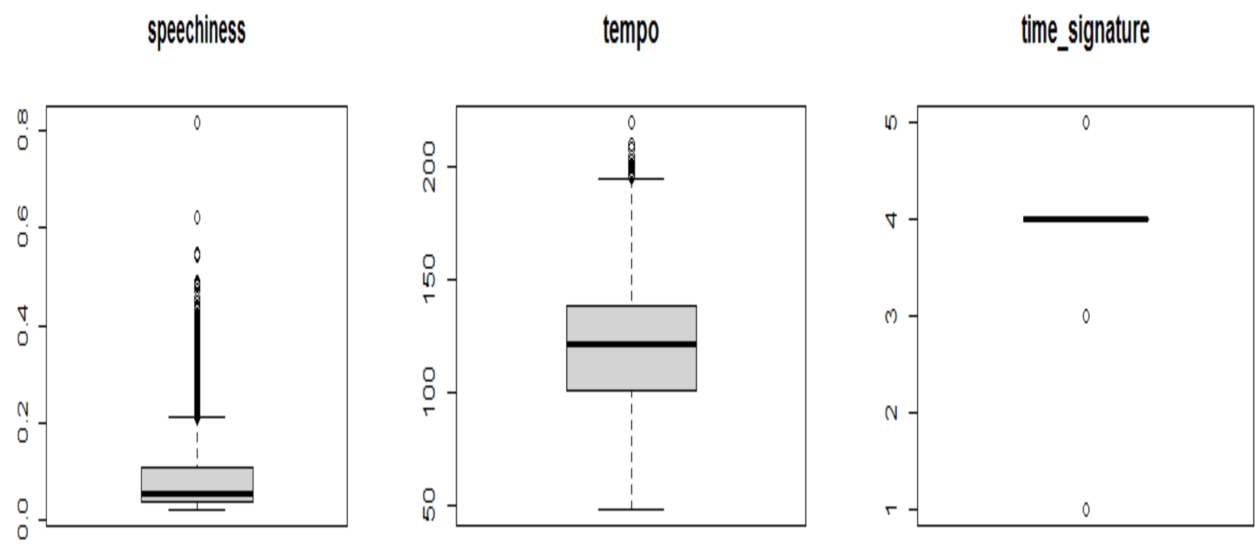
```
##   acousticness        danceability     duration_ms          energy
##  Min.   :0.0000028   Min.   :0.1220   Min.   :  16042   Min.   :0.0148
##  1st Qu.:0.0096300   1st Qu.:0.5140   1st Qu.: 200015   1st Qu.:0.5630
##  Median :0.0633000   Median :0.6310   Median : 229261   Median :0.7150
##  Mean   :0.1875900   Mean   :0.6184   Mean   : 246306   Mean   :0.6816
##  3rd Qu.:0.2650000   3rd Qu.:0.7380   3rd Qu.: 270333   3rd Qu.:0.8460
##  Max.   :0.9950000   Max.   :0.9840   Max.   :1004627   Max.   :0.9980


##  instrumentalness         key            liveness          loudness
##  Min.   :0.0000000   Min.   : 0.000   Min.   :0.0188   Min.   :-33.097
##  1st Qu.:0.0000000   1st Qu.: 2.000   1st Qu.:0.0923   1st Qu.: -8.394
##  Median :0.0000762   Median : 6.000   Median :0.1270   Median : -6.248
##  Mean   :0.1332855   Mean   : 5.343   Mean   :0.1908   Mean   : -7.086
##  3rd Qu.:0.0540000   3rd Qu.: 9.000   3rd Qu.:0.2470   3rd Qu.: -4.746
##  Max.   :0.9760000   Max.   :11.000   Max.   :0.9690   Max.   : -0.307


##       mode          speechiness           tempo        time_signature
##  Min.   :0.0000   Min.   :0.02310   Min.   : 47.86   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:0.03750   1st Qu.:100.19   1st Qu.:4.000
```

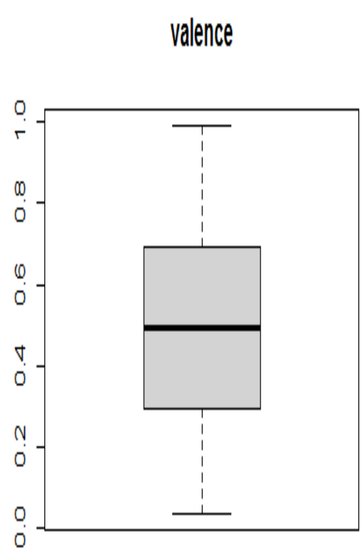
```
##  Median :1.0000   Median :0.05490   Median :121.43   Median :4.000
##  Mean   :0.6123   Mean   :0.09266   Mean   :121.60   Mean   :3.968
##  3rd Qu.:1.0000   3rd Qu.:0.10800   3rd Qu.:137.85   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :0.81600   Max.   :219.33   Max.   :5.000
##     valence           target
##  Min.   :0.0348   Min.   :0.0000
##  1st Qu.:0.2950   1st Qu.:0.0000
##  Median :0.4920   Median :1.0000
##  Mean   :0.4968   Mean   :0.5057
##  3rd Qu.:0.6910   3rd Qu.:1.0000
##  Max.   :0.9920   Max.   :1.0000
```

```
# There are no missing value in the data. The number of liked and disliked
songs is almost similar. The audio features are in range 0 to 1 except
Duration, Key, Loudness, Tempo and Time_signature.

##DATA VISUALIZATION
#Boxplot
for(i in 1:13){
  boxplot(data1[,i],main=names(data1)[i])

}
```

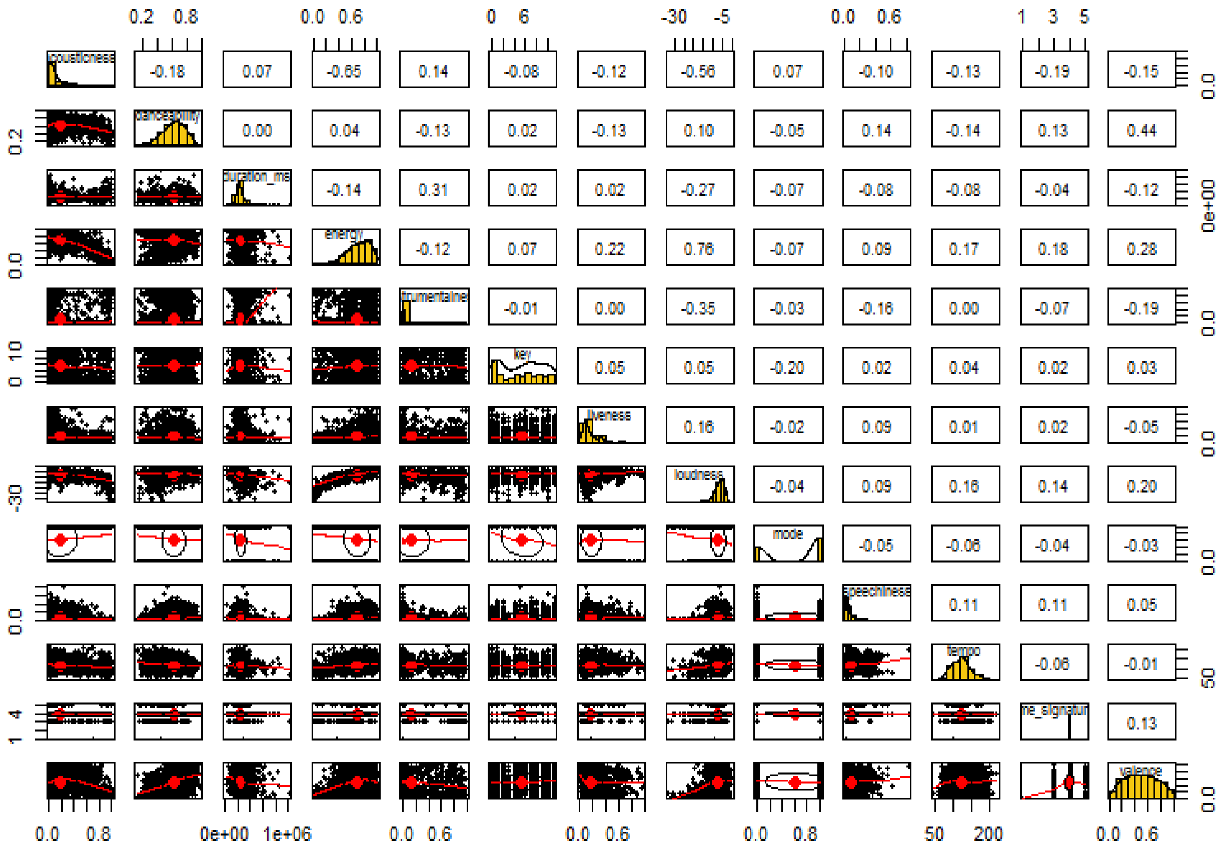



energy
instrumentalness
key
liveness
loudness
mode
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
-30
-20
-10


speechiness
0.0
0.2
0.4
0.6
0.8
tempo
50
100
150
200
time_signature
1
2
3
4
5


valence
0.0
0.2
0.4
0.6
0.8
1.0

```r
#Examining Correlation between variables
library(psych)

pairs.panels(data1[,1:13],method="pearson",hist.col=7,
             density=TRUE, ellipses=TRUE)
```



 # We can observe that Acousticness and Energy are negatively correlated i.e. the less acoustic the song more energetic it is and vice-versa. There is considerable negative correlation between loudness and acousticness. Also, increase in energy implies louder song and vice-versa. All the other factors may be considered uncorrelated.

```r
##Data Normalization
library(caret)

preproc=preProcess(data1[,c(3,6,8,11,12)], method=c("center","scale"))
norm=predict(preproc, data1)
```

```r
##splitting dataset in train and test set
library(caTools)

data_sample=sample(1:nrow(norm),size = ceiling(0.75*nrow(norm)),replace = FALSE)
train_data=norm[data_sample,]
test_data=norm[-data_sample,]

##Logistic Regression
full=glm(target~., family=binomial,data=train_data)
summary(full)

##
## Call:
## glm(formula = target ~ ., family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0596  -1.0577   0.4194   1.0461   2.2246
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.22779    0.47488  -4.691 2.71e-06 ***
## acousticness     -1.49426    0.30340  -4.925 8.43e-07 ***
## danceability      1.95395    0.43511   4.491 7.10e-06 ***
## duration_ms       0.23833    0.06447   3.697 0.000218 ***
## energy            0.65426    0.50816   1.288 0.197916
## instrumentalness  1.14780    0.24292   4.725 2.30e-06 ***
## key               0.01561    0.05654   0.276 0.782429
## liveness          0.48443    0.37877   1.279 0.200913
## loudness         -0.49367    0.10287  -4.799 1.60e-06 ***
## mode             -0.04547    0.11606  -0.392 0.695205
## speechiness       3.71157    0.68916   5.386 7.22e-08 ***
## tempo             0.13035    0.05859   2.225 0.026094 *
## time_signature   -0.01742    0.05859  -0.297 0.766290
## valence           0.68571    0.27401   2.503 0.012331 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2097.1  on 1512  degrees of freedom
## Residual deviance: 1880.9  on 1499  degrees of freedom
## AIC: 1908.9
## Number of Fisher Scoring iterations: 4
```

#The deviance residuals are centered around zero and are symmetrically distributed, which is good for model. The song attributes like aucousticness, danceability, duration_ms, instrumentallness, loudness and speechiness are significant for α=0.001. Therefore applying step-wise logistic regression we get the following result.

```
library(MASS)
step=stepAIC(full,trace=FALSE)
summary(step)
```

```
##
## Call:
## glm(formula = target ~ acousticness + danceability + duration_ms +
##     instrumentalness + liveness + loudness + speechiness + tempo +
##     valence, family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0598  -1.0535   0.4199   1.0440   2.1540
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.77610    0.27768  -6.396 1.59e-10 ***
## acousticness     -1.64205    0.27695  -5.929 3.05e-09 ***
## danceability      1.80802    0.41813   4.324 1.53e-05 ***
## duration_ms       0.24359    0.06409   3.801 0.000144 ***
## instrumentalness  1.23011    0.23475   5.240 1.60e-07 ***
## liveness          0.54619    0.37587   1.453 0.146194
## loudness         -0.40702    0.07742  -5.257 1.46e-07 ***
## speechiness       3.78276    0.68703   5.506 3.67e-08 ***
## tempo             0.13308    0.05829   2.283 0.022436 *
## valence           0.80271    0.25693   3.124 0.001783 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2097.1  on 1512  degrees of freedom
## Residual deviance: 1883.0  on 1503  degrees of freedom
## AIC: 1903
##
## Number of Fisher Scoring iterations: 4
```

```
step$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## target ~ acousticness + danceability + duration_ms + energy +
##     instrumentalness + key + liveness + loudness + mode + speechiness +
##     tempo + time_signature + valence
##
## Final Model:
## target ~ acousticness + danceability + duration_ms + instrumentalness +
##     liveness + loudness + speechiness + tempo + valence
##
##
##              Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                                     1499   1880.949 1908.949
## 2            - key  1 0.07625517      1500   1881.025 1907.025
## 3 - time_signature  1 0.08448598      1501   1881.110 1905.110
## 4           - mode  1 0.19458315      1502   1881.304 1903.304
## 5         - energy  1 1.66648193      1503   1882.971 1902.971
```

```
res_step=predict(step,test_data,target="response")
(table(actualvalue=test_data$target,predictedvalue=res_step>0.5))
```

```
##            predictedvalue
## actualvalue FALSE TRUE
##           0   219   34
##           1   149  102
```

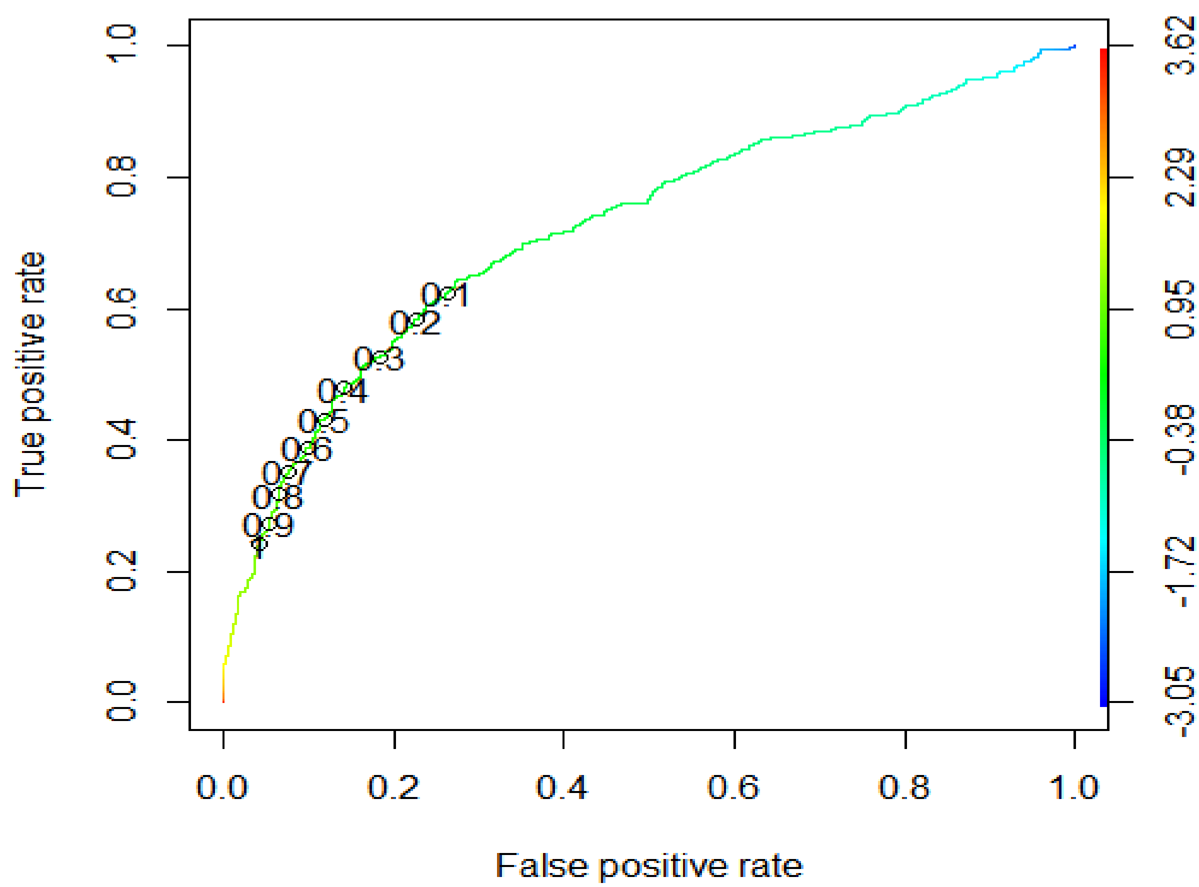
```
##(219+102)/(219+34+149+102)
##0.63690476
```

#Applying Step-wise regression we get the final model as,
target ~ acousticness + danceability + duration_ms + instrumentalness + liveness + loudness + speechiness + tempo + valence

#We notice decrease in the residual Deviance and AIC value which is a good indication.

#The accuracy of the model is 63.6905%. We observe that 149 songs which are disliked and classified as liked (false positive). Therefore we can change the threshold value (which is set to 0.5) to 0.3 and check the changes occurred. We changed the threshold to 0.3 after observing the following AUC-ROC curve. In order to decrease the false positive rate we may increase true positive rate. This may alter the accuracy.

```
res=predict(step,train_data,target="response")
library(ROCR)

ROCPred=prediction(res,train_data$target)
ROCPref=performance(ROCPred,"tpr","fpr")
plot(ROCPref,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))
```




```
res_step=predict(step,test_data,target="response")
(table(actualvalue=test_data$target,predictedvalue=res_step>0.3))

##             predictedvalue
## actualvalue FALSE TRUE
##           0   207   46
##           1   129  122

##(207+122)/(207+122+129+46)
```

```
##0.65277778

#We find that there is increase in accuracy by 1.587%. Therefore the logistic
 model is optimized.

##NEURAL NETWORKS
library(neuralnet)

library(tictoc)

tic()
nn=neuralnet(target~.-c("key","time_signature","energy","mode") ,data=train_data,hid
den=5,act.fct="logistic",
linear.output=FALSE)
plot(nn)
toc()

## 153.59 sec elapsed

pred=predict(nn,test_data)
table(test_data$target,pred[,1]>0.5)

##
##     FALSE TRUE
##   0   177   76
##   1    75  176

#Applying Neural networks to the dataset was much better idea than the
logistic regression. The accuracy of 70.04% was obtained. However, it took
much time to train Neural network.

##Random Forest
train_data$target <- as.character(train_data$target)
train_data$target <- as.factor(train_data$target)
library(randomForest)

tic()
rf=randomForest(target~acousticness + danceability + duration_ms + energy +
                  instrumentalness + liveness + loudness + mode + speechiness +
                  valence,data=train_data)
toc()

## 3.63 sec elapsed

rf
```

```
##
## Call:
##  randomForest(formula = target ~ acousticness + danceability +  duration_ms +
       energy + instrumentalness + liveness + loudness + mode + speechiness +
       valence, data = train_data)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 21.94%
## Confusion matrix:
##     0   1 class.error
## 0 580 164   0.2204301
## 1 168 601   0.2184655
```

#The random forest model obtained the accuracy of 78.056%. It is faster than Neural network.