Final Project

Asopjio Lea

TITLE: AN ANALYSIS OF SPOTIFY TRACKS TO UNDERSTAND MUSIC TRENDS AND PREFERENCES.

Introduction

In an era where music streaming services like Spotify define the contours of the music industry, understanding what makes a song resonate with listeners is more crucial than ever. Spotify, as a global leader in music streaming, offers a unique vantage point to analyze music trends, artist popularity, and the intrinsic qualities that make a track successful. With millions of tracks and equally diverse user preferences, the platform serves as a dynamic dataset ripe for exploration.

Music, a universal language, has evolved significantly with the advancement of technology. From vinyl records to cassettes, CDs to MP3s, and now streaming services, the way we consume music has changed, but our love for it remains constant. Today, Spotify leads this transformation, curating experiences for listeners and providing artists with a platform to showcase their creativity. This project aims to analyze a dataset of Spotify tracks to uncover trends in music preferences and track features over recent years.

PART I: Data set presentation and preparation.

1. Upload and View Dataset

```
spotify_data <- read_csv("spotify_playlist_2000to2023.csv")
```

```
## Rows: 2400 Columns: 23
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): playlist_url, track_id, track_name, album, artist_id, artist_name,...
## dbl (16): year, track_popularity, artist_popularity, danceability, energy, k...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
view(spotify_data)
```

2. Data Wrangling

i. Selecting useful features

```
wrangled_spotify_data <- spotify_data %>%
  select(year, track_name, artist_name, artist_genres, artist_popularity, track_popularity, danceability, instrumenta

view(wrangled_spotify_data)
```

ii) Features explanation.

1. year: Top hit year of the playlist
2. track_name: The name of the track
3. artist_name: The artists who performed the track.
4. artist_genres: A list of the genres the artist is associated with.
5. artist_popularity: The popularity of the artist. The value will be between 0 and 100, with 100 being the most popular.
6. track_popularity: The popularity of the track. The value will be between 0 and 100, with 100 being the most popular.
7. danceability: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo.
8. intrumentalness: The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content.
9. valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more.
10. tempo: The overall estimated tempo of a track in beats per minute (BPM).
11. duration_ms: The duration of the track in milliseconds.
12. energy: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.

iii. Data transformation.

```
# Converting duration_ms from milliseconds to minutes

wrangled_spotify_data <- wrangled_spotify_data %>%
  mutate(duration_min = round(duration_ms / 60000, 2)) %>%
  select(-duration_ms)

head(wrangled_spotify_data)
```

```
## # A tibble: 6 × 12
##    year track_name  artist_name artist_genres artist_popularity track_popularity
##   <dbl> <chr>       <chr>       <chr>                     <dbl>            <dbl>
## 1  2000 Oops!...I … Britney Sp… ['dance pop'…                81               81
## 2  2000 All The Sm… blink-182   ['alternativ…                79               83
## 3  2000 Breathe     Faith Hill  ['contempora…                62               66
## 4  2000 It's My Li… Bon Jovi    ['glam metal…                79               81
## 5  2000 Bye Bye Bye *NSYNC      ['boy band',…                70               75
## 6  2000 Thong Song  Sisqo       ['contempora…                58               71
## # ℹ 6 more variables: danceability <dbl>, instrumentalness <dbl>,
## #   valence <dbl>, tempo <dbl>, energy <dbl>, duration_min <dbl>
```

> The column duration_ms in milliseconds was replaced by the column duration_min in minutes.

```
# Categorizing tempo into 'low', 'moderate', and 'fast'
wrangled_spotify_data <- wrangled_spotify_data %>%
  mutate(tempo = case_when(
    tempo < 100 ~ "low",
    tempo >= 100 & tempo < 200 ~ "moderate",
    tempo >= 200 ~ "fast",
    TRUE ~ NA_character_
  ))

head(wrangled_spotify_data)
```

```
## # A tibble: 6 × 12
##    year track_name  artist_name artist_genres artist_popularity track_popularity
##   <dbl> <chr>       <chr>       <chr>                     <dbl>            <dbl>
## 1  2000 Oops!...I … Britney Sp… ['dance pop'…                81               81
## 2  2000 All The Sm… blink-182   ['alternativ…                79               83
## 3  2000 Breathe     Faith Hill  ['contempora…                62               66
## 4  2000 It's My Li… Bon Jovi    ['glam metal…                79               81
## 5  2000 Bye Bye Bye *NSYNC      ['boy band',…                70               75
## 6  2000 Thong Song  Sisqo       ['contempora…                58               71
## # ℹ 6 more variables: danceability <dbl>, instrumentalness <dbl>,
## #   valence <dbl>, tempo <chr>, energy <dbl>, duration_min <dbl>
```

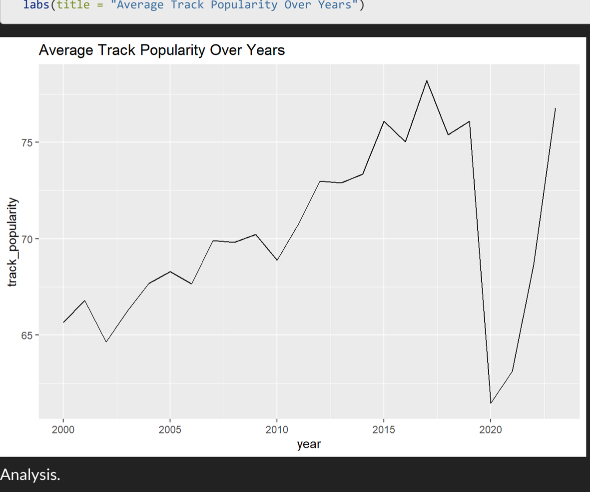
> The numerric values for tempo were categorized into low, moderate and fast.

PART II: Research Questions and Data Visualization.

Research question 1: What years had the highest and lowest average track popularity?

1. Plot time series of average track popularity by year.

```
ggplot(wrangled_spotify_data, aes(x = year, y = track_popularity)) +
  geom_line(stat = "summary", fun = mean) +
  labs(title = "Average Track Popularity Over Years")
```



Analysis.

The above time serie plot shows that the highest average track popularity was in 2017 and the lowest average track popularity was in 2020.

A decline in track popularity in 2020 on platforms like Spotify could be influenced by several factors. Considering the unique global circumstances of that year, some specific possibilities include:

1. COVID-19 Pandemic: The outbreak of the COVID-19 pandemic had profound impacts on almost all sectors, including the music industry. Lockdowns and social distancing measures led to:

- Changes in Listening Habits: People were spending more time at home, potentially altering their music consumption habits. There may have been a shift towards more calming, soothing music or podcasts, which could affect the popularity metrics of tracks that were previously popular.
- Disruption of Music Production: Many artists and producers faced challenges in recording and releasing new music due to restrictions on gatherings, affecting the availability of new, popular tracks.

2. Economic Downturn: The economic impact of the pandemic may have led consumers to cut back on discretionary spending, including subscriptions to paid streaming services, potentially affecting the algorithms that drive track popularity and visibility.
3. Shift in Public Events: The cancellation of live events, festivals, and concerts, which often drive up the popularity of associated tracks, likely had a significant impact. Without these events, many tracks may not have received the usual publicity boost.

Research question 2: What common variables exist between the top hits of 2017 and 2020.

1). 2017 data wrangling, tidying and visualization.

i. Artists with the most Top Hits in 2017

```
artist_data2017 <- wrangled_spotify_data %>%
  group_by(artist_name) %>%
  filter(year == 2017) %>%
  summarize(number_of_songs = n(), .groups = "drop") %>%
  arrange(desc(number_of_songs))

head(artist_data2017)
```

```
## # A tibble: 6 × 2
##   artist_name     number_of_songs
##   <chr>                     <int>
## 1 Ed Sheeran                    5
## 2 Kendrick Lamar                4
## 3 Imagine Dragons               3
## 4 Avicii                        2
## 5 Bruno Mars                    2
## 6 Calvin Harris                 2
```

ii). Table showing one song of each top 10 artists in 2017 with their variables.

```
top_ten_artists_data2017 <- wrangled_spotify_data %>%
  filter(year == 2017 &
         artist_name %in% c("Ed Sheeran", "Kendrick Lamar", "Imagine Dragons", "Avicii", "Bruno Mars",
                            "Calvin Harris", "Clean Bandit", "Drake","Khalid" , "Lorde")) %>%
  group_by(artist_name) %>%
  arrange(desc(track_popularity), desc(year)) %>%  # Arrange by popularity and year within groups
  slice_head(n = 1) %>%
  select(track_name, year, track_popularity, danceability, valence, instrumentalness, energy)
```

```
## Adding missing grouping variables: `artist_name`
```

```
head(top_ten_artists_data2017)
```

```
## # A tibble: 6 × 8
## # Groups:   artist_name [6]
##   artist_name   track_name            year track_popularity danceability valence
##   <chr>         <chr>                <dbl>            <dbl>        <dbl>   <dbl>
## 1 Avicii        Without You (feat. …  2017               80        0.662   0.295
## 2 Bruno Mars    That's What I Like    2017               87        0.853   0.86
## 3 Calvin Harris Feels (feat. Pharre…  2017               76        0.893   0.872
## 4 Clean Bandit  Symphony (feat. Zar…  2017               77        0.715   0.454
## 5 Drake         Passionfruit          2017               84        0.809   0.364
## 6 Ed Sheeran    Perfect               2017               87        0.599   0.168
## # ℹ 2 more variables: instrumentalness <dbl>, energy <dbl>
```

iii). Tidy 2017 data using pivot

```
# Pivot the table

top_ten_artists_data2017_tidy <- top_ten_artists_data2017 %>%
  pivot_longer(names_to = "features",
               values_to = "numbers",
               cols = c(danceability, valence, energy, instrumentalness))

head(top_ten_artists_data2017_tidy)
```

```
## # A tibble: 6 × 6
## # Groups:   artist_name [2]
##   artist_name track_name                 year track_popularity features numbers
##   <chr>       <chr>                     <dbl>            <dbl> <chr>      <dbl>
## 1 Avicii      Without You (feat. Sandro…  2017               80 danceab…   0.662
## 2 Avicii      Without You (feat. Sandro…  2017               80 valence    0.295
## 3 Avicii      Without You (feat. Sandro…  2017               80 energy     0.858
## 4 Avicii      Without You (feat. Sandro…  2017               80 instrum…   0
## 5 Bruno Mars  That's What I Like          2017               87 danceab…   0.853
## 6 Bruno Mars  That's What I Like          2017               87 valence    0.86
```

iv). Facet Barplot

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
ggplot(subset(top_ten_artists_data2017_tidy), aes(x = artist_name, y = numbers, fill = features)) +
  geom_col(position = position_dodge(width = 0.5)) +
  facet_wrap(~ features, scales = "free_x") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        strip.text.x = element_text(size = 10, face = "bold")) +
  scale_y_continuous(labels = label_number())
```

2. 2020 data wrangling, tidying and visualization.

i). Artists with the most Top Hits in 2020

```
artist_data2020 <- wrangled_spotify_data %>%
  group_by(artist_name) %>%
  filter(year == 2020) %>%
  summarize(number_of_songs = n(), .groups = "drop") %>%
  arrange(desc(number_of_songs))

head(artist_data2020)
```

```
## # A tibble: 6 × 2
##   artist_name   number_of_songs
##   <chr>                   <int>
## 1 Bad Bunny                   4
## 2 Billie Eilish               4
## 3 Justin Bieber               4
## 4 Ariana Grande               3
## 5 The Weeknd                  3
## 6 BLACKPINK                   2
```

ii). Table showing one song of each top 10 artists in 2017 with their variables.

```
top_ten_artists_data2020 <- wrangled_spotify_data %>%
  filter(year == 2020 &
         artist_name %in% c("Bad Bunny", "Billie Eilish", "Justin Bieber", "Ariana Grande", "The Weeknd",
                            "BLACKPINK", "BTS", "Black Eyed Peas","Doja Cat" , "Drake")) %>%
  group_by(artist_name) %>%
  arrange(desc(track_popularity), desc(year)) %>%
  slice_head(n = 1) %>%
  select(track_name, year, track_popularity, danceability, valence, instrumentalness, energy)
```

```
## Adding missing grouping variables: `artist_name`
```

```
head(top_ten_artists_data2020)
```

```
## # A tibble: 6 × 8
## # Groups:   artist_name [6]
##   artist_name     track_name           year track_popularity danceability valence
##   <chr>           <chr>               <dbl>            <dbl>        <dbl>   <dbl>
## 1 Ariana Grande   "Stuck with U (wi…  2020               80        0.597   0.537
## 2 BLACKPINK       "Ice Cream (with …  2020               62        0.79    0.904
## 3 BTS             "Dynamite"          2020                0        0.746   0.737
## 4 Bad Bunny       "D\xc1KITI"         2020               78        0.731   0.145
## 5 Billie Eilish   "everything i wan…  2020               83        0.704   0.243
## 6 Black Eyed Peas "MAMACITA"          2020               65        0.894   0.428
## # ℹ 2 more variables: instrumentalness <dbl>, energy <dbl>
```

iii). Tidy 2020 data using pivot

```
top_ten_artists_data2020_tidy <- top_ten_artists_data2020 %>%
  pivot_longer(names_to = "features",
               values_to = "numbers",
               cols = c(danceability, valence, energy, instrumentalness))

head(top_ten_artists_data2017_tidy)
```

```
## # A tibble: 6 × 6
## # Groups:   artist_name [2]
##   artist_name track_name                 year track_popularity features numbers
##   <chr>       <chr>                     <dbl>            <dbl> <chr>      <dbl>
## 1 Avicii      Without You (feat. Sandro…  2017               80 danceab…   0.662
## 2 Avicii      Without You (feat. Sandro…  2017               80 valence    0.295
## 3 Avicii      Without You (feat. Sandro…  2017               80 energy     0.858
## 4 Avicii      Without You (feat. Sandro…  2017               80 instrum…   0
## 5 Bruno Mars  That's What I Like          2017               87 danceab…   0.853
## 6 Bruno Mars  That's What I Like          2017               87 valence    0.86
```

iv). Facet Barplot

```
library(ggplot2)
library(scales)

ggplot(subset(top_ten_artists_data2020_tidy), aes(x = artist_name, y = numbers, fill = features)) +
  geom_col(position = position_dodge(width = 0.5)) +
  facet_wrap(~ features, scales = "free_x") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate and adjust x-axis text
        strip.text.x = element_text(size = 10, face = "bold")) +
  scale_y_continuous(labels = label_number())
```

Analysis:

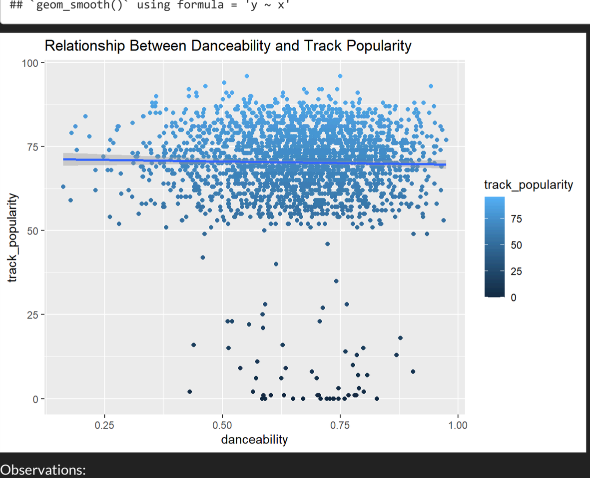
- Both 2017 and 2020 popular track show to be suitable for dancing as they all show high values (>.6) of danceability.
- It is observed that tracks in 2020 tends to have a higher energy and valence as compared to the tracks in 2017. A possible explanation to this could be that during times of crisis and uncertainty, such as the COVID-19 pandemic, people often seek out music that uplifts and energizes them, providing a sense of comfort and escape from the stresses of daily life. With lockdowns and social distancing measures in place during 2020, people spent more time at home and online, potentially leading to changes in music listening habits. The higher energy and valence of tracks in 2020 could be a reflection of a collective need for positivity and strength.
- Overall, the barplots reveal that the least common and least important feature to the popularity of a track both in 2017 and in 2020 is intsrumentalness meanwhile, danceability, valence and energy are all common variables in popular tracks.

PART III: Regression Analysis

1. Scatter plot with regression line to examine relationships between track_popularity and danceability within the years 2000 to 2023.

```
ggplot(wrangled_spotify_data, aes(x = danceability, y = track_popularity)) +
  geom_point(aes(color = track_popularity)) +
  geom_smooth(method = "lm") +
  labs(title = "Relationship Between Danceability and Track Popularity")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Observations:

1. **High Concentration of Points**: Most of the data points are concentrated around the regression line, covering the range of approximately 50 to 90 on the track popularity (y-axis) and approximately 0.1 to 0.9 on danceability (x-axis).
2. **Few Outliers**: There are relatively few points not around the regression line, which are in the lower popularity range approximately from 1 to 30 on the y-axis and a moderate to high danceability range approximately from 0.4 to 0.83 on the x-axis.

Analysis:

1. The high concentration of data points around the regression line in the specified ranges suggests a strong positive correlation between danceability and track popularity. This implies that as a track's danceability increases, its popularity tends to increase, particularly within this danceability range.
2. The approximated range of danceability from 0.1 to 0.9 captures most types of music suitable for dancing, from lightly rhythmic to highly rhythmic, which are likely favored by listeners, thereby boosting their popularity scores.

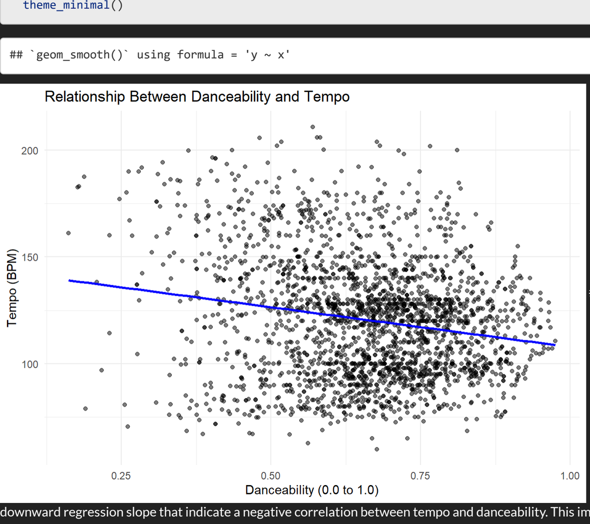
Implication of Data Spread - The dense clustering of points around the regression line indicates a consistent pattern where danceability is a significant factor in determining a track's popularity. This consistency suggests that for a majority of tracks, higher danceability could be a crucial element in their appeal and market success.

Analysis of Lower Popularity Tracks - The few points that do not align with the regression line, which fall into the approximated lower popularity range (1 to 30) but still possess a moderate to relatively high danceability (0.4 to 0.83), suggest exceptions to the general trend. This could be interpreted in several ways: - **Other Influencing Factors**: These tracks, despite having decent danceability, might be lacking in other areas that affect popularity, such as genre appeal, or artist popularity

2. Scatter plot with regression line examining how the tempo of a track influence its danceability.

```
ggplot(spotify_data, aes(x = danceability, y = tempo)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Relationship Between Danceability and Tempo",
       x = "Danceability (0.0 to 1.0)",
       y = "Tempo (BPM)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Analysis: - We can observe a downward regression slope that indicate a negative correlation between tempo and danceability. This implies that as tempo decreases, danceability increase and as tempo increases, danceability decrease. I am very surprised as I was expecting the opposit of this i.e tempo and danceability increasing and decreasing together.

Hypothesis Testing to statistically test if tracks with a tempo above a certain threshold (e.g., 120 BPM) have significantly higher danceability scores.

```
# Split the data into two groups based on tempo
group_low_tempo <- filter(spotify_data, tempo <= 120)
group_high_tempo <- filter(spotify_data, tempo > 120)

# Perform a t-test to compare danceability scores between the two groups
t_test_results <- t.test(group_low_tempo$danceability, group_high_tempo$danceability, alternative = "less")

print(t_test_results)
```

```
##
##  Welch Two Sample t-test
##
## data:  group_low_tempo$danceability and group_high_tempo$danceability
## t = 6.2661, df = 2382.8, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf 0.04497232
## sample estimates:
## mean of x mean of y
## 0.6795620 0.6439433
```

Results Interpretation:

1. **p-value**: The p-value is reported as 1, which is highly unusual and typically indicates no effect or no significant difference under the tested hypothesis. In the context of hypothesis testing, this p-value suggests that the data do not provide sufficient evidence to reject the null hypothesis. For this test setup where the alternative hypothesis is that the true difference in means is less than 0 (indicating group_low_tempo would have a lower mean danceability than group_high_tempo), the results do not support this.
2. **Confidence Interval**: The 95% confidence interval for the difference in means ranges from negative infinity to 0.04497232. This interval includes 0, further supporting the conclusion that there is no significant difference where the danceability of the low-tempo group is less than that of the high-tempo group. Since zero is included and the interval is skewed positive, it suggests that the low-tempo group could have equal or slightly higher danceability, but not significantly less.

Based on the p-value and confidence interval provided by your t-test, there is no statistical evidence to conclude that tracks with tempos over 120 BPM have a lower danceability than tracks with tempos at or below 120 BPM. In fact, the data suggest the opposite might be true, but not to a statistically significant degree under the specified conditions of the test.

PART IV: Concluding Remarks and Key Takeaways from the Analysis of Spotify Tracks

Insights from Track Popularity Over the Years

Our analysis revealed that **2017** marked the peak of average track popularity, while **2020** experienced the lowest. This dip in popularity during 2020 can largely be attributed to the global outbreak of COVID-19, which fundamentally altered listening habits, artist production capabilities, and the economic conditions influencing consumer spending on entertainment. The pandemic likely shifted listener preferences towards more soothing, home-friendly music and podcasts, impacting the popularity metrics traditionally dominated by upbeat or live performance-driven tracks.

Common Characteristics of Top Hits in 2017 and 2020

Despite the tumultuous landscape of 2020, top hits maintained high danceability, suggesting a consistent preference for tracks that facilitate a positive and energetic experience. Interestingly, tracks in 2020 displayed higher energy and valence compared to those in 2017, possibly reflecting a collective psychological response to seek uplifting content amidst the pandemic's challenges. This underscores the role of music as a therapeutic and unifying force during hard times.

Danceability and Track Popularity

The regression analysis highlighted a strong positive correlation between danceability and track popularity, with dense clustering around the regression line confirming danceability as a critical factor in a track's market success. However, the presence of some tracks with moderate to high danceability but low popularity indicates that other factors also play significant roles.

The Surprising Relationship Between Tempo and Danceability

Contrary to expectations, our analysis suggested a negative correlation between tempo and danceability, indicating that slower tempos might enhance a track's danceability. This counterintuitive finding suggests that while fast beats are typically associated with dance tracks, slower rhythms might allow for more expressive and varied dance styles.

Hypothesis Testing on Danceability Across Tempos

The hypothesis testing did not find significant differences in danceability between tracks above and below 120 BPM, indicating no substantial effect of tempo on danceability across these groups. This outcome suggests that tempo alone does not dictate a track's danceability, and listeners might appreciate a broader range of tempos in dance music than traditionally assumed.

Overall Conclusion

This analysis underscores the complexity of musical preferences and trends, revealing that while certain attributes like danceability consistently influence track popularity, external factors such as global crises can significantly alter listening habits. Additionally, unexpected findings like the negative correlation between tempo and danceability challenge traditional notions and invite further exploration into how and why people interact with music in various contexts.

Moving forward, stakeholders in the music industry, from producers to marketers, can leverage these insights to better align their offerings with listener preferences, potentially using targeted strategies to cater to changing tastes and conditions. Moreover, continuing to analyze emerging data will be crucial in staying responsive to the dynamic landscape of music consumption.