30549 - MS Project
Songs Popularity

Leonardo Saveri

BAI | Fall 2021 | Bocconi University

Abstract

The following project aims to find whether there is a correlation between a song danceability and acousticness, and its popularity. First I will describe the dataset, then I will prepare the data and analyze it graphically. Secondly, I will set a hypothesis and perform a test to either reject or retain the null hypothesis, and then I will create a regression model and check if it is statistically significant. Lastly, I will create a simple cross-validation script to predict a song popularity based on its danceability, acousticness, key, mode, and music genre.

1 The Dataset

The dataset I choose to use for this project is the "***Prediction of music genre***[1]". This dataset contains 50005 songs, was collected using Spotify API and categorizes the data in these 18 different features:

- instance_id
- artist_name
- track_name
- popularity
- acousticness
- danceability
- duration_ms
- energy
- instrumentalness
- key
- liveness
- loudness
- mode
- speechiness
- tempo
- obtained_date
- valence
- music_genre

Since I will be dealing only with six of these features, I will describe only those.

- **popularity:** ranging from 0 *(least popular)* to 100 *(most popular)*
- **acousticness:** ranging from 0 *(least acoustic)* to 1 *(most acoustic)*
- **danceability:** ranging from 0 *(least danceable)* to 1 *(most danceable)*
- **key:** the key of each song *(12 in total)*
- **mode:** wether a song is on a *Minor* or a *Major* scale
- **music_genre:** the genre of each song *(10 in total)*

2 Preparing the Data

I first use the function *options(scipen=999)* to avoid dealing with data in exponential form, then I import the csv data:

```
data <- read.csv(file.choose(), sep=",",
    dec=".", header=TRUE)
```

The first step to do is to clean the dataset. I remove rows that have either n.a. or empty cells:

```
data <- na.omit(data)
```

I then chose to extract only the rows that are relevant to my study to light up the data:

```
data <- data[c('popularity',
    'acousticness', 'danceability', 'key',
    'mode', 'music_genre')]
```

I use the *attach(data)* formula to make the code more readable.

3 Understanding the Data

To have a graphical understanding of the data, I first plot a scatterplot showing the relation between the danceability and the popularity of songs:

```
plot(danceability, popularity,
    main="Relation Popularity-Danceability",
     xlab="Danceability ",
         ylab="Popularity", pch=19)
```

I then use the *abline* and *lm* command to plot the correlation as a red line and add it to the graph:

```
abline(lm(popularity ~ danceability), col
    = "red", lwd = 3)
```

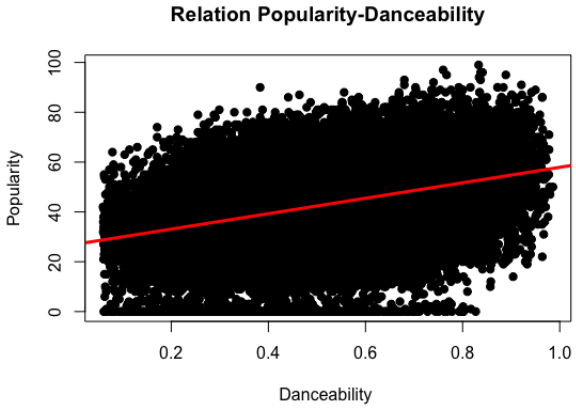

Figure 1: *ScatterPlot Popularity-Danceability*

I proceed to do the same but for acousticness instead of danceability:

```
plot(acousticness, popularity,
    main="Relation Popularity-Acousticness",
     xlab="Acousticness ",
         ylab="Popularity", pch=19)

abline(lm(popularity ~ acousticness), col
    = "red", lwd = 3)
```

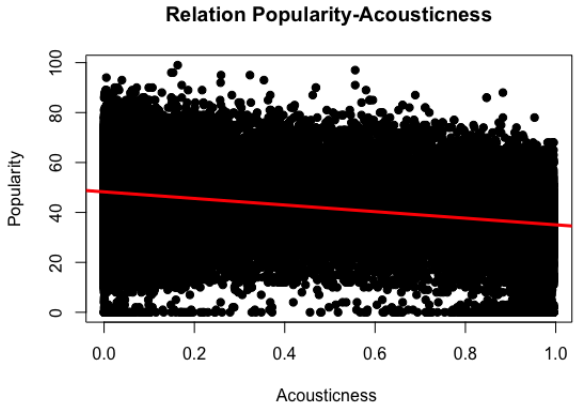

Figure 2: *ScatterPlot Popularity-Acousticness*

What we can deduce from the above plot visualizations is that, *in general*, a **more danceable** song is **more popular**, while a **more acoustic** track is **less popular**.

It can also be interesting to graph and understand the correlation between the acousticness and the danceability of a song *(less acoustic, more danceable and vice-versa)*.

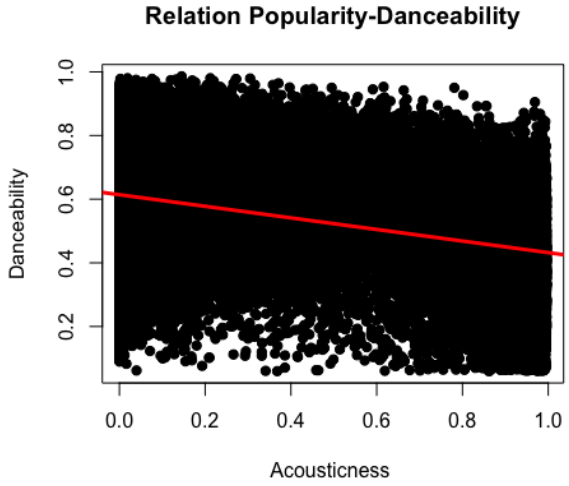

Figure 3: *ScatterPlot Danceability-Acousticness*

Let's now produce a 3d scatterplot, where we can see the relation between both acousticness and danceability, and popularity. We will also take in consideration the different music genres, that are given by the different colors in the graph.

```
install.packages("plotly")
library(plotly)
```

```
fig <- plot_ly(data, x = ~acousticness, z
    = ~popularity, y = ~danceability, color
    = ~music_genre,
    text = ~paste('</br> Artist: ',
        artist_name, '</br> Song: ',
        track_name))
fig <- fig %>% add_markers()
fig <- fig %>%
    layout(scene = list(xaxis = list(title
        = 'acousticness'),
                        yaxis = list(title =
                            'danceability'),
                        zaxis = list(title =
                            'popularity')))
```

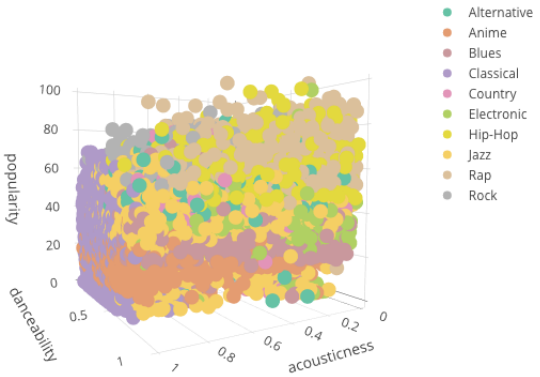

Figure 4: *3d ScatterPlot*

It is difficult to find conclusions by simply looking at this graph, but I think it is interesting to see how, *looking at the big picture,* most of the classical songs are the most acoustic *(as expected)*, and how most of the more popular ones are either Hip-Hop or Rap.

4 Test Statistic

4.1 Hypothesis

Hypothesis: *I expect the popularity of songs in minor and major mode to have same mean.*
Null Hypothesis: *Major and minor songs do not have the same mean popularity.*
Alternate Hypothesis: *Major and minor songs have the same mean popularity.*

4.2 The experiment

I perform a **two-sample t-test**. I can do so because the values of the popularity are independent (a song is not both minor and major in the dataset) We set the α value equal to 0.05, and use a two-sided t-test with different variances *(since the variances of minor_pop and major_pop are different)*:

```
alpha = .05
#since the variances are different
t.test(popularity ~ mode, data = data,
    var.equal = FALSE, conf.level = 1-alpha)
```

The output of the code is the following:

```
        Welch Two Sample t-test

data:  popularity by mode
t = -2.9992, df = 36889, p-value = 0.002709
alternative hypothesis: true difference in means
 between group Major and group Minor is not equal
 to 0
95 percent confidence interval:
 -0.7197427 -0.1508151
sample estimates:
mean in group Major mean in group Minor
           44.06458            44.49986
```

Figure 5: *Result of the Two-Sided T-Test*

Since the **p-value is < 0.05,** we can **reject the null hypothesis.** This means that we can say that there is enough evidence to support that Major and Minor songs have same mean popularity.

5 Linear Regression

I want to create a multiple linear regression model that, given the acousticness and the danceability of a song, could predict its popularity. To do so we use the formula:

```
lm1 <-
    lm(popularity~danceability+acousticness,
    data = data)
```

However, before considering this regression model, we need to check that it is statistically significant.
To do so we use:

```
summary(lm1)
```

And we get the output:

```
Call:
lm(formula = popularity ~ danceability + acousticness, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-52.876  -9.156   0.849   9.876  49.832

Coefficients:
             Estimate Std. Error t value            Pr(>|t|)
(Intercept)   32.7506     0.2489  131.56 <0.0000000000000002 ***
danceability  25.2802     0.3807   66.41 <0.0000000000000002 ***
acousticness  -8.6252     0.1992  -43.30 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.26 on 49997 degrees of freedom
Multiple R-squared:  0.1586,    Adjusted R-squared:  0.1585
F-statistic:  4711 on 2 and 49997 DF,  p-value: < 0.00000000000000022
```

Figure 6: *Summary of the Linear Regression*

Since both the p-values for the danceability and the acousticness are **less than 0.05** *(a lot less)* we can say that the model is statistically significant. **The danceability and acousticness variables are significant in the model**
The estimator of the standard deviation of the data is 14.26.
The $R^2 = 0.1586$ means that the covariates explain 15.86% of the variation in danceability and acousticness. The adjusted $R^2 = 0.1585$, takes into account the the size of the model.
These all means that the linear regression provides a **not so good fit.**
The p-value corresponding to the F-statistic is extremely low *(< 0.00000000000000022)* which provides sufficient evidence in support of this linear model versus no model at all.

6 Cross Validation

Let's now perform an easy cross-validation script to predict the *popularity* of a song given its *danceability, acousticness, key, mode and music genre.*
To do so we first load the required package:

```
install.packages("caret")
library(caret)
```

We then prepare the data with only the columns of the features I choose to use:

```
data <- data[c('popularity',
    'acousticness', 'danceability', 'key',
    'mode', 'music_genre')]
```

We use these data to produce the model using 80% of the data as training data and 20% as testing data, using 123 as seed:

```
set.seed(123)
random_sample <- createDataPartition(data
    $ popularity, p = 0.8, list = FALSE)
training_dataset <- data[random_sample, ]
testing_dataset <- data[-random_sample, ]
model <- lm(popularity ~., data =
    training_dataset)
predictions <- predict(model,
    testing_dataset)
```

We have now created a cross validation model, but to understand if it is good, we need to compute the R^2 and the RMSE:

```
data.frame( R2 = R2(predictions,
    testing_dataset $ popularity),
          RMSE = RMSE(predictions,
              testing_dataset $ popularity))
```

We obtain the following result:

R^2	RMSE
0.6176715	9.668931

The R^2 shows how well the data fits the model. Since it is ~ 0.62 the model is not terrible but it is still *far from optimal.*
The RMSE tells us the average distance between the predicted values from the model and the actual values in the dataset. Since the *popularity* data has a range 0-100, an RMSE of ~ 10 is not too bad, but again, far from optimal.

R Code

The R Code for this project, along with the dataset[1], can be found at this git repository[2].

```
options(scipen=999)
data <- read.csv(file.choose(), sep=",", dec=".", header=TRUE)
#Clean Data
data <- na.omit(data)

#Select only the columns I need
data <- data[c('artist_name', 'track_name', 'popularity', 'acousticness', 'danceability',
    'key', 'mode', 'music_genre')]


View(data)
detach(data)
attach(data)
names(data)

#GRAPHS
#Scatterplot, Popularity-Danceability
plot(danceability, popularity, main="Relation Popularity-Danceability",
     xlab="Danceability ", ylab="Popularity", pch=19)

#Correlation Popularity-Danceability
abline(lm(popularity ~ danceability), col = "red", lwd = 3)

#Scatterplot, Popularity-Acousticness
plot(acousticness, popularity, main="Relation Popularity-Acousticness",
     xlab="Acousticness ", ylab="Popularity", pch=19)

#Correlation Popularity-Acousticness
abline(lm(popularity ~ acousticness), col = "red", lwd = 3)

#Scatterplot, Acousticness-Danceability
plot(acousticness, danceability, main="Relation Popularity-Danceability",
     xlab="Acousticness ", ylab="Danceability", pch=19)

#Correlation Acousticness-Danceability
abline(lm(danceability ~ acousticness), col = "red", lwd = 3)


#3d Scatterplot Showing the relation between Acousticness, Danceability, music_genre and
    Popularity
install.packages("plotly")
library(plotly)

fig <- plot_ly(data, x = ~acousticness, z = ~popularity, y = ~danceability, color =
    ~music_genre,
               text = ~paste('</br> Artist: ', artist_name,
                             '</br> Song: ', track_name))
fig <- fig %>% add_markers()
fig <- fig %>% layout(scene = list(xaxis = list(title = 'acousticness'),
                                   yaxis = list(title = 'danceability'),
                                   zaxis = list(title = 'popularity')))

fig
```

```r
#Two-sample t-test
#H0: mu_major_pop != mu_minor_pop, H1: mu_major_pop = mu_minor_pop
alpha = .05
#since the variances are different
t.test(popularity ~ mode, data = data, var.equal = FALSE, conf.level = 1-alpha)
#we reject the null hypothesis because the p value < 0.05



#Multiple Linear Regression
lm1 <- lm(popularity~danceability+acousticness, data = data)
summary(lm1)

#Cross Validation
install.packages("caret")
library(caret)

#tanking only the columns I need
data <- data[c('popularity', 'acousticness', 'danceability', 'key', 'mode',
               'music_genre')]

#setting the seed
set.seed(123)

random_sample <- createDataPartition(data $ popularity, p = 0.8, list = FALSE)
training_dataset <- data[random_sample, ]
testing_dataset <- data[-random_sample, ]
model <- lm(popularity ~., data = training_dataset)
predictions <- predict(model, testing_dataset)

#understanding the data
data.frame( R2 = R2(predictions, testing_dataset $ popularity),
          RMSE = RMSE(predictions, testing_dataset $ popularity))
```

Contents

References

[1] *Prediction of Music Genre Dataset*. URL: https://www.kaggle.com/vicsuperman/prediction-of-music-genre.

[2] *GitHub Repository*. URL: https://github.com/leonardosaveri/MS_Project_Fall2021.