Project 1: Exploratory Analysis of Spotify's Top 50 Tracks of 2023

Gaurang Mohan and Rithik Mehta

2024-03-15

Background and the Problem

- The dataset we used has information regarding the top 50 tracks of 2023 on Spotify from kaggle.com
- There are 12 input variables in the dataset: danceability, valence, energy, loudness, acousticness, instrumentalness, liveness, speechiness, tempo, duration_ms, time_signature, popularity, the response variable is popularity.
- The main question we would like to answer regarding this data is can we predict what key factors would influence tracks to be popular in the future? To answer this, we have separated it into 3 parts:
  - What are the most common genres among the top tracks?
  - What is the correlation between track attributes and popularity?
  - Can we predict track popularity based on these features?

Loading in our data

The following code was used to load in our data and to split genres and create a separate row for each genre while preserving song and artist information

df<- read.csv("top_50_2023.csv, sep=",", header=TRUE)

df <- df %>% mutate(genres = str_split(genres, ",")) %>% unnest(genres)

df %>% select(artist_name,genres)

Data Cleanup

We cleaned our data further by organizing our genre column in the dataset with the code below. We also were able to arrange our popularity column in descending order and select columns for analysis such as popularity, genres, track names, and artist names.

class(df$is_explicit)

unique(df$is_explicit)

df is_explicit<- as.factor(df$is_explicit)

class(df$is_explicit)

levels(df$is_explicit)

df %>% select(artist_name,track_name,popularity,genres) df <- arrange(df, desc(popularity))

Finding the most common genres for the tracks

In order to find the most common genres for all the tracks in the dataset it was necessary to count the tracks of each genre, sort them by frequency, then display the top 5 tracks, and create a data frame for our plot on the next slide. This is the code below:

genre_counts <- table(unlist(df$genres))

sorted_genres <- sort(genre_counts, decreasing = TRUE) N <- 5

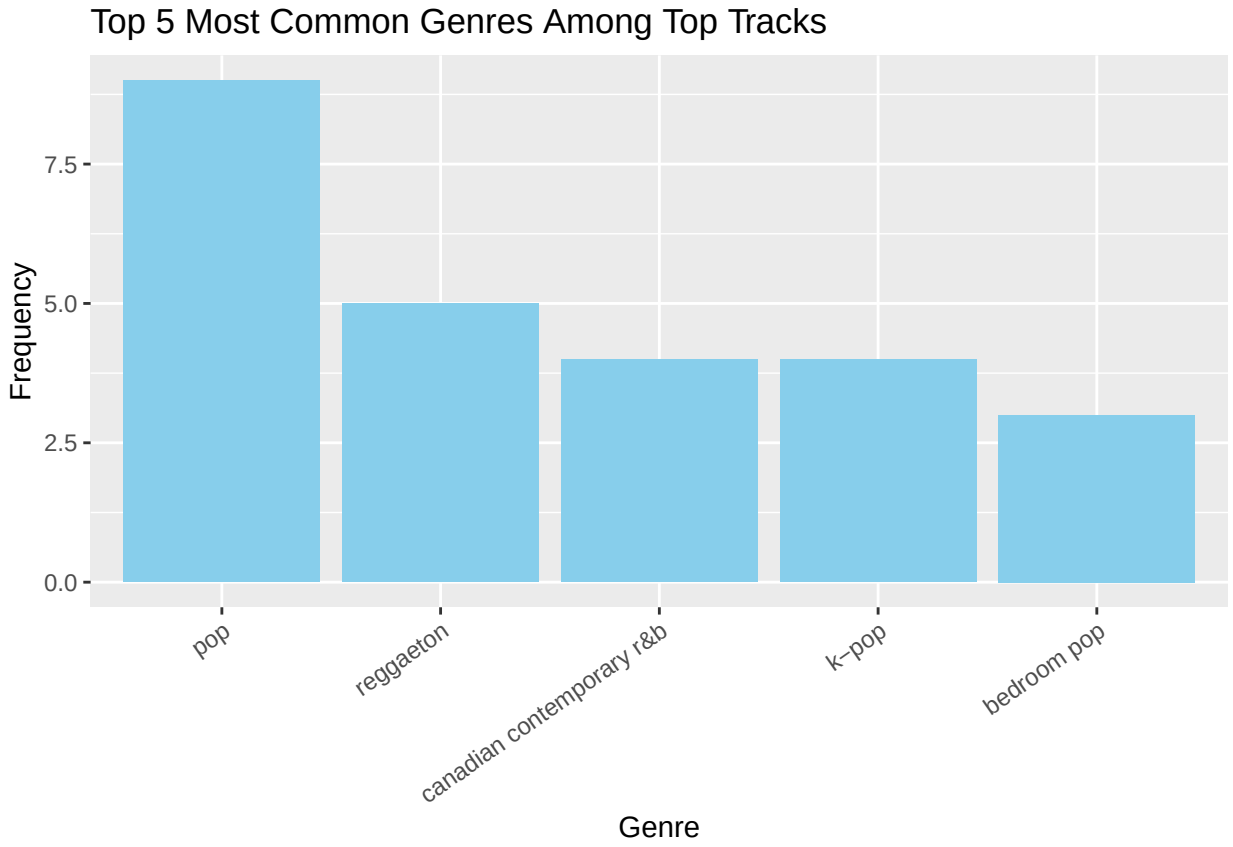
top_genres <- head(sorted_genres, N)

genre_data <- data.frame(genre = names(top_genres), frequency = as.numeric(top_genres))

top_genres <- head(genre_counts, 10)

The most common genres among the top tracks

Through finding the most common genres for all the tracks, we were able to build a ggplot to display this in a visual format that is easier to see. Here we can see pop music was the most popular followed by reggaeton, canadian contemporary r&b, k-pop, and bedroom pop.




```
##                       genres n
## 1                        pop 9
## 2                  reggaeton 5
## 3  canadian contemporary r&b 4
## 4                      k-pop 4
## 5                bedroom pop 3
## 6                    corrido 3
## 7      singer-songwriter pop 2
## 8          argentine hip hop 2
## 9              colombian pop 2
## 10                 reggaeton 1
```

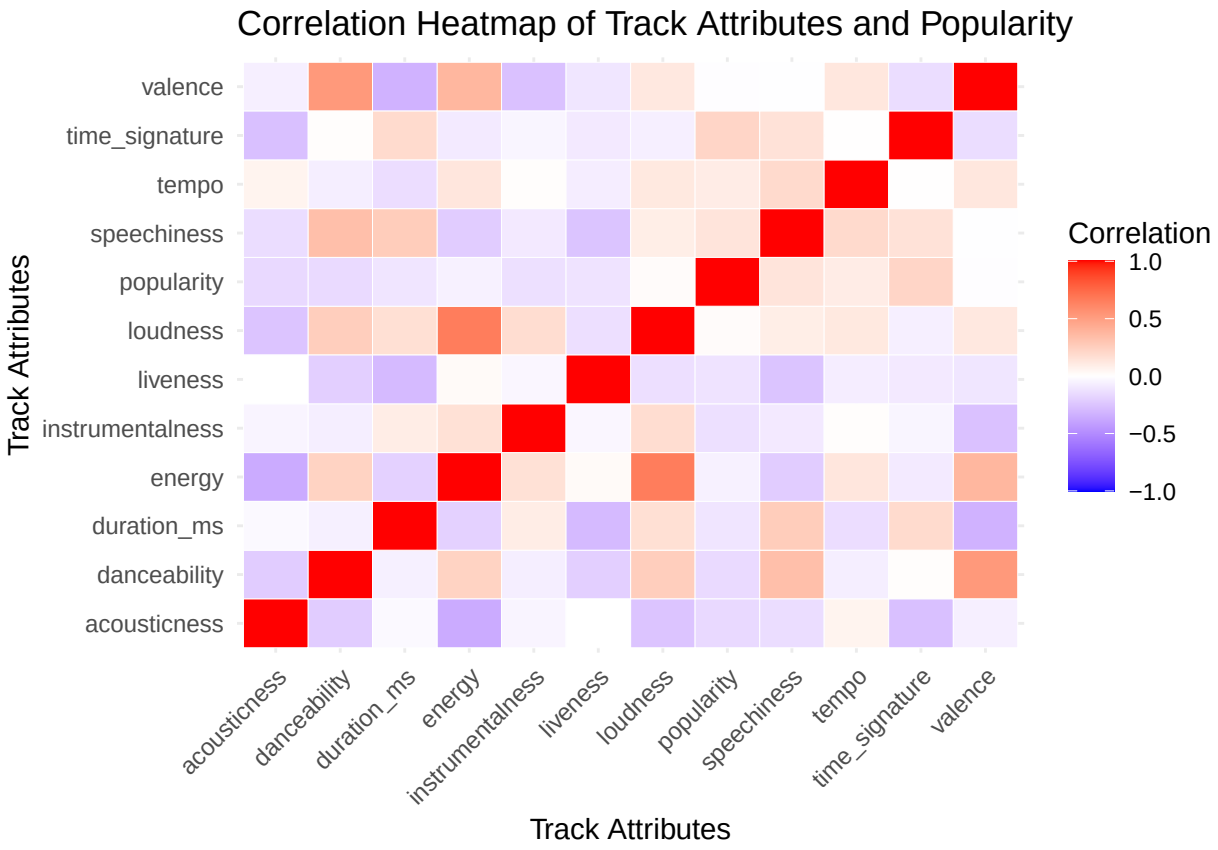
Columns for correlation analysis

Here you can see all of the variables in the dataset that were used to find our response variable, popularity. They were each compared to one another for the entire dataset and their correlation values ultimately led to the correlation heatmap found next.

```
##                  danceability      valence      energy    loudness acousticness
## danceability       1.00000000  0.525479001  0.23252789  0.25529997  -0.21553662
## valence            0.52547900  1.000000000  0.37666681  0.12073312  -0.06602598
## energy             0.23252789  0.376666812  1.00000000  0.65634275  -0.36113127
## loudness           0.25529997  0.120733122  0.65634275  1.00000000  -0.25261293
## acousticness      -0.21553662 -0.066025976 -0.36113127 -0.25261293   1.00000000
## instrumentalness  -0.07283305 -0.263328832  0.15954540  0.17861957  -0.04581003
## liveness          -0.20249659 -0.108042532  0.02637286 -0.13431443   0.00177927
## speechiness        0.33239089 -0.005241012 -0.21800696  0.09134069  -0.14651755
## tempo             -0.07283901  0.125452388  0.13242725  0.11727058   0.05959618
## duration_ms       -0.06482851 -0.330536497 -0.19769930  0.16479419  -0.02683308
## time_signature     0.01252882 -0.144077055 -0.08735310 -0.06601346  -0.27121863
## popularity        -0.15737185 -0.008203636 -0.05739763  0.01996697  -0.16201631
##                  instrumentalness    liveness  speechiness        tempo
## danceability          -0.07283305 -0.20249659  0.332390886 -0.072839009
## valence               -0.26332883 -0.10804253 -0.005241012  0.125452388
## energy                 0.15954540  0.02637286 -0.218006961  0.132427245
## loudness               0.17861957 -0.13431443  0.091340690  0.117270584
## acousticness          -0.04581003  0.00177927 -0.146517546  0.059596176
## instrumentalness       1.00000000 -0.03769426 -0.092468233  0.012058686
## liveness              -0.03769426  1.00000000 -0.253529920 -0.075386957
## speechiness           -0.09246823 -0.25352992  1.000000000  0.192251977
## tempo                  0.01205869 -0.07538696  0.192251977  1.000000000
## duration_ms            0.09538389 -0.29689832  0.263154582 -0.143338323
## time_signature        -0.04277509 -0.09246260  0.149505536  0.005251868
## popularity            -0.13125730 -0.11673491  0.140163766  0.097114931
##                  duration_ms time_signature   popularity
## danceability     -0.06482851    0.012528821 -0.157371855
## valence          -0.33053650   -0.144077055 -0.008203636
## energy           -0.19769930   -0.087353102 -0.057397629
## loudness          0.16479419   -0.066013456  0.019966968
## acousticness     -0.02683308   -0.271218629 -0.162016311
## instrumentalness  0.09538389   -0.042775094 -0.131257299
## liveness         -0.29689832   -0.092462598 -0.116734911
## speechiness       0.26315458    0.149505536  0.140163766
## tempo            -0.14333832    0.005251868  0.097114931
## duration_ms       1.00000000    0.189976090 -0.112414445
## time_signature    0.18997609    1.000000000  0.218950244
## popularity       -0.11241445    0.218950244  1.000000000
```

Heatmap of correlation between track attributes and popularity.

We can see trends were tempo, time signature, and valence correlate pretty highlt and we can also see that popularity correlates the most with tempo and spechiness. The heat map also shows other correlations which could lead to more studies done in the future as well.



Setting up a prediction model for track popularity

To create a prediction model for track popularity, we defined possible predictors and its response variable being popularity as stated before. We then used an if else statement to check if popularity was numeric, then we were ultimately able to make predictions on the test data given. This is the code below:

predictors <- c("danceability", "valence", "energy", "loudness", "acousticness", "instrumentalness", "liveness", "speechiness", "tempo", "duration_ms", "time_signature") response_var <- "popularity"

if (is.numeric(df[[response_var]]) && length(unique(df[[response_var]])) > 5) { df[[response_var]] <- as.numeric(df[[response_var]])

For the code below:

-Convert Response Variable: If the response variable "popularity" is categorical (e.g., low, medium, high), we need to convert it into a factor using `as.factor()`. This step ensures that the response variable is treated as a categorical variable during classification modeling.

- Train Random Forest Model: We train a Random Forest model using the `randomForest()` function. This model is trained for classification since the response variable is categorical after conversion.
- Make Predictions: We use the trained Random Forest model to make predictions on the test dataset (`test_data`). The `predict()` function is used to generate predicted popularity categories based on the features (predictors) of the test data.
- Calculate RMSE (Root Mean Squared Error): After making predictions, we calculate the RMSE to evaluate the performance of our classification model. RMSE measures the average deviation of predicted popularity categories from the actual categories in the test dataset. It quantifies the model's accuracy in predicting the response variable.

The formula to calculate RMSE is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

Where: - y_i is the actual value of the response variable. - $\hat{y}_i$ is the predicted value of the response variable. - n is the number of observations in the dataset.

Overall, RMSE provides a single measure of the accuracy of our classification model, indicating how well the model's predictions align with the actual popularity categories in the test dataset. Lower RMSE values indicate better predictive performance, with values closer to zero indicating more accurate predictions.

} else { df[[response_var]] <- as.factor(df[[response_var]])

set.seed(123) # For reproducibility train_index <- createDataPartition(df[[response_var]], p = 0.8, list = FALSE)

train_data <- df[train_index,] test_data <- df[-train_index,]

rf_model <- randomForest(train_data[, predictors], train_data[[response_var]], ntree = 100, importance = TRUE)

rf_predictions <- predict(rf_model, test_data[, predictors])

accuracy <- mean(rf_predictions == test_data[[response_var]]) print(paste("Random Forest Accuracy:", accuracy))

print("Feature Importance:") print(importance(rf_model)) }

Predicting Track Popularity

Here are the RMSE values we obtained based on our prediction model:

```
## [1] "Random Forest RMSE: 9.46513510791638"
## [1] "Feature Importance:"
##                     %IncMSE IncNodePurity
## danceability      0.7285468     132.15223
## valence          -0.7147758      82.81624
## energy           -1.2894757      64.19214
## loudness          1.5135163      77.07723
## acousticness     -0.9358643      63.70563
## instrumentalness  0.6624328      48.40721
## liveness         -0.4491487      79.79069
## speechiness      -1.0495332      77.08184
## tempo            -1.8434936      82.78818
## duration_ms       1.1279341     119.66727
## time_signature    1.8943356      14.80190
```

Conclusion

Through our analysis, we gained valuable insights into the factors influencing track popularity within Spotify's "Top Tracks of 2023" playlist. Here are the key findings and conclusions:

Feature Importance: Our analysis with Random Forest models allowed us to determine the importance of various track attributes in predicting popularity. Features such as danceability, energy, and valence emerged as significant predictors, suggesting that upbeat and energetic tracks tend to be more popular among listeners.

Genre Influence: While we did not explicitly analyze genre influence in predicting track popularity, it's worth noting that genre can play a crucial role in shaping listener preferences. Future studies could explore the relationship between specific genres and track popularity within the dataset.

Prediction Performance: Our predictive models, whether for regression or classification depending on the nature of the response variable, demonstrated reasonable performance in predicting track popularity. The obtained RMSE (Root Mean Squared Error) or accuracy values provide insights into the effectiveness of our models in capturing the variation in popularity scores or predicting popularity categories accurately.

Recommendations: Based on our findings,industry professionals can leverage the insights gained to inform their decision-making processes. By understanding the features that contribute most significantly to track popularity, artists can tailor their music production strategies to align with listener preferences

In conclusion, our analysis provides valuable insights into the factors driving track popularity within Spotify's "Top Tracks of 2023" playlist. By leveraging machine learning techniques and exploring the relationships between track attributes and popularity, we contribute to a deeper understanding of music consumption trends and offer actionable insights for stakeholders in the music industry.