ANALYZING MILLENNIAL PREFERENCES ACROSS SOCIAL MEDIA PLATFORMS

A.Banu Harshini
121323030026

CONTENT:

INTRODUCTION

Millennials, a generation deeply immersed in the digital age, have shaped the landscape of social media. This analysis dives into their preferences across platforms like Instagram and Facebook, highlighting how factors such as device usage and gender influence their online behavior. By examining platform engagement and content trends, this study offers a concise look at how millennials connect and interact in today's social media-driven world.

OBJECTIVE:

- Understand Platform Preferences:
- Identify which social media platforms (e.g., Instagram, Facebook) are most preferred by millennials across different segments.
- Segment-based Analysis:
- Analyze platform preferences across various user segments, such as:
    - Mobile vs. Web users: Evaluate how platform preferences differ between users accessing social media via mobile and web platforms.
    - Demographic segments: Examine differences in platform preferences across gender (e.g., male vs. female respondents).
- Behavioral Insights:
- Assess how different user segments respond to notification badges across platforms and whether certain platforms drive more engagement.
- Platform Dominance:
- Identify the leading platform among millennial respondents based on count and percentage preferences.
- Preference Variability:
- Explore the variability in platform preferences across different device types and demographic groups, helping in targeted marketing strategies.

DATASET:

Question

Segment Type

Segment Description

Answer

COUNT

PERCENTAGE

TOOLS

PYTHON

Python is a high-level, general-purpose programming language known for its simplicity and readability. It's widely used in fields like web development, automation, data analysis, machine learning, and scientific computing.

JUPYTER NOTEBOOK

Jupyter Notebook is an open-source interactive computing environment where users can combine live code, equations, visualizations, and narrative text in a single document.

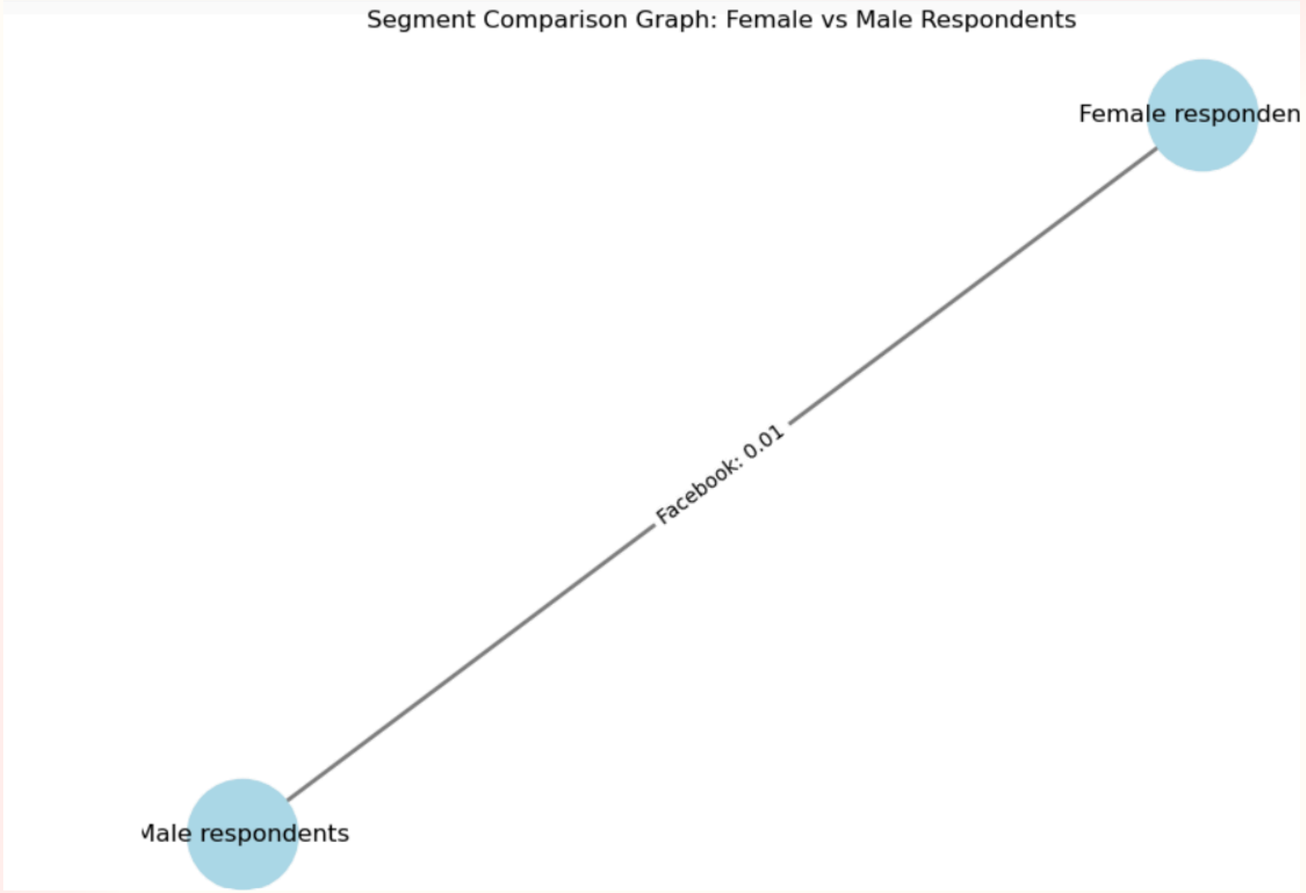
NETWORKX

NetworkX is a Python library used for creating, manipulating, and analyzing complex networks (graphs). It supports both undirected and directed graphs and can handle many types of real-world network data.

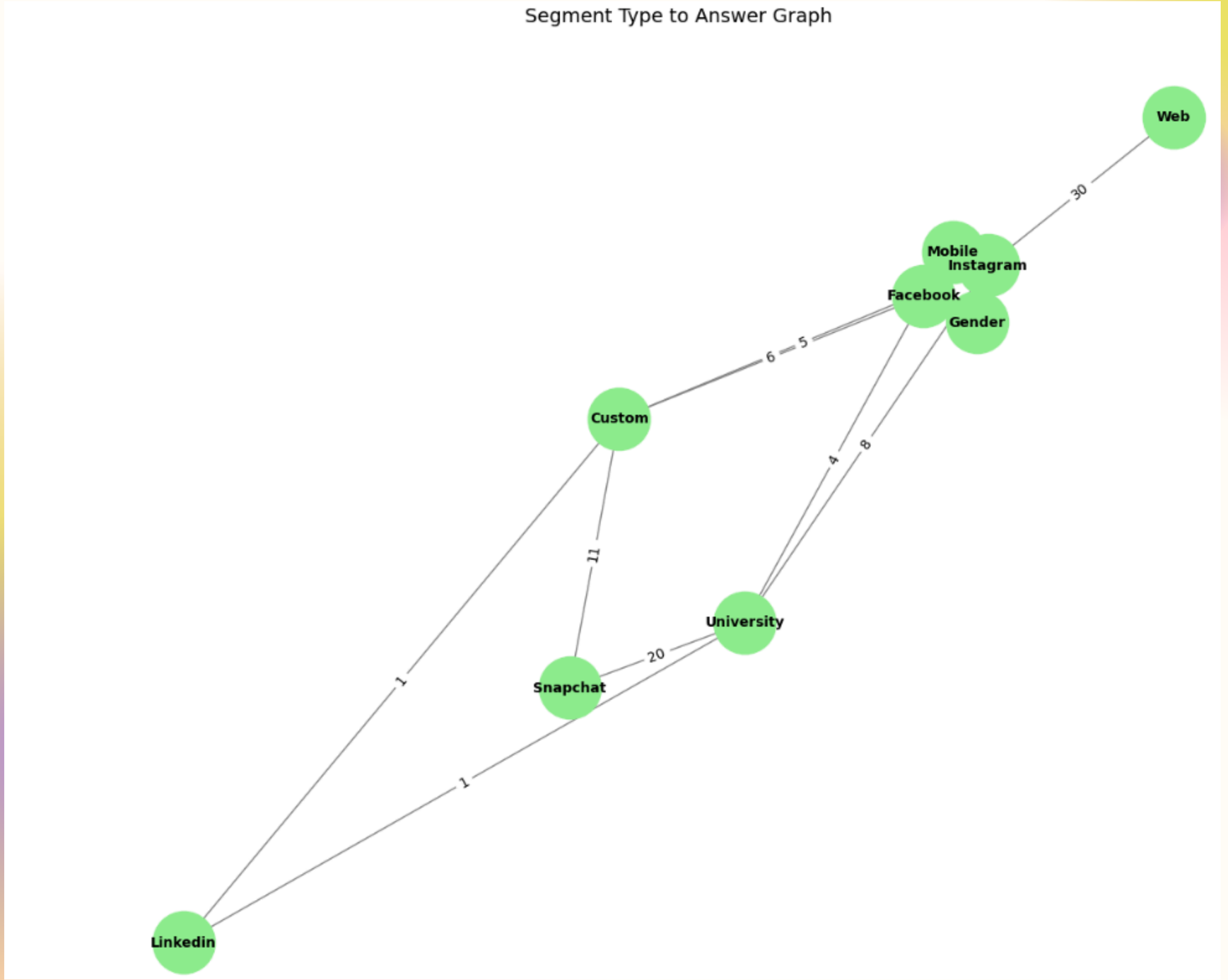
NETWORK ANALYSIS:

```python
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Load dataset
file_path = 'Dataset.csv'
data = pd.read_csv(file_path)

# Define segments to compare (e.g., Female respondents, Male respondents)
segments = ["Female respondents", "Male respondents"]

# Create an empty graph
G = nx.Graph()

# Filter dataset to include only the specified segments
filtered_data = data[data['Segment Description'].isin(segments)]

# Create a dictionary to store the answers for each segment
segment_answers = {}
for segment in segments:
    segment_data = filtered_data[filtered_data['Segment Description'] == segment]
    segment_answers[segment] = {row['Answer']: row['Count'] for _, row in segment_data.iterrows()}

# Compare segments based on common answers and calculate the difference in their responses
for answer in set(segment_answers[segments[0]].keys()).intersection(segment_answers[segments[1]].keys()):
    count_female = segment_answers['Female respondents'].get(answer, 0)
    count_male = segment_answers['Male respondents'].get(answer, 0)

    # Calculate the difference or similarity in response counts
    similarity = 1 / (1 + abs(count_female - count_male))  # Inverse of absolute difference

    # Add an edge between the segments with the weight as the similarity score
    G.add_edge('Female respondents', 'Male respondents', weight=similarity, label=f'{answer}: {similarity:.2f}')
```

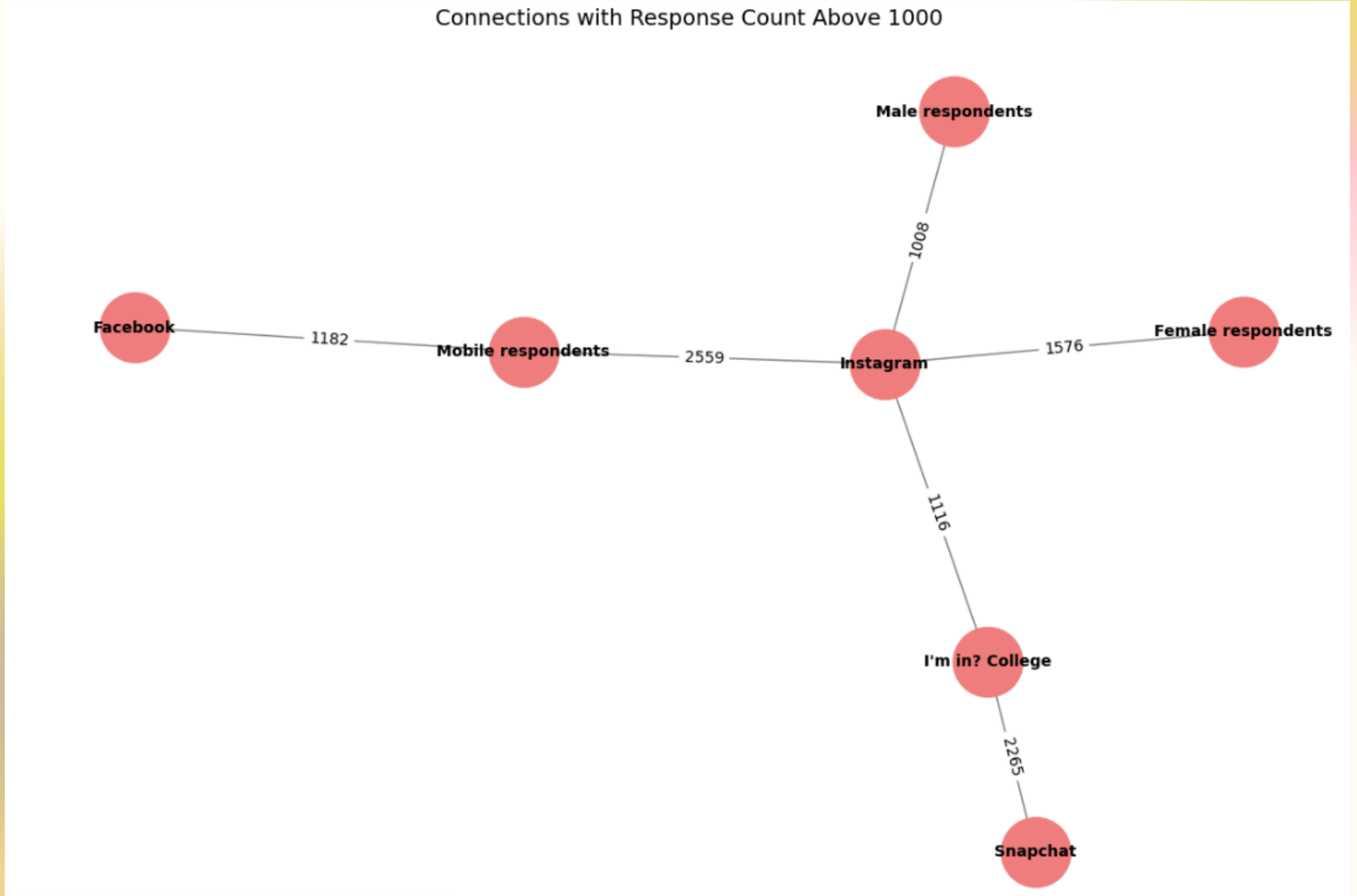
```python
# Use a layout and draw the graph
pos = nx.spring_layout(G)
plt.figure(figsize=(8, 6))

# Draw nodes and edges with weights
nx.draw(G, pos, with_labels=True, node_color='lightblue', node_size=3000, edge_color='gray', width=2)

# Draw edge labels (showing the answer and similarity score)
edge_labels = nx.get_edge_attributes(G, 'label')
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_size=10)

# Display the graph
plt.title("Segment Comparison Graph: Female vs Male Respondents")
plt.show()
```



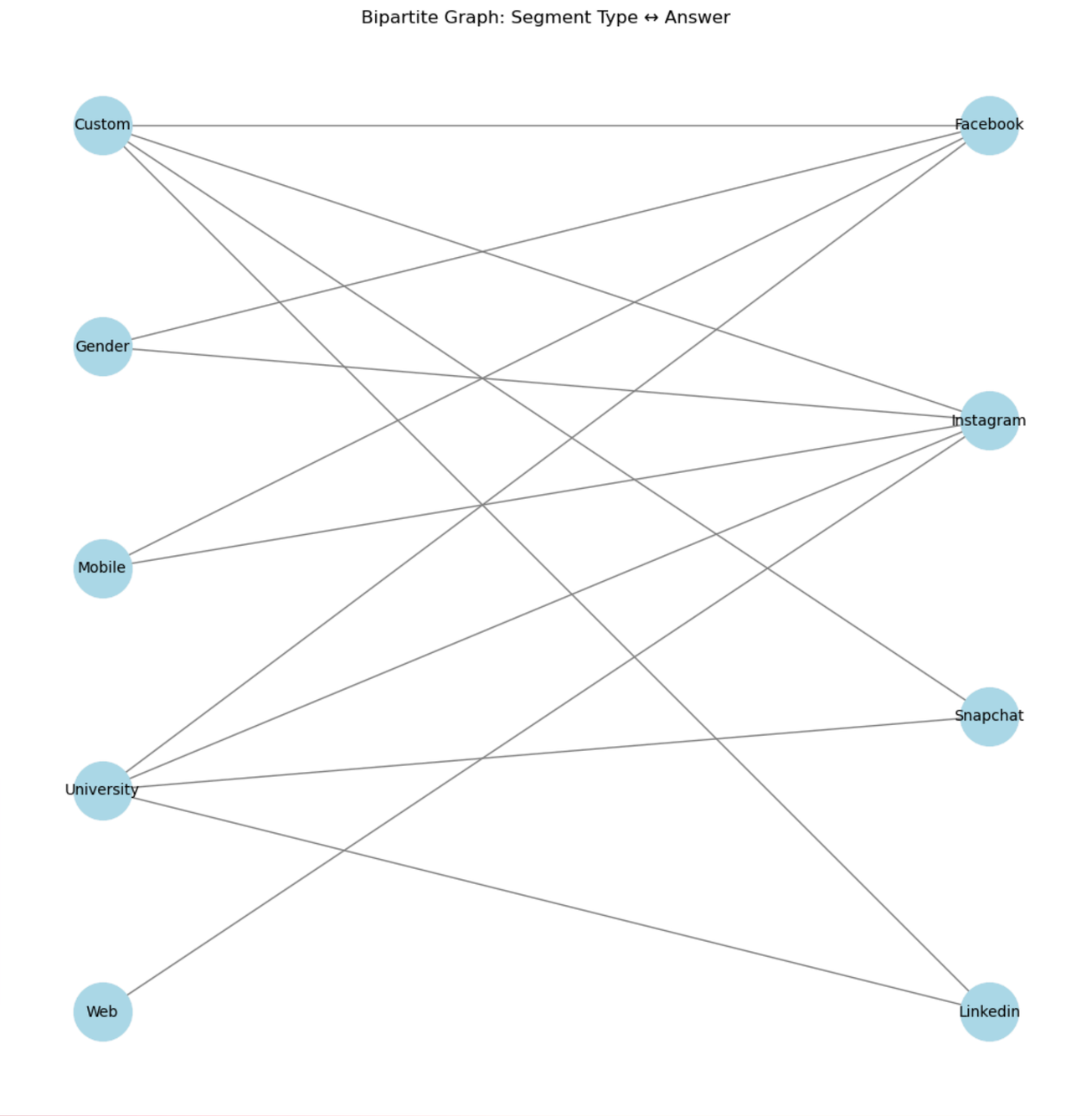
```python
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Load dataset
file_path = 'Dataset.csv'
data = pd.read_csv(file_path)

# Create an empty graph
G = nx.Graph()

# Iterate through the dataset and add edges between "Segment Type" and "Answer"
for _, row in data.iterrows():
    segment_type = row['Segment Type']  # Example: Mobile, Web, Gender
    answer = row['Answer']               # Example: Instagram, Facebook
    count = row['Count']                 # Count of responses

    # Add an edge between segment type and answer with the weight as the count
    G.add_edge(segment_type, answer, weight=count)

# Use a layout to avoid overlap
pos = nx.spring_layout(G, k=0.5)  # k controls the distance between nodes

# Increase figure size for better visualization
plt.figure(figsize=(12, 10))

# Draw nodes and edges
nx.draw(G, pos, with_labels=True, node_color='lightgreen', node_size=2000, edge_color='gray', font_size=10, font_weight='bold')

# Draw edge labels (showing the response count)
edge_labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_size=10)

# Display the graph
plt.title("Segment Type to Answer Graph", fontsize=14)
plt.axis("off")
plt.show()
```


Segment Type to Answer Graph
Web
Mobile
Instagram
Facebook
Gender
Custom
University
Snapchat
Linkedin
30
6
5
4
8
11
20
1
1

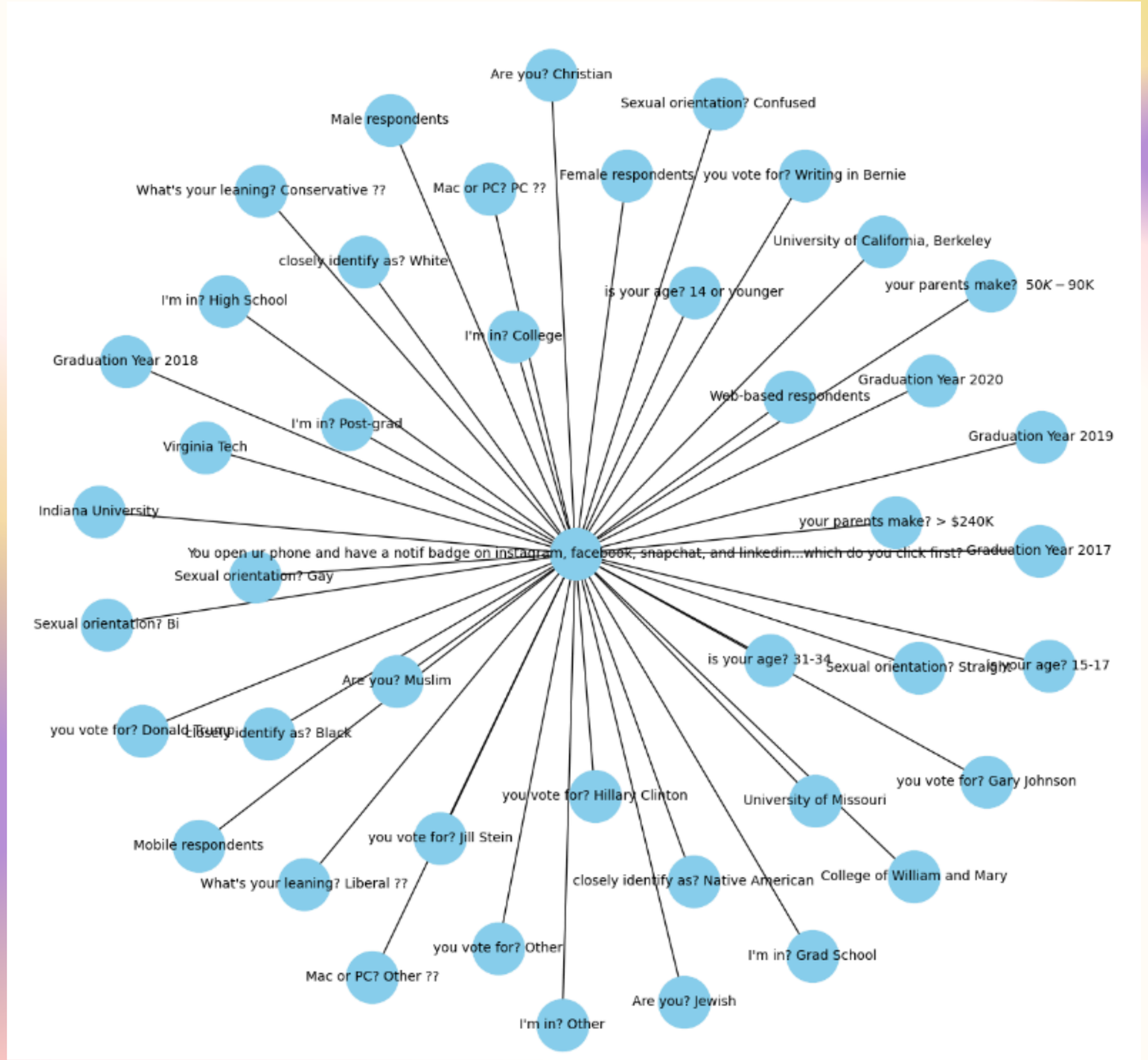
```python
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Load dataset
file_path = 'Dataset.csv'
data = pd.read_csv(file_path)

# Create an empty graph
G = nx.Graph()

# Set your response count threshold
response_count_threshold = 1000  # Adjust this value as needed

# Iterate through the dataset to add edges between "Segment Description" and "Answer" if above the threshold
for _, row in data.iterrows():
    segment_description = row['Segment Description']  # Example: Mobile respondents, Female respondents
    answer = row['Answer']                             # Example: Instagram, Facebook
    count = row['Count']                               # Count of responses

    # Only add an edge if the count is above the threshold
    if count > response_count_threshold:
        G.add_edge(segment_description, answer, weight=count)

# Use spring layout for the graph
pos = nx.spring_layout(G, k=0.5, iterations=50)

# Increase figure size for better visualization
plt.figure(figsize=(12, 8))

# Draw nodes and edges
nx.draw(G, pos, with_labels=True, node_color='lightcoral', node_size=2000, edge_color='gray', font_size=10, font_weight='bold')

# Draw edge labels (showing the response count)
edge_labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_size=10)

# Display the graph
plt.title(f"Connections with Response Count Above {response_count_threshold}", fontsize=14)
plt.axis("off")
plt.show()
```


Connections with Response Count Above 1000
Male respondents
1008
Facebook
1182
Mobile respondents
2559
Instagram
1576
Female respondents
1116
I'm in? College
2265
Snapchat

```python
from networkx.algorithms import bipartite

# Create a bipartite graph
B = nx.Graph()

# Add nodes with the bipartite node attribute
segment_types = df['Segment Type'].unique()
answers = df['Answer'].unique()

B.add_nodes_from(segment_types, bipartite=0)  # One set of nodes
B.add_nodes_from(answers, bipartite=1)  # Another set of nodes

# Add edges based on the relationships between Segment Type and Answer
B.add_edges_from(zip(df['Segment Type'], df['Answer']))

# Plot the bipartite graph
plt.figure(figsize=(10, 10))
pos = nx.bipartite_layout(B, segment_types)  # Layout for bipartite graph

nx.draw(B, pos, with_labels=True, node_color='lightblue', edge_color='gray', node_size=1500, font_size=10)
plt.title('Bipartite Graph: Segment Type ↔ Answer')
plt.show()
```


Bipartite Graph: Segment Type ↔ Answer
Custom
Gender
Mobile
University
Web
Facebook
Instagram
Snapchat
Linkedin

```python
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Load your dataset
data = pd.read_csv('Dataset.csv')

# Create a new graph
G = nx.Graph()

# Iterate through the dataset and add nodes and edges for 'Question' and 'Segment Description'
for index, row in data.iterrows():
    question = row['Question']  # Replace with the correct column name for 'Question'
    segment_desc = row['Segment Description']  # Replace with the correct column name for 'Segment Description'

    # Add 'Question' and 'Segment Description' as nodes
    G.add_node(question, label='Question')
    G.add_node(segment_desc, label='Segment Description')

    # Add an edge between 'Question' and 'Segment Description'
    G.add_edge(question, segment_desc)

# Draw the graph
plt.figure(figsize=(12, 12))
nx.draw(G, with_labels=True, node_color="skyblue", node_size=1500, font_size=10)
plt.show()
```


Are you? Christian
Male respondents
Sexual orientation? Confused
What's your leaning? Conservative ??
Mac or PC? PC ??
Female respondents
you vote for? Writing in Bernie
University of California, Berkeley
closely identify as? White
is your age? 14 or younger
your parents make? 50K – 90K
I'm in? High School
I'm in? College
Graduation Year 2018
Graduation Year 2020
Web-based respondents
I'm in? Post-grad
Graduation Year 2019
Virginia Tech
Indiana University
your parents make? > $240K
You open ur phone and have a notif badge on instagram, facebook, snapchat, and linkedin...which do you click first?
Graduation Year 2017
Sexual orientation? Gay
Sexual orientation? Bi
is your age? 31-34
Sexual orientation? Straight
is your age? 15-17
Are you? Muslim
you vote for? Donald Trump
closely identify as? Black
you vote for? Gary Johnson
you vote for? Hillary Clinton
University of Missouri
Mobile respondents
you vote for? Jill Stein
College of William and Mary
What's your leaning? Liberal ??
closely identify as? Native American
you vote for? Other
I'm in? Grad School
Mac or PC? Other ??
Are you? Jewish
I'm in? Other

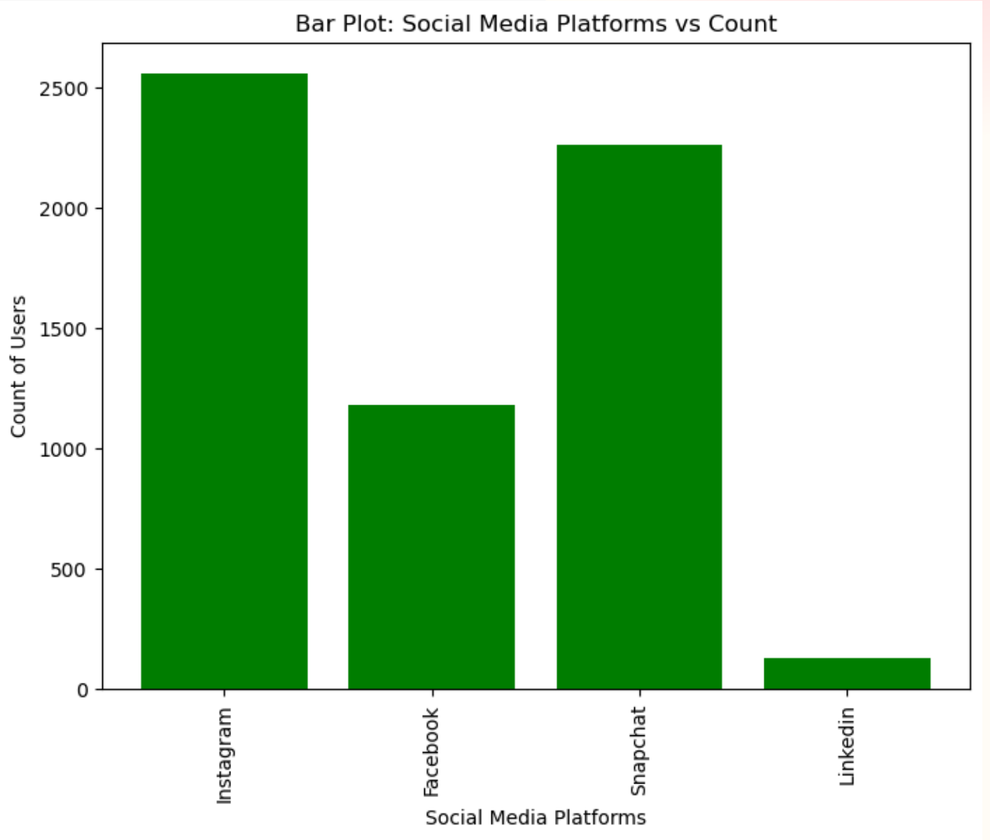
```python
import networkx as nx
import matplotlib.pyplot as plt

# Assuming your DataFrame is already loaded and contains 'Segment Type', 'Answer', and 'Count'

# Create a graph
G_type_answer = nx.Graph()

# Add edges between Segment Type and Answer, weighted by Count
for index, row in df.iterrows():
    G_type_answer.add_edge(row['Segment Type'], row['Answer'], weight=row['Count'])

# Get edge weights (Count) for edge thickness
weights = [d['weight'] for (u, v, d) in G_type_answer.edges(data=True)]

# Normalize weights for better visualization
max_weight = max(weights) if weights else 1  # Avoid division by zero
normalized_weights = [w / max_weight * 5 for w in weights]  # Scale weights between 1 and 5

# Draw the graph
plt.figure(figsize=(10, 10))

# Get positions for nodes using spring layout (force-directed layout)
pos = nx.spring_layout(G_type_answer)

# Draw nodes (Segment Type and Answer)
nx.draw_networkx_nodes(G_type_answer, pos, node_color='lightblue', node_size=1200)

# Draw edges with widths proportional to the 'Count' values
nx.draw_networkx_edges(G_type_answer, pos, width=normalized_weights, edge_color='gray')

# Draw labels
nx.draw_networkx_labels(G_type_answer, pos, font_size=10)

# Title for the graph
plt.title('Network Graph: Segment Type to Answer (Edge Thickness by Count)')
plt.show()
```


Network Graph: Segment Type to Answer (Edge Thickness by Count)
Web
Gender
Facebook
Instagram
Mobile
Custom
University
Snapchat
Linkedin

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Segment Description', y='Answer', size='Count', data=df, sizes=(20, 200), palette='cool')

plt.title('Scatterplot: Segment Description vs Answer (Sized by Count)')
plt.xlabel('Segment Description')
plt.ylabel('Answer')
plt.xticks(rotation=90)  # Rotate x-axis labels for readability
plt.show()
```

Scatterplot: Segment Description vs Answer (Sized by Count)
Instagram
Facebook
Snapchat
Linkedin
Answer
Count
0
500
1000
1500
2000
2500
Mobile respondents
Web-based respondents
Female respondents
Male respondents
Indiana University
University of California, Berkeley
College of William and Mary
University of Missouri
Virginia Tech
you vote for? Writing in Bernie
you vote for? Hillary Clinton
Graduation Year 2019
closely identify as? White
I'm in? Post-grad
Mac or PC? PC ??
Graduation Year 2018
your parents make? > $240K
is your age? 31-34
What's your leaning? Liberal ??
What's your leaning? Conservative ??
is your age? 15-17
is your age? 14 or younger
I'm in? College
you vote for? Other
you vote for? Gary Johnson
Are you? Muslim
Are you? Jewish
Sexual orientation? Bi
Sexual orientation? Straight
Sexual orientation? Gay
I'm in? Other
Are you? Christian
you vote for? Jill Stein
closely identify as? Black
Sexual orientation? Confused
I'm in? Grad School
Graduation Year 2020
you vote for? Donald Trump
Graduation Year 2017
closely identify as? Native American
I'm in? High School
your parents make? 50K – 90K
Mac or PC? Other ??
Segment Description

```
plt.figure(figsize=(8,6))
plt.bar(platforms, counts, color='green')
plt.title('Bar Plot: Social Media Platforms vs Count')
plt.xlabel('Social Media Platforms')
plt.ylabel('Count of Users')
plt.xticks(rotation=90)
plt.show()
```



INFERENCE:

Platform Popularity: Instagram seems to be one of the top platforms used by different respondent segments (mobile, web, gender), with a higher percentage of responses compared to other platforms like Facebook.

Segment Differences: The data is segmented by different categories (Mobile, Web, Gender), which indicates that platform preferences may vary across these user segments. For example, Instagram appears popular across all categories, but Facebook has varying levels of usage, being less preferred in the "Mobile" and "Female" segments.

CONCLUSION:

Instagram Dominance: Based on the response count and percentage, Instagram is a dominant platform across various segments, particularly among mobile and female users.

Facebook's Decline: Facebook shows a lower engagement compared to Instagram, especially among mobile users and females. This could imply shifting user preferences, where newer platforms are favored by certain demographics.

Thank you