User Engagement As A Function Of Audio Attributes Using Multivariate Regression

Jack Fahrnow, Luke Atkins, Akshika Rai, Yichen Gao

***Abstract*—Through the use of the Regression model of the multivariable variety, conclusions can be reached about the relationships between these Spotify attributes and the Listeners. Comprehending the relationship between these two will help us predict a listener's next song on the list and make that next song be better than what the Listener plays already.**

I. Background

Spotify is the world's largest music streaming service provider and has 381 million monthly active listeners which also includes 171 million paid subscribers. Spotify had around 1.1 billion downloads on the google store in 2021 alone. So, millions of people listen to music all day including me. As an analyst, what's better than exploring quantifying data music and drawing valuable insights. Celebrity artists on Spotify would continuously be making tracks and hoping one day one of their albums or songs will eventually go viral. For their songs to go viral, they would have to upgrade and make better songs and constantly test their music until they find a gold mine. Artists post their tracks on Spotify using a separate software called, Ditto, which helps artists release unlimited tracks in 100 different apps or stores. It varies how the Spotify artist gets paid however. It depends on what location of where the listener is listening to the track and whether the stream came from a paid or free subscription account. In order for an artist to go viral on Spotify, they would need to focus on a certain niche for their song. They have the opportunity to explore and create content beyond music by incorporating their passions and interests into their daily conversation with their fans.

II. Goal & Potential Uses

The goal of the project is to predict the niche a musical artist has on spotify. We would like to find the attribute of their music that contributes to the popularity of their songs. We measure this with the error of the model by sum square error, and find which attribute, when removed, increases the error the most. A model which can find the niche of an artist could have applications in artist and fan engagement. Another possible outcome could be improved music recommendation methods and more efficient music promotion on streaming platforms.

III. Dataset/Coding

The two datasets used to create the linear models were collected directly from spotify. Each artist had around 200 songs. The API was able to collect all of the public music published by each artist from their account on the service.

```python
sp = spotipy.Spotify(auth_manager=SpotifyClientCredentials(client_id=client_id,
                                                           client_secret=client_pass))
# artist Id's of Megan The Stallion, Father John Misty, Foo Fighters
megan, fjMisty, fooFight = '181bsRPaVXVlUKXrxwZfHK', '2kGBy2WHvF0VdZyqiVCkDT','7jy3rLJdDQY21OgRLCZ9sD'

# megan the stallion...
megans = sp.artist_albums(artist_id=megan, country = 'US')
mItems = megans['items']
meganSongIDs = []

# iterate over the albums of the artists or any type, this includes singles and mixtapes
for i in mItems:
   albumID = i['id']
   tracks = sp.album_tracks(album_id=albumID)
   # create a list of the song id's
   for j in range(len(tracks['items'])):
        meganSongIDs.append((tracks['items'][j]['id']))
mfeature = [sp.audio_features(meganSongIDs[id])[0] for id in range(len(meganSongIDs))]
with open('MeganTheStallion.csv', 'w', newline='') as output_file:
    dict_writer = csv.DictWriter(output_file, keys)
    dict_writer.writeheader()
    dict_writer.writerows(mfeature)
```

FooFighters

danceability	energy	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	popularity
0.277	0.992	-6.237	0.0856	3.55E-06	0.836	0.272	0.148	103.494	98293	44
0.38	0.969	-6.147	0.0649	0.000764	0.0241	0.304	0.352	132.869	313373	47
0.17	0.998	-4.585	0.232	0.000136	0.352	0.757	0.06	162.402	211000	40
0.111	0.978	-5.33	0.0737	8.76E-06	0.76	0.257	0.303	165.373	259813	41
0.253	0.949	-6.126	0.111	9.72E-06	0.00809	0.135	0.16	161.611	272880	38

IV. Methods and Measures Used

This project used the Python programming software to create the two datasets and we created the two regression models using R Studio. Spotipy, a python library, was used to interact with the API and collect data. We use R to create the linear regression. All data was stored as CSV files.

V. Spotify Attributes and What They Mean

A. The Spotify API attributes used in these Models

The Spotify API attributes we are using are danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_ms, and rank. These attributes help us determine what the next recommended song will be for the listener, or for the artist's perspective, when their next song will go viral. Danceability describes how suitable a track is for dancing based on a combination of music elements, meaning the strength of the beat and the overall regularity. Loudness tells us the decibels of the track.

speechiness detects the presence of spoken words, the lyrics in the music. Acousticness measures the confidence measure whether the track is acoustic. Instrumentalness predicts whether a track contains no vocals, so whenever there are no lyrics, that is how it detects the instrumentalness. Liveness detects the presence of an audience, shows that the track was performed live, so if there are clapping and people cheering, liveness detects that. Valence shows the musical positiveness conveyed by a track, a high valence means the music sounds happy as for the negative being sad. Tempo is the overall beats per minute/pace in a track. Duration_ms is the length of the track. Rank is the ranking of the track from top to bottom of the data. It would also mean the popularity of a song, or how popular the song is.

B. Attributes that weren't being used

There were several attributes that we didn't include such as mode, key, URI, e.t.c. because these attributes do not work best with the regression model. Mode indicates if the track is major or minor, Major indicating 1 and Minor indicating 2. And the key just tells us the pitch class with some notations. The regression model works best with continuous variables, so we had no other choice but to remove them in order for these models to work.

VI. Data Preparations and Regression Model

```
x = read.csv2("FatherJohnMisty.csv",stringsAsFactors=FALSE,sep=",");
x = sapply(x, as.character)
x = data.matrix(x)
x = apply(x, 2, as.numeric) #changing the data type from character to numeric

new_data = scale(x[,9:11]) #Standardizing the dataset
View(new_data)
new_x = cbind(x[,1:8],new_data) #Binding the columns together
View(new_x)

X = as.matrix(new_x[,1:10]);
y = as.vector(new_x[,11]);
a = solve(t(X) %*% X, t(X) %*% y)
paste("coefficients are:")
a
```

```
yhat = X %*% a
sse = sum((y - yhat) ^ 2)
paste("sse =", sse) #Finding the sum of squared errors

new_sse = rep(0, 10)
diff = rep(0, 10)
for (i in 1:10) {
    newX = X[,-i]
    a = solve(t(newX) %*% newX, t(newX) %*% y)
    yhat = newX %*% a
    new_sse[i] = sum((y - yhat) ^ 2)
    diff[i] = abs(new_sse[i] - sse) #Computing the new sum of squared errors by omitting one attribute at a
time.
    cat(
        paste0(
            "Omit attribute", i , ": new sse = ", new_sse[i],", difference with old sse = ",diff[i],".\n" )
    )
```

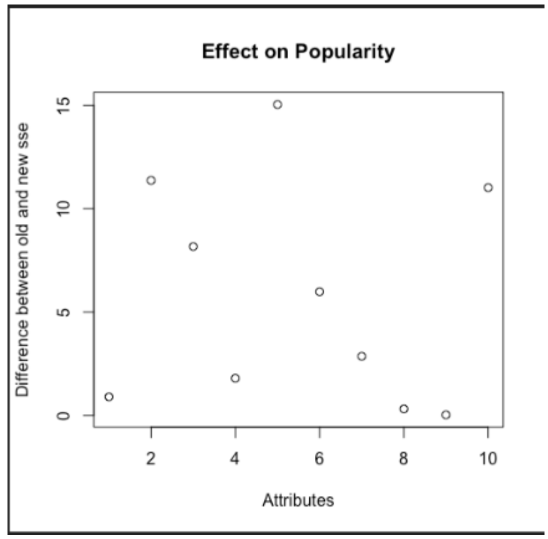
```
}
cat("Most difference is", max(diff), "\n") #Popularity most affected by this attribute

cat("Least difference is", min(diff), "\n") #Popularity least affected by this attribute

plot(diff, main = "Effect on Popularity", ylab = "Difference between old and new sse", xlab = "Attributes")
#Goal of the model is to output the attribute which causes the most and least difference between the new sse and
the old sse
#Which will help us determine the attribute with the most and least effect on the popularity of an artist's track
```

VI. Results and Interpretation

```
Most difference is 1.911314
Least difference is 0.005982919
```

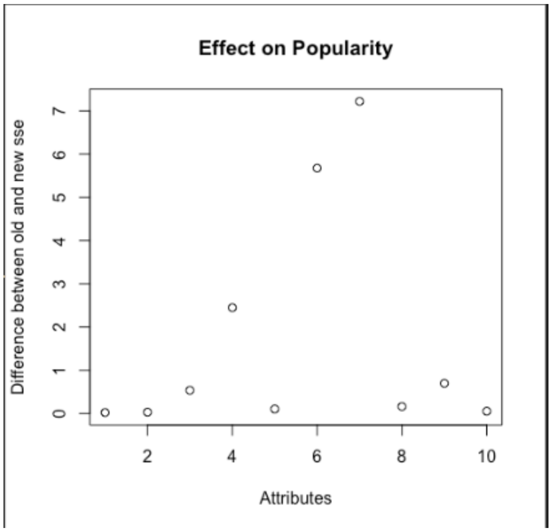


Father John Misty Results and Interpretation

Father John Misty is an Alternative/Indie artist whose songs' popularity, as observed in the plot above, is most affected by the attribute acousticness. On the contrary, the attribute tempo least affects the popularity of his songs.

As an Alternative/Indie artist, his songs use acoustic and non-electronic musical instruments. Therefore, the acousticness of his songs is high leading to a high level of popularity because fans of the genre prefer a high level of acousticness in tracks. Whereas the tempo of a song is high for rock music involving electronic musical instruments, it is low for songs that use acoustic and non-electronic musical instruments. Therefore, having less effect on the popularity of an Alternative/Indie artist's tracks.

```
Most difference is 7.220158
Least difference is 0.02211429
```



Foo Fighters Results and Interpretation

Foo Fighters is a Rock band whose songs' popularity, as observed in the plot, is most affected by the liveness of their tracks. On the contrary, a track's popularity is least affected by its danceability.

Foo Fighters fans or fans of Rock music appreciate a track more when it has a higher level of liveness because there is a genre-specific link between liveness and authenticity in rock music. Rock fans highly value that authenticity. Therefore, the attribute of liveness affects the popularity of Foo Fighters' songs the most. In contrast, the danceability of a track does not affect the popularity of the Rock band's tracks because the usual tempo range for danceable tracks is 128 bpm (beats per minute). All Foo Fighters tracks are over this level, averaging 130 bpm. Therefore the popularity of their tracks is least affected by the attribute of danceability. Their fans expect their music to be something other than danceable.

VI. Challenges and Future Direction

What we could have done differently or the next model/algorithm we would use in the future would be the approximate nearest-neighbor search algorithm. This method would’ve been way easier because it helps group songs up for the listener and the listeners together based on shared attributes or qualities. As for the future directions of the research, our team would like to explore how we could incorporate their challenging parameters into our model. Creating a decision tree or looking for a new type of model are potential next steps.

VII. Contributions of the Authors

These are the group members that helped analyze the listeners contributions to the group’s regression analysis of the Spotify data attributes. Luke Atkins wrote the Python code for the API calls to Spotify and generated the raw data set with some initial cleaning while creating presentation slides and assisting the final report. Jack Fahrnow helped with the presentation and the final report of the project, he also assisted in formatting the final report into an IEEE style format. Akshika Rai created the multi-linear regressions in R for both artists and generated outcomes while creating presentation slides and aiding in the writing of the report.

References

[1] Ashrith. (2019, March 22). *What makes a song likeable?* Medium. Retrieved December 5,2022, from https://towardsdatascience.com/what-makes-a-song-likeable-dbfdb7abe404

[2] Plantinga, B. (2018, April 29). *What do Spotify's audio features tell us about this year's Eurovision Song Contest? 🤔*. Medium. Retrieved December 5, 2022, from https://medium.com/@boplantinga/what-do-spotifys-audio-features-tell-us-about-this-year-s-eurovision-song-contest-66ad188e112a#:~:text=Spotify%20Audio%20Analysis&text=A%20value%20of%200.0%20is,fast%2C%20loud%2C%20and%20noisy

[3] Bode, A. (2021, October 1). *Spotify API and audio features*. Medium. Retrieved December 5, 2022, from https://towardsdatascience.com/spotify-api-audio-features-5d8bcbd780b2

[4] Ditto Inc. (n.d.). *The simple way to get your music on Spotify: Ditto music*. How To Get Your Music on Spotify | Keep 100% | Ditto Music. Retrieved December 5, 2022, from https://dittomusic.com/en/sell-your-music/spotify/

[5] Ostrow, J. (2014, February 17). *Finding and nurturing your musical niche*. Disc Makers Blog. Retrieved December 5, 2022, from https://blog.discmakers.com/2014/01/finding-and-nurturing-your-musical-niche/#:~:text=By%20focusing%20on%20your%20niche,daily%20conversation%20with%20your%20fans.