

# Spotify® & YouTube **Analysis**

Devi Somalinga Bhuvanesh  
Gururup Kaur Gurbir Singh Soi  
Vy Hoang Mai Duong

# Report Summary

## Exploratory Analysis & Data preparation

Combine and delete variables  
Duplicated, Missing values, Outliers &  
New Variable Column

## Preliminary Analysis

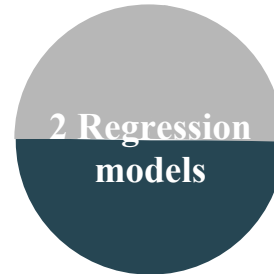
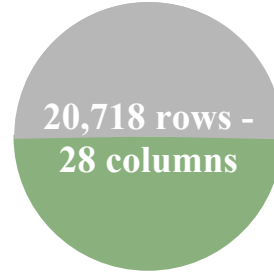
Features of Songs, Artists, Albums and  
Channels

## Analysis & Modelling

Logistic Regression & Confusion matrix  
Linear Regression & Correlation Matrix  
t-test of dependence samples

## Conclusion and Recommendations

Proposals based on Analysis and  
Conclusion



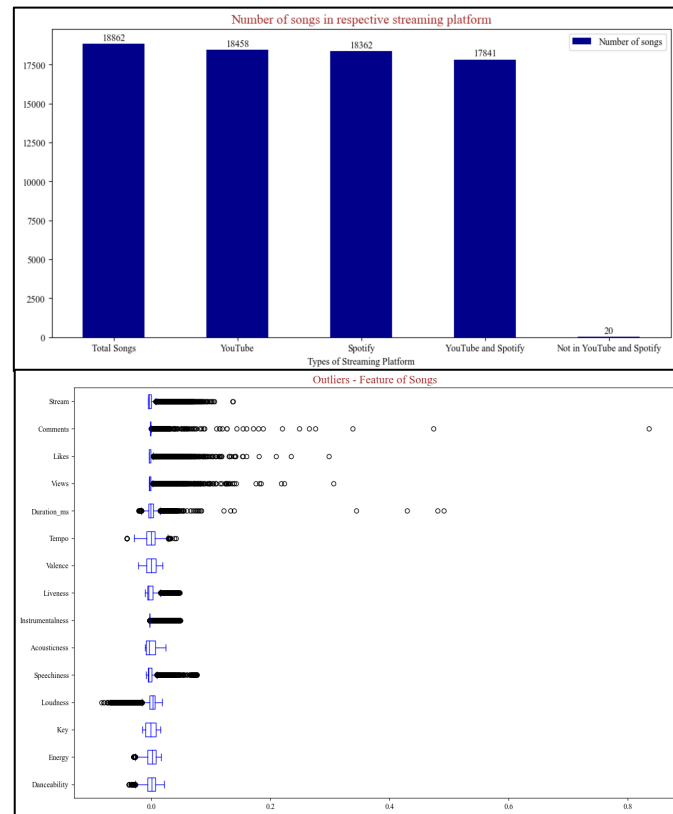
# Exploratory Analysis & Data preparation

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	20718 non-null	int64
1	Artist	20718 non-null	object
2	Url_spotify	20718 non-null	object
3	Track	20718 non-null	object
4	Album	20718 non-null	object
5	Album_type	20718 non-null	object
6	Uri	20718 non-null	object
7	Danceability	20716 non-null	float64
8	Energy	20716 non-null	float64
9	Key	20716 non-null	float64
10	Loudness	20716 non-null	float64
11	Speechiness	20716 non-null	float64
12	Acousticness	20716 non-null	float64
13	Instrumentalness	20716 non-null	float64
14	Liveness	20716 non-null	float64
15	Valence	20716 non-null	float64
16	Tempo	20716 non-null	float64
17	Duration_ms	20716 non-null	float64
18	Url_youtube	20248 non-null	object
19	Title	20248 non-null	object
20	Channel	20248 non-null	object
21	Views	20248 non-null	float64
22	Likes	20177 non-null	float64
23	Comments	20149 non-null	float64
24	Description	19842 non-null	object
25	Licensed	20248 non-null	object
26	official_video	20248 non-null	object
27	Stream	20142 non-null	float64

Rename & Remove  
columns

Identify duplication  
and merge values

Define Outliers




# Exploratory Analysis & Data preparation

**Add column**  
Likes-Views ratio  
for YouTube  
videos



To **indicate** the popularity of a songs with the positive reception of that song

**Divide** datasets  
into 2 sets  
according analysis  
purposes



Multiple artists performed the same song resulting in duplication of rows with the same song and its features. => **Artists set**

The same song/track was performed by different artists but had different features => **Songs set**

# Glimpse of the Cleaned Dataset

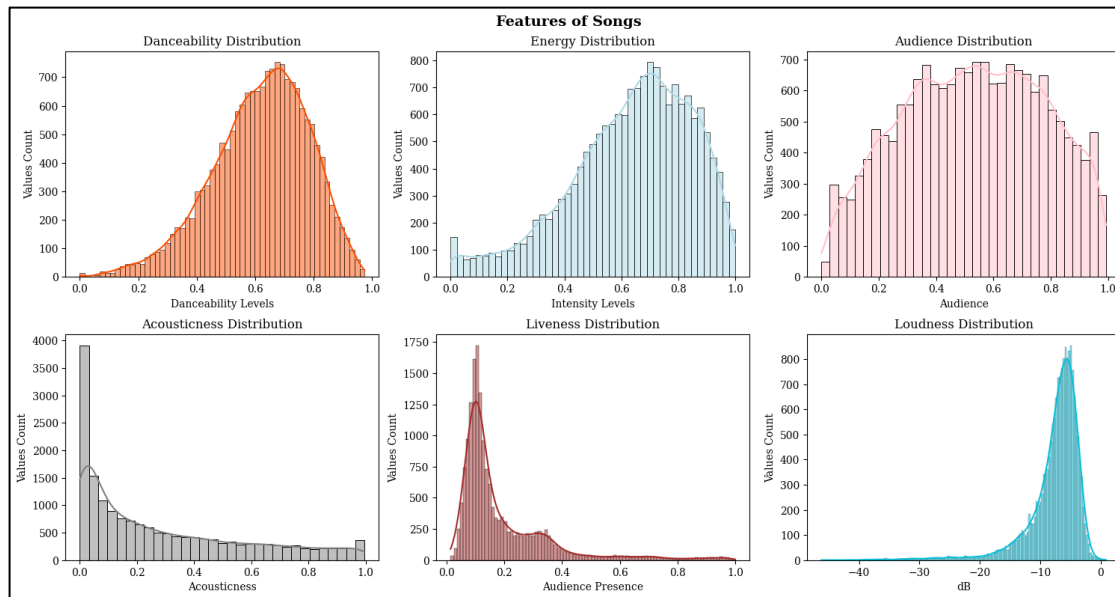
Int64Index: 18862 entries, 0 to 20717

Data columns (total 26 columns):

#	Column	Non-Null Count	Dtype
0	Artist	18862 non-null	object
1	Track	18862 non-null	object
2	Album	18862 non-null	object
3	Album_type	18862 non-null	object
4	Uri	18862 non-null	object
5	Danceability	18860 non-null	float64
6	Energy	18860 non-null	float64
7	Key	18860 non-null	float64
8	Loudness	18860 non-null	float64
9	Speechiness	18860 non-null	float64
10	Acousticness	18860 non-null	float64
11	Instrumentalness	18860 non-null	float64
12	Liveness	18860 non-null	float64
13	Valence	18860 non-null	float64
14	Tempo	18860 non-null	float64
15	Duration_ms	18860 non-null	float64
16	Title	18458 non-null	object
17	Channel	18458 non-null	object
18	Views	18458 non-null	float64
19	Likes	18389 non-null	float64
20	Comments	18362 non-null	float64
21	Description	18074 non-null	object
22	Licensed	18458 non-null	object
23	official_video	18458 non-null	object
24	Stream	18312 non-null	float64
25	Likes_to_views_ratio	18388 non-null	float64

dtypes: float64(16), object(10)

memory usage: 3.9+ MB



# Preliminary Analysis

01

Artists on  
Spotify &  
YouTube



02

Channels &  
Albums on  
Spotify &  
YouTube



03

Songs on  
Spotify &  
YouTube



- **Spotify** Streams peak at ~3.4 billion counts, average stream counts of 18,312 tracks is 131,664,214.

- **YouTube** Views peaks at 8.9+ billion counts, averaging at 89,044,026 which is less than Spotify average stream counts.  
+ Views and Comments are indicators of **YouTube**

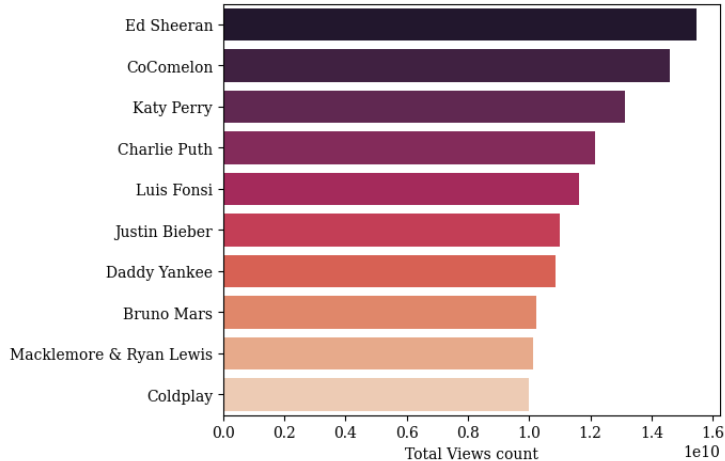
	Stream	Likes	Views	Comments
count	1.831200e+04	1.838900e+04	1.845800e+04	1.836200e+04
mean	1.316642e+08	6.229439e+05	8.904403e+07	2.652553e+04
std	2.365431e+08	1.681334e+06	2.607889e+08	1.922104e+05
min	6.574000e+03	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.704808e+07	2.068000e+04	1.754198e+06	4.930000e+02
50%	4.864910e+07	1.169800e+05	1.368734e+07	3.125500e+03
75%	1.346372e+08	4.846190e+05	6.657492e+07	1.362050e+04
max	3.386520e+09	5.078863e+07	8.079647e+09	1.608314e+07

# Preliminary Analysis



## Top Artists on Spotify & YouTube

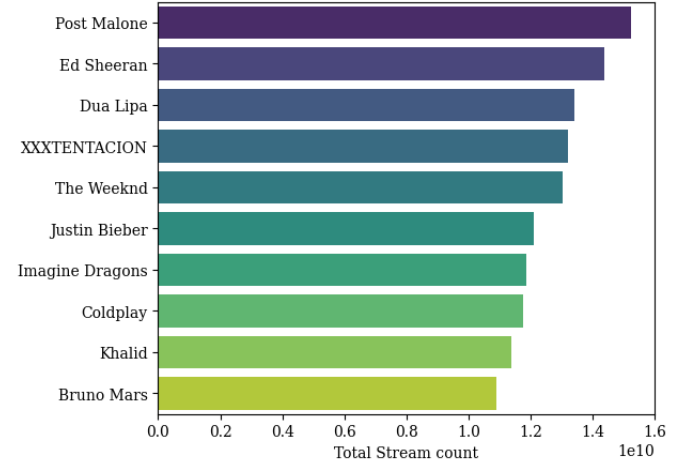
Top 10 Best Performed Artist Based on YouTube Views



Best Performed Artist on **YouTube** is Ed Sheeran with an accumulated 15.5 billion views count for all his Songs

In second place is CoComelon, a channel dedicated to children's content, with 14,6 billion views count

Top 10 Most Streamed Artist on Spotify Stream



Post Malone is the most successful artist on **Spotify** with ~15.3 billion streams in total.

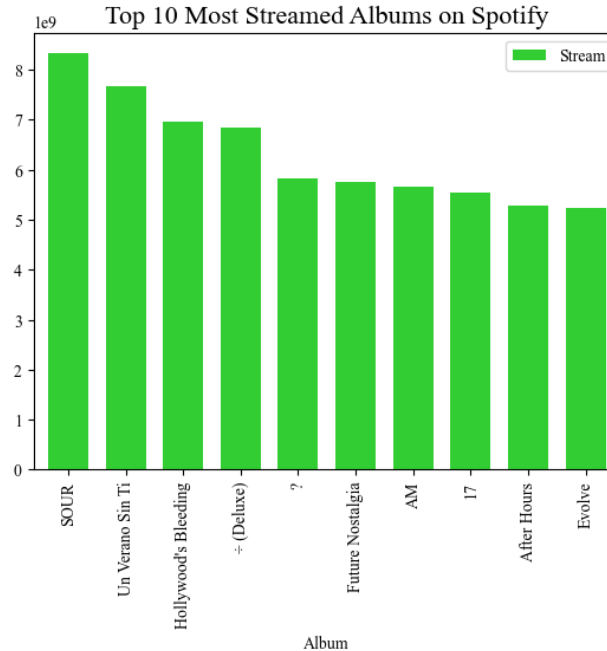
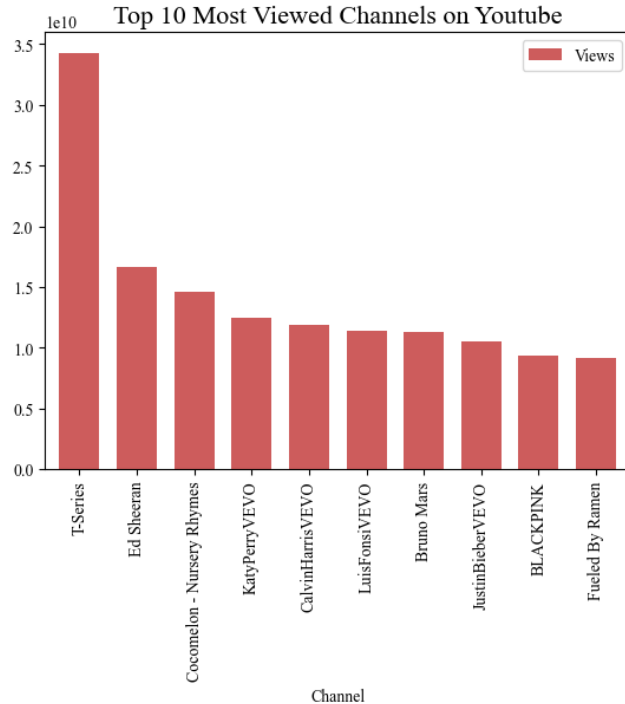
In second, is Ed Sheeran with 14.4 billion streams

Bruno Mars  
Coldplay  
Ed Sheeran  
Justin Bieber

# Preliminary Analysis



## Top Channels & Albums on Spotify & YouTube



	Channel	Title
4807	T-Series	173
4667	SonyMusicIndiaVEVO	66
4668	SonyMusicSouthVEVO	62
4120	RHINO	62
5608	Zee Music Company	48
4418	SMTOWN	47
2027	HYBE LABELS	43
1339	DisneyMusicVEVO	43
426	Atlantic Records	42
4662	Sony Music India	34

Entertainment companies are taking advantages of YouTube platform to reach more audience

Channel with most uploads are all Entertainment Music companies.

(\*At time of data collected)

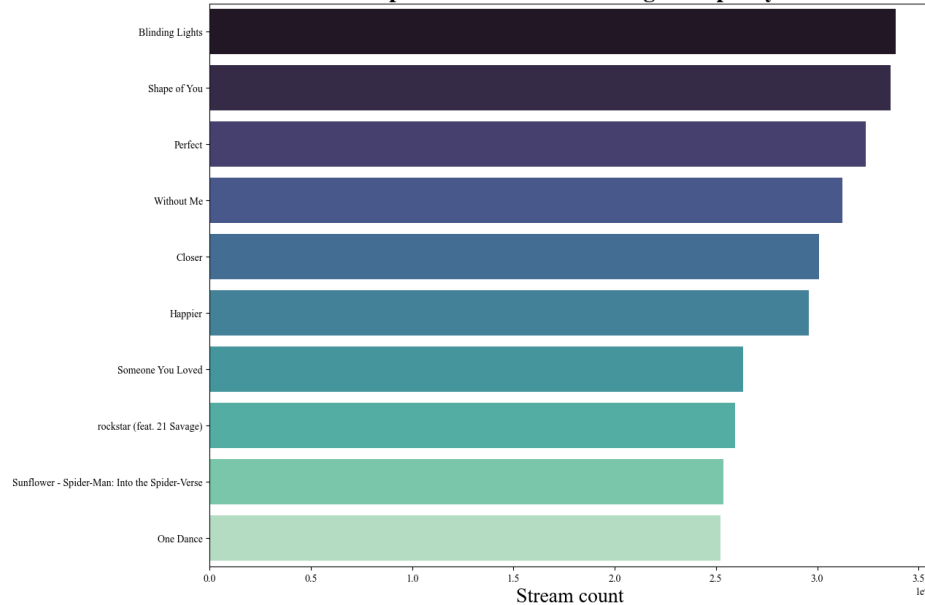


# Preliminary Analysis

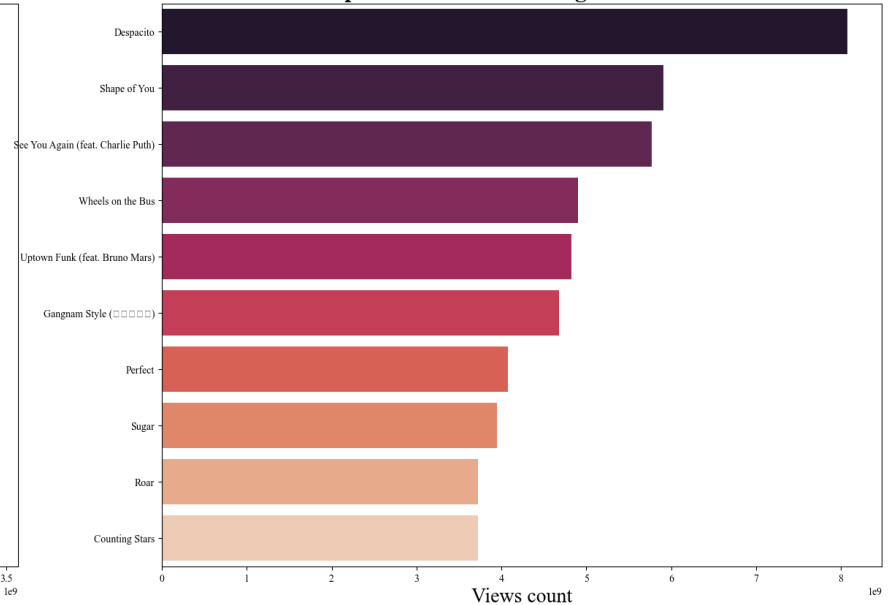


## Top Songs on Spotify & YouTube

Top 10 Most Streamed Songs on Spotify



Top 10 Most Viewed Songs on YouTube



2 songs that make it to the Top 10 played in 2 platforms, both from Ed Sheeran: {'Perfect', 'Shape of You'}

# Preliminary Analysis

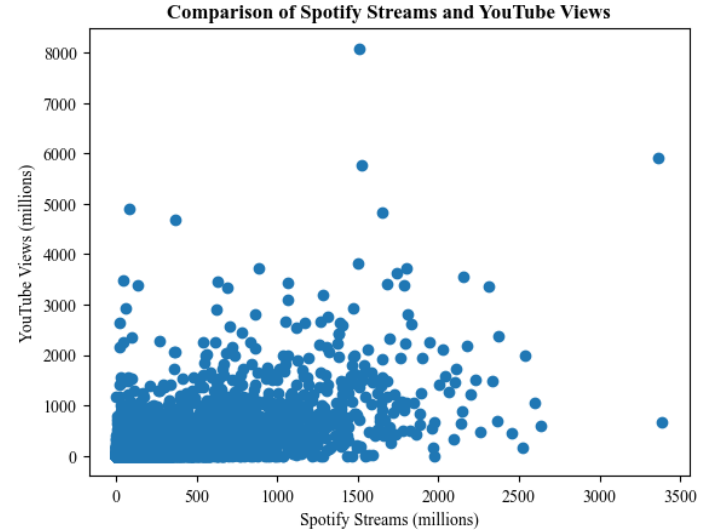
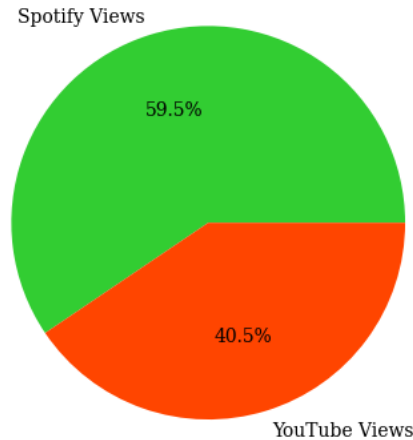


## Total YouTube Views and Spotify Stream

Generally, Spotify total views is higher than YouTube total views in this dataset

Spotify views and YouTube views has a moderately positive correlation of 0.60.

### Total YouTube Views vs Spotify Stream



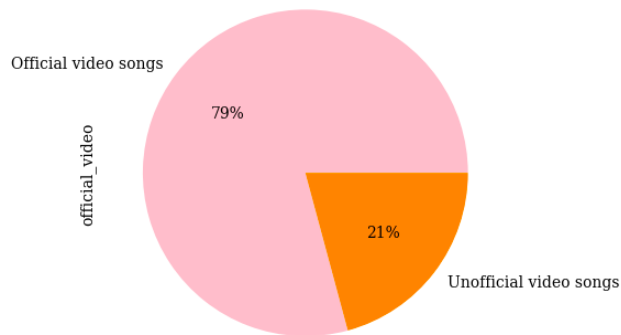
# Preliminary Analysis

## Songs & Albums on Spotify & YouTube



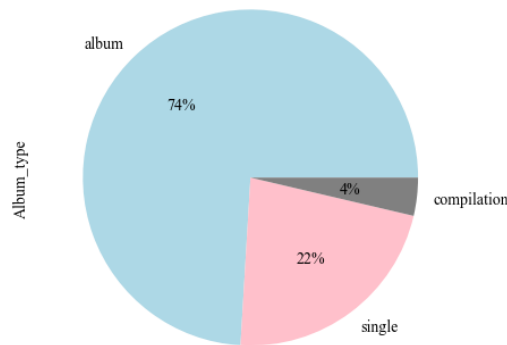
Most videos on YouTube are official videos (79%)  
21% is unofficial uploads

Proportion of Official vs. Unofficial Songs



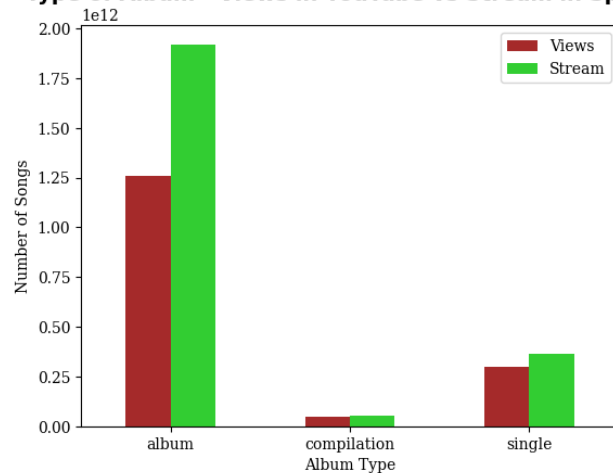
74% songs belong to albums  
22% songs are singles  
4% are compilation

Type of Songs



Spotify has more streams than views on YouTube for each album type

Type of Album - Views in YouTube vs Stream in Spotify



# Preliminary Analysis



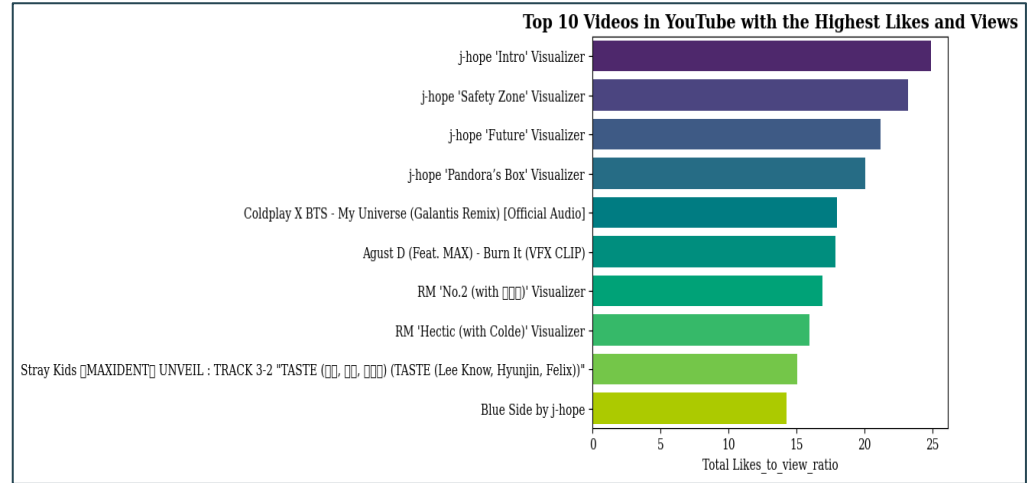
Gauge of how much an audience is watching and appreciating a video.

A glimpse of trendy music consumption can be seen:

- This period favor B.T.S boyband as 8 of the top videos had their members appearance (RM, J-Hope or all group)
- The top 4 Likes/Views videos were all from J-hope, a B.T.S member, ranging from 24.92% to 20.05%, followed is Coldplay X BTS.

'Likes' and 'Views' have a significant positive linear association (coefficient of 0.886). This indicates that a rise in views is related to an increase in likes, and vice versa.

## Top Likes-Views ratio videos



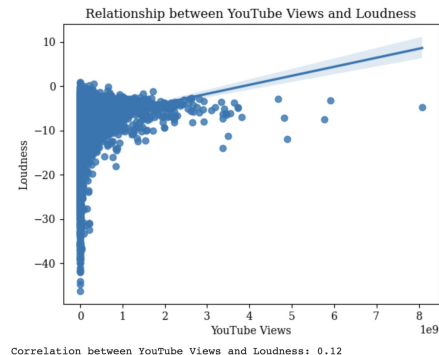
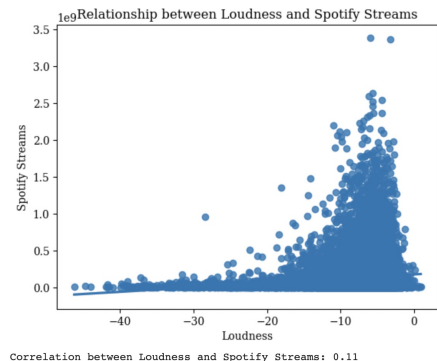
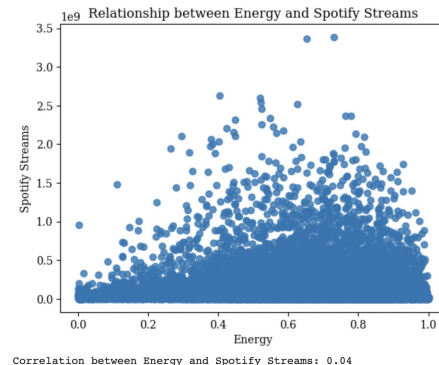
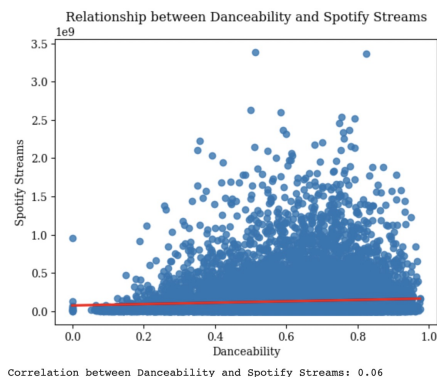
```
print('Correlation coefficient:', correlation_coef)
```

```
Correlation coefficient: 0.885976986339338
```

# Preliminary Analysis

## Linear Relationship

- Danceability and Spotify streams have a 0.06 correlation coefficient, which suggests a marginally favorable association
- A very slight positive link between energy and Spotify streams—with a correlation coefficient of 0.04—can be seen.
- Spotify streams and loudness have a weakly positive correlation (0.11),
- The correlation coefficient between YouTube views and loudness, which is 0.12, also shows a weakly positive association between the two variables.



# Inferential Analysis & Prediction Modelling

01

## Logistic Regression

Analyze the effect of licensing on views:

- Number of views: *predictor*
- Whether a song is licensed or not: *response* variable.

02

## Linear Regression

Relationship between the song characteristics and their popularity

Prediction modelling the song's popularity based on songs characteristic

03

## Paired t-test

To infer if there is a significant difference between Spotify Streams and YouTube Views from this dataset sample

# Inferential Analysis



## Paired test: YouTube Views and Spotify Stream

### The question:

Artists and Entertainment companies might be interested in if there is a significant difference between Spotify Streams and YouTube Views.

The method: t-test of dependence samples (Given same artists and songs, different platform)

### Conducting the test:

#### **Hypothesis stating:**

- *Null Hypothesis* is assuming the Views and Streams of the artists are equal:  $H_0: M_1 = M_2$
- The *alternative hypothesis* would be that the 2 means are not equal as stated in the null hypothesis:  $H_1: M_1 \neq M_2$

```
TtestResult(statistic=-15.363285756565345, pvalue=1.7806596212483008e-51, df=3251)
```

### Test Result:

The test statistic and p-value is -15.363285756565345 and 1.781, which are very large t-statistics (assume  $\alpha=0.01$ )

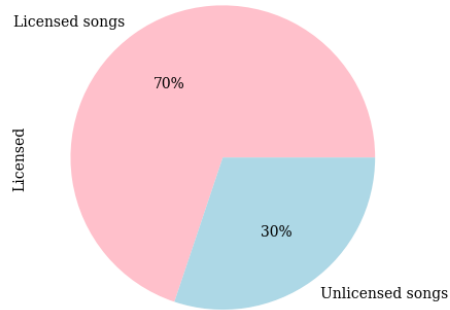
The Conclusion: we will reject the Null hypothesis and conclude that there is a significant difference between the reception Artists receive on YouTube and Spotify

# Prediction Modelling

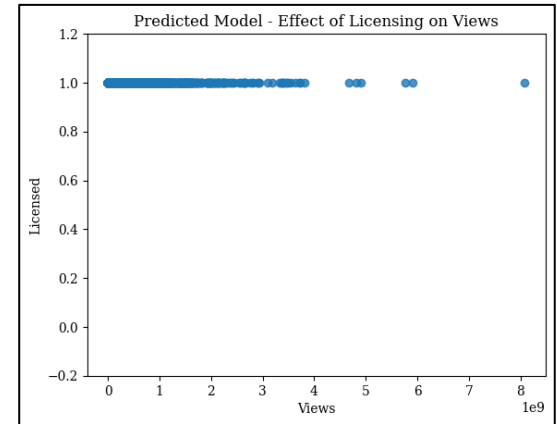
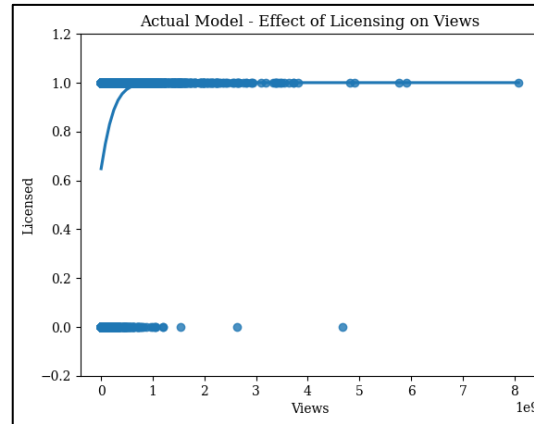
01

## Logistic Regression

Proportion of Licensed vs. Unlicensed Songs



The dataset has a greater proportion of *licensed* songs compared to *unlicensed* ones



**Logistic regression** model was used to analyze the effect of licensing on views (category variable):

- Number of views: predictor variable
- Whether a song is licensed or not: response variable



# Prediction Modelling

01

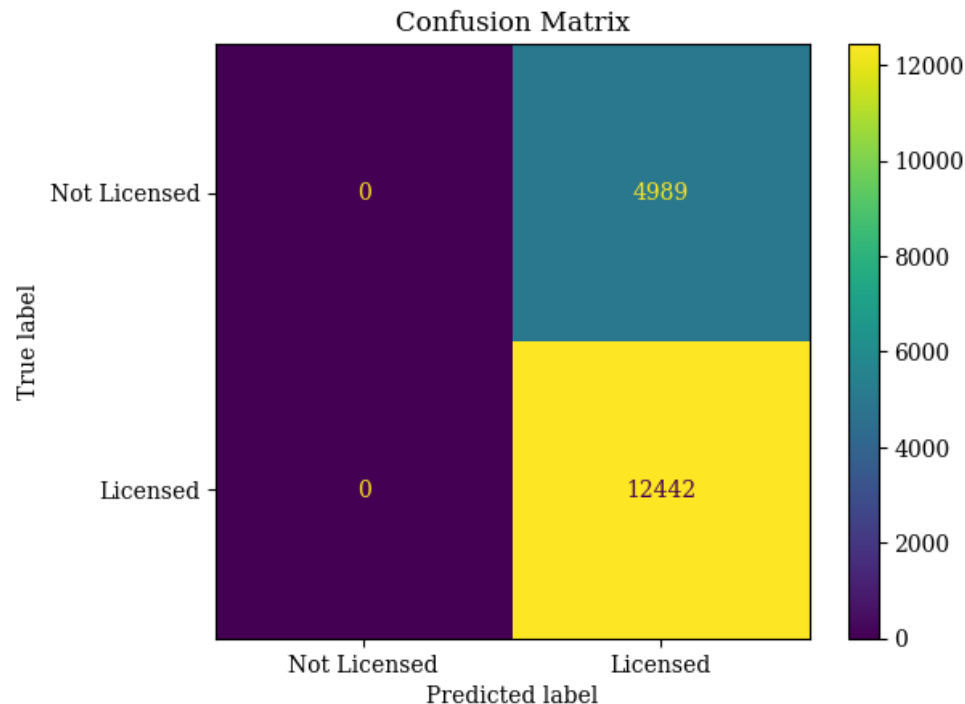
## Logistic Regression

	precision	recall	f1-score	support
0	0.00	0.00	0.00	4989
1	0.71	1.00	0.83	12442
accuracy			0.71	17431
macro avg	0.36	0.50	0.42	17431
weighted avg	0.51	0.71	0.59	17431

**Precision:** from all the True-Positive predicted, 71% was actually licensed.

**Recall:** in all the licensed songs, we predicted correctly 100%

**Accuracy:** total accurate predictions was 71%



# Prediction Modelling

02

## Linear Regression

### The question:

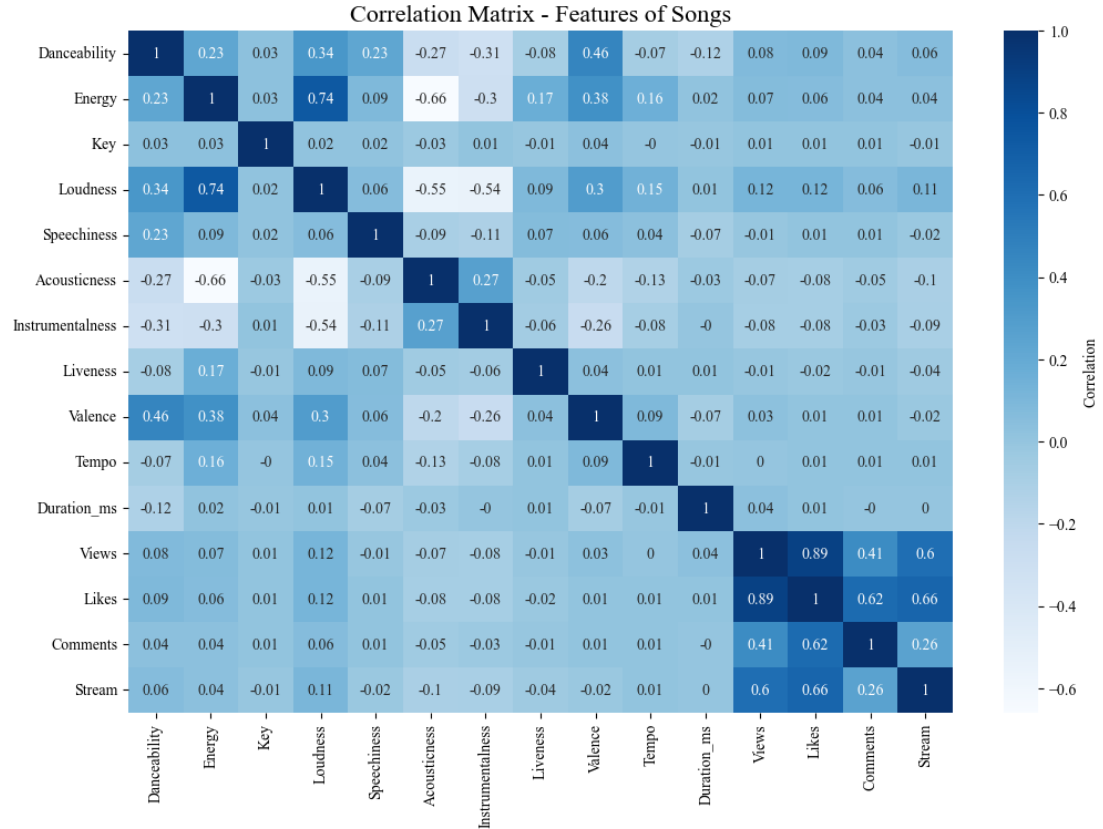
Uncovering if the more catchy, more danceable, or faster, etc. a song is, the easier it is to be well-received by population

### The method: Multi Linear Regression

From correlation matrix, we attempt to modeling YouTube Views prediction and Spotify Streams based on songs features

*(Note: exclude attributes with less than 0.05 correlation with the dependent variables)*

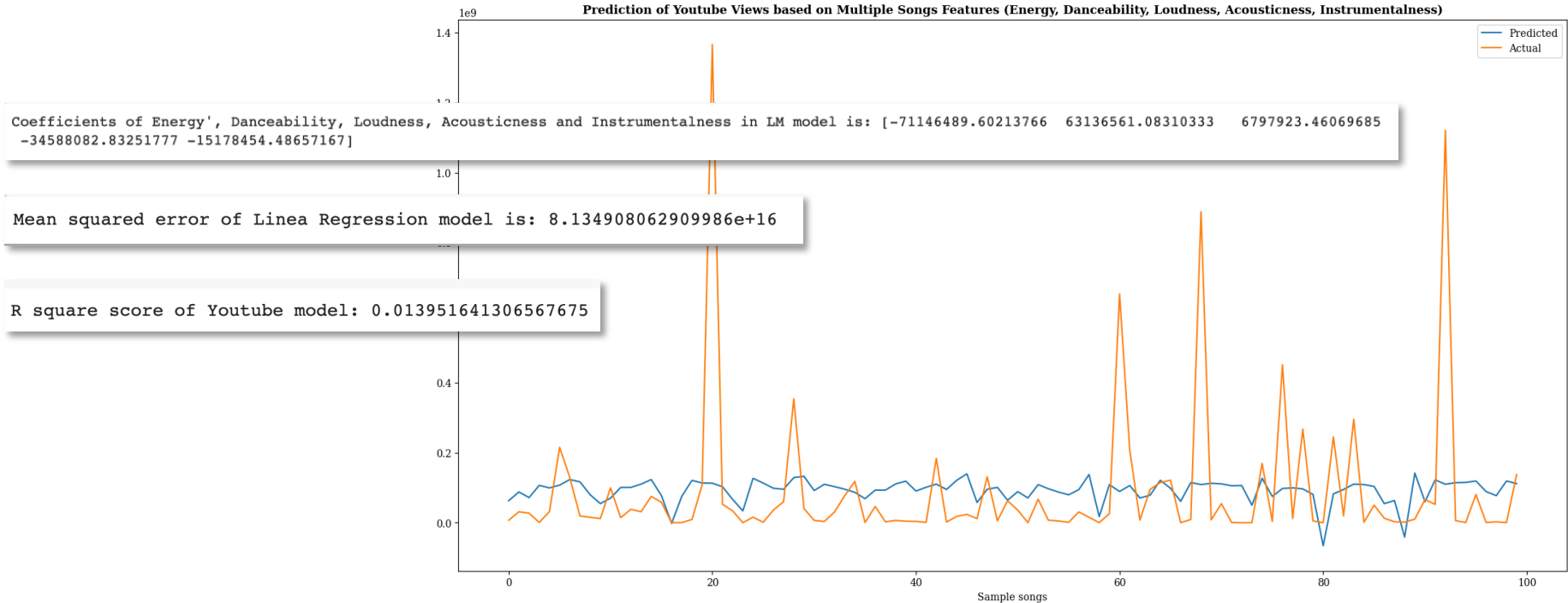
Model was built on train data (70%) and concluded on test data (30%)



# Prediction Modelling

02

## Linear Regression



# Prediction Modelling

02

## Linear Regression

Prediction of Spotify Stream based on Multiple Songs Features (Danceability, Loudness, Acousticness, Instrumentalness)



Coefficients of Danceability, Loudness, Acousticness and Instrumentalness in LM model is: [ 30264107.80828344 2524502.73699581 -52850144.71244186 -46538934.51105024]

Mean squared error of Linea Regression model is:  $6.092714296666391 \times 10^{16}$

R square score of Spotify model: 0.013408349636507055

# Prediction Modelling

02

Linear Regression

Reading the statistic of the two models, we can *conclude* that:

- The prediction ability of YouTube model is around 1.4% and Spotify model is 1.3%. Both are low
- The mean squared error values are both very high, meaning a high error values of predicted values compared to observed
- The graphs also demonstrate that even though the models can make some predictions matching the test data, the accuracy is very low.

# Conclusions



Songs on Spotify had more views (59.5%) than on YouTube.

The presence of an *official video significantly influenced* a song's popularity.

Energy and loudness showed weak or no correlation with Spotify streams and YouTube views

Ed Sheeran was the top Artist on YouTube, followed by CoComelon, a children's content channel.

On Spotify, Post Malone was the most successful Artist while Ed Sheeran ranked second.

These highlight *differences in consumption* trend of 2 platforms

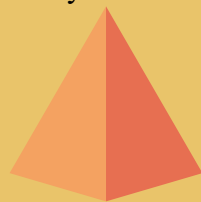


Challenges in accurately predicting licensed & unlicensed songs due to an *imbalanced dataset*

t-test confirmed a significant *difference in audience engagement* and consumption between 2 platforms.

Linear regression models built on song features (Danceability, Energy, and Loudness) exhibited *low prediction*

This implies that factors beyond these characteristics contribute more significantly to a song's popularity.



# Recommendations

## Official content

Official videos significantly influenced a song's popularity

## Cross platform

Maximize their reach and exposure due to differences in consumption patterns

## Licenses

Licensing will allow for wider distribution and monetization

## Continuously Analyze & Adapt

To effectively navigate the changing landscape and maximize chances of success

## Diverse factors for popularity

Marketing, audience targeting, and promotion strategies, to enhance success

## Engagement metrics

Correlations between views and likes indicate the importance of viewer engagement

## International Markets

Exploring international markets by capitalizing on the YouTube diverse global reach

## Balanced Dataset

Inclusive to enhance prediction model capacity & Avoid bias

# References

*Spotify and Youtube*. (2023b, March 20). Kaggle. Retrieved on April 24, from <https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube>

GeeksforGeeks. (2022). How to Find Drop duplicate columns in a Pandas DataFrame. *GeeksforGeeks*. Retrieved on April 24, from <https://www.geeksforgeeks.org/how-to-find-drop-duplicate-columns-in-a-pandas-dataframe/>

Zach. (2021). How to Drop Duplicate Columns in Pandas (With Examples). *Statology*. Retrieved on April 24, 2023 from <https://www.statology.org/pandas-drop-duplicate-columns/>

*Merge, join, concatenate and compare — pandas 2.0.1 documentation*. (n.d.). Retrieved on April 24, 2023 from [https://pandas.pydata.org/docs/user\\_guide/merging.html](https://pandas.pydata.org/docs/user_guide/merging.html)

Kleppen, E. (2023, April 5). How To Find Outliers in Data Using Python (and How To Handle Them). *CareerFoundry*. Retrieved on April 24, 2023 from <https://careerfoundry.com/en/blog/data-analytics/how-to-find-outliers/>

Stack Overflow. *Compare two lists in python and return matches*. (n.d.). Retrieved on April 24, 2023 from <https://stackoverflow.com/questions/1388818/how-can-i-compare-two-lists-in-python-and-return-matches>

Northeastern University, ALY6015 - Intermediate Analytics (2023). Retrieved on April 24, 2023.

Python, R. (2022). Logistic Regression in Python. *realpython.com*. [https://realpython.com/logistic-regression-python/#:~:text=The%20logistic%20regression%20function%20%F0%9D%91%9D\(%F0%9D%90%B1\)%20is%20the%20sigmoid%20function,that%20the%20output%20is%200.](https://realpython.com/logistic-regression-python/#:~:text=The%20logistic%20regression%20function%20%F0%9D%91%9D(%F0%9D%90%B1)%20is%20the%20sigmoid%20function,that%20the%20output%20is%200.)

GeeksforGeeks. (2022b). Create a correlation Matrix using Python. *GeeksforGeeks*. <https://www.geeksforgeeks.org/create-a-correlation-matrix-using-python/>

GeeksforGeeks. (2023a). Linear Regression Python Implementation. *GeeksforGeeks*. <https://www.geeksforgeeks.org/linear-regression-python-implementation/>

Narkhede, S. (2021, June 15). Understanding Confusion Matrix - Towards Data Science. *Medium*. <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

PyCoach. (2022b, January 4). A Simple Guide to Linear Regression using Python - Towards Data Science. *Medium*. <https://towardsdatascience.com/a-simple-guide-to-linear-regression-using-python-7050e8c751c1>

*sklearn.linear\_model.LogisticRegression*. (n.d.). Scikit-learn. [https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

GeeksforGeeks. (2021a). Bar Plot in Matplotlib. *GeeksforGeeks*. <https://www.geeksforgeeks.org/bar-plot-in-matplotlib/>

*Matplotlib Pie Charts*. (n.d.). [https://www.w3schools.com/python/matplotlib\\_pie\\_charts.asp](https://www.w3schools.com/python/matplotlib_pie_charts.asp)

*Scatter plot — Matplotlib 3.7.1 documentation*. (n.d.). [https://matplotlib.org/stable/gallery/shapes\\_and\\_collections/scatter.html](https://matplotlib.org/stable/gallery/shapes_and_collections/scatter.html)



