TheresaAlroukaibe_Phase1

2023-10-19

CSC463 Data Mining ~ Project Phase I

Name: Theresa Al Roukaibe

Student ID: 202001675

I. Dataset Description

Context

The dataset I chose is titled "Spotify Songs". As the name implies it contains a comprehensive collection of information related to music tracks available on the Spotify platform, which is a music streaming platform. This dataset has been chosen as it contains a variety of attributes (qualitative and quantitative). In addition it allows for a quantitative response in the form of "track_popularity" which peaked my intereset on seing the column in the dataset.

Can we really predict how popular a song might be based on its features?

I imagine answering this question would be very useful and interesting for upcoming and new artists wishing to make an entrance. Which is why this data is a valuable resource in my aim to explore the relationships between a song popularity and its various musical features, to *then build a predictive model to estimate track popularity.*

Dataset Features

Below is a description of all the attributes found in the dataset :

- `track_id`: A unique identifier for each song assigned by the Spotify platform.
- `track_name`: The name of the song.
- `track_artist`: The artist(s) or author(s) of the song.
- `track_popularity`: A quantitative measure of the track's popularity between 0-100 where higher is better or more popular.
- `track_album_id`: Unique Identifier for the album that contains the song. This identifier is assigned by the Spotify platform.
- `track_album_name`: The name of the album in which this song is found.
- `track_album_release_date`: The release date of the album in which this song is found.
- `playlist_name`: The name of the playlist the song belongs to.
- `playlist_id`: Unique Identifier for the playlist the song belongs to.
- `playlist_genre`: The genre of the playlist the song belongs to.
- `playlist_subgenre`: The subgenre of the playlist the song belongs to.
- `danceability`: A measure of how suitable a song 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.

- `energy`: A measure from 0.0 to 1.0 representing 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.
- `key`: The key of the song or overall note used (each number maps to a key such as 0=C, 1=C#, etc…)
- `loudness`: The overall or average loudness of a song in decibels (dB). Values typical range between -60 and 0 db.
- `mode`: 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.
- `speechiness`: 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.
- `acousticness`: A confidence measure from 0.0 to 1.0 of whether the song is acoustic (unamplified sounds). 1.0 represents high confidence the track is acoustic.
- `instrumentalness`: Predicts whether a song contains no vocals. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content.
- `liveness`: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.
- `valence`: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a song or "happy vibes". Songs with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
- `tempo`: The overall estimated tempo of a song 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.
- `duration_ms`: Duration of the song in milliseconds

II. Data Mining Tasks

a. Data preprocessing, visualization, and exploration techniques

Preprocessing

1. I started by installing necessary packages since I am working on my home computer and loaded my dataset into a dataframe called "songs" and visualized it as table to see my attributes clearer

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become e
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
setwd('C:/Users/there/OneDrive/Desktop/Phase1')
songs <- read.csv("spotify_songs.csv", header =TRUE)
View(songs)
```

- The data I have seems very clean however some instances of missing values can be found 15 to be exact. Below is the code where I removed these rows so they do not mess with my explorations further on.

```
sum(is.na(songs))
```

```
## [1] 15
```

```
which(is.na(songs))
```

```
##  [1]  40985  42116  42117  52402  52645  73818  74949  74950  85235  85478
## [11] 172317 173448 173449 183734 183977
```

- Upon ordering the songs by title I noticed duplicates of the same song due to the fact that it was added to several playlists. I procceeded to remove these duplicates in order for my results to be more accurate. I used the combination of name and artist of the song as unique identifiers.

```
clean_songs <-songs[!duplicated(songs[c("track_name", "track_artist")]), ]
View(clean_songs)
```

- Other than that the types of the data I need are all assigned correctly having checked using the function below:

```
str(clean_songs)
```

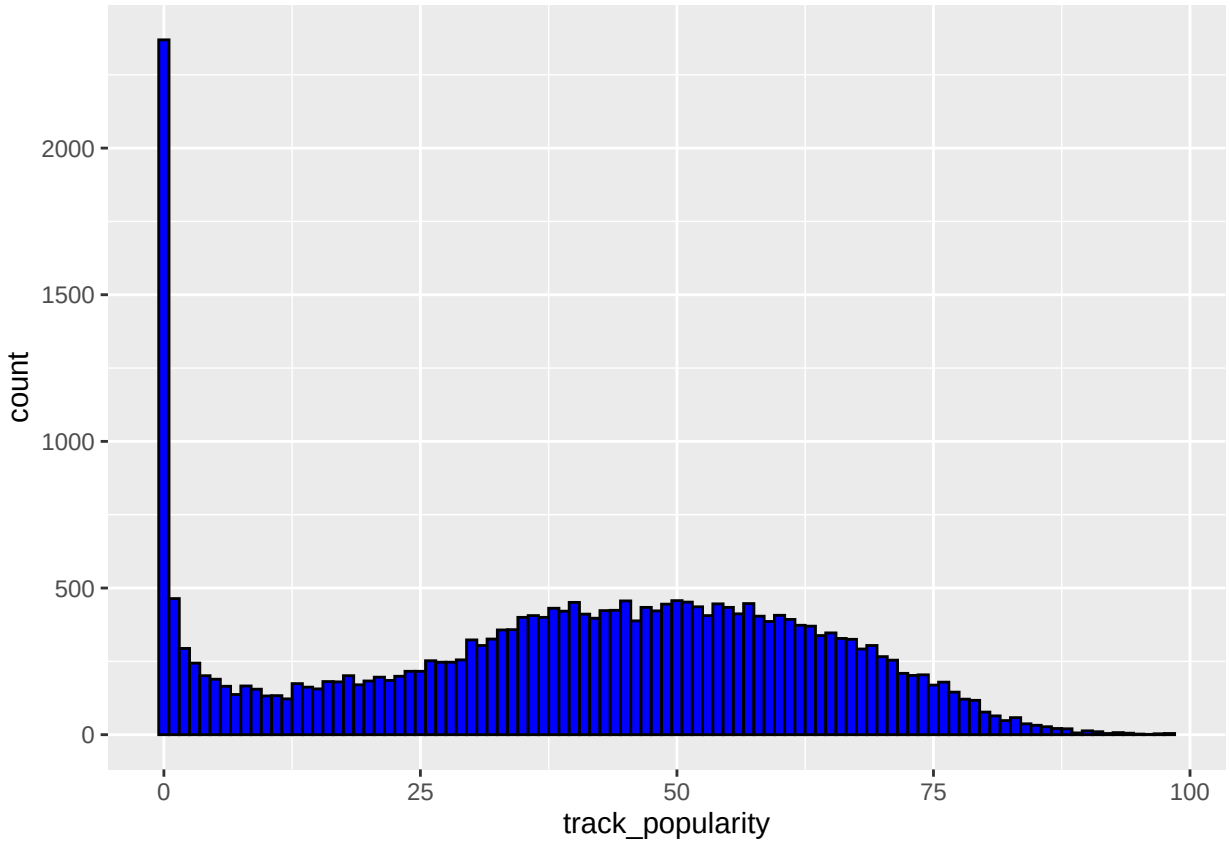
```
## 'data.frame':    26230 obs. of  23 variables:
##  $ track_id                : chr  "6f807x0ima9a1j3VPbc7VN" "0r7CVbZTWZgbTCYdfa2P31" "1z1Hg7Vb0AhHDiEm
##  $ track_name              : chr  "I Don't Care (with Justin Bieber) - Loud Luxury Remix" "Memories -
##  $ track_artist            : chr  "Ed 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" "1HoSmj2eLcsrR0vl
##  $ track_album_name        : chr  "I Don't Care (with Justin Bieber) [Loud Luxury Remix]" "Memories
##  $ 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" "37i9dQZF1DXcZDD
##  $ 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 0.056
##  $ 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
##  $ 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 253
```

Exploration and Visualizations

I chose the following attributes that I suspect popularity is correlated to: **Danceability, Energy, Valence and Tempo** since songs having high values in these attributes might be more upbeat and popular among people.

As a first step of exploring I notice my dataset has a much larger number of less popular songs that high popularity ones. Very rare are tracks that have popularity higher than 75 which might be a problem in building my model.

```
ggplot(clean_songs, aes(x = track_popularity)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black")
```

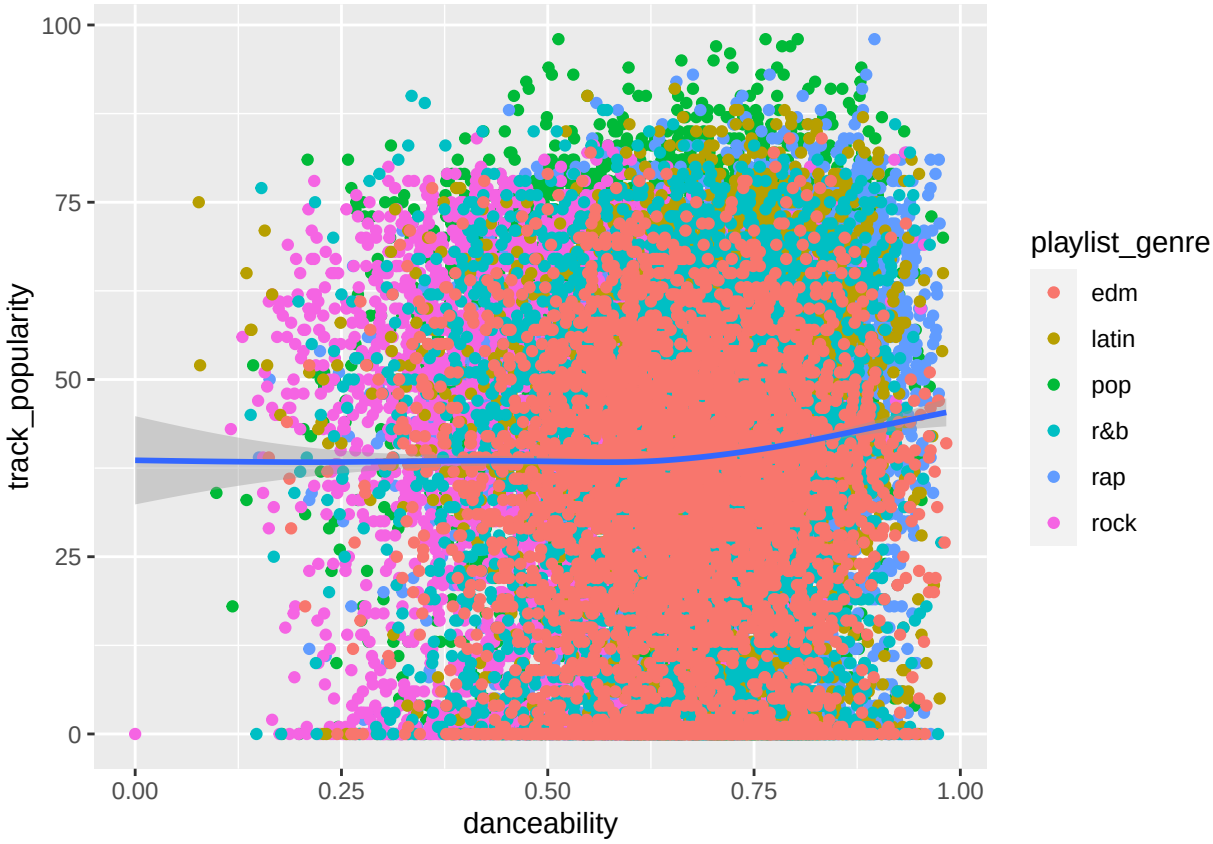


- **Danceability vs. Popularity exploration**

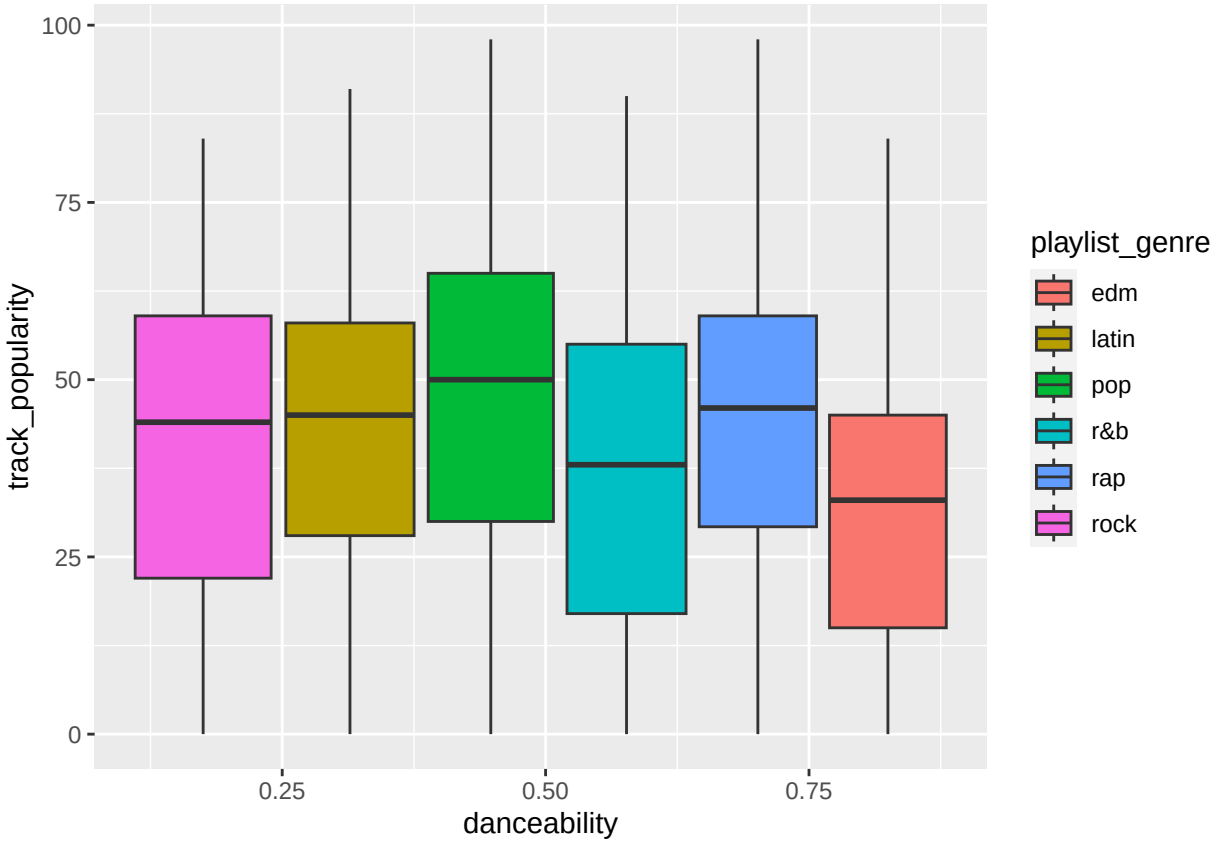
  Based on my explorations we can see from the dot plot below that popularity seems constant until danceability has values above 0.75 where we notice a slight increase in the curve meaning we can see a slight increase of popularity as the track becomes more "danceable". We cannot see a clear trend using only the dots as they are very clustered which is why I added the curve. I also noticed that tracks that belonged to the genre "pop" seem to be concentrated at the top right of my plot, they seem to be the most popular in my dataset in addition to their danceability being high. I wanted to visualize this, so I ran a boxplot for the different genres found below showing pop as having the overall highest track popularity.

```
ggplot(data = clean_songs) +
    geom_point(mapping = aes(x = danceability, y = track_popularity, color=playlist_genre)) +
    geom_smooth(mapping = aes(x = danceability, y = track_popularity))
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
ggplot(data = clean_songs, mapping = aes(x = danceability, y = track_popularity, fill=playlist_genre)) +
      geom_boxplot()
```

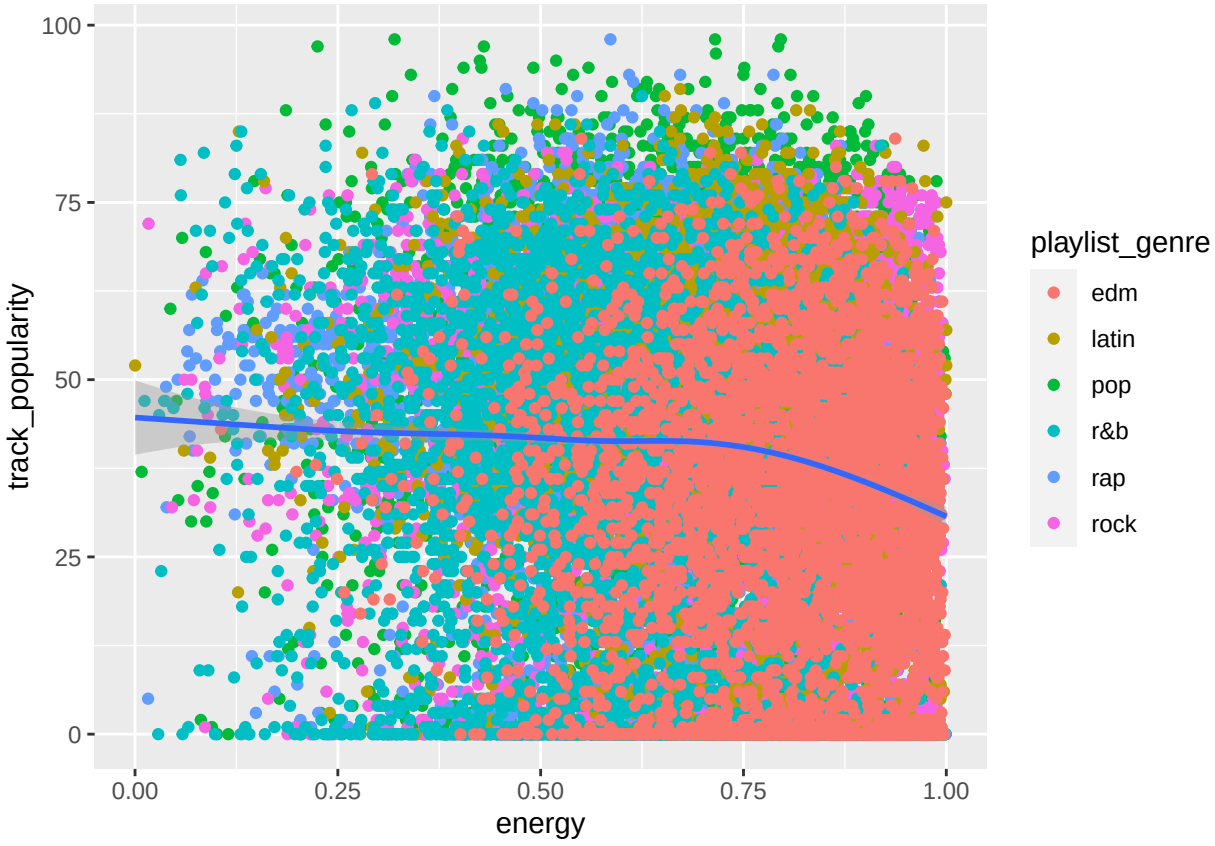


- **Energy vs. Popularity exploration**

  As energy increases, we can see from the curve that popularity is slightly decreasing which might be due to the fact that genres who have higher energy (as we see the red dots representing edm) are concentrated to the left and are not as popular as the other genres of tracks. i.e Not everyone is fan of EDM music.

```
ggplot(data = clean_songs) +
     geom_point(mapping = aes(x = energy, y = track_popularity, color=playlist_genre)) +
     geom_smooth(mapping = aes(x = energy, y = track_popularity))
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

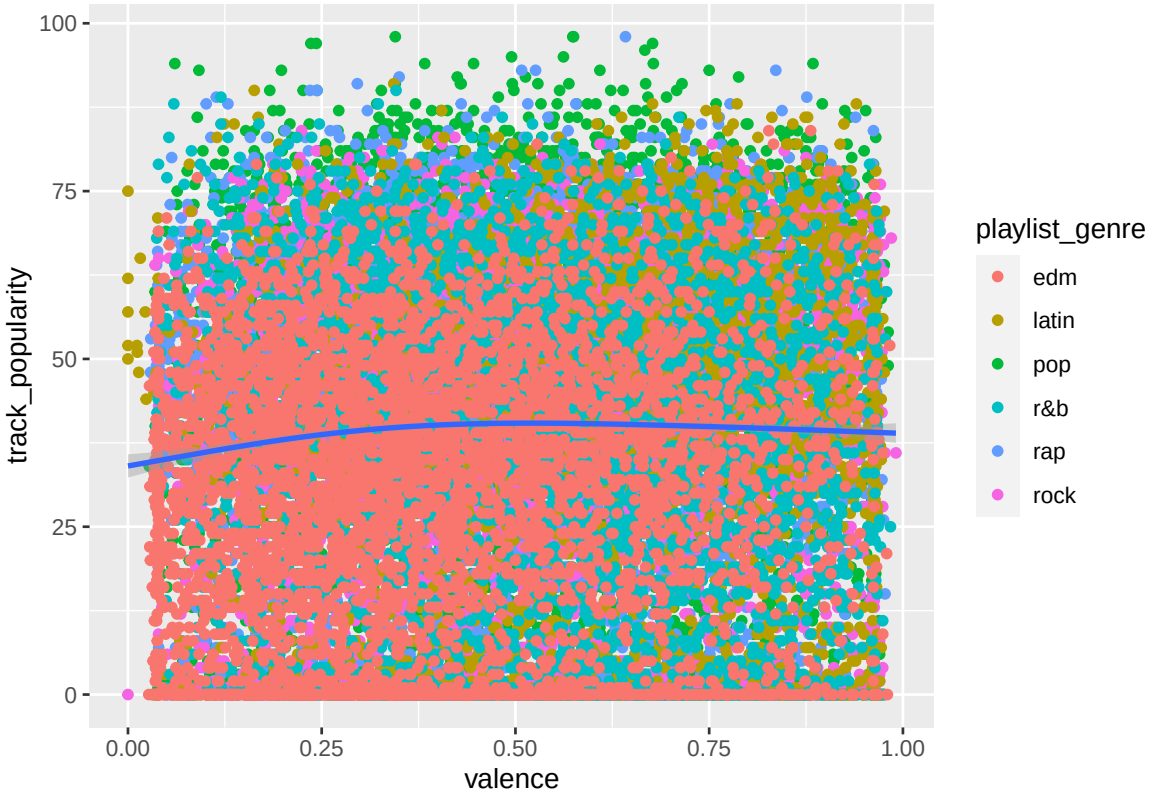


- **Valence vs. Popularity exploration**

  Upon further investigations in valence no clear trend is clear in the graph I obtained, however further correlations analysis should be made to see if we should drop this attribute.

```
ggplot(data = clean_songs) +
  geom_point(mapping = aes(x = valence, y = track_popularity, color=playlist_genre)) +
  geom_smooth(mapping = aes(x = valence, y = track_popularity))
```

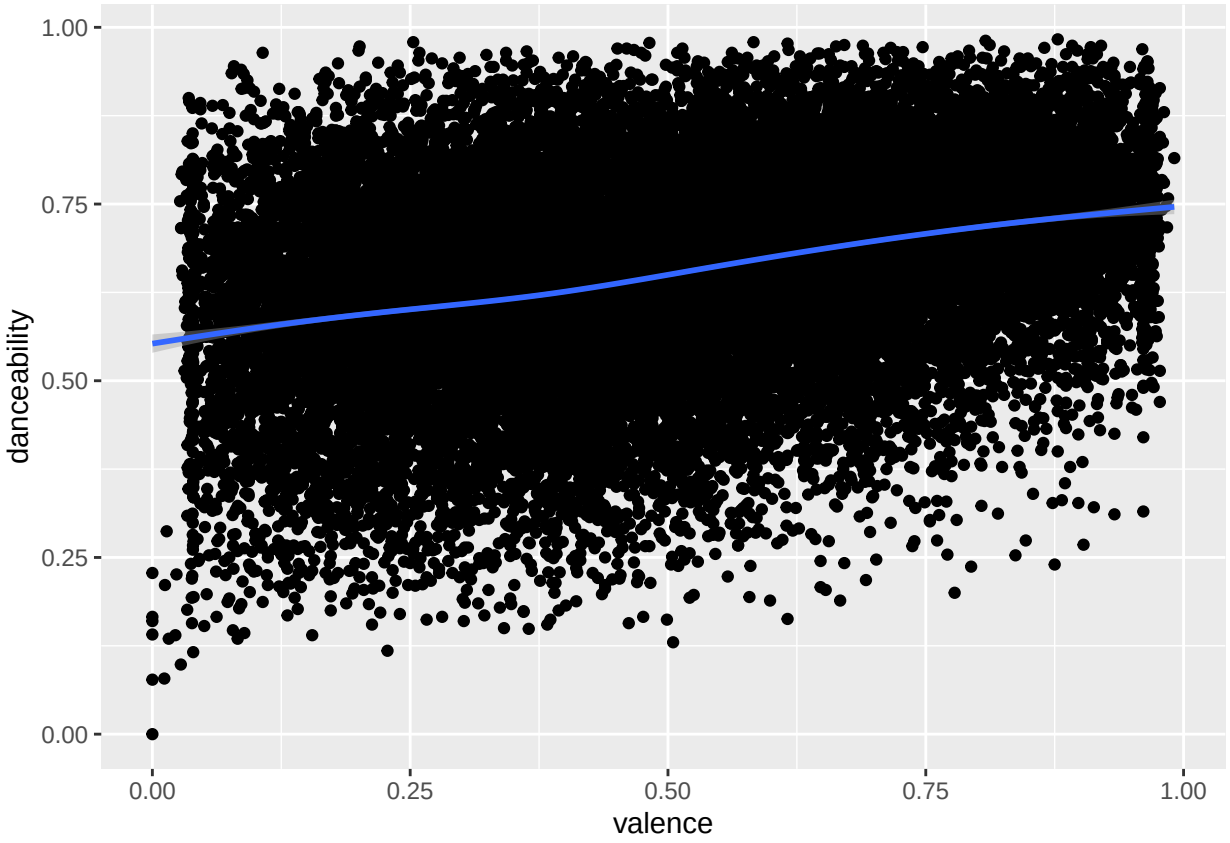
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



However, if we plot valence against danceability we can notice a strong correlation between them, which might be another ouput to explore. [SEE BELOW]

```
ggplot(data = clean_songs) +
  geom_point(mapping = aes(x = valence, y = danceability)) +
  geom_smooth(mapping = aes(x = valence, y = danceability))
```

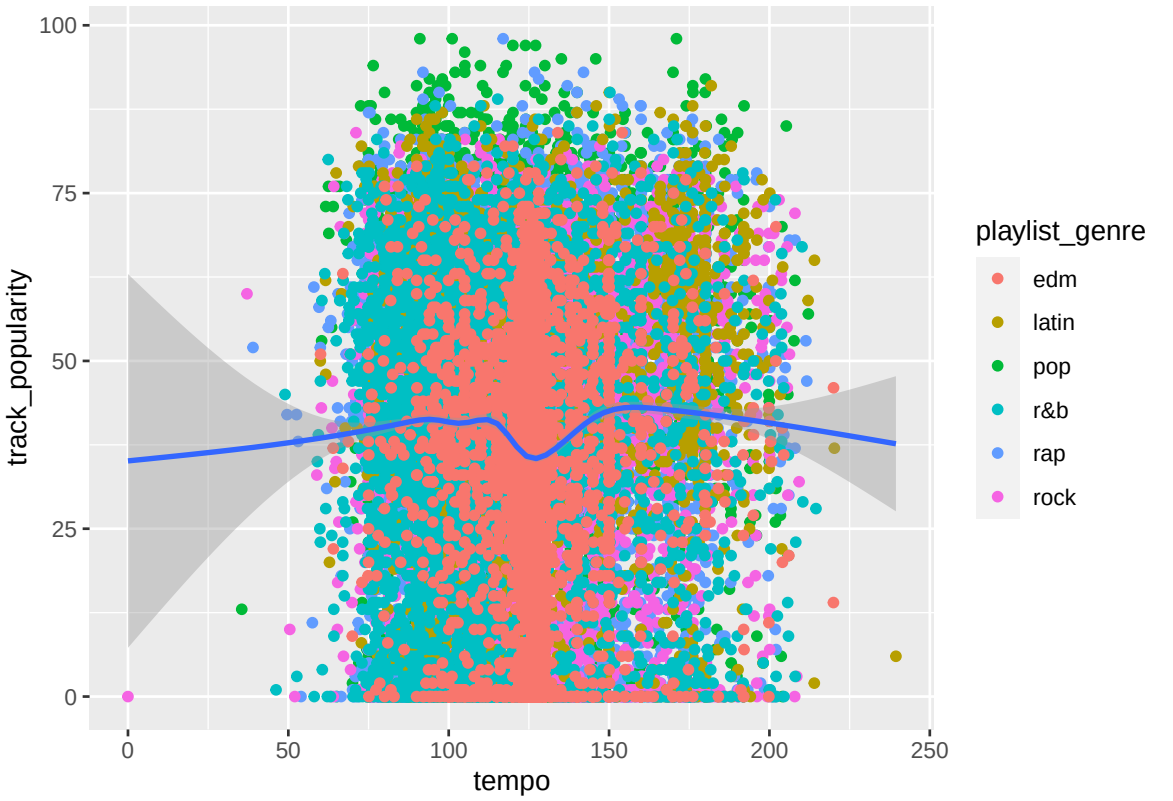
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



- **Tempo vs. Popularity exploration**

```
ggplot(data = clean_songs) +
  geom_point(mapping = aes(x = tempo, y = track_popularity, color=playlist_genre)) +
  geom_smooth(mapping = aes(x = tempo, y = track_popularity))
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



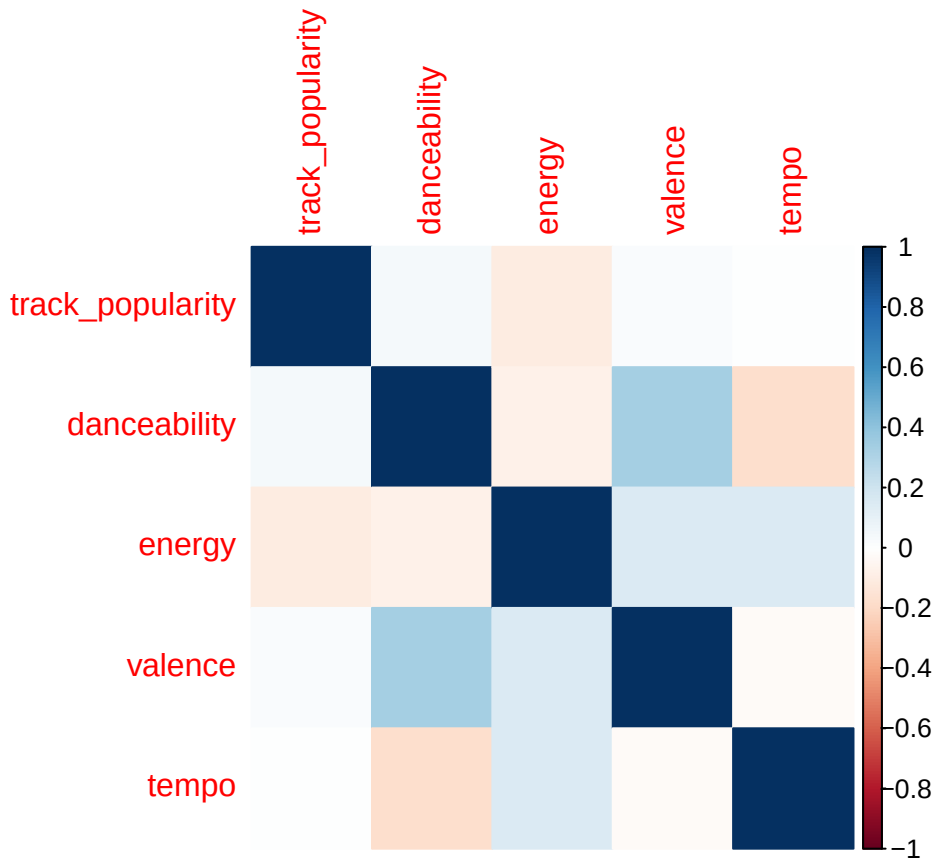
One conclusion I can make from the results above is the lower overall popularity of the edm genre. As we can see the red dots belonging to this genre are concentrated in the middle exactly where we see a drop of track popularity.

b. Subsequent Hypotheses and Correlations

The results I got do not highlight a very strong proof that these attributes affect popularity, maybe due to the fact of the data being heavily concentrated in the lower popularity spectrum.

Let us conduct a correlation matrix to be sure of the strongness of my attributes and choose the ones I want to include in my model. I used the corrplot library to plot a heatmap of the correlations that will show me where the correlation is highest.

```
correlation_matrix <- cor(clean_songs[c("track_popularity", "danceability", "energy", "valence", "tempo
corrplot(correlation_matrix, method = "color")
```



Although faint, valence and danceability seem to have the greatest correlation to track popularity while we can see that higher energy lowers the popularity. So I will drop my study of tempo for now.

Let us look at these correlations in more detail:

```
cor.test(clean_songs$danceability, clean_songs$track_popularity)
```

```
##
##  Pearson's product-moment correlation
##
## data:  clean_songs$danceability and clean_songs$track_popularity
## t = 8.0304, df = 26228, p-value = 1.012e-15
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03744532 0.06158971
## sample estimates:
##        cor
## 0.04952475
```

```
cor.test(clean_songs$energy, clean_songs$track_popularity)
```

```
##
##  Pearson's product-moment correlation
##
## data:  clean_songs$energy and clean_songs$track_popularity
```

```
## t = -17.294, df = 26228, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.11812956 -0.09419865
## sample estimates:
##        cor
## -0.1061795
```

```
cor.test(clean_songs$valence, clean_songs$track_popularity)
```

```
##
##  Pearson's product-moment correlation
##
## data:  clean_songs$valence and clean_songs$track_popularity
## t = 4.8462, df = 26228, p-value = 1.266e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.0178150 0.0419971
## sample estimates:
##        cor
## 0.02991043
```

- Danceability has a t-statistic of approximately 8.0304, which suggests a positive relationship between to popularity. In addition the p-value of 1.012e-15 is low, indicating that the correlation is statistically significant. These results indicate that there is a statistically significant, but relatively weak, correlation between "danceability" and "track_popularity" as we noticed from the exploration phase. This means that songs with higher danceability tend to be slightly more popular, but the strength of this relationship is not very strong. This might mean that many other factors likely influence a song's popularity and should be included in our model.
- Energy has a negative t-value of -17.294 which indicates a negative relationship between "energy" and "track_popularity." Meaning that as "energy" increases, "track_popularity" tends to decrease. The p-value is also relatively small meaning these results are significant.
- Valence also has high t-statistic (altough less than danceability) and a low p-value. The correlation of popularity and valence seems less important than that of popularity with danceability.

  However, let me note that the values in the correlation matrix are very low (run code below for matrix) meaning there is some kind of relationship but it is very weak. This again might be due to the fact of the distribution of the dataset in popularity (very low number of tracks with high popularity)

  ```
  correlation_matrix
  ```

  ```
  ##                  track_popularity danceability     energy     valence
  ## track_popularity      1.000000000   0.04952475 -0.1061795  0.02991043
  ## danceability          0.049524754   1.00000000 -0.0733992  0.33474276
  ## energy               -0.106179481  -0.07339920  1.0000000  0.15005138
  ## valence               0.029910429   0.33474276  0.1500514  1.00000000
  ## tempo                 0.004413484  -0.17461960  0.1520423 -0.02170116
  ##                          tempo
  ## track_popularity  0.004413484
  ## danceability     -0.174619595
  ## energy            0.152042253
  ## valence          -0.021701164
  ## tempo             1.000000000
  ```

c. Linear Regression Techniques

```
mlr_model <- lm (track_popularity ~ danceability + energy + valence, data= clean_songs)
summary(mlr_model)
```

```
##
## Call:
## lm(formula = track_popularity ~ danceability + energy + valence,
##     data = clean_songs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.241 -16.807   2.858  18.012  61.108
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   44.2207     0.8801  50.244  < 2e-16 ***
## danceability   4.6758     1.0494   4.456 8.39e-06 ***
## energy       -13.7901     0.7878 -17.504  < 2e-16 ***
## valence        3.6233     0.6564   5.520 3.42e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.1 on 26226 degrees of freedom
## Multiple R-squared:  0.01417,    Adjusted R-squared:  0.01406
## F-statistic: 125.7 on 3 and 26226 DF,  p-value: < 2.2e-16
```

I chose a multiple linear regression model in order to include all the attribtutes. Because from the graph we suspected a linear relationship between the predictors and the track popularity.

III. Model Performance and Results

- **Coefficient Significances**:

  Danceability: Significant with a low p-value (8.39e-06), we can expect an approximate 4 unit increase in popularity as danceability increases.

  Energy: Highly significant with a very low p-value (< 2e-16), a negative relationship meaning popularity shall decrease 17 units as energy increases.

  Valence: Highly significant with a very low p-value (< 2e-16).

- **R-squared Value:**

  R-squared is 0.01417, so my chosen attributes explain only about 1.42% of the variance in popularity. This is relatively very weak and indicates that the variance in popularity is explained by much more than these attributes. Even tough these attributes are significant they account only for a slight variance in the popularity and other factors seem to be involved in determining a track popularity.

- **Residual Standard Error:** The RSE is a measure of the differences between the predicted values and the actual observed values. In my output, the RSE is approximately 23.1. This means that, on average, predictions of track popularity have an error of around 23.1 units. Which is not very desirable however in contrast to the medical field error in my case is more tolerated than other cases so in this context this might be acceptable.

- **F-statistic:** The F-statistic tests the overall significance of the model in my case it is 125.7, and the associated p-value is $<$ 2.2e-16, suggesting that at least one predictor variable is significant. This indicates that the model as a whole is statistically significant because some factor contributes to explaining the variation of popularity.

IV. Conclusion

In this analysis, I explored the dataset titled "Spotify Songs" with the primary objective to investigate the relationships between the attributes of songs and their popularity in order to build a predictive model for estimating track popularity. Upon further analysis, I noticed danceability and valence had the highest correlations with track popularity, while energy showed a negative relationship. A multiple linear regression model was constructed to predict track popularity using danceability, energy, and valence as predictor variables. The results indicated that these attributes significantly influenced track popularity. However, the model's R-squared value was low, suggesting that they explained only a small portion of the variation in popularity. The F-statistic was statistically significant, suggesting that the model as a whole had explanatory power.

In conclusion, while the attributes explored in this analysis have a statistical impact on track popularity, their influence is relatively weak. To build a more robust predictive model, it's essential to consider other variables and factors that contribute to the complex landscape of music popularity.