## **EXPERIMENT NO. 2**

**Aim:** Data Collection – Select the social media platforms of your choice (Twitter, Facebook, LinkedIn, YouTube, Web blogs, etc.), connect to and capture social media data for business. (scraping, crawling, parsing)

**Software tools:**

* GOOGLE CLOUD API
* GOOGLE COLAB

**Theory:**

* **Scraping:** Scraping refers to the process of extracting data from websites. This could involve pulling information from web pages, APIs, or any other online source. Web scraping can be done manually, but it's often automated using software tools known as web scrapers or crawlers. Scraping can involve extracting text, images, links, or any other content available on the web.
* **Crawling:** Crawling is the process of systematically browsing the internet to discover and index web pages. Search engines like Google use crawlers (also known as spiders or bots) to traverse the web, following links from one page to another and gathering information about each page they visit. Crawling is a crucial step in indexing web content, enabling search engines to provide relevant results to users.
* **Parsing:** Parsing involves breaking down a piece of data into its component parts and interpreting its structure. In the context of web scraping, parsing is often used to extract specific information from HTML or other markup languages. For example, a parser might extract the text of a particular HTML element, extract attributes from tags, or identify patterns within the text. Parsing is essential for extracting meaningful data from raw web pages during the scraping process.

1. **Scraping YouTube Video Comments:**

**Step 1:** Get API Key – Go to the [Google Cloud Console](https://console.cloud.google.com/) Google Cloud Console and sign in with your Google Account.

**Step 2:** Click the project drop-down menu in the top bar and select or create the project you want to use.

**Step 3:** Click the hamburger menu in the top left and select APIs & Services > Select YouTube Data API V3 > Credentials.

**Step 4:** Click the Create credentials button and select API key. The API key will be displayed in a pop-up window. You can click the RESTRICT KEY button to restrict the API keys usage, such as by IP address or referrer. Click the COPY button to copy the API key to your clipboard.

**Step 5:** You can use the API key in your application to access the Google Cloud APIs. Be sure to keep the API key confidential, as it can be used to access your Google Cloud resources.

* **Use API key and video id:**

**Input:**

import requests

import json

video\_id = "Ct8Gxo8StBU"

api\_key = "AIzaSyAMgD\_1ZKx5x\_5wkZSawSZGhT8hihz0q2I"

video\_info\_url = f"https://www.googleapis.com/youtube/v3/videos?part=snippet&id={video\_id}&key={api\_key}"

video\_info\_response = requests.get(video\_info\_url)

video\_info\_data = video\_info\_response.json()

print(json.dumps(video\_info\_data, indent=4))

**Output:**



* **Retrieve video information:**

**Input:**

if 'items' in video\_info\_data and video\_info\_data['items']:

    title = video\_info\_data['items'][0]['snippet']['title']

    description = video\_info\_data['items'][0]['snippet']['description']

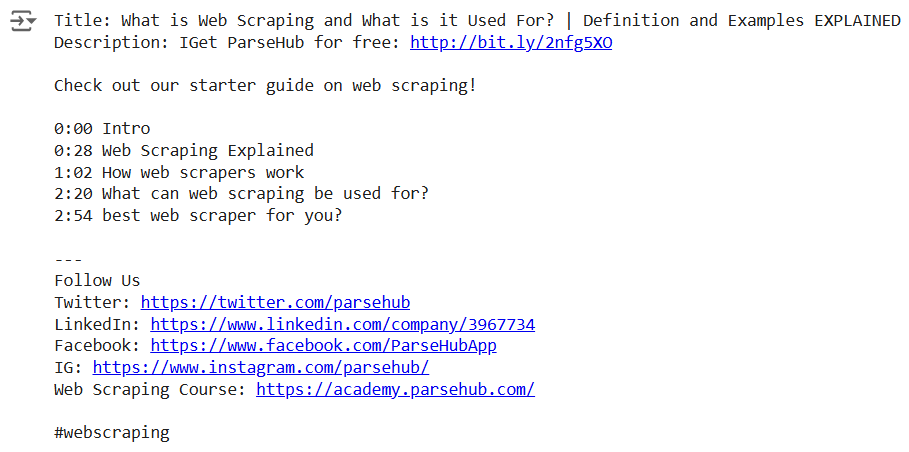
    print("Title:", title)

    print("Description:", description)

else:

    print("Video information not found. Check the video ID and API key.")

**Output:**



* **Retrieve and Extract video comments:**

**Input:**

comments\_url =f"https://www.googleapis.com/youtube/v3/commentThreads?part=snippet&videoId={video\_id}&key={api\_key}"

comments\_response= requests.get(comments\_url)

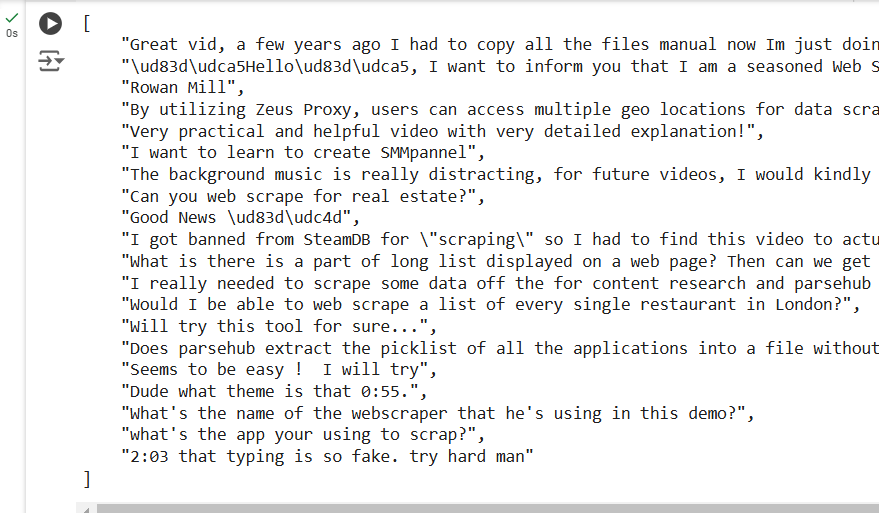
comments\_data = comments\_response.json()

**# Extract the comments**

comments = [item["snippet"]["topLevelComment"]["snippet"]["textOriginal"] for item in comments\_data["items"]]

print(json.dumps(comments, indent=4))

**Output:**



* **Analysis of YouTube comments:**

**Input:**

from textblob import TextBlob

def get\_comment\_sentiment(comment):

  analysis = TextBlob(comment)

  if  analysis.sentiment.polarity > 0:

    return "Positive"

  elif analysis.sentiment.polarity == 0:

    return "neutral"

  else:

    return "negative"

* **Displaying the sentiment of YouTube video comments:**

**Input:**

comment\_list = []

sentiment\_list= []

for comment in comments:

sentiment = get\_comment\_sentiment(comment)

comment\_list.append(comment)

sentiment\_list.append(sentiment)

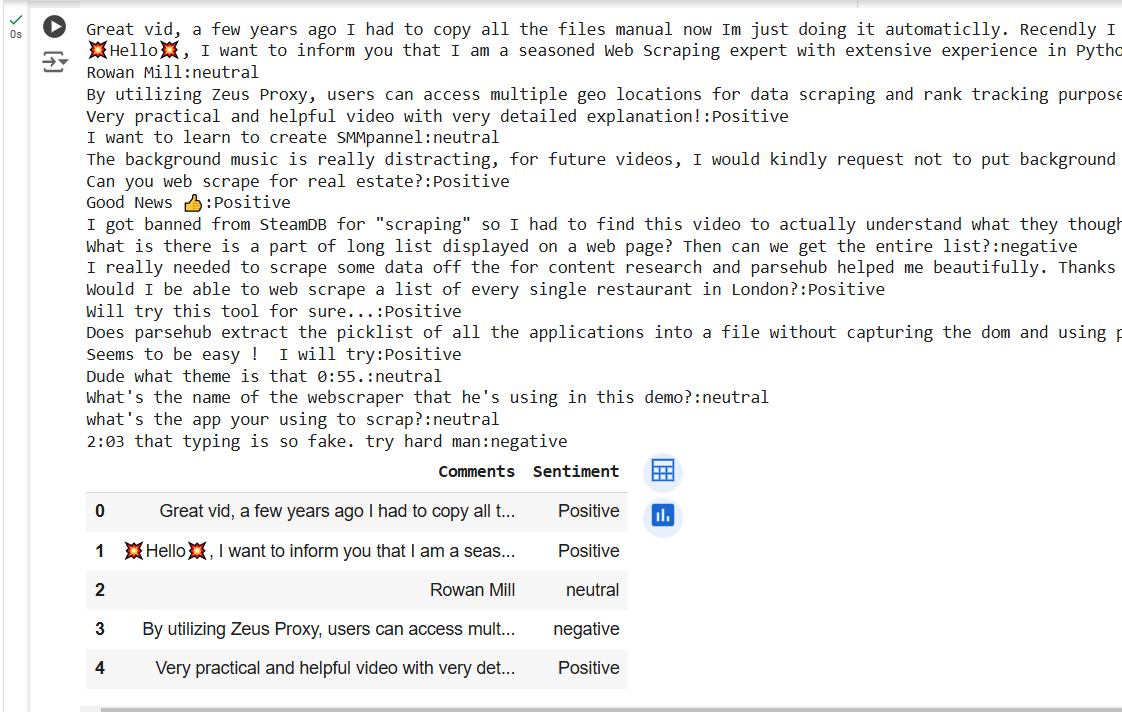
print(f"{comment}:{sentiment}")

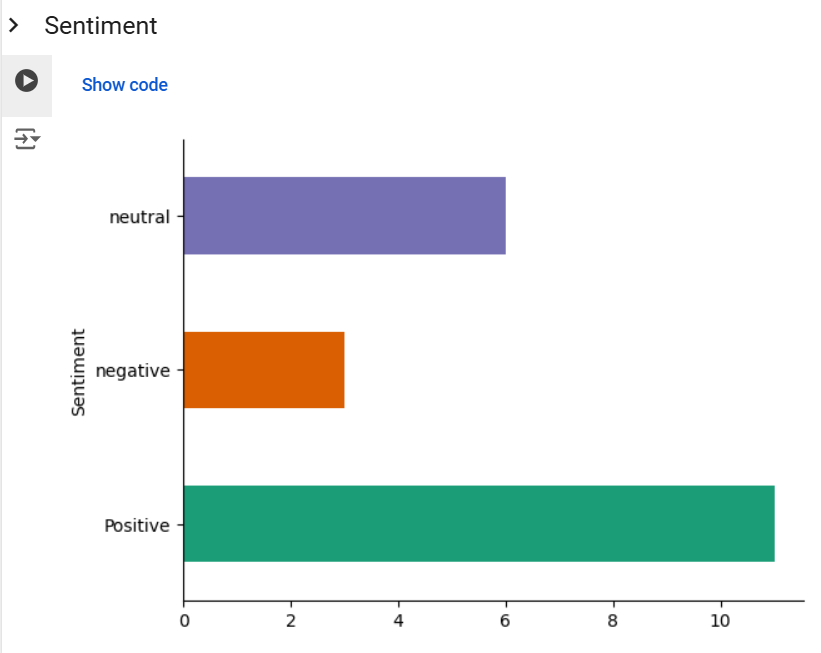
import pandas as pd

sentiment\_df = pd.DataFrame({"Comments": comment\_list,"Sentiment": sentiment\_list})

sentiment\_df.head()

**Output:**



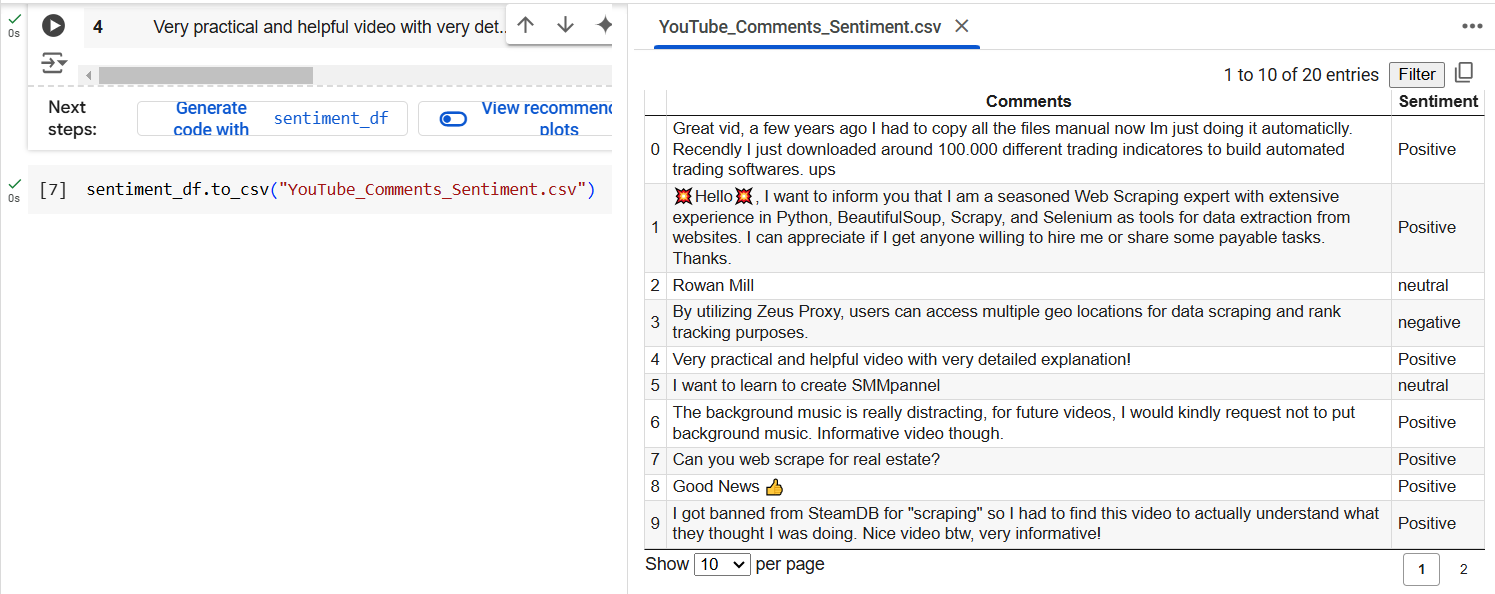


* **Save comments to CSV file:**

**Input:**

sentiment\_df.to\_csv("YouTube\_Comments\_Sentiment.csv")

**Output:**



**Conclusion:**

Hence, successfully scraped the comments of YouTube videos using API key.