

**Assignment No. 1-2-3**

Semester	B.E. Semester VIII – Computer Engineering
Subject	Data Science Honor
Subject Professor In-charge	Prof. Amit Alyani
Academic Year	2024-25
Student Name	Deep Salunkhe
Roll Number	21102A0014

**Title:** YouTube video sentiment analysis

---

**Implementation:**

**1. Introduction**

Social Media Analytics involves extracting meaningful insights from user-generated content on platforms like YouTube, Twitter, and Facebook. **Sentiment analysis** helps determine whether the comments are positive, negative, or neutral, providing valuable feedback for content creators and marketers.

In this experiment, we extract **YouTube comments**, preprocess them using **text mining techniques**, and apply **machine learning classification algorithms** to analyze sentiment.

---

Product details



## YouTube Data API v3

[Google](#)

The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists,...

MANAGE

TRY THIS API

API Enabled

OVERVIEW

DOCUMENTATION

SUPPORT

RELATED PRODUCTS

### Overview

The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists, and channels.

#### About Google

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google

### Additional details

Type: [SaaS & APIs](#)

Last product update: 7/22/22

Category: [YouTube](#)

Service name: youtube.googleapis.com

## 2. Extracting YouTube Comments

### Tools & Methods

To extract YouTube comments, we use:

1. **YouTube Data API v3** – Provides structured access to YouTube comments.
2. **Google Colab** – For Python-based implementation.
3. **Pandas & NLP Libraries** – For data processing and analysis
4. .

### Steps for Extraction

1. **Enable YouTube Data API** from Google Cloud Console.
2. **Obtain API Key** for authentication.
3. **Use Python script** to fetch comments from a YouTube video.

75

```

from googleapiclient.discovery import build

# API Key and Video ID
api_key = "AIzaSyB4Pok_vojBvhTXAxZ0t9iCwJlVfMCLhW4"
video_id = "byJ7pxxhaDY"

# Build YouTube service
youtube = build("youtube", "v3", developerKey=api_key)

# Fetch comments
def fetch_comments(video_id):
    comments = []
    request = youtube.commentThreads().list(
        part="snippet",
        videoId=video_id,
        maxResults=100
    )
    response = request.execute()

    while response:
        for item in response["items"]:
            comment = item["snippet"]["topLevelComment"]["snippet"]["textDisplay"]
            comments.append(comment)

            # Check for next page token
            if "nextPageToken" in response:
                request = youtube.commentThreads().list(
                    part="snippet",
                    videoId=video_id,
                    pageToken=response["nextPageToken"],
                    maxResults=100
                )
                response = request.execute()
            else:
                break

    return comments

# Fetch and print comments
comments = fetch_comments(video_id)
for idx, comment in enumerate(comments[:100], start=1):
    print(f"{idx}: {comment}")

```

1: Watch Solo Leveling Season 2 here! <a href="http://got.cr/cc-sl2pv">got.cr/cc-sl2pv</a>  
2: Yep the animation is really nice they should've not remove some of the Gore stuff, it makes the video more epic... Just  
3: Ant dungeon in season 3 and double gate dungeon in season 4  
4: Every one is taking About the Aries ok Dona pubg mobile  
5: Mommy cha in the thumbnail is crazy approved baddie hehe  
6: <a href="https://www.youtube.com/watch?v=byJ7pXXhADY&t=59">0:59</a> is chedzu?  
7: Will Beru appear in a second part ?  
8: I caaaaaan't believe I cried over the end of the trailer  
9: Sister Leveling SEASON 2 IS BACK BABY!!!  
10: Is he gonna rewind time in S2?  
11: This is peak  
12: Feels like a sun wookong esk character that will eventually rebel against the gods and become too power  
13: I completed 200 episodes in manhwa  
14: Damn  
15: They ate they ate they ate  
16: Cool  
17: HELL YEAAAAAH  
18: A R I S E ! ! !  
19: Solo leveling season 2=<br>jin woo arua farm  
20: Bruhh!!! Did they animate everything already???  
21: The quality is incredible. I keep rewatching the episodes over and over again.  
22: God damn it, the plap scene is probably like idk, 5 or even 10 fking years from now  
23: Arise!  
24: so season will end before the jeju arc or after?  
25: He's aura not purple yet?  
26: <a href="https://www.youtube.com/watch?v=byJ7pXXhADY&t=18">0:18</a> bro went from gang to full on legion in just a few mont  
27: Please upload season 2  
28: I read the manhwa and I can say that this anime will be the best for 2025 and other seasons will be perfect.  
29: Shadow king has returned  
30: &quot;Only on Crunchyroll&quot;<br><br>LOL!   
31: Nice, I haven't heard TK since Tokyo Ghoul, and its goated opening.  
32: Lisa is top-notch in this song

### 3. Text Mining on Extracted Comments

Text mining involves **cleaning, preprocessing, and transforming** textual data into a structured format.

#### Preprocessing Steps

- **Remove Punctuation & Special Characters**
- **Convert to Lowercase**
- **Remove Stopwords**
- **Tokenization & Lemmatization**

```
100: Kinda glad they didnt change the names to Japanese like in first season
```

```
from textblob import TextBlob

def analyze_sentiment(comment):
    # Create a TextBlob object
    blob = TextBlob(comment)

    # Get polarity: -1 = negative, 0 = neutral, 1 = positive
    polarity = blob.sentiment.polarity

    # Determine sentiment based on polarity
    if polarity > 0:
        sentiment = "Positive"
    elif polarity < 0:
        sentiment = "Negative"
    else:
        sentiment = "Neutral"

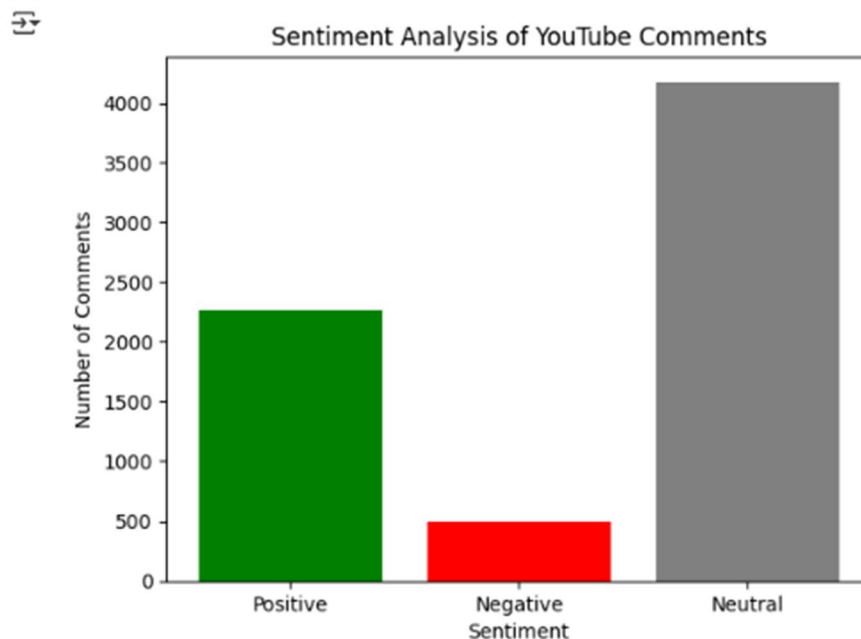
    return sentiment, polarity

# Example: Analyze sentiment of the first 10 comments
for idx, comment in enumerate(comments, start=1):
    sentiment, polarity = analyze_sentiment(comment)
    print(f"Comment {idx}: Sentiment = {sentiment}, Polarity = {polarity}")
```

```
Comment 6780: Sentiment = Neutral, Polarity = 0.0
Comment 6781: Sentiment = Neutral, Polarity = 0.0
Comment 6782: Sentiment = Neutral, Polarity = 0.0
Comment 6783: Sentiment = Neutral, Polarity = 0.0
Comment 6784: Sentiment = Positive, Polarity = 0.25
Comment 6785: Sentiment = Positive, Polarity = 0.2857142857142857
Comment 6786: Sentiment = Positive, Polarity = 0.35
Comment 6787: Sentiment = Neutral, Polarity = 0.0
Comment 6788: Sentiment = Neutral, Polarity = 0.0
Comment 6789: Sentiment = Neutral, Polarity = 0.0
Comment 6790: Sentiment = Neutral, Polarity = 0.0
Comment 6791: Sentiment = Neutral, Polarity = 0.0
Comment 6792: Sentiment = Neutral, Polarity = 0.0
Comment 6793: Sentiment = Neutral, Polarity = 0.0
Comment 6794: Sentiment = Positive, Polarity = 0.25
Comment 6795: Sentiment = Neutral, Polarity = 0.0
Comment 6796: Sentiment = Neutral, Polarity = 0.0
Comment 6797: Sentiment = Neutral, Polarity = 0.0
Comment 6798: Sentiment = Neutral, Polarity = 0.0
Comment 6799: Sentiment = Neutral, Polarity = 0.0
Comment 6800: Sentiment = Positive, Polarity = 0.6
Comment 6801: Sentiment = Neutral, Polarity = 0.0
Comment 6802: Sentiment = Positive, Polarity = 0.7
Comment 6803: Sentiment = Neutral, Polarity = 0.0
```

```
Comment 6837: Sentiment = Neutral, Polarity = 0.0  
Comment 6838: Sentiment = Neutral, Polarity = 0.0
```

```
25 import matplotlib.pyplot as plt  
  
# Analyze sentiment of the comments  
sentiments = {"Positive": 0, "Negative": 0, "Neutral": 0}  
  
for comment in comments:  
    sentiment, _ = analyze_sentiment(comment)  
    sentiments[sentiment] += 1  
  
# Create a bar graph for the sentiments  
labels = list(sentiments.keys())  
values = list(sentiments.values())  
  
plt.bar(labels, values, color=["green", "red", "gray"])  
plt.title("Sentiment Analysis of YouTube Comments")  
plt.xlabel("Sentiment")  
plt.ylabel("Number of Comments")  
plt.show()
```



## 6. Conclusion

This experiment demonstrated:

- **YouTube comment extraction** using **YouTube Data API**.
- **Text mining** for cleaning and preprocessing.
- **Sentiment analysis** using **machine learning classification models**.



**Insights:**

- Most of the audience thought the trail of average not too good not to bad
- Very few thought it was bad adaptation.