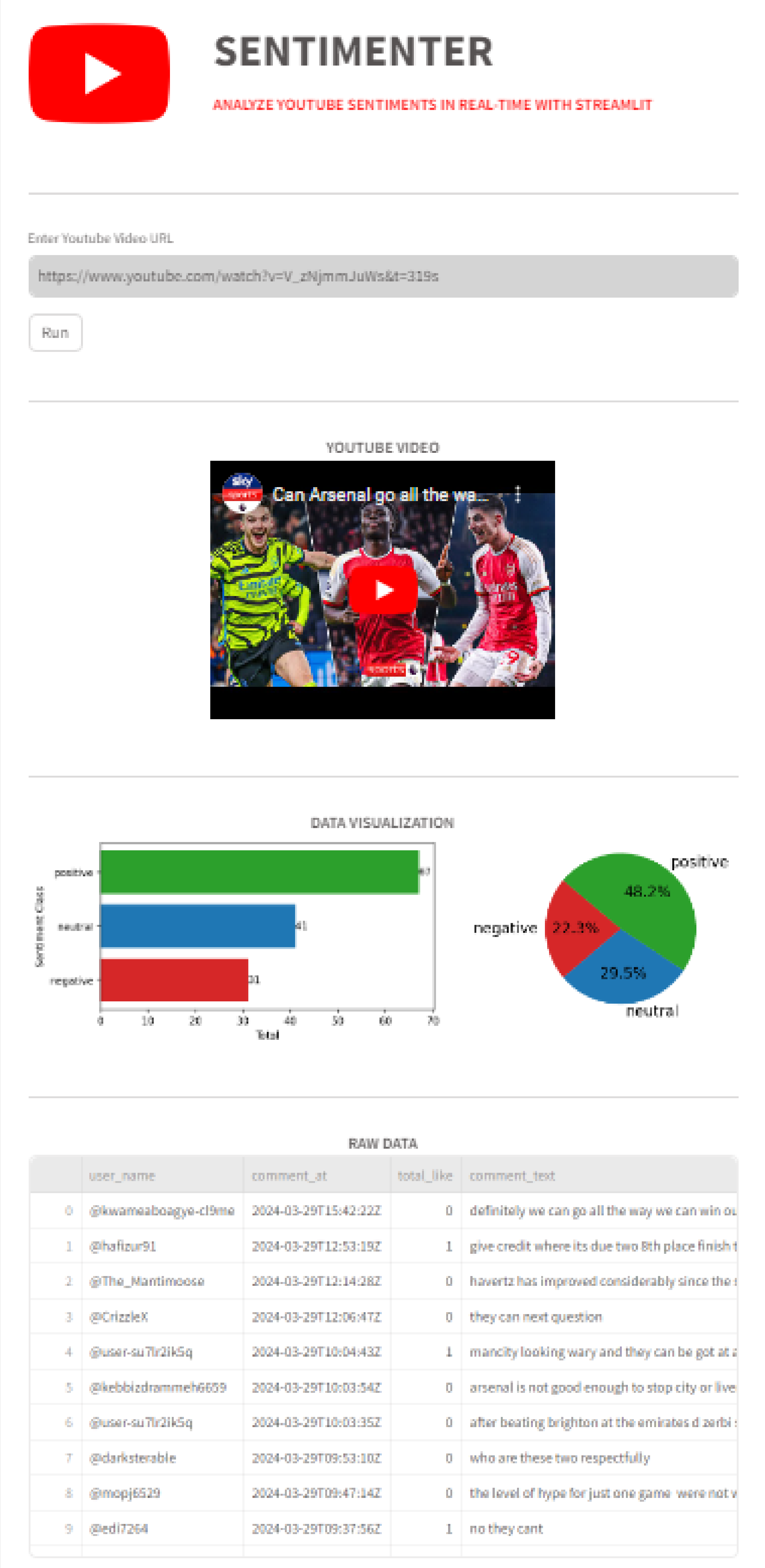
YOUTUBE SENTIMENT ANALYSIS:

UNVEILING THE POWER OF STREAMLIT WEB APPS!

Streamlit

PUBLISHED BY : ALFIANSYAH



SENTIMENTER

ANALYZE YOUTUBE SENTIMENTS IN REAL-TIME WITH STREAMLIT

Enter Youtube Video URL

https://www.youtube.com/watch?v=V_zNjmmJuWs&t=319s

Run

YOUTUBE VIDEO

DATA VISUALIZATION

RAW DATA

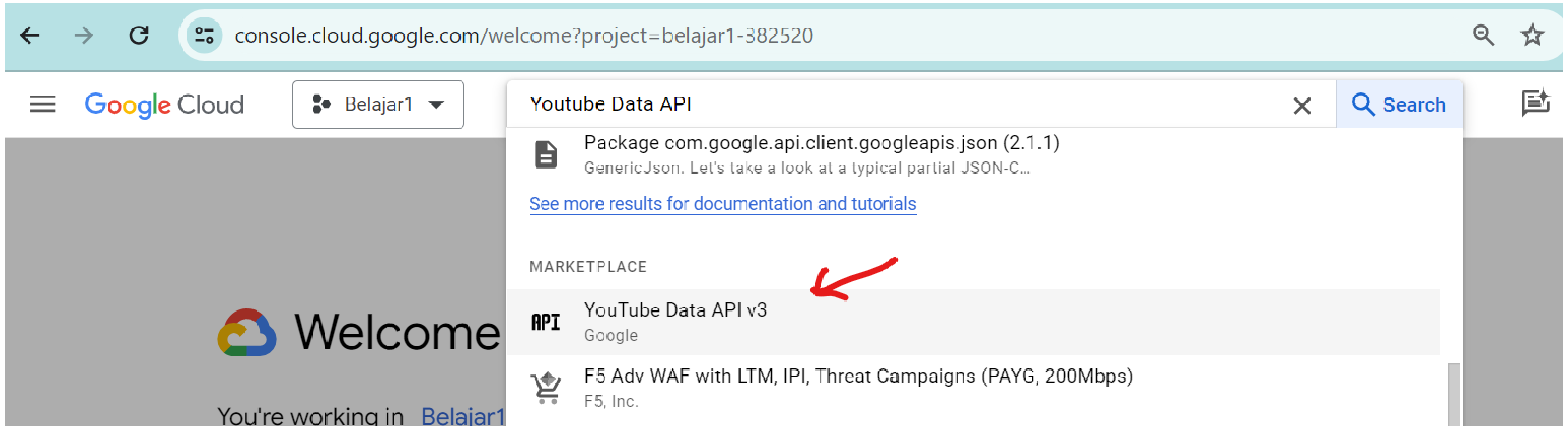
	user_name	comment_at	total_like	comment_text
0	@kwameaboagye-cl9me	2024-03-29T15:42:22Z	0	definitely we can go all the way we can win ou
1	@hafizur91	2024-03-29T12:53:19Z	1	give credit where its due two 8th place finish t
2	@The_Mantimoose	2024-03-29T12:14:28Z	0	havertz has improved considerably since the s
3	@CrizzleX	2024-03-29T12:06:47Z	0	they can next question
4	@user-su7lr2ik5q	2024-03-29T10:04:43Z	1	mancity looking wary and they can be got at a
5	@kebbizdrammeh6659	2024-03-29T10:03:54Z	0	arsenal is not good enough to stop city or live
6	@user-su7lr2ik5q	2024-03-29T10:03:35Z	0	after beating brighton at the emirates d zerbi s
7	@darksterable	2024-03-29T09:53:10Z	0	who are these two respectfully
8	@mopj6529	2024-03-29T09:47:14Z	0	the level of hype for just one game were not v
9	@edi7264	2024-03-29T09:37:56Z	1	no they cant

PROJECT FLOW

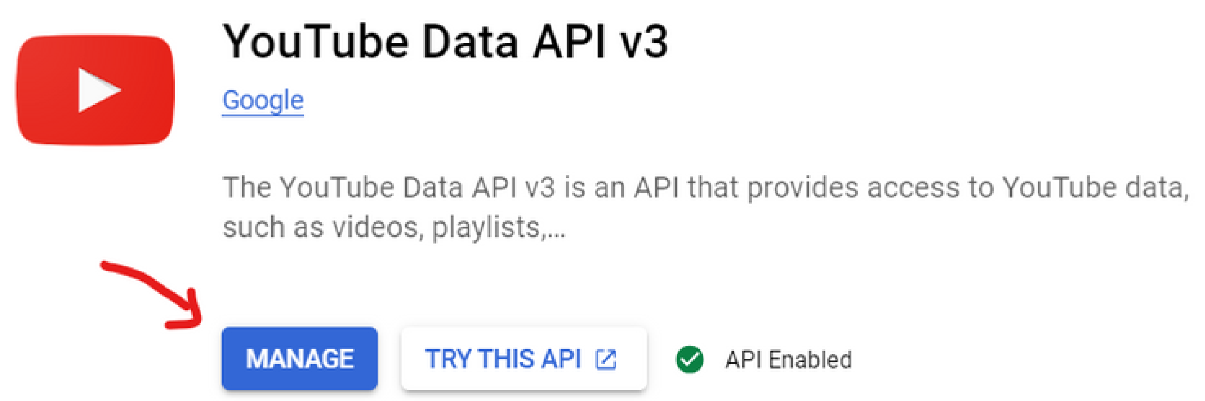
1	DATA COLLECTION	We will utilize the YouTube Data API v3 to collect data (comment) from the YouTube video titled "Can Arsenal go all the way this season? 🔴🏆" by the Sky Sport Premier League channel.
2	DATA PREPROCESSING	We will examine the summary of our data, converting all comments to lowercase and removing special characters, symbols, and punctuation marks.
3	SENTIMENT ANALYSIS	We will conduct sentiment analysis utilizing VADER (Valence Aware Dictionary and sEntiment Reasoner) scoring methodology.
4	DATA VISUALIZATION	We will create some visualization using Matplotlib.
5	BUILD WEB APP	We will build Web App using Streamlit.

DATA COLLECTION

Before we start the project, we need to generate an API Key from the YouTube Data API v3 to gather data from YouTube. We can acquire the API Key for free from https://console.cloud.google.com.


console.cloud.google.com/welcome?project=belajar1-382520
Google Cloud
Belajar1
Youtube Data API
Search
Package com.google.api.client.googleapis.json (2.1.1)
GenericJson. Let's take a look at a typical partial JSON-C...
See more results for documentation and tutorials
MARKETPLACE
API
YouTube Data API v3
Google
F5 Adv WAF with LTM, IPI, Threat Campaigns (PAYG, 200Mbps)
F5, Inc.
Welcome
You're working in Belajar1

and then we click manage

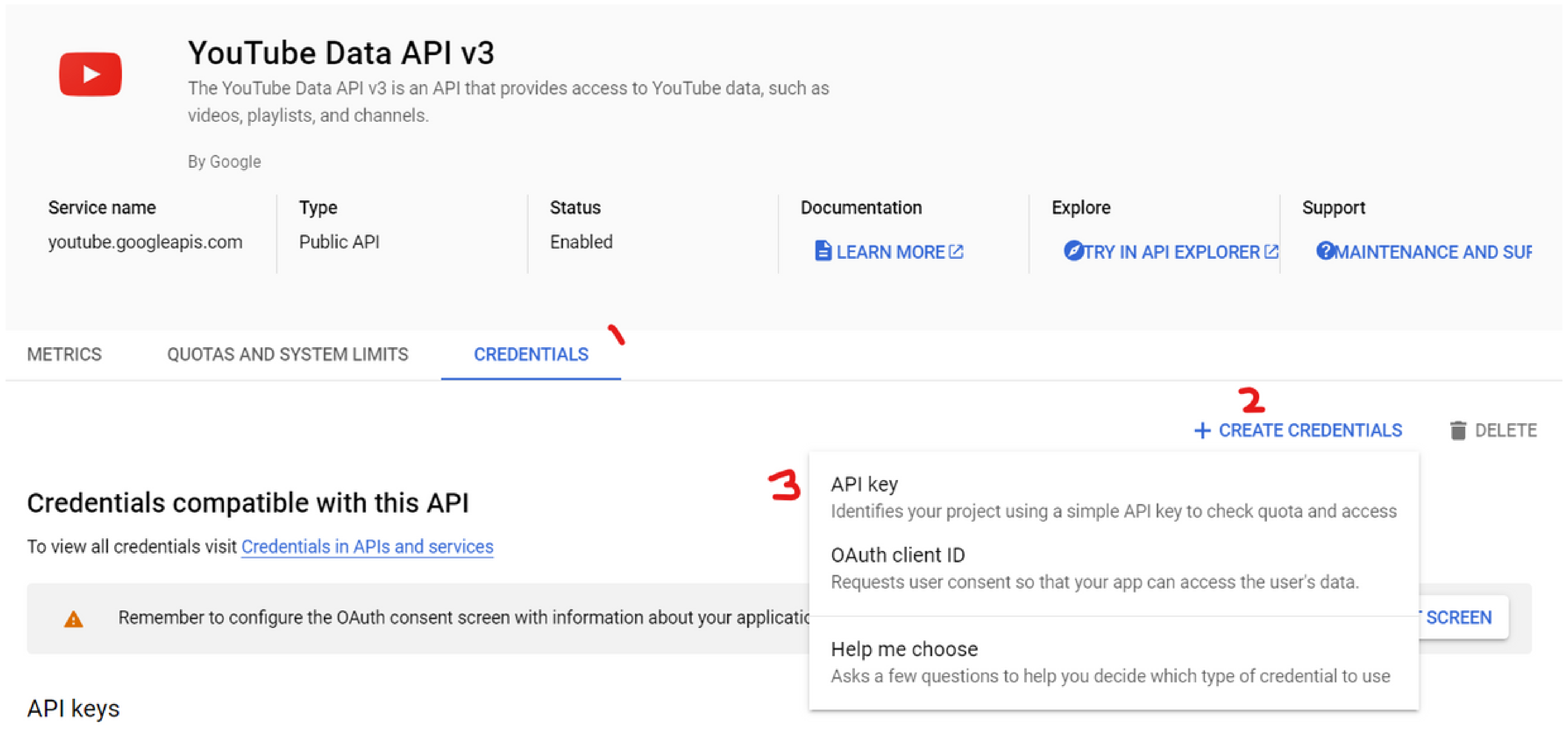

YouTube Data API v3

Google

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

DATA COLLECTION

Now, navigate to the credentials section and proceed to create new credentials. Select "API key" and your API key will be generated. Simply copy it for future use.



YouTube Data API v3

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

By Google

Service name	Type	Status	Documentation	Explore	Support
youtube.googleapis.com	Public API	Enabled	LEARN MORE	TRY IN API EXPLORER	MAINTENANCE AND SUF

Credentials compatible with this API

To view all credentials visit Credentials in APIs and services

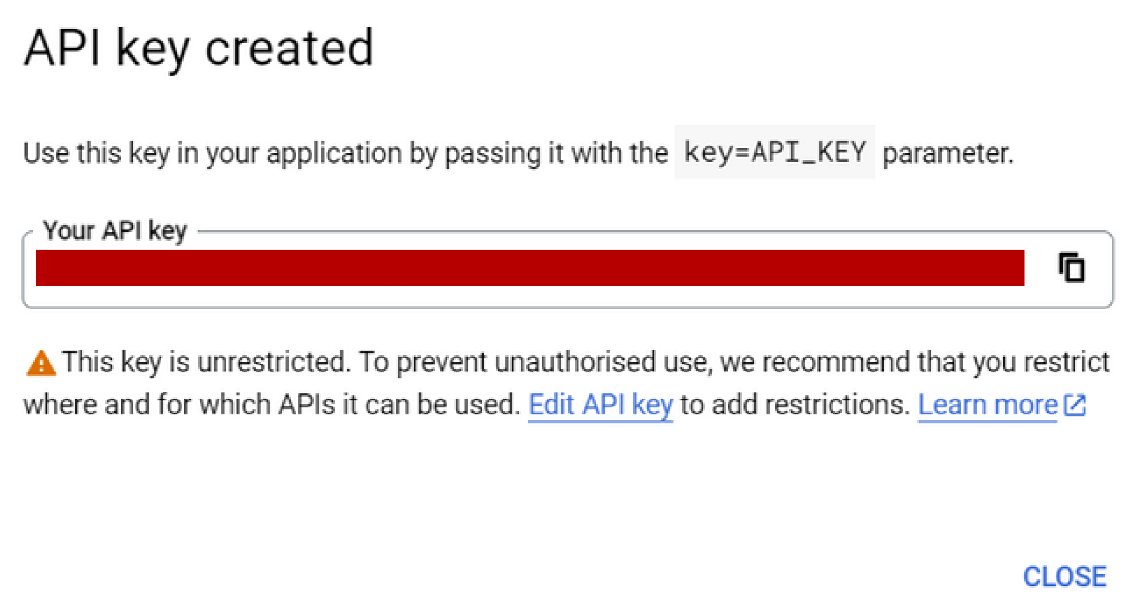
Remember to configure the OAuth consent screen with information about your applicatic

API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so that your app can access the user's data.

Help me choose
Asks a few questions to help you decide which type of credential to use

API keys



API key created

Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key
[redacted]

This key is unrestricted. To prevent unauthorised use, we recommend that you restrict where and for which APIs it can be used. Edit API key to add restrictions. Learn more

DATA COLLECTION

Let’s start coding!

First, import the essentials libraries.

```python
import googleapiclient.discovery
import pandas as pd
```

- googleapiclient.discovery is part of the Google API Client Library for Python. This library facilitates interaction with the YouTube Data API, enabling us to make requests to retrieve data from YouTube.
- pandas is a powerful data manipulation and analysis library in Python. With Pandas, we can efficiently convert the collected data into dataframes for subsequent analysis.

Then, let‘s call the Youtube API.

```python
api_service_name = "youtube"
api_version = "v3"
api_key = "Your API Key"

youtube = googleapiclient.discovery.build(
    api_service_name, api_version, developerKey=api_key)
```

Fill the api_key with your API key from the previous step.

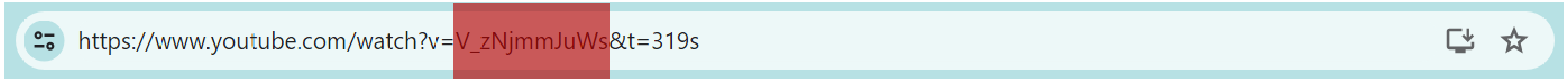
DATA COLLECTION

Let's proceed by making a request to the API to retrieve a response for a specific YouTube video.

```python
video_id = "V_zNjmmJuWs"
request = youtube.commentThreads().list(
    part="snippet",
    videoId= video_id,
    maxResults=100
)

response = request.execute()
```

You can get video_id from Youtube video URL, see the image bellow.



https://www.youtube.com/watch?v=V_zNjmmJuWs&t=319s

If we check the response, we'll find a lot of information presented in JSON format (Dictionary).

```python
response
```

```
{'kind': 'youtube#commentThreadListResponse',
 'etag': 'O-igwhZ0K1hxd7_3P2FKLFcvsec',
 'nextPageToken': 'Z2V0X25ld2VzdF9maXJzdC0tQ2dnSWdBUVZGN2ZST0JJRkNJZ2dHQUFTQlFpSklCZ0FFZ1VJblNBWUFSSUZDS2dnR0FBU0JRaUhJQmdBR0FBaURnb01DS1BRbGJBR0VJaTd
vX29C',
 'pageInfo': {'totalResults': 100, 'resultsPerPage': 100},
 'items': [{'kind': 'youtube#commentThread',
```

DATA COLLECTION

Let's gather only the essential information related to the comments: author, date, total likes, and the comment.

```python
comments = []

for item in response['items']:
    comment = item['snippet']['topLevelComment']['snippet']
    comments.append([
        comment['authorDisplayName'],
        comment['publishedAt'],
        comment['likeCount'],
        comment['textOriginal'],
    ])
```

If we check the results, we already have all that essential information.

```python
comments
```

```
[['@Iamsnowhitesamuel',
  '2024-03-29T04:49:13Z',
  0,
  'It was so surprising how Arsenal now became talking point in every pundit 😊😊'],
 ['@CEO786',
  '2024-03-29T03:47:19Z',
  0,
  'We’re guaranteed the league and most likely we’ll win the champions league as well finally! 🏆🏆'],
```

DATA COLLECTION

Currently, we can only get a max of 100 results per request. Therefore, to obtain all the comments, we need to create a loop function.

```
while (1 == 1):
  try:
   nextrun = response['nextPageToken']
  except KeyError:
   break
  nextrun = response['nextPageToken']
  nextRequest = youtube.commentThreads().list(part="snippet", videoId=video_id, maxResults=100, pageToken=nextrun)
  response = nextRequest.execute()
  for item in response['items']:
    comment = item['snippet']['topLevelComment']['snippet']
    comments.append([
        comment['authorDisplayName'],
        comment['publishedAt'],
        comment['likeCount'],
        comment['textOriginal'],
    ])
```

Then we can convert the results into a DataFrame using pandas and add column id from index.

```
df = pd.DataFrame(comments, columns=['user_name', 'comment_at', 'total_like', 'comment_text'])
df['id'] = df.index
```

DATA COLLECTION

Preview the data

```
df.head(5)
```

	user_name	comment_at	total_like	comment_text	id
0	@user-yk8sy9if4l	2024-03-29T06:02:37Z	0	I dont think so look top on goal diffrence the...	0
1	@Iamsnowhitesamuel	2024-03-29T04:49:13Z	0	It was so surprising how Arsenal now became ta...	1
2	@CEO786	2024-03-29T03:47:19Z	0	We're guaranteed the league and most likely we...	2
3	@LP-ct6xe	2024-03-29T03:35:47Z	0	hang on a second - why do we always have a wom...	3
4	@faisalmohamed7125	2024-03-29T00:47:49Z	0	COYG	4

Now that we already have all the data in a DataFrame, we can proceed to the next step for preprocessing the data.

DATA PREPROCESSING

Let's start by looking at information related to our data.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121 entries, 0 to 120
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   user_name     121 non-null    object
 1   comment_at    121 non-null    object
 2   total_like    121 non-null    int64 
 3   comment_text  121 non-null    object
 4   id            121 non-null    int64 
dtypes: int64(2), object(3)
memory usage: 4.9+ KB
```

As we can observe,

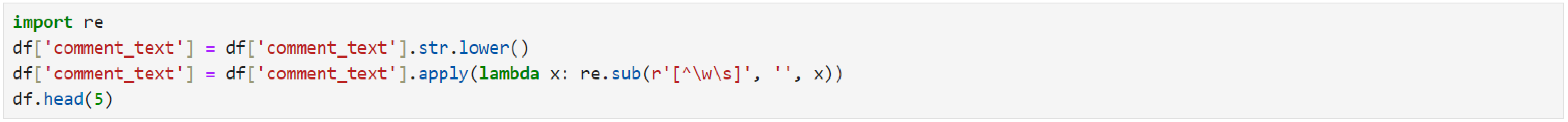
- **we have a total of 121 data entries,**
- **distributed across 4 columns,**
- **all without any null values.**
- **The columns "user_name", "comment_at", and "comment_text" are of object data type,**
- **while "total_like" and "id" is of int64 data type.**

So the data looks good.

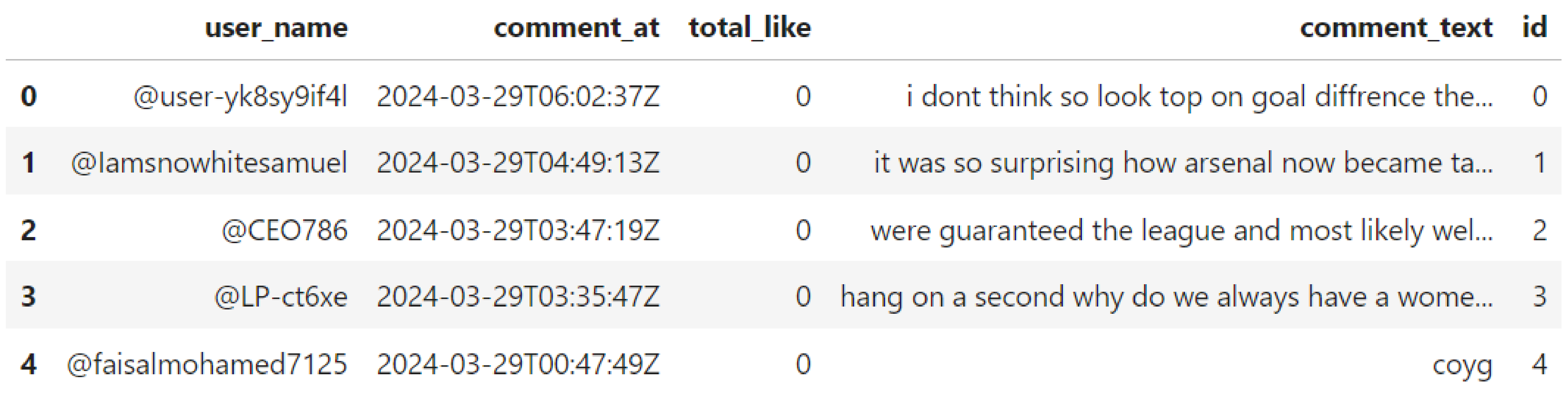
DATA PREPROCESSING

Now, we will preprocess our 'comment_text' column by converting the text to lowercase and removing special characters, symbols, and punctuation using the 're' library.

- **re library refers to Python's built-in regular expression library, which is used for pattern matching in strings. Specifically, the functions from the "re" library that would likely be used for this task include "re.sub()" for substituting patterns in strings.**



```python
import re
df['comment_text'] = df['comment_text'].str.lower()
df['comment_text'] = df['comment_text'].apply(lambda x: re.sub(r'[^\w\s]', '', x))
df.head(5)
```



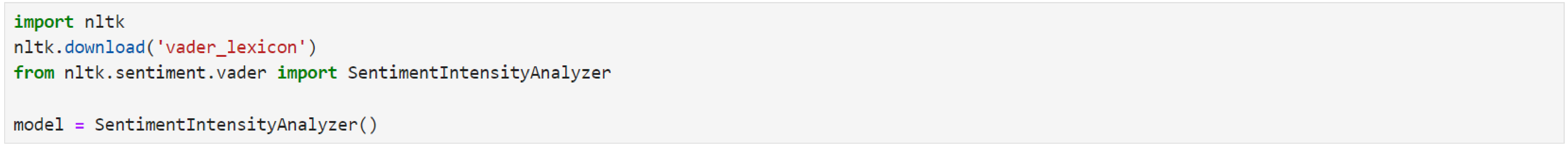
	user_name	comment_at	total_like	comment_text	id
0	@user-yk8sy9if4l	2024-03-29T06:02:37Z	0	i dont think so look top on goal diffrence the...	0
1	@lamsnowhitesamuel	2024-03-29T04:49:13Z	0	it was so surprising how arsenal now became ta...	1
2	@CEO786	2024-03-29T03:47:19Z	0	were guaranteed the league and most likely wel...	2
3	@LP-ct6xe	2024-03-29T03:35:47Z	0	hang on a second why do we always have a wome...	3
4	@faisalmohamed7125	2024-03-29T00:47:49Z	0	coyg	4

The comment_text looks better now, so we can continue to create sentiment analysis.

SENTIMENT ANALYSIS

We will user VADER scoring for sentiment analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a sentiment analysis tool that is specifically designed for analyzing the sentiment of text. It works by analyzing the polarity (positive, negative, or neutral) of individual words in a text and then calculates an overall sentiment score for the entire text. It provides a score ranging from -1 to 1, where -1 indicates extremely negative sentiment, 0 indicates neutral sentiment, and 1 indicates extremely positive sentiment.

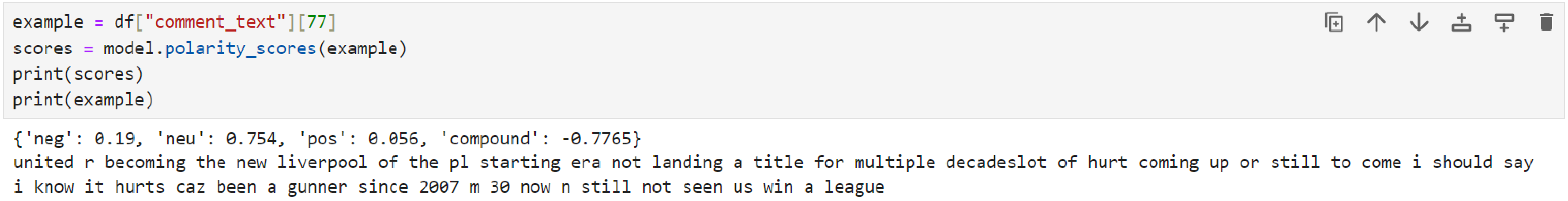
We can use VADER model from NLTK library. NLTK (Natural Language Toolkit) is a Python library specifically designed to work with human language data.



```
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer

model = SentimentIntensityAnalyzer()
```

Now, let's try to perform sentiment analysis on a random comment.



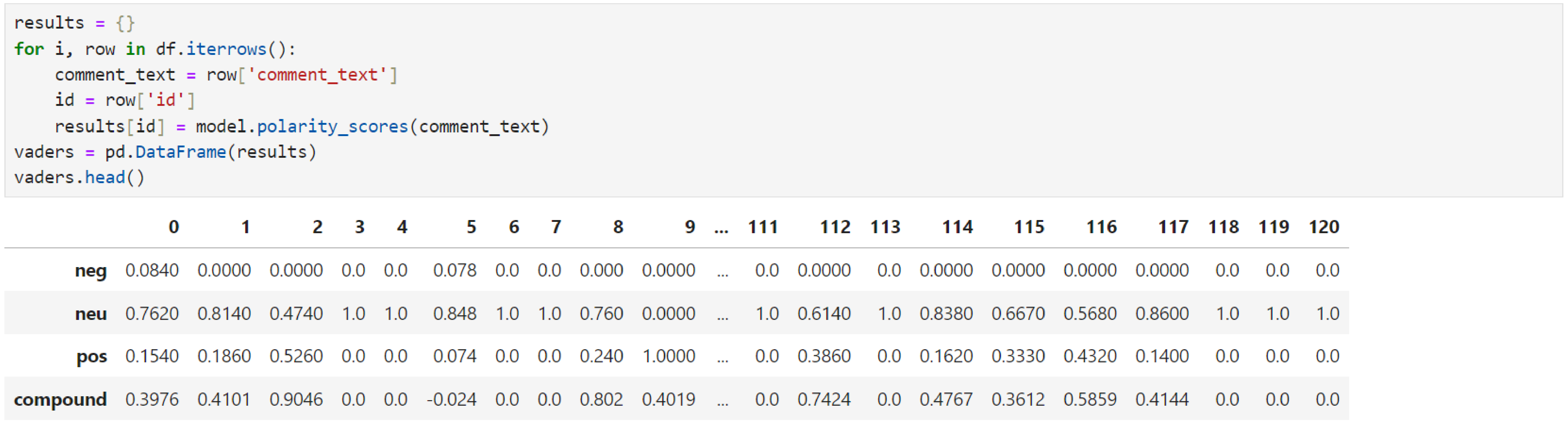
```
example = df["comment_text"][77]
scores = model.polarity_scores(example)
print(scores)
print(example)
```

```
{'neg': 0.19, 'neu': 0.754, 'pos': 0.056, 'compound': -0.7765}
united r becoming the new liverpool of the pl starting era not landing a title for multiple decadeslot of hurt coming up or still to come i should say
i know it hurts caz been a gunner since 2007 m 30 now n still not seen us win a league
```

Based on the results, the comment appears to be negative with a compound score of -0.7765.
If we look at the comment, it seems to be from an Arsenal fan who never seen Arsenal win the league.

SENTIMENT ANALYSIS

Next, we will perform sentiment analysis on all 'comment_text' entries in the dataframe.



```python
results = {}
for i, row in df.iterrows():
    comment_text = row['comment_text']
    id = row['id']
    results[id] = model.polarity_scores(comment_text)
vaders = pd.DataFrame(results)
vaders.head()
```

	0	1	2	3	4	5	6	7	8	9	...	111	112	113	114	115	116	117	118	119	120
neg	0.0840	0.0000	0.0000	0.0	0.0	0.078	0.0	0.0	0.000	0.0000	...	0.0	0.0000	0.0	0.0000	0.0000	0.0000	0.0000	0.0	0.0	0.0
neu	0.7620	0.8140	0.4740	1.0	1.0	0.848	1.0	1.0	0.760	0.0000	...	1.0	0.6140	1.0	0.8380	0.6670	0.5680	0.8600	1.0	1.0	1.0
pos	0.1540	0.1860	0.5260	0.0	0.0	0.074	0.0	0.0	0.240	1.0000	...	0.0	0.3860	0.0	0.1620	0.3330	0.4320	0.1400	0.0	0.0	0.0
compound	0.3976	0.4101	0.9046	0.0	0.0	-0.024	0.0	0.0	0.802	0.4019	...	0.0	0.7424	0.0	0.4767	0.3612	0.5859	0.4144	0.0	0.0	0.0

As we can see from the results, it appears that the dataframe is flipped between the columns and the row. Next, we will fix that.

SENTIMENT ANALYSIS

Fix the dataframe and show first 2 row.

```
vaders = vaders.T
vaders.head(2)
```

	neg	neu	pos	compound
0	0.084	0.762	0.154	0.3976
1	0.000	0.814	0.186	0.4101

and then merge it with original df using id

```
vaders = vaders.reset_index().rename(columns={'index': 'id'})
df = df.merge(vaders, how='left', on='id')
df.head(5)
```

	user_name	comment_at	total_like	comment_text	id	neg	neu	pos	compound
0	@user-yk8sy9if4l	2024-03-29T06:02:37Z	0	i dont think so look top on goal diffrence the...	0	0.084	0.762	0.154	0.3976
1	@Iamsnowhitesamuel	2024-03-29T04:49:13Z	0	it was so surprising how arsenal now became ta...	1	0.000	0.814	0.186	0.4101
2	@CEO786	2024-03-29T03:47:19Z	0	were guaranteed the league and most likely wel...	2	0.000	0.474	0.526	0.9046
3	@LP-ct6xe	2024-03-29T03:35:47Z	0	hang on a second why do we always have a wome...	3	0.000	1.000	0.000	0.0000
4	@faisalmohamed7125	2024-03-29T00:47:49Z	0	coyg	4	0.000	1.000	0.000	0.0000

SENTIMENT ANALYSIS

Now, let’s finish the sentiment analysis by making sentiment class using compound value.

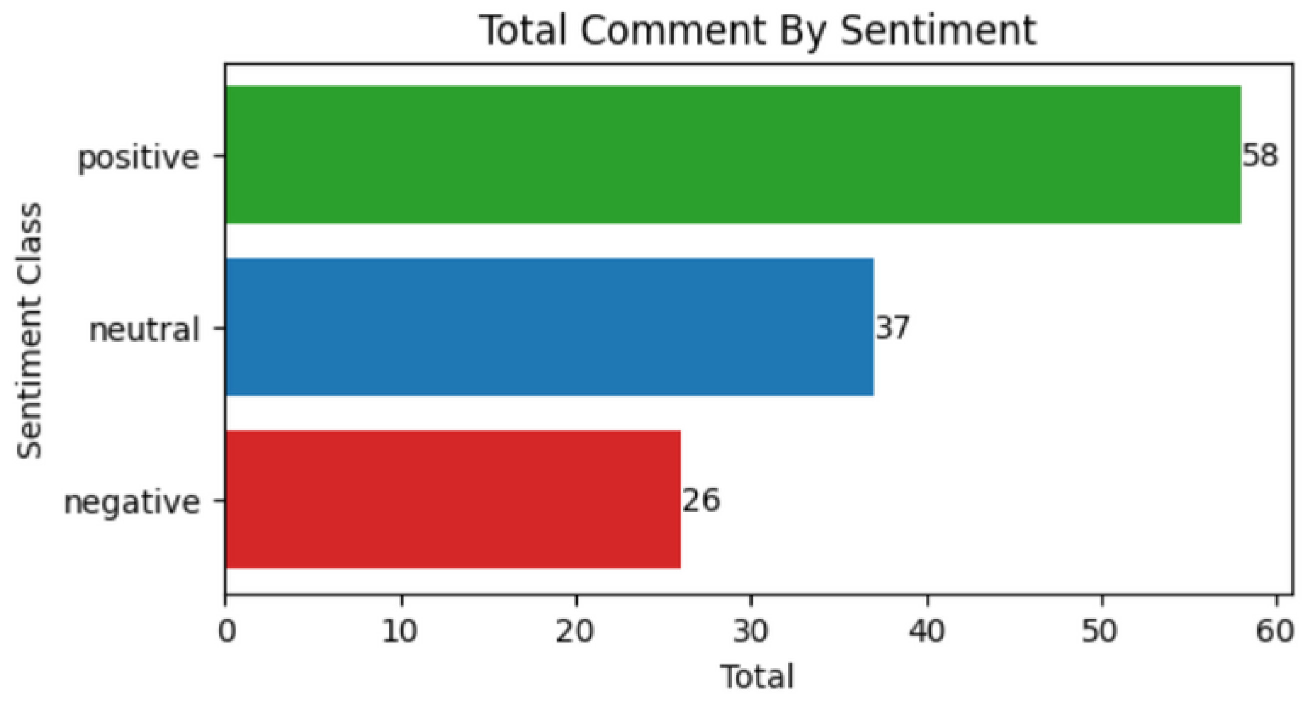
```
df['sentiment_class'] = 'neutral'
df.loc[df['compound'] > 0, 'sentiment_class'] = 'positive'
df.loc[df['compound'] < 0, 'sentiment_class'] = 'negative'
df
```

	user_name	comment_at	total_like	comment_text	id	neg	neu	pos	compound	sentiment_class
0	@user-yk8sy9if4l	2024-03-29T06:02:37Z	0	i dont think so look top on goal diffrence the...	0	0.084	0.762	0.154	0.3976	positive
1	@lamsnowhitesamuel	2024-03-29T04:49:13Z	0	it was so surprising how arsenal now became ta...	1	0.000	0.814	0.186	0.4101	positive
2	@CEO786	2024-03-29T03:47:19Z	0	were guaranteed the league and most likely wel...	2	0.000	0.474	0.526	0.9046	positive
3	@LP-ct6xe	2024-03-29T03:35:47Z	0	hang on a second why do we always have a wome...	3	0.000	1.000	0.000	0.0000	neutral
4	@faisalmohamed7125	2024-03-29T00:47:49Z	0	coyg	4	0.000	1.000	0.000	0.0000	neutral
...	...	...	...	...	...	...	...	...	...	...
116	@LT-mg1vs	2024-03-28T12:37:29Z	71	arsenal will win the league button	116	0.000	0.568	0.432	0.5859	positive
117	@SixtoVerslues	2024-03-28T12:37:21Z	0	the diversity of perspectives woven into this ...	117	0.000	0.860	0.140	0.4144	positive
118	@zsino	2024-03-28T12:37:18Z	0	coyg i believe we can we start with city	118	0.000	1.000	0.000	0.0000	neutral
119	@user-vo8de7sh8c	2024-03-28T12:37:10Z	3	my arsenal	119	0.000	1.000	0.000	0.0000	neutral
120	@Alan-hb2ng	2024-03-28T12:37:04Z	0	first	120	0.000	1.000	0.000	0.0000	neutral

DATA VISUALIZATION

Next, let's create data visualizations using Matplotlib.

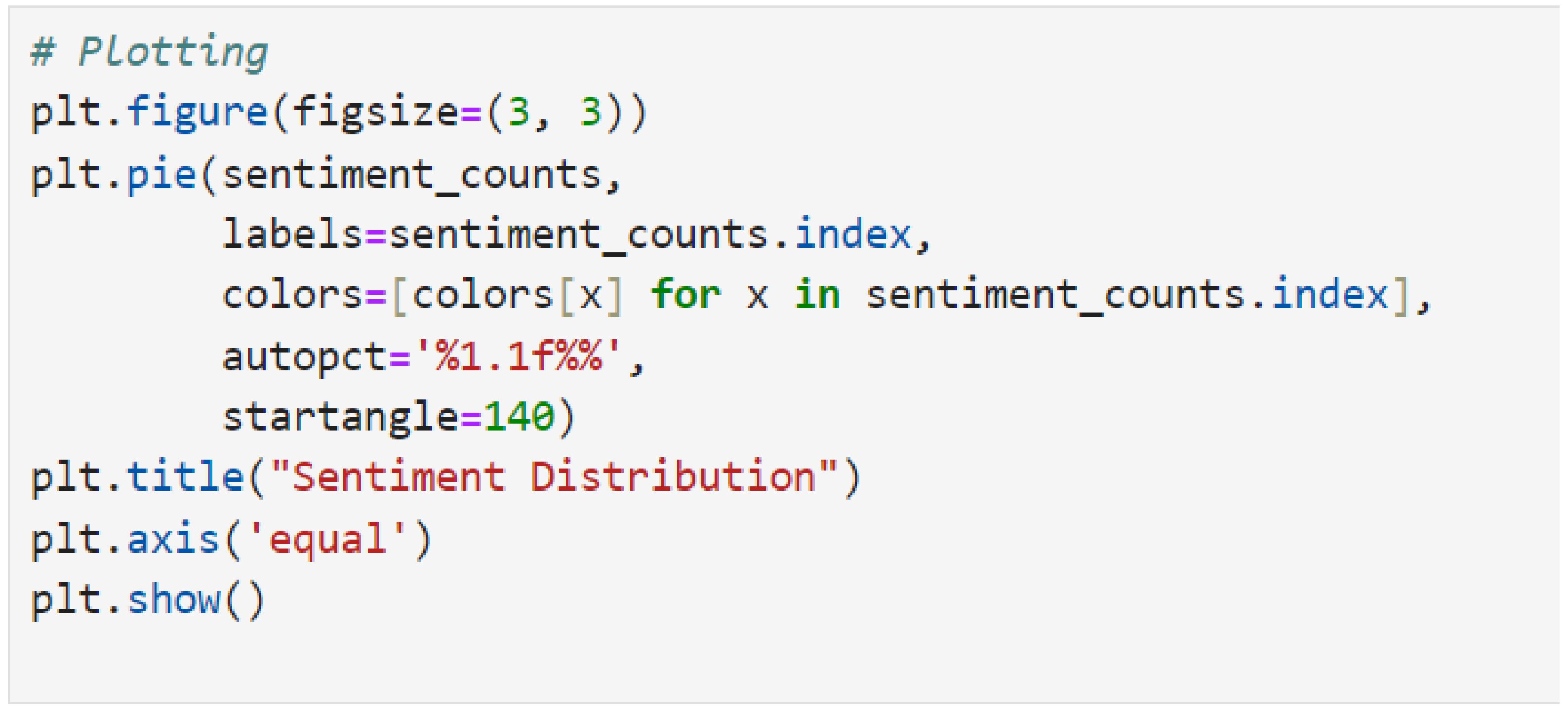
```python
import matplotlib.pyplot as plt
colors = {'positive': 'tab:green', 'negative': 'tab:red', 'neutral': 'tab:blue'}
sentiment_counts = df['sentiment_class'].value_counts().sort_index()
plt.figure(figsize=(6, 3))
for sentiment_class, count in sentiment_counts.items():
    plt.barh(sentiment_class, count, color=colors[sentiment_class])
    plt.text(count, sentiment_class, str(count), va='center')
plt.title("Total Comment By Sentiment")
plt.xlabel('Count')
plt.ylabel('Sentiment Class')
plt.show()
```



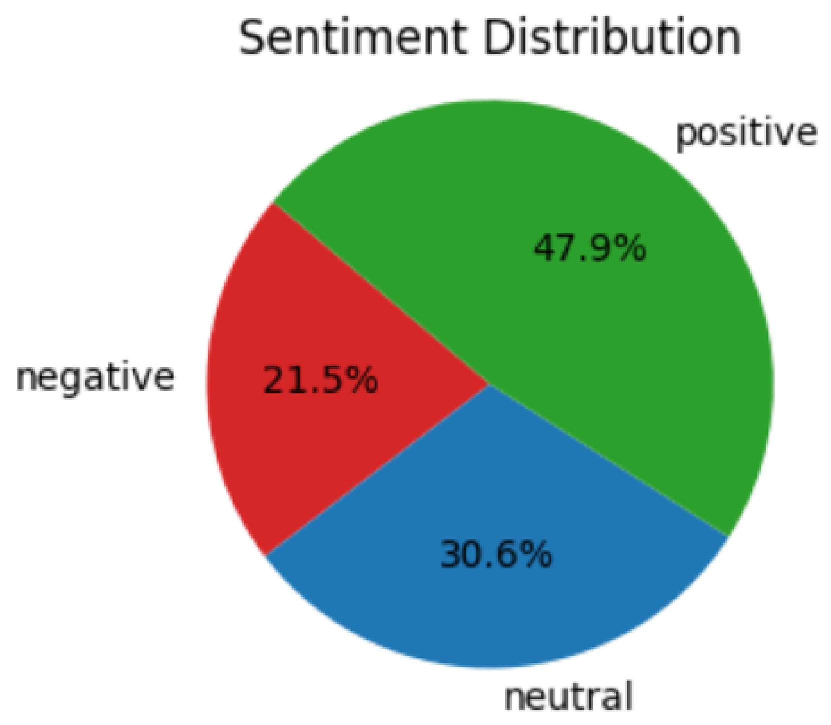
So based on the visualization above, we can see that our video has more positive sentiments with a total of 58 comments, followed by neutral with 37 comments, and negative with 26 comments.

DATA VISUALIZATION

Now, let's create another data visualization with a pie chart to check the percentage of each sentiment.



```
# Plotting
plt.figure(figsize=(3, 3))
plt.pie(sentiment_counts,
        labels=sentiment_counts.index,
        colors=[colors[x] for x in sentiment_counts.index],
        autopct='%1.1f%%',
        startangle=140)
plt.title("Sentiment Distribution")
plt.axis('equal')
plt.show()
```



We can see that we have 47.9% positive sentiment, followed by 30.6% neutral sentiment, and 21.5% negative sentiment.

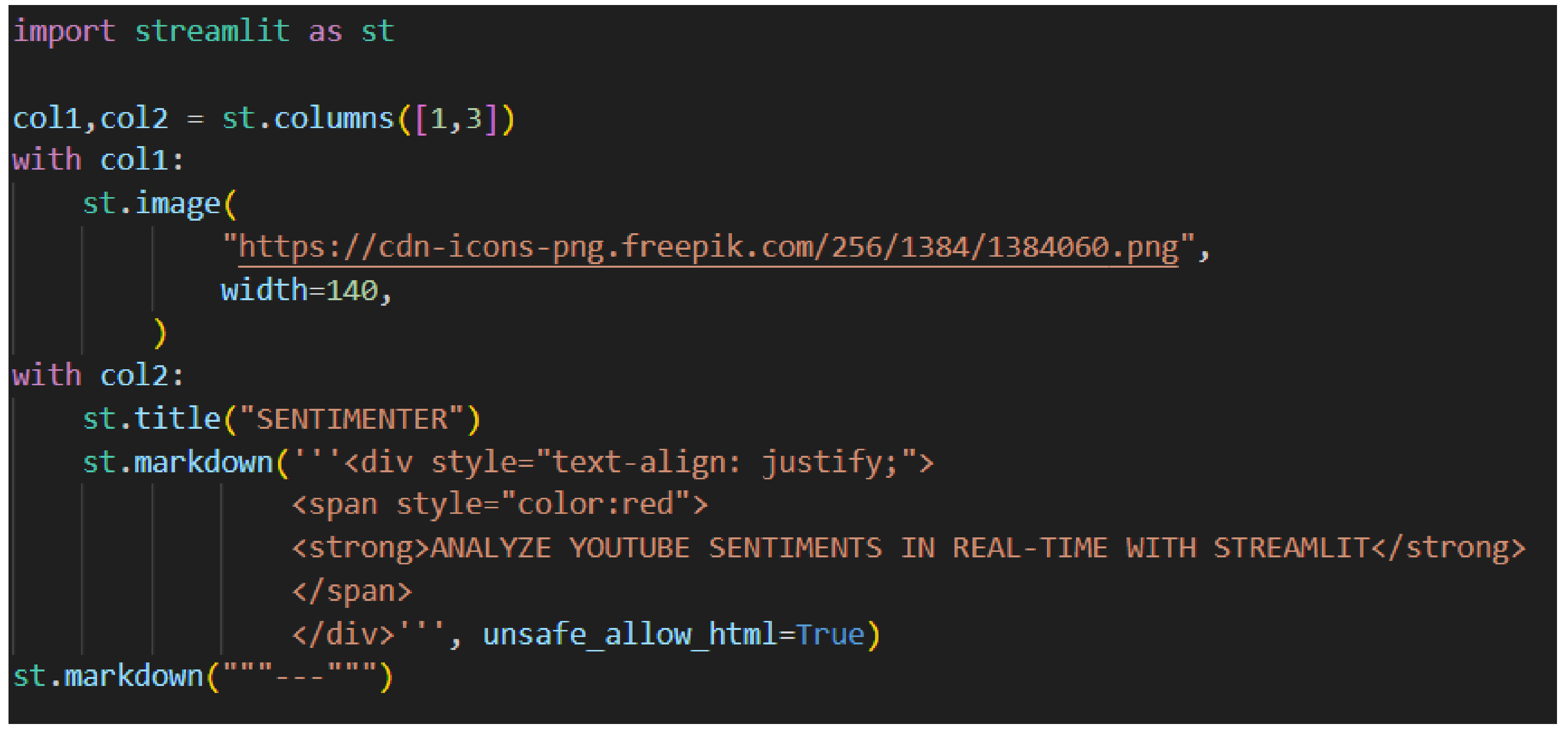
So that are the results of the sentiment analysis for the YouTube video from SKY Sport Premier League titled "Can Arsenal go all the way this season? 🔴🏆".

- 47.9% positive,
- 30.6% neutral and,
- 21.5% negative

BUILD WEB APP

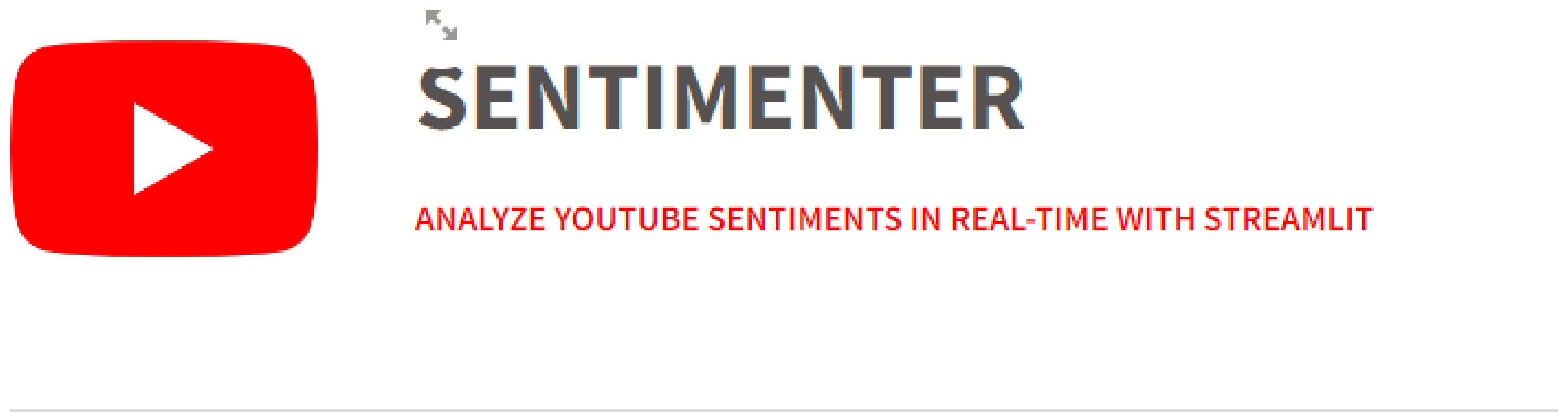
Now, let's build a web app for this project using Streamlit.

First, I'll show the YouTube logo using st.image and create a title using st.title and st.markdown. This logo and title are placed in separate columns using st.columns with a ratio of 1:3.



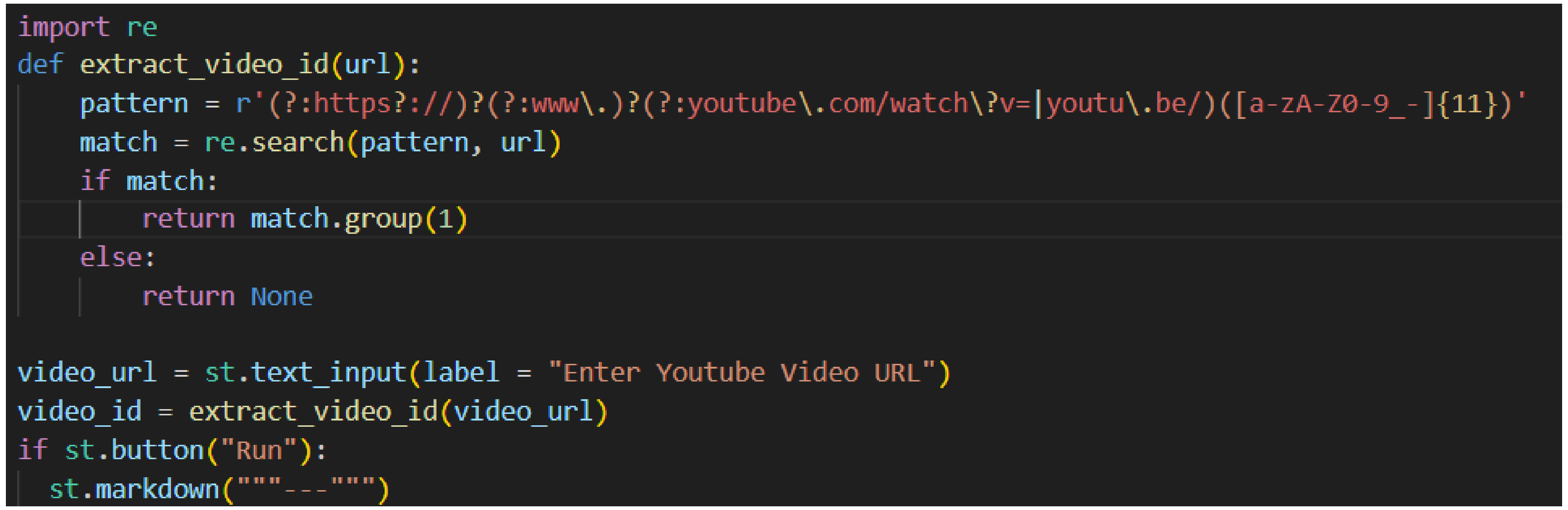
```
import streamlit as st

col1,col2 = st.columns([1,3])
with col1:
    st.image(
            "https://cdn-icons-png.freepik.com/256/1384/1384060.png",
            width=140,
        )
with col2:
    st.title("SENTIMENTER")
    st.markdown('''<div style="text-align: justify;">
                <span style="color:red">
                <strong>ANALYZE YOUTUBE SENTIMENTS IN REAL-TIME WITH STREAMLIT</strong>
                </span>
                </div>''', unsafe_allow_html=True)
st.markdown("""---""")
```



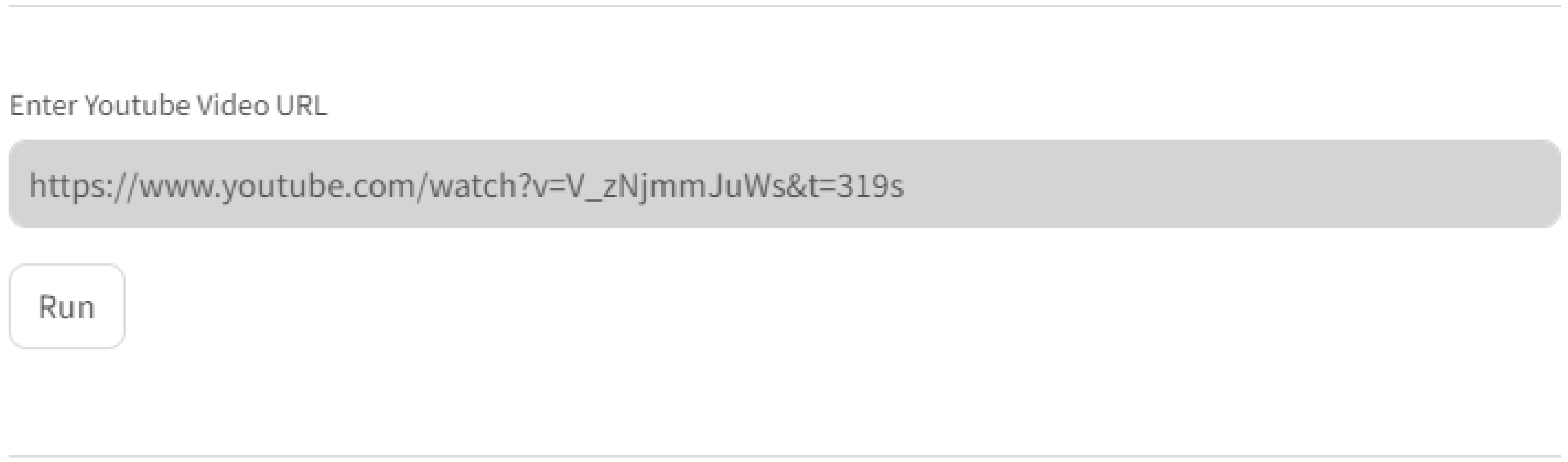
BUILD WEB APP

Next, let’s create a form for users to input the YouTube video URL using st.text_input and add a run button with st.button to initiate the process. We also need to create a function to extract the video ID from the URL, as it is required for the next process.



```python
import re
def extract_video_id(url):
    pattern = r'(?:https?://)?(?:www\.)?(?:youtube\.com/watch\?v=|youtu\.be/)([a-zA-Z0-9_-]{11})'
    match = re.search(pattern, url)
    if match:
        return match.group(1)
    else:
        return None

video_url = st.text_input(label = "Enter Youtube Video URL")
video_id = extract_video_id(video_url)
if st.button("Run"):
  st.markdown("""---""")
```

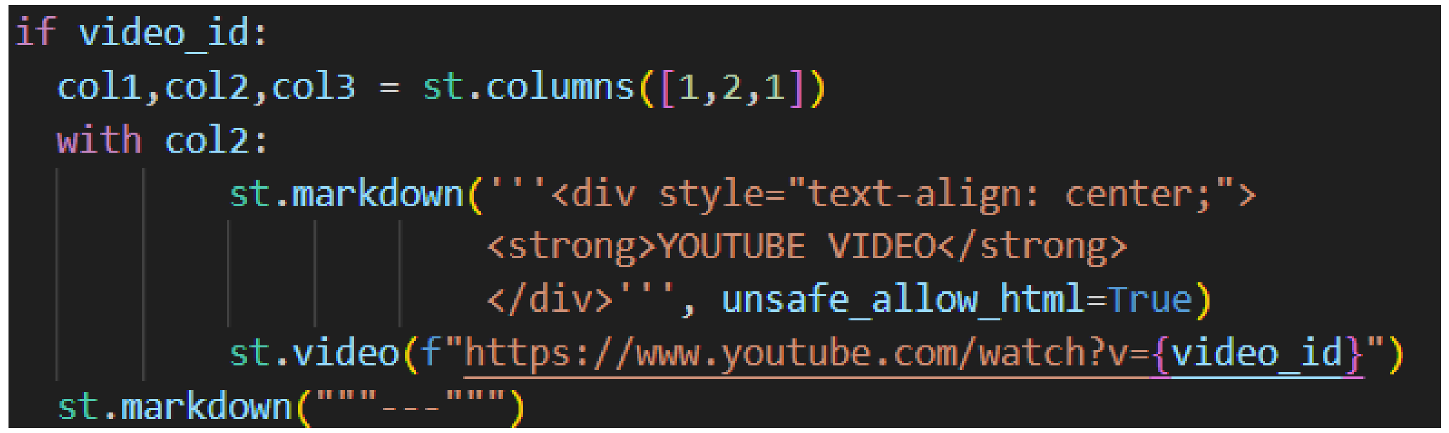


Enter Youtube Video URL

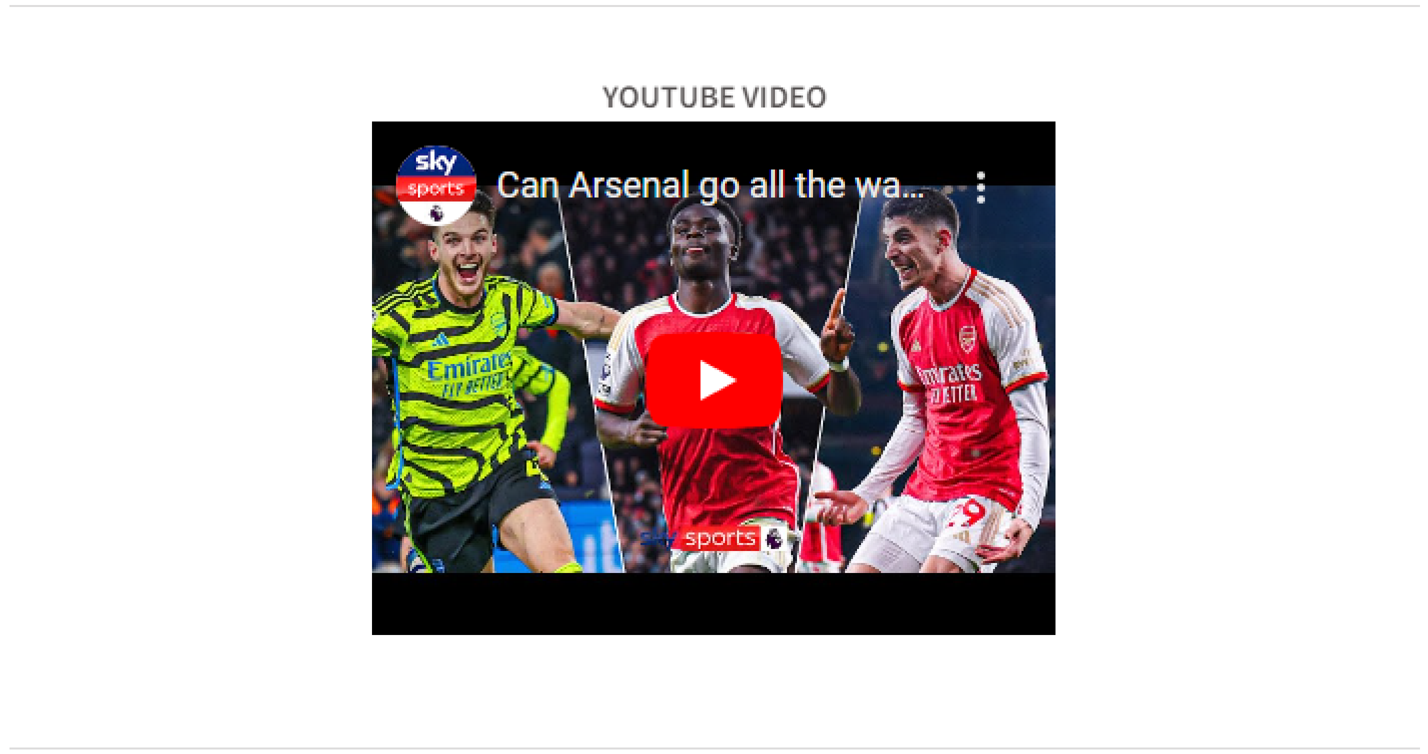
https://www.youtube.com/watch?v=V_zNjmmJuWs&t=319s

BUILD WEB APP

After users press the 'Run' button, we will display the YouTube video using st.video. I divide the page into three columns with a ratio of 1:2:1 using st.columns to make the video is centered by placing it in the second column (col2).

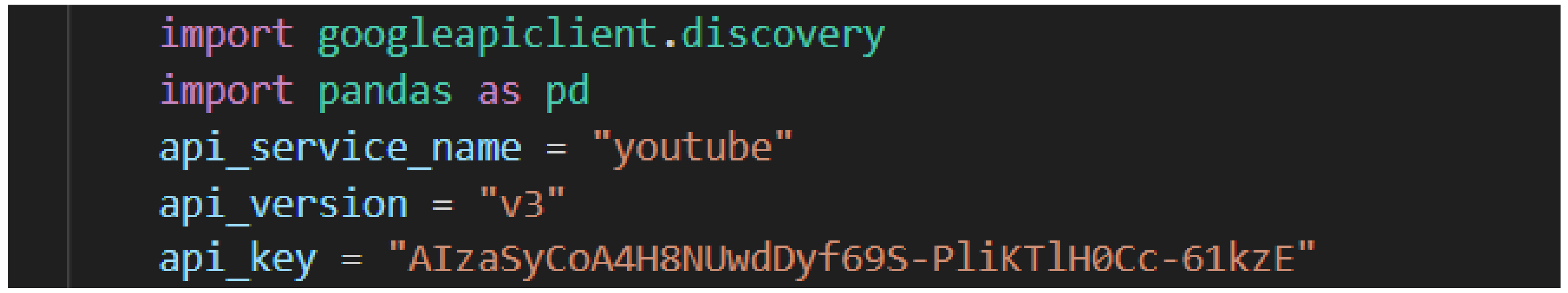


```
if video_id:
  col1,col2,col3 = st.columns([1,2,1])
  with col2:
          st.markdown('''<div style="text-align: center;">
                      <strong>YOUTUBE VIDEO</strong>
                      </div>''', unsafe_allow_html=True)
          st.video(f"https://www.youtube.com/watch?v={video_id}")
  st.markdown("""---""")
```



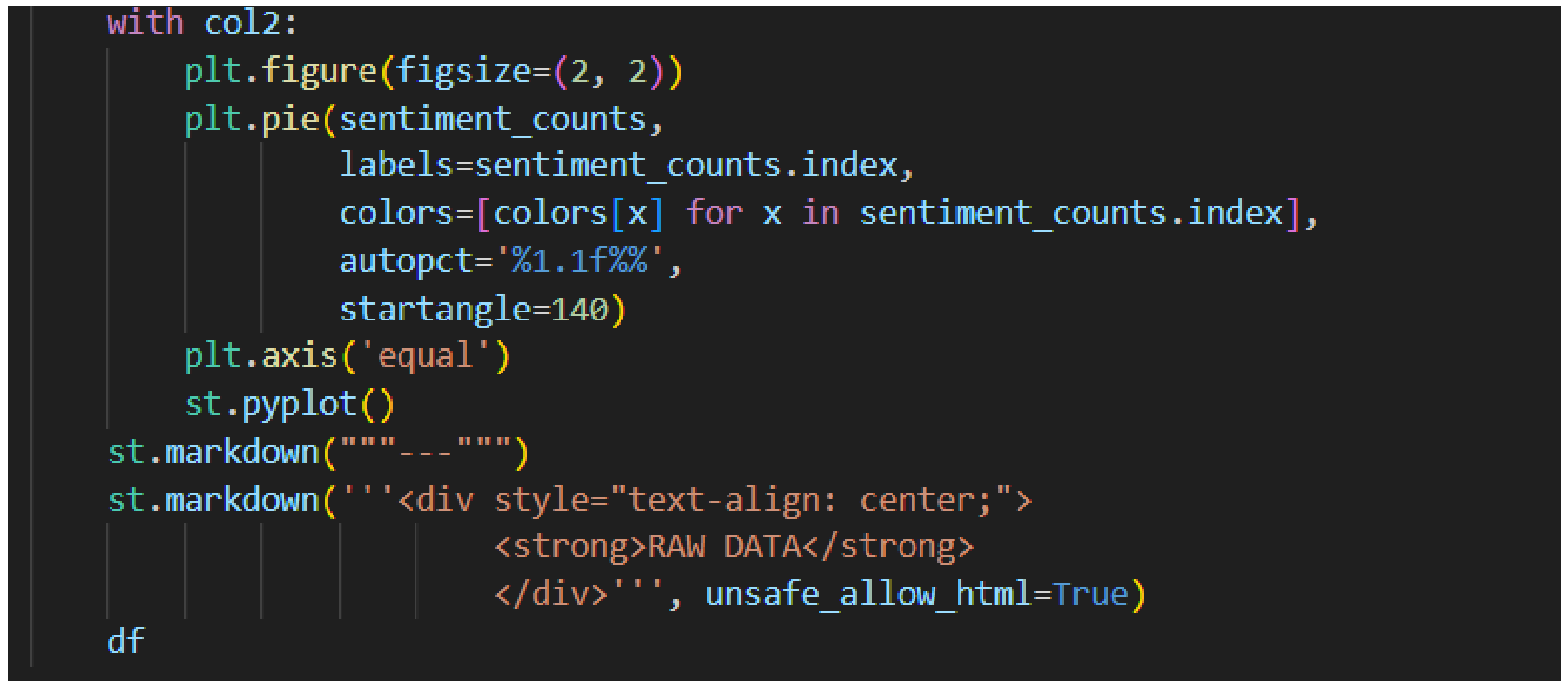
BUILD WEB APP

Finally, we will display some data visualizations similar to the previous process and present a table containing the raw data resulting from sentiment analysis. The application is ready to be deployed, and you can find the full code on my GitHub page.



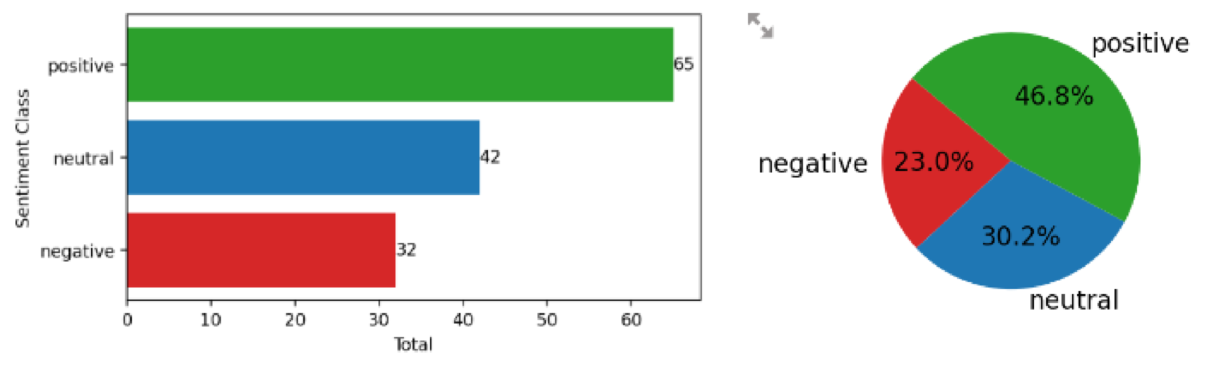
```
    import googleapiclient.discovery
    import pandas as pd
    api_service_name = "youtube"
    api_version = "v3"
    api_key = "AIzaSyCoA4H8NUwdDyf69S-PliKTlH0Cc-61kzE"
```

● ● ●

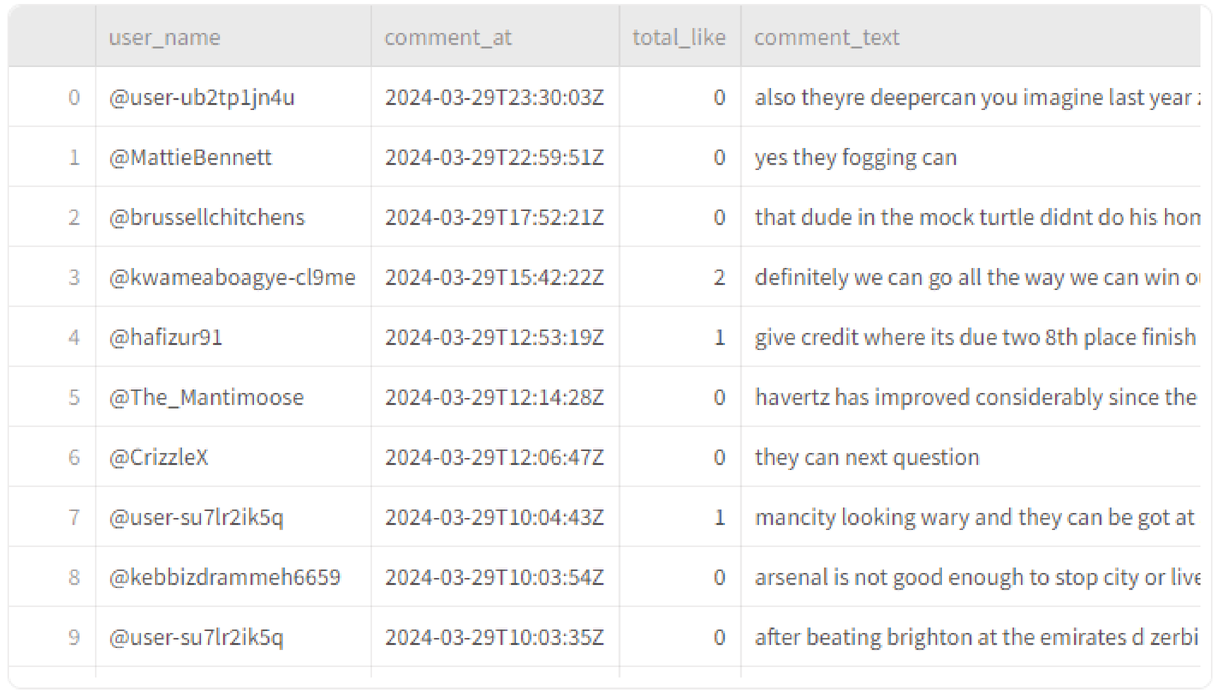


```
with col2:
    plt.figure(figsize=(2, 2))
    plt.pie(sentiment_counts,
            labels=sentiment_counts.index,
            colors=[colors[x] for x in sentiment_counts.index],
            autopct='%1.1f%%',
            startangle=140)
    plt.axis('equal')
    st.pyplot()
st.markdown("""---""")
st.markdown('''<div style="text-align: center;">
                <strong>RAW DATA</strong>
                </div>''', unsafe_allow_html=True)
df
```

DATA VISUALIZATION



RAW DATA



	user_name	comment_at	total_like	comment_text
0	@user-ub2tp1jn4u	2024-03-29T23:30:03Z	0	also theyre deepercan you imagine last year
1	@MattieBennett	2024-03-29T22:59:51Z	0	yes they fogging can
2	@brussellchitchens	2024-03-29T17:52:21Z	0	that dude in the mock turtle didnt do his hon
3	@kwameaboagye-cl9me	2024-03-29T15:42:22Z	2	definitely we can go all the way we can win o
4	@hafizur91	2024-03-29T12:53:19Z	1	give credit where its due two 8th place finish
5	@The_Mantimoose	2024-03-29T12:14:28Z	0	havertz has improved considerably since the
6	@CrizzleX	2024-03-29T12:06:47Z	0	they can next question
7	@user-su7lr2ik5q	2024-03-29T10:04:43Z	1	mancity looking wary and they can be got at
8	@kebbizdrammeh6659	2024-03-29T10:03:54Z	0	arsenal is not good enough to stop city or live
9	@user-su7lr2ik5q	2024-03-29T10:03:35Z	0	after beating brighton at the emirates d zerbi

Thank you for taking the time to read through!

Feel free to access the web app by clicking here.
If you're interested in exploring the full code, you can find it on my GitHub profile.
Let's connect on LinkedIn as well!