Song vs. Music Video Popularity

Team: Checked-out second semester seniors

Meredith Grife and Anjali Sharma

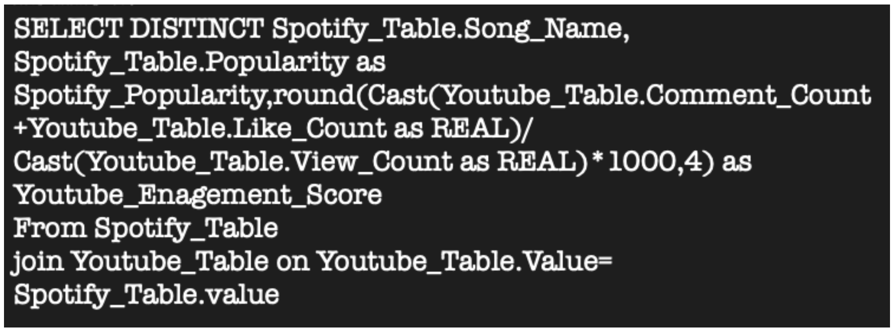
The aim of our project was to examine the correlation between a song's popularity and the popularity of its corresponding music video on YouTube. There are several factors to consider in the relationship between a song's success on Spotify versus YouTube. We hypothesized that songs might be more popular on YouTube due to its more engaging features such as likes, dislikes, and comments. However, the convenience of Spotify, which does not require visual content and can be accessed on mobile devices even when offline, may also contribute to its popularity. To conduct our analysis, we utilized the Spotify and YouTube APIs.

We gathered data from the Spotify and Youtube APIs. We utilized a csv file called charts.csv that has the songs from the Billboards top 100s. With this we used Spotify's API to collect the Artist Name, Name of the Song, and its popularity. From Youtube's API we collected the music video name, view count, like count, dislike count, and comment count.

Our initial approach to comparing the success of a song on Spotify vs. Youtube was to simply compare the number of plays of the track and views of its music video. However, after researching the Spotify API, we learned that it won't let you extract the number of plays of a song. The only information that it will give is its popularity score. This is determined algorithmically by Spotify based on total streams, how recently played, and frequency played. However, we don't know how it's calculated. Thus, we had to shift our plan and utilize Spotify's popularity score and give Youtube its own Engagement Score.

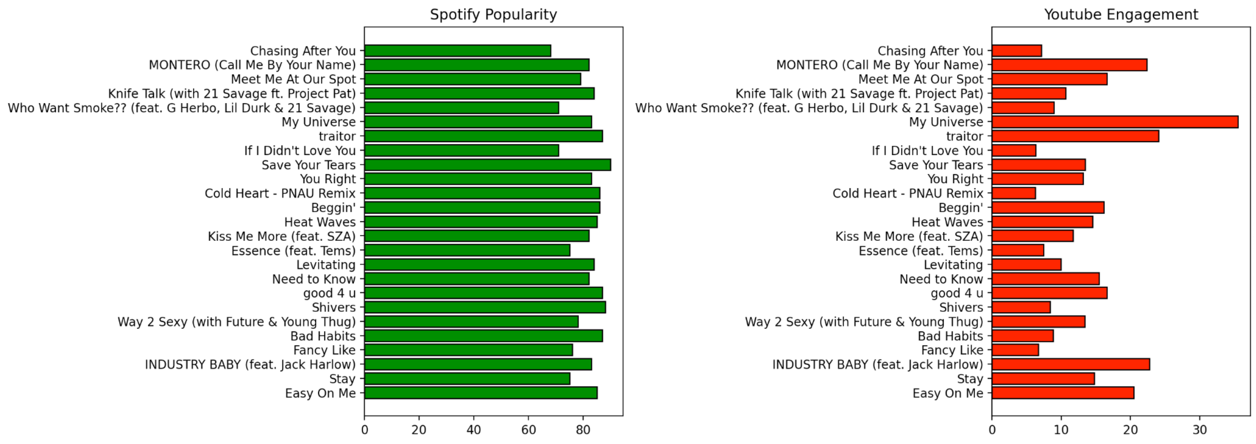
We calculated the Youtube engagement as 100*(number of likes + number of comments) / (number of views) using the Select and Join statement pictured below.

The calculations from the data in the database

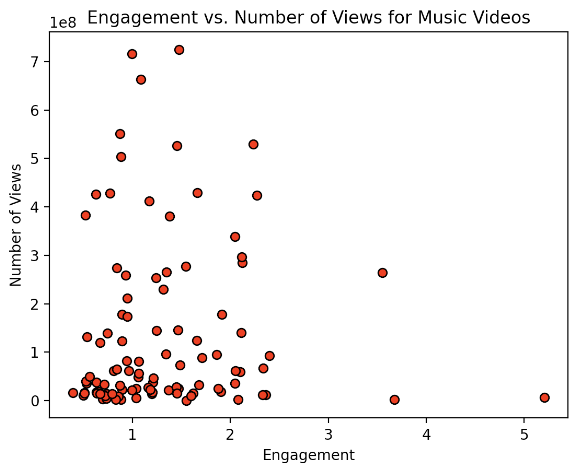


```
SELECT DISTINCT Spotify_Table.Song_Name,
Spotify_Table.Popularity as
Spotify_Popularity,round(Cast(Youtube_Table.Comment_Count
+Youtube_Table.Like_Count as REAL)/
Cast(Youtube_Table.View_Count as REAL)*1000,4) as
Youtube_Enagement_Score
From Spotify_Table
join Youtube_Table on Youtube_Table.Value=
Spotify_Table.value
```

From there, we created two visualizations: side-by-side vertical bar graphs of selected songs' Spotify popularity and Youtube engagement and a scatter plot of the engagement vs. the number of views on the music videos. Both graphs are pictured below.



Some songs with high popularity do not have music videos with high engagement, such as "If I Didn't Love You" and "Fancy Like". This could be due to the music videos not being well-received by viewers or not interesting enough for people to like and comment on the Youtube video. Some songs have very high engagement, like "My Universe" by Coldplay X BTS. This could be due to the popularity of the groups, which can encourage more people to like and comment on the video.



The scatter plot above shows that music videos with a very high number of views do not appear to have a very high engagement rate. This could be because maybe people replay the video many times without commenting each time. Additionally, a user can only "like" the video once, so even if they really enjoy a music video and watch it often, they can't re-like it. In contrast, there are a few music videos that have very high engagement, but low viewership. This could be due to maybe more dedicated fans willing to like and comment on the video. It could also be that the video isn't as popular, so not many people view it.

Instructions for Running Code:

Download "charts.csv"

```
Open "SI206FinalProject.py"
        Run     early = get_track_ids(read_in_top_songs())
                get_popularity_score(early)
        Run
                ids = read_in("charts.csv")
                vids = get_video_stats(ids)
                write_out("youtube.csv", vids)
        Run     create_Spotify_db()
        Run     create_Youtube_db()

Open "database_and_visuals.py"
        Run     create_text_file_from_DB()
        Run     processed = process_data("Spotify_And_Youtube_DB.csv")
                double_bar(processed)
        Run     y_data = process_youtube("youtube.csv")
                scatter(y_data)
```

Function Documentation:

database_and_visuals.py:

```
def create_text_file_from_DB()
        Input: Nothing
        Output: Nothing

def process_data()
        Input: filename (Type: String)
        Output: songs, popularity, engagement score (Type:Tuple)

def double_bar()
        Input: songs, popularity, engagement score (Type:Tuple)
        Output: nothing
def process_youtube()
        Input: filename (Type: String)
        Output: views, engagement (Type: Tuple)
```

def scatter():
Input: views, engagement (Type: Tuple)
Output: nothing

SI206FinalProject.py:

def CreateToken()
Input: Nothing
Output: Spotify access Token (Type: String)

def read_in_top_songs()
Input: Nothing
Output: 100 top song name from Charts.csv (Type: List)

def get_track_ids()
Input: List of song names
Output: Spotify Track IDs (Type: List)

def get_popularity_score()
Input: List of Spotify Track IDs
Output: Nothing

def read_in():
Input: filename (Type: String)
Output: video ids (Type: List)

def write_out():
Input: filename (Type: String), tup_in (Type: Tuple)
Output: nothing

def get_video_stats():
Input: Youtube ids (Type: List)
Output: Statistics of Youtube ids (Type: List)

def create_Spotify_db()
Input: Nothing
Output: Nothing

def create_Youtube_db()
Input: Nothing
Output: Nothing

Resources Used

Date	**Issue Description**	**Location of Resource**	**Result (did it solve the issue)**
04/15	Spotify Access token expiring every hour	Spotify for Developers	Yes
04/16	Unable to access Youtube private data	Youtube API Documentation	Yes
04/16	Error: No such file or directory: ‘Charts.csv’	w3schools Python File handling	Yes
04/17	Struggled with the formatting of the visuals	Matplotlib.org	Yes