VIT-AP
UNIVERSITY

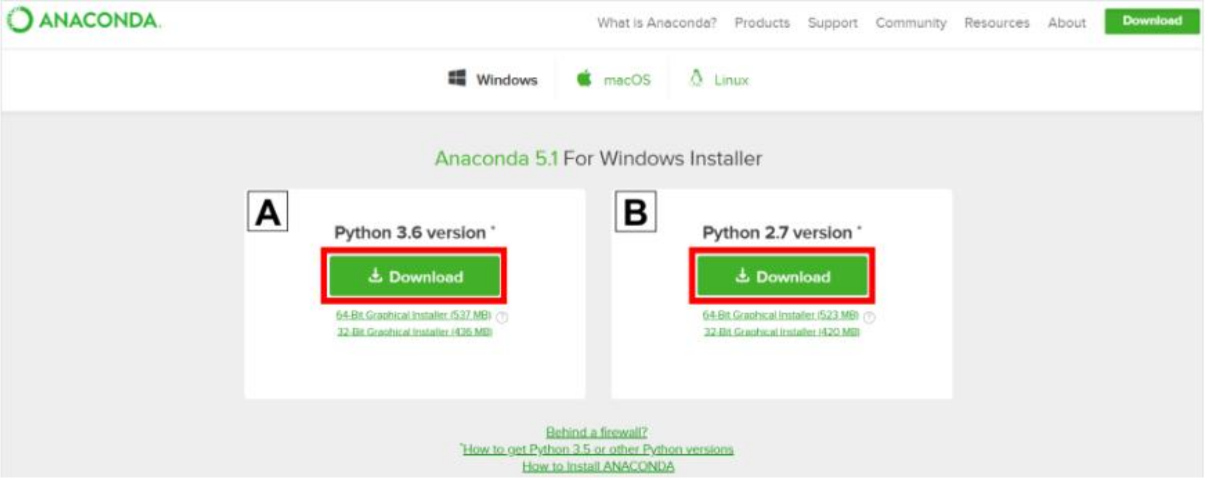
Social Network Analysis

Name: Gorla Sai Dheraz
Registration Number: 21BCE8701
Faculty: Meghavathu S. S. Nayak

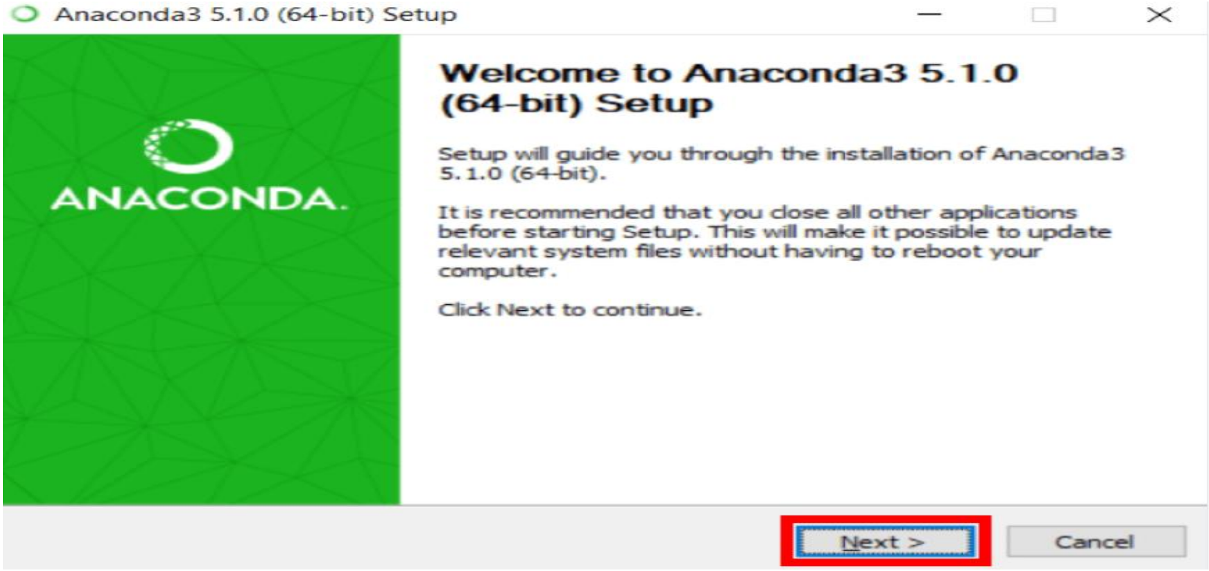
CONTENT

Install Python Software and its GUI tools.
The procedure of installing Anaconda:

1. Go to the **Anaconda Website** and choose a Python 3. x graphical installer. Locate your download and double-click it


ANACONDA.
What is Anaconda?
Products
Support
Community
Resources
About
Download
Windows
macOS
Linux
Anaconda 5.1 For Windows Installer
A
Python 3.6 version *
Download
64-Bit Graphical Installer (537 MB)
32-Bit Graphical Installer (436 MB)
B
Python 2.7 version *
Download
64-Bit Graphical Installer (523 MB)
32-Bit Graphical Installer (420 MB)
Behind a firewall?
*How to get Python 3.5 or other Python versions
How to Install ANACONDA

2. When the screen below appears, click on Next.


Anaconda3 5.1.0 (64-bit) Setup

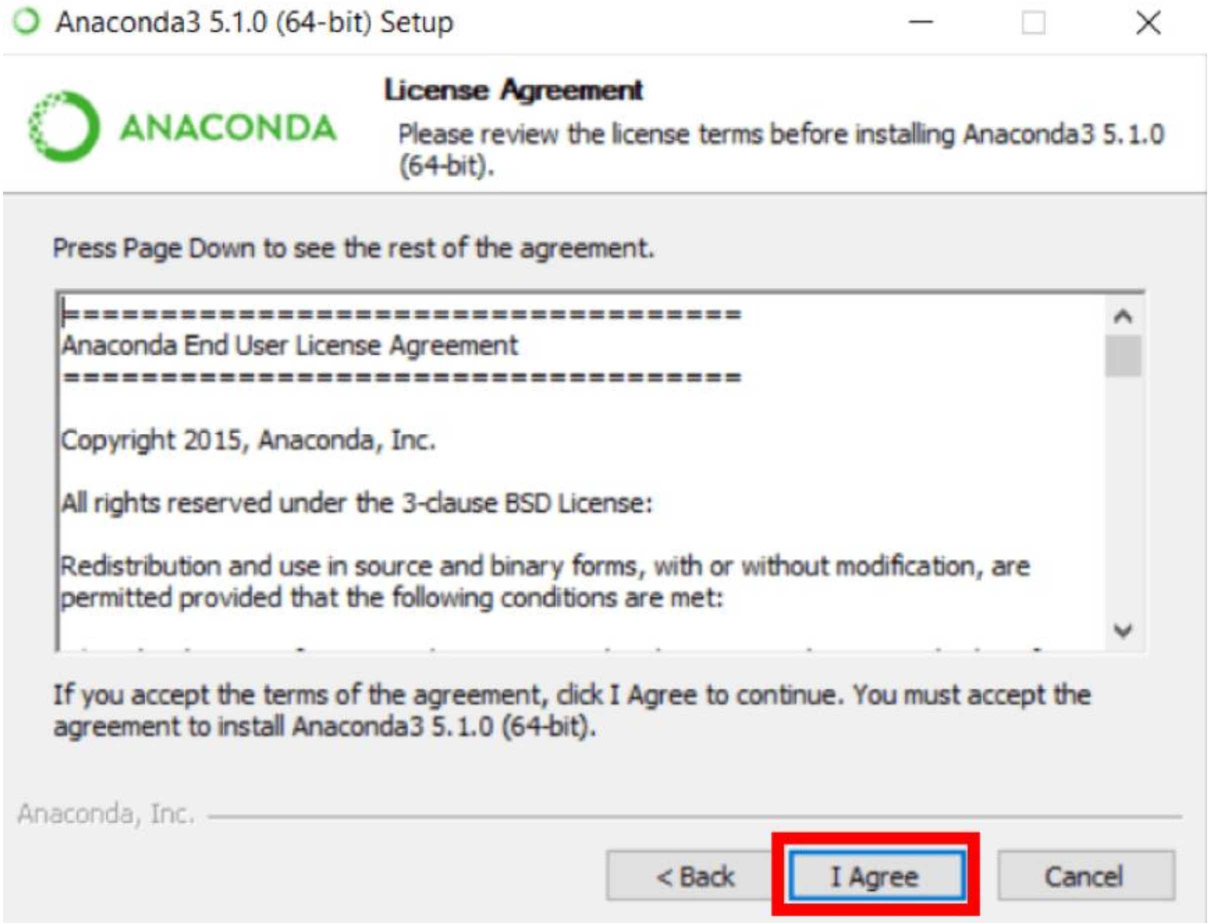
Welcome to Anaconda3 5.1.0 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 5.1.0 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

3. Read the license agreement and click on I Agree.



License Agreement

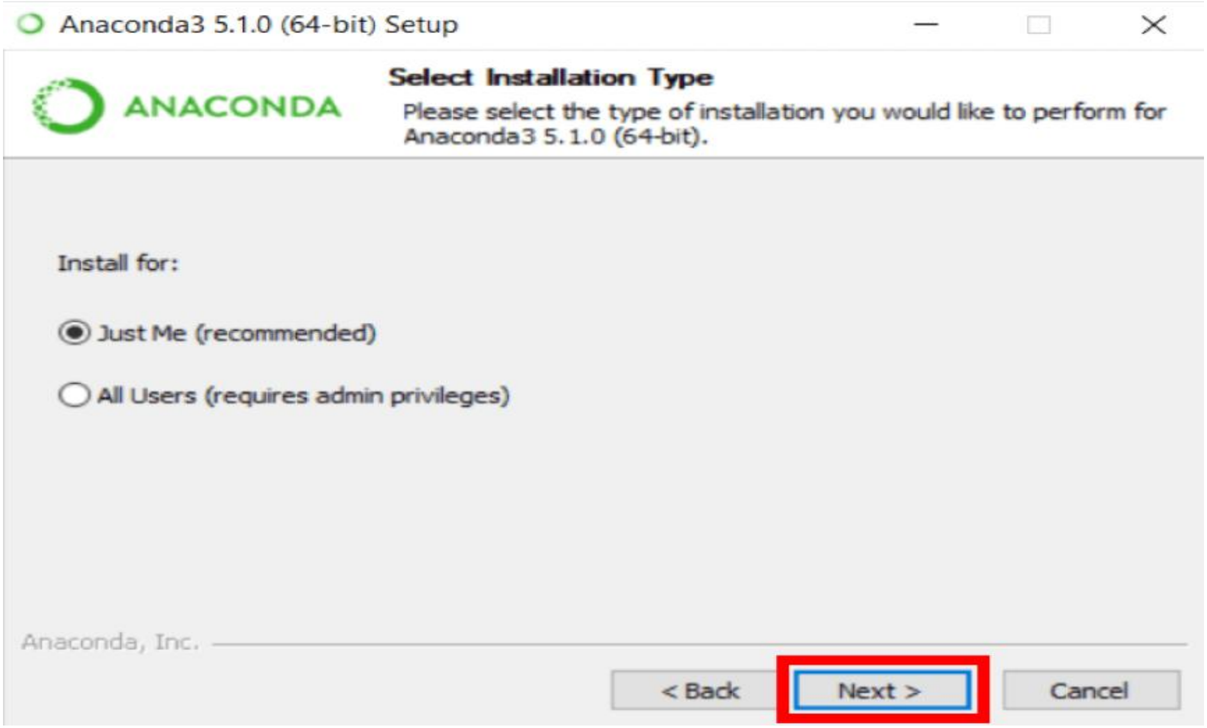
Please review the license terms before installing Anaconda3 5.1.0 (64-bit).

Press Page Down to see the rest of the agreement.

```
===================================
Anaconda End User License Agreement
===================================

Copyright 2015, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 5.1.0 (64-bit).

4. Click on Next.



Select Installation Type

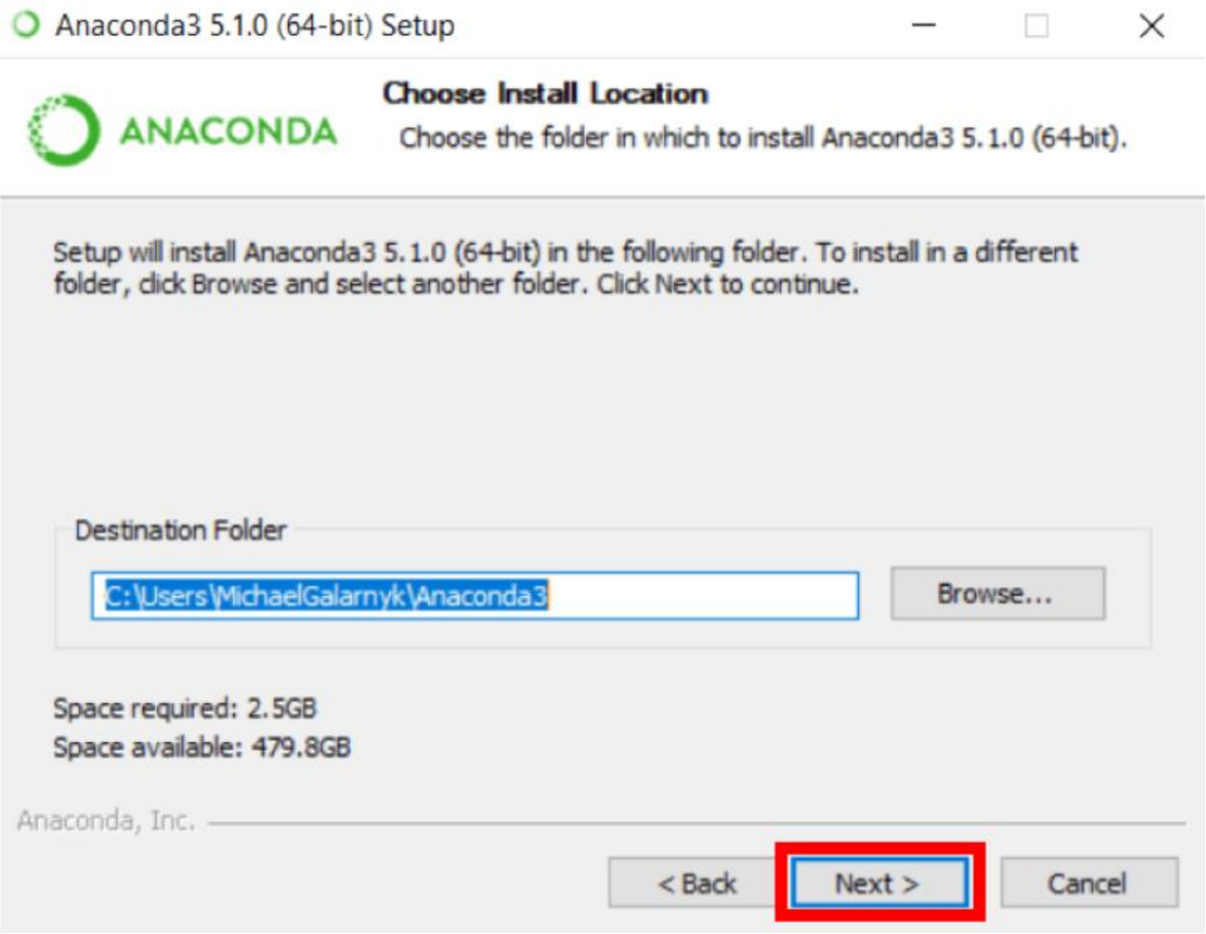
Please select the type of installation you would like to perform for Anaconda3 5.1.0 (64-bit).

Install for:

● Just Me (recommended)

○ All Users (requires admin privileges)

5. Note your installation location and then click Next.



Anaconda3 5.1.0 (64-bit) Setup

Choose Install Location

Choose the folder in which to install Anaconda3 5.1.0 (64-bit).

Setup will install Anaconda3 5.1.0 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

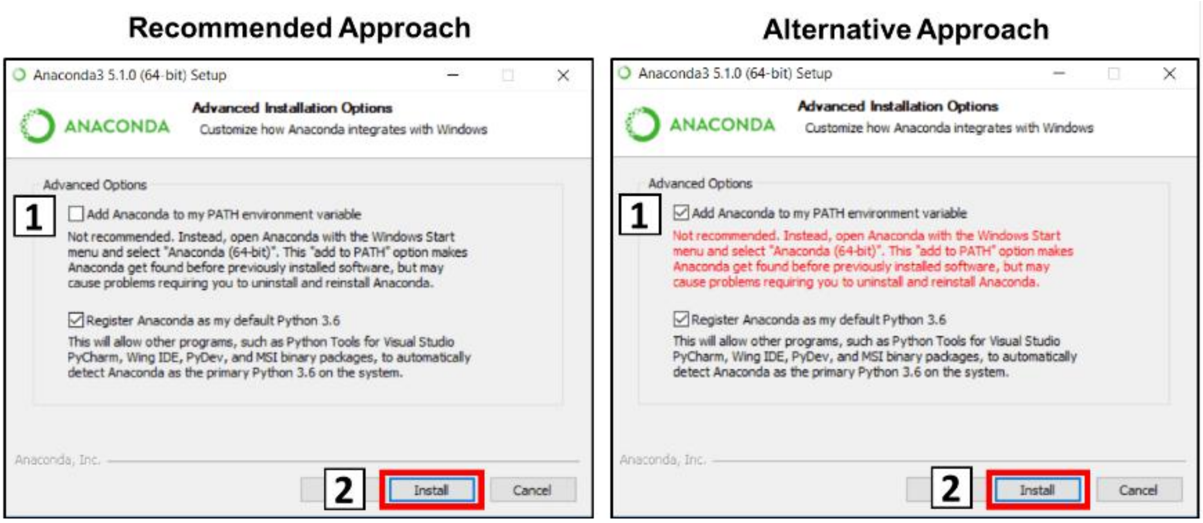
Destination Folder

C:\Users\MichaelGalarnyk\Anaconda3

Space required: 2.5GB
Space available: 479.8GB

Anaconda, Inc.

6. This is an important part of the installation process. The recommended approach is to not check the box to add Anaconda to your path. This means you will have to use Anaconda Navigator or the Anaconda Command Prompt (located in the Start Menu under "Anaconda") when you wish to use Anaconda (you can always add Anaconda to your PATH later if you don't check the box). If you want to be able to use Anaconda in your command prompt (or git bash, cmd, PowerShell, etc), please use the alternative approach and check the box



Recommended Approach

Anaconda3 5.1.0 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

1 □ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.6

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

Anaconda, Inc.

Alternative Approach

Anaconda3 5.1.0 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

1 ☑ Add Anaconda to my PATH environment variable

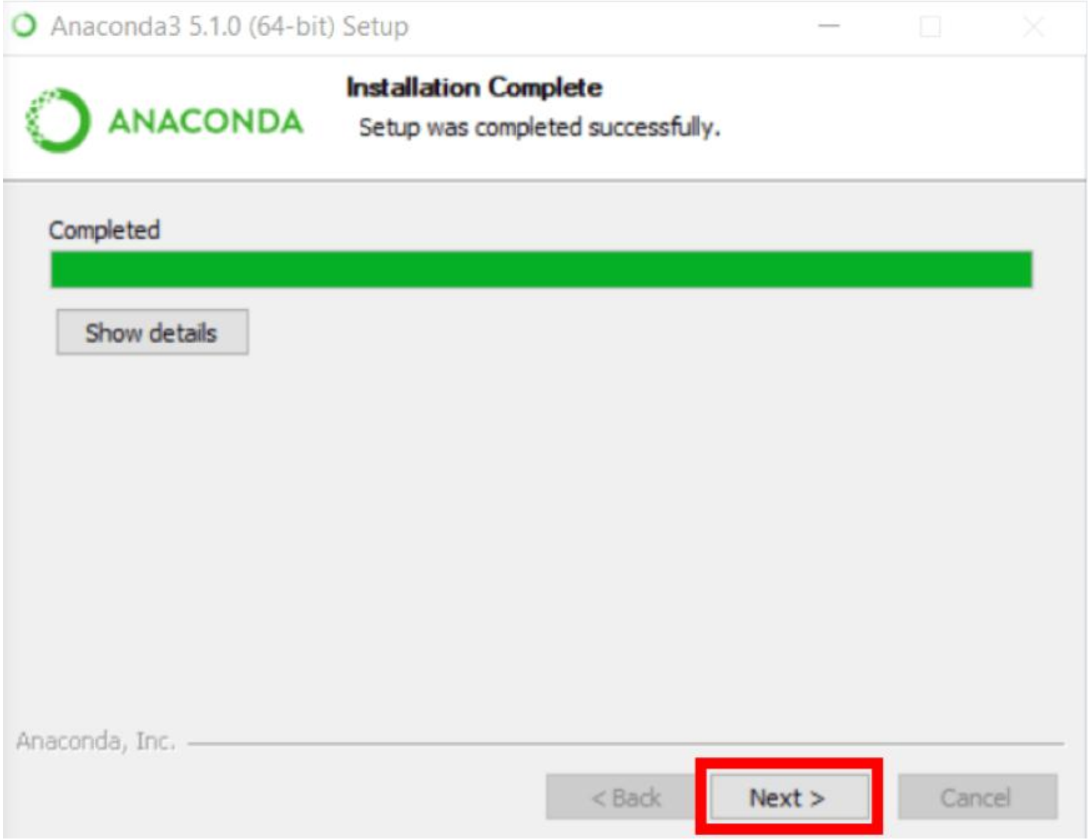
Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.6

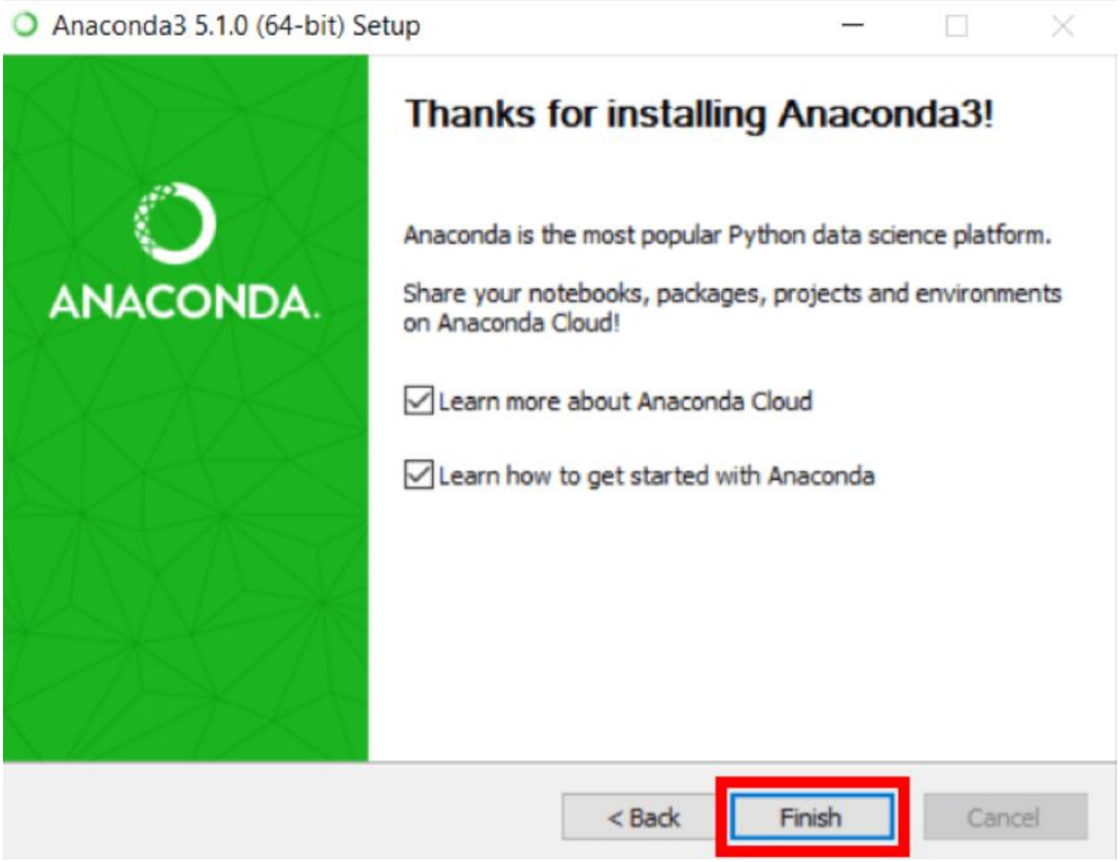
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

Anaconda, Inc.

7. Click on Next.



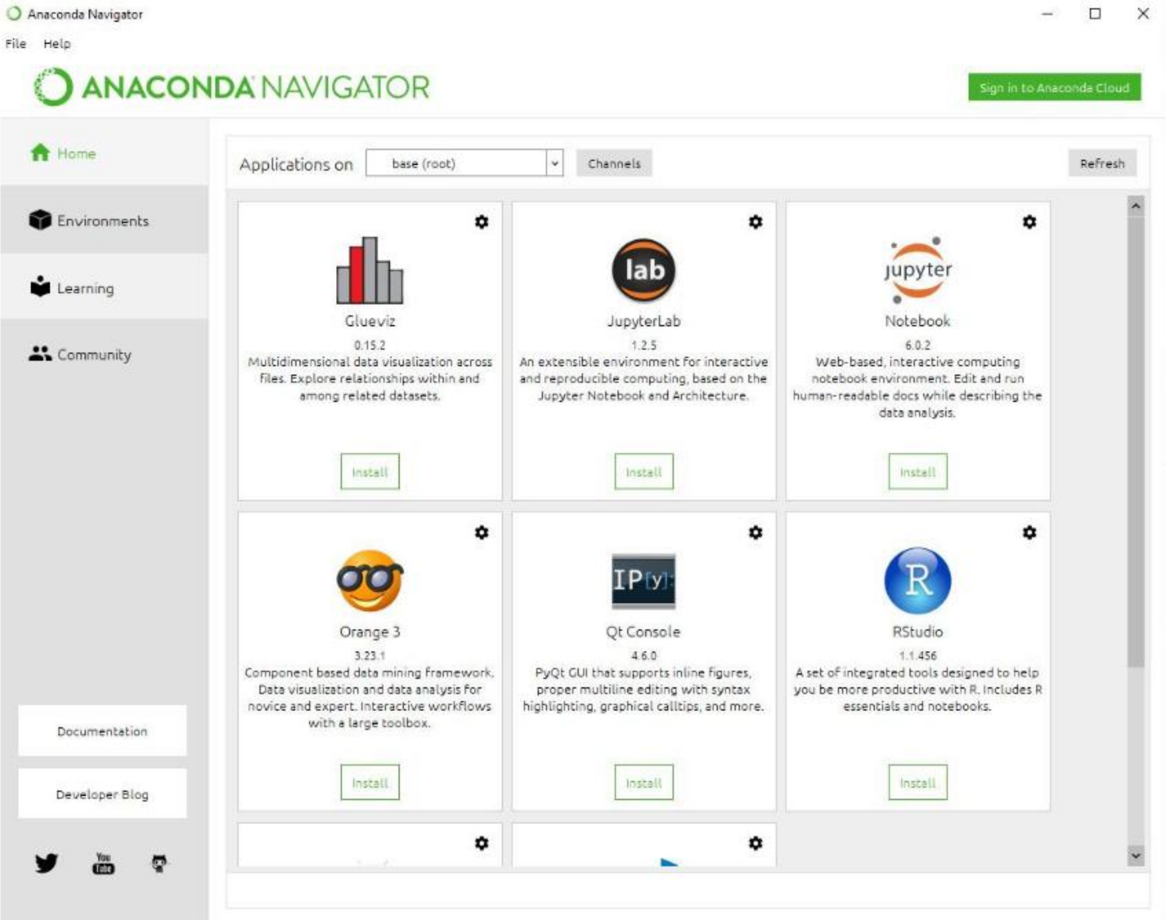
8. Click on Finish.



Anaconda3 5.1.0 (64-bit) Setup
ANACONDA.
Thanks for installing Anaconda3!
Anaconda is the most popular Python data science platform.
Share your notebooks, packages, projects and environments on Anaconda Cloud!
☑ Learn more about Anaconda Cloud
☑ Learn how to get started with Anaconda
< Back
Finish
Cancel

Installing Jupyter Notebook using Anaconda

1: First, Launch the Anaconda Navigator



Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab
1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

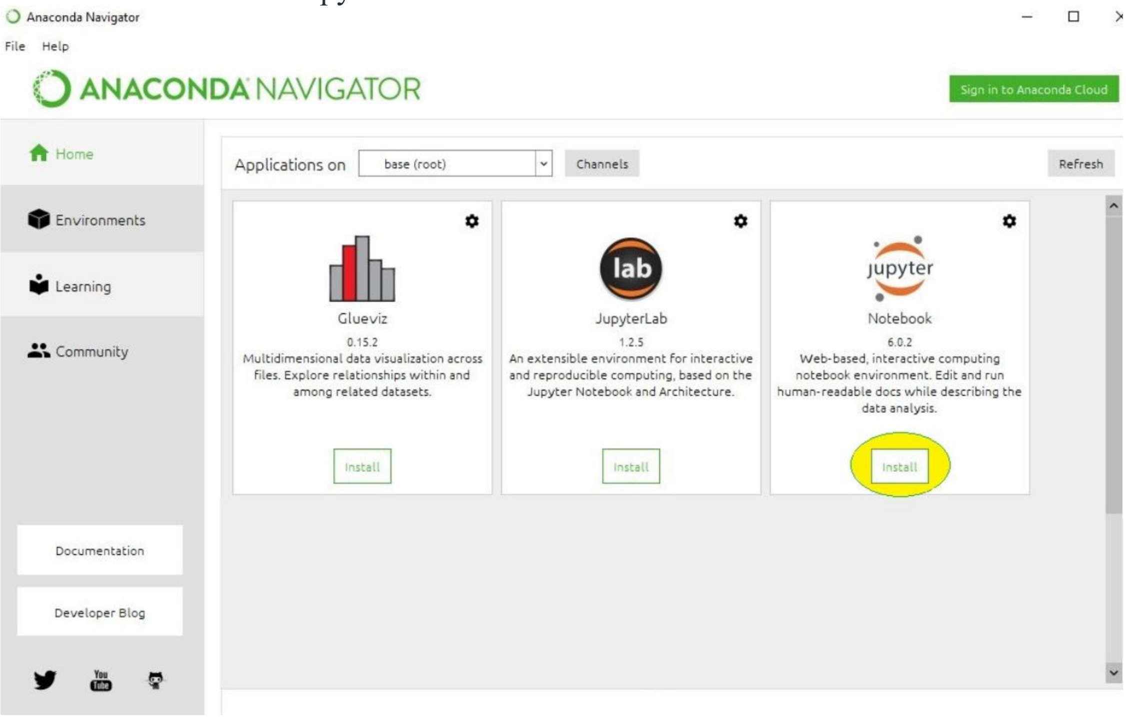
Notebook
6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Orange 3
3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Qt Console
4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

2. Click on the Install Jupyter Notebook Button



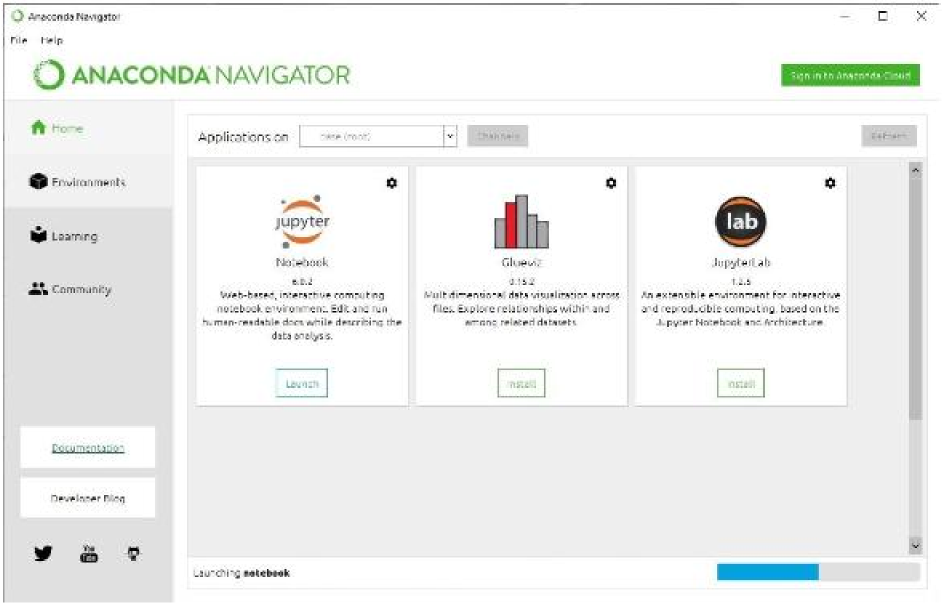
Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab
1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

3. The installation process has begun to Start! Finished Installation

4. Now, click on the Launch button to Launch the Jupyter.



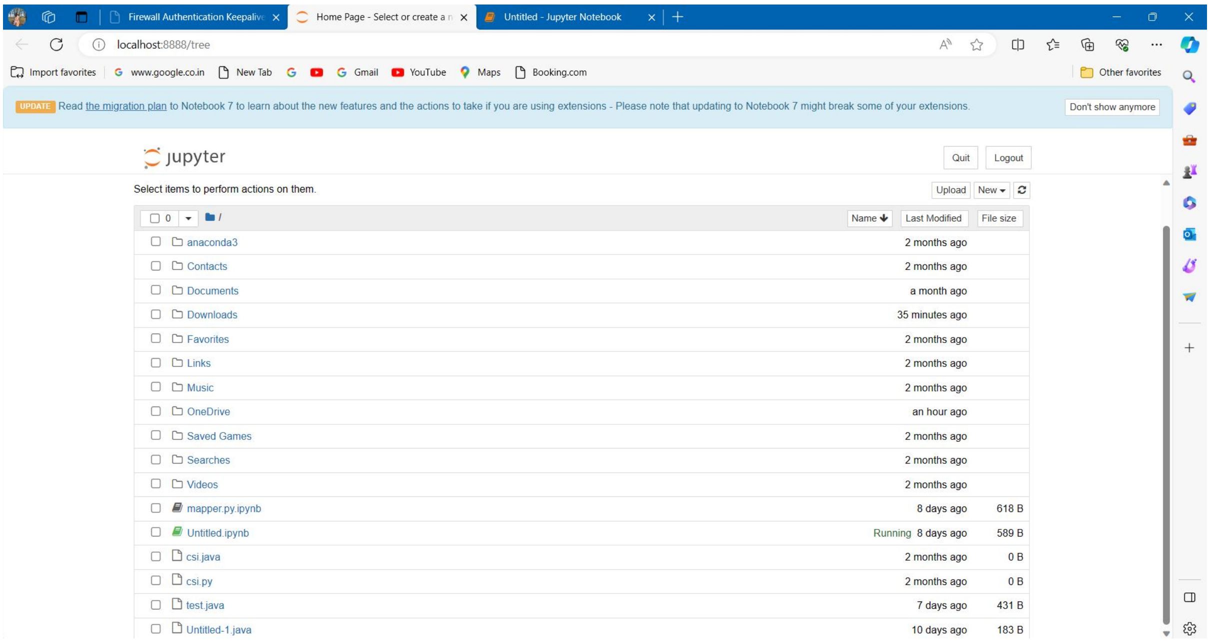
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Multidimensional data visualization across files. Explore relationships within and among related datasets.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launching notebook

Opening jupyter:



Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

Select items to perform actions on them.

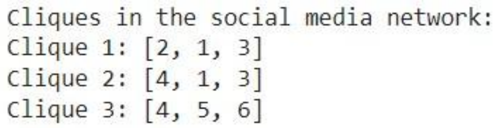
Name	Last Modified	File size
anaconda3	2 months ago	
Contacts	2 months ago	
Documents	a month ago	
Downloads	35 minutes ago	
Favorites	2 months ago	
Links	2 months ago	
Music	2 months ago	
OneDrive	an hour ago	
Saved Games	2 months ago	
Searches	2 months ago	
Videos	2 months ago	
mapper.py.ipynb	8 days ago	618 B
Untitled.ipynb	Running 8 days ago	589 B
csi.java	2 months ago	0 B
csi.py	2 months ago	0 B
test.java	7 days ago	431 B
Untitled-1.java	10 days ago	183 B

Write a Python program for clique in a graph in a social media Network.

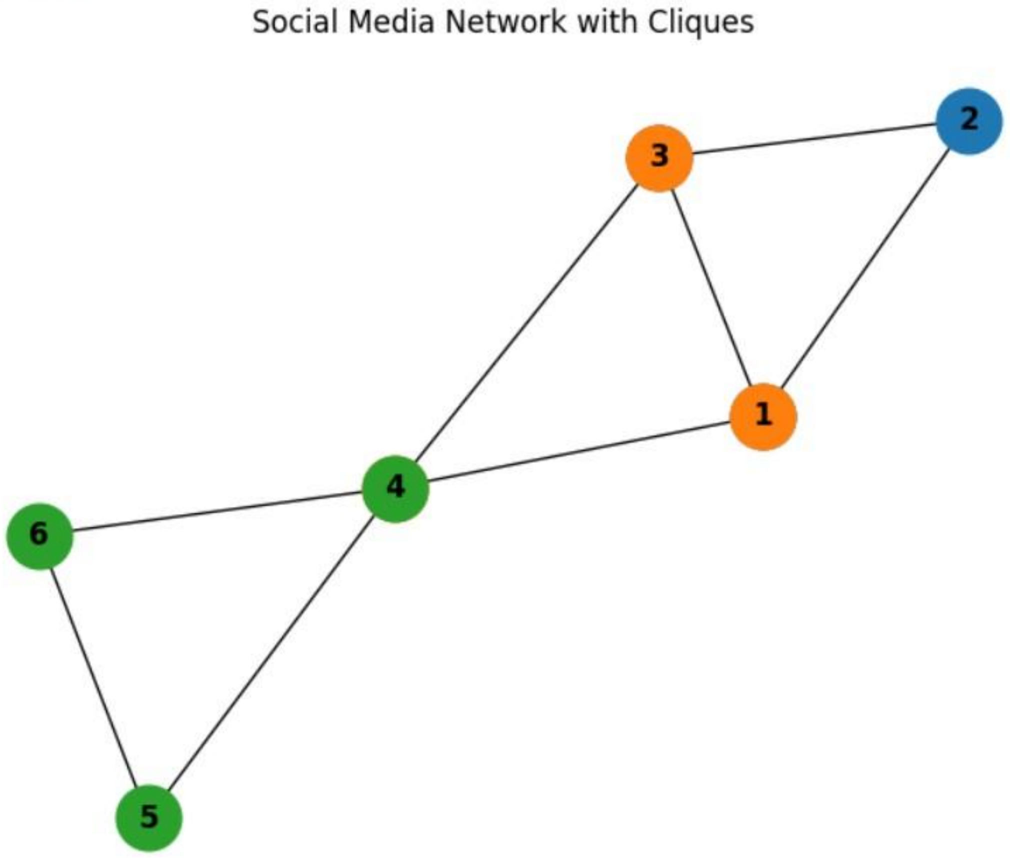
Code:

```
import networkx as nx
import matplotlib.pyplot as plt
def find_cliques(graph):
   cliques = list(nx.find_cliques(graph))
   return cliques
def draw_graph_with_cliques(graph, cliques):
   pos = nx.spring_layout(graph)
   nx.draw(graph, pos, with_labels=True, font_weight='bold')
   for i, clique in enumerate(cliques):
       nx.draw_networkx_nodes(graph, pos, nodelist=clique, node_color=f'C{i}', node_size=700)
   plt.title("Social Media Network with Cliques")
   plt.show()
def main():
   social_graph = nx.Graph()
   edges = [(1, 2), (1, 3),(1, 4), (2, 3), (3, 4), (4, 5), (4, 6), (5, 6)]
   social_graph.add_edges_from(edges)
   cliques = find_cliques(social_graph)
   print("Cliques in the social media network:")
   for i, clique in enumerate(cliques, start=1):
       print(f"Clique {i}: {clique}")
   draw_graph_with_cliques(social_graph, cliques)
if __name__ == "__main__":
   main()
```

Output:



```
Cliques in the social media network:
Clique 1: [2, 1, 3]
Clique 2: [4, 1, 3]
Clique 3: [4, 5, 6]
```



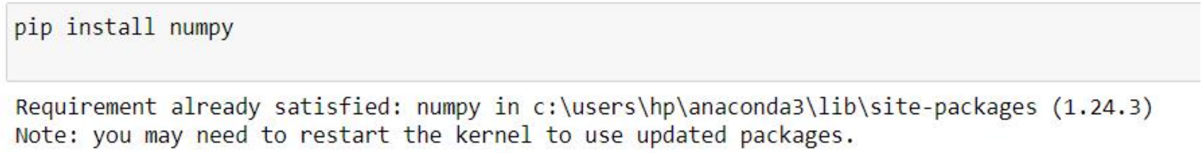
Social Media Network with Cliques

Install required Python packages.

1. NumPy:

NumPy is a fundamental package for scientific computing in Python. It supports large, multi-dimensional arrays and matrices of numerical data, as well as a large library of mathematical functions to operate on these arrays. The package is particularly useful for performing mathematical operations on large datasets and is widely used in machine learning, data analysis, and scientific computing.

pip install numpy



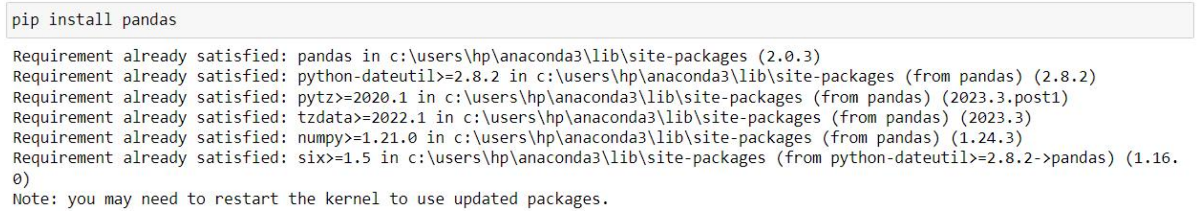
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\hp\anaconda3\lib\site-packages (1.24.3)
Note: you may need to restart the kernel to use updated packages.
```

2. Pandas:

Pandas is a powerful data manipulation library for Python that provides fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data easy and intuitive. The package is particularly well-suited for working with tabular data, such as spreadsheets or SQL tables, and provides powerful data cleaning, transformation, and wrangling capabilities.

pip install pandas



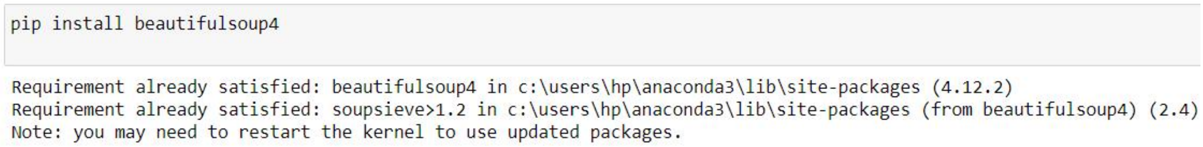
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\hp\anaconda3\lib\site-packages (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.
0)
Note: you may need to restart the kernel to use updated packages.
```

3. BeautifulSoup:

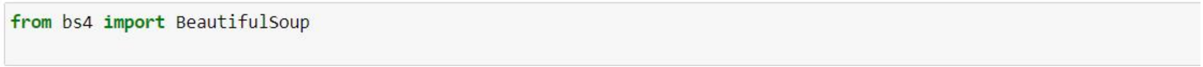
BeautifulSoup is a Python library for parsing HTML and XML documents. It creates parse trees from the documents that can be used to extract data from HTML and XML files with a simple and intuitive API. BeautifulSoup is commonly used for web scraping and data extraction.

pip install beautifulsoup4
from bs4 import BeautifulSoup



```
pip install beautifulsoup4
```

```
Requirement already satisfied: beautifulsoup4 in c:\users\hp\anaconda3\lib\site-packages (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\hp\anaconda3\lib\site-packages (from beautifulsoup4) (2.4)
Note: you may need to restart the kernel to use updated packages.
```

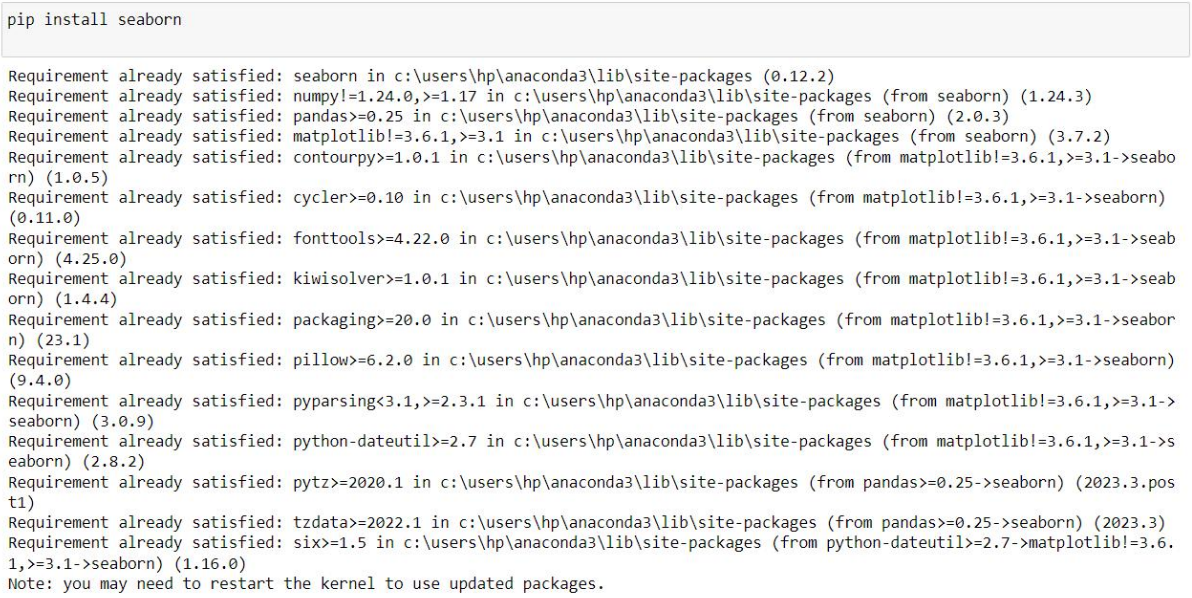


```
from bs4 import BeautifulSoup
```

4. Seaborn:

Seaborn is a library for creating attractive and informative statistical graphics in Python. The library is built on top of Matplotlib and provides a high-level interface for creating complex visualizations, such as heat maps, violin plots, and scatter plots. Seaborn is particularly well-suited for visualizing complex datasets and is often used in data exploration and analysis.

pip install seaborn



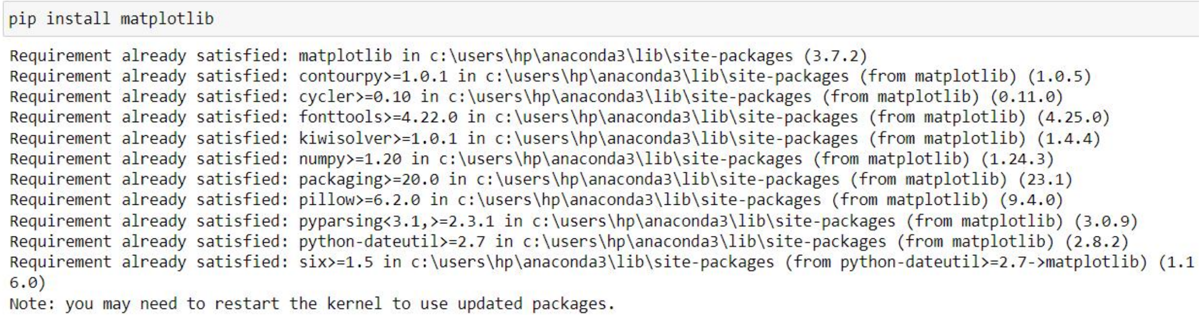
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\hp\anaconda3\lib\site-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\users\hp\anaconda3\lib\site-packages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in c:\users\hp\anaconda3\lib\site-packages (from seaborn) (2.0.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\hp\anaconda3\lib\site-packages (from seaborn) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seabo
rn) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seab
orn) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seab
orn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seabor
n) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->
seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->s
eaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=0.25->seaborn) (2023.3.pos
t1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.
1,>=3.1->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

3. Matplotlib:

Matplotlib is a plotting library for Python that provides an extensive API for creating static, animated, and interactive visualizations. The library is highly customizable, and users can create a wide range of plots, including line plots, scatter plots, bar plots, histograms, and heat maps. Matplotlib is a great tool for data visualization and is widely used in data analysis, scientific computing, and machine learning.

pip install matplotlib



```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib\site-packages (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.24.3)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.1
6.0)
Note: you may need to restart the kernel to use updated packages.
```

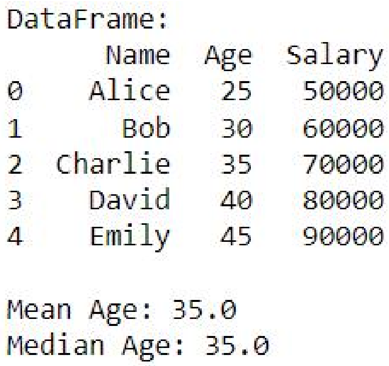
Upload the data and explore the installed packages.

Exploring all five installed packages:
Code:

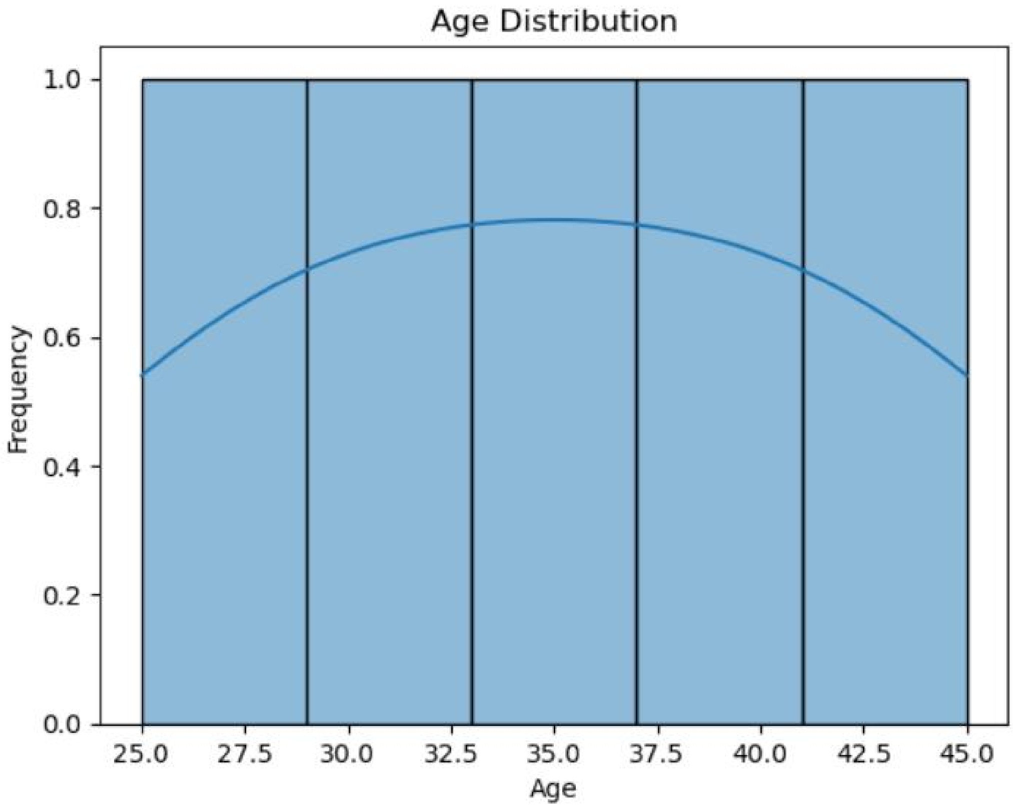
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from bs4 import BeautifulSoup

# Generate some sample data
data = {
   'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],
   'Age': [25, 30, 35, 40, 45],
   'Salary': [50000, 60000, 70000, 80000, 90000]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Display the DataFrame
print("DataFrame:")
print(df)

# Calculate mean and median of the 'Age' column using numpy
age_mean = np.mean(df['Age'])
age_median = np.median(df['Age'])
print("\nMean Age:", age_mean)
print("Median Age:", age_median)

# Visualize the distribution of the 'Age' column using seaborn
sns.histplot(df['Age'], bins=5, kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

# Parse a simple HTML snippet using BeautifulSoup
html_doc = """
<html>
<body>
<h1>Welcome to BeautifulSoup</h1>
<p>This is a simple HTML document.</p>
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
```

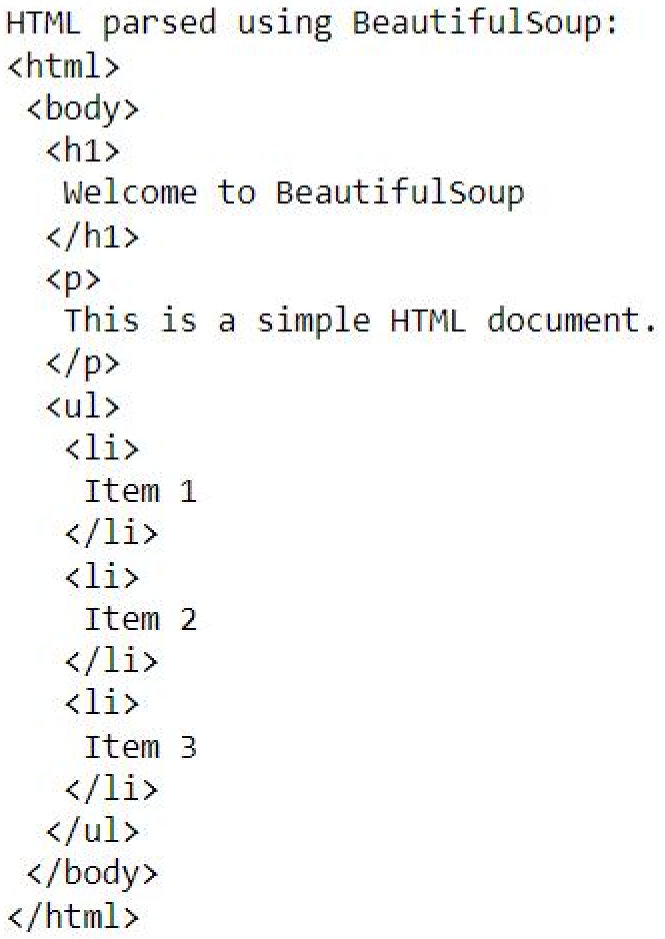
```
</ul>
</body>
</html>
"""

soup = BeautifulSoup(html_doc, 'html.parser')
print("\nHTML parsed using BeautifulSoup:")
print(soup.prettify())
```

Output:



```
DataFrame:
      Name  Age  Salary
0    Alice   25   50000
1      Bob   30   60000
2  Charlie   35   70000
3    David   40   80000
4    Emily   45   90000

Mean Age: 35.0
Median Age: 35.0
```





```
HTML parsed using BeautifulSoup:
<html>
 <body>
  <h1>
   Welcome to BeautifulSoup
  </h1>
  <p>
   This is a simple HTML document.
  </p>
  <ul>
   <li>
    Item 1
   </li>
   <li>
    Item 2
   </li>
   <li>
    Item 3
   </li>
  </ul>
 </body>
</html>
```

Practice various plot on given datasets.

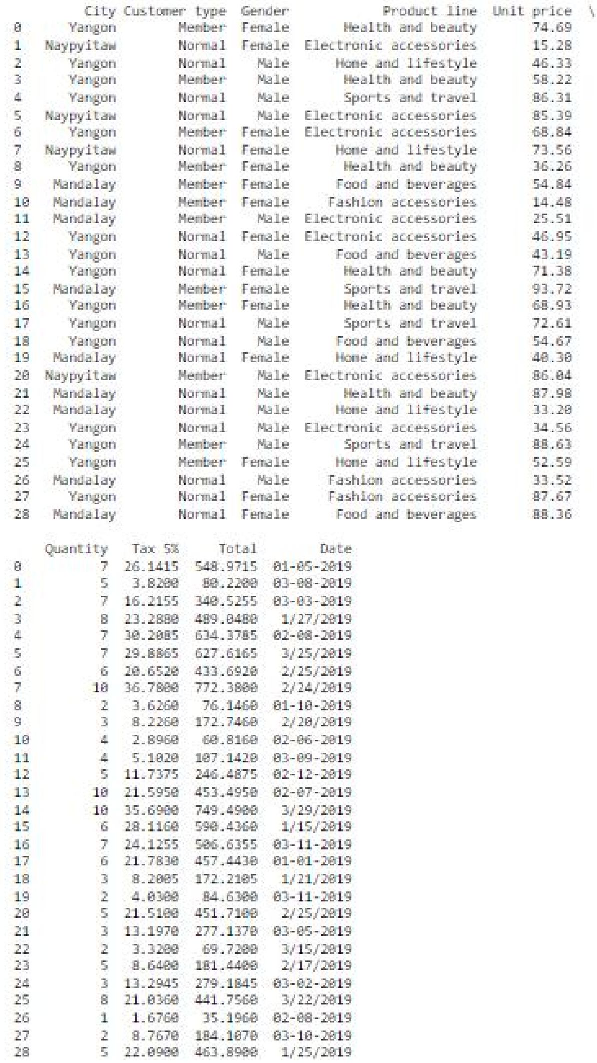
- **Line graph**
- **Data points in graph**
- **Bar Plots**
- **Histograms**
- **Pie Plots**
- **Area Plots**
- **Scatter Plots**
- **Time Series Graph**

Uploading dataset:

Code:

```
import pandas as pd
file_path = "supermarket_sales - Sheet1.csv"
sales_df = pd.read_csv(file_path)
print(sales_df)
```

Output:



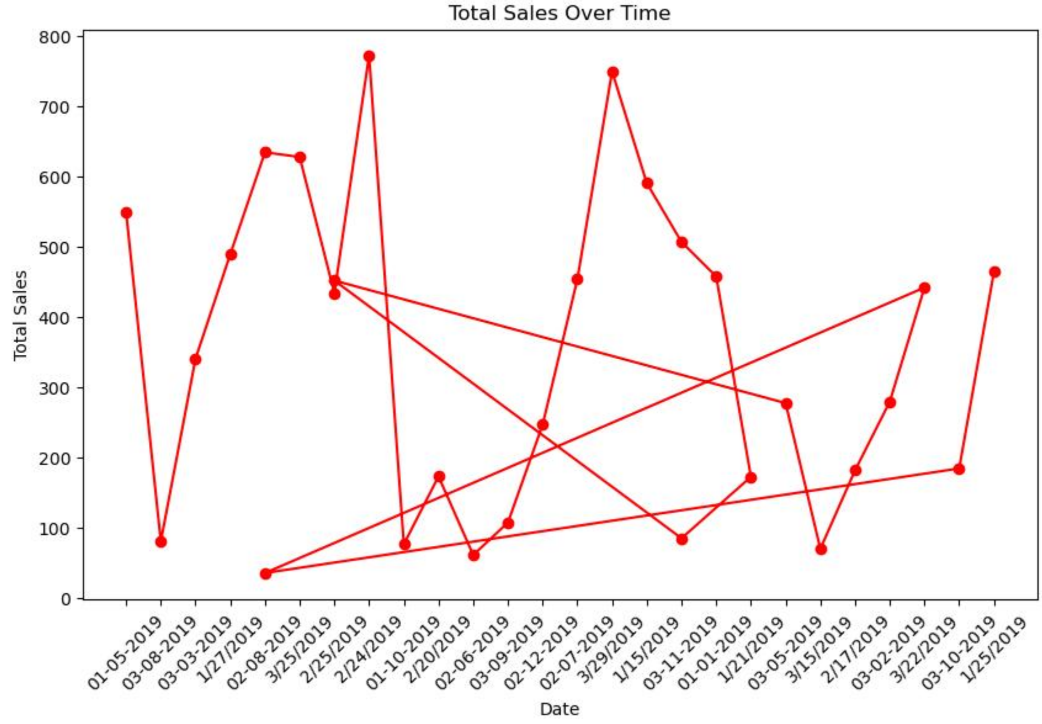
```
        City Customer type  Gender            Product line  Unit price  \
0     Yangon        Member  Female       Health and beauty       74.69
1  Naypyitaw        Normal  Female  Electronic accessories       15.28
2     Yangon        Normal    Male      Home and lifestyle       46.33
3     Yangon        Member    Male       Health and beauty       58.22
4     Yangon        Normal    Male       Sports and travel       86.31
5  Naypyitaw        Normal    Male  Electronic accessories       85.39
6     Yangon        Member  Female  Electronic accessories       68.84
7  Naypyitaw        Normal  Female      Home and lifestyle       73.56
8     Yangon        Member  Female       Health and beauty       36.26
9   Mandalay        Member  Female      Food and beverages       54.84
10  Mandalay        Member  Female     Fashion accessories       14.48
11  Mandalay        Member    Male  Electronic accessories       25.51
12    Yangon        Normal  Female  Electronic accessories       46.95
13    Yangon        Normal    Male      Food and beverages       43.19
14    Yangon        Normal  Female       Health and beauty       71.38
15  Mandalay        Member  Female       Sports and travel       93.72
16    Yangon        Member  Female       Health and beauty       68.93
17    Yangon        Normal    Male       Sports and travel       72.61
18    Yangon        Normal    Male      Food and beverages       54.67
19  Mandalay        Normal  Female      Home and lifestyle       40.30
20 Naypyitaw        Member    Male  Electronic accessories       86.04
21  Mandalay        Normal    Male       Health and beauty       87.98
22  Mandalay        Normal    Male      Home and lifestyle       33.20
23    Yangon        Normal    Male  Electronic accessories       34.56
24    Yangon        Member    Male       Sports and travel       88.63
25    Yangon        Member  Female      Home and lifestyle       52.59
26  Mandalay        Normal    Male     Fashion accessories       33.52
27    Yangon        Normal  Female     Fashion accessories       87.67
28  Mandalay        Normal  Female      Food and beverages       88.36

    Quantity   Tax 5%     Total        Date
0          7  26.1415  548.9715  01-05-2019
1          5   3.8200   80.2200  03-08-2019
2          7  16.2155  340.5255  03-03-2019
3          8  23.2880  489.0480   1/27/2019
4          7  30.2085  634.3785  02-08-2019
5          7  29.8865  627.6165   3/25/2019
6          6  20.6520  433.6920   2/25/2019
7         10  36.7800  772.3800   2/24/2019
8          2   3.6260   76.1460  01-10-2019
9          3   8.2260  172.7460   2/20/2019
10         4   2.8960   60.8160  02-06-2019
11         4   5.1020  107.1420  03-09-2019
12         5  11.7375  246.4875  02-12-2019
13        10  21.5950  453.4950  02-07-2019
14        10  35.6900  749.4900   3/29/2019
15         6  28.1160  590.4360   1/15/2019
16         7  24.1255  506.6355  03-11-2019
17         6  21.7830  457.4430  01-01-2019
18         3   8.2005  172.2105   1/21/2019
19         2   4.0300   84.6300  03-11-2019
20         5  21.5100  451.7100   2/25/2019
21         3  13.1970  277.1370  03-05-2019
22         2   3.3200   69.7200   3/15/2019
23         5   8.6400  181.4400   2/17/2019
24         3  13.2945  279.1845  03-02-2019
25         8  21.0360  441.7560   3/22/2019
26         1   1.6760   35.1960  02-08-2019
27         2   8.7670  184.1070  03-10-2019
28         5  22.0900  463.8900   1/25/2019
```

1. Line graph:

Code:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.plot(sales_df["Date"], sales_df["Total"], marker='o', color='r', linestyle='-')
plt.title('Total Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.grid(False)
plt.show()
```

Output:

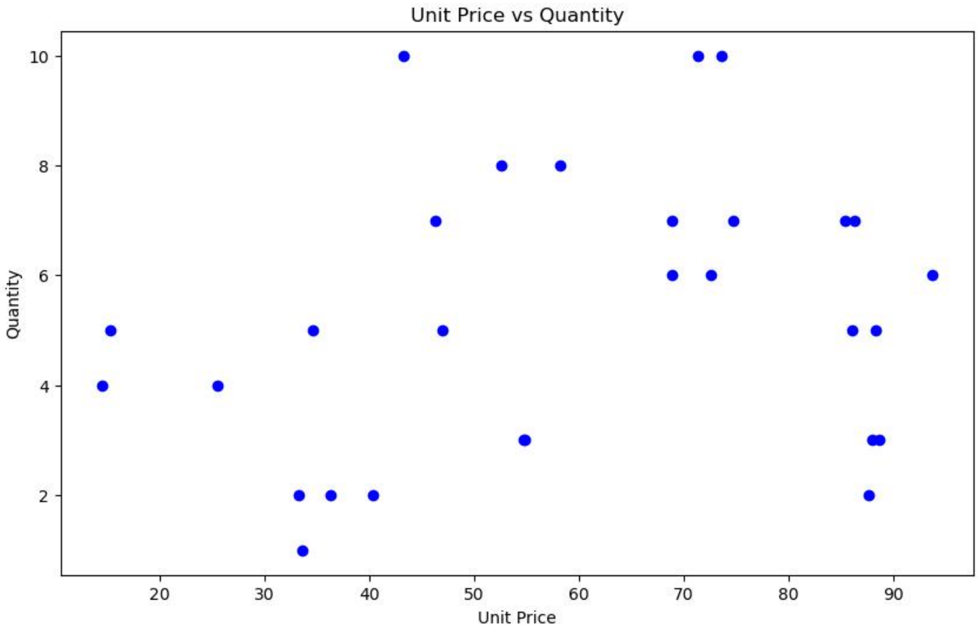


2. Data points in graph:

Code:

```
plt.figure(figsize=(10, 6))
plt.scatter(sales_df["Unit price"], sales_df["Quantity"], color='b')
plt.title('Unit Price vs Quantity')
plt.xlabel('Unit Price')
plt.ylabel('Quantity')
plt.grid(False)
plt.show()
```

Output:

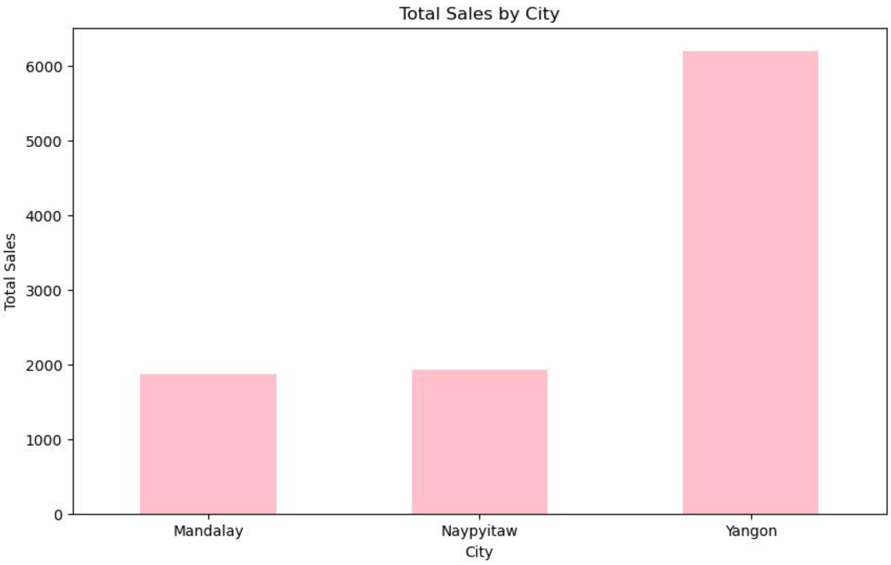


3. Bar plots:

Code:

```
total_sales_by_city = sales_df.groupby("City")["Total"].sum()
plt.figure(figsize=(10, 6))
total_sales_by_city.plot(kind='bar', color='pink')
plt.title('Total Sales by City')
plt.xlabel('City')
plt.ylabel('Total Sales')
plt.xticks(rotation=0)
plt.grid(False)
plt.show()
```

Output:

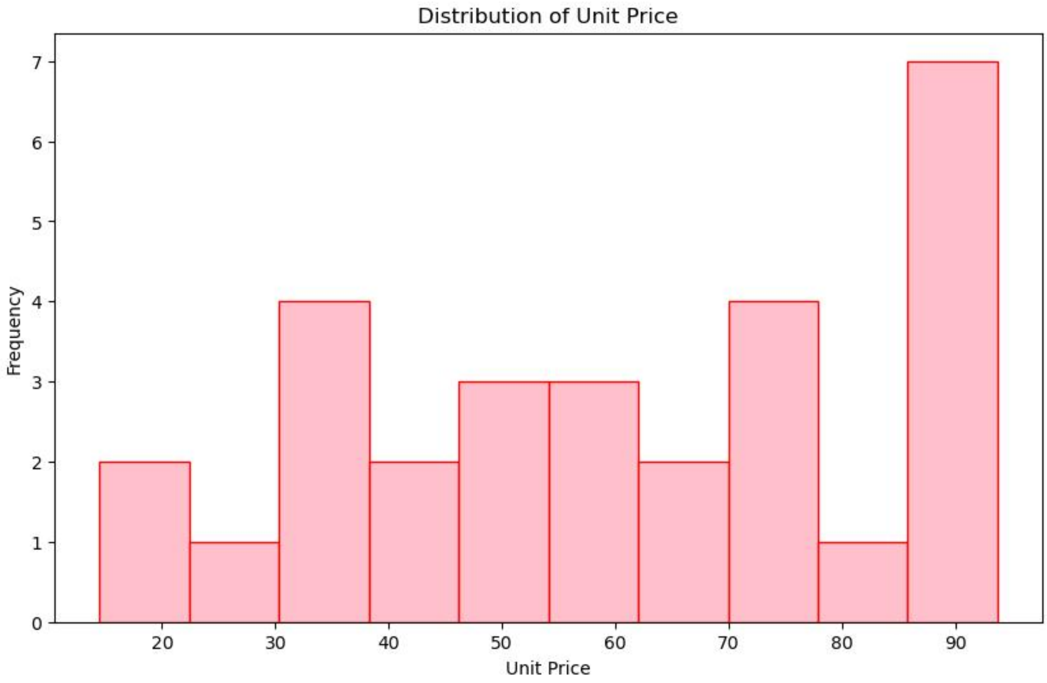


4. Histograms

Code:

```
plt.figure(figsize=(10, 6))
plt.hist(sales_df["Unit price"], bins=10, color='pink',edgecolor='red')
plt.title('Distribution of Unit Price')
plt.xlabel('Unit Price')
plt.ylabel('Frequency')
plt.grid(False)
plt.show()
```

Output:

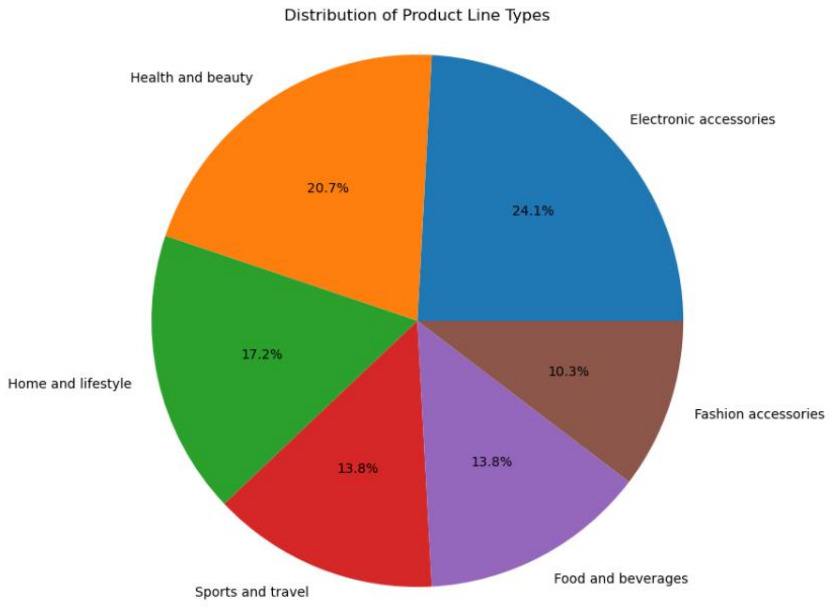


5. Pie plots:

Code:

```
product_line_counts = sales_df["Product line"].value_counts()
plt.figure(figsize=(10, 8))
plt.pie(product_line_counts, labels=product_line_counts.index, autopct='%1.1f%%')
plt.title('Distribution of Product Line Types')
plt.axis('equal')
plt.show()
```

Output:

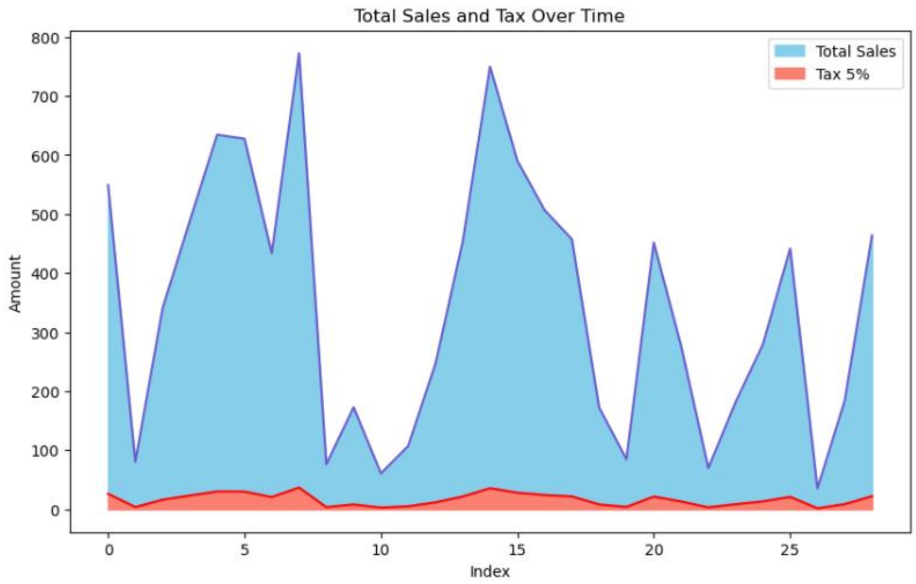


6. Area plots:

Code:

```
plt.figure(figsize=(10, 6))
plt.fill_between(sales_df.index, sales_df["Total"], color='skyblue',label='Total Sales')
plt.plot(sales_df.index, sales_df["Total"], color='Slateblue')
plt.fill_between(sales_df.index, sales_df["Tax 5%"], color='salmon', label='Tax 5%')
plt.plot(sales_df.index, sales_df["Tax 5%"], color='red')
plt.title('Total Sales and Tax Over Time')
plt.xlabel('Index')
plt.ylabel('Amount')
plt.legend()
plt.grid(False)
plt.show()
```

Output:

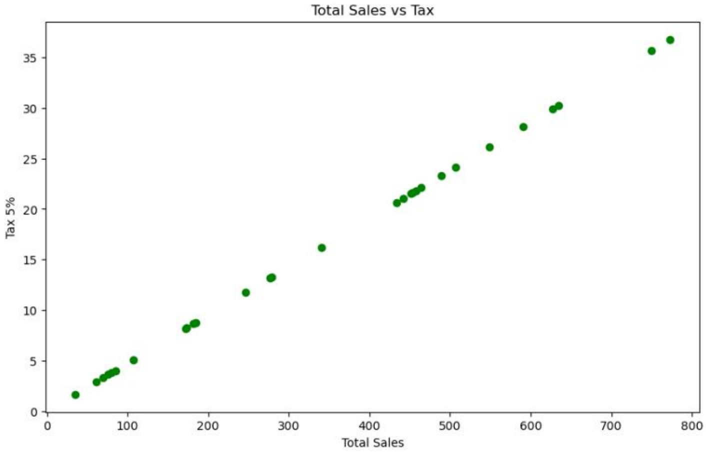


7. Scatter plot:

Code:

```
plt.figure(figsize=(10, 6))
plt.scatter(sales_df["Total"], sales_df["Tax 5%"], color='green')
plt.title('Total Sales vs Tax')
plt.xlabel('Total Sales')
plt.ylabel('Tax 5%')
plt.grid(False)
plt.show()
```

Output:

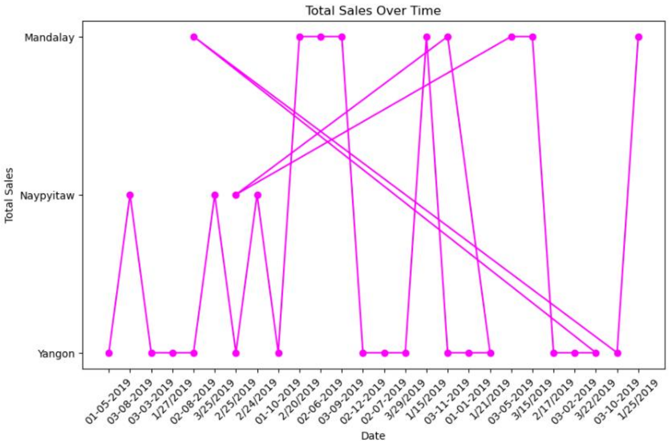


8. Time Series Graph:

Code:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.plot(sales_df["Date"], sales_df["City"], marker='o', color='magenta')
plt.title('Total Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.grid(False)
plt.show()
```

Output:



Collect the data from social media on YouTube using YouTube API.

Generating YouTube API:

YouTube Data API is an instrument that allows developers all around the world to effectively work with the YouTube data and create their own apps based on it. Google YouTube API works with a quota to guarantee that developers use the service properly and do not create any app that makes service quality worse or limits access for others. All API requests, including invalid requests, cost at least a one-point quota.

YouTube Data API key gives wide opportunities for interacting with YouTube content:

Collect information about specific channels

Get an access to videos and playlists

Explore views, descriptions, comments, likes and dislikes

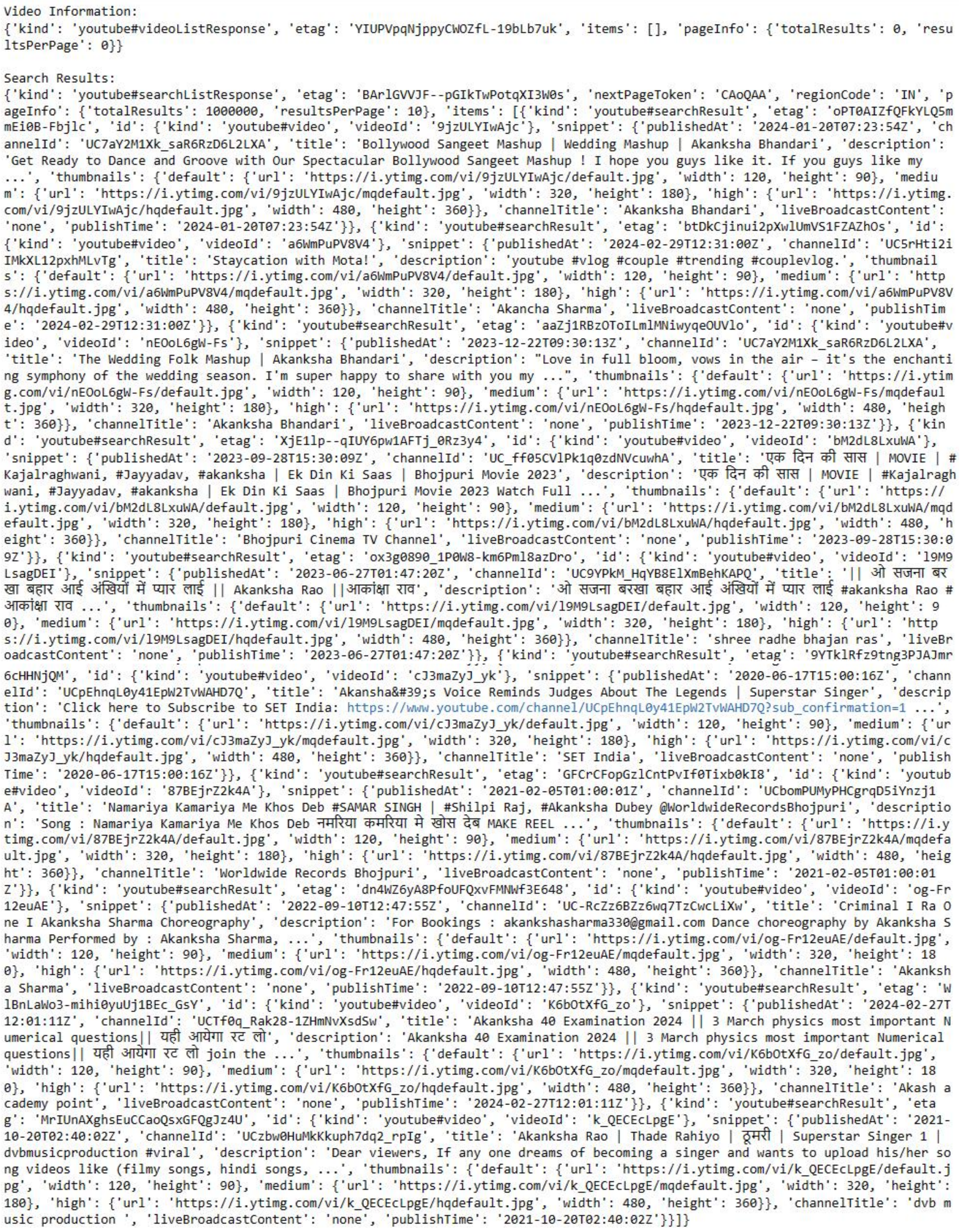
Code:

This code will retrieve information about a specific video with the provided ID and also search for videos related to the specified query.

Once you receive the data from the API, you'll need to parse and process it according to your requirements. You may want to store it in a database, analyze it, or display it on a website or application.

```
from googleapiclient.discovery import build
api_key = "AIzaSyDh4UWPLG22TIBA5lLxN7N4zcPPjJAVtvs"
youtube = build("youtube", "v3", developerKey=api_key)
video_id = "7j1uoMvpKv0?si=Iyfb9k724LI-BycC"
video_response = youtube.videos().list(
   part="snippet",
   id=video_id
).execute()
search_query = "akanksha"
search_response = youtube.search().list(
   q=search_query,
   part="snippet",
   type="video",
   maxResults=10
).execute()
print("Video Information:")
print(video_response)
print("\nSearch Results:")
print(search_response)
```

Output:



```
Video Information:
{'kind': 'youtube#videoListResponse', 'etag': 'YIUPVpqNjppyCWOZfL-19bLb7uk', 'items': [], 'pageInfo': {'totalResults': 0, 'resu
ltsPerPage': 0}}

Search Results:
{'kind': 'youtube#searchListResponse', 'etag': 'BArlGVVJF--pGIkTwPotqXI3W0s', 'nextPageToken': 'CAoQAA', 'regionCode': 'IN', 'p
ageInfo': {'totalResults': 1000000, 'resultsPerPage': 10}, 'items': [{'kind': 'youtube#searchResult', 'etag': 'oPT0AIZfQFkYLQ5m
mEi0B-Fbjlc', 'id': {'kind': 'youtube#video', 'videoId': '9jzULYIwAjc'}, 'snippet': {'publishedAt': '2024-01-20T07:23:54Z', 'ch
annelId': 'UC7aY2M1Xk_saR6RzD6L2LXA', 'title': 'Bollywood Sangeet Mashup | Wedding Mashup | Akanksha Bhandari', 'description':
'Get Ready to Dance and Groove with Our Spectacular Bollywood Sangeet Mashup ! I hope you guys like it. If you guys like my
...', 'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/9jzULYIwAjc/default.jpg', 'width': 120, 'height': 90}, 'mediu
m': {'url': 'https://i.ytimg.com/vi/9jzULYIwAjc/mqdefault.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'https://i.ytimg.
com/vi/9jzULYIwAjc/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'Akanksha Bhandari', 'liveBroadcastContent':
'none', 'publishTime': '2024-01-20T07:23:54Z'}}, {'kind': 'youtube#searchResult', 'etag': 'btDkCjinui2pXwlUmVS1FZAZhOs', 'id':
{'kind': 'youtube#video', 'videoId': 'a6WmPuPV8V4'}, 'snippet': {'publishedAt': '2024-02-29T12:31:00Z', 'channelId': 'UC5rHti2i
IMkXL12pxhMLvTg', 'title': 'Staycation with Mota!', 'description': 'youtube #vlog #couple #trending #couplevlog.', 'thumbnail
s': {'default': {'url': 'https://i.ytimg.com/vi/a6WmPuPV8V4/default.jpg', 'width': 120, 'height': 90}, 'medium': {'url': 'http
s://i.ytimg.com/vi/a6WmPuPV8V4/mqdefault.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'https://i.ytimg.com/vi/a6WmPuPV8V
4/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'Akancha Sharma', 'liveBroadcastContent': 'none', 'publishTim
e': '2024-02-29T12:31:00Z'}}, {'kind': 'youtube#searchResult', 'etag': 'aaZj1RBzOToILmlMNiwyqeOUVlo', 'id': {'kind': 'youtube#v
ideo', 'videoId': 'nEOoL6gW-Fs'}, 'snippet': {'publishedAt': '2023-12-22T09:30:13Z', 'channelId': 'UC7aY2M1Xk_saR6RzD6L2LXA',
'title': 'The Wedding Folk Mashup | Akanksha Bhandari', 'description': "Love in full bloom, vows in the air – it's the enchanti
ng symphony of the wedding season. I'm super happy to share with you my ...", 'thumbnails': {'default': {'url': 'https://i.ytim
g.com/vi/nEOoL6gW-Fs/default.jpg', 'width': 120, 'height': 90}, 'medium': {'url': 'https://i.ytimg.com/vi/nEOoL6gW-Fs/mqdefaul
t.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'https://i.ytimg.com/vi/nEOoL6gW-Fs/hqdefault.jpg', 'width': 480, 'heigh
t': 360}}, 'channelTitle': 'Akanksha Bhandari', 'liveBroadcastContent': 'none', 'publishTime': '2023-12-22T09:30:13Z'}}, {'kin
d': 'youtube#searchResult', 'etag': 'XjE1lp--qIUY6pw1AFTj_0Rz3y4', 'id': {'kind': 'youtube#video', 'videoId': 'bM2dL8LxuWA'},
'snippet': {'publishedAt': '2023-09-28T15:30:09Z', 'channelId': 'UC_ff05CVlPk1q0zdNVcuwhA', 'title': 'एक दिन की सास | MOVIE | #
Kajalraghwani, #Jayyadav, #akanksha | Ek Din Ki Saas | Bhojpuri Movie 2023', 'description': 'एक दिन की सास | MOVIE | #Kajalragh
wani, #Jayyadav, #akanksha | Ek Din Ki Saas | Bhojpuri Movie 2023 Watch Full ...', 'thumbnails': {'default': {'url': 'https://
i.ytimg.com/vi/bM2dL8LxuWA/default.jpg', 'width': 120, 'height': 90}, 'medium': {'url': 'https://i.ytimg.com/vi/bM2dL8LxuWA/mqd
efault.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'https://i.ytimg.com/vi/bM2dL8LxuWA/hqdefault.jpg', 'width': 480, 'h
eight': 360}}, 'channelTitle': 'Bhojpuri Cinema TV Channel', 'liveBroadcastContent': 'none', 'publishTime': '2023-09-28T15:30:0
9Z'}}, {'kind': 'youtube#searchResult', 'etag': 'ox3g0890_1P0W8-km6Pml8azDro', 'id': {'kind': 'youtube#video', 'videoId': 'l9M9
LsagDEI'}, 'snippet': {'publishedAt': '2023-06-27T01:47:20Z', 'channelId': 'UC9YPkM_HqYB8ElXmBehKAPQ', 'title': '|| ओ सजना बर
खा बहार आई अंखियों में प्यार लाई || Akanksha Rao ||आकांक्षा राव', 'description': 'ओ सजना बरखा बहार आई अंखियों में प्यार लाई #akanksha Rao #
आकांक्षा राव ...', 'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/l9M9LsagDEI/default.jpg', 'width': 120, 'height': 9
0}, 'medium': {'url': 'https://i.ytimg.com/vi/l9M9LsagDEI/mqdefault.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'http
s://i.ytimg.com/vi/l9M9LsagDEI/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'shree radhe bhajan ras', 'liveBr
oadcastContent': 'none', 'publishTime': '2023-06-27T01:47:20Z'}}, {'kind': 'youtube#searchResult', 'etag': '9YTklRfz9tng3PJAJmr
6cHHNjQM', 'id': {'kind': 'youtube#video', 'videoId': 'cJ3maZyJ_yk'}, 'snippet': {'publishedAt': '2020-06-17T15:00:16Z', 'chann
elId': 'UCpEhnqL0y41EpW2TvWAHD7Q', 'title': 'Akansha&#39;s Voice Reminds Judges About The Legends | Superstar Singer', 'descrip
tion': 'Click here to Subscribe to SET India: https://www.youtube.com/channel/UCpEhnqL0y41EpW2TvWAHD7Q?sub_confirmation=1 ...',
'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/cJ3maZyJ_yk/default.jpg', 'width': 120, 'height': 90}, 'medium': {'ur
l': 'https://i.ytimg.com/vi/cJ3maZyJ_yk/mqdefault.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'https://i.ytimg.com/vi/c
J3maZyJ_yk/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'SET India', 'liveBroadcastContent': 'none', 'publish
Time': '2020-06-17T15:00:16Z'}}, {'kind': 'youtube#searchResult', 'etag': 'GFCrCFopGzlCntPvIf0Tixb0kI8', 'id': {'kind': 'youtub
e#video', 'videoId': '87BEjrZ2k4A'}, 'snippet': {'publishedAt': '2021-02-05T01:00:01Z', 'channelId': 'UCbomPUMyPHCgrqD5iYnzj1
A', 'title': 'Namariya Kamariya Me Khos Deb #SAMAR SINGH | #Shilpi Raj, #Akanksha Dubey @WorldwideRecordsBhojpuri', 'descriptio
n': 'Song : Namariya Kamariya Me Khos Deb नमरिया कमरिया मे खोस देब MAKE REEL ...', 'thumbnails': {'default': {'url': 'https://i.y
timg.com/vi/87BEjrZ2k4A/default.jpg', 'width': 120, 'height': 90}, 'medium': {'url': 'https://i.ytimg.com/vi/87BEjrZ2k4A/mqdefa
ult.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'https://i.ytimg.com/vi/87BEjrZ2k4A/hqdefault.jpg', 'width': 480, 'heig
ht': 360}}, 'channelTitle': 'Worldwide Records Bhojpuri', 'liveBroadcastContent': 'none', 'publishTime': '2021-02-05T01:00:01
Z'}}, {'kind': 'youtube#searchResult', 'etag': 'dn4WZ6yA8PfoUFQxvFMNWf3E648', 'id': {'kind': 'youtube#video', 'videoId': 'og-Fr
12euAE'}, 'snippet': {'publishedAt': '2022-09-10T12:47:55Z', 'channelId': 'UC-RcZz6BZz6wq7TzCwcLiXw', 'title': 'Criminal I Ra O
ne I Akanksha Sharma Choreography', 'description': 'For Bookings : akankshasharma330@gmail.com Dance choreography by Akanksha S
harma Performed by : Akanksha Sharma, ...', 'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/og-Fr12euAE/default.jpg',
'width': 120, 'height': 90}, 'medium': {'url': 'https://i.ytimg.com/vi/og-Fr12euAE/mqdefault.jpg', 'width': 320, 'height': 18
0}, 'high': {'url': 'https://i.ytimg.com/vi/og-Fr12euAE/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'Akanksh
a Sharma', 'liveBroadcastContent': 'none', 'publishTime': '2022-09-10T12:47:55Z'}}, {'kind': 'youtube#searchResult', 'etag': 'W
lBnLaWo3-mihi0yuUj1BEc_GsY', 'id': {'kind': 'youtube#video', 'videoId': 'K6bOtXfG_zo'}, 'snippet': {'publishedAt': '2024-02-27T
12:01:11Z', 'channelId': 'UCTf0q_Rak28-1ZHmNvXsdSw', 'title': 'Akanksha 40 Examination 2024 || 3 March physics most important N
umerical questions|| यही आयेगा रट लो', 'description': 'Akanksha 40 Examination 2024 || 3 March physics most important Numerical
questions|| यही आयेगा रट लो join the ...', 'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/K6bOtXfG_zo/default.jpg',
'width': 120, 'height': 90}, 'medium': {'url': 'https://i.ytimg.com/vi/K6bOtXfG_zo/mqdefault.jpg', 'width': 320, 'height': 18
0}, 'high': {'url': 'https://i.ytimg.com/vi/K6bOtXfG_zo/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'Akash a
cademy point', 'liveBroadcastContent': 'none', 'publishTime': '2024-02-27T12:01:11Z'}}, {'kind': 'youtube#searchResult', 'eta
g': 'MrIUnAXghsEuCCaoQsxGFQgJz4U', 'id': {'kind': 'youtube#video', 'videoId': 'k_QECEcLpgE'}, 'snippet': {'publishedAt': '2021-
10-20T02:40:02Z', 'channelId': 'UCzbw0HuMkKkuph7dq2_rpIg', 'title': 'Akanksha Rao | Thade Rahiyo | ठुमरी | Superstar Singer 1 |
dvbmusicproduction #viral', 'description': 'Dear viewers, If any one dreams of becoming a singer and wants to upload his/her so
ng videos like (filmy songs, hindi songs, ...', 'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/k_QECEcLpgE/default.j
pg', 'width': 120, 'height': 90}, 'medium': {'url': 'https://i.ytimg.com/vi/k_QECEcLpgE/mqdefault.jpg', 'width': 320, 'height':
180}, 'high': {'url': 'https://i.ytimg.com/vi/k_QECEcLpgE/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'dvb m
usic production ', 'liveBroadcastContent': 'none', 'publishTime': '2021-10-20T02:40:02Z'}}]}
```

Analyze the YouTube Trending Videos on Indian Videos.

Goals of the Analysis on Indian Movies dataset.

a. How many views do our trending videos have?

b. Do most of them have many views? Is having many views required for a video to become trending?

c. The same questions above but applied to likes and comment count instead of views.

d. Which video remained the most on the trending-videos list?

e. How many trending videos contain a fully capitalized word in their titles?

f. What are the lengths of trending video titles? Is this length related to the video becoming trendy?

g. How are views, likes, dislikes, comment count, title length, and other attributes correlate with (relate to) each other? How are they connected?

h. What are the most common words in trending video titles?

i. Which YouTube channels have the largest number of trending videos?

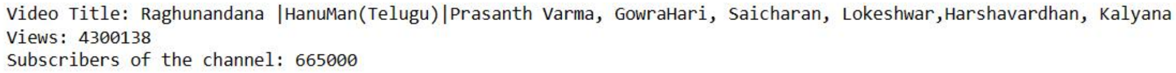
j. Which video category (e.g. Entertainment, Gaming, Comedy, etc.) has the largest number of trending videos?

k. When were the trending videos published? On which days of the week? at which times of the day?

Code:

```
from googleapiclient.discovery import build
api_key = "AIzaSyDh4UWPLG22TIBA5lLxN7N4zcPPjJAVtvs"
youtube = build('youtube', 'v3', developerKey=api_key)
def get_video_info(video_id):
    video_response = youtube.videos().list(
        part="snippet, statistics",
        id=video_id
    ).execute()
    if 'items' in video_response:
        video_info = video_response['items'][0]
        snippet = video_info['snippet']
        statistics = video_info['statistics']
        title = snippet['title']
        channel_id = snippet['channelId']
        views = statistics.get('viewCount', 0)
        channel_response = youtube.channels().list(
            part="snippet, statistics",
            id=channel_id
        ).execute()
        if 'items' in channel_response:
            channel_info = channel_response['items'][0]
            channel_statistics = channel_info['statistics']
            subscribers = channel_statistics.get('subscriberCount', 0)
            return title, views, subscribers
        else:
            return title, views, None
    else:
        return None, None, None
video_id = "hpNNFazxKo8"
video_title, video_views, channel_subscribers = get_video_info(video_id)
print("Video Title:", video_title)
print("Views:", video_views)
print("Subscribers of the channel:", channel_subscribers)
```

Output:



```
Video Title: Raghunandana |HanuMan(Telugu)|Prasanth Varma, GowraHari, Saicharan, Lokeshwar,Harshavardhan, Kalyana
Views: 4300138
Subscribers of the channel: 665000
```

Collect the data from social media on Facebook using Facebook API.

Procedure:

As of the last update in January 2022, Facebook offers an API (Application Programming Interface) that allows developers to access certain data from the platform. However, it is essential to note that Facebook's API access is subject to their platform policies and restrictions. Also, Facebook frequently updates its API, so it is crucial to refer to their official documentation for the most up-to-date information on accessing data.

Here is a general outline of how you might collect data from Facebook using their API:

Create a Facebook Developer Account: Before you can access the Facebook API, you need to create a developer account on Facebook's Developer Portal: https://developers.facebook.com/

Create an App: Once you have a developer account, you will need to create a new app. This app will be used to authenticate your requests to the Facebook API.

Set up App Permissions: Depending on the type of data you want to access; you will need to request specific permissions from Facebook. These permissions determine what data your app can access on behalf of users.

Authenticate Your App: Facebook's API typically uses OAuth authentication. This means you will need to implement OAuth in your application to authenticate users and obtain access tokens for making API requests.

Make API Requests: Once your app is authenticated and permissions are granted, you can start making API requests to fetch the desired data. The Graph API is Facebook's primary API for accessing data, and it allows you to retrieve information about users, posts, comments, pages, and more.

Handle Rate Limits and Data Storage: Facebook's API has rate limits and data usage restrictions that you need to be aware of. Make sure to handle rate limits gracefully and consider how you will store and manage the data you collect.

Comply with Facebook's Policies: Ensure that your app complies with Facebook's platform policies and data usage terms. Violating these policies can result in your app being restricted or banned.

It is important to approach data collection from social media platforms ethically and responsibly, respecting user privacy and platform policies. Additionally, keep in mind that collecting data from Facebook may require approval and may be subject to restrictions based on Facebook's policies and regulations. Always refer to Facebook's official documentation and guidelines for the most accurate and up-to-date information.

Step-1

We need to login Facebook developer page

Step-2

Then select the app explorer option

Step :03

Then we need to create app and need to permission

Step:04

Generate the token

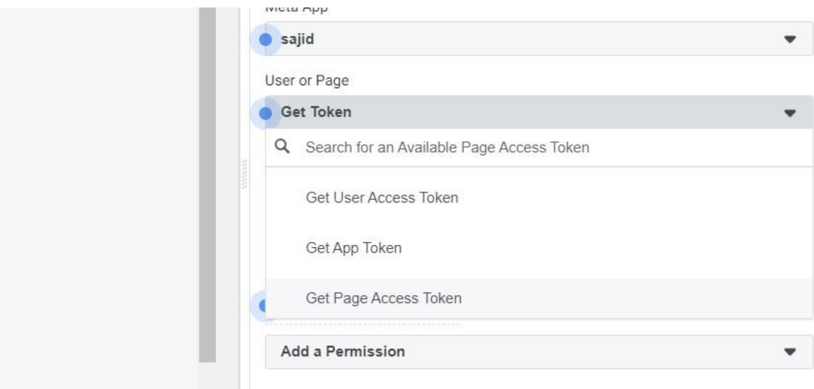
Step: 5

Then Submit the token

Step: 06

Then We got every data and we also remotely post also those account.

Select Get Page Access Token



Meta App

sajid

User or Page

Get Token

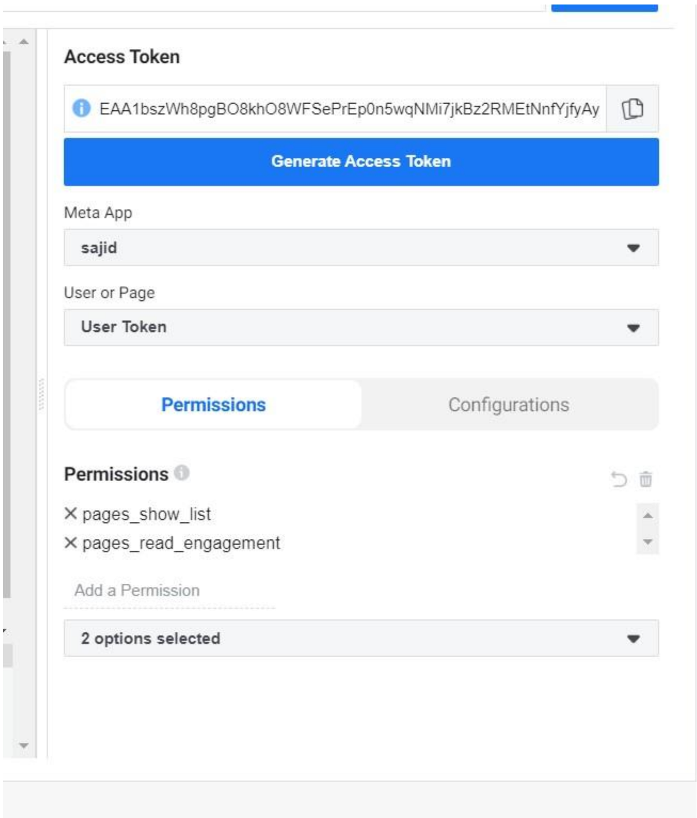
Search for an Available Page Access Token

Get User Access Token

Get App Token

Get Page Access Token

Add a Permission



Access Token

EAA1bszWh8pgBO8khO8WFSePrEp0n5wqNMi7jkBz2RMEtNnfYjfyAy

Generate Access Token

Meta App

sajid

User or Page

User Token

Permissions | Configurations

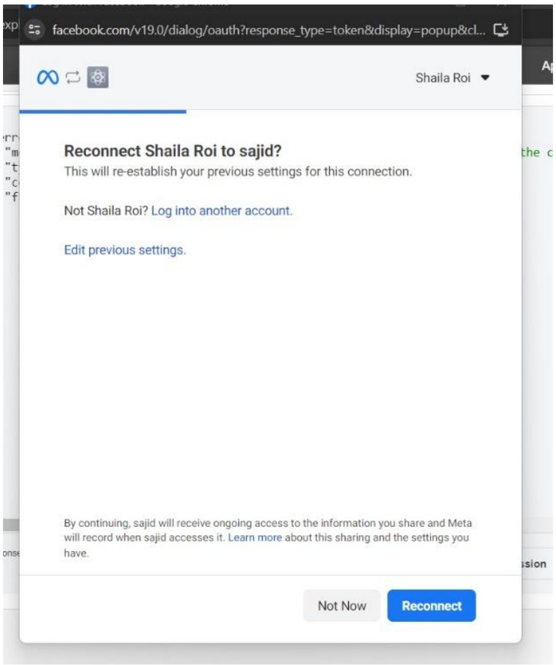
Permissions

× pages_show_list

× pages_read_engagement

Add a Permission

2 options selected



facebook.com/v19.0/dialog/oauth?response_type=token&display=popup&cl...

Shaila Roi

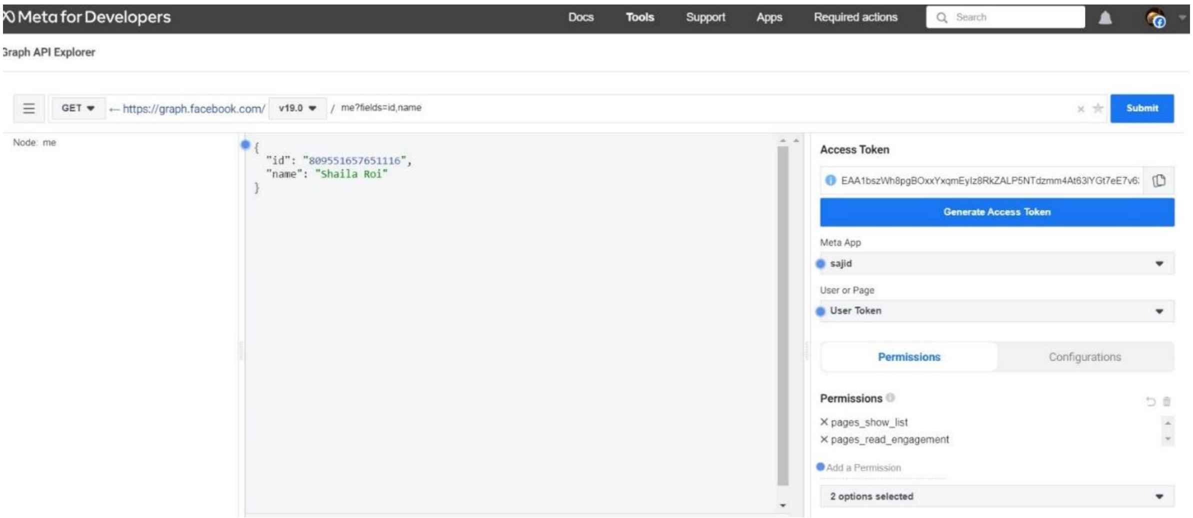
Reconnect Shaila Roi to sajid?

This will re-establish your previous settings for this connection.

Not Shaila Roi? Log into another account.

Edit previous settings.

By continuing, sajid will receive ongoing access to the information you share and Meta will record when sajid accesses it. Learn more about this sharing and the settings you have.



GET ← https://graph.facebook.com/ v19.0 / me?fields=id,name

Node: me

```
{
  "id": "809551657651116",
  "name": "Shaila Roi"
}
```

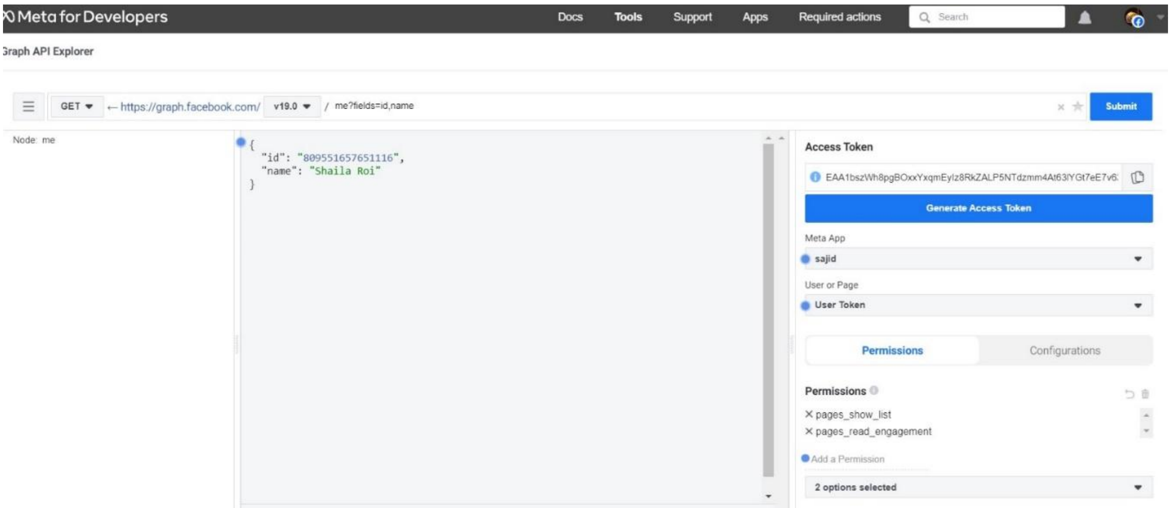
Access Token: EAA1bszWh8pgBOxxYxqmEylz8RkZALP5NTdzmm4At63lYGt7eE7v6

Meta App: sajid

User or Page: User Token

Permissions
× pages_show_list
× pages_read_engagement

2 options selected



GET ← https://graph.facebook.com/ v19.0 / me?fields=id,name

Node: me

```
{
  "id": "809551657651116",
  "name": "Shaila Roi"
}
```

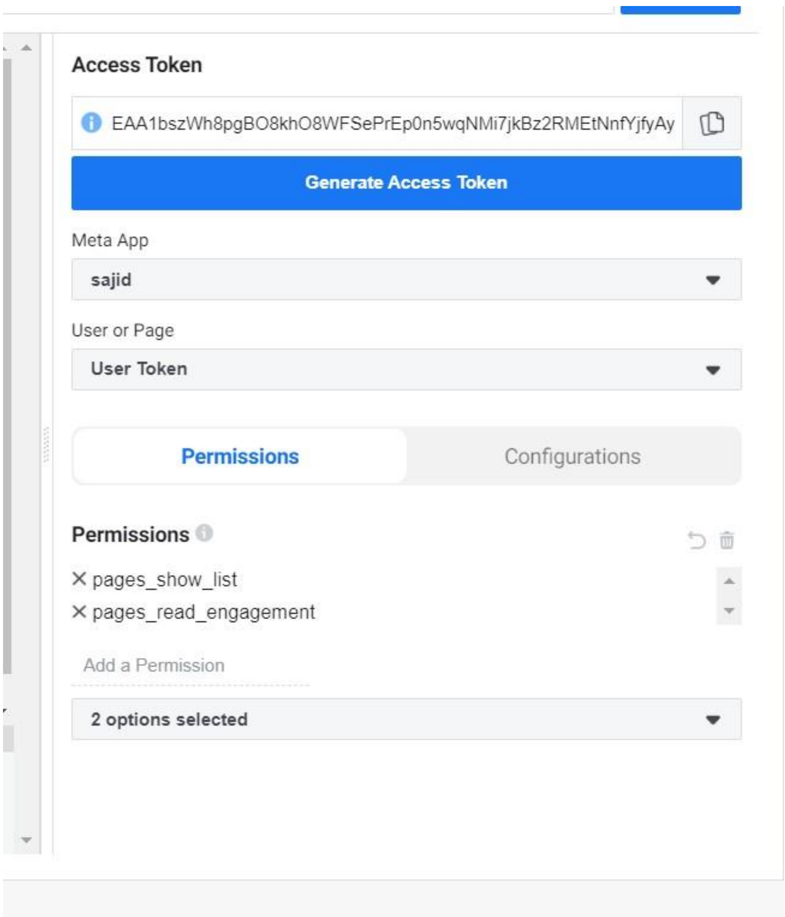
Access Token: EAA1bszWh8pgBOxxYxqmEylz8RkZALP5NTdzmm4At63lYGt7eE7v6

Meta App: sajid

User or Page: User Token

Permissions
× pages_show_list
× pages_read_engagement

2 options selected



Access Token

EAA1bszWh8pgBO8khO8WFSePrEp0n5wqNMi7jkBz2RMEtNnfYjfyAy

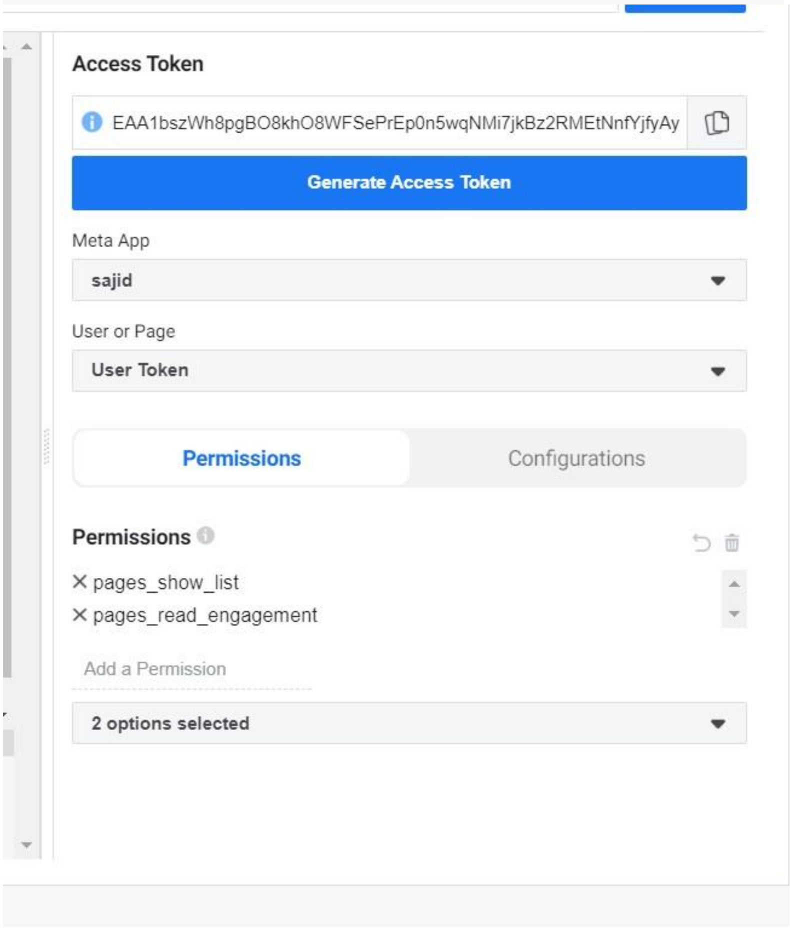
Meta App: sajid

User or Page: User Token

Permissions

- ✕ pages_show_list
- ✕ pages_read_engagement

2 options selected



Access Token

EAA1bszWh8pgBO8khO8WFSePrEp0n5wqNMi7jkBz2RMEtNnfYjfyAy

Meta App: sajid

User or Page: User Token

Permissions

- ✕ pages_show_list
- ✕ pages_read_engagement

2 options selected

Analyse the Facebook data for sentiment analysis.

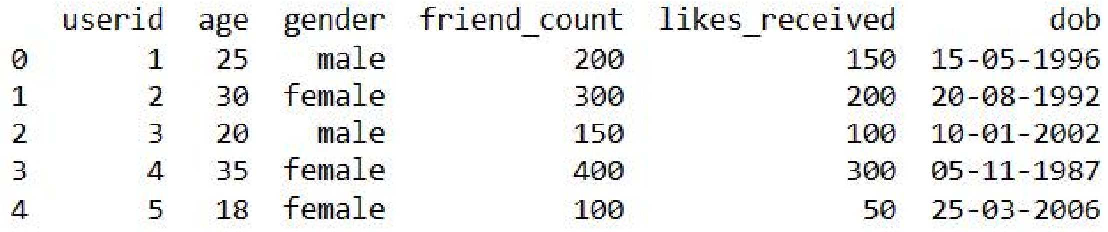
Goals of the Analysis on Facebook dataset.

a. Load the pseudo_facebook.csv dataset.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('pseudo_facebook.csv')
print(df.head())
```

Output:



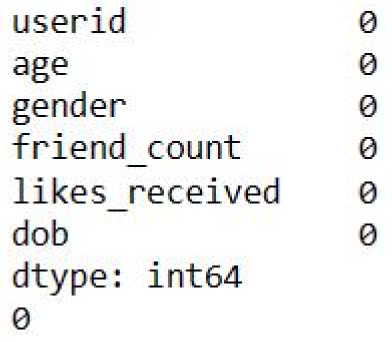
```
   userid  age  gender  friend_count  likes_received         dob
0       1   25    male           200             150  15-05-1996
1       2   30  female           300             200  20-08-1992
2       3   20    male           150             100  10-01-2002
3       4   35  female           400             300  05-11-1987
4       5   18  female           100              50  25-03-2006
```

b. Preprocess the dataset. Find null values, duplicates process those values.

Code:

```
print(df.isnull().sum())
print(df.duplicated().sum())
df = df.drop_duplicates()
```

Output:



```
userid            0
age               0
gender            0
friend_count      0
likes_received    0
dob               0
dtype: int64
0
```

c. Making a new column for different age group

Code:

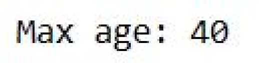
```
bins = [0, 19, 26, 36, 51, float('inf')]
labels = ['0-18', '19-25', '26-35', '36-50', '51+']
df['age_group'] = pd.cut(df['age'], bins=bins, labels=labels, right=False)
```

d. Find max age of a person

Code:

```
max_age = df['age'].max()
print("Max age:", max_age)
```

Output:



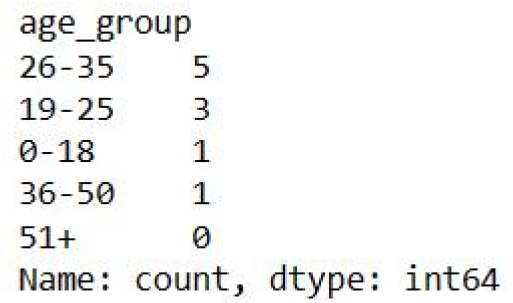
```
Max age: 40
```

e. Display the age groups analyse its count.

Code:

```
print(df['age_group'].value_counts())
```

Output:



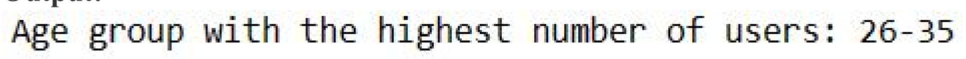
```
age_group
26-35    5
19-25    3
0-18     1
36-50    1
51+      0
Name: count, dtype: int64
```

f. Find which age group has the highest number of users

Code:

```
highest_age_group = df['age_group'].value_counts().idxmax()
print("Age group with the highest number of users:", highest_age_group)
```

Output:



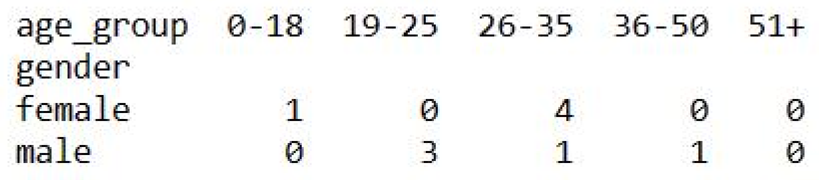
```
Age group with the highest number of users: 26-35
```

g. Which gender aged has higher number of users

Code:

```
gender_age_group_counts = df.groupby(['gender', 'age_group']).size().unstack()
print(gender_age_group_counts)
```

Output:



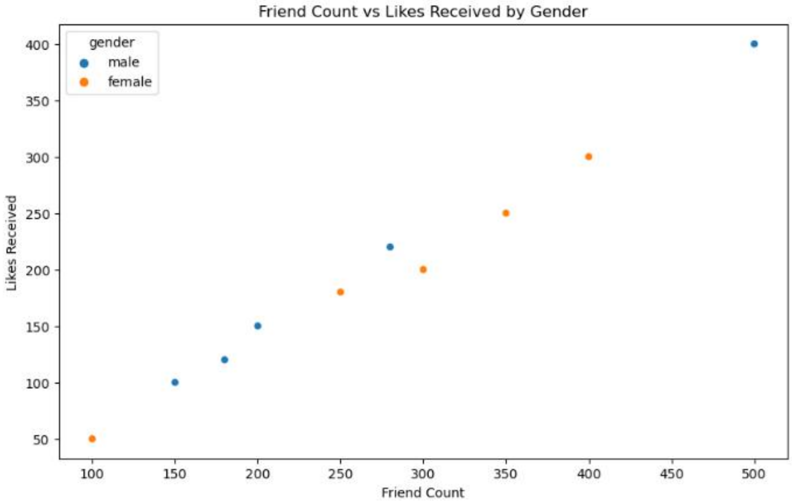
```
age_group  0-18  19-25  26-35  36-50  51+
gender
female        1      0      4      0    0
male          0      3      1      1    0
```

h. Which gender has the greatest number of friends using scatter plot?

Code:

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='friend_count', y='likes_received', data=df, hue='gender')
plt.xlabel('Friend Count')
plt.ylabel('Likes Received')
plt.title('Friend Count vs Likes Received by Gender')
plt.show()
```

Output:

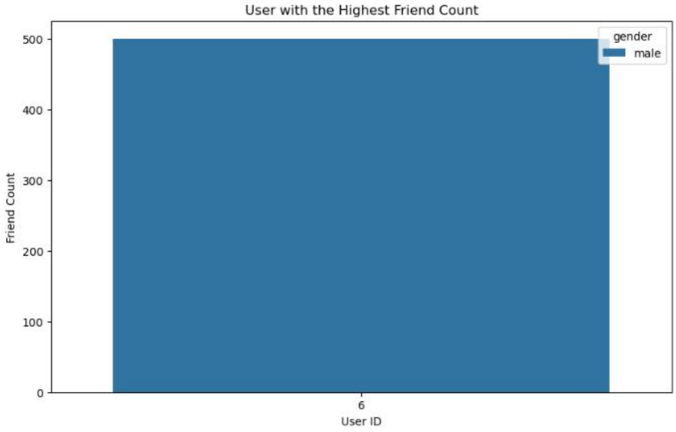


i. Who has the highest friend count? Using bar plot

Code:

```
highest_friend_count = df[df['friend_count'] == df['friend_count'].max()]
plt.figure(figsize=(10, 6))
sns.barplot(x='userid', y='friend_count', data=highest_friend_count, hue='gender')
plt.xlabel('User ID')
plt.ylabel('Friend Count')
plt.title('User with the Highest Friend Count')
plt.show()
```

Output:

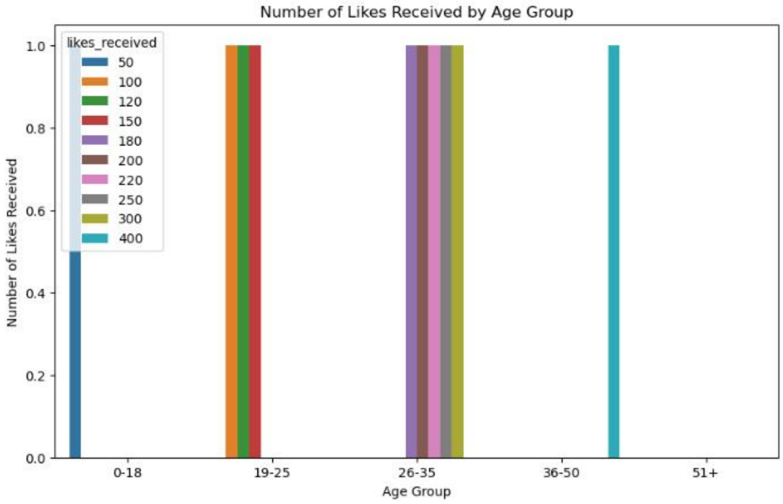


j. Which age group has highest number of likes received? Using count plot.

Code:

```
plt.figure(figsize=(10, 6))
sns.countplot(x='age_group', data=df, hue='likes_received')
plt.xlabel('Age Group')
plt.ylabel('Number of Likes Received')
plt.title('Number of Likes Received by Age Group')
plt.show()
```

Output:

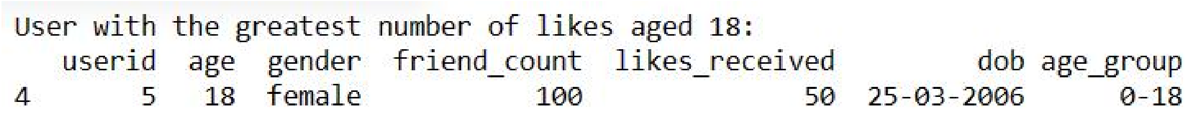


k. Who received the greatest number of likes aged 18?

Code:

```
greatest_likes_age_18 = df[df['age'] == 18].sort_values(by='likes_received', ascending=False).head(1)
print("User with the greatest number of likes aged 18:")
print(greatest_likes_age_18)
```

Output:



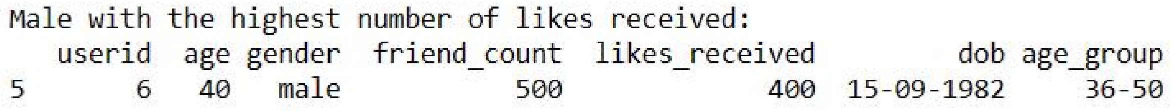
```
User with the greatest number of likes aged 18:
   userid  age  gender  friend_count  likes_received         dob age_group
4       5   18  female           100              50  25-03-2006      0-18
```

l. Who male has the highest number of likes received.

Code:

```
male_highest_likes = df[df['gender'] == 'male'].sort_values(by='likes_received', ascending=False).head(1)
print("Male with the highest number of likes received:")
print(male_highest_likes)
```

Output:



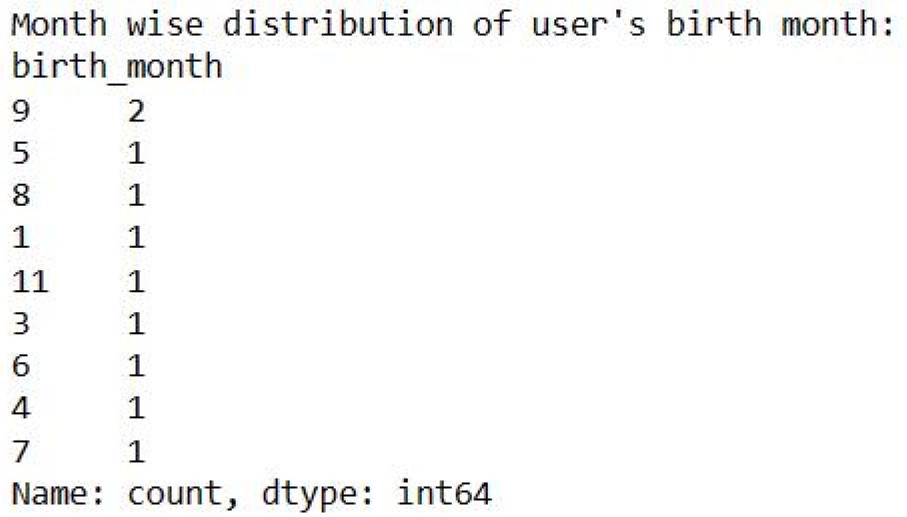
```
Male with the highest number of likes received:
   userid  age gender  friend_count  likes_received         dob age_group
5       6   40   male           500             400  15-09-1982     36-50
```

m. Which month are Facebook users born?

Code:

```
df['dob'] = pd.to_datetime(df['dob'])
df['birth_month'] = df['dob'].dt.month
birth_month_counts = df['birth_month'].value_counts()
print("Month wise distribution of user's birth month:")
print(birth_month_counts)
```

Output:



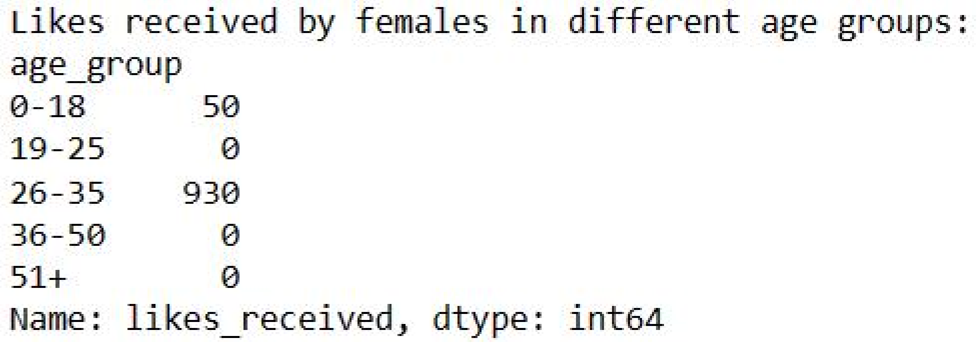
```
Month wise distribution of user's birth month:
birth_month
9     2
5     1
8     1
1     1
11    1
3     1
6     1
4     1
7     1
Name: count, dtype: int64
```

n. Which Females age groups has received a greater number of likes through mobile and web.

Code:

```
female_likes = df[df['gender'] == 'female']
female_likes_by_age_group = female_likes.groupby('age_group')['likes_received'].sum()
print("Likes received by females in different age groups:")
print(female_likes_by_age_group)
```

Output:



```
Likes received by females in different age groups:
age_group
0-18      50
19-25      0
26-35    930
36-50      0
51+        0
Name: likes_received, dtype: int64
```

Collect the web community data/archives using any web crawler tool/software, and perform sentiment analysis using Scrapy

Scrapy is an open-sourced framework that runs on Python. The library offers a ready-to- use structure for programmers to customize a web crawler and extract data from the web on a large scale.

A. Spacy sentiment analysis on Amazon reviews, scraped using Scrapy tool Scrapy is an open-source, web-crawling tool used to scrape data.

B. Data has been scraped from Amazon reviews page and stored in reviews.csv.

C. The multi-class problem is converted to binary classification problem for simplicity. By default, 5/4/3 stars are considered as POSITIVE and 1/2 stars as NEGATIVE.

D. Preprocess the dataset file (like class imbalance/missing values/noise values)

E. After that, the dataset is ready to be used for training and testing.

F. Spacy is used for custom text classification.

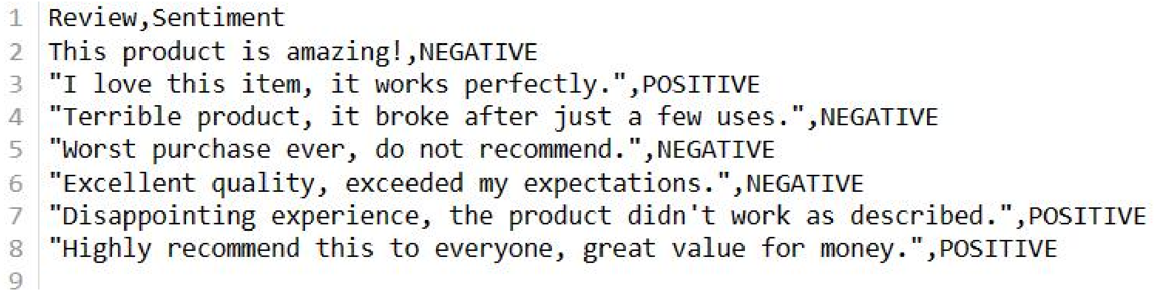
G. Report the precision, recall, F1 score and loss show how our model is performing.

Code:

```
import pandas as pd
import random
reviews = [
    "This product is amazing!",
    "I love this item, it works perfectly.",
    "Terrible product, it broke after just a few uses.",
    "Worst purchase ever, do not recommend.",
    "Excellent quality, exceeded my expectations.",
    "Disappointing experience, the product didn't work as described.",
    "Highly recommend this to everyone, great value for money."
]
sentiments = ['POSITIVE', 'NEGATIVE']
labels = [random.choice(sentiments) for _ in range(7)]
data = pd.DataFrame({'Review': reviews, 'Sentiment': labels})
data.to_csv('reviews.csv', index=False)
print("Random reviews generated and saved to 'reviews.csv'.")
```

Output:

```
Random reviews generated and saved to 'reviews.csv'.
```



```
Review,Sentiment
This product is amazing!,NEGATIVE
"I love this item, it works perfectly.",POSITIVE
"Terrible product, it broke after just a few uses.",NEGATIVE
"Worst purchase ever, do not recommend.",NEGATIVE
"Excellent quality, exceeded my expectations.",NEGATIVE
"Disappointing experience, the product didn't work as described.",POSITIVE
"Highly recommend this to everyone, great value for money.",POSITIVE
```

Step 1: Scrape Amazon Reviews with Scrapy

Code:

```
import scrapy
class AmazonReviewsSpider(scrapy.Spider):
    name = 'amazon_reviews'
    start_urls = ['https://www.amazon.com/product-reviews/B09B3Y81BC']
    def parse(self, response):
        for review in response.css('div.review'):
            yield {
                'review_text': review.css('span.review-text::text').get(),
                'star_rating': review.css('i.review-rating span.a-icon-alt::text').re_first(r'(\d\.\d|\d)'),
            }
        next_page = response.css('li.a-last a::attr(href)').get()
        if next_page is not None:
            yield response.follow(next_page, self.parse)
```

Step 2: Preprocess Data

After scraping the data, preprocess it to handle class imbalance, missing values, and noise. For example, you can handle missing star ratings or remove reviews with insufficient information.

Step 3: Convert to Binary Classification

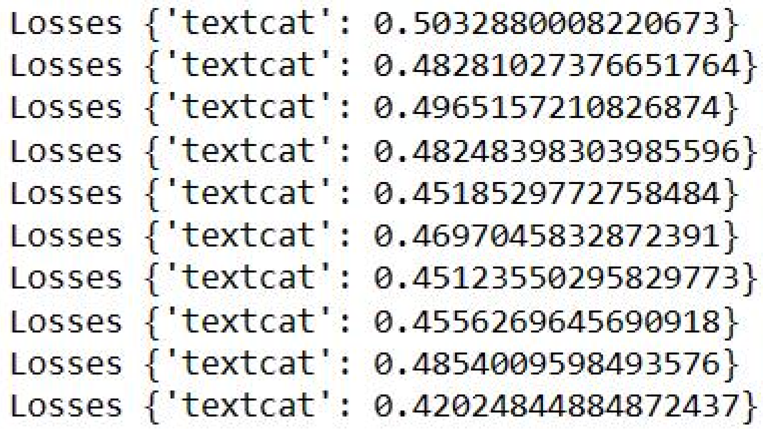
Convert the star ratings to binary labels (positive/negative) based on the provided mapping (5/4/3 stars as POSITIVE and 1/2 stars as NEGATIVE).

Step 4: Train Spacy Model for Sentiment Analysis

Code:

```
import spacy
from spacy.training.example import Example
from spacy.util import minibatch, compounding
import random
nlp = spacy.blank("en")
textcat = nlp.add_pipe("textcat")
textcat.add_label("POSITIVE")
textcat.add_label("NEGATIVE")
train_data = [
    ("This product is amazing!", {"cats": {"POSITIVE": 0, "NEGATIVE": 1}}),
    ("I love this item, it works perfectly.", {"cats": {"POSITIVE": 1, "NEGATIVE": 0}}),
    ("Terrible product, it broke after just a few uses.", {"cats": {"POSITIVE": 0, "NEGATIVE": 1}}),
    ("Worst purchase ever, do not recommend.", {"cats": {"POSITIVE": 0, "NEGATIVE": 1}}),
    ("Excellent quality, exceeded my expectations.", {"cats": {"POSITIVE": 1, "NEGATIVE": 0}}),
    ("Disappointing experience, the product didn't work as described.", {"cats": {"POSITIVE": 0, "NEGATIVE": 1}}),
    ("Highly recommend this to everyone, great value for money.", {"cats": {"POSITIVE": 1, "NEGATIVE": 0}})
]
train_examples = []
for text, annotation in train_data:
    example = Example.from_dict(nlp.make_doc(text), annotation)
    train_examples.append(example)
n_iter = 10
other_pipes = [pipe for pipe in nlp.pipe_names if pipe != "textcat"]
with nlp.disable_pipes(*other_pipes):
    optimizer = nlp.begin_training()
    for i in range(n_iter):
        losses = {}
        random.shuffle(train_examples)
        batches = minibatch(train_examples, size=compounding(4.0, 32.0, 1.001))
        for batch in batches:
            nlp.update(batch, drop=0.5, losses=losses)
        print("Losses", losses)
nlp.to_disk("amazon_sentiment_model")
```

Output:



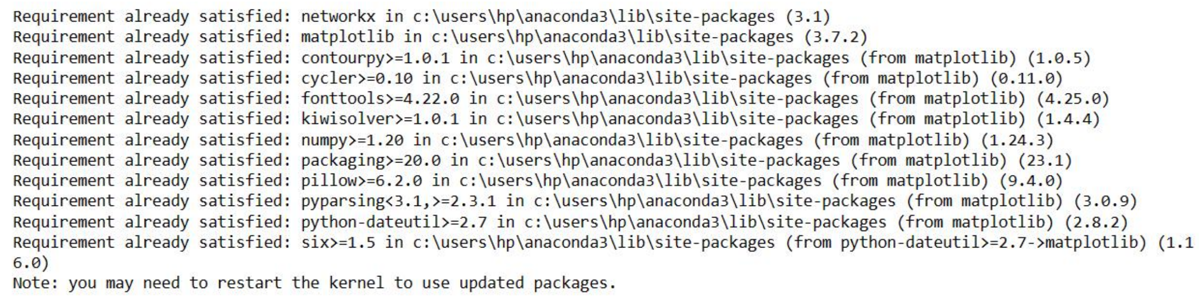
```
Losses {'textcat': 0.5032880008220673}
Losses {'textcat': 0.48281027376651764}
Losses {'textcat': 0.4965157210826874}
Losses {'textcat': 0.48248398303985596}
Losses {'textcat': 0.4518529772758484}
Losses {'textcat': 0.4697045832872391}
Losses {'textcat': 0.45123550295829773}
Losses {'textcat': 0.4556269645690918}
Losses {'textcat': 0.4854009598493576}
Losses {'textcat': 0.42024844884872437}
```

Perform network analysis using Network X API

Code:

pip install networkx matplotlib

Output:



```
Requirement already satisfied: networkx in c:\users\hp\anaconda3\lib\site-packages (3.1)
Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib\site-packages (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.24.3)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.1
6.0)
Note: you may need to restart the kernel to use updated packages.
```

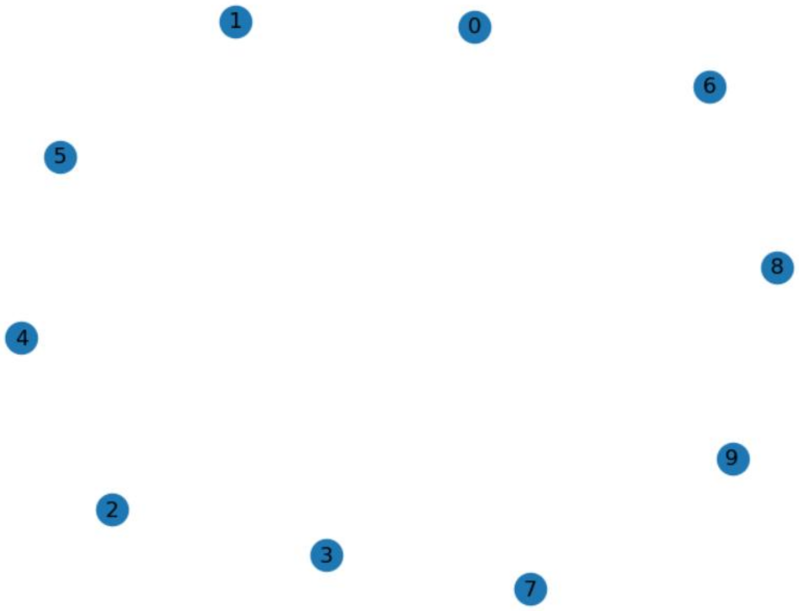
A. Create a graph (min 8 to 10 nodes)

Code:

```
import networkx as nx
G = nx.Graph()
nodes = range(10)
G.add_nodes_from(nodes)
print("Nodes:", G.nodes())
nx.draw(G, with_labels=True)
```

Output:

```
Nodes: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

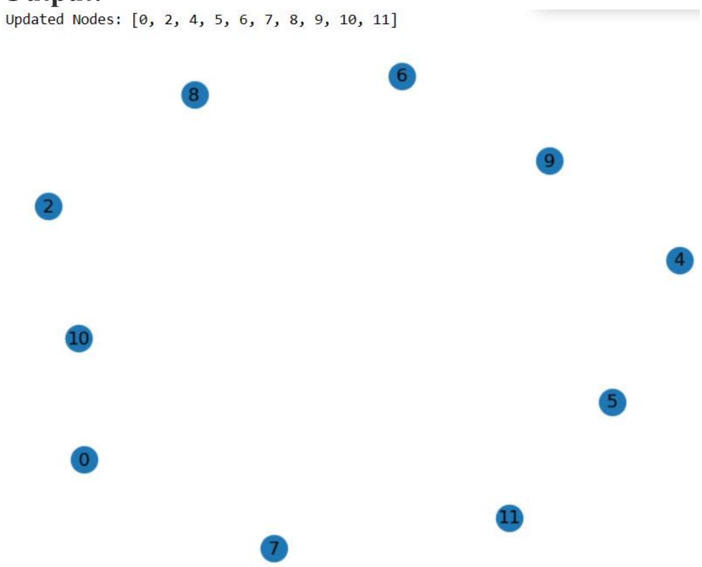


B. Add and delete nodes

Code:

```
G.add_node(10)
G.add_node(11)
G.remove_node(1)
G.remove_node(3)
print("Updated Nodes:", G.nodes())
nx.draw(G, with_labels=True)
```

Output:



```
Updated Nodes: [0, 2, 4, 5, 6, 7, 8, 9, 10, 11]
```

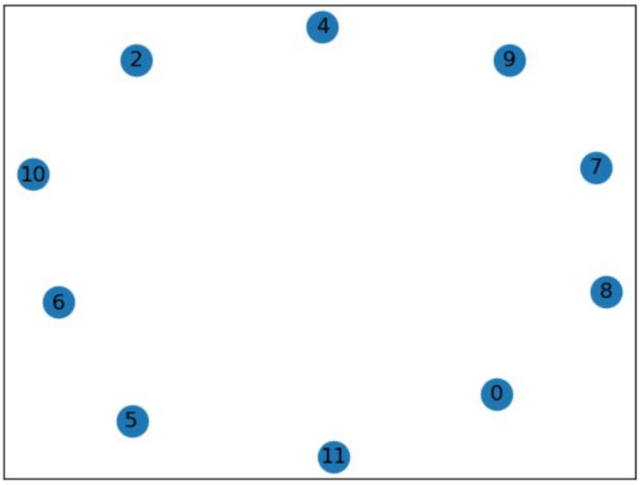
C. Visualize the graph with loaded datasets

- **Node-link diagram**

Code:

```
nx.draw_networkx(G, with_labels=True)
```

Output:

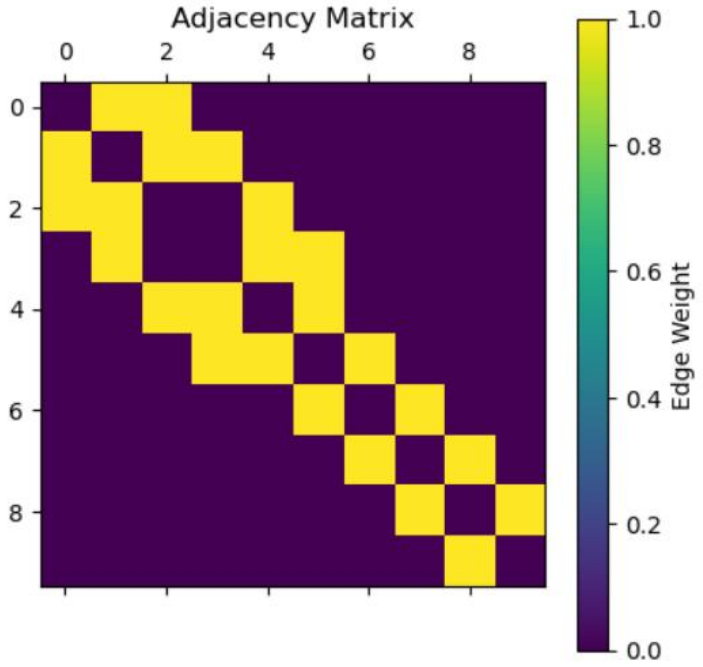


- **Matrix plot**

Code:

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
G = nx.Graph()
G.add_edges_from([(1,2), (1,3), (2,3), (2,4), (3,5), (4,5), (4,6), (5,6), (6,7), (7,8), (8,9), (9,10)])
adj_matrix = nx.to_numpy_array(G)
plt.matshow(adj_matrix, cmap='viridis')
plt.colorbar(label='Edge Weight')
plt.title('Adjacency Matrix')
plt.show()
```

Output:

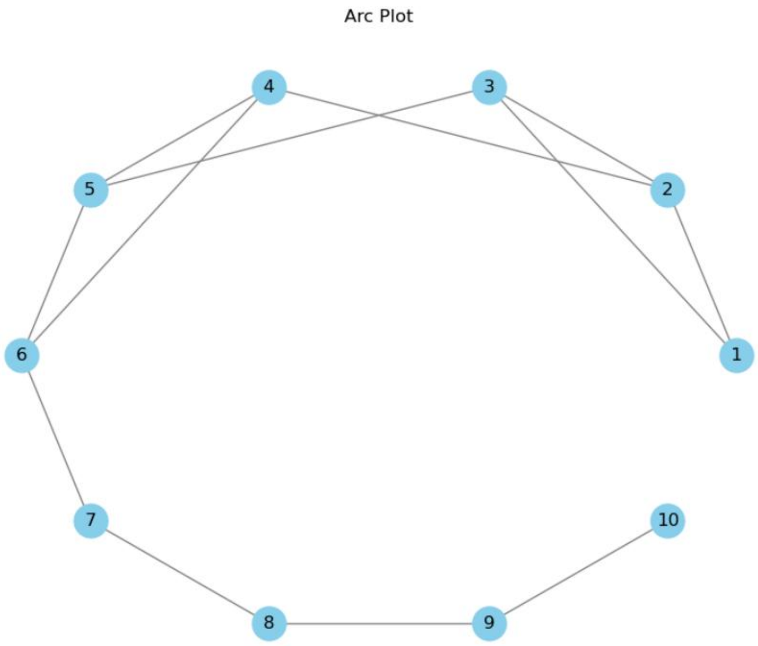


- **Arc plot**

Code:

```
from matplotlib import pyplot as plt
plt.figure(figsize=(8, 6))
nx.draw_circular(G, with_labels=True, node_size=500, node_color='skyblue', edge_color='gray')
plt.title('Arc Plot')
plt.show()
```

Output:

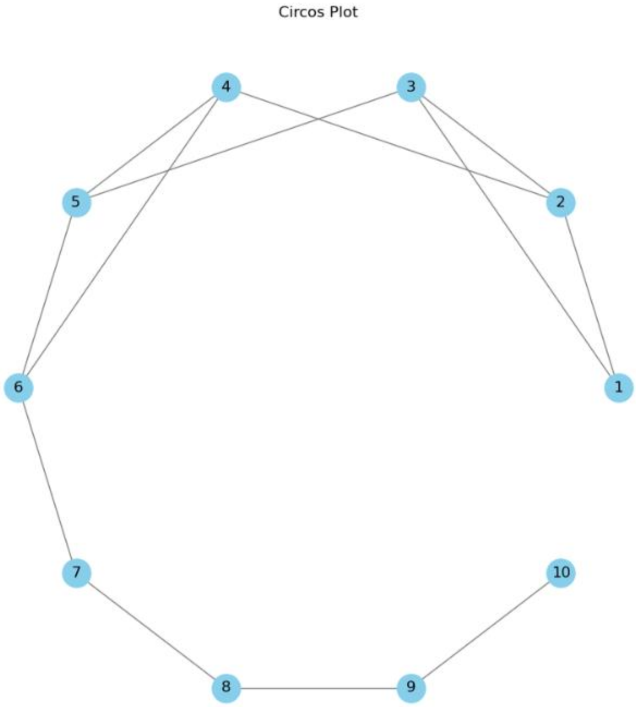


- **Circus plot**

Code:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 8))
nx.draw_circular(G, with_labels=True, node_size=500, node_color='skyblue', edge_color='gray')
plt.title('Circos Plot')
plt.show()
```

Output:

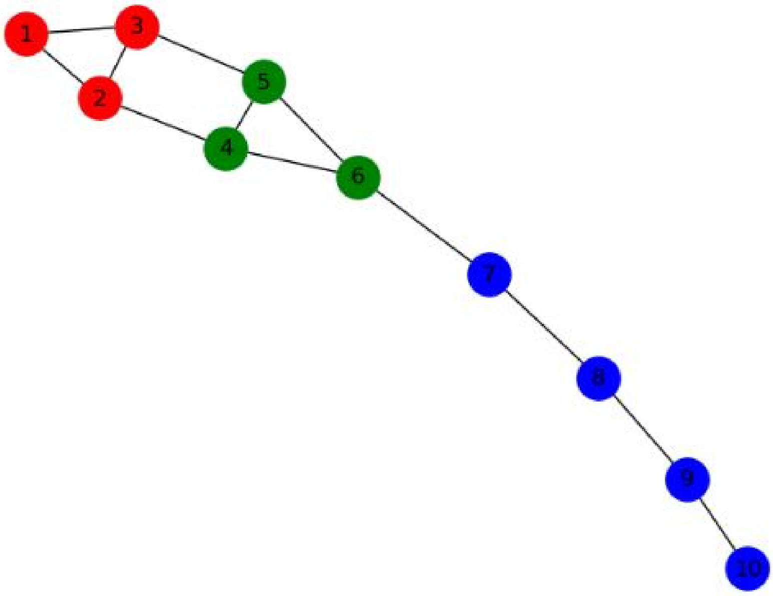


- **Hive plot**

Code:

```
pos = nx.spring_layout(G)
nx.draw(G, pos, node_color='skyblue', node_size=500, with_labels=True)
nx.draw_networkx_nodes(G, pos, nodelist=[1,2,3], node_color='red', node_size=500)
nx.draw_networkx_nodes(G, pos, nodelist=[4,5,6], node_color='green', node_size=500)
nx.draw_networkx_nodes(G, pos, nodelist=[7,8,9,10], node_color='blue', node_size=500)
plt.show()
```

Output:


1
3
2
5
4
6
7
8
9
10