Spotify And Youtube Dataset

Sophie Nguyen

Georgia State University

CSC 1302: Principles of Computer Science II

Dr. Tushara Sadasivuni

17 April 2023

About Dataset

1. **Dataset of songs of various artists, and each song is present:**
    - Statistics of the music version on Spotify.
    - The number of views of the music video on Youtube.
2. **Contents used in this report:**
    - *Acousticness:* indicates whether the track is acoustic.
    - *Album_type:* indicates if the song is released on Spotify as a single or contained in an album.
    - *Danceability:* describes how suitable a track is for dancing.
    - *Energy:* represents a perceptual measure of intensity and activity.
    - *Key:* the key the track is in, using Standard Pitch Class Notation.
    - *Licensed:* indicates whether the video represents licensed content.
    - *Loudness:* the overall loudness of a track in decibels (dB).
    - *official_video:* indicates if the video found is the official video of the song.
    - *Stream:* number of streams of the song on Spotify.
    - *Track:* name of the song.
    - *Valence:* describes the musical positiveness conveyed by a track.
    - *Views:* number of views.

Spotify And Youtube Dataset

1. Data preprocessing: *Import pandas library.*

- Read the dataset: *pd.read_csv()*
- Remove unused columns: *drop()*
- Fill missing values with the mean of the columns: *fillna(mean())*
- Remove duplicates: *drop_duplicates()*

2. Calculate mean, median, variance, and standard deviation of these 2 features (Energy and Key): *Import pandas library to use mean(), median(), var(), and std() functions.*

a. Energy:

Mean of Energy column is: 0.6353

Median of Energy column is: 0.6660

Variance of Energy column is: 0.0459

Standard deviation of Energy column is: 0.2141

=> Energy column of the dataset says the following:

- The mean energy value means that, on average, the majority of the songs in the dataset have a moderate level of perceived intensity and activity.
- The median is higher than the mean, suggesting that there may be some songs in the dataset with relatively high energy levels that are pulling the median up.
- There are relatively small variance and standard deviation, indicating that the energy values in the dataset are not widely spread out from the mean.

b. Key:

Mean of Key column is: 5.3003

Median of Key column is: 5.0000

Variance of Key column is: 12.7910

Standard deviation of Key column is: 3.5764

=> Key column of the dataset says the following:

- The mean suggests that the average key of the tracks in the dataset is around F♯/G ♭ (since 5 represents F and 6 represents G in the pitch class notation).
- The median is lower than the mean value, suggesting that the distribution of the Key column is slightly skewed towards lower values.

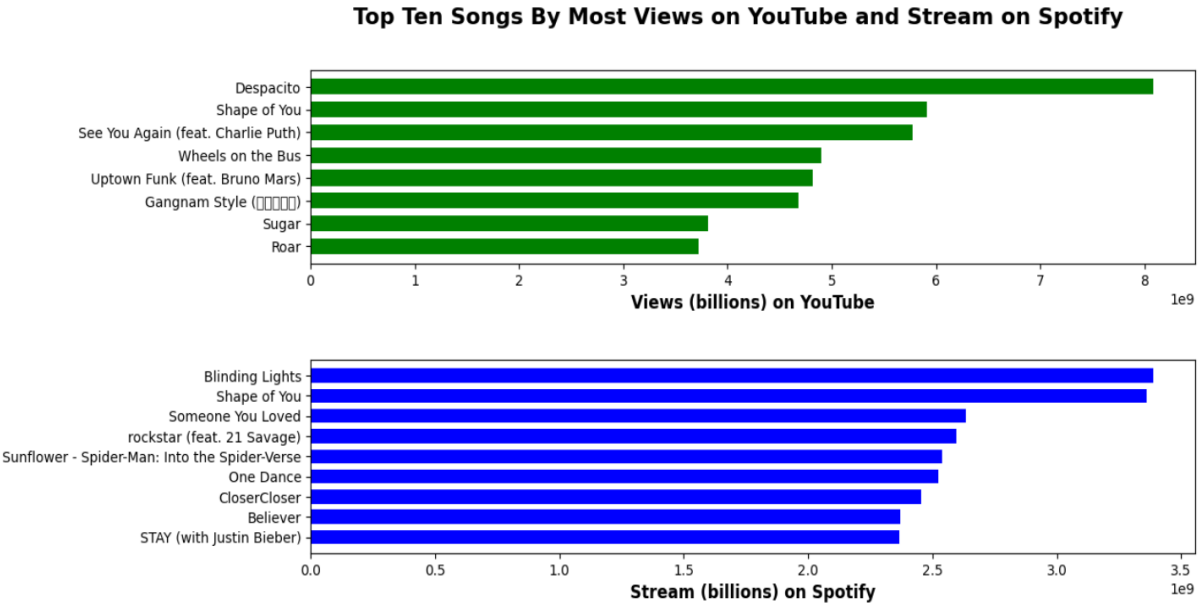
- There are large variance and standard deviation, which indicates that the key signatures of the tracks are quite spread out. In other words, the Key column has a relatively high level of variation.

3. Visualize the data: *Import matplotlib library.*

a. Bar chart (only 1 horizontal bar at 1 item): **=> *The chart says the following:***

- Create subplots: *subplots()*
- Create horizontal bars: *barh()*
- Set label and title: *set_xlabel(), suptitle()*
- Invert y-axis: *invert_yaxis()*
- Adjust the spacing: *subplots_adjust()*

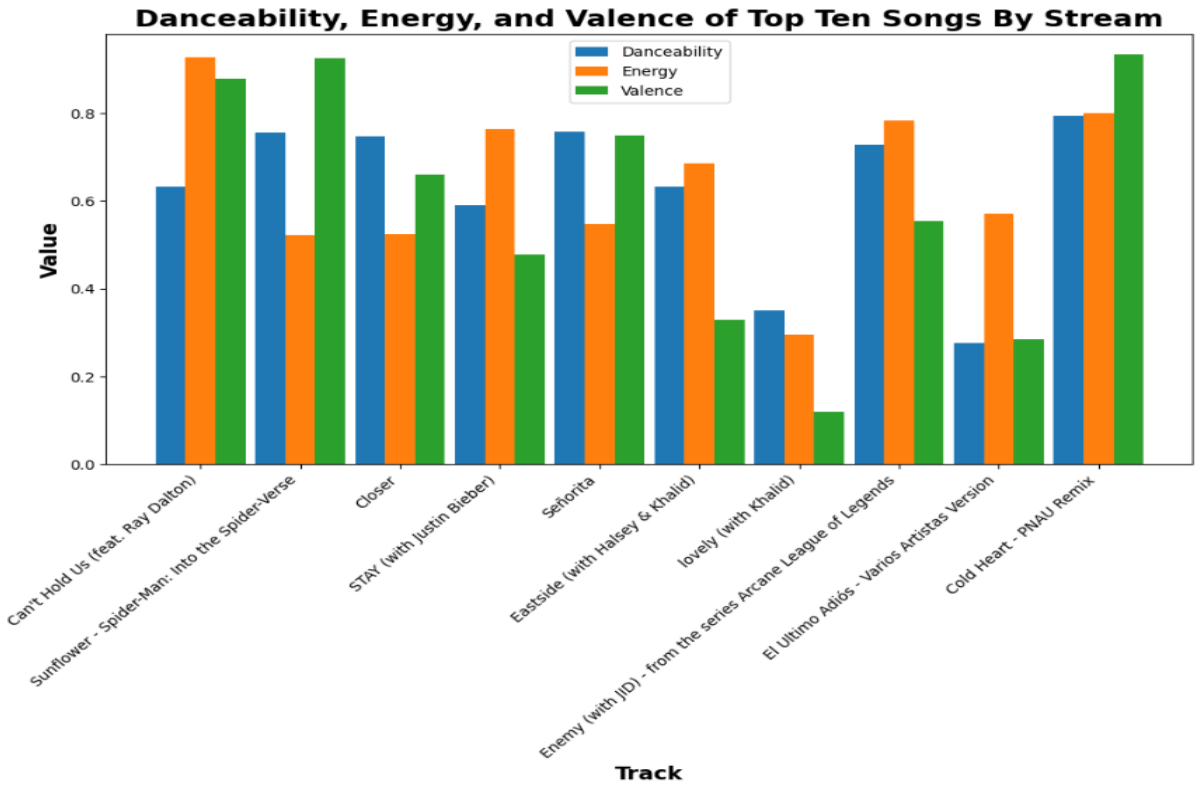
- Despacito holds 1st place by being the top song by most views on Youtube with 8 billion views.
- Blinding Lights holds 1st place by being the top song by most streams on Spotify with 3.4 billion streams, followed by Shape of You with 3.3 billion streams.
- Shape of You holds 2nd place by being one of the top ten songs by most views on Youtube and stream on Spotify.



b. Bar chart (3 vertical bars at 1 item): **=> *The chart says the following:***

- Create 3 vertical bars at 1 item: *bar(index), bar(index+0.3), bar(index+0.6)*
- Set title and labels: *set_title(), set_xlabel(), set_ylabel()*
- Set the x-tick labels: *set_xticks(index+0.3), xticks(rotation=45, ha='right')*
- Add a legend: *legend()*

- These features (Danceability, Energy, and Valence) are relevant to the ranking of songs.
- Songs with dance beats, high energy, and high valence are more popular.
- Can't Hold Us holds 1st place by being the top song by stream with the highest energy and valence.

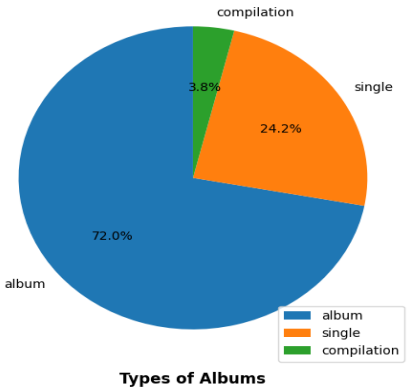


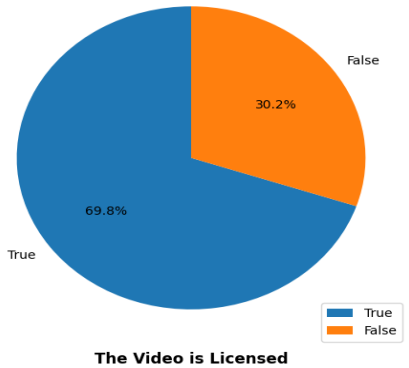
c. 3 Pie charts:

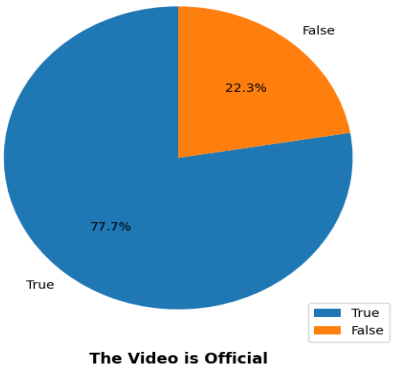
- Create 3 subplots: *subplots(1, 3)*
- Create pie chart: *pie()*
- Set titles: *set_xlabel()*
- Add a legend and set it to lower right: *legend(loc="lower right")*

=> The chart says the following:

- Albums are the most popular form of music release with 72.0%, followed by singles with 24.2%.
- Most of the videos, 69.8%, are licensed, representing licensed content.
- 77.7% of the videos found are the official videos of the songs.





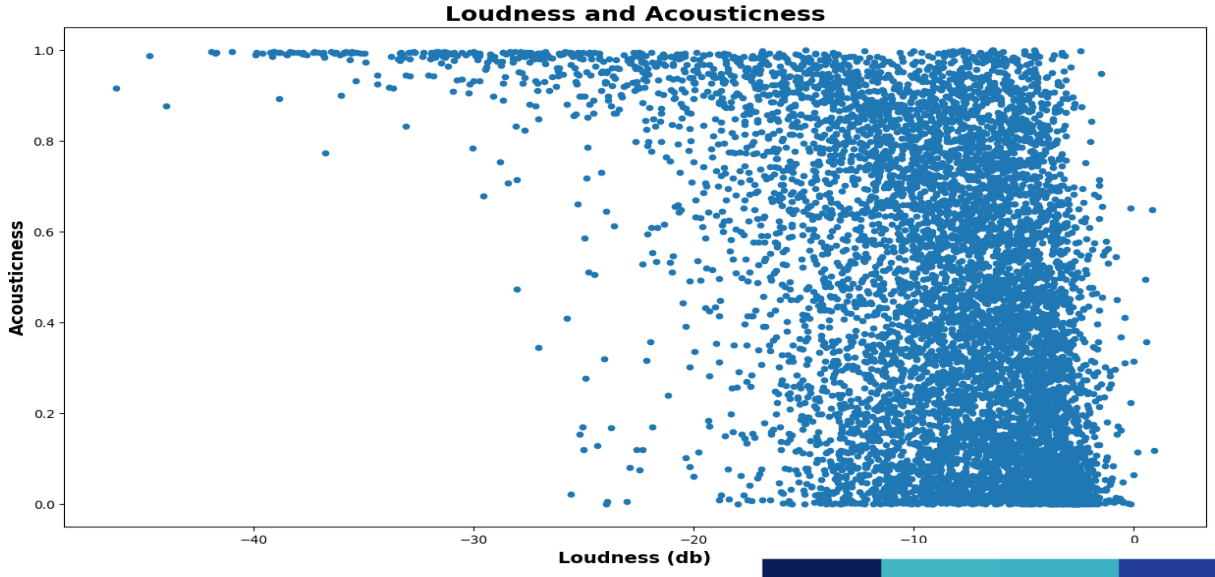


d. Scatter chart:

- Create scatter chart: *scatter()*
- Set title and labels: *title(), xlabel(), ylabel()*

=> *The chart says the following:*

- Loudness and Acousticness are 2 audio features that are commonly used to describe music.
- Songs with high loudness tend to have high acousticness as well since they are often more intense and dynamic.
- The graph seems to be scattered to the right as most of the data plots on the right. It appears to be skewed from the left.

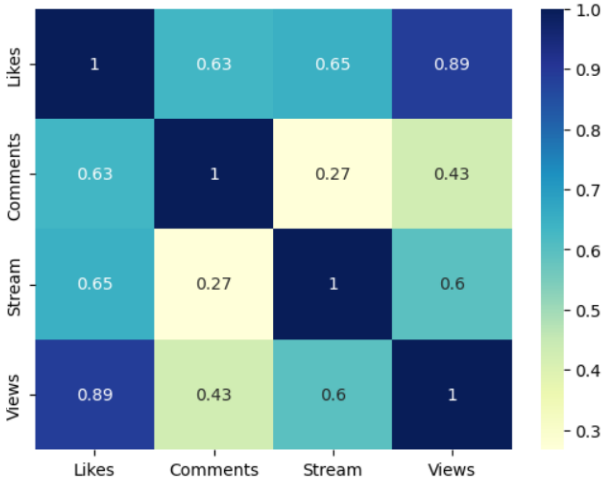


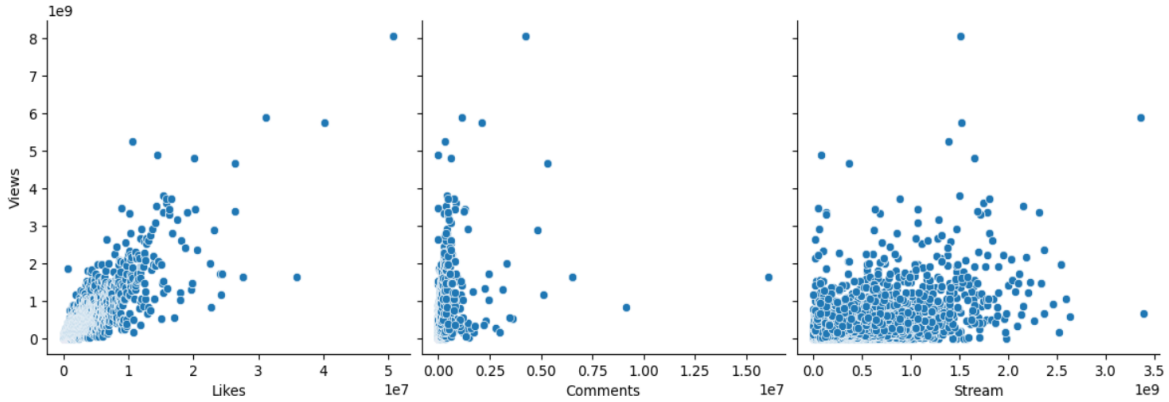
4. Bonus: Build a linear regression model:

a. Find feature variable, which is to find which column is the most correlated to Views column: *import seaborn library to use heatmap() and pairplot() functions.*

According to the graphs, the Likes column seems most correlated to Views column (0.89).

=> *The Likes column is used as a feature variable.*





b. Create training set and validation set:

from sklearn.model_selection import train_test_split to use train_test_split() function.

```
X_train:
8525      86697.0
10278    144692.0
1885       2579.0
6565      46235.0
4293      16086.0
           ...
16304    749929.0
79          19.0
12119    243916.0
14147    443507.0
5640      31153.0
Name: Likes, Length: 14080, dtype: float64

y_train:
8525      21746563.0
10278      9563176.0
1885        553760.0
6565       4028896.0
4293        679667.0
            ...
16304    180138758.0
79           5292.0
12119     23329150.0
14147     34360662.0
5640       1825136.0
Name: Views, Length: 14080, dtype: float64
```

c. Find and visualize the regression line:

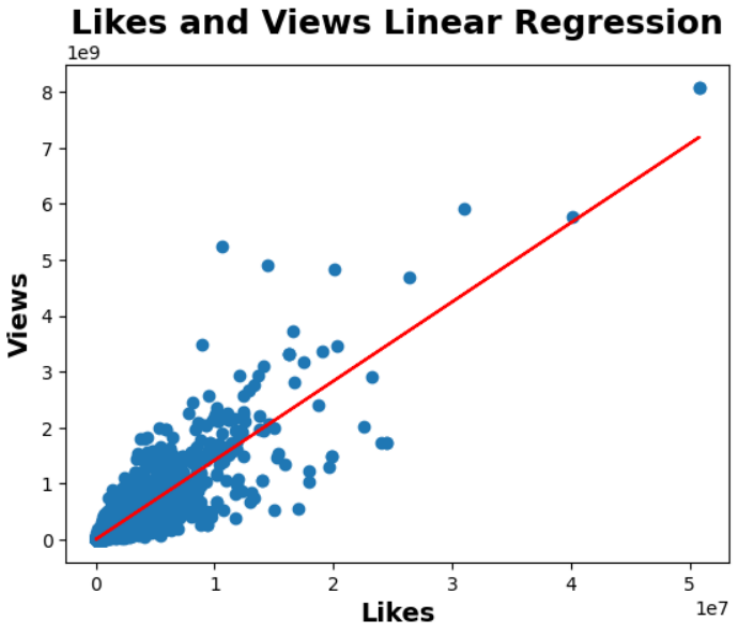
- *From sklearn.linear_model import LinearRegression to use LinearRegression() and fit() functions.*
- *Import matplotlib library to create scatter chart.*

Intercept: 1822708.6585386395

Slope: [141.48864498]

=> The regression line:

Views = 1822709 + 141.4886 * Like



d. Find MSE of the model: *from sklearn.linear_model import LinearRegression to use predict() function, from sklearn.metrics import mean_squared_error to use mean_squared_error().*

MSE of training set is: 15565279074256570.00000

MSE of valid set is: 19125417327917868.00000

References

Rastelli, S., Sallustio, M., & Guarisco, M. (2023, March 20). *Spotify and YouTube*. Kaggle. Retrieved April 8, 2023, from https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube