Optimizing Music Recommendation for User Engagement

Presented by Group 7: James Nwizu, Jyneasha Johnson, Marvin Lomo, and Quentin Daniel

11/29/23| ISM 645 | University of North Carolina at Greensboro

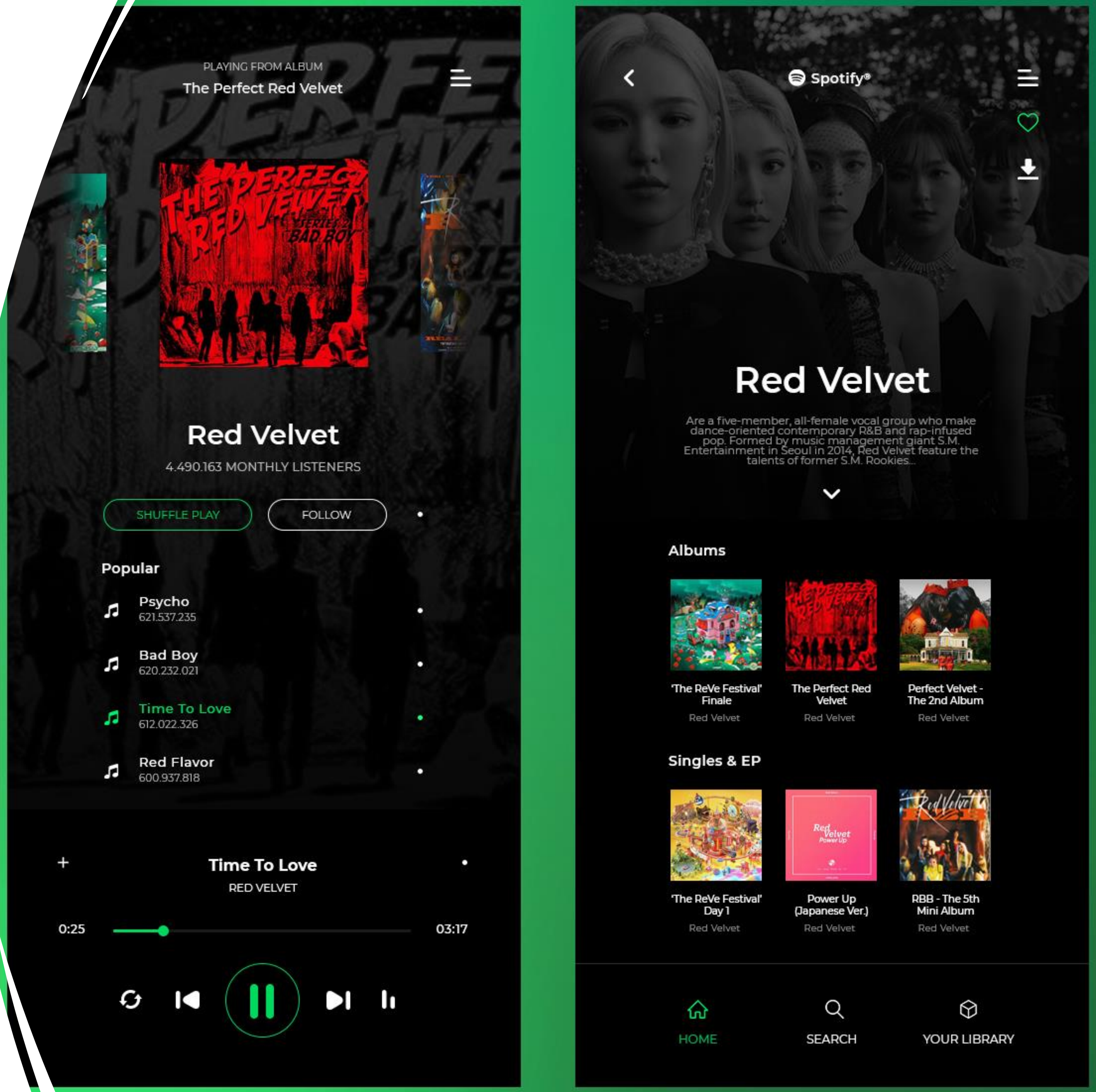
Agenda

- ❑ Introduction and Motivation
- ❑ Research Questions
- ❑ Dataset
- ❑ Model
- ❑ Results and Discussion
- ❑ Conclusion

Introduction

- Spotify, a leading music streaming platform, aims to enhance user engagement and retention by delivering personalized music recommendations.

- The challenge lies in creating a recommendation system that not only suggests music that users are likely to enjoy but also keeps them engaged with the platform for longer periods.

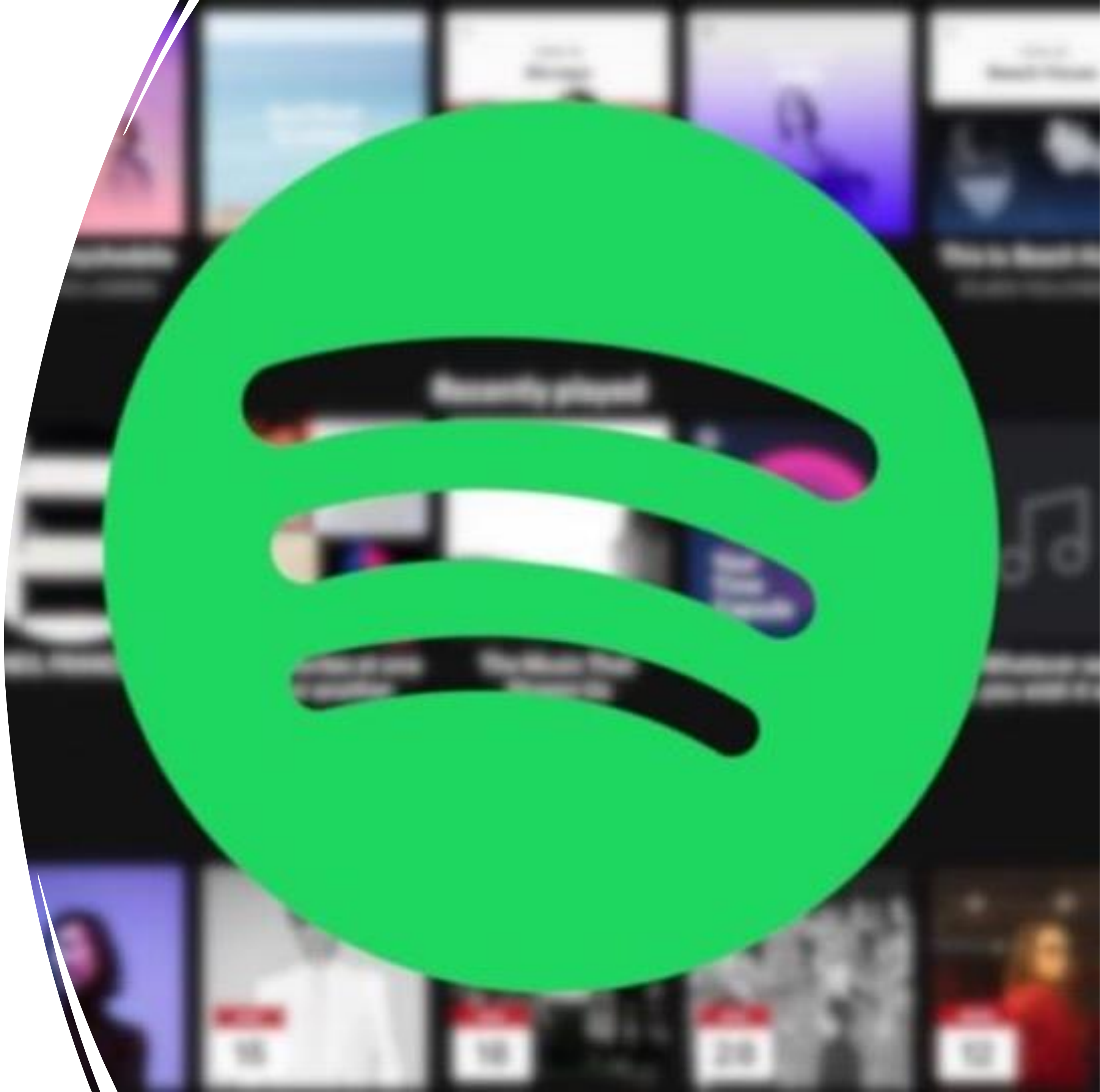


by rareddeer

Motivation

- ❑Exploring a topic related to Spotify allows us to work on something that we all enjoy, which is music. And it helps that Spotify is one of the largest and most popular music streaming platforms globally.

- ❑Using Spotify for the development of our project will be beneficial because it is something that we are genuinely intrigued by and there is a plethora of data to help in our project.



Research Questions

1. How can Spotify implement advanced **machine learning algorithms (unsupervised)** to improve its recommendation engine leading to reduced Churn rate, improved music discovery, and heightened user engagement?

2. What are the **key features** in determining the **similarity of recommended music**?

3. What features are **most/least useful for predicting a track's popularity**?

Research Questions Cont.

To answer the first two questions, we implement a clustering algorithm. We aim to discover subgroups in the data that span the traditional grouping according to genre and popularity, valence, key, and other variables.

- To perform clustering, we will utilize:
  - K-means Clustering: Here, we will seek to partition the observations into a pre-specified number of clusters. To find the optimal number of clusters in our music data set we utilize 'the elbow' technique
- To answer the third question, we utilize two (2) supervised learning models (chosen to enhance explainability):
  - A **logistic regression model** to show the statistically significant features that predict track popularity.
  - A **decision tree model** to validate features that predict track popularity

Dataset

Link to the Dataset: spotify-tracks-dataset

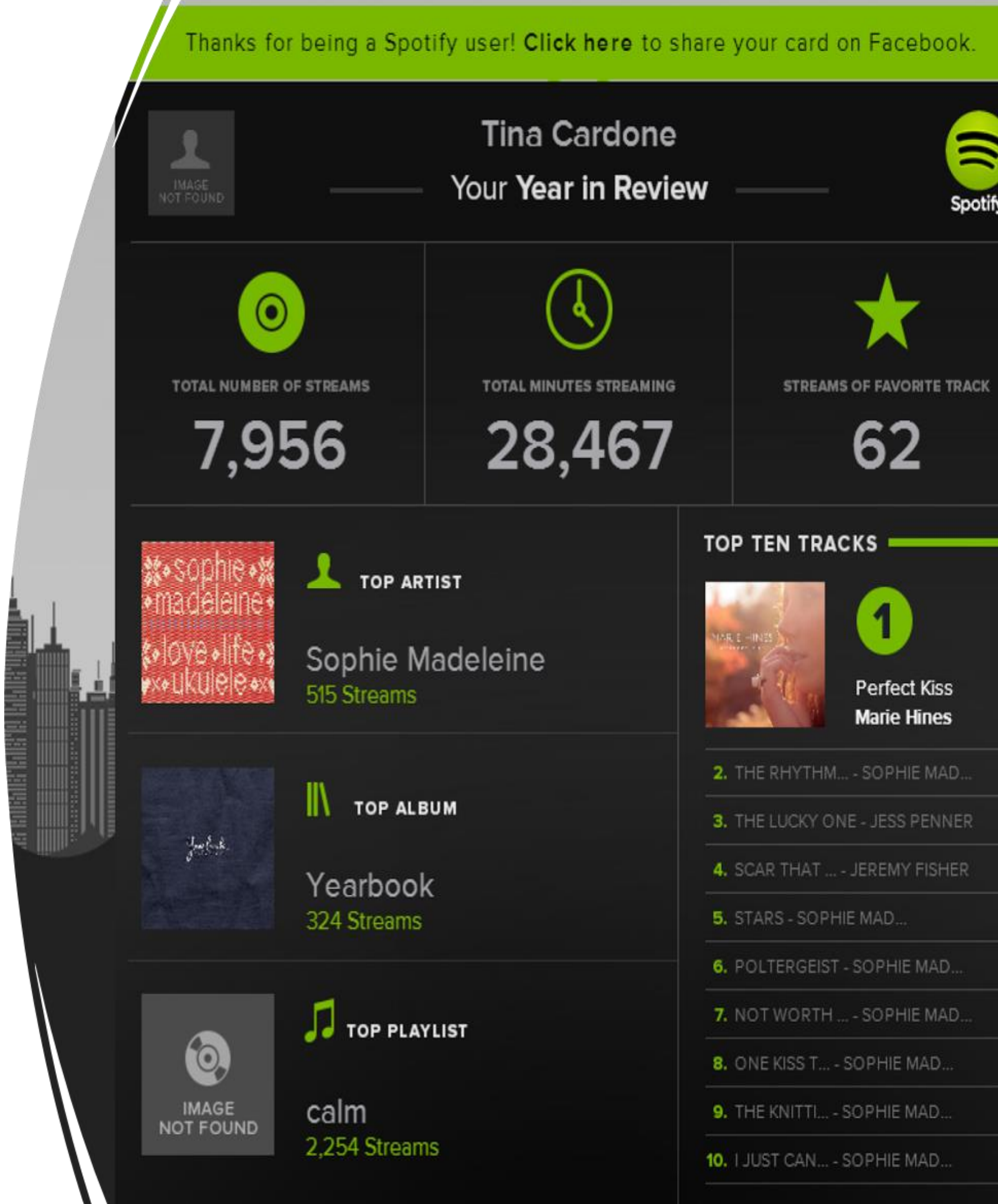
```r
spotify_df <- read_csv('spotify_data.csv', show_col_types = FALSE)

# Set Seed and Select 50_000 records
set.seed(123)

# Load the dataset
spotify_df <- sample_n(spotify_df, 50000)

# Preview the dataset
head(spotify_df)
```

A tibble: 6 × 20

track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liv
<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<lgl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
PDj0uocOvtSJCdOt65qy	Yo Yo Honey Singh	Bhaag Johnny	Aankhon Aankhon	59	244831	FALSE	0.746	0.988	10	-4.565	0	0.0637	0.290000	2.18e-03	
ıDL5VREpThmMOa1pta	Shakti Sivamani	Ailesa	Ailesa	42	250601	FALSE	0.628	0.790	5	-7.159	0	0.0655	0.270000	6.83e-04	
L8He4J2hrTWl9rxbE3DY	Babasónicos	Desde Adentro - Impuesto de Fe (En Vivo)	El Maestro - En Vivo	44	170866	FALSE	0.650	0.915	4	-5.761	0	0.0368	0.446000	0.00e+00	
VgwcEKnjPCcSEsJ5fWY2	Control Freak	Sable Valley Summer Vol. 2	No Chill	39	178285	FALSE	0.593	0.937	1	-6.703	0	0.0566	0.000397	6.28e-01	
VjreSYeix5YZRbwsbPv6T	LA INDIA	The Greatest Salsa Ever	Nunca Voy A Olvidarte	34	299613	FALSE	0.546	0.628	6	-10.289	1	0.0441	0.625000	1.10e-02	

Dataset

- track_id:
- artists:
- album_name:
- track_name:
- popularity:
- duration_ms:
- explicit:
- danceability:
- energy:
- key:
- loudness:
- mode:
- speechiness:
- acousticness:
- instrumentalness:
- liveness:
- valence:
- tempo:
- time_signature:
- track_genre:



Glimpse of the Dataset

```r
# Taking a glimpse of the Dataset
glimpse(spotify_df)
```

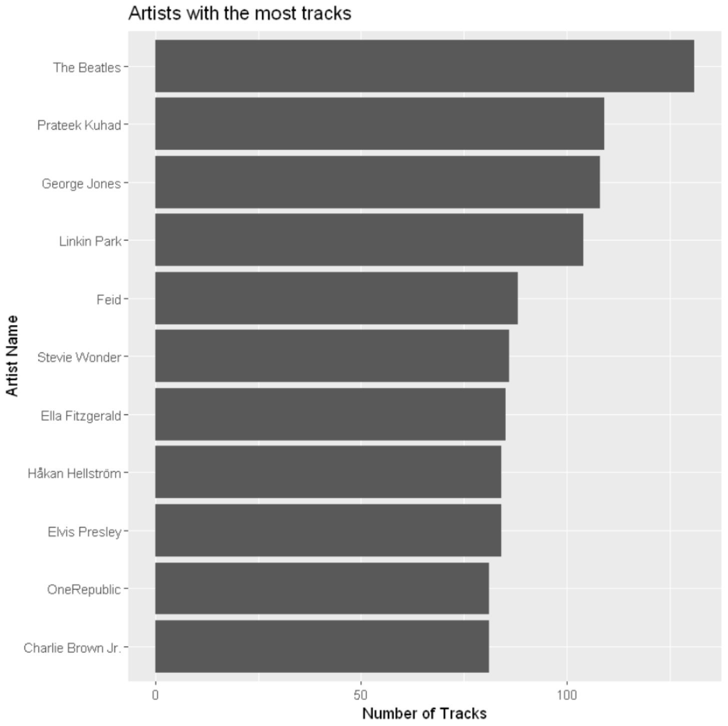
```
Rows: 50,000
Columns: 20
$ track_id         <chr> "78PDj0uocOvtSJCdOt65qy", "1hP0hDL5VREpThmMOa1pta", "…
$ artists          <chr> "Yo Yo Honey Singh", "Shakti Sivamani", "Babasónicos"…
$ album_name       <chr> "Bhaag Johnny", "Ailesa", "Desde Adentro - Impuesto d…
$ track_name       <chr> "Aankhon Aankhon", "Ailesa", "El Maestro - En Vivo", …
$ popularity       <dbl> 59, 42, 44, 39, 34, 4, 0, 55, 35, 57, 21, 29, 46, 5, …
$ duration_ms      <dbl> 244831, 250601, 170866, 178285, 299613, 168828, 19401…
$ explicit         <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS…
$ danceability     <dbl> 0.746, 0.628, 0.650, 0.593, 0.546, 0.676, 0.787, 0.60…
$ energy           <dbl> 0.9880, 0.7900, 0.9150, 0.9370, 0.6280, 0.3380, 0.872…
$ key              <dbl> 10, 5, 4, 1, 6, 9, 9, 2, 0, 7, 5, 2, 11, 11, 0, 0, 0,…
$ loudness         <dbl> -4.565, -7.159, -5.761, -6.703, -10.289, -7.960, -2.1…
$ mode             <dbl> 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,…
$ speechiness      <dbl> 0.0637, 0.0655, 0.0368, 0.0566, 0.0441, 0.0284, 0.143…
$ acousticness     <dbl> 2.90e-01, 2.70e-01, 4.46e-01, 3.97e-04, 6.25e-01, 3.7…
$ instrumentalness <dbl> 2.18e-03, 6.83e-04, 0.00e+00, 6.28e-01, 1.10e-02, 5.5…
$ liveness         <dbl> 0.1310, 0.6060, 0.9100, 0.3020, 0.0801, 0.4680, 0.136…
$ valence          <dbl> 0.6100, 0.4700, 0.8850, 0.0333, 0.7080, 0.3210, 0.696…
$ tempo            <dbl> 130.042, 132.983, 112.028, 139.933, 173.872, 92.029, …
$ time_signature   <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 4, 4, 4, 4,…
$ track_genre      <chr> "hip-hop", "indie", "alt-rock", "dubstep", "salsa", "…
```

Research Strategy

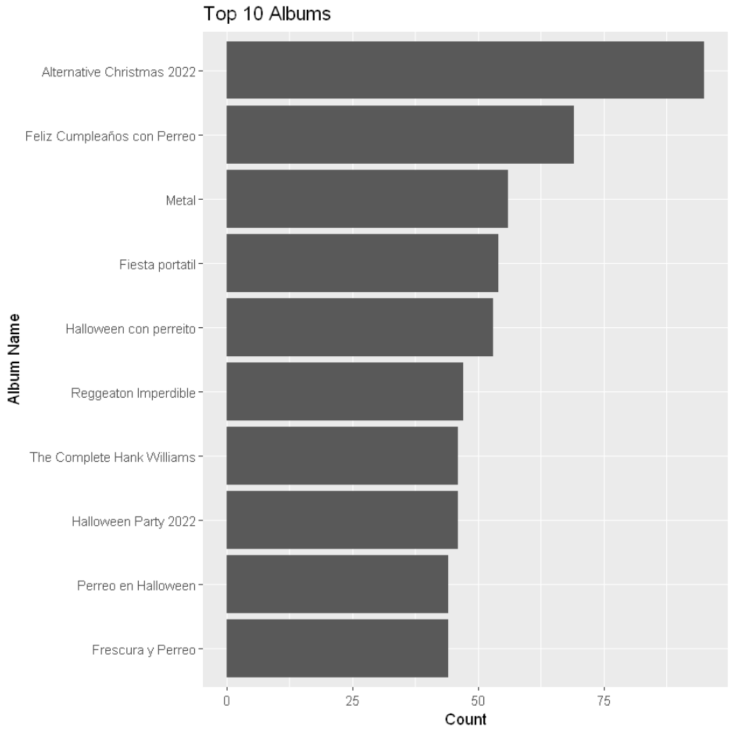
- We followed the CRISP-DM Framework in conducting this research.
  - **Data Understanding:** We explored the data to understand each feature, **removed duplicates, null-values** and prepared the data for modeling with *tidymodel recipes* by using **normalization** and **dummying categorical features**.
  - **Modeling:** We modeled our data using the **k-means clustering** algorithm. To ensure the generalizability of our model, we utilized **k-fold cross-validation to select the right number of clusters.**
  - **Evaluation:** To evaluate our model, we previewed the clusters predicted/assigned and checked for similarity among the tracks/music records. For our supervised learning models, we used the accuracy of a dummy classifier as a baseline for comparing the accuracy of the predictive models.

Data Understanding

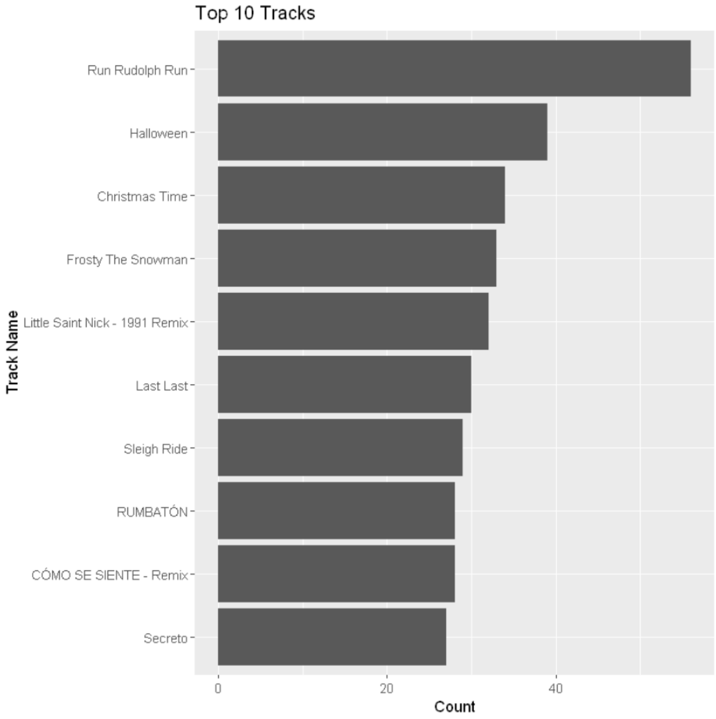
Selecting by count



Selecting by count



Selecting by count



Modeling

```
# Preparing the dataset pipeline
clust_rec  <-   recipe(~ ., data = spotify_df) |>
                    # Update the roles of variables - These are not used in the Clustering Procedure
                    update_role(track_id, new_role = 'ID') |>
                    update_role(track_name, new_role = 'track_name') |>
                    update_role(artists, new_role = 'artist_name') |>
                    update_role(album_name, new_role = 'album_name') |>
                    # Mutate the logical column - explicit
                    step_mutate(explicit = as.numeric(explicit)) |>
                    # Creating dummy variables with the track_genre
                    step_other(track_genre, threshold = 0.009) |>
                    # Here, we dummy all the predictors
                    step_dummy(all_nominal_predictors()) |>
                    # kmeans is sensitive to distances so we normalize the data
                    step_normalize(all_numeric_predictors()) |>
                    # Remove null values from the predictors
                    step_naomit(all_predictors())
```

```
Rows: 49,910
Columns: 46
$ track_id                <fct> 78PDj0uocOvtSJCdOt65qy, 1hP0hDL5VREpThmMOa1pta…
$ artists                 <fct> "Yo Yo Honey Singh", "Shakti Sivamani", "Babas…
$ album_name              <fct> "Bhaag Johnny", "Ailesa", "Desde Adentro - Imp…
$ track_name              <fct> "Aankhon Aankhon", "Ailesa", "El Maestro - En …
$ popularity              <dbl> 1.15220769, 0.38983418, 0.47952518, 0.25529768…
$ duration_ms             <dbl> 0.15064399, 0.20293212, -0.51963180, -0.452400…
$ explicit                <dbl> -0.3051644, -0.3051644, -0.3051644, -0.3051644…
$ danceability            <dbl> 1.0313247, 0.3490043, 0.4762166, 0.1466212, -0…
$ energy                  <dbl> 1.375316404, 0.587560326, 1.084881082, 1.17240…
$ key                     <dbl> 1.32401277, -0.08085819, -0.36183238, -1.20475…
$ loudness                <dbl> 0.73460869, 0.21448138, 0.49479671, 0.30591470…
$ mode                    <dbl> -1.3235551, -1.3235551, -1.3235551, -1.3235551…
$ speechiness             <dbl> -0.200261451, -0.183081236, -0.457010215, -0.2…
$ acousticness            <dbl> -0.07265370, -0.13292259, 0.39744366, -0.94535…
$ instrumentalness        <dbl> -0.4964816, -0.5013259, -0.5035361, 1.5286675,…
$ liveness                <dbl> -0.43395559, 2.07332135, 3.67797859, 0.4686641…
$ valence                 <dbl> 0.52282835, -0.01721733, 1.58363236, -1.701774…
$ tempo                   <dbl> 0.27001075, 0.36847284, -0.33308213, 0.6011527…
$ time_signature          <dbl> 0.2228434, 0.2228434, 0.2228434, 0.2228434, 0.…
$ track_genre_bluegrass   <dbl> -0.09729053, -0.09729053, -0.09729053, -0.0972…
$ track_genre_breakbeat   <dbl> -0.09570417, -0.09570417, -0.09570417, -0.0957…
$ track_genre_british     <dbl> -0.0973954, -0.0973954, -0.0973954, -0.0973954…
$ track_genre_club        <dbl> -0.09538385, -0.09538385, -0.09538385, -0.0953…
```

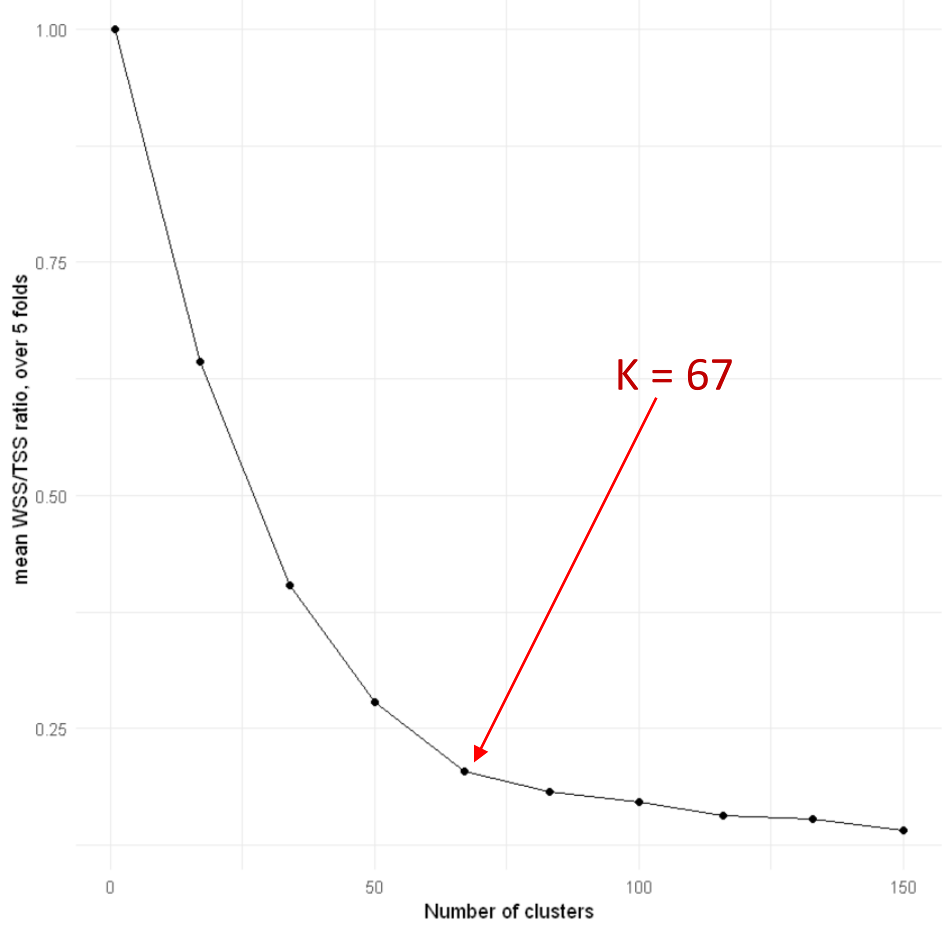
Modeling

```r
# Specifying 5 fold Cross Validation
clust_cv <- vfold_cv(spotify_df, v = 5)
```

```r
# Specifying the model and the tuning parameter
model_kmeans <- k_means(num_clusters = tune()) |>
                    # Set the Engine
                    set_engine('stats')
```

```r
set.seed(123)
# Set the Pipeline
kmeans_wflow <- workflow() |>
                    # Add the model
                    add_model(model_kmeans) |>
                    # Add the pipeline
                    add_recipe(clust_rec)


# Number of clusters
clust_num_grid <- grid_regular(
                        # Setting the number of clusters
                        num_clusters(c(1L, 150L)),
                        # Levels
                        levels = 10
                              )
```

```r
# Lets tune the cluster
res <- tune_cluster(
                        kmeans_wflow,
                        resamples = clust_cv,
                        grid = clust_num_grid,
                        control = control_grid(save_pred = TRUE, extract = identity),
                        metrics = cluster_metric_set(sse_within_total, sse_total, sse_ratio)
                    )
```



Evaluation

pred_cluster	count
<fct>	<int>
Cluster_1	1510
Cluster_2	1464
Cluster_3	1063
Cluster_4	1353
Cluster_5	459
Cluster_6	1332
Cluster_7	1295
Cluster_8	460
Cluster_9	275
Cluster_10	453
Cluster_11	844
Cluster_12	1266
Cluster_13	460
Cluster_14	1084
Cluster_15	1002
Cluster_16	1510
Cluster_17	1313
Cluster_18	471
Cluster_19	468
Cluster_20	1017
Cluster_21	455
Cluster_22	1042
Cluster_23	1130
Cluster_24	476
Cluster_25	1200
Cluster_26	989
Cluster_27	903
Cluster_28	1283
Cluster_29	461
Cluster_30	1586
⋮	⋮
Cluster_38	457
Cluster_39	806
Cluster_40	1323
Cluster_50	144
Cluster_51	453
Cluster_52	666
Cluster_53	166
Cluster_54	1025
Cluster_55	535
Cluster_56	83
Cluster_57	464
Cluster_58	839
Cluster_59	773
Cluster_60	358
Cluster_61	469
Cluster_62	308
Cluster_63	147
Cluster_64	263
Cluster_65	183
Cluster_66	234
Cluster_67	14

pred_cluster	artists	album_name	track_name
<fct>	<fct>	<fct>	<fct>
Cluster_2	Shakti Sivamani	Ailesa	Ailesa
Cluster_2	Babasónicos	Desde Adentro - Impuesto de Fe (En Vivo)	El Maestro - En Vivo
Cluster_2	Afrojack;R3HAB;Au/Ra	Electro sounds	Worlds On Fire
Cluster_2	Barão Vermelho	Balada MTV	Puro êxtase
Cluster_2	Bobby Rydell	The Best Of Bobby Rydell	Sway
Cluster_2	Official HIGE DANdism	One-Man Tour 2021-2022 -Editorial-@Saitama Super Arena (LIVE)	Bedroom Talk - LIVE
Cluster_2	Ritviz	Mimmi	Pukaar
Cluster_2	Sonu Nigam	Devi Bhajan - Sonu Nigam	Nanhe Nanhe Paon Mere (From "Meri Maa")
Cluster_2	Biquini Cavadão	Ao Vivo	Múmias
Cluster_2	Tobee	Après Ski Fussball Party 2022 - Wir feiern auf den Hütten	Weltmeister werden wir in diesem Jahr
Cluster_2	Dorgival Dantas;Flávio José	Minha Música, Nossa História	Passei a Noite no Forró - Ao Vivo
Cluster_2	KSHMR;Bassjackers;Sirah	Memories (feat. Sirah) [Radio Edit]	Memories (feat. Sirah) - Radio Edit
Cluster_2	Turma do Pagode	Turma no Quintal EP 3 (Ao Vivo)	Bebida com Carência (Ao Vivo)
Cluster_2	Sorriso Maroto	Escondido dos seus Pais (Ao Vivo)	Escondido dos seus Pais - Ao Vivo
Cluster_2	Juan D'Arienzo;Mario Bustos;Jorge Valdez	Tango Classics 084: Milonga del recuerdo	Baldosa floja
Cluster_2	Charlie Brown Jr.	Acústico (Ao Vivo)	Só Por Uma Noite - Ao Vivo

Results and Discussion

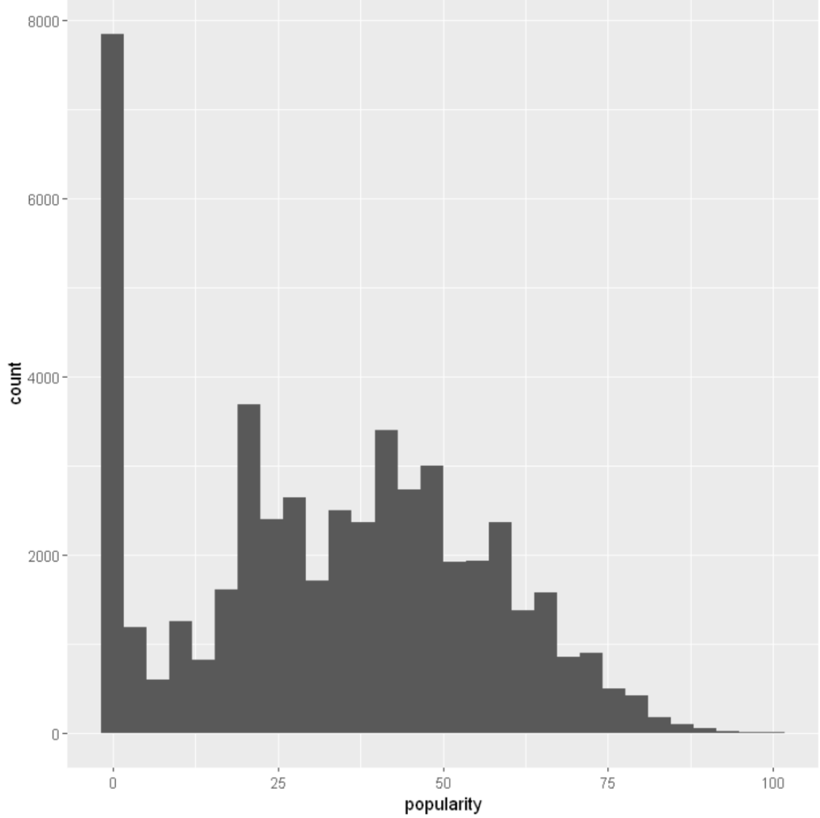
- RQ1: From the model above, a **cluster number of 67** optimizes the WSS/TSS ratio**.**

- RQ2: What are the most key features in determining similarity of recommended music?
  - The features are:
    - Popularity, duration_ms, explicit, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature, track_genre

Modeling Popularity

```r
spotify_df |>
    ggplot(
        aes(x=popularity)
    ) +
    geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binw
```




```r
# Due to the presence of large counts of tracks with low popularity (0) we will treat this as a binary classification problem
pred_spotify_df <- spotify_df |>
                        # Create a binary variable for popularity where tracks with  >= 50 popularity rating are popular and other are not
                        mutate(popularity = as.factor(ifelse(test = popularity >= 50, yes = 1, no = 0)))
```

```r
# Dummy Classifier - Always Predicting the Majority Class
pred_spotify_df |>
    summarise(.by = popularity, count = (n() / nrow(pred_spotify_df)) )
```

A tibble: 2 × 2

popularity	count
<fct>	<dbl>
1	0.2586055
0	0.7413945

> From the above when we always predict the majority class (not popular) we will have an accuracy of `74.14%`

We convert the popularity to a categorical variable due to the high percentage of low popularity (0) tracks in the dataset.

Modeling Popularity

```
# Preparing Data for prediction
pred_rec  <-   recipe( popularity ~ ., data = pred_spotify_df) |>
          # Update the roles of variables - These are not used in the Predictive Pipeline
          update_role(track_id, new_role = 'ID') |>
          update_role(track_name, new_role = 'track_name') |>
          update_role(artists, new_role = 'artist_name') |>
          update_role(album_name, new_role = 'album_name') |>
          # logistic regression optimizer is sensitive to distances so we normalize the data
          step_normalize(all_numeric_predictors()) |>
          # Mutate the logical column - explicit
          step_mutate(explicit = as.numeric(explicit)) |>
          # Create dummy variables for the track_genre
          step_other(track_genre, threshold = 0.009) |>
          # Here, we dummy all the predictors
          step_dummy(all_nominal_predictors()) |>
          # Remove null values from the predictors
          step_naomit(all_predictors())
```

```
Rows: 49,910
Columns: 46
$ track_id              <fct> 78PDj0uocOvtSJCdOt65qy, 1hP0hDL5VREpThmMOa1pta…
$ artists               <fct> "Yo Yo Honey Singh", "Shakti Sivamani", "Babas…
$ album_name            <fct> "Bhaag Johnny", "Ailesa", "Desde Adentro - Imp…
$ track_name            <fct> "Aankhon Aankhon", "Ailesa", "El Maestro - En …
$ duration_ms           <dbl> 0.15064399, 0.20293212, -0.51963180, -0.452400…
$ explicit              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ danceability          <dbl> 1.0313247, 0.3490043, 0.4762166, 0.1466212, -0…
$ energy                <dbl> 1.375316404, 0.587560326, 1.084881082, 1.17240…
$ key                   <dbl> 1.32401277, -0.08085819, -0.36183238, -1.20475…
$ loudness              <dbl> 0.73460869, 0.21448138, 0.49479671, 0.30591470…
$ mode                  <dbl> -1.3235551, -1.3235551, -1.3235551, -1.3235551…
$ speechiness           <dbl> -0.200261451, -0.183081236, -0.457010215, -0.2…
$ acousticness          <dbl> -0.07265370, -0.13292259, 0.39744366, -0.94535…
$ instrumentalness      <dbl> -0.4964816, -0.5013259, -0.5035361, 1.5286675,…
$ liveness              <dbl> -0.43395559, 2.07332135, 3.67797859, 0.4686641…
$ valence               <dbl> 0.52282835, -0.01721733, 1.58363236, -1.701774…
$ tempo                 <dbl> 0.27001075, 0.36847284, -0.33308213, 0.6011527…
$ time_signature        <dbl> 0.2228434, 0.2228434, 0.2228434, 0.2228434, 0.…
$ popularity            <fct> 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0…
$ track_genre_bluegrass <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ track_genre_breakbeat <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ track_genre_british   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
```

Evaluating Popularity

```r
# Building a logistic regression model
log_reg_model <- logistic_reg() |>
            set_engine('glm')


# Linear model workflow
log_reg_wflow <- workflow() |>
                  # add_recipe
                  add_recipe(pred_rec) |>
                 # Add model
                  add_model(log_reg_model)

# Fit the  workflow to the data
log_reg_fit <- fit(log_reg_wflow, data = pred_spotify_df)
```

```r
# preview the model
log_reg_fit |>
    extract_fit_engine() |>
    summary()
```

Call:

```
Call:
stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)               -3.193701   0.247969 -12.879  < 2e-16 ***
duration_ms               -0.178054   0.015792 -11.275  < 2e-16 ***
explicit                   0.375297   0.039157   9.584  < 2e-16 ***
danceability               0.101158   0.013745   7.360 1.84e-13 ***
energy                    -0.094715   0.022889  -4.138 3.50e-05 ***
key                        0.006451   0.010811   0.597 0.550690
loudness                   0.083083   0.020675   4.019 5.85e-05 ***
mode                      -0.045424   0.010915  -4.162 3.16e-05 ***
speechiness               -0.216678   0.015365 -14.102  < 2e-16 ***
acousticness              -0.143021   0.017246  -8.293  < 2e-16 ***
instrumentalness          -0.164494   0.014265 -11.532  < 2e-16 ***
liveness                  -0.188205   0.013004 -14.473  < 2e-16 ***
valence                   -0.209797   0.013975 -15.012  < 2e-16 ***
tempo                     -0.011766   0.011420  -1.030 0.302871
time_signature             0.051115   0.012155   4.205 2.61e-05 ***
track_genre_bluegrass     -0.211577   0.368159  -0.575 0.565502
track_genre_breakbeat      0.093327   0.341864   0.273 0.784857
track_genre_british        3.318223   0.265492  12.498  < 2e-16 ***
track_genre_club           0.827568   0.294444   2.811 0.004945 **
track_genre_dance          2.003899   0.268429   7.465 8.31e-14 ***
track_genre_dub            1.856843   0.269861   6.881 5.95e-12 ***
track_genre_folk           2.387911   0.268644   8.889  < 2e-16 ***
track_genre_french         2.497011   0.267180   9.346  < 2e-16 ***
track_genre_grindcore    -13.340850 111.040779  -0.120 0.904369
track_genre_groove         2.540818   0.266576   9.531  < 2e-16 ***
track_genre_guitar         0.248411   0.338082   0.735 0.462483
track_genre_happy          0.259953   0.326055   0.797 0.425296
track_genre_hardcore       1.988236   0.271609   7.320 2.48e-13 ***
track_genre_iranian      -13.095675 108.905306  -0.120 0.904286
track_genre_j.idol        -0.427222   0.383872  -1.113 0.265739
track_genre_k.pop          3.998687   0.269274  14.850  < 2e-16 ***
track_genre_mandopop       2.828614   0.265966  10.635  < 2e-16 ***
track_genre_metal          3.317307   0.266351  12.455  < 2e-16 ***
```

```r
# Confusion Matrix
conf_mat(final_df, truth = popularity, estimate = log_reg_pred)
```

```
          Truth
Prediction     0     1
         0 36232 11688
         1   771  1219
```

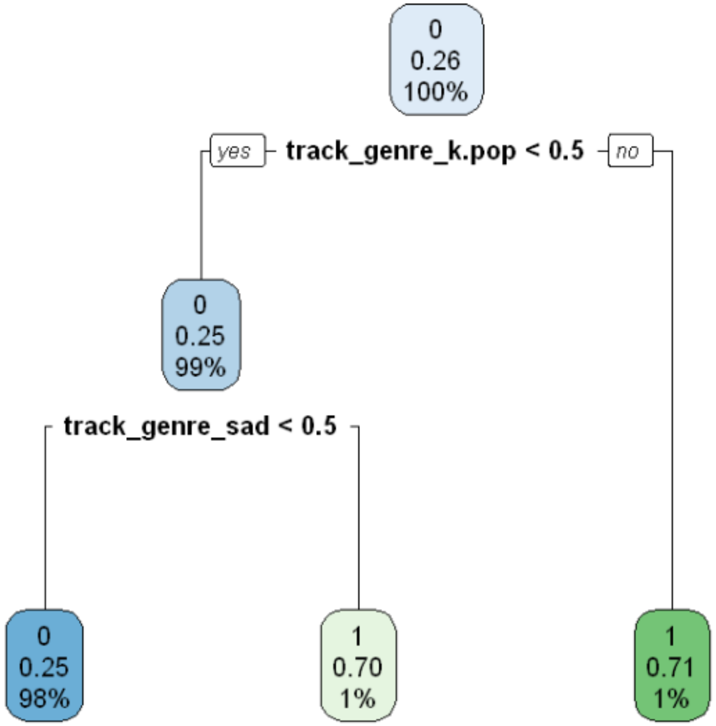
```r
# Accuracy of the Prediction
accuracy(final_df, truth = popularity, estimate = log_reg_pred)
```

A tibble: 1 × 3

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
accuracy	binary	0.7503707

Evaluating Popularity

```r
dec_tree_class <- decision_tree(mode = 'classification') |>
                  set_engine('rpart')


# Linear model workflow
dec_tree_class_wflow <- workflow() |>
                        # add_recipe
                        add_recipe(pred_rec) |>
                        # Add model
                        add_model(dec_tree_class)


# Fit the  workflow to the data
dec_tree_class_fit <- fit(dec_tree_class_wflow, data = pred_spotify_df)
```



Results and Discussion II

- RQ3: What features are most/least useful for predicting a track's popularity?
  - Our analysis show that the ff features are most useful in predicting a tracks popularity:
    - duration_ms, explicit, danceability, energy, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, time_signature, track_genre

Conclusion

From our analysis, we can conclude the ff:

1. K-means clustering can be utilized to produce sub-groups/clusters of similar music that can be recommended to listeners based on the listener's playlist history.

2. To determine similarity between songs, going beyond traditional track genres and including other features like *popularity, duration_ms, explicit, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature* will improve the similarity in recommended music.

3. From our analysis, we reveal that songs that are explicit, danceable, and loud increase the probability of popularity.

Thank you

Presented by Group 7:

James Nwizu,

Jyneasha Johnson,

Marvin Lomo,

Quentin Daniel

11/29/23| ISM 645 | University of North Carolina at Greensboro