Social Media Analytics Lab (CDP-042)

LAB FILE

B. Tech: 3rd YEAR
SEMESTER: 6th

SESSION: 2024-25

Submitted By:
Priyansh Singhal
(2022529691)
SECTION: M

Submitted To:
Dr. Avinash Kumar Sharma
(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING SHARDA SCHOOL OF ENGINEERING & TECHNOLOGY

SHARDA UNIVERSITY, GREATER NOIDA

INDEX

S.NO	AIM	DATE	REMARKS

Experiment No.1

AIM- Create a blog and link it with Google Analytics using R

Objective: -

- To authenticate and retrieve data from Google Analytics using the googleAnalyticsR package.
- To perform basic analysis and visualize the data, including metrics like active users, sessions, and page views.

Steps:

1. Download and Install R

R is a system for statistical computation and graphics. It provides, among other things, a programming language, high level graphics, interfaces to other languages and debugging facilities.

2. Download and Install R Studio

R Studio is where the magic happens. It is an IDE (Interactive Development Environment). This is the software user interface that you'll be working in and where you'll run your scripts.

3. Save the R Script

Launch R Studio and in the top menu go to File > New File > New R Script. This will open a blank window in the top left pane of R Studio. Copy and paste the code below into the blank window. Save the R Script.

4. Install and Load Required Packages

```
install.packages("googleAuthR")
install.packages("googleAnalyticsR")
install.packages("gargle")
install.packages("ggplot2")
library(ggplot2)
library(gargle)
library(googleAuthR)
library(googleAnalyticsR)
```

5. Authenticate Google Analytics Account

ga_auth(email="singhalpriyansh2005@gmail.com")
ga_account_list("ga4")

```
# A tibble: 1 × 4
  account_name    accountId property_name propertyId
  <chr>           <chr>     <chr>         <chr>
1 Priyansh Singhal 342047172 Sharda        473961637
```

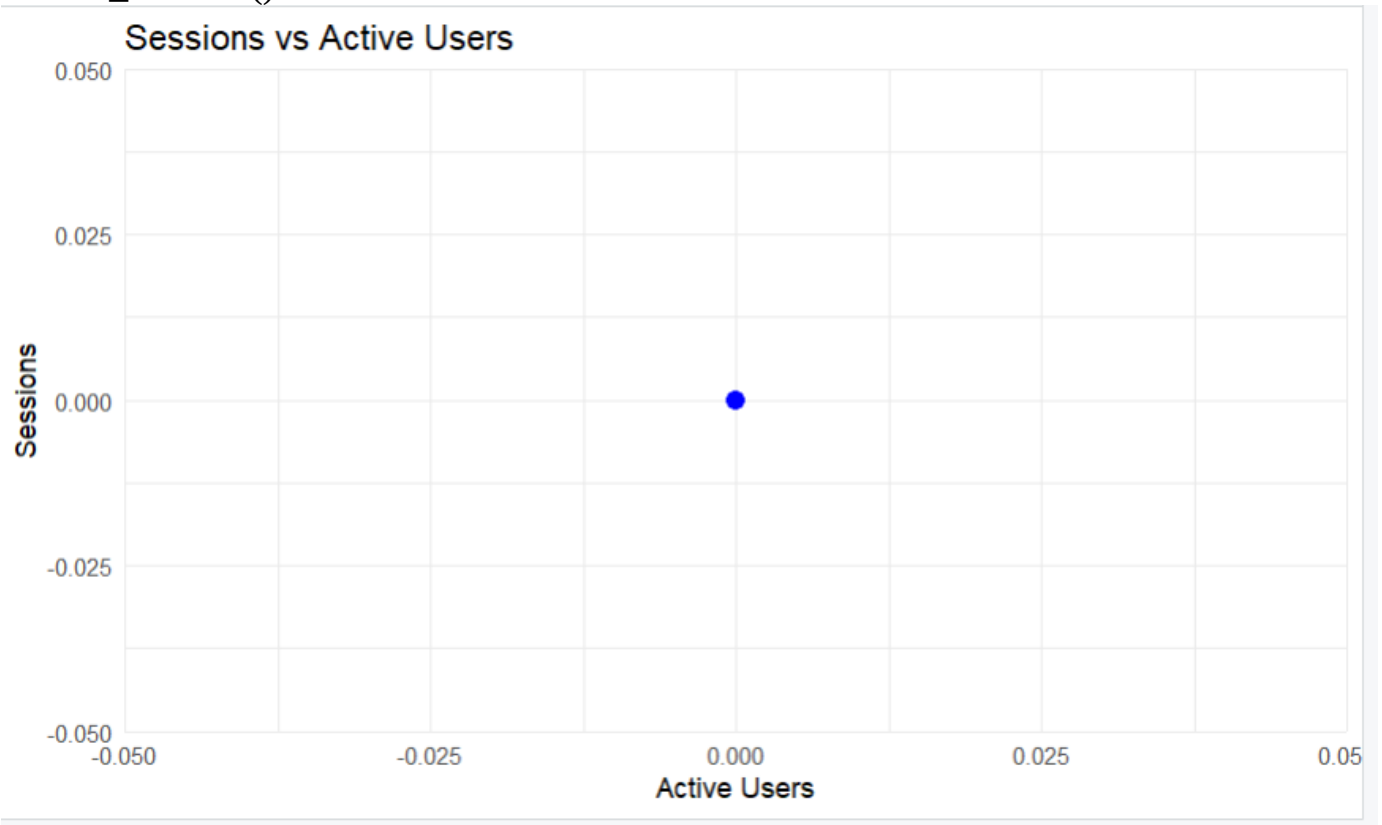
6. Define the Property ID, Date Range and Fetch Google Analytics Data

```
my_property_id <- 473961637
from_date <- "2025-01-21"
to_date <- "2025-01-27"
overall <- ga_data(
my_property_id,
  metrics = c("activeUsers", "newUsers","sessions", "screenPageViews"),
  date_range = c(from_date, to_date)
)
print(overall)
```

```
A tibble: 1 × 4
activeUsers newUsers sessions screenPageViews
      <dbl>    <dbl>    <dbl>           <dbl>
          0        0        0               0
```

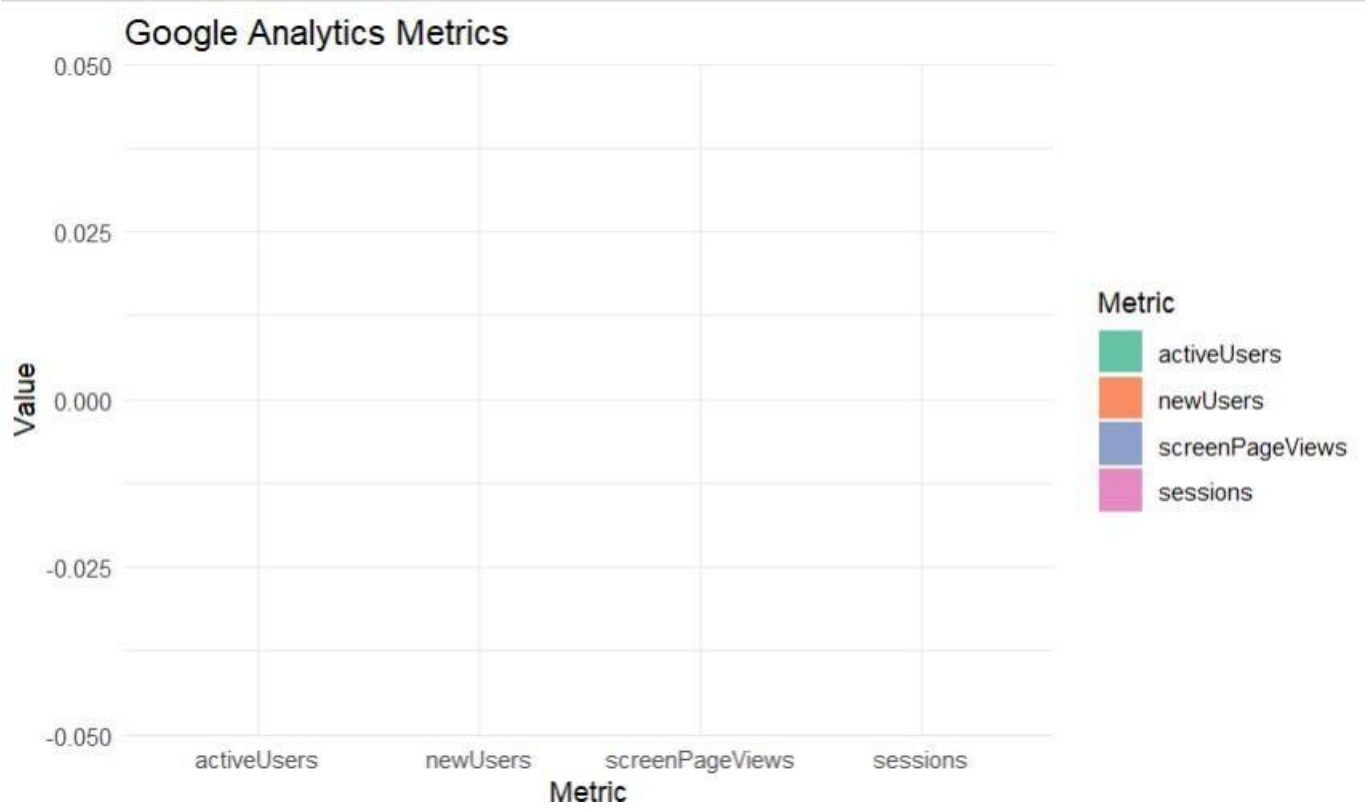
7. Visualize the Relationship Between Active Users and Sessions

```
ggplot(overall, aes(x = activeUsers, y = sessions)) +
  geom_point(color = "blue", size = 3) +
  labs(
    title = "Sessions vs Active Users",
    x = "Active Users",
    y = "Sessions"
  ) +
  theme_minimal()
```



8. Create a Bar Chart for Metrics

```
ggplot(overall_long, aes(x = Metric, y = Value, fill = Metric)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Google Analytics Metrics",
    x = "Metric",
    y = "Value"
  ) +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```



Experiment No. 2

Objective: Facebook metric analysis using Facebook Insights.

Facebook Page Insights vs. Facebook Audience Insights

Facebook Page Insights and Facebook Audience Insights are both Facebook analytics tools. There is a little bit of overlap between the two of them, but they serve very different purposes. The key to how to use each is in the name of the tool:

● Facebook Page Insights gives you detailed analytics for your *Facebook Page*, so you can track what works, learn how people interact with your content, and improve your results over time.

● Facebook Audience Insights helps you understand your *Facebook audience* so you can better target ads and create more relevant content.

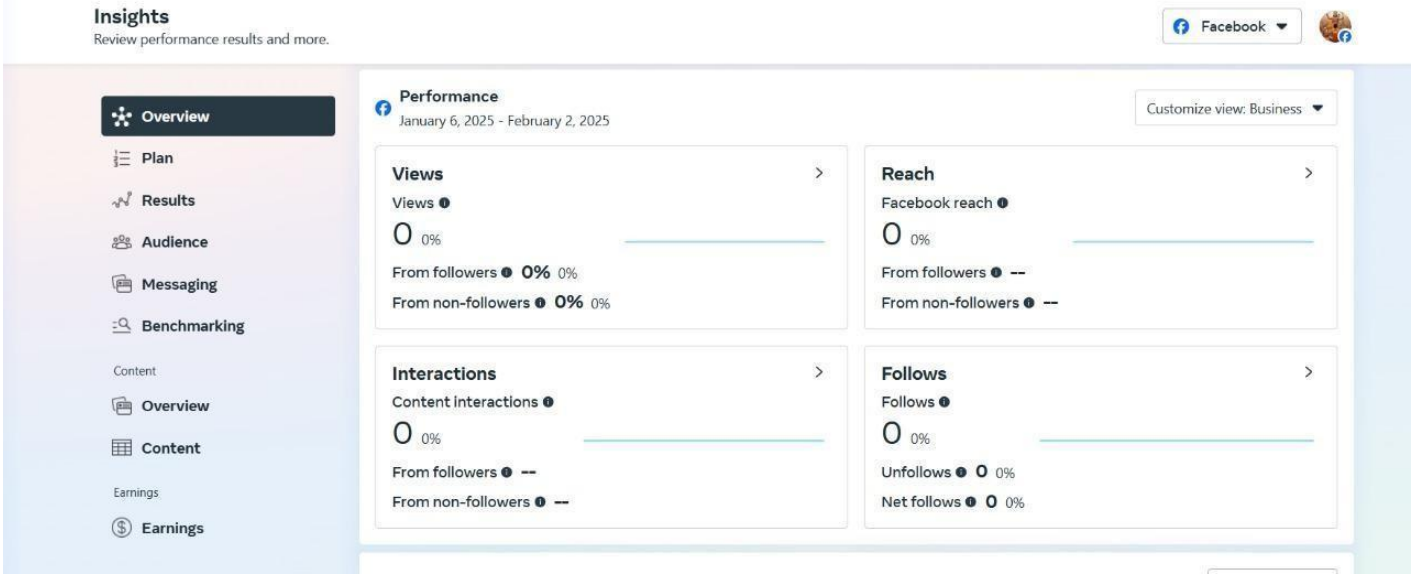
In this post, we walk you through how to use Facebook Page Insights. Looking for information on how to track and analyze your Facebook audience? We've got a whole blog post about how to set up and use Facebook Audience Insights to get you started.

How to use Facebook analytics

First things first. To access Facebook Page Insights, go to your Facebook Page and click Insights in the top menu. If you don't see Insights in the menu, click More to bring it up.

You'll be taken straight to your Overview, which you can access again at any time by clicking Overview in the left-hand menu.

Overview: The big picture

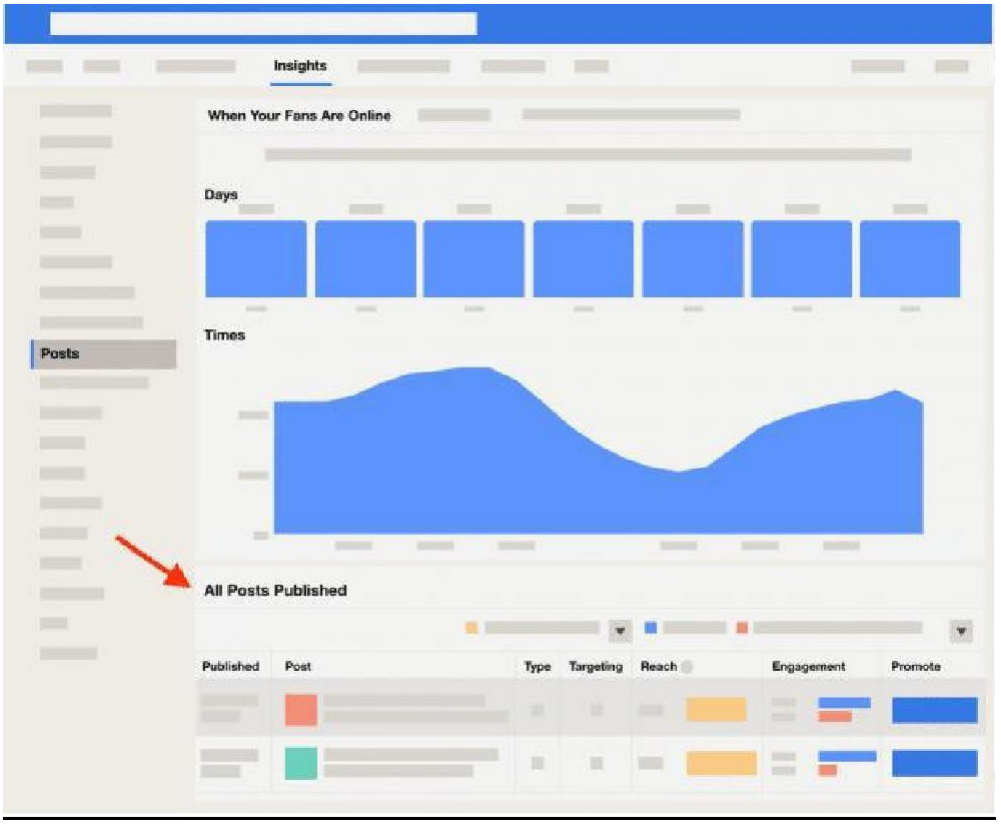


The Overview gives you a bird's-eye view of everything that's happening with your Facebook Page. You can choose to view data for the last seven or 28 days, for the current day, or for the previous day.
The Overview is broken down into three sections, starting with the Page Summary. Here, you'll see a set of graphs with top-level metrics for a number of categories:
● Actions on Page: The combined total clicks for your contact information and call-to-action button
● Page views: Total views of your Facebook Page, including by people not logged into Facebook.
● Page Previews: The number of times people hovered their mouse over your Page information to see a preview of your Page.

● Page Likes: The number of new likes.
● Post reach: The number of people who saw your posts in their timeline.
● Story reach: The number of people who saw your Stories.
● Recommendations: The number of people who recommended your Page.
● Post engagement: A combined total of post likes, comments, shares, and other engagements.
● Responsiveness: An evaluation of how often and how fast you respond to messages.
● Videos: The number of video views of three seconds of more.
● Page followers: The number of new followers
● Orders: Your orders and earnings.

You can click on any of these charts to get more detailed information, or click on the corresponding item in the left-hand menu.
The second section is called Your 5 Most Recent Posts. This gives you the reach and engagement numbers for your latest posts. There's also an option to boost your best performing posts right from this screen.
Finally, you'll see a section called Pages to Watch. In this section, you can manually add in Pages you want to compare to your own. For example, you could add a competitor's page to see how you measure up. This can be a good way to benchmark your results against the major players in your industry.
Directly under the page summary, you'll see a box with the heading Ad Results Have Moved. All tracking for Facebook ads has moved to a separate ad analytics page called Ad Center. You can find it in the top navigation menu of your Facebook Page by clicking More.
Posts: Detailed analysis
The Posts section of your Facebook page analytics dashboard gives you tons of important information about your posts and the activity on your Page, divided into three tabs:
● When Your Fans Are Online
● Post Types
● Top Posts from Pages You Watch

It’s a bit confusing, because only the information in the box at the top of the screen changes when you click on one of these tabs. The information in the section below this top box, called All Posts Published, stays the same.



We’ll start with the All-Posts Published section before moving on to what you can learn from the individual tabs at the top of the page.

All posts published

Here, you can review the results for all the posts you’ve published on your Page. For each post, you can see:

● Type: Was it a link post? A photo post? A video? This column can help you understand which types of posts appeal most to your audience.

● Targeting: Was it a public post? Did you target a specific audience? When comparing results, make sure you take the targeted group into account.

● Reach: By default, this column shows the number of people who say your post. To see the breakdown of paid versus organic reach, hover your mouse over the yellow bar. At the top of the All Posts Published section, there’s a drop-down menu that you can use to change what’s displayed in this column. Your other options are to view impressions (organic vs. paid) or reach among fans vs. non-fans.

● Engagement: Here you see the number of clicks each post got, as well as the combined number of reactions, comments, and shares.

There’s also an option to boost your posts right from this screen.

Experiment No. 8

Objective: Link a website to Google Analytics and Use Web browser to Analyse the statistics of a web site using Google Analytics

Google Analytics is a free website analytics dashboard that provides a wealth of insights about your website and its visitors, including those who find you through social media. For instance, you can track:

• Total traffic to your site and traffic sources (including social networks)

• Individual page traffic

• Number of leads converted and where those leads come from

• Whether your traffic comes from mobile or desktop.

Step 1: Create a Google Analytics account

1. Create a Google Analytics account by clicking the Start measuring button to sign up on the GA page.

2. Enter your account name and choose your data sharing settings. When you're ready, click Next.

3. This is where you have to pay attention to get the Universal Analytics tracking code. Under Property name, enter the name of your website or business (not your URL). Choose your time zone and currency. Then, click Show advanced options.

4. Switch the toggle on for Create a Universal Analytics property. Enter your website URL. Leave the radio button selected for Create both a Google Analytics 4 and a Universal Analytics property. Double-check the settings, then click Next.

5. On the next screen, you can enter information about your business, but you don't have to. Once you've entered as much detail as you'd like, click Create, then accept the Terms of Service Agreement in the pop-up box.

6. In the bottom left corner of the Google Analytics dashboard, click Admin. Select the account and property you're looking for. In the Property column, click Tracking Info.

7. Click Tracking code to get your tracking ID.

Step 2: Set up Google Tag Manager Google

Tag Manager allows you to send data to Google Analytics without coding knowledge.

1. Create an account on the Google Tag Manager dashboard. Choose a good account name, the country your business is in, and whether or not you want to share your data with Google to enable benchmarking.

2. Scroll down to the Container Setup section. A container holds all the macros, rules, and tags needed to track data for your website. Enter a name you'd like for your container and choose Web as your Target platform, then click Create. Review the Terms of Service in the pop-up and click Yes.

3. Copy and paste the code from the Install Google Tag Manager pop-up box onto your website. The first snippet goes in the section of your page, and the second in the

section. The code has to go on every page of your website, so it's best if you can add it to the templates of your content management system (CMS). 4. Once you've added the code to your website, return to the Tag Manager workspace and click Submit on the top right of the screen.

Step 3: Create your analytics tags

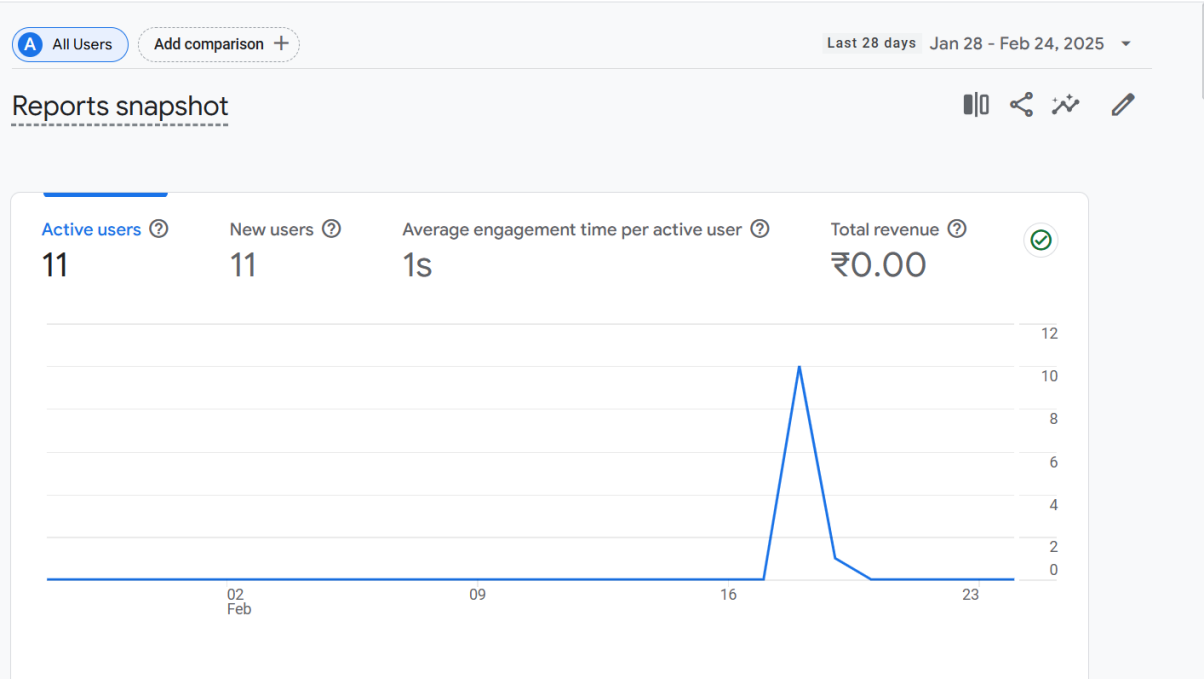
Now it's time to merge Google Tag Manager with Google Analytics.

1. Go to your Google Tag Manager workspace and click Add a new tag. There are two areas of the tag you'll be able to customize: • Configuration. Where the data collected by the tag will go. • Triggering. What type of data you want to collect.

2. Click Tag Configuration and choose Google Analytics: Universal Analytics.

3. Choose the type of data you want to track and then choose New Variable… from the dropdown menu under Google Analytics Settings.

4. Head back to the Triggering section to select the data you want to send to Google Analytics. Select All Pages to send data from all your web pages, then click Add. Click Save and voila! You have a new Google Tag tracking and sending data to Google Analytics.



Experiment 6

Objective: Develop a dashboard and reporting tool based on real time social media data.

A social media dashboard monitors your social media performance metrics like engagement, subscriber or follower count, and audience insights. Social media dashboards bring together metrics from platforms like Facebook, Twitter, and YouTube to display your social media marketing performance in a single view. When you track your metrics on a social media dashboard, you have quick access to insights that will help you make smart, data-driven marketing decisions.

Use Of social media dashboard A social media monitoring dashboard displays all of your metrics in a single view. Use your social media metrics to shape your marketing strategy, engage with your audience, increase your conversion rates, and generate revenue. Social media dashboards allow you to gain insight with a single glance and share your performance with your team so you can stay on top of your social media strategies.

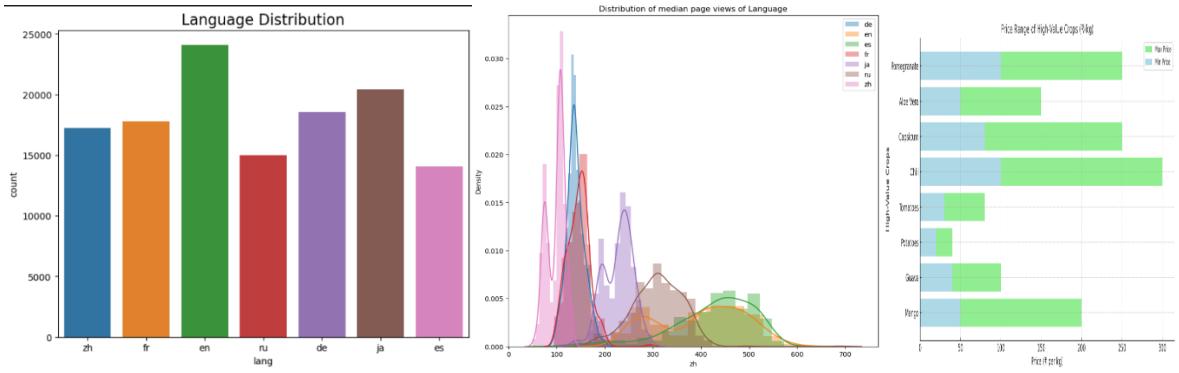
How do I create a social media dashboard?

Before you start building a social media dashboard, ask yourself these questions.

- Who is the audience for the marketing dashboard?
- What information do they need?
- What do they already know about the marketing metrics?
- What is their level of experience with marketing data?

These questions will help you narrow the scope of your dashboard and identify the top social media marketing metrics to track. A well-designed social media dashboard will communicate a clear message, so create your dashboard based on your intended audience. From there, choose the right visualizations and ensure your dashboard is easy to read.

Dashboard

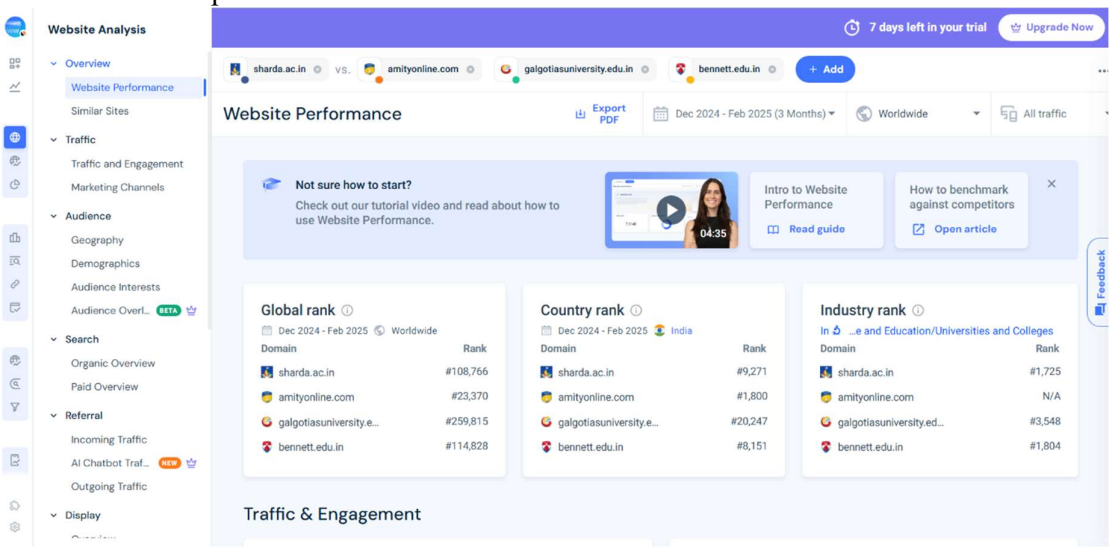


EXPERIMENT- 7

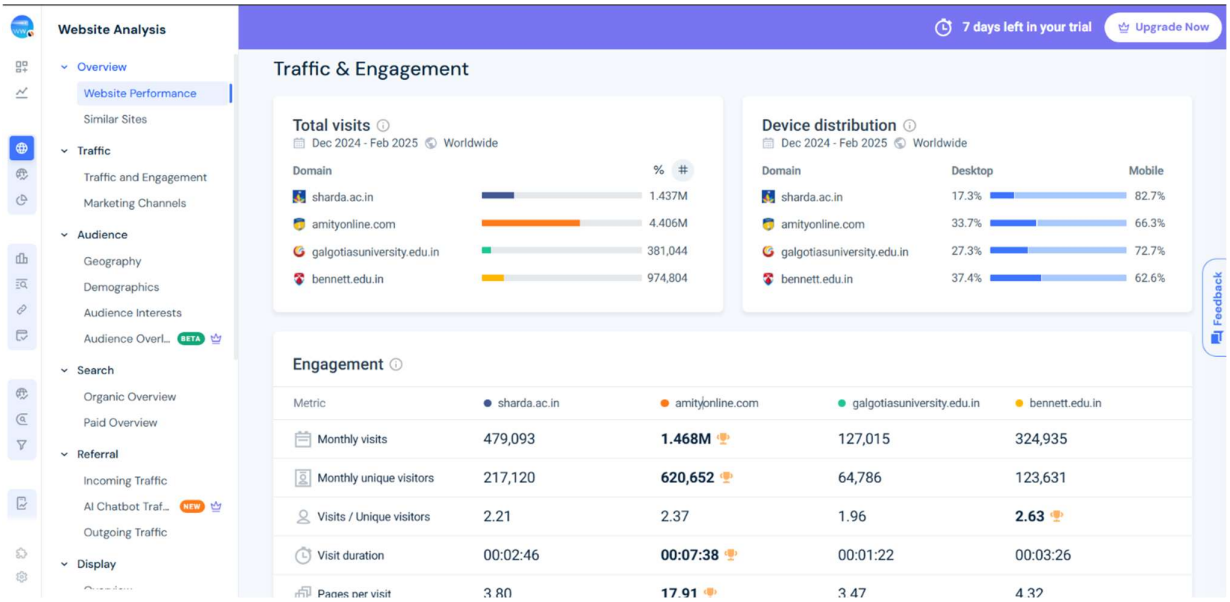
AIM: Analyze competitor activities using social media data.

THEORY:

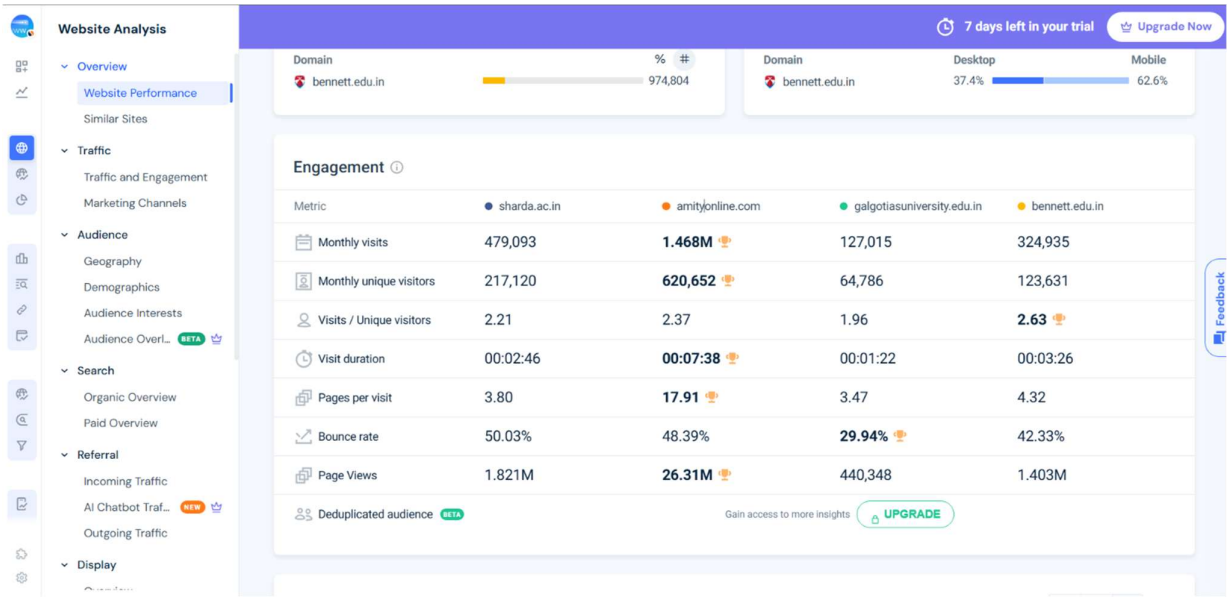
1. Open a web browser and navigate to the SimilarWeb website.
2. Enter the competitor's website URL into the search bar.



3. Analyze the Traffic Overview section to observe total visits and traffic trends over the past six months.

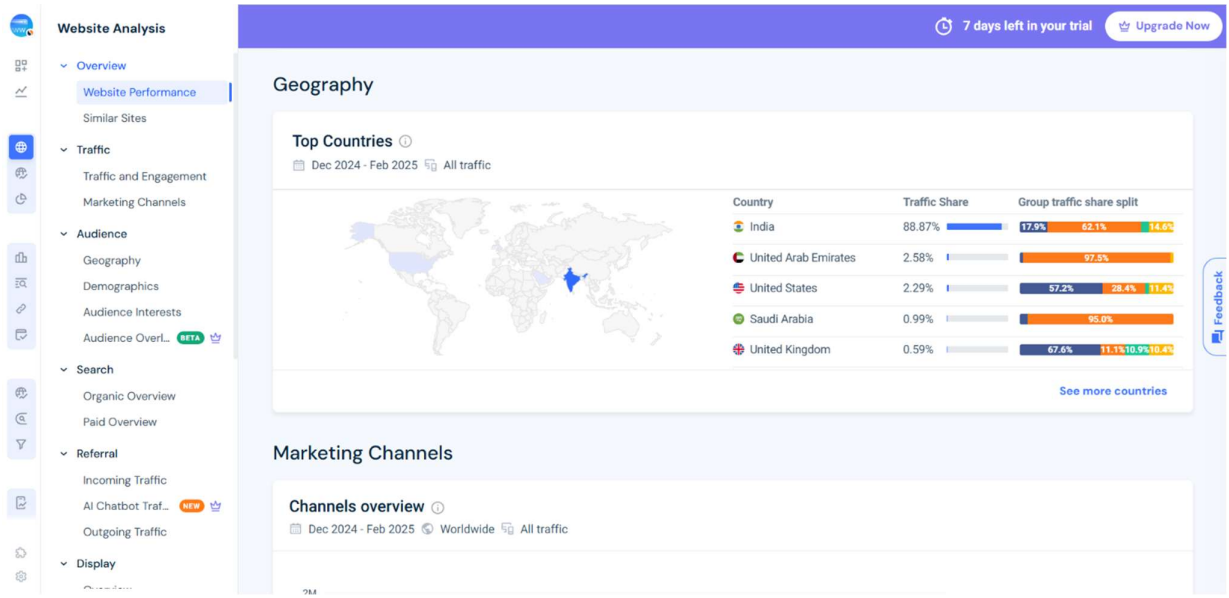


4. Examine the Engagement Metrics to evaluate factors such as average visit duration, page views per visit, and bounce rate.

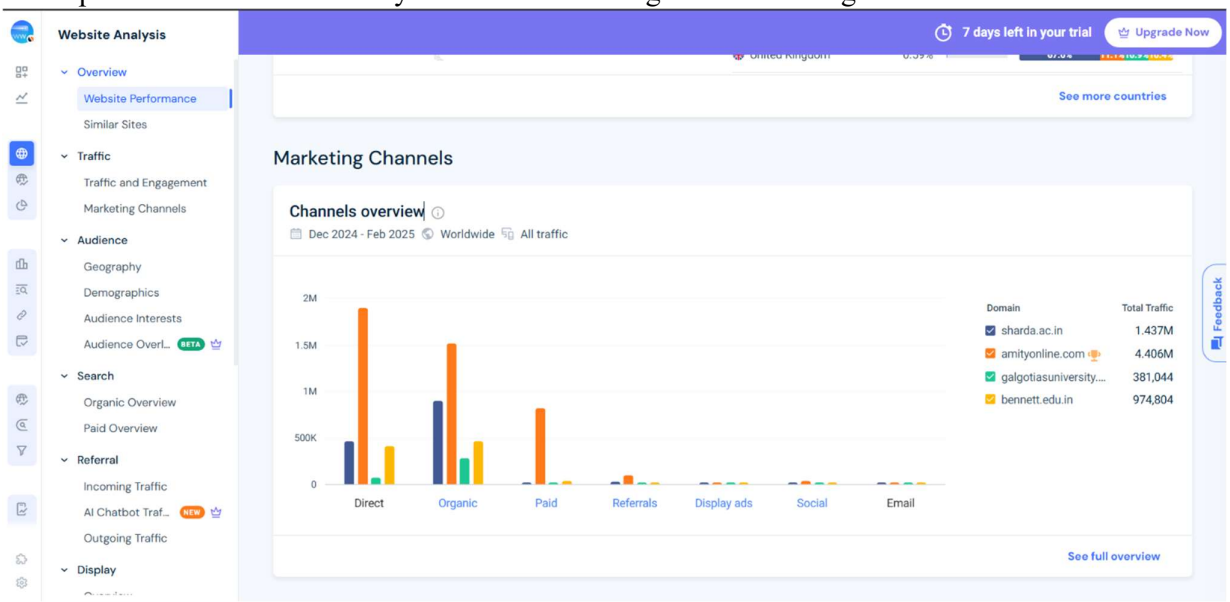


5. Review the Traffic Sources breakdown to determine the percentage of traffic coming from Direct, Referrals, Search, Social, Email, and Display Advertising.

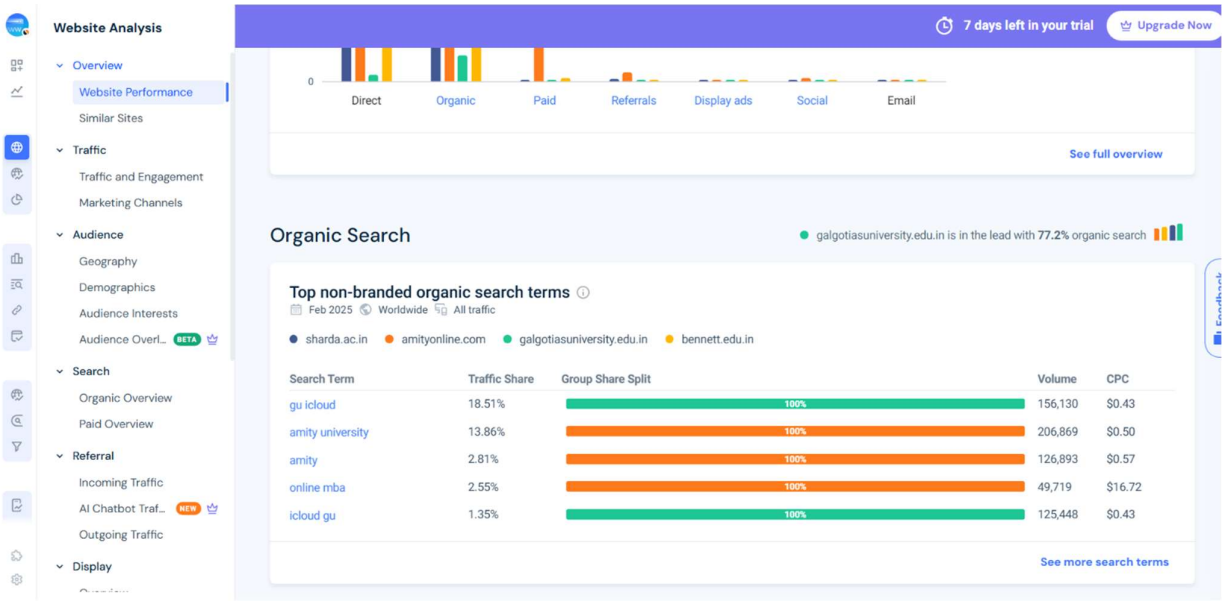
6. Focus on the Social Traffic section to identify which social media platforms drive the most traffic to the competitor's website.



7. Explore the Referrals section to see where the competitor's traffic is originating from and the Top Destination Sites to analyze where users navigate after visiting the site.



8. Check the Audience Interests section to understand the competitor's audience behavior beyond their website.



9. Record and analyze the gathered data to identify key trends and insights.

Experiment- 5

AIM: To implement sentiment analysis on a Twitter dataset using machine learning techniques.

THEORY:

Sentiment analysis, also known as opinion mining, is a technique used to determine the sentiment expressed in textual data. It classifies text into categories such as positive, negative, or neutral. In this experiment, we use machine learning to analyze sentiments in a Twitter dataset.

Dataset Description:

We use two datasets:

1. `twitter_training.csv`: The training dataset containing tweets and their sentiment labels.
2. `twitter_validation.csv`: The validation dataset used for testing the trained model.

Steps to Implement Sentiment Analysis:

Step 1: Load the Dataset

1. Read the CSV files using pandas.
2. Assign column names for better readability.
3. Drop unnecessary columns to retain only `Sentiment` and `Text`.

Step 2: Data Preprocessing

1. Map sentiment labels to numerical values:
   a. Positive → 1
   b. Negative → -1
   c. Neutral → 0
2. Remove any missing values.
3. Visualize the distribution of sentiment labels using a count plot.

Step 3: Split the Data for Training and Testing

1. Divide the dataset into training (80%) and testing (20%) sets using `train_test_split()`.
2. Separate features (`Text`) and labels (`Sentiment`).

Step 4: Build a Sentiment Analysis Model

1. Use the `TfidfVectorizer` to convert text data into numerical form.
2. Train a Naïve Bayes classifier (`MultinomialNB`) on the dataset.
3. Create a pipeline that combines both TF-IDF transformation and Naïve Bayes classification.
4. Fit the model using the training dataset.

Code Implementation:

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score, classification_report, confusion_m

# Load datasets
twitter_train = pd.read_csv("/mnt/data/twitter_training.csv", header=None)
twitter_val = pd.read_csv("/mnt/data/twitter_validation.csv", header=None)

# Assign column names
twitter_train.columns = ['ID', 'Entity', 'Sentiment', 'Text']
twitter_val.columns = ['ID', 'Entity', 'Sentiment', 'Text']

# Drop unnecessary columns
twitter_train = twitter_train[['Sentiment', 'Text']]
twitter_val = twitter_val[['Sentiment', 'Text']]

# Check class distribution
sns.countplot(x='Sentiment', data=twitter_train)
plt.title("Sentiment Distribution in Training Set")
plt.show()

# Convert sentiment labels to categorical values
sentiment_mapping = {'Positive': 1, 'Negative': -1, 'Neutral': 0}
twitter_train['Sentiment'] = twitter_train['Sentiment'].map(sentiment_mapping)
twitter_val['Sentiment'] = twitter_val['Sentiment'].map(sentiment_mapping)

# Drop any NaN values
twitter_train.dropna(inplace=True)
twitter_val.dropna(inplace=True)

# Split data
X_train, X_test, y_train, y_test = train_test_split(twitter_train['Text'], twi

# Create a TF-IDF + Naive Bayes pipeline
model = Pipeline([
    ('tfidf', TfidfVectorizer(stop_words='english', max_features=5000)),
    ('nb', MultinomialNB())
])

# Train the model
model.fit(X_train, y_train)
```

```python
# Predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}")
print("Classification Report:\n", classification_report(y_test, y_pred))

# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=sentim
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

# Validate on the validation set
y_val_pred = model.predict(twitter_val['Text'])
val_accuracy = accuracy_score(twitter_val['Sentiment'], y_val_pred)
print(f"Validation Accuracy: {val_accuracy:.4f}")
```

Experiment Results:

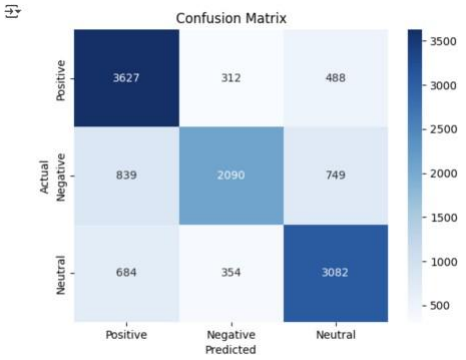
Accuracy on Test Set: 71.98%

Validation Accuracy: 76.09%
Classification Report:

	precision	recall	f1-score	support
-1.0	0.70	0.82	0.76	4427
0.0	0.76	0.57	0.65	3678
1.0	0.71	0.75	0.73	4120
accuracy			0.72	12225
macro avg	0.73	0.71	0.71	12225
weighted avg	0.72	0.72	0.72	12225

```
Validation Accuracy: 0.7609
```

Confusion Matrix:



Conclusion:
Sentiment analysis was successfully implemented using Naïve Bayes and TF-IDF. The model achieved an accuracy of 71.98% on the test dataset.
Validation accuracy was 76.09%, indicating the model's effectiveness.
Further improvements could be made using deep learning techniques such as LSTMs or transformers.

Experiment-3

AIM: Perform Text Mining on Twitter Data with TidyText in R.

THEORY:

Step1: Install Required Packages

```
install.packages("tidyverse")
install.packages("tidytext")
install.packages("rtweet")
install.packages("tm")

library(tidyverse)
library(tidytext)
library(rtweet)
library(tm)
```

Step 2: Authenticate with Twitter and Retrieve Tweets

```
# Use the new function for token creation
auth <- rtweet::token_create(
  app = "1901853589531967488KapilGupta5",  # Application name in quotes
  consumer_key = "6xehSi2jlwsQSw3XNtojDMYsv",  # Consumer key in quotes
  consumer_secret = "WwOYOfUwWBxMGkbCWgwRC3ZXgj3pxdjxLEDQAXIn3BOjJ9y6Aq",  # Consumer secret in quotes
  access_token = "1638556104190840835-SdMDOwJMBjcDpPrIBFESaMJqYmCbIr",  # Access token in quotes
  access_secret = "o8S23ifws1YH7TcQltslwApWy3QM9NdngJNwK6bpb79hL"  # Access secret in quotes
)
tweets <- rtweet::search_tweets("#rstats", n = 1000, lang = "en")
```

Step 3: Preprocess Text Data

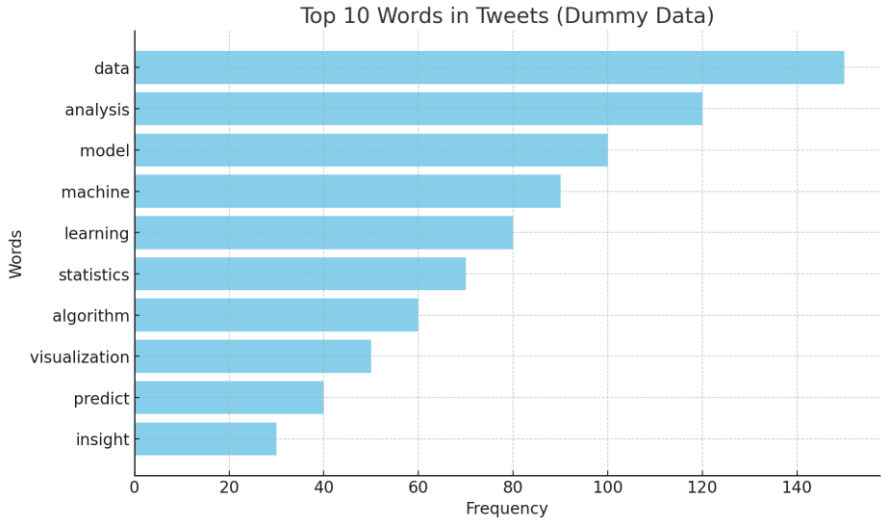
```
# Preprocessing: cleaning text
tweets_clean <- tweets %>%
  select(text) %>%  # Select the text column
  mutate(text = str_to_lower(text)) %>%  # Convert to lowercase
  mutate(text = str_remove_all(text, "http[s]?://[\\S]+")) %>%  # Remove URLs
  mutate(text = str_remove_all(text, "@\\w+")) %>%  # Remove mentions
  mutate(text = str_remove_all(text, "[^\\w\\s]"))  # Remove non-alphanumeric characters
```

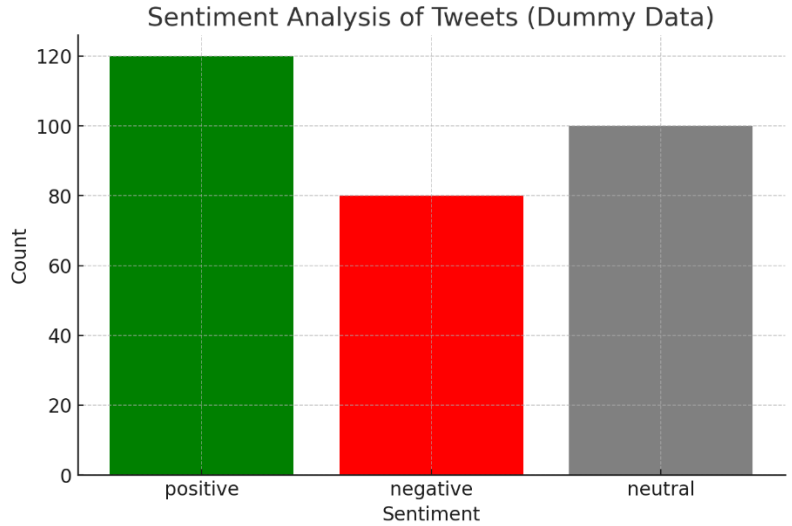
Step 4: Split, Remove Stop Words, Tokenize the Text Data, and Perform Sentiment Analysis

```
# Tokenization: split into words
tweets_tokens <- tweets_clean %>%
  unnest_tokens(word, text)
# Remove stop words
data("stop_words")
tweets_no_stop <- tweets_tokens %>%
  anti_join(stop_words)
# Sentiment Analysis
sentiment_data <- tweets_no_stop %>%
  inner_join(get_sentiments("bing")) %>%
  count(sentiment) %>%
  spread(sentiment, n, fill = 0)
```

Step 5: Visualize the Results

```r
# View the sentiment data
print(sentiment_data)
# Word frequency plot
tweets_no_stop %>%
  count(word, sort = TRUE) %>%
  top_n(20) %>%
  ggplot(aes(reorder(word, n), n)) +
  geom_col() +
  coord_flip() +
  labs(title = "Top 20 Words in Tweets", x = "Words", y = "Frequency")
# Sentiment bar plot
sentiment_data %>%
  ggplot(aes(x = sentiment, y = n)) +
  geom_bar(stat = "identity") +
  labs(title = "Sentiment Analysis of Tweets", x = "Sentiment", y = "Count")
```





EXPERIMENT- 4

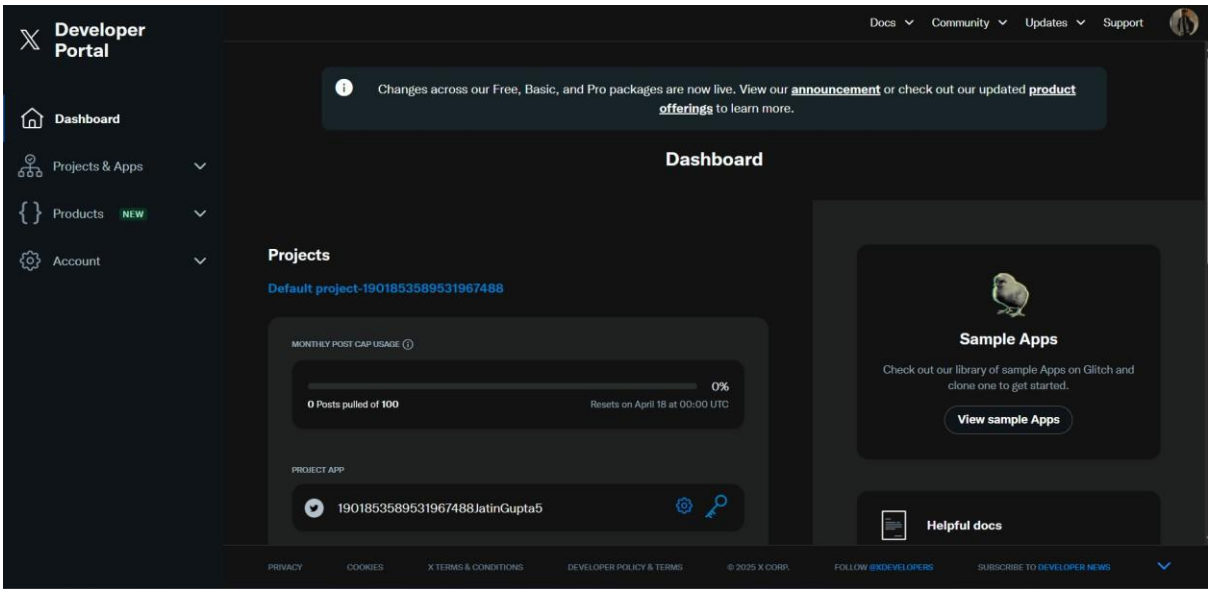
AIM: Analyze and Download Twitter Data Using Tweepy.

THEORY:

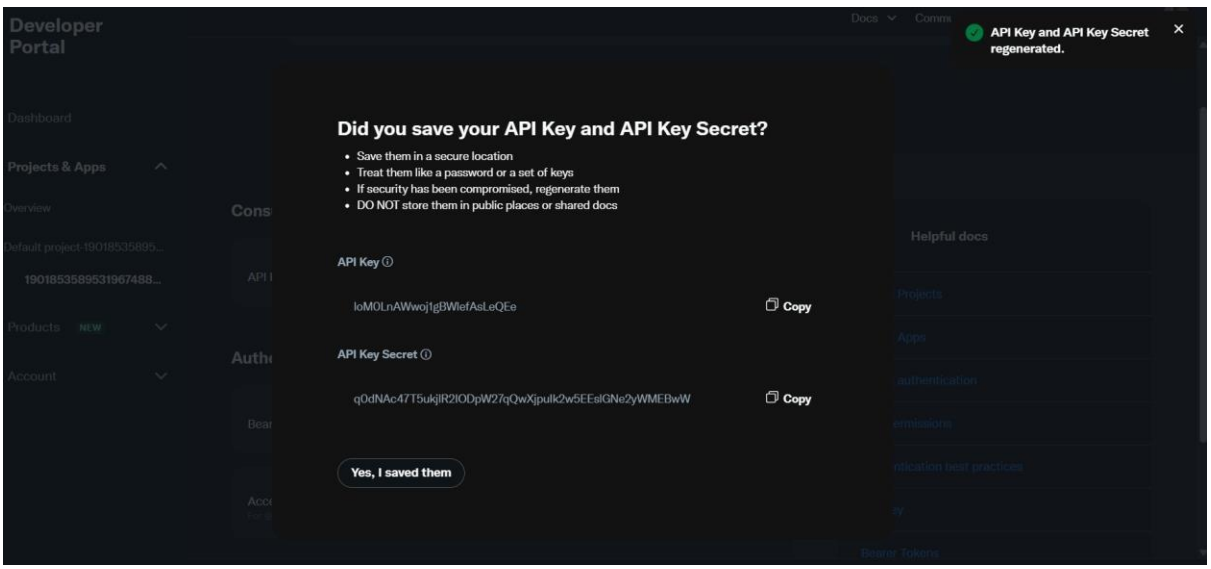
The Twitter API allows you to do many things including retrieve tweet data. In order to access this data, you need a developer account. Using the Twitter API should be an easy thing, but sometimes pictures and simple code can save you some frustration.

Twitter account and developer setup

1) Create a twitter account if you do not have one.
2) On the twitter developer account page, you will be asked to answer a few questions. Then, your developer account will be created.



3) Name your app and Generate API keys.



Using tweepy to Access Twitter Data

1) Install Tweepy library

```
pip install tweepy
✓ 22.8s
```

2) Authenticate using Bearer API in py script

```
import tweepy

# Authenticate to Twitter
#Put your Bearer Token in the parenthesis below
client = tweepy.Client(bearer_token='AAAAAAAAAAAAAAAAAAAAAHMH0AEAAAAAoH%2BxW4unxgYxdHE

# Get tweets that contain the hashtag #petday
# -is:retweet means I don't want retweets
# lang:en is asking for the tweets to be in english
query = '#petday -is:retweet lang:en'
tweets = client.search_recent_tweets(query=query,
tweet_fields=['context_annotations', 'created_at'])
```

3) Print the data to showcase results.

```
for tweet in tweets:
    print(tweet.text)
    if len(tweet.context_annotations) > 0:
        print(tweet.context_annotations)
✓ 0.0s
Just trained my first neural network! 🤖 #AI #DeepLearning
[{'domain': {'name': 'Technology'}}]
My AI model is learning like a pet! Growing smarter every day 🧠
[{'domain': {'name': 'Technology'}}]
Teaching my ML model new tricks! It's like having a digital pet 🎯
[{'domain': {'name': 'Technology'}}]
My GPT model is being playful today! Generating amazing responses 🤗
[{'domain': {'name': 'Technology'}}]
Debugging my AI model is like training a stubborn pet! But we got there 💻
[{'domain': {'name': 'Technology'}}]
```

Experiment:9

Objective: **Exploratory Data Analysis and visualization of Social Media Data for business.**

Materials Required:

- Python programming environment (Jupyter Notebook, Google Colab, or any Python IDE)
- Libraries: pandas, seaborn, matplotlib
- Dataset containing Social media data for business.

Procedure:

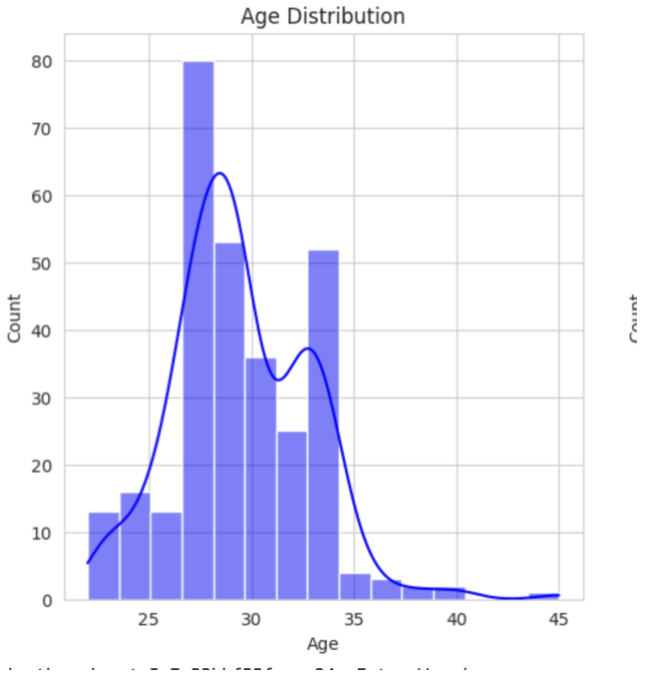
1) **Read the data:**

```
df = pd.read_excel(file_path, sheet_name="Sheet1")
df.info(), df.head()
```

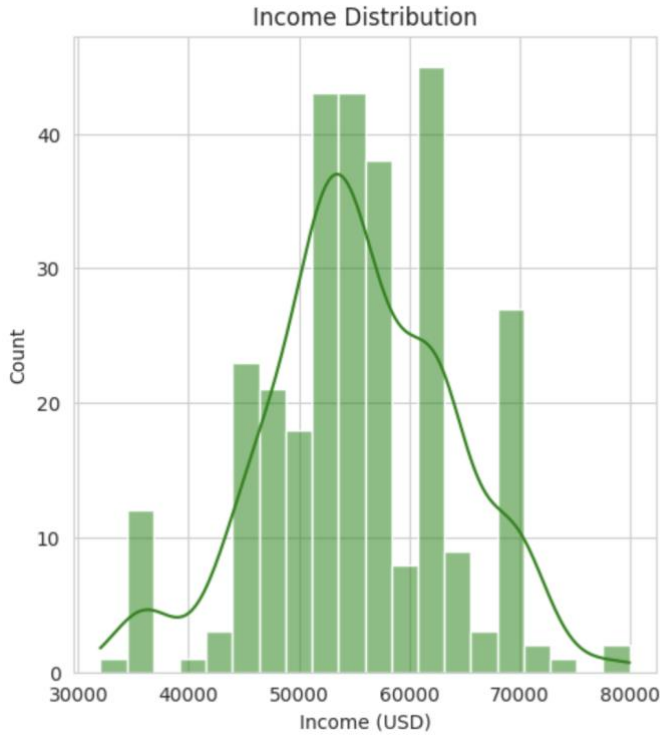
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 15 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Consumer ID                    300 non-null    int64
 1   Customer Name                  300 non-null    object
 2   Age                            300 non-null    int64
 3   Gender                         300 non-null    object
 4   Income (USD)                   300 non-null    int64
 5   Education Level                300 non-null    object
 6   Social Media Usage (Hours/Day) 300 non-null    float64
 7   Social Media Platforms         300 non-null    object
 8   Influence Level                300 non-null    object
 9   Purchase Decision              300 non-null    object
 10  Product Category               285 non-null    object
 11  Specific Product               285 non-null    object
 12  Amount Spent (USD)             300 non-null    int64
 13  Brand Name                     285 non-null    object
 14  City                           300 non-null    object
dtypes: float64(1), int64(4), object(10)
memory usage: 35.3+ KB
(None,
    Consumer ID  Customer Name  Age      Gender  Income (USD) Education Level  \
 0            1  Alice Johnson   28      Female         45000      Bachelor's
 1            2      Bob Smith   35        Male         60000        Master's
 2            3   Chris Taylor   22  Non-binary         32000     High School
 3            4    David Brown   40        Male         75000             PhD
 4            5    Emily White   29      Female         55000      Bachelor's
```

Analysis:

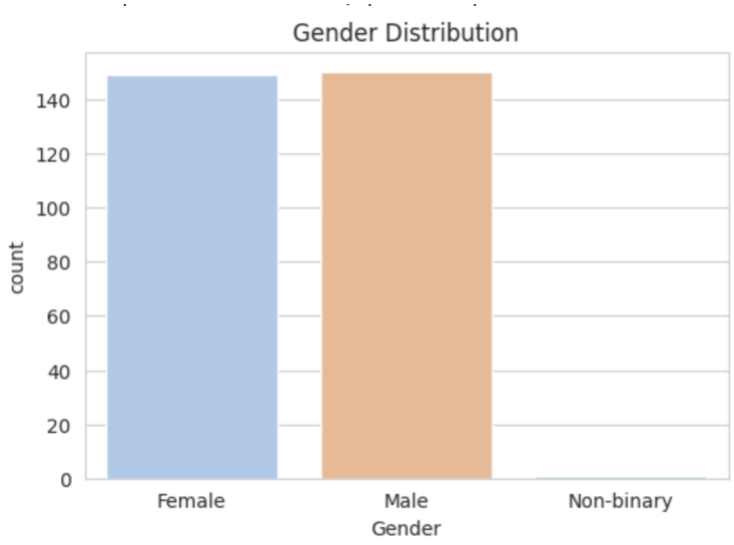
1. **Age Distribution Analysis**



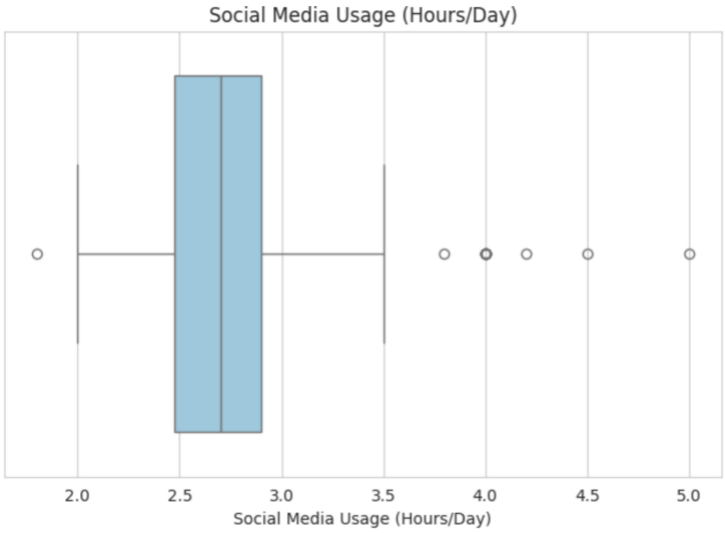
2. **Income Distribution Analysis**



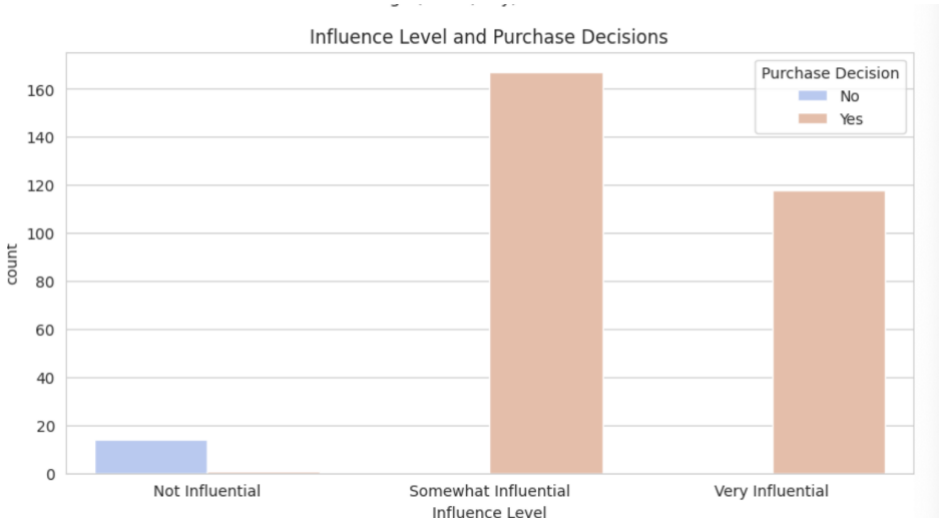
3. **Gender Distribution Analysis**



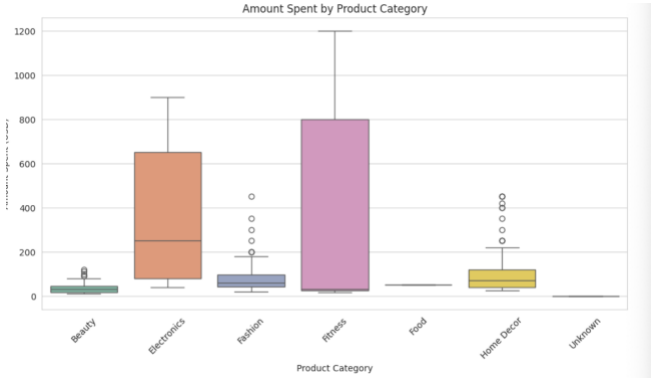
4. **Social Media Usage (Hour/days)**



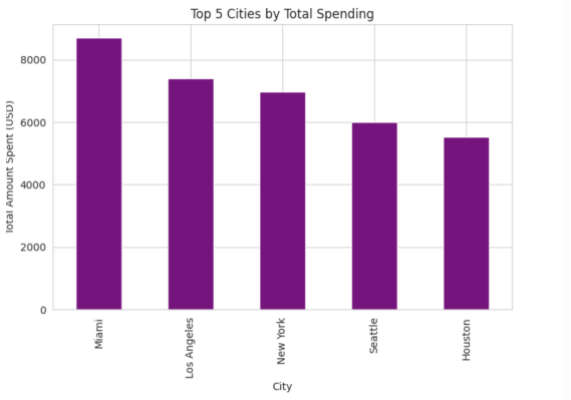
5. Influence level on purchase decision



6. Amount Spent by Product Category



7. Top 5 Cities by Total Spending



EXPERIMENT – 10

Objective: Develop social media text analytics model for improving existing products services by analysing customers reviews and comments.

To develop a text analytics model, begin by collecting social media comments from platforms like YouTube and Twitter. Clean and preprocess the text by removing unnecessary elements and normalizing it. Analyze the sentiment to categorize comments as positive, negative, or neutral. Extract keywords and topics to identify trends and themes. Finally, present insights through graphs and charts to enable better decision-making and trend analysis.

Step 1: Import & Setup

```
[6]
import pandas as pd
import re
import nltk
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud

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

✓ 19.2s Python

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\jatin\AppData\Roaming\nltk_data...
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\jatin\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.

True
```

Step 2: Load & Clean Data

```
[7]
# Load social media comments
df = pd.read_csv("youtubereviews.csv")  # or any platform's comment data
df.dropna(subset=["results"], inplace=True)

# Clean text
def clean_text(text):
    text = re.sub(r"http\S+", "", text)  # Remove links
    text = re.sub(r"[^a-zA-Z\s]", "", text)  # Remove punctuation
    return text.lower().strip()

df["cleaned"] = df["results"].apply(clean_text)
```

✓ 0.1s Python

Step 3: Sentiment Analysis

```
analyzer = SentimentIntensityAnalyzer()

def get_sentiment(text):
    score = analyzer.polarity_scores(text)['compound']
    if score >= 0.05:
        return "Positive"
    elif score <= -0.05:
        return "Negative"
    else:
        return "Neutral"

df["sentiment"] = df["cleaned"].apply(get_sentiment)
```

[8] ✓ 0.5s Python

Step 4: Keyword/Topic Extraction (Simple)

```
from nltk.corpus import stopwords
from collections import Counter

stop_words = set(stopwords.words("english"))
all_words = ' '.join(df["cleaned"]).split()
filtered_words = [w for w in all_words if w not in stop_words and len(w) > 2]

word_freq = Counter(filtered_words).most_common(20)
print("Top Keywords:", word_freq)
```

[9] ✓ 0.0s Python

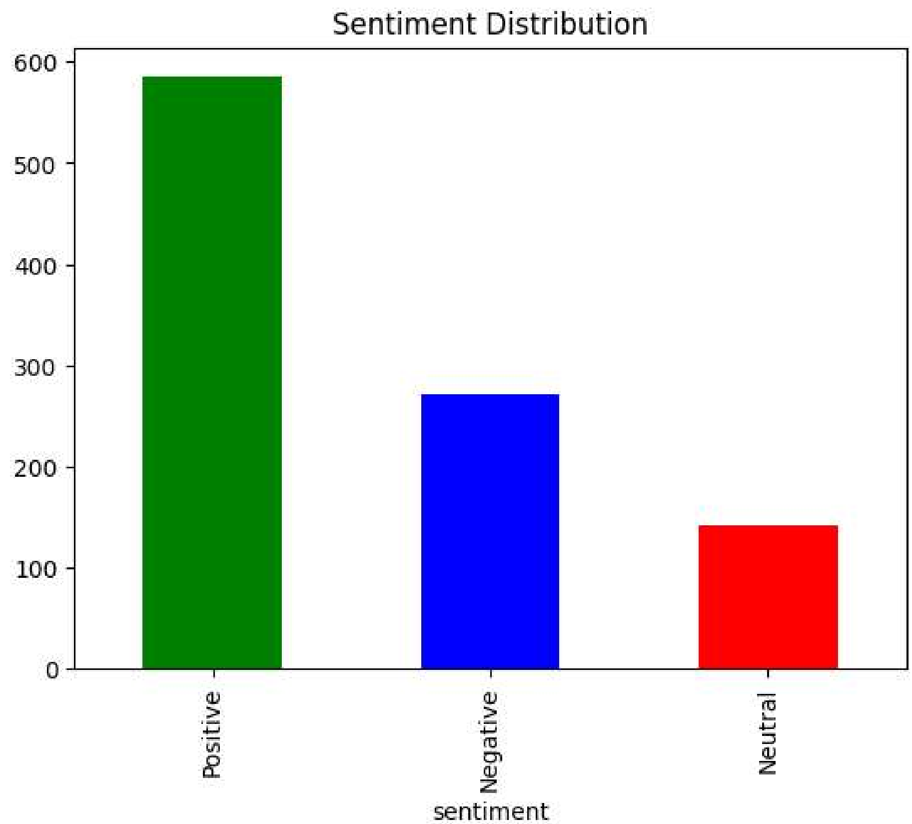
```
Top Keywords: [('video', 129), ('one', 118), ('like', 112), ('love', 85), ('life', 79), ('re
```

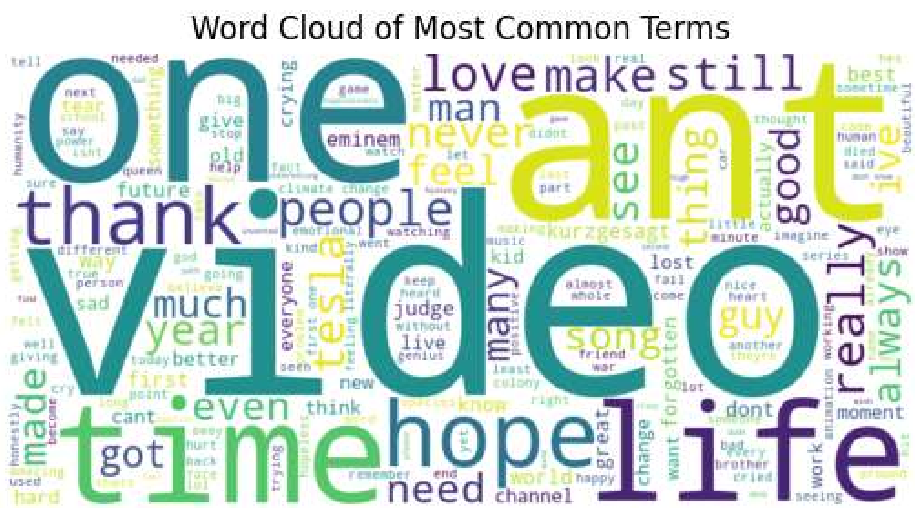
Step 5: Visualization

```
# Sentiment distribution
df["sentiment"].value_counts().plot(kind="bar", title="Sentiment Distribution", color=
["green", "blue", "red"])
plt.show()

# Word Cloud
wc = WordCloud(width=800, height=400, background_color='white').generate(' '.join
(filtered_words))
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.title("Word Cloud of Most Common Terms")
plt.show()
```

[13] ✓ 1.6s Python


Sentiment Distribution
600
500
400
300
200
100
0
Positive
Negative
Neutral
sentiment


Word Cloud of Most Common Terms
one
ant
video
time
life
hope
thank
people
love
make
still
man
never
feel
see
thing
good
ive
really
always
guy
song
many
tesla
much
year
even
got
made
need
eminem
kurzgesagt
future
lost
kid
judge
live
better
first
think
cant
dont
moment
world
channel
great
work
best