Youtube Comments Sentiment Analysis

Khushi Masarani
Asmitha Gopal
Norberto Limon

Agenda:

01 - Introduction

02 - Data Preprocessing

03 - Sentiment Analysis

04 - Word Frequency

05 - Conclusion

Introduction

Discover the power of sentiment analysis in understanding the opinions and emotions of viewers on YouTube. Analyze data to gain insights.

Research Question

"How can sentiment analysis help to inform YouTube content creators and marketers?"

Objective

- The objective of our research was to learn about sentiment analysis using VADER.
- Our source data was a sample dataset of YouTube comments and interactions.
- Using this dataset we classify comments by sentiment (neg, neu, pos) and identify the channels with the highest positivity rating in its comment pool.

YouTube Datasets

- It consists of two datasets-
  - Youtube vidoes
  - Youtube comments
- The videos dataset has 8 variables and 8,000 instances.
- The comments dataset has 4 variables and 120,000 instances.

```
US_comments = pd.read_csv('/content/drive/MyDrive/datasets/UScomments.csv', error_bad_lines=False)
US_comments.head()
```

```
Skipping line 41589: expected 4 fields, saw 11
Skipping line 51628: expected 4 fields, saw 7
Skipping line 114465: expected 4 fields, saw 5
```

	video_id	comment_text	likes	replies
0	XpVt6Z1Gjjo	Logan Paul it's yo big day !!!!!!	4	0
1	XpVt6Z1Gjjo	I've been following you from the start of your...	3	0
2	XpVt6Z1Gjjo	Say hi to Kong and maverick for me	3	0
3	XpVt6Z1Gjjo	MY FAN . attendance	3	0
4	XpVt6Z1Gjjo	trending 😉	3	0

```
[ ] US_videos = pd.read_csv('/content/drive/MyDrive/datasets/USvideos.csv', error_bad_lines=False)
```

```
Skipping line 2401: expected 11 fields, saw 21
Skipping line 2800: expected 11 fields, saw 21
```

YT Videos Data Description

- Youtube Videos Data consists of:
  - **Video ID:** A unique identifier for each video on YouTube.
  - **Title:** The title of the video.
  - **Category:** The category or genre to which the video belongs.
  - **Tags:** A list of keywords or tags associated with the video.
  - **View Count:** The number of times the video has been viewed.
  - **Like Count:** The number of likes received for the video.
  - **Dislike Count:** The number of dislikes received for the video.
  - **Comment Count:** The number of comments posted on the video.

```
US_videos.head()
```

	video_id	title	channel_title	category_id	tags	views	likes	dislikes	comment_total	thur
0	XpVt6Z1Gjjo	1 YEAR OF VLOGGING -- HOW LOGAN PAUL CHANGED Y...	Logan Paul Vlogs	24	logan paul vlog|logan paul|logan|paul|olympics...	4394029	320053	5931	46245	https://i.ytimg.com/vi/XpVt6Z1G
1	K4wEI5zhHB0	iPhone X — Introducing iPhone X — Apple	Apple	28	Apple|iPhone 10|iPhone Ten|iPhone|Portrait Lig...	7860119	185853	26679	0	https://i.ytimg.com/vi/K4wEI5zhHE
2	cLdxuaxaQwc	My Response	PewDiePie	22	[none]	5845909	576597	39774	170708	https://i.ytimg.com/vi/cLdxuaxaQv
3	WYYvHb03Eog	Apple iPhone X first look	The Verge	28	apple iphone x hands on|Apple iPhone X|iPhone ...	2642103	24975	4542	12829	https://i.ytimg.com/vi/WYYvHb03Ec
4	sjlHnJvXdQs	iPhone X (parody)	jacksfilms	23	jacksfilms|parody|parodies|iphone|iphone x|iph...	1168130	96666	568	6666	https://i.ytimg.com/vi/sjlHnJvXdC

YT Comments Data Description

- Our Youtube Comments Data consists of:
  - **video_id:** Identifier for the video being commented on
  - **comment_text:** Text content of the comment
  - **likes:** Number of likes received for the comment
  - **replies:** Number of replies received for the comment


```
US_comments.head()
```

	video_id	comment_text	likes	replies
0	XpVt6Z1Gjjo	Logan Paul it's yo big day !!!!!!	4	0
1	XpVt6Z1Gjjo	I've been following you from the start of your...	3	0
2	XpVt6Z1Gjjo	Say hi to Kong and maverick for me	3	0
3	XpVt6Z1Gjjo	MY FAN . attendance	3	0
4	XpVt6Z1Gjjo	trending 😉	3	0

Libraries

Importing The Libaries

```
[ ]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import re
     import csv

     %matplotlib inline
     import warnings
     warnings.filterwarnings("ignore")
```

```
[ ]  pd.set_option('display.max_columns',None)
```

Reading the datasets

```
US_comments = pd.read_csv('/content/drive/MyDrive/datasets/UScomments.csv', error_bad_lines=False)
US_comments.head()
```

```
Skipping line 41589: expected 4 fields, saw 11
Skipping line 51628: expected 4 fields, saw 7
Skipping line 114465: expected 4 fields, saw 5
```

	video_id	comment_text	likes	replies
0	XpVt6Z1Gjjo	Logan Paul it's yo big day !!!!!!	4	0
1	XpVt6Z1Gjjo	I've been following you from the start of your...	3	0
2	XpVt6Z1Gjjo	Say hi to Kong and maverick for me	3	0
3	XpVt6Z1Gjjo	MY FAN . attendance	3	0
4	XpVt6Z1Gjjo	trending 😉	3	0

```
[ ]  US_videos = pd.read_csv('/content/drive/MyDrive/datasets/USvideos.csv', error_bad_lines=False)
```

```
Skipping line 2401: expected 11 fields, saw 21
Skipping line 2800: expected 11 fields, saw 21
Skipping line 5297: expected 11 fields, saw 12
Skipping line 5299: expected 11 fields, saw 12
Skipping line 5300: expected 11 fields, saw 12
Skipping line 5301: expected 11 fields, saw 12
```

YVideos Descriptive Analytics

```
US_videos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7992 entries, 0 to 7991
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   video_id        7992 non-null   object 
 1   title           7992 non-null   object 
 2   channel_title   7992 non-null   object 
 3   category_id     7992 non-null   int64  
 4   tags            7992 non-null   object 
 5   views           7992 non-null   int64  
 6   likes           7992 non-null   int64  
 7   dislikes        7992 non-null   int64  
 8   comment_total   7992 non-null   int64  
 9   thumbnail_link  7992 non-null   object 
 10  date            7992 non-null   float64
dtypes: float64(1), int64(5), object(5)
memory usage: 686.9+ KB
```

```
[ ] US_videos.shape
```

```
(7992, 11)
```

```
US_videos.nunique()
```

```
video_id          2364
title             2398
channel_title     1230
category_id         16
tags              2204
views             7939
likes             6624
dislikes          2531
comment_total     4152
thumbnail_link    2364
date                40
dtype: int64
```

YComments Descriptive Analytics

```
US_comments.nunique()
```

```
video_id          472
comment_text    80752
likes             328
replies           141
dtype: int64
```

```
US_comments.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 120663 entries, 0 to 120663
Data columns (total 4 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   video_id      120663 non-null  object
 1   comment_text  120663 non-null  object
 2   likes         120663 non-null  object
 3   replies       120663 non-null  object
dtypes: object(4)
memory usage: 4.6+ MB
```

```
US_comments.shape
```

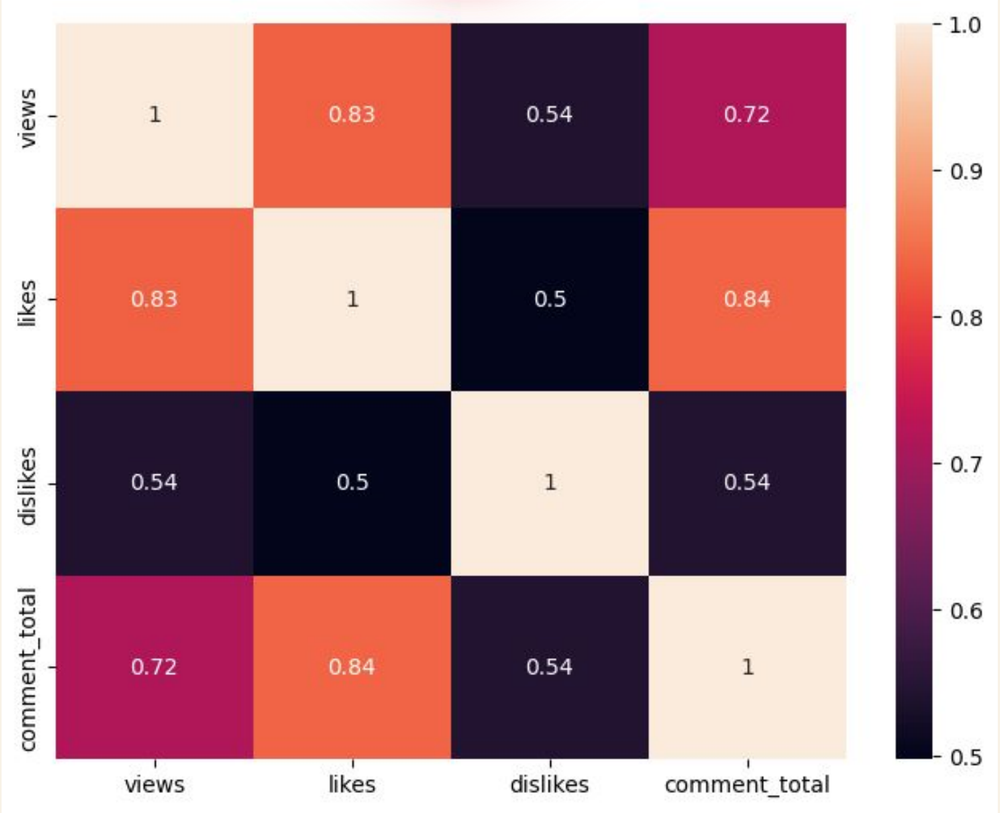
```
(120665, 4)
```

```
US_comments.isnull().sum()
```

```
video_id        0
comment_text    2
likes           1
replies         1
dtype: int64
```

YVideos Correlation Matrix

- This heatmap visualization illustrates the correlation levels between numeric features.
- According to this chart there is a strong relationship between:
  - likes and views
  - as well as likes and total number of comments
- This highlights the importance of exposure and engagement.

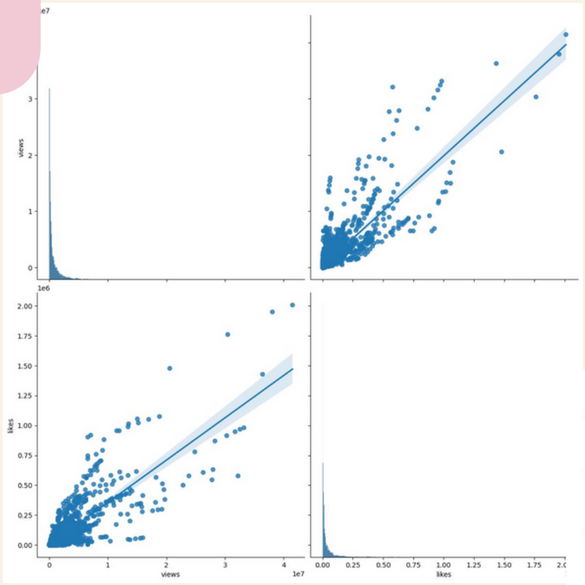


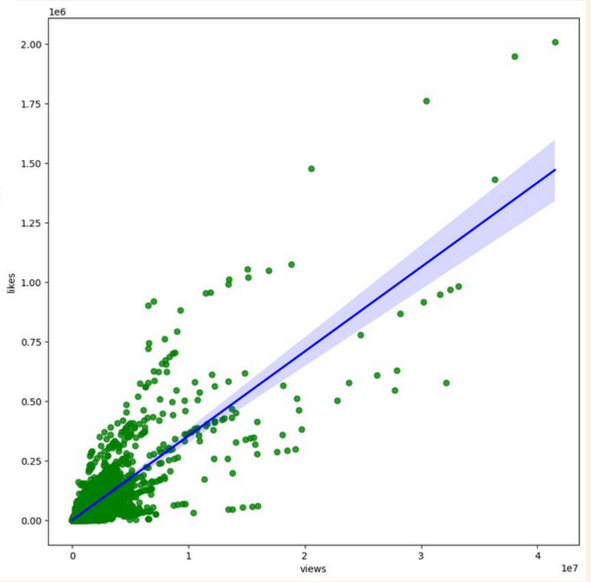
YVideos Correlation

We can take a closer look at the relationship between likes and view count with the following scatter plots.

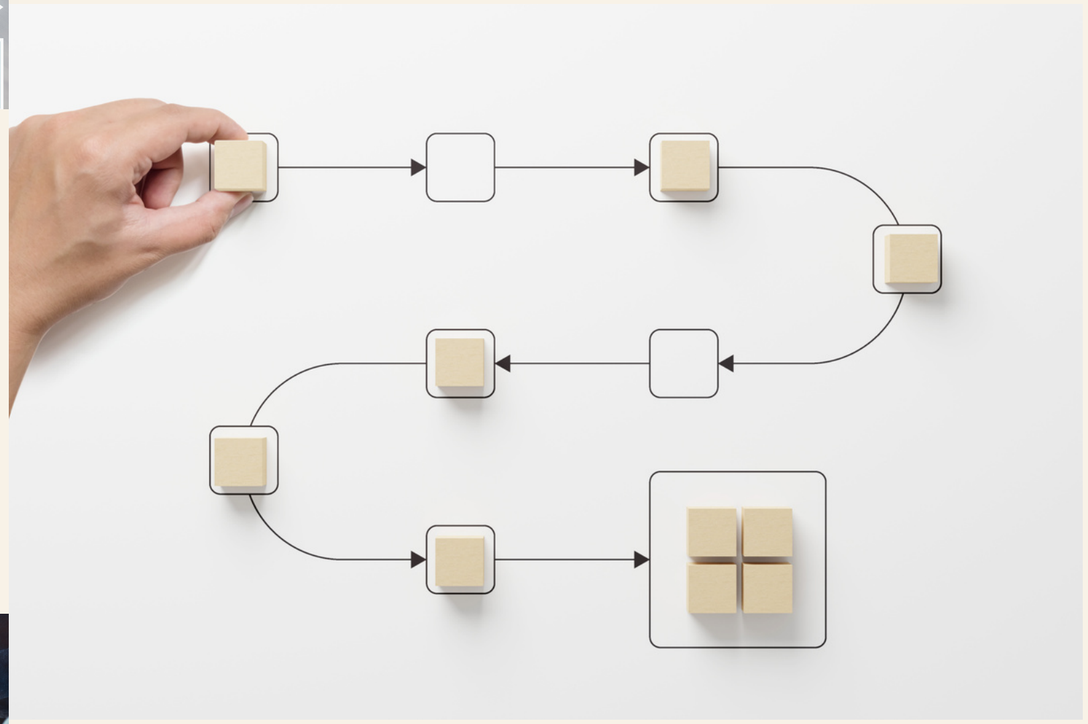
This suggests that there is a strong relationship between the two variables, highlighted with a trendline.

This makes intuitive sense, suggesting the greater the views, the greater the likes and vice versa.





Data Processing



Refine your data with preprocessing techniques to prepare our dataset for sentiment analysis

02 - Data Preprocessing

This step will consist of two phases: data cleaning and preprocessing

Data Cleaning will begin by removing:

- Defective Rows
- Unused Columns
- Null Values

Followed by our NLP, which starts by eliminating:

- Punctuation
- Numbers
- Special Characters
- and Stop Words.

This leaves us with a final body of text upon which we can do sentiment analysis.

VComments - Marking and Dropping Defective Rows

- At this step of our preprocessing, we removed all rows with corrupt data entries.
- Entries with values in the unnamed columns are marked and discarded.
- This and the integer assignments to our numeric columns mean we can now prepare our comments for sentiment analysis.

```
# Marking defective rows
indices = []

for index, row in US_comments.iterrows():
    if pd.isnull(US_comments['Unnamed: 4'][index]) == False:
        print("Error detected in line number {}".format(index))
        indices.append(index)
```

```
Error detected in line number 245210
Error detected in line number 388422
```

```
[12] # Dropping Marked Index

     for i in indices:
         US_comments.drop(i, inplace=True)
         print("Dropped index {}".format(i))
```

```
Dropped index 245210
Dropped index 388422
```

Dropping Unused Columns

```
# With the faulty entries removed, we have no more use for the unamed columns.

US_comments = pd.DataFrame(US_comments, columns=columns)
US_comments.dropna(inplace=True)
US_comments = US_comments.reset_index().drop('index',axis=1)
```

Dropping Null Values

```
[ ]  US_comments.dropna(inplace=True)
```

```
US_comments.isnull().sum()
```

```
video_id        0
comment_text    0
likes           0
replies         0
dtype: int64
```

```
[ ]  US_comments.shape
```

```
(120663, 4)
```

Normalizing Data for Sentiment Analysis

Removing Punctuations, Numbers and Special Characters.

```
[ ]  US_comments['comment_text'] = US_comments['comment_text'].str.replace("[^a-zA-Z#]", " ")
```

Removing Short Words.

```
[ ]  US_comments['comment_text'] = US_comments['comment_text'].apply(lambda x: ' '.join([w for w in x.split() if len(w)>3]))
```

Changing the text to lower case.

```
US_comments['comment_text'] = US_comments['comment_text'].apply(lambda x:x.lower())
```

Tokenization and Lemmatization

- **Tokenization** is the step by which the character string in a text segment is turned into units – tokens – for further analysis.
- **Lemmatization** links similar meaning words as one word, making tools such as chatbots and search engine queries more effective and accurate.
    - The goal of lemmatization is to reduce a word to its root form, also called a lemma.

Tokenization

```
[ ] tokenized_tweet = US_comments['comment_text'].apply(lambda x: x.split())
    tokenized_tweet.head()
```

```
0                                          [logan, paul]
1    [been, following, from, start, your, vine, cha...
2                                       [kong, maverick]
3                                          [attendance]
4                                            [trending]
Name: comment_text, dtype: object
```

Lemmatization

```
[ ] from nltk.stem import WordNetLemmatizer
    from nltk.corpus import stopwords
```

```
[ ] wnl = WordNetLemmatizer()
```

```
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
tokenized_tweet.apply(lambda x: [wnl.lemmatize(i) for i in x if i not in set(stopwords.words('english'))])
tokenized_tweet.head()
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
0                                          [logan, paul]
1    [been, following, from, start, your, vine, cha...
2                                       [kong, maverick]
3                                          [attendance]
4                                            [trending]
Name: comment_text, dtype: object
```

Implications of the findings

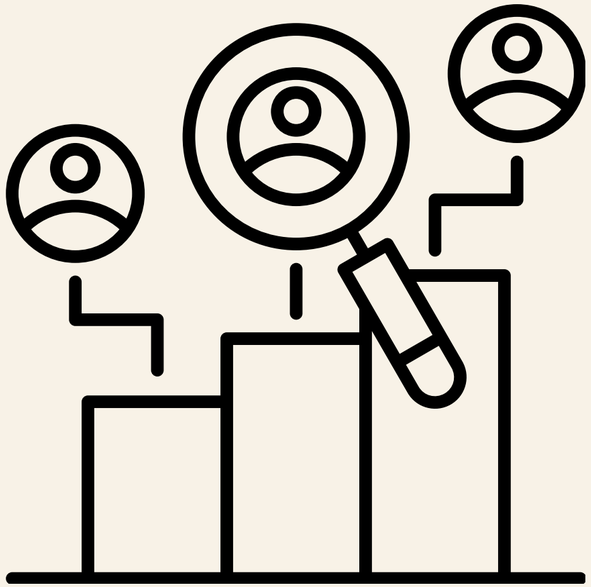
Sentiment Analysis Provides valuable feedback for quick, aggregate analysis. This has important implications for content creators and marketers alike.

03 - Sentiment Analysis

- Sentiment analysis, also known as opinion mining, is a natural language processing technique that aims to determine the emotional tone and attitudes expressed in a piece of text.

Why Sentiment Analysis?

- It helps to identify whether the text conveys positive, negative, neutral, or mixed emotions, enabling businesses to gauge customer satisfaction, public opinion, and brand perception.
- Sentiment analysis plays a crucial role in business and marketing strategies.
- By analyzing customer feedback, social media posts, product reviews, and survey responses, companies can gain valuable insights into consumer sentiments and preferences.



Calculating Sentiment Score

```
[ ]  for i in range(len(tokenized_tweet)):
         tokenized_tweet[i] = ' '.join(tokenized_tweet[i])
```

```
[ ]  US_comments['comment_text'] = tokenized_tweet
```

Let's do the Sentiment Analysis on the US Comments Dataset

```
import nltk
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
True
```

```
[ ]  from nltk.sentiment.vader import SentimentIntensityAnalyzer
     sia = SentimentIntensityAnalyzer()
```

Setting The Sentiment Scores

```
[ ]  US_comments['Sentiment Scores'] = US_comments['comment_text'].apply(lambda x:sia.polarity_scores(x)['compound'])
```

Classifying Sentiments

Classifying the Sentiment scores as Positive, Negative and Neutral

```
[ ]  US_comments['Sentiment'] = US_comments['Sentiment Scores'].apply(lambda s : 'Positive' if s > 0 else ('Neutral' if s == 0 else 'Negative'))
```

```
[ ]  US_comments.head()
```

	video_id	comment_text	likes	replies	Sentiment Scores	Sentiment
0	XpVt6Z1Gjjo	logan paul	4	0	0.0	Neutral
1	XpVt6Z1Gjjo	been following from start your vine channel ha...	3	0	0.0	Neutral
2	XpVt6Z1Gjjo	kong maverick	3	0	0.0	Neutral
3	XpVt6Z1Gjjo	attendance	3	0	0.0	Neutral
4	XpVt6Z1Gjjo	trending	3	0	0.0	Neutral

```
[ ]  US_comments.Sentiment.value_counts()
```

```
Positive    54112
Neutral     45053
Negative    21497
Name: Sentiment, dtype: int64
```

Positive comments examples

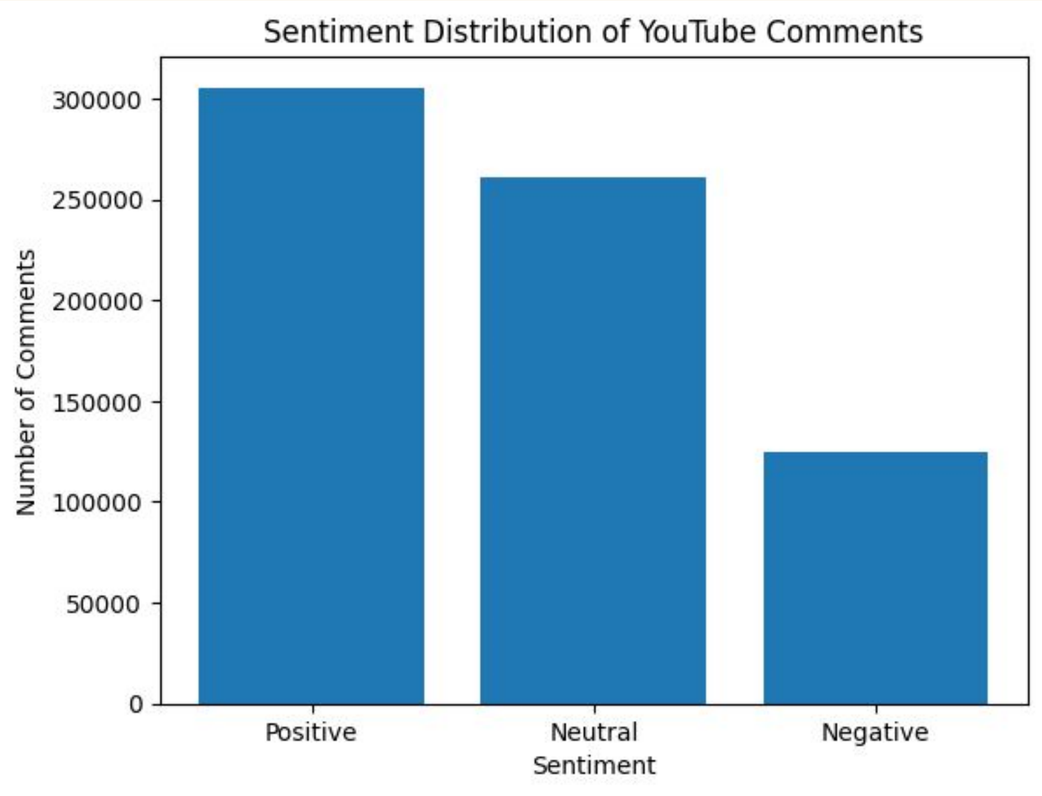
- **I love this video! It's so inspiring!**
- This is exactly what I needed to hear today. Thank you for sharing!
- Your channel always puts me in a good mood, keep up the great work!

Negative comments examples

"Waste of time, didn't learn anything new."

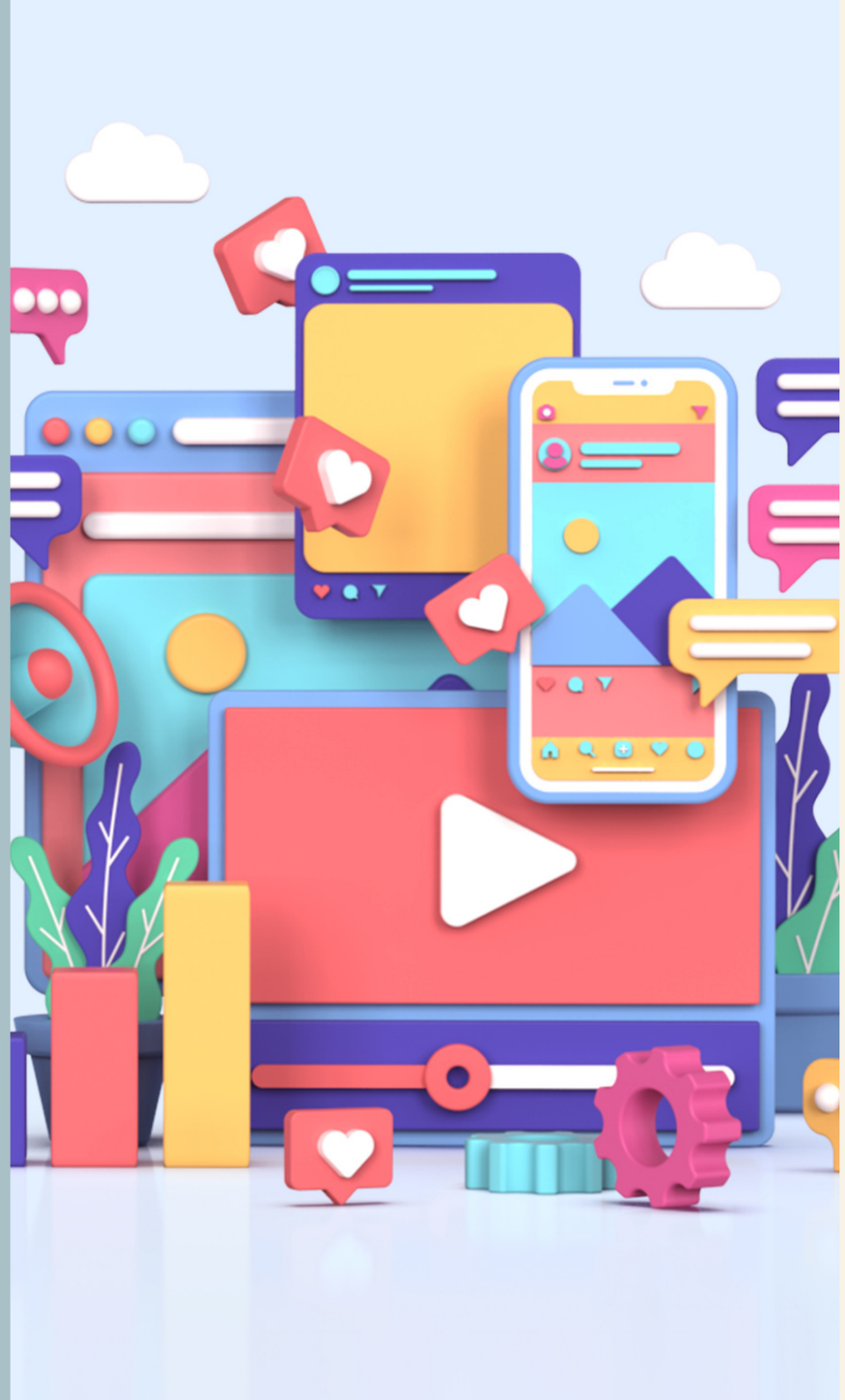
"This video is terrible, the speaker is so boring."

"I can't believe how bad the production quality is."



Sentiment Distribution

The sentiment analysis of YouTube comments shows that 44% of the reactions were positive, 38% were neutral, and 18% were negative.



Word Frequency

- Use natural language processing (NLP) tools to analyze text data.
- After Lemmatization, Tokenization, Sentiment Analysis and Classification, our next step was to obtain the leading topic words by frequency.
- The words with the highest frequency can be visualized using a word cloud.

04 - WordClouds

Most frequent words

- A word cloud in sentiment analysis provides a visually appealing representation of the most frequent words or phrases found in the analyzed text.
- The size of each word in the cloud corresponds to its frequency of occurrence, allowing users to quickly identify the most prevalent sentiments expressed in the text.

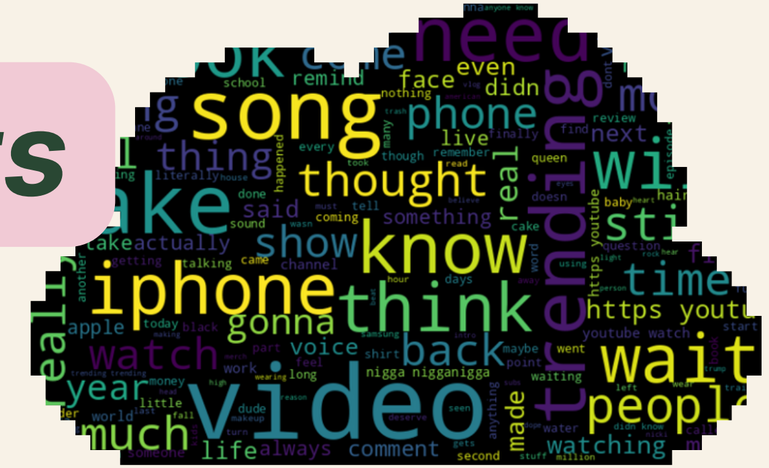

```
all_words = ' '.join([text for text in US_comments['comment_text']])
from wordcloud import WordCloud
wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(all_words)

plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```

WordClouds (cont.)

Word clouds condense large volumes of text data into a concise and interpretable format. By highlighting prominent sentiment-related words, users can grasp the prevailing emotions or opinions within the text without having to read through the entire content, making it a valuable tool for summarizing sentiment patterns.

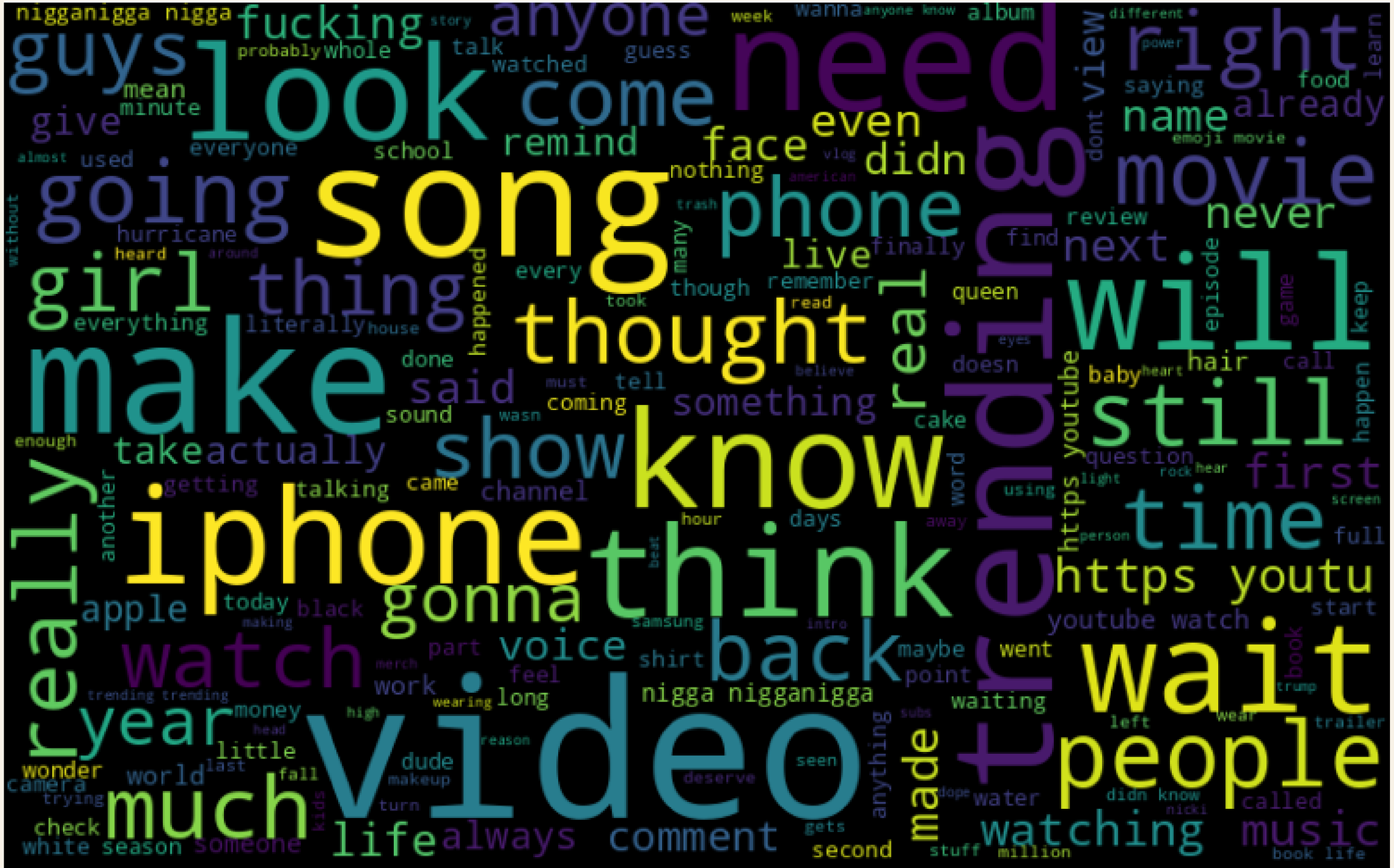
Positive Comments

Neutral Comments



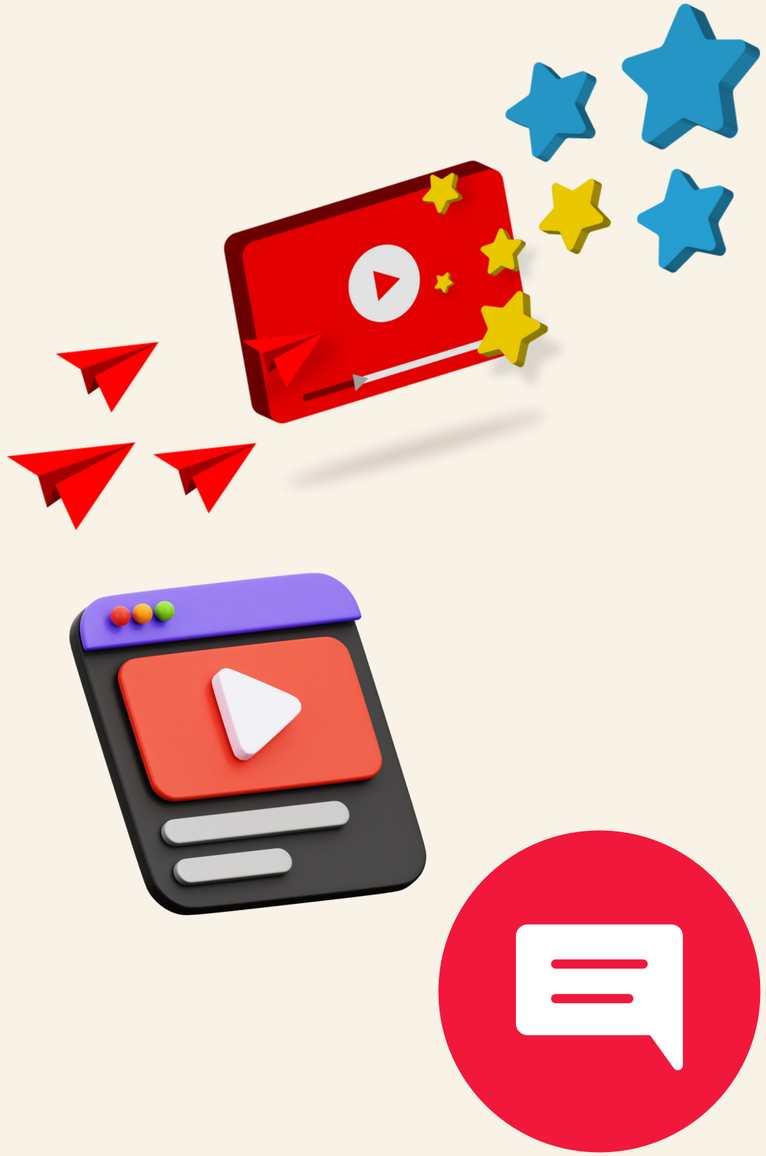
WordCloud for Positive Comments

WordCloud for Neutral Comments



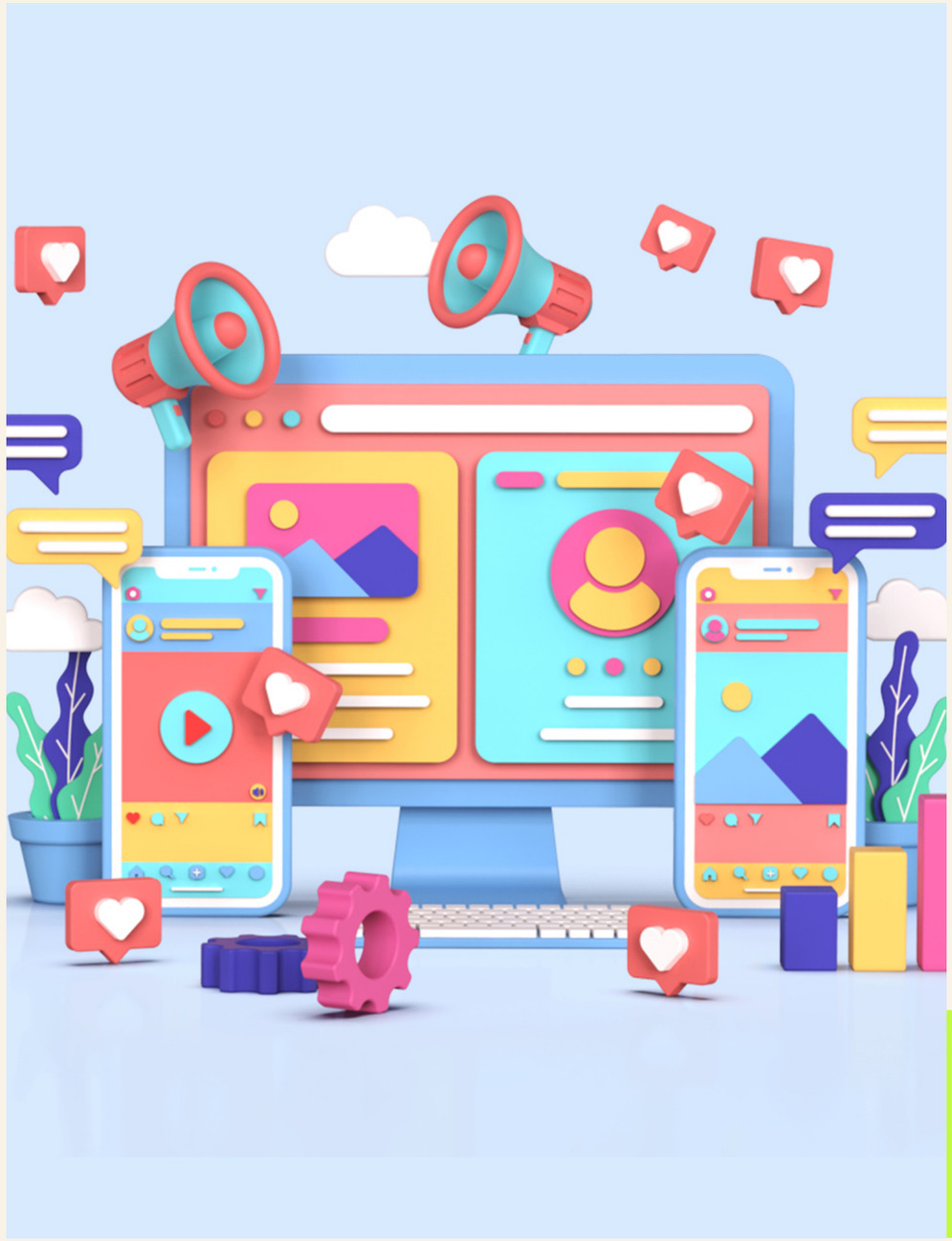
05 - Findings & Conclusion

- *By conducting this analysis, we achieved the following outcomes:*
- *Insight into User Sentiments:*
  - *We gained a holistic view of how users perceive and react to YouTube videos.*
  - *Understanding the distribution of sentiments helped identify the overall sentiment balance and potential areas for content improvement.*
- *Feedback for Video Creators:*
  - *The sentiment analysis results offered constructive feedback for content creators to gauge the impact of their videos on the audience.*
  - *Positive sentiments highlighted successful aspects that could be emphasized in future content, while negative sentiments indicated areas requiring enhancement or change.*
- *Audience Engagement Strategies:*
  - *By identifying the most common positive and negative keywords through word clouds, creators can tailor their content to resonate with the audience effectively.*
  - *This valuable insight can help shape engagement strategies and lead to more compelling video content.*



Potential uses for businesses and individuals

- Monitor your brand’s reputation and customer satisfaction.
- Gain insights into what your audience likes and dislikes.
- Identify trends and opportunities for improvement.

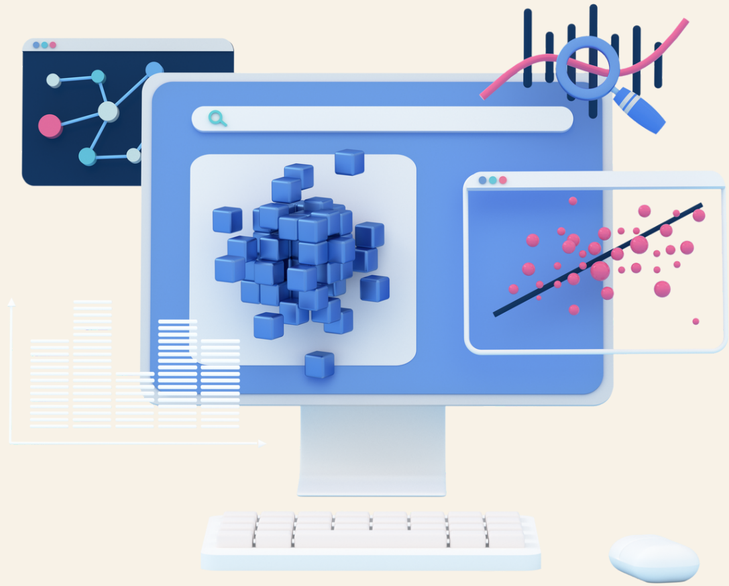


Future directions for research

- Sentiment analysis can help identify trends, customer preferences, and areas for improvement. Future research can explore multilingual analysis and video content analysis.

Takeaways

- In conclusion, this Python code for YouTube comments sentiment analysis proved to be a powerful tool for gaining insights into user sentiments, providing invaluable feedback to content creators, and guiding audience engagement strategies.
- By harnessing the power of NLP and sentiment analysis, we can enhance the YouTube experience, create impactful content, and foster a stronger connection with viewers.



Thank you for allowing us to share these insights with you today!