Spotify Data Analysis

Introduction

- The Spotify Dataset comes from Spotify via the spotifyr package.
- This package was authorized to make it easier to get either your own data or general metadata around songs from Spotify's API.
- We want to answer the questions:
- What characteristics of a song can determine its popularity?
- Which genres and subgenres predict popularity of the songs?
- Data Preparation, EDA and Modelling.
- **Linear Regression, KNN, Logistic Regression, SVM** and **Tree Models.**
- Spotify will be able to provide more accurate predictions of a new song's potential popularity.

Variable Name	Description
track_id	unique ID
track_name	Song Name
track_artist	Song Artist
track_popularity	Song Popularity (0-100) where higher is better
track_album_id	Album unique ID
track_album_name	Song album name
track_album_release_date	Date when album released
playlist_name	Name of playlist
playlist_id	Playlist ID
playlist_genre	Playlist genre
playlist_subgenre	Playlist subgenre
danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements. A value of 0.0 is least danceable and 1.0 is most danceable.
energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.
key	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation .
loudness	The overall loudness of a track in decibels (dB).
mode	Mode indicates the modality (major or minor) of a track
speechiness	Speechiness detects the presence of spoken words in a track.
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
instrumentalness	Predicts whether a track contains no vocals.
liveness	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.
valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive.
tempo	The overall estimated tempo of a track in beats per minute (BPM).
duration_ms	Duration of song in milliseconds

Packages Required

- **Tidyverse**: assists with data import, tidying, manipulation, and data visualization.
- **ggplot2**: package for producing statistical, or data, graphics.
- **kknn**: performs k-nearest neighbor classification.
- **corrplot**: graphical display of a correlation matrix, confidence interval.
- **readr**: provides a fast way to read rectangular data.
- **rpart**: implements the classification and regression tree algorithm (CART).
- **rpart.plot**: An Enhanced Plotting Package for rpart.

Data Preparation

- **Data loading**: `spotify <- read.csv("spotify.csv")`
- **Data Examination**: `head(spotify)`

```
### Checking dimension of Data
dim(spotify)
```

```
## [1] 32833    23
```

```
#### Checking column name
names(spotify)
```

```
##  [1] "track_id"                "track_name"
##  [3] "track_artist"            "track_popularity"
##  [5] "track_album_id"          "track_album_name"
##  [7] "track_album_release_date" "playlist_name"
##  [9] "playlist_id"             "playlist_genre"
## [11] "playlist_subgenre"       "danceability"
## [13] "energy"                  "key"
## [15] "loudness"                "mode"
## [17] "speechiness"             "acousticness"
## [19] "instrumentalness"        "liveness"
## [21] "valence"                 "tempo"
## [23] "duration_ms"
```

Data Cleaning

- ```
  ### Checking structure of Data
  str(spotify)
  ```
- We need to change *track_album_release_date* from **chr** to **date** variable
- We will also change *playlist_genre* from **chr** to **factor**, for future plotting
- ```
  #summary statistics
  summary(spotify)
  ```
  : Min, Q1, Median, Mean, Q3 and Max of each variable
- We can see that there are probably some **outliers**
- *duration_ms* has a Max of **517810**
- *tempo* has a Max of **239.44**
- Since *track_popularity* is continuous we assume **lm** and **regression trees** are the best model

STR()

```
'data.frame':   32833 obs. of  23 variables:
 $ track_id                : chr  "6f807x0ima9a1j3VPbc7VN" "0r7CVbZTWZgbTCYdfa2P31" "1z1Hg7Vb0AhHDiEmnDE79l"
"75FpbthrwQmzHlBJLuGdC7" ...
 $ track_name              : chr  "I Don't Care (with Justin Bieber) - Loud Luxury Remix" "Memories - Dillon Francis Remix"
"All the Time - Don Diablo Remix" "Call You Mine - Keanu Silva Remix" ...
 $ track_artist            : chr  "Ed Sheeran" "Maroon 5" "Zara Larsson" "The Chainsmokers" ...
 $ track_popularity        : int  66 67 70 60 69 67 62 69 68 67 ...
 $ track_album_id          : chr  "2oCs0DGTsRO98Gh5ZSl2Cx" "63rPSO264uRjW1X5E6cWv6" "1HoSmj2eLcsrR0vE9gThr4"
"1nqYsOef1yKKuGOVchbsk6" ...
 $ track_album_name        : chr  "I Don't Care (with Justin Bieber) [Loud Luxury Remix]" "Memories (Dillon Francis Remix)"
"All the Time (Don Diablo Remix)" "Call You Mine - The Remixes" ...
 $ track_album_release_date: chr  "2019-06-14" "2019-12-13" "2019-07-05" "2019-07-19" ...
 $ playlist_name           : chr  "Pop Remix" "Pop Remix" "Pop Remix" "Pop Remix" ...
 $ playlist_id             : chr  "37i9dQZF1DXcZDD7cfEKhW" "37i9dQZF1DXcZDD7cfEKhW" "37i9dQZF1DXcZDD7cfEKhW"
"37i9dQZF1DXcZDD7cfEKhW" ...
 $ playlist_genre          : chr  "pop" "pop" "pop" "pop" ...
 $ playlist_subgenre       : chr  "dance pop" "dance pop" "dance pop" "dance pop" ...
 $ danceability            : num  0.748 0.726 0.675 0.718 0.65 0.675 0.449 0.542 0.594 0.642 ...
 $ energy                  : num  0.916 0.815 0.931 0.93 0.833 0.919 0.856 0.903 0.935 0.818 ...
 $ key                     : int  6 11 1 7 1 8 5 4 8 2 ...
 $ loudness                : num  -2.63 -4.97 -3.43 -3.78 -4.67 ...
 $ mode                    : int  1 1 0 1 1 1 0 0 1 1 ...
 $ speechiness             : num  0.0583 0.0373 0.0742 0.102 0.0359 0.127 0.0623 0.0434 0.0565 0.032 ...
 $ acousticness            : num  0.102 0.0724 0.0794 0.0287 0.0803 0.0799 0.187 0.0335 0.0249 0.0567 ...
 $ instrumentalness        : num  0.00 4.21e-03 2.33e-05 9.43e-06 0.00 0.00 0.00 4.83e-06 3.97e-06 0.00 ...
 $ liveness                : num  0.0653 0.357 0.11 0.204 0.0833 0.143 0.176 0.111 0.637 0.0919 ...
 $ valence                 : num  0.518 0.693 0.613 0.277 0.725 0.585 0.152 0.367 0.366 0.59 ...
 $ tempo                   : num  122 100 124 122 124 ...
 $ duration_ms             : int  194754 162600 176616 169093 189052 163049 187675 207619 193187 253040 ...
```

SUMMARY()

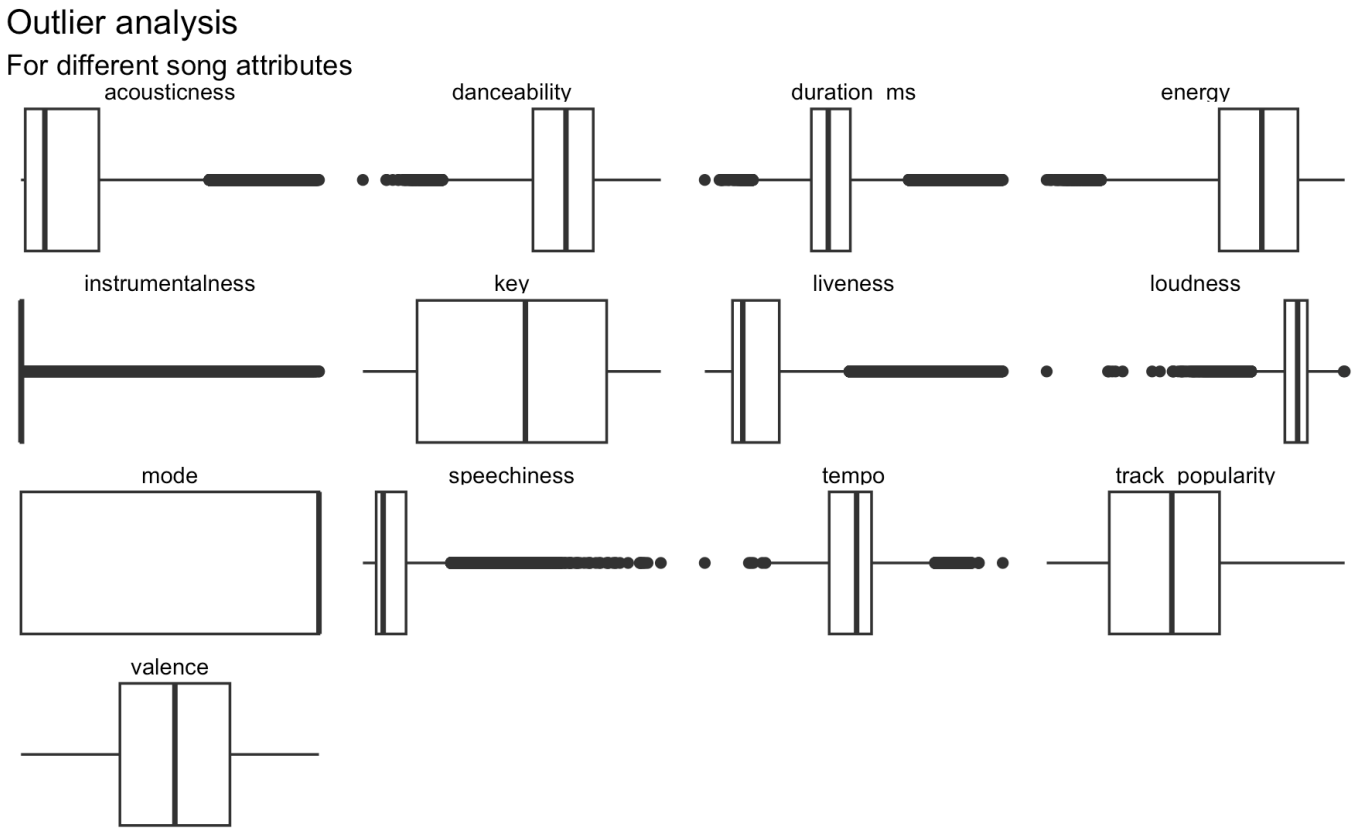
```
playlist_genre playlist_subgenre   danceability        energy
edm  :6043     Length:32833       Min.   :0.0000   Min.   :0.000175
latin:5155     Class :character   1st Qu.:0.5630   1st Qu.:0.581000
pop  :5507     Mode  :character   Median :0.6720   Median :0.721000
r&b  :5431                        Mean   :0.6548   Mean   :0.698619
rap  :5746                        3rd Qu.:0.7610   3rd Qu.:0.840000
rock :4951                        Max.   :0.9830   Max.   :1.000000

      key            loudness           mode          speechiness
Min.   : 0.000   Min.   :-46.448   Min.   :0.0000   Min.   :0.0000
1st Qu.: 2.000   1st Qu.: -8.171   1st Qu.:0.0000   1st Qu.:0.0410
Median : 6.000   Median : -6.166   Median :1.0000   Median :0.0625
Mean   : 5.374   Mean   : -6.720   Mean   :0.5657   Mean   :0.1071
3rd Qu.: 9.000   3rd Qu.: -4.645   3rd Qu.:1.0000   3rd Qu.:0.1320
Max.   :11.000   Max.   :  1.275   Max.   :1.0000   Max.   :0.9180

 acousticness    instrumentalness       liveness         valence
Min.   :0.0000   Min.   :0.0000000   Min.   :0.0000   Min.   :0.0000
1st Qu.:0.0151   1st Qu.:0.0000000   1st Qu.:0.0927   1st Qu.:0.3310
Median :0.0804   Median :0.0000161   Median :0.1270   Median :0.5120
Mean   :0.1753   Mean   :0.0847472   Mean   :0.1902   Mean   :0.5106
3rd Qu.:0.2550   3rd Qu.:0.0048300   3rd Qu.:0.2480   3rd Qu.:0.6930
Max.   :0.9940   Max.   :0.9940000   Max.   :0.9960   Max.   :0.9910

    tempo          duration_ms
Min.   :  0.00   Min.   :  4000
1st Qu.: 99.96   1st Qu.:187819
Median :121.98   Median :216000
Mean   :120.88   Mean   :225800
3rd Qu.:133.92   3rd Qu.:253585
Max.   :239.44   Max.   :517810
```

Data Cleaning

- **Duplicates** : `dups_id <- sum(duplicated(spotify$track_id))`
- **4477** observations have **duplicates** on *track_id*
- We remove them
- **Missing values**: `sum(is.na(spotify))`
- **1693** missing values in this dataset
- We will not remove them

Data Cleaning

- Visual **Boxplot exploration** to identify outliers

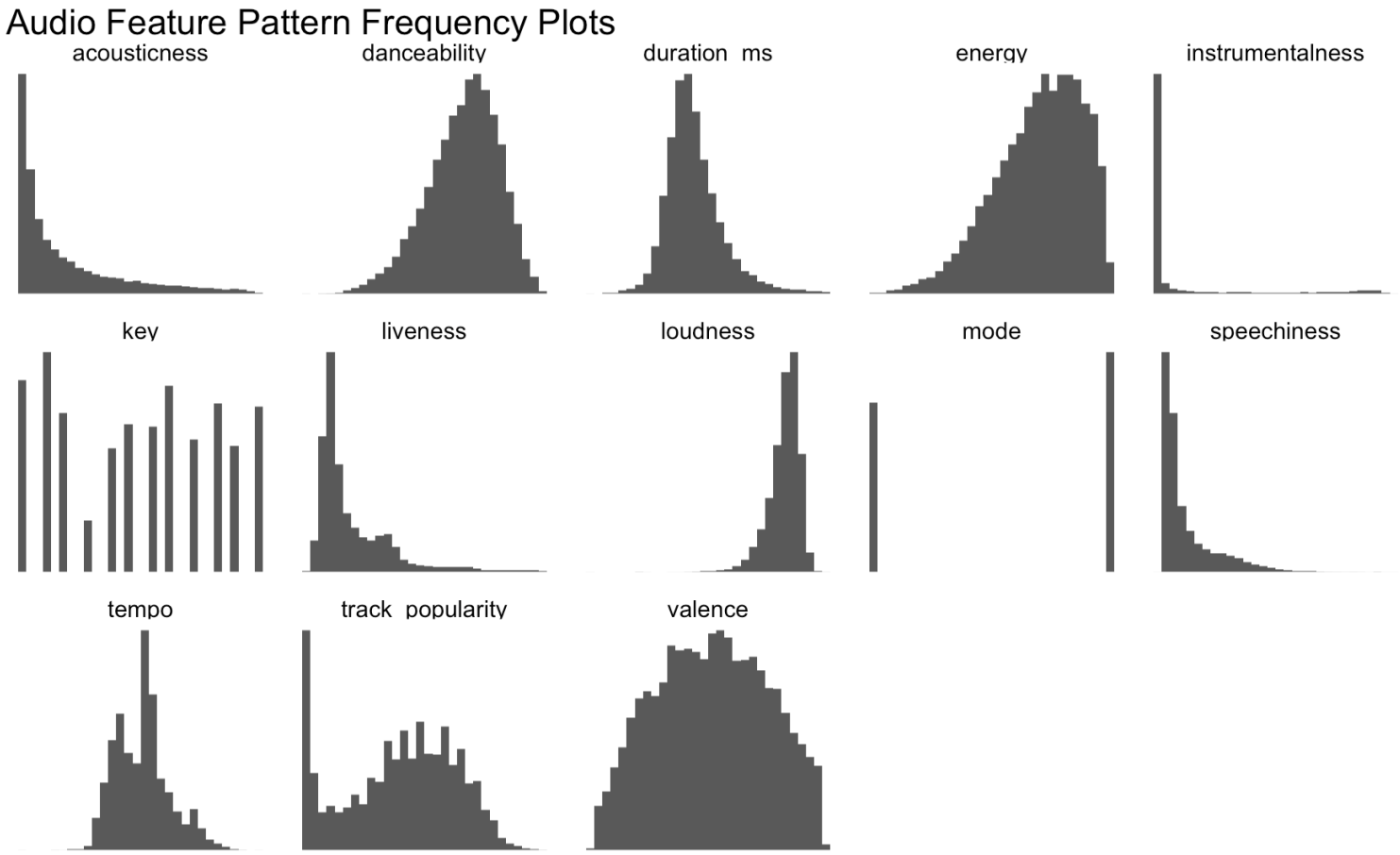


- We will **truncate** *energy speechiness, acousticness, instrumentalness* and *liveness*

- We will **winsorize** *loudness, tempo* and *duration*

Exploratory Data Analysis

- Visual **histogram exploration** to understand the data set

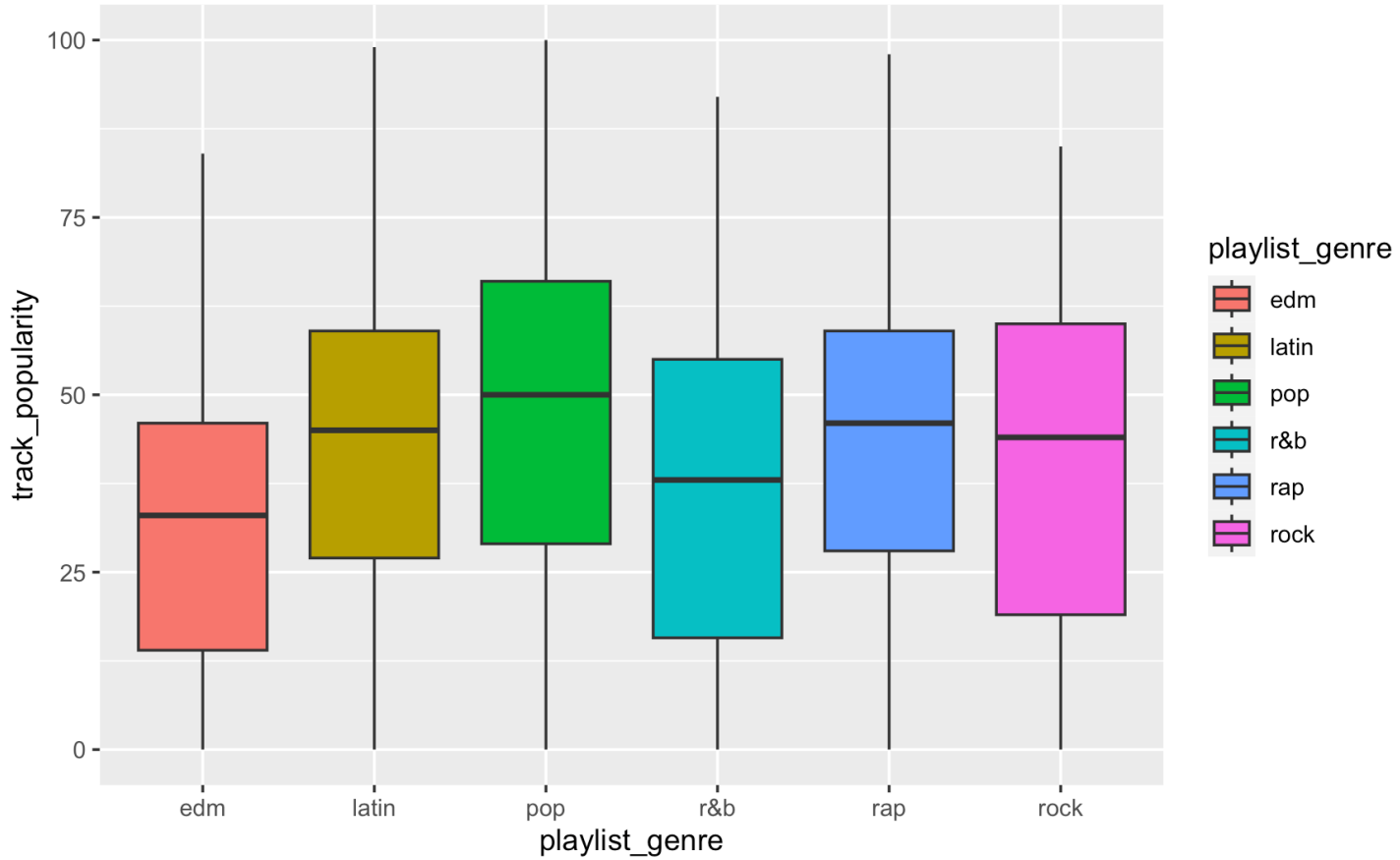


- *Duration* , *Valence* and *tempo* are **normally distributed**
- *Danceability*, *Energy* and *Loudness* are **left-skewed**
- *Acousticness*, *Liveness* and *Speechiness* are **right-skewed**
- In *track_popularity* the max is around 50-60, and it is normally distributed. There are a lot of zero values, possible **missing values or outliers**

Exploratory Data Analysis

- **Boxplot** to plot different genres vs popularity

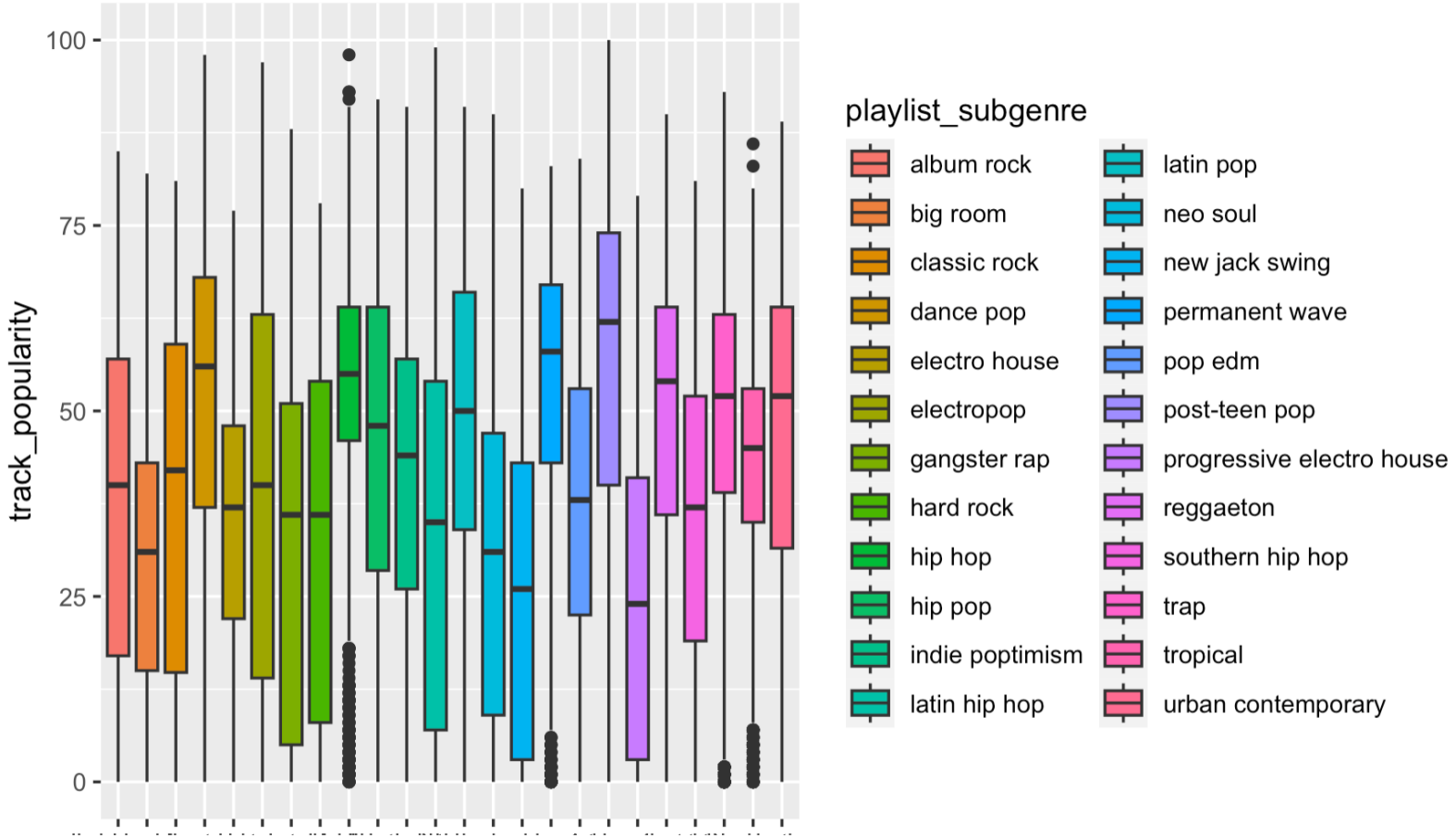
```
ggplot(spotify,aes(x = playlist_genre, y = track_popularity, fill = playlist_genre)) +geom_boxplot()
```



- *Pop* is the **most popular** genre
- Followed by *latin* and *rock*

Exploratory Data Analysis

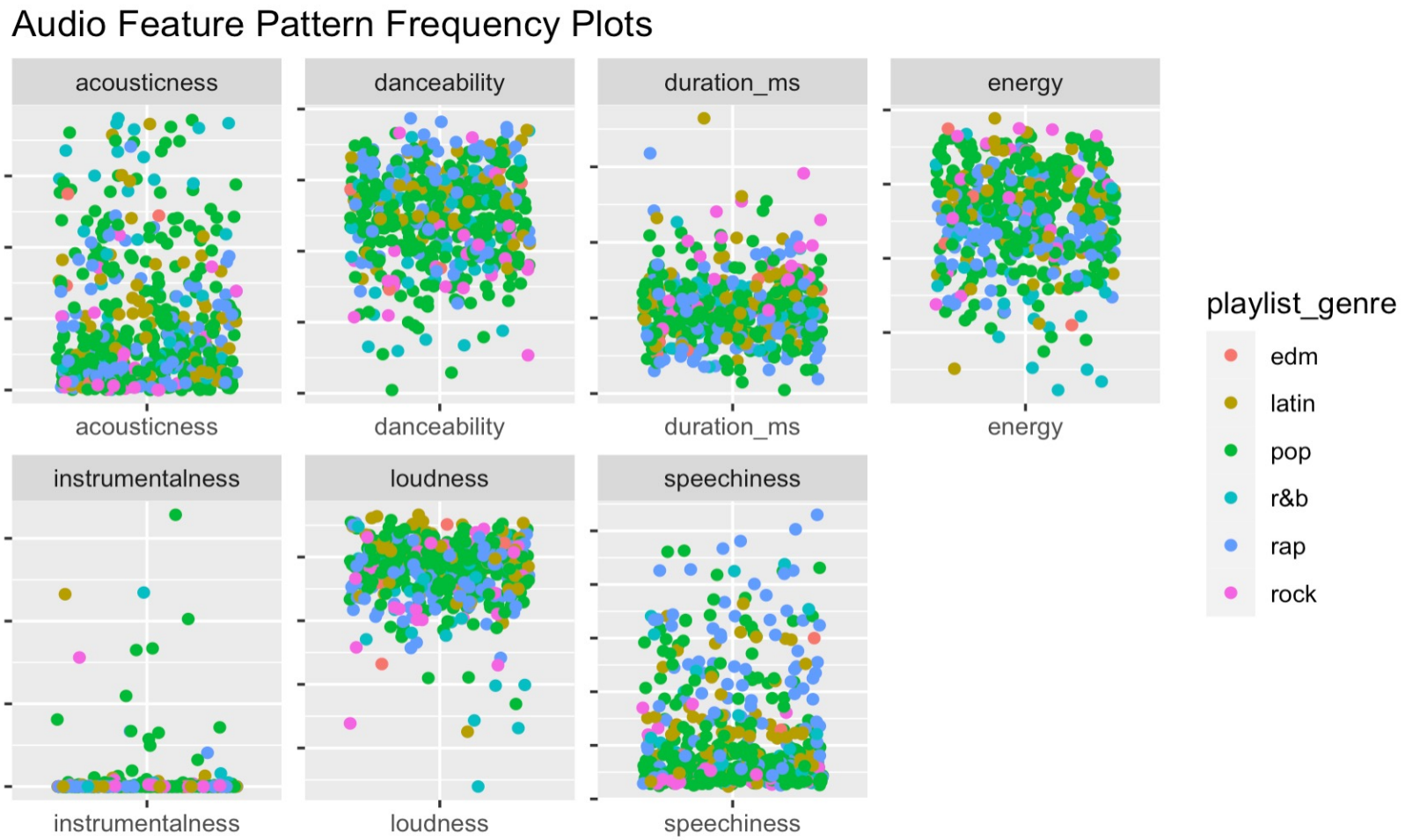
- **Boxplot** to plot different subgenres vs popularity



- *Post-Teen-Pop* is the **most popular** subgenre
- Followed by *Dance-Pop* and *Permanent-Wave*

Exploratory Data Analysis

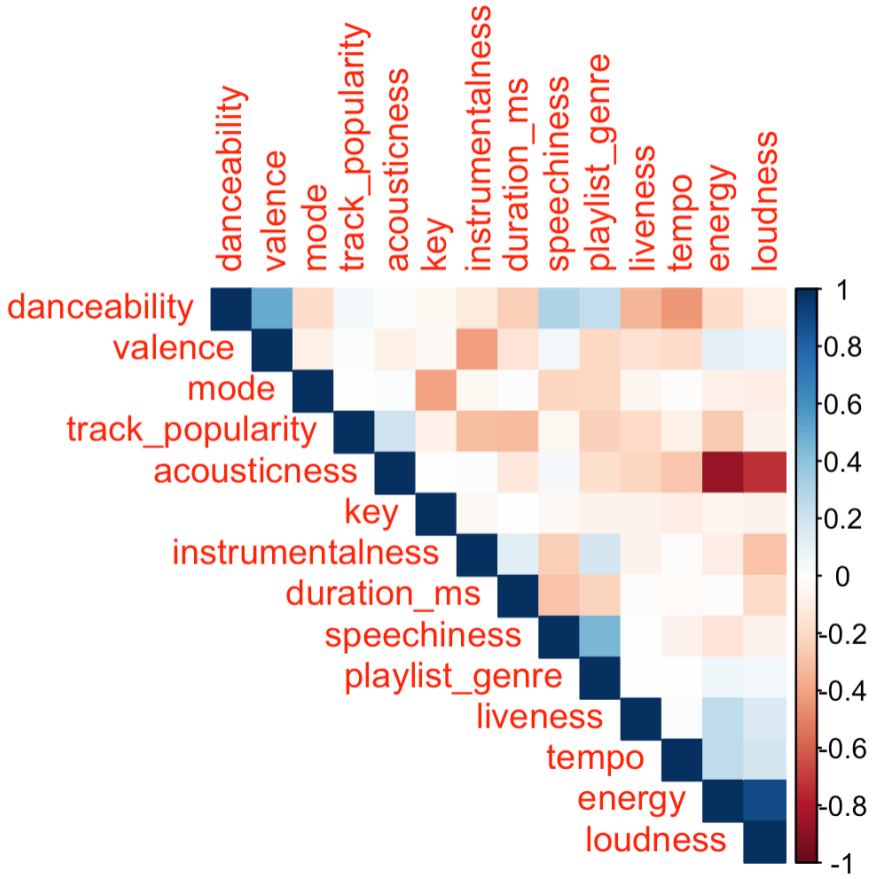
- **Scatterplots** comparing parameters to popularity



- These graphics contain data from the **500 most popular songs**
- Since *Instrumentalness, Speechiness* and *Loudness* have a smaller interval, we would consider them to **best predict** *track_popularity.*

Exploratory Data Analysis

- **Correlation Plot**



- The variables that **mostly correlate** with *track_popularity* are the following:

1. *Duration* (**-0.1396**)

2. *Instrumentalness* **(-0.1244**)

3. *Energy* (**-0.1036**)

4. *Acousticness* (**0.0917**)

5. *playlist_genre* (**-0.0734**)

Modelling

- **1st Linear Regression Model**

- We first split the data into **training (70%)** and **testing set (30%)**
- We build regression model with **continuous** variables
- All variables are significant, **except** *Key*
- Low **R^2 = 0.06003** , only 6% of the of the data's **variability** can be explained by the regression model
- *instrumentalness, speechiness* and *duration_ms* have **highest coefficients**
- **Training MSE=** 529.94**; Testing MSE=** 527.51

```
Call:
lm(formula = track_popularity ~ danceability + energy + key +
    loudness + mode + speechiness + acousticness + instrumentalness +
    liveness + valence + tempo + duration_ms, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-54.757 -17.360   2.935  18.119  60.604

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       6.684e+01  2.045e+00  32.682  < 2e-16 ***
danceability      4.170e+00  1.288e+00   3.237  0.00121 **
energy           -2.318e+01  1.461e+00 -15.864  < 2e-16 ***
key               1.398e-02  4.595e-02   0.304  0.76088
loudness          1.132e+00  7.802e-02  14.512  < 2e-16 ***
mode              7.590e-01  3.357e-01   2.261  0.02378 *
speechiness      -7.397e+00  1.657e+00  -4.464 8.09e-06 ***
acousticness      4.934e+00  8.895e-01   5.547 2.95e-08 ***
instrumentalness -9.426e+00  7.437e-01 -12.676  < 2e-16 ***
liveness         -4.420e+00  1.077e+00  -4.104 4.08e-05 ***
valence           1.997e+00  7.861e-01   2.540  0.01110 *
tempo             2.808e-02  6.261e-03   4.485 7.33e-06 ***
duration_ms      -4.277e-05  2.726e-06 -15.689  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.03 on 19836 degrees of freedom
Multiple R-squared:  0.06003,   Adjusted R-squared:  0.05946
F-statistic: 105.6 on 12 and 19836 DF,  p-value: < 2.2e-16
```

Modelling

- **2nd Linear Regression Model**

- We build regression model with **categorical** variables
- *Genre* and *Subgenre*
- All variables are significant, **except** *classic rock* and *hip hop*
- Bigger **R^2 = 0.1362** , 13.62% of the of the data's **variability** can be explained by the regression model
- *pop, rap , r&b , jack swing*, *electropop* and *neo soul* have **highest coefficients**
- **Training MSE=** 486.43**; Testing MSE=** 485.8

```
Call:
lm(formula = track_popularity ~ playlist_genre + playlist_subgenre,
    data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-55.418 -16.339   2.905  16.931  60.098

Coefficients: (5 not defined because of singularities)
                                          Estimate Std. Error t value
(Intercept)                                23.9894     0.6853  35.005
playlist_genrelatin                        18.0795     1.0338  17.489
playlist_genrepop                          31.4286     1.0654  29.501
playlist_genrer&b                          22.2155     1.0313  21.541
playlist_genrerap                          24.4461     1.0227  23.902
playlist_genrerock                         13.5668     1.0741  12.631
playlist_subgenrebig room                   4.4078     1.0705   4.117
playlist_subgenreclassic rock              -0.2592     1.1543  -0.225
playlist_subgenredance pop                 -3.0791     1.1009  -2.797
playlist_subgenreelectro house              9.7948     0.9814   9.981
playlist_subgenreelectropop               -16.3501     1.0989 -14.878
playlist_subgenregangster rap             -15.9361     1.0535 -15.127
playlist_subgenrehard rock                 -4.2979     1.1230  -3.827
playlist_subgenrehip hop                    4.6597     1.0591   4.400
playlist_subgenrehip pop                   -1.8091     1.2060  -1.500
playlist_subgenreindie poptimism          -14.2180     1.0637 -13.367
playlist_subgenrelatin hip hop             -9.5706     1.0786  -8.873
playlist_subgenrelatin pop                  4.3781     1.1083   3.950
playlist_subgenreneo soul                 -16.3026     1.0344 -15.760
playlist_subgenrenew jack swing           -19.9120     1.1214 -17.757
playlist_subgenrepermanent wave            14.2161     1.1811  12.036
playlist_subgenrepop edm                   12.5587     1.0861  11.563
playlist_subgenrepost-teen pop                 NA         NA      NA
playlist_subgenreprogressive electro house     NA         NA      NA
playlist_subgenrereggaeton                  4.2245     1.2535   3.370
```

Modelling: Linear Models

- **3rd Model (with interaction)**

```
lm_model_int <- lm(track_popularity ~ danceability + loud
ness + danceability*loudness, data = spotify)
```

- **Lower R^2 =** 0.004887
- Interaction and variables **are significant**
- **Training MSE=** 567.09
- **Testing MSE=** 554.33

- **4th Model (no interaction)**

```
lm_model_no_int <- lm(track_popularity ~ danceability + l
oudness , data = spotify)
summary(lm_model_no_int)
```

- **R^2 =** 0.003439
- Both variables **are significant**
- **Training MSE=** 566.13
- **Testing MSE=** 555.23

Modelling: KNN

- **1st Model (k=5)**

```
spotify_knn_model <- kknn(track_popularity ~ danceability
+ energy + key + loudness + mode + speechiness + acoustic
ness + instrumentalness + liveness + valence + tempo + du
ration_ms, train = train_data, test = train_data, k = 5)
```

- **Training MSE=** 194.904
- **Testing MSE=** 687.024

- **2nd Model (k=20)**

```
spotify_knn_model_20 <- kknn(track_popularity ~ danceabil
ity + energy + key + loudness + mode + speechiness + acou
sticness + instrumentalness + liveness + valence + tempo
+ duration_ms, train = train_data, test = train_data, k =
20)
```

- **Training MSE=** 389.061
- **Testing MSE=** 556.982

Modelling: KNN > Different Variables

- **3rd Model (k=5; unstandardized)**

```
stand_knn <- kknn(track_popularity ~ danceability + energ
y + loudness + speechiness + instrumentalness + liveness
+ valence + duration_ms, train = train_new.x, test = trai
n_new.x, k = 5)
```

- **Training MSE=** 217.942
- **Testing MSE=** 213.310

- **4th Model (k=5; standardized)**

```
knn_unstand <- kknn(track_popularity ~  danceability + en
ergy + loudness + speechiness + instrumentalness + livene
ss + valence + duration_ms, train = train_new.x, test = t
rain_new.x, k = 5, scale = FALSE)
```

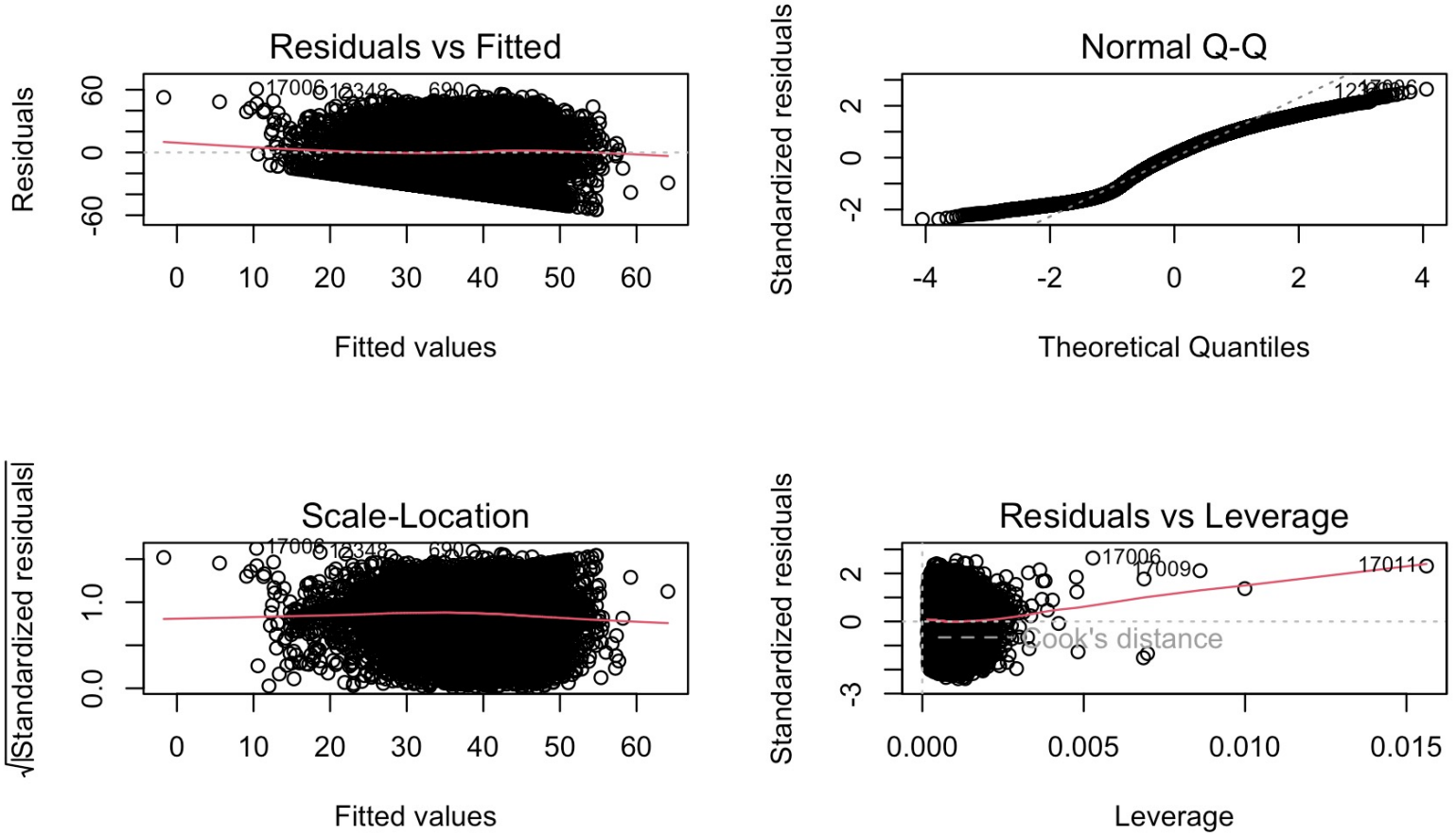
- **Training MSE=** 206.747
- **Testing MSE=** 201.450

Modelling

- **Diagnostic Plots**

```{r}
# diagnostic plot
par(mfrow = c(2, 2))
plot(lm_model)
```

- From the **diagnostic plot**, we see that the model **does not** meet all four assumptions
- **linear regression model** is not the best fit for this data.



Modelling

- **Comparing Results**

	Lm 1 (Cont)	Lm 2 (Cat)	Lm 3 (Int)	Lm 4 (No Int)	Knn (k=5) (Cont)	Knn (k=20) (Cont)	Knn(k=5) (diff. var) / unst.	Knn(k=5) (diff. var) / stand.
Training MSE	529.94	486.43	567.09	566.13	**194.904**	389.061	217.942	206.747
Testing MSE	**527.51**	485.8	554.33	555.23	687.024	556.982	213.310	201.450

- Out of the **continuous** models (with same variables) , **LM** had the **best Out-of-Sample MSE**
- Out of the continuous models (with same variables), **KNN** had the **best In-Sample MSE (with k=5)**
- In KNN **standardized** model performed better, but small difference between both (they have lower **MSE** , because we used less variables on these models)

- **Linear Model** is still **better** model that KNN
- Smallest Testing MSE
- Better for **continuous** dependent variables

Modelling : Logistic Regression

- **1st Model (num. var)**

- For **logistic Regression** we have to convert *track_popularity* **(dependent variable)** to **categorical**

```
spotify_copy$track_popularity <- ifelse(spotify_copy$track_popularity >=70, 1 , 0)
```

- We are substituting values above 70 with **1** (popular) and below 70 with **0** (unpopular)

Modelling

- **1st Model (num. var)**

- *Key, danceability* and *acousticness* are **not significant**
- *Key* has the highest coefficient with -**9.644**, followed by *speechiness* and *liveness*
- **Training AUC = 0.6855; Testing AUC = 0.6610**
- We define **optimal p.cut = 0.16**

TRAINING	**TESTING**
MR:0.1534	**MR: 0.1523**
FPR:0.08957	**FPR: 0.0871**
FNR:0.7757	**FNR: 0.7909**
cost: 0.4418	**Cost: 0.4457**

```
Call:
glm(formula = track_popularity ~ danceability + energy + key +
    loudness + mode + speechiness + acousticness + instrumentaln
    liveness + valence + tempo + duration_ms, family = binomial,
    data = spotify_copy.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1331  -0.5000  -0.4123  -0.2795   4.2276

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       1.066e+00  3.085e-01   3.455 0.000551 ***
danceability      3.319e-01  1.902e-01   1.745 0.081027 .
energy           -3.095e+00  2.169e-01 -14.269  < 2e-16 ***
key              -9.644e-03  6.523e-03  -1.478 0.139311
loudness          2.076e-01  1.310e-02  15.843  < 2e-16 ***
mode              1.286e-01  4.805e-02   2.676 0.007446 **
speechiness      -7.592e-01  2.401e-01  -3.162 0.001568 **
acousticness      1.473e-01  1.228e-01   1.199 0.230523
instrumentalness -3.071e+00  2.814e-01 -10.916  < 2e-16 ***
liveness         -4.376e-01  1.710e-01  -2.558 0.010518 *
valence           4.779e-01  1.159e-01   4.125 3.71e-05 ***
tempo             2.652e-03  8.573e-04   3.093 0.001980 **
duration_ms      -1.883e-06  4.619e-07  -4.076 4.57e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Modelling

- **2nd Model (cat. var)**

- *big room, classic rock, hip pop,* and *electro house* are **not significant**
- *Pop* has the highest coefficient with 4.9005, followed by *r&b* and *rap*
- **Training AUC = 0.7594 ; Testing AUC = 0.754**
- We define **optimal p.cut = 0.17**

TRAINING	**TESTING**
MR:0.1513	**MR: 0.1571**
FPR:0.1047	**FPR: 0.1047**
FNR: 0.6695	**FNR: 0.6692**
cost: 0.4003	**Cost: 0.4053**

```
Call:
glm(formula = track_popularity ~ playlist_genre + playlist_subgenre,
    family = binomial, data = spotify_copy.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9468  -0.5258  -0.3472  -0.1463   3.3090

Coefficients: (5 not defined because of singularities)
                                          Estimate Std. Error z value
(Intercept)                               -5.4706     0.4474 -12.229
playlist_genrelatin                        2.1066     0.4832   4.360
playlist_genrepop                          4.9005     0.4532  10.813
playlist_genrer&b                          3.5616     0.4578   7.781
playlist_genrerap                          3.4258     0.4587   7.469
playlist_genrerock                         3.1048     0.4647   6.681
playlist_subgenrebig room                  0.1464     0.6715   0.218
playlist_subgenreclassic rock             -0.1167     0.1790  -0.652
playlist_subgenredance pop                -0.6406     0.1037  -6.178
playlist_subgenreelectro house             0.9380     0.5332   1.759
playlist_subgenreelectropop               -1.1165     0.1131  -9.868
playlist_subgenregangster rap             -1.1283     0.1873  -6.025
playlist_subgenrehard rock                -0.7950     0.2064  -3.853
playlist_subgenrehip hop                   0.4123     0.1312   3.142
playlist_subgenrehip pop                   0.1436     0.1470   0.977
playlist_subgenreindie poptimism          -2.2086     0.1422 -15.535
playlist_subgenrelatin hip hop             0.9102     0.2183   4.169
playlist_subgenrelatin pop                 1.9038     0.2023   9.410
playlist_subgenreneo soul                 -1.8874     0.2208  -8.549
playlist_subgenrenew jack swing           -3.2198     0.4588  -7.018
playlist_subgenrepermanent wave            0.7548     0.1584   4.766
playlist_subgenrepop edm                   2.7406     0.4715   5.812
playlist_subgenrepost-teen pop                NA         NA      NA
playlist_subgenreprogressive electro house    NA         NA      NA
```

Modelling: SVM

- **1st Model (no weights and numeric)**

```
spotify_svm = svm({{as.factor(track_popularity)}} ~ dance
ability + energy + key + loudness + mode + speechiness +
acousticness + instrumentalness + liveness + valence + te
mpo + duration_ms, data = spotify_svm_train, kernel = 'li
near')
```

- **Training MR=** 0.0929
- **Testing MR=** 0.0928

```
      pred
true     0     1
   0 20575     0
   1  2109     0
```

```
      pred
true     0     1
   0  5146     0
   1   526     0
```

- **Observations:**
- Model **with weights** had better **MR** and **AUC**

- **2nd Model (weight and numeric)**

```
spotify_svm_asymmetric = svm(as.factor(track_popularity)
~  danceability + energy + loudness + speechiness + mode
+ key + acousticness+ instrumentalness + liveness + valen
ce + tempo +duration_ms, data = spotify_svm_train, kernel
= 'linear', class.weights = c("0" = 1, "1" = 2))
```

- **Training MR=** 0.0930
- **Testing MR=** 0.0927

- **AUC Training** = 0.5867
- **AUC Testing** = 0.5699

```
      pred
true     0     1
   0 20575     0
   1  2109     0
```

```
      pred
true     0     1
   0  5146     0
   1   526     0
```

Modelling: Classification Tree

- **1st Model (no weights)**

- We also have to convert *track popularity* into a **dummy variable**
- Split into **training** and **testing** data
- ```
  fit_tree_dec <- rpart(as.factor(track_popularity) ~ danceability + energy + speechiness + lo
  udness + instrumentalness+ mode+ key + acousticness+ liveness + valence + tempo, data=spotif
  y_dec.train)
  ```
- **MR Training**: 0.0935; **MR Testing**: 0.0920. **AUC Train**: 0.5; **AUC Test**:0.5
- Confusion matrix has **0** TP and **0** FP on training & testing data

true	pred 0	1
0	15422	0
1	1591	0

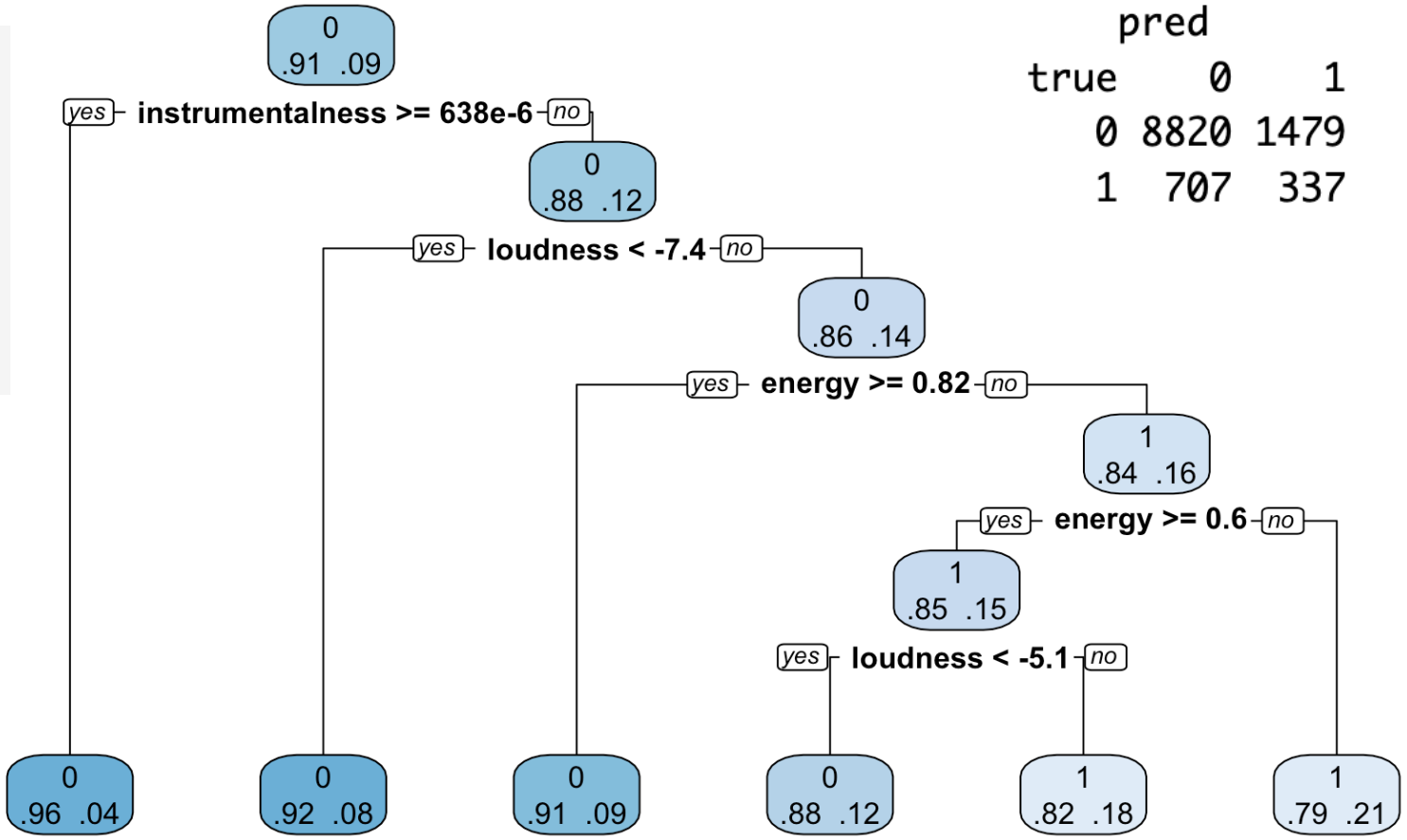
true	pred 0	1
0	10299	0
1	1044	0
  ```

Modelling: Classification Tree

- **2nd Model (with weights of 1 and 6)**

```
cost_matrix <- matrix(c(0, 1,  # cost of 1 for FP
                        6, 0),  # cost of 6 for FN
                      byrow = TRUE, nrow = 2)
fit_tree_asym <- rpart(as.factor(track_popularity) ~ danc
eability + energy + speechiness + loudness + instrumental
ness+ mode+ key + acousticness+ liveness + valence + temp
o, data=spotify_dec.train, parms = list(loss = cost_matri
x))

rpart.plot(fit_tree_asym,extra=4, yesno=2)
```



pred		
true	0	1
0	8820	1479
1	707	337

- **MR Training**:0.1951; **MR Testing**: 0.1927
- **AUC Train**: 0.6614; **AUC Test**:0. 0.6451

- **2nd Model** had higher MR, but **performed better**, since it had TP and TN

Conclusion

- **Comparing All Results**

- **Logistic Regression**

	Training	Testing
AUC	0.6856	0.6610
MR	0.1534	0.1523

- **Observations:**
- **Regression Tree** had the **best** Out-of-Sample **MR**
- **Logistic Regression** had best **AUC**
- **Tree** is the best model, since it can handle continuous data as the dependent variable

- **SVM**

	Training	Testing
AUC	0.587	0.5909
MR1	0.0929	0.0928
MR2	0.0930	0.0927

- **Regression Tree**

	Training	Testing
AUC 1	0.5	0.5
AUC 2	0.6614	0.6451
MR1	0.0955	0.0920
MR2	0.1951	0.1927

Conclusion

- **Comparing All Results**

- **Final Thoughts:**
- **Tree** is the best model out of the last three models
- **LM** performs better than **KNN**
- **Tree** is best one, since LM does not fit four assumptions and overfits data (based on residual plots)

- **Best** *Track Popularity* predictors**:**
- **Plots** and **Models** agree that *instrumentallness*, *speechiness* and *liveness* are the variables that **best predict popularity**
- *Pop* and *dance pop* are the **genre** and **subgenre** that predicts popularity the most
- From linear and logistic models, *key* is not significant and does not influence in a songs **popularity**