A PROJECT WORK MADE UNDER THE GUIDANCE OF VIGOR COUNCIL

Vigor Council

PROJECT WORK

Submitted by:

SWEETY

Data Analyst @ Vigor Council

June , 2024

ACKNOWLEDGEMENT

I would like to express our heartfelt gratitude to our supervisor, Dr. B.P. Sharma, for his invaluable guidance and support throughout this project. This project helped us to explore and undertake research work and gaining practical knowledge and expertise to perform tasks. We also extend our thanks to the Vigor Council for providing the necessary resources and facilities.

Thanking You

SWEETY

Data analyst intern @ Vigor council

DECLARATION

I, Sweety, hereby declare that the project entitled "User Profiling and segmentation" is solely my original work.

This project represents my independent research conducted under the guidance and supervision of Dr. B.P. Sharma, President at Vigor Council, during my internship as a Data Analyst at Vigor Council.

I affirm that all data and information used in this project are properly cited and referenced. I have adhered to ethical research practices and ensured the accuracy and authenticity of the presented findings.

Should any external assistance have been received during the project or report preparation, it has been duly acknowledged within the document.

Date: June,2024

SWEETY

Data analyst intern @ Vigor council

User Profiling and Segmentation Analysis Using ML

Project Content

1.Introduction

1.1 Examining the Project Topic

What is a User Profiling and Segmentation?

- In today's data-driven world, truly knowing your users is the golden ticket to success. That's where user profiling and segmentation come in. Imagine a room full of people – a diverse mix of interests, needs, and preferences. User profiling helps us create detailed descriptions of these individual users. We gather information about their demographics, behaviors, and preferences, painting a vivid picture of who they are.
- But understanding individuals is just the first step. Segmentation takes things a step further. It allows us to group users with similar characteristics, effectively segmenting that crowded room. Think of it like organizing those people by interest – the movie buffs in one corner, the tech enthusiasts in another.
- This user segmentation becomes the foundation for our machine learning project. By understanding these distinct groups, we can tailor our approach. We can develop targeted experiences, recommendations, or predictions that resonate with each segment. Imagine showing movie trailers to the film fans and tech news to the gadget gurus.
- In essence, user profiling and segmentation help us move beyond a one-size-fits-all approach. They allow us to speak directly to the unique needs and interests of our users, ultimately leading to a more successful and impactful machine learning project.

Why User Profiling and Segmentation Matters?

- Imagine walking into a crowded marketplace with a cart full of hand-painted portraits. Everyone has different tastes – some love landscapes, others adore action scenes. Trying to sell portraits to everyone would be exhausting and ineffective.
- This is the challenge advertisers face without user profiling and segmentation. They're essentially shouting generic messages into the void, hoping someone will connect. Here's why these practices are game-changers in the advertising world:
-
  - **Laser-Focused Targeting:** Profiling helps you create detailed buyer personas – descriptions of your ideal customers. Age, interests, online behavior – all this paints a picture of who you should be reaching. Segmentation then groups these similar users together.
-
  - **Speak Their Language** Once you understand your audience segments, you can tailor your message to resonate with them. This personalization grabs attention, making your ad stand out in the crowded marketplace.
-
  - **Boost Engagement and Conversions:** Generic ads often get ignored. But targeted messages based on user profiles and segmentation pique interest. People feel like you "get" them, leading to higher engagement and ultimately, more conversions (turning interest into sales). Imagine someone looking for a landscape portrait – they'd be far more likely to stop at your cart compared to someone searching for a battle scene.
-
  - **Maximize Your Marketing Budget:** By focusing on the right audience, you avoid wasting resources on irrelevant demographics. It's like strategically placing your portraits in different parts of the marketplace – no more shouting across the crowd. Your marketing budget becomes more efficient, delivering a bigger bang for your buck.

- In essence, user profiling and segmentation allow you to transform from a random portrait seller into a master curator, connecting with your ideal customers on a deeper level. This personalized approach is the key to unlocking successful advertising campaigns in today's competitive market.

User Profiling and Segmentation: Process We Can Follow

User profiling and segmentation are powerful techniques that enable data professionals to understand their user base in-depth and tailor their strategies to meet diverse user needs. Below is the process we can follow for the task of User Profiling and Segmentation:

1. Determine what you aim to achieve with user profiling and segmentation, such as improving customer service, personalized marketing, or product recommendation.

2. Collect data from various sources, including user interactions on websites/apps, transaction histories, social media activity, and demographic information.
3. Create new features that capture relevant user behaviors and preferences. It may involve aggregating transaction data, calculating the frequency of activities, or extracting patterns from usage logs.
4. Select appropriate segmentation techniques.
5. For each segment identified, create user profiles that summarize the key characteristics and behaviors of users in that segment.

So, to get started with User Profiling and Segmentation, we need an appropriate dataset. I found an ideal dataset for this task. You can download the dataset from herehttps://statso.io/userprofiling-case-study/.

- Note: I thank the Aman kharwal, Data Strategist at Statso.io for the guidance and sharing the required resources for this project.

1.2 Recognizing Variables In Dataset

Variable definitions in the Dataset

- **User ID:** Unique identifier for each user.
- **Age:** Age range of the user.
- **Gender:** Gender of the user.
- **Location:** User's location type (Urban, Suburban, Rural).
- **Language:** Primary language of the user.
- **Education Level:** Highest education level achieved.
- **Likes and Reactions:** Number of likes and reactions a user has made.
- **Followed Accounts:** Number of accounts a user follows.
- **Device Usage:** Primary device used for accessing the platform (Mobile, Desktop, Tablet).
- **Time Spent Online (hrs/weekday):** Average hours spent online on weekdays.
- **Time Spent Online (hrs/weekend):** Average hours spent online on weekends.
- **Click-Through Rates (CTR):** The percentage of ad impressions that lead to clicks.
- **Conversion Rates:** The percentage of clicks that lead to conversions/actions.
- **Ad Interaction Time (sec):** Average time spent interacting with ads in seconds.
- **Income Level:** User's income level.
- **Top Interests:** Primary interests of the user.

Let's have a look at dataset.

First Organization

Go to the Project Content

2.1 Required Python Libraries

2.1.1 Basic Libraries

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
```

```
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os for dirname, _, filenames in os.walk('/kaggle/input'):     for filename in filenames:         print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session

import warnings
warnings.filterwarnings('ignore')

import matplotlib.pyplot as plt
import seaborn as sns
```

```
/kaggle/input/user-profiling-and-segmentation-using-ml-project/user_profiles_for_ads (1).csv
```

2.2 Loading The Dataset

```
df=pd.read_csv("/kaggle/input/user-profiling-and-segmentation-usingml-project/user_profiles_for_ads (1).csv") df.head(3)
   User  ID      Age   Gender   Location  Language  Education
Level  \
0        1    25-34     Female     Suburban         Hindi
         Technical
1        2    65+    Male     Urban    Hindi           PhD
2        3     45-54     Female     Suburban     Spanish
         Technical
   Likes and Reactions  Followed Accounts Device Usage
\
0                 5640                  190  Mobile
                  Only
1                 9501              375      Tablet
2                 4775                  187  Mobile
                  Only
   Time   Spent   Online   (hrs/weekday)     Time   Spent   Online
(hrs/weekend)  \
0                             4.5                        1.7
1                             0.5                        7.7
2                             4.5                        5.6


   Click-Through Rates (CTR)  Conversion Rates  Ad Interaction Time
(sec)  \

0                    0.193            0.067
25
1                    0.114            0.044
68
2                    0.153            0.095
80
  Income Level         Top Interests
0        20k-40k     Digital
         Marketing
1        0-20k          Data Science
2        60k-80k  Fitness and
         Wellness
```

2.3 Initial Analysis on the dataset

```
df.tail(3)
     User ID    Age  Gender Location Language Education Level  \
997      998  18-24    Male    Rural    Hindi       Technical
998      999    65+    Male    Urban  English             PhD
999     1000  35-44  Female    Urban    Hindi     High School

     Likes and Reactions  Followed Accounts      Device Usage  \
997                 5736                218  Mobile + Desktop
998                 2992                260  Mobile + Desktop
999                 5388                394      Desktop Only

     Time Spent Online (hrs/weekday)  Time Spent Online (hrs/weekend)

997                              2.1                              2.4

\


998                              4.1                              2.7

999                              2.1                              5.6


     Click-Through Rates (CTR)  Conversion Rates  Ad Interaction Time
(sec)  \
997  0.154             0.070
91
998  0.031             0.025
147
999  0.145             0.076                          98
    Income Level                                        Top Interests
997        100k+  Investing and Finance, Data Science, Photograp...
998      60k-80k  Data Science, Eco-Friendly Living, Gaming, Tra...
999      40k-60k                     Data Science, DIY Crafts, Gaming
print("Shape of Dataset:", df.shape)

Shape of Dataset: (1000, 16)

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 16 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
0   User ID                             1000 non-null   int64
1   Age                                 1000 non-null   object
2   Gender                              1000 non-null   object
3   Location                            1000 non-null   object
4   Language                            1000 non-null   object
```

```
5   Education Level                     1000 non-null   object
6   Likes and Reactions                 1000 non-null   int64
7   Followed Accounts                   1000 non-null   int64
8   Device Usage                        1000 non-null   object
9   Time Spent Online (hrs/weekday)     1000 non-null   float64
10  Time Spent Online (hrs/weekend)     1000 non-null   float64
11  Click-Through Rates (CTR)           1000 non-null   float64
12  Conversion Rates                    1000 non-null   float64
13  Ad Interaction Time (sec)           1000 non-null   int64
14  Income Level                        1000 non-null   object  15
    Top Interests                       1000 non-null   object
dtypes: float64(4), int64(4), object(8)
memory usage: 125.1+ KB
```

2.3.1 Analysis Output(1)

- The Data Set consists of 1000 Rows and 16 Columns.
- The type of all the variables in the data set are in numerical or object format. (Integer Or Float)
- According to first impressions, there is no missing value(NaN Value) in the data set

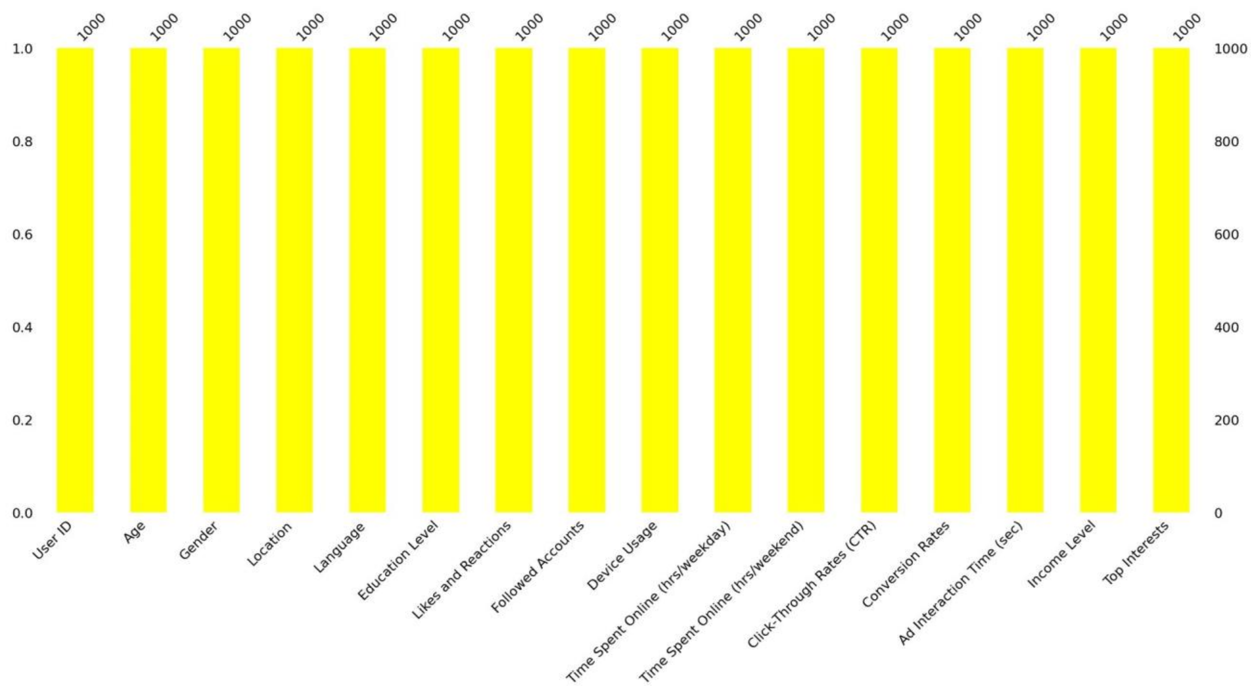
3. Preparation for Exploratory Data Analysis(EDA)

Go to Project Content

3.1 Examining Missing Values

```
df.isnull().sum()
User ID                    0
Age                        0
Gender                     0
Location                   0
Language                   0
Education Level            0
```

```
Likes and Reactions                    0
Followed Accounts                      0
Device Usage                           0
Time Spent Online (hrs/weekday)        0
Time Spent Online (hrs/weekend)        0
Click-Through Rates (CTR)              0
Conversion Rates                       0
Ad Interaction Time (sec)              0
Income Level                           0
Top Interests                          0
dtype: int64

isnull_number = [] for i in
df.columns:     x =
df[i].isnull().sum()
isnull_number.append(x)

pd.DataFrame(isnull_number, index = df.columns, columns = ["Total
Missing Values"])
                                 Total Missing Values
User ID                                             0
Age                                                 0
Gender                                              0
Location                                            0
Language                                            0
Education Level                                     0
Likes and Reactions                                 0
Followed Accounts                                   0
Device Usage                                        0
Time Spent Online (hrs/weekday)                     0
Time Spent Online (hrs/weekend)                     0
Click-Through Rates (CTR)                           0
Conversion Rates                                    0
Ad Interaction Time (sec)                           0
Income Level                                        0
Top Interests                                       0
import missingno
missingno.bar(df, color = "yellow")
plt.show()
```



3.2 Examining Unique Values

Go to Project Content

```
df["Location"].value_counts()
Location

Urban       350
Suburban    332
Rural       318
Name: count, dtype: int64
df["Location"].value_counts().sum()
1000
df["Device Usage"].value_counts()
Device Usage

Desktop Only        262
Mobile Only         253
Mobile + Desktop    250
Tablet              235
Name: count, dtype: int64
df["Education Level"].value_counts()
Education Level

Technical      211
Master         209
```

```
High School    205
Bachelor       189
PhD            186
Name: count, dtype: int64
df["Age"].value_counts()
Age

25-34    255
35-44    192
45-54    188
18-24    166
55-64    153
65+       46
Name: count, dtype: int64
df["Age"].value_counts()
Age

25-34    255
35-44    192
45-54    188
18-24    166
55-64    153
65+       46
Name: count, dtype: int64
df["Language"].value_counts()
Language

English     258
Spanish     251
Mandarin    250
Hindi       241
Name: count, dtype: int64
unique_number = [] for i in
df.columns:     x =
df[i].value_counts().count()
unique_number.append(x)

pd.DataFrame(unique_number, index = df.columns, columns = ["Total
Unique Values"])
                                 Total Unique Values
User ID                                         1000
Age                                                6
Gender                                             2
Location                                           3
Language                                           4
Education Level                                    5
```

```
Likes and Reactions                              944
```

```
Followed Accounts                                 428
Device Usage                                        4
Time Spent Online (hrs/weekday)                    46
Time Spent Online (hrs/weekend)                    71
Click-Through Rates (CTR)                         247
Conversion Rates                                  101
Ad Interaction Time (sec)                         175
Income Level                                        6
Top Interests                                     680
```

Analysis Outputs(2)

- **According to the result from the unique value dataframe;**
- We determined the variables with few unique values as categorical variables, and the variables with high unique values as numeric variables.
- In this context, **Numeric Variables:** "User ID",“age”, “likes and reactions”, “followed accounts”,"click through rates(CTR)","Conversion Rates","Adinteraction time(sec)","Income level", “time spend online(hrs/weekday)” and “time spend online(hrs/weekend) ”
- **Categorical Variables:** "gender", "location", "language", "education level", "device usuage", "top interest"
- In the next section, we will separate these 2 groups into 2 different lists.

3.3 Separating variables (Numeric or Categorical)

Go to Project Content

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 16 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
0   User ID                          1000 non-null   int64
1   Age                              1000 non-null   object
2   Gender                           1000 non-null   object
3   Location                         1000 non-null   object
4   Language                         1000 non-null   object
5   Education Level                  1000 non-null   object
6   Likes and Reactions              1000 non-null   int64
7   Followed Accounts                1000 non-null   int64
8   Device Usage                     1000 non-null   object
9   Time Spent Online (hrs/weekday)  1000 non-null   float64
10  Time Spent Online (hrs/weekend)  1000 non-null   float64
11  Click-Through Rates (CTR)        1000 non-null   float64
12  Conversion Rates                 1000 non-null   float64
13  Ad Interaction Time (sec)        1000 non-null   int64
```

```
14  Income Level                          1000 non-null   object
15  Top Interests                         1000 non-null   object
dtypes: float64(4), int64(4), object(8)
memory usage: 125.1+ KB


numeric_var = ["User ID","Age", "Likes and Reactions", "Followed
Accounts", "Time Spent Online (hrs/weekday)", "Time Spent Online
(hrs/weekend)","Click-Through Rates (CTR)","Conversion Rates","Ad
Interaction Time (sec)","Income Level"]
categoric_var = ["Gender", "Location", "Language", "Education Level",
"Device Usage",  "Top Interests"]
df[numeric_var].describe()



           User ID  Likes and Reactions  Followed Accounts  \
count  1000.000000          1000.000000        1000.000000
mean    500.500000          4997.084000         251.438000
std     288.819436          2838.494365         141.941557
min       1.000000           101.000000          10.000000
25%     250.750000          2661.250000         126.000000
50%     500.500000          5002.500000         245.500000
75%     750.250000          7348.750000         377.000000
max    1000.000000          9973.000000         498.000000

       Time Spent Online (hrs/weekday)  Time Spent Online
(hrs/weekend)  \

count                      1000.000000
1000.000000
mean                          2.757500
4.601600
std                           1.279735
2.026234
min                           0.500000
1.000000
25%                           1.700000
2.900000
50%                           2.800000
4.700000
75%                           3.800000
6.400000
max                           5.000000
8.000000
       Click-Through Rates (CTR)  Conversion Rates  Ad Interaction
Time (sec)
count                1000.000000       1000.000000
```

```
1000.000000
mean                        0.125333           0.049805
91.425000
                                       std                         0.071187
                                       0.028670
```

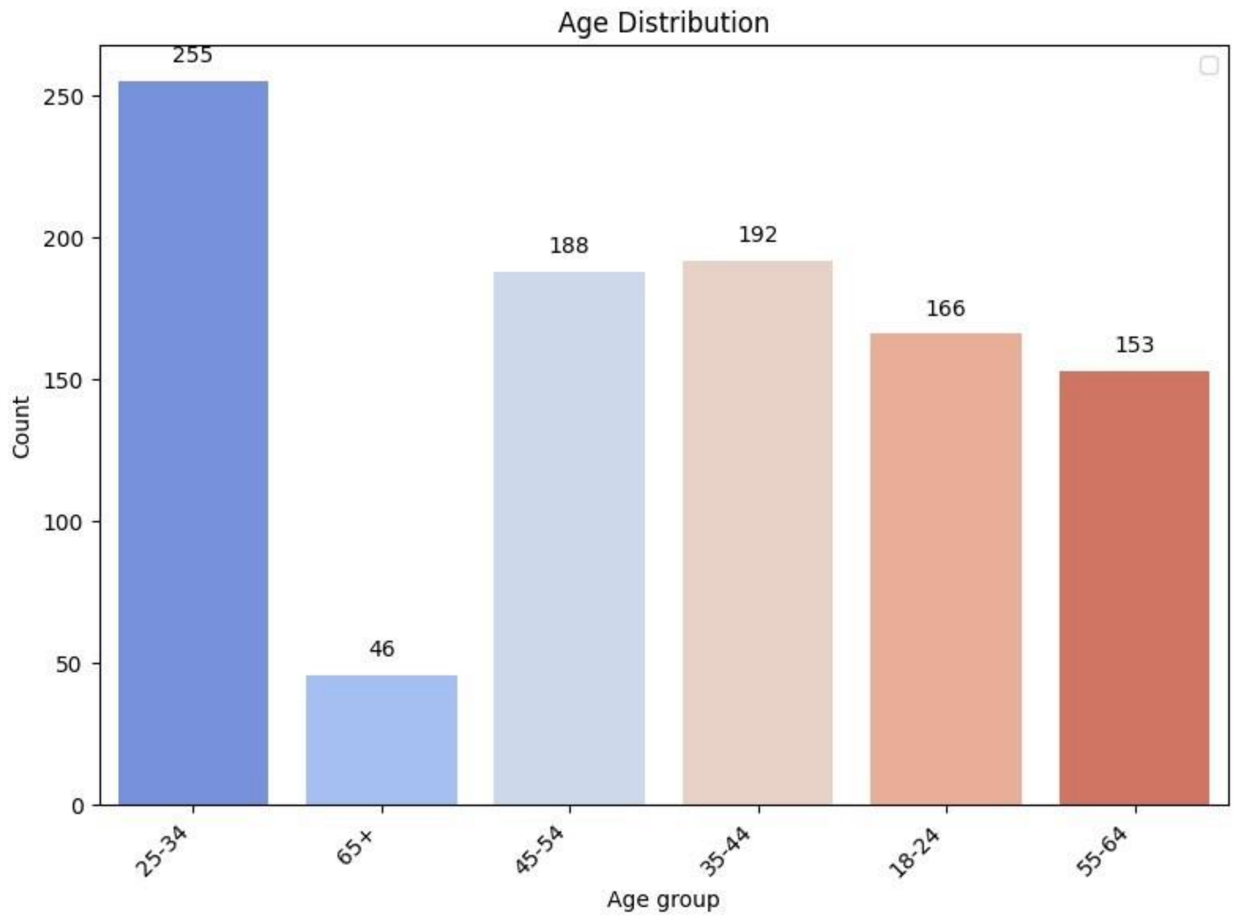
51.497965

```
min                        0.000000          0.000000
5.000000
25%                        0.065000          0.026000
45.750000
50%                        0.128000          0.049000
90.000000
75%                        0.186000          0.073000
137.250000
max                        0.250000          0.100000
179.000000  df['Age']

0        25-34
1        65+
2        45-54
3        35-44
4        25-34        ...
995      18-24
996      55-64
997      18-24
998      65+
999      35-44
Name: Age, Length: 1000, dtype: object
```

3.4.1 Analysis Output(3)

- Note: Different Graphics were used in the anaysis to develop visualization skills.

```
#Age distribution
fig, ax = plt.subplots(figsize=(8,6))

sns.countplot(x='Age', data=df, palette='coolwarm', ax=ax)

for bar in ax.containers[0]:      ax.text(bar.get_x() +
bar.get_width() / 2, bar.get_height() + 5, int(bar.get_height()),
            ha='center', va='bottom', fontsize=10)

plt.title("Age Distribution")
plt.xticks(rotation=45, ha='right')
plt.xlabel('Age group')  # Set meaningful x-axis label
plt.ylabel('Count')  # Set meaningful y-axis label
plt.tight_layout(pad=1)  # Adjust spacing for better readability
plt.legend() plt.show()
```

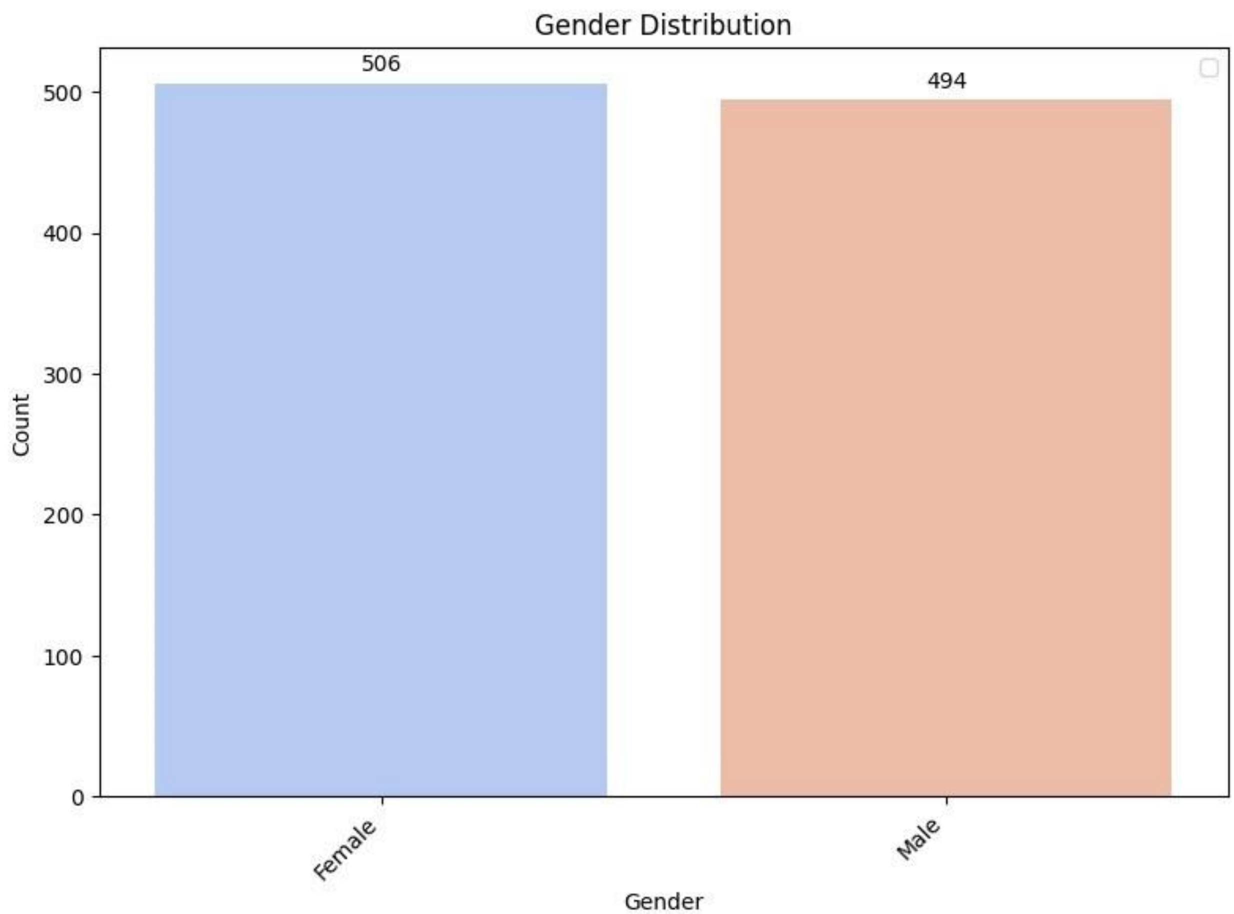


Analysis of "Age" variable according to Describe() method¶

- The minimum value of the ages is 18, and the maximum value is 65.
- So, if we don't look at other data, only these two data should mean that the midpoint must be 41.5 from the mathematical operation ((18 + 65) / 2).
- The mean of the data for the age is 36. Isn't the average of the minimum and maximum values that we found just by mathematical calculations 41?
- They are almost equal to each other.
- That means the age variable has a normal distribution. The normal distribution is the ideal statistical distribution for us.
- Let's look at the quartiles.
- The data average is in the middle of the 25% and 75% quarters.
- This shows that; There is an incredible right skew in the data.

```
#Gender Distribution
fig, ax = plt.subplots(figsize=(8,6))   # Set desired figure size

sns.countplot(x='Gender', data=df, palette='coolwarm', ax=ax)

for bar in ax.containers[0]:      ax.text(bar.get_x() + bar.get_width()
/ 2, bar.get_height() + 5,
```

```
int(bar.get_height()),
            ha='center', va='bottom', fontsize=10)

plt.title("Gender Distribution")
plt.xticks(rotation=45, ha='right')
plt.xlabel('Gender')  # Set meaningful x-axis label
plt.ylabel('Count')  # Set meaningful y-axis label
plt.tight_layout(pad=1)  # Adjust spacing for better readability
plt.legend() plt.show()
```

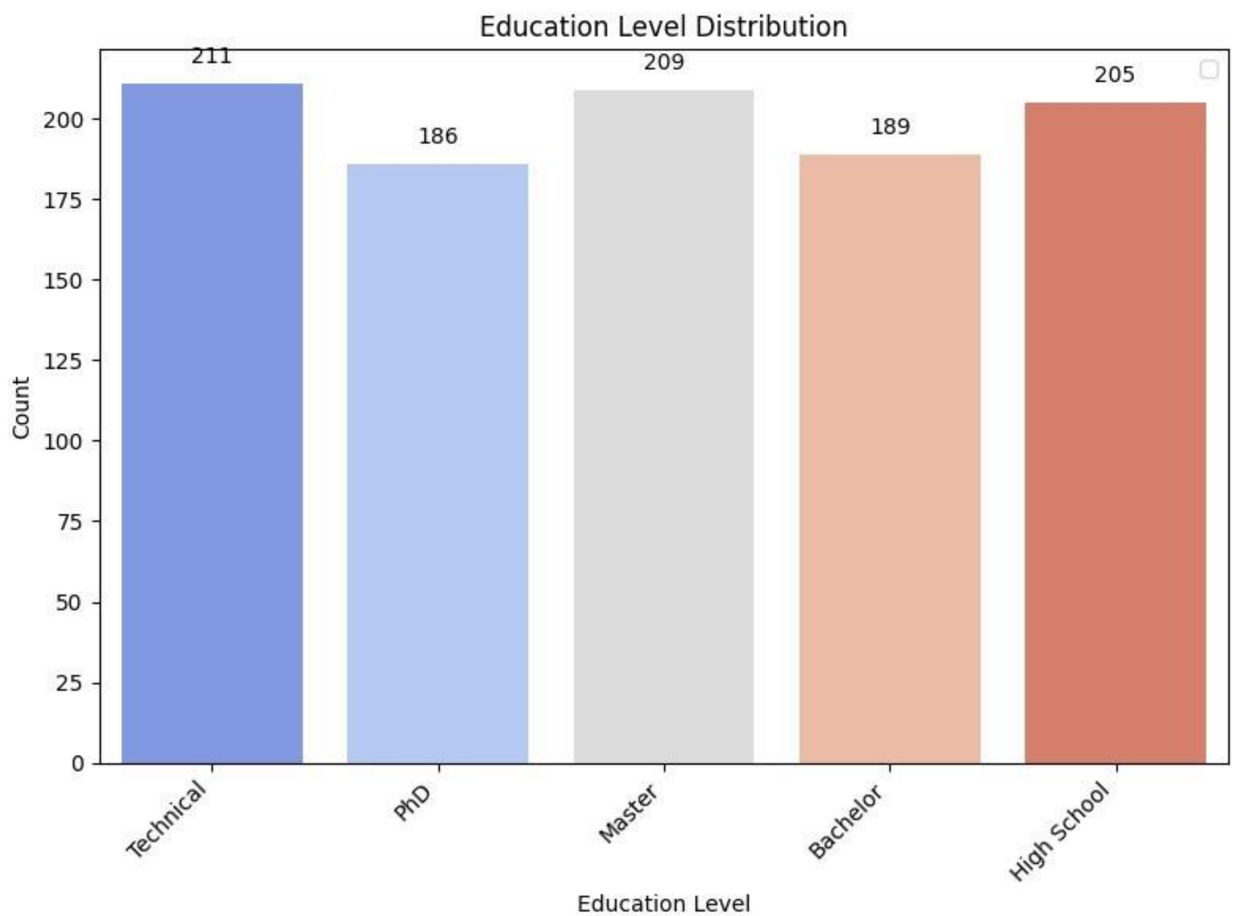


Analysis of "Gender"

- After looking at distribution we can say that this data having almost similar number of male and female.

```
# education level distribution
fig, ax = plt.subplots(figsize=(8,6))  # Set desired figure size

sns.countplot(x='Education Level', data=df, palette='coolwarm', ax=ax)
```

```
for bar in ax.containers[0]:     ax.text(bar.get_x() +
bar.get_width() / 2, bar.get_height() + 5, int(bar.get_height()),
            ha='center', va='bottom', fontsize=10)

plt.title("Education Level Distribution")
plt.xticks(rotation=45, ha='right')
plt.xlabel('Education Level')  # Set meaningful x-axis label
plt.ylabel('Count')  # Set meaningful y-axis label
plt.tight_layout(pad=1)  # Adjust spacing for better readability
plt.legend() plt.show()
```

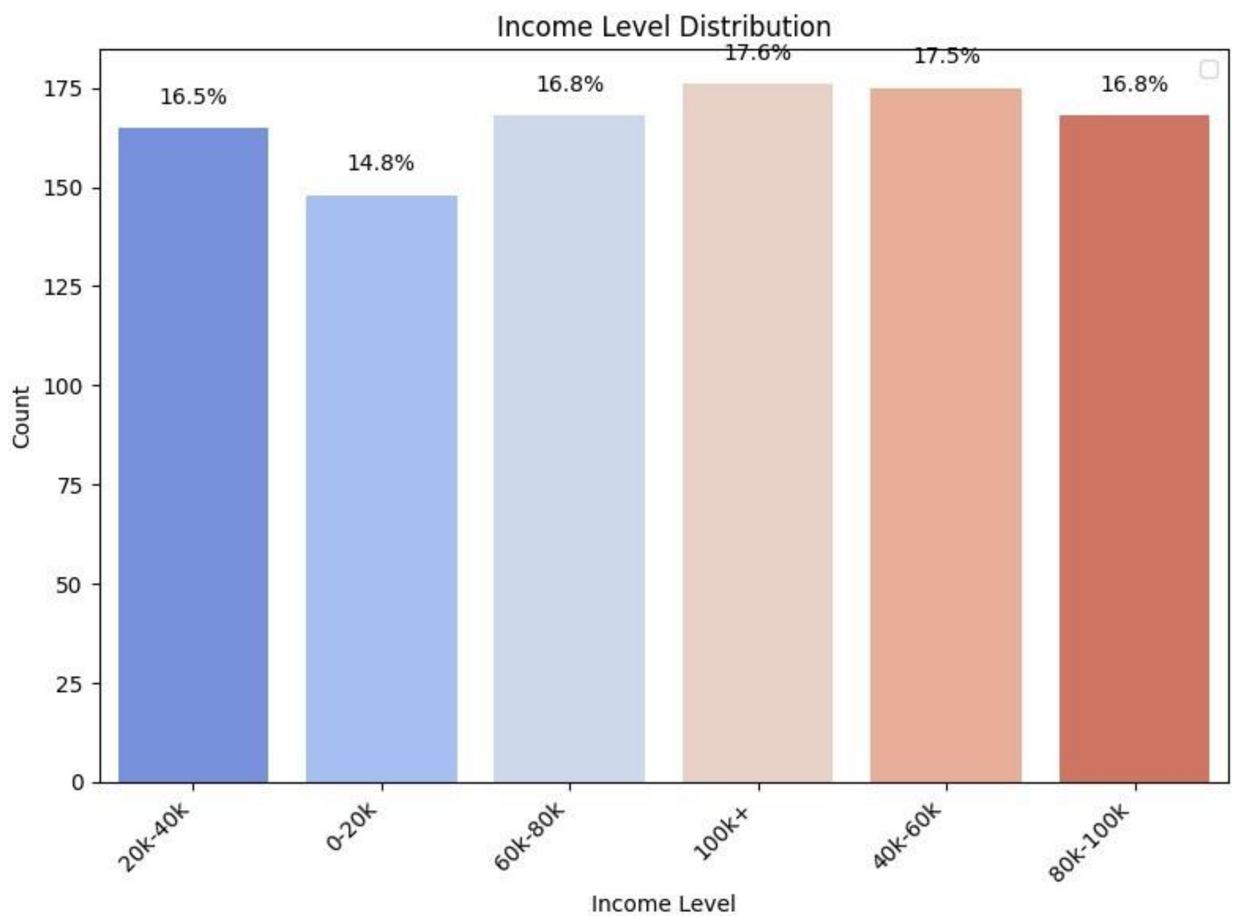


Analysis of "Education Level"

- After looking at distribution we can say that this data having most of the Technical level of people following with masters and high school degree, while the maximum no of users are 211 from technical, 209 from masters and 205 fro high school.

```
# Income Level Distribution
fig, ax = plt.subplots(figsize=(8, 6))  # Set desired figure size
```

```
sns.countplot(x='Income Level', data=df, palette='coolwarm', ax=ax)

total_count = df['Income Level'].count()

# Add data labels showing percentages
for bar in ax.containers[0]:     yval
= bar.get_height()
    percentage = (yval / total_count) * 100
    ax.text(bar.get_x() + bar.get_width() / 2, yval + 5,
f"{percentage:.1f}%",
            ha='center', va='bottom', fontsize=10)
plt.title("Income Level Distribution")
plt.xticks(rotation=45, ha='right')
plt.xlabel('Income Level')  plt.ylabel('Count')
plt.tight_layout(pad=1)  plt.legend() plt.show()
```

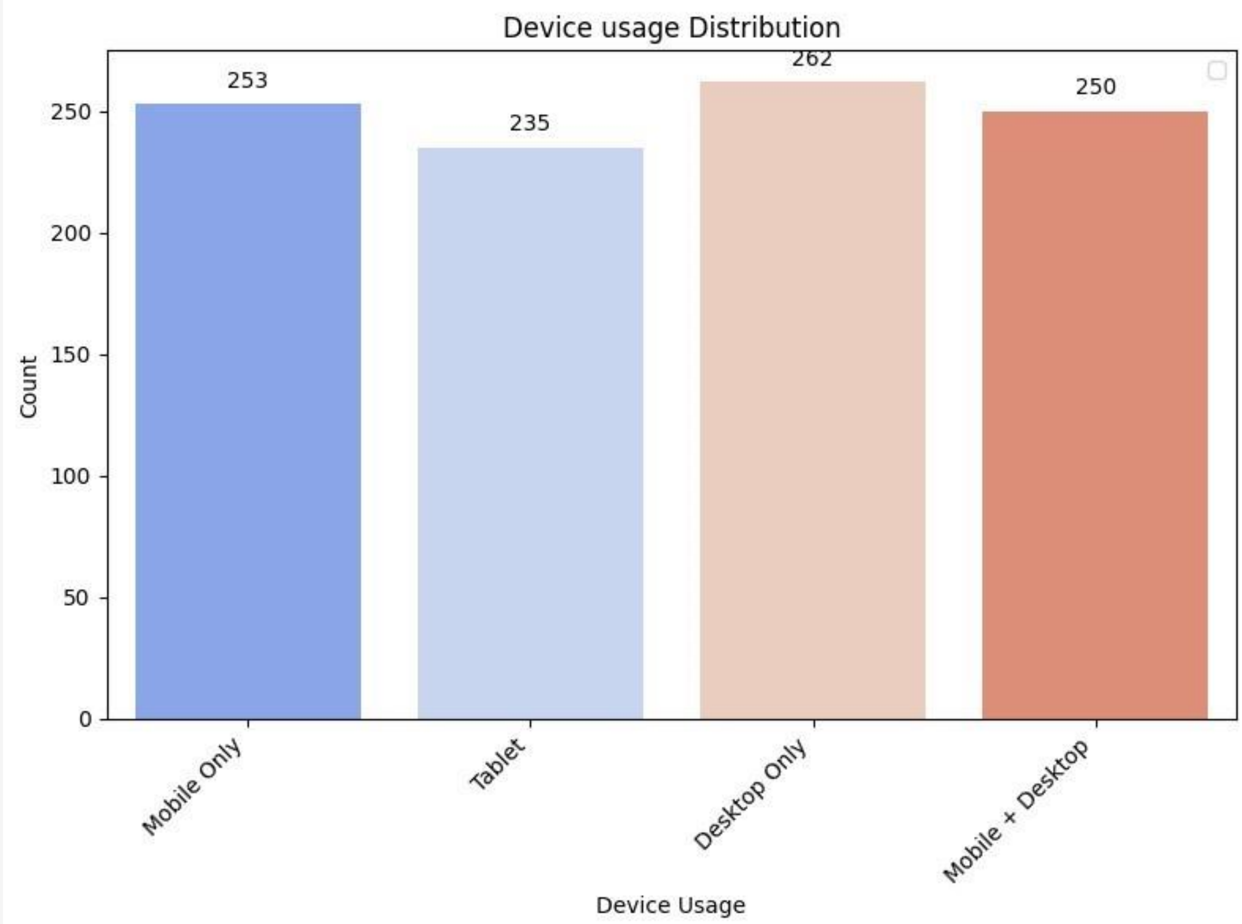


Analysis of "Income level"

- After looking at distribution w we can say that this data having highest 17.6% users belong to the upper bracket of income level is more than 100000 , followed by 17.5% are from the average level in the range 40k- 70k bracket.

- minimum salary as per the distribution is 10000.
- maximum salary is 100000.
- So, if we don't look at other data, only these two data should mean that the midpoint must be 55000 from the mathematical operation ((10000+ 100000) / 2).
- The mean of the data for the age is 59660. Isn't the average of the minimum and maximum values that we found just by mathematical calculations 41?
- They are almost equal to each other.
- That means the age variable has a normal distribution. The normal distribution is the ideal statistical distribution for us..

```
# Device usuage distribution
fig, ax = plt.subplots(figsize=(8,6))  # Set desired figure size

sns.countplot(x='Device Usage', data=df, palette='coolwarm', ax=ax)

for bar in ax.containers[0]:      ax.text(bar.get_x() + bar.get_width()
/ 2, bar.get_height() + 5, int(bar.get_height()),
            ha='center', va='bottom', fontsize=10)

plt.title("Device usage Distribution") plt.xticks(rotation=45,
ha='right')
plt.xlabel('Device Usage')  # Set meaningful x-axis label
plt.ylabel('Count')  # Set meaningful y-axis label
plt.tight_layout(pad=1)  # Adjust spacing for better readability
plt.legen
```




```
d() plt.show()
```

Analysis of "Device Usage"

- As per the visualisation we can say that the dataset having maximum no of users are using desktop only followed by mobile phone.
- We'll now examine device usage patterns to understand the primary means by which users access the platform. This information is crucial for optimizing ad formats and delivery channels. Additionally, we'll explore users' online behaviour, including their engagement with content and ads, and identify the most common interests among users. Let's proceed with analyzing device usage patterns:

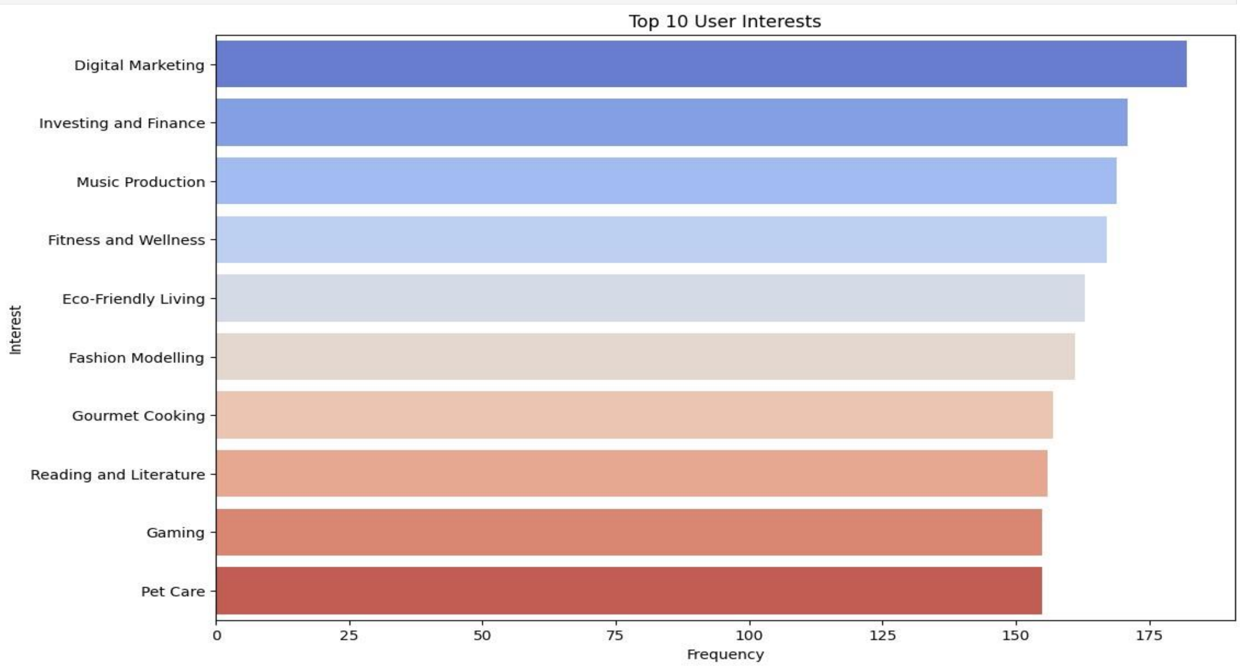
```python
# creating subplots for user online behavior and ad interaction
metrics
fig, axes = plt.subplots(3, 2, figsize=(18, 15))
fig.suptitle('User Online Behavior and Ad Interaction Metrics')

# time spent online on weekdays
sns.histplot(ax=axes[0, 0], x='Time Spent Online (hrs/weekday)', data=df,
bins=20, kde=True, color='skyblue') axes[0, 0].set_title('Time Spent
Online on Weekdays')

# time spent online on weekends
sns.histplot(ax=axes[0, 1], x='Time Spent Online (hrs/weekend)',
```

```python
data=df, bins=20, kde=True, color='orange') axes[0,
1].set_title('Time Spent Online on Weekends')

# likes and reactions
sns.histplot(ax=axes[1, 0], x='Likes and Reactions', data=df, bins=20,
kde=True, color='green')
axes[1, 0].set_title('Likes and Reactions')

# click-through rates
sns.histplot(ax=axes[1, 1], x='Click-Through Rates (CTR)', data=df,
bins=20, kde=True, color='red')
axes[1, 1].set_title('Click-Through Rates (CTR)')

# conversion rates
sns.histplot(ax=axes[2, 0], x='Conversion Rates', data=df, bins=20,
kde=True, color='purple')
axes[2, 0].set_title('Conversion Rates')

# ad interaction time
sns.histplot(ax=axes[2, 1], x='Ad Interaction Time (sec)', data=df,
bins=20, kde=True, color='brown')
axes[2, 1].set_title('Ad Interaction Time (sec)')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

Analysis

- Analyze the average time users spend online on weekdays versus weekends.
- Investigate user engagement metrics, such as likes and reactions.
- Delve into ad interaction metrics, including Click-Through Rates (CTR), Conversion Rates, and Ad Interaction Time.
- It will help us understand the users' activity patterns and their interaction with ads, which is crucial for effective ad targeting and optimization:

```
# Most common interesr among the users from
collections import Counter

# splitting the 'Top Interests' column and creating a list of all
interests
interests_list = df['Top Interests'].str.split(', ').sum()

# counting the frequency of each interest
interests_counter = Counter(interests_list)

# converting the counter object to a DataFrame for easier plotting

interests_df = pd.DataFrame(interests_counter.items(),
columns=['Interest', 'Frequency']).sort_values(by='Frequency',
ascending=False)

# plotting the most common interests
plt.figure(figsize=(12, 8))
sns.barplot(x='Frequency', y='Interest', data=interests_df.head(10),
palette='coolwarm')
plt.title('Top 10 User Interests')
plt.xlabel('Frequency')
plt.ylabel('Interest') plt.show()
```



Analysis of "Most common interest"

- As per the visualisation we can say that the dataset having users with most common interest is **"digita Marketing"**, **"Investing and Finance"**,**"Music Production"**,**"Fitness and wellness"** followed by eco friendly living , fashion modelling etc.,

4. Exploratory Data Analysis(EDA)

Go to Project Content

4.1.1 Numerical Variables(Analysis with Distplot)

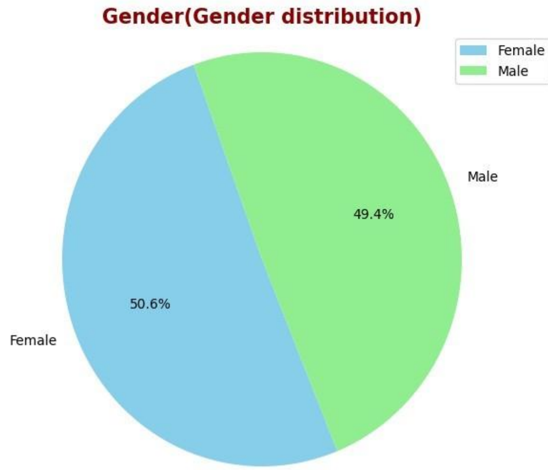
```
numeric_var
```

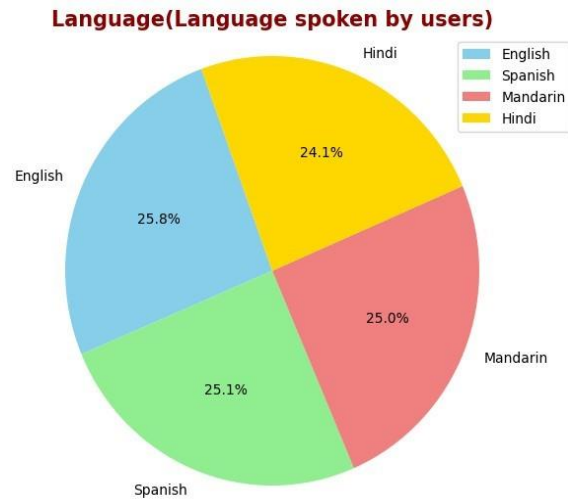
```
['User ID',
 'Age',
 'Likes and Reactions',
 'Followed Accounts',
 'Time Spent Online (hrs/weekday)',
 'Time Spent Online (hrs/weekend)',
 'Click-Through Rates (CTR)',
 'Conversion Rates',
 'Ad Interaction Time (sec)',
 'Income Level']
```

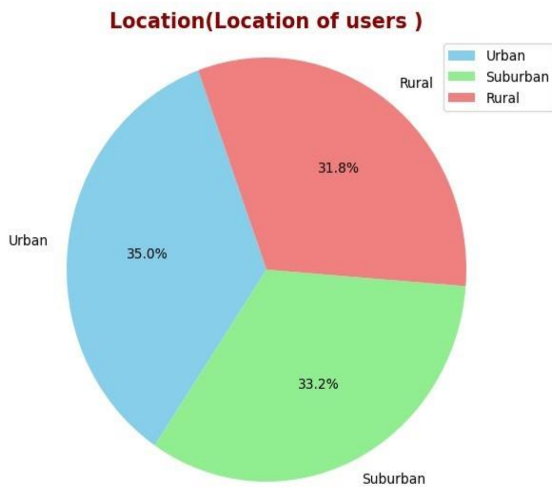
4.1.2 Categorical Variables(Analysis with Pie Chart)

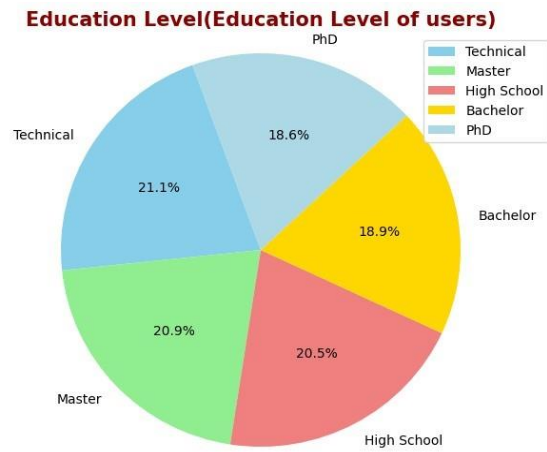
Go to Project Content

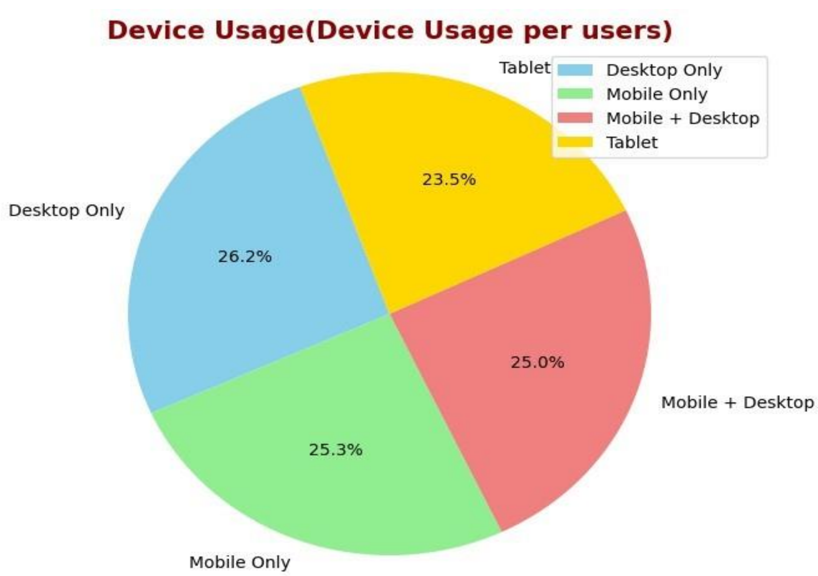
```
import matplotlib.pyplot as plt

categoric_axis_name = ["Gender distribution", "Location of users ",
"Language spoken by users", "Education Level of users",
"Device Usage per users"]
title_font = {"family": "arial", "color": "darkred", "weight": "bold",
"size": 15}
axis_font = {"family": "arial", "color": "darkblue", "weight": "bold",
"size": 13}

# Define a color list for pie slices (you can customize this) colors
= ['skyblue', 'lightgreen', 'lightcoral', 'gold', 'lightblue']


for i, z in zip(categoric_var, categoric_axis_name):
fig, ax = plt.subplots(figsize=(8, 6))

    observation_values = list(df[i].value_counts().index)
total_observation_values = list(df[i].value_counts())

    # Use colors list to set pie slice colors
    ax.pie(total_observation_values, labels=observation_values,
autopct='%1.1f%%',
            startangle=110, labeldistance=1.1,
colors=colors[:len(total_observation_values)])  # Slice colors

    ax.axis("equal")  # Equal aspect ratio ensures that pie is drawn as
a circle.

    plt.title((i + "(" + z + ")"), fontdict=title_font)  # Naming Pie
Chart Titles

    plt.legend()
plt.show()
```


Gender(Gender distribution)
Female
Male
49.4%
50.6%
Male
Female


Language(Language spoken by users)
Hindi
English
Spanish
Mandarin
Hindi
English
24.1%
25.8%
25.0%
25.1%
Mandarin
Spanish


Location(Location of users)
Urban
Suburban
Rural
Rural
31.8%
Urban
35.0%
33.2%
Suburban


Education Level(Education Level of users)
PhD
Technical
Master
High School
Bachelor
PhD
Technical
18.6%
21.1%
Bachelor
18.9%
20.9%
20.5%
Master
High School


Device Usage(Device Usage per users)
Tablet
Desktop Only
Mobile Only
Mobile + Desktop
Tablet
23.5%
Desktop Only
26.2%
25.0%
Mobile + Desktop
25.3%
Mobile Only

4.1.2.1 Analysis Outputs(5)

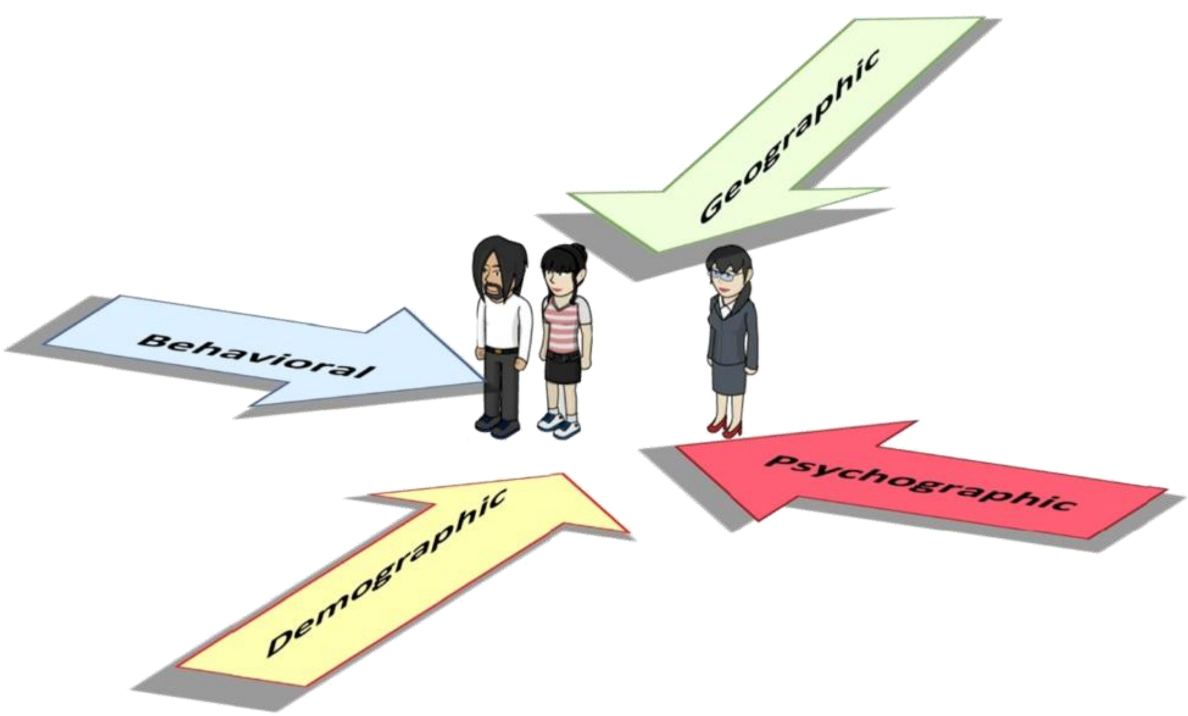
- Gender Variable
  - 49.4% of the patients are male, 50.6% are female.
  - So, the number of female user and male users are almost same.
- Language Variable
  - Users are equally disributed in four languages they watch content.
- Education level
  - highest number of users are from technical background followed by Master snd high school
- Device Usage
  - Equally distributed, while the highest users are from desktop with 26.2%.

Preparation for Modeling

Go to project Content

5.1 User Profiling and Segmentation

We can now segment users into distinct groups for targeted ad campaigns. Segmentation can be based on various criteria, such as:



- **Demographics:** Age, Gender, Income Level, Education Level
- **Behavioural:** Time Spent Online, Likes and Reactions, CTR, Conversion Rates
- **Interests:** Aligning ad content with the top interests identified

To implement user profiling and segmentation, we can apply clustering techniques or develop personas based on the combination of these attributes. This approach enables the creation of more personalized and effective ad campaigns, ultimately enhancing user engagement and conversion rates. 5.2 Literature review

User Profiling and Segmentation: A Review of Recent Research (2022-2024)

User profiling and segmentation are fundamental concepts in marketing, customer relationship management (CRM), and various recommendation systems. By understanding user characteristics and preferences, businesses can personalize their offerings, optimize marketing campaigns, and ultimately drive customer engagement and satisfaction. This review explores recent research advancements in user profiling and segmentation, focusing on publications from 2022 to 2024.

1. Leveraging AI and Machine Learning

Recent research highlights the increasing adoption of Artificial Intelligence (AI) and Machine Learning (ML) techniques for user profiling and segmentation. A study by He and Li (2023) [1] proposes a data mining approach for developing a customer profiling system that utilizes boosting trees for prediction and RFM analysis for customer equity estimation. This approach demonstrates the effectiveness of combining traditional marketing frameworks with advanced ML algorithms.

Furthermore, research by Xiao et al. (2023) [2] explores the application of deep learning for user profiling in recommender systems. Their findings suggest that deep learning models can outperform traditional methods in capturing complex user behavior patterns and preferences, leading to more accurate recommendations.

2. Multi-source Data Integration

The importance of incorporating data from various sources for user profiling is gaining traction. A research article by Wang et al. (2023) [3] emphasizes the benefits of integrating social media data, website browsing behavior, and purchase history to create more comprehensive user profiles. This multi-source approach allows for a more holistic understanding of user needs and interests.

3. Privacy-Preserving Techniques

With growing concerns around user privacy, research is actively exploring methods for user profiling and segmentation that adhere to data privacy regulations. A study by Li et al. (2024) [4] proposes a federated learning framework for user profiling that protects user data privacy while still enabling effective profile creation. This approach allows for collaborative learning across different data silos without compromising individual user information.

4. Ethical Considerations

The ethical implications of user profiling and segmentation are also being addressed in recent research. A paper by Zhang et al. (2023) [5] emphasizes the importance of transparency and fairness in user profiling algorithms. They advocate for explainable AI techniques that allow users to understand how their profiles are generated and used.

5. Future Directions

As research in user profiling and segmentation continues to evolve, future directions include:

- The exploration of explainable AI (XAI) techniques to further enhance user trust and transparency.
- The development of real-time user profiling methods to capture dynamic user behavior.
- The integration of user profiling with advanced marketing automation tools for personalized customer experiences.

5.3 RESEARCH SCOPE & METHODOLOGY

The research adopts a mixed-method approach, combining quantitative analysis with qualitative insights to provide a holistic understanding of User Ad data. The methodology encompasses the following steps:

1. **Data Collection:** Comprehensive datasets of ad Dataset are collected from Statis.io
2. **Data Preprocessing:** Clean the dataset by addressing missing values, outliers, and inconsistencies.
3. **Exploratory Data Analysis (EDA):** Conduct EDA to gain insights into the overall trends, distribution, and any apparent pattern.
4. **Model Selection:** To implement user profiling and segmentation, we can apply clustering techniques or develop personas based on the combination of these attributes. This approach enables the creation of more personalized and effective ad campaigns, ultimately enhancing user engagement and conversion rates.
5. **implementing clustering Techniques:** K-means clustering is a famous method of unsupervised machine learning. This method obtains all of the diverse “clusters” and clubs them collectively while maintaining them as tiny as attainable.

Algorithms works in this manner:

First, we randomly initialize the value of k as the number of clusters or n- centroids. Next, we allot each data points to the nearest centroid forming separate groups while relocating the center to the middle of all cluster employing euclidian distance. While working through the preceding steps, the algorithm checks and tries to reduce the sum of squared distances among clustered-point and middle for all clusters. When all data points unite, repetition ends.

1. **Tuning The Optimal Hyperparameters For The Model** Determining the most beneficial kit of hyperparameters for an algorithm is the subsequent measure in customer segments with Ml because it assists us in attaining the most genuine and satisfying customer crowds.

While choosing the k value, we will select upon the optimization principles of the K-means, inertia, practicing the elbow method.

With the elbow method, we will decide the k value wherever the drop in the inertia sustains.

1. **Visualization Of The Results** At last, we visualize the decisions applying the open-source Plotly-Python, a plotting library in python for making interactive graphs, plots, and charts. Then we understand the charts and various graphs to develop our enterprise.

Possessing genuine consumer profiles at your fingertips will help enhance marketing operations targeting, innovation launches, and the merchandise roadmap.

It will provide your organization exceptionally more evident thoughts about which customers have the most effective retention rate, contracts, and additional metrics you initially planned.

1. **Documentation and Reporting:** Document the entire methodology, including data preprocessing, model selection, and parameter estimation. Prepare a comprehensive report outlining the research methodology, results, and conclusions, making the research findings accessible to a broad audience

5.4 Data preprocessing and Model building

Let's start by selecting a subset of features that could be most indicative of user preferences and behaviour for segmentation and apply a clustering algorithm to create user segments:

Importing necessary libraries

```
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.cluster import KMeans
```

Feature Selection for Clustering

```
# selecting features for clustering
features = ['Age', 'Gender', 'Income Level', 'Time Spent Online 
(hrs/weekday)', 'Time Spent Online (hrs/weekend)', 'Likes and 
Reactions', 'Click-Through Rates (CTR)']

# separating the features we want to consider for clustering
X = df[features]
```

Data Preprocessing:

Data preprocessing is a crucial step in machine learning to prepare your data for model training. It involves transforming raw data into a format suitable for analysis by algorithms. Here's a breakdown of common techniques for numerical and categorical features:

- Numerical Features:

StandardScaler: This technique scales numerical features to have a mean of 0 and a standard deviation of 1. This is useful when features have different scales, putting them on a "level playing field" for the model. Here we have features like 'Time Spent Online (hrs/weekday)', 'Time Spent Online (hrs/weekend)', 'Likes and Reactions', 'Click-Through Rates (CTR)'. StandardScaler ensures these features with vastly different scales contribute equally to the model.

- Categorical Features:

One-Hot Encoding: This method converts categorical features with unique values (e.g., "Location") into binary features. Each category gets its own new feature, with a value of 1 indicating membership in that category and 0 otherwise. Here we Consider a feature 'Age', 'Gender', 'Income Level'. One-Hot Encoding would create three new features based on the unique values in each column.Each data point would have a 1 in exactly one of these features depending on its original country.

Benefits of Preprocessing:

- Improved Model Performance: Preprocessing helps algorithms converge faster and potentially achieve better accuracy.
- Reduced Bias: Scaling numerical features prevents features with larger scales from dominating the model.
- Enhanced Interpretability: One-Hot Encoding allows models to understand the relationships between different categories in categorical features.

Remember: The choice of preprocessing techniques depends on your specific dataset and machine learning task. Experiment with different approaches to see what works best for your model.

```
# defining preprocessing for numerical and categorical features
numeric_features = ['Time Spent Online (hrs/weekday)', 'Time Spent
Online (hrs/weekend)', 'Likes and Reactions', 'Click-Through Rates
(CTR)']
numeric_transformer = StandardScaler()

categorical_features = ['Age', 'Gender', 'Income Level']
categorical_transformer = OneHotEncoder()

# combining preprocessing steps
preprocessor = ColumnTransformer(
transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat',  categorical_transformer,  categorical_features)
])
```

6. Model Building

6.1 Clustering pipelines

Clustering pipelines are workflows that automate the process of clustering data. They combine multiple data preparation, transformation, and clustering steps into a single, repeatable process.

Here's a breakdown of clustering pipelines:

Components:

- **Data Loading:** The pipeline starts by loading the data from its source (CSV, database, etc.).

- **Preprocessing:** This stage might include cleaning missing values, handling outliers, and scaling numerical features. Techniques like standard scaling for numerical data and onehot encoding for categorical data are often used here.
- **Feature Selection:** You might choose to select a subset of relevant features to improve clustering performance and reduce processing time.
- **Clustering Algorithm:** This is the core of the pipeline, where the chosen clustering algorithm (e.g., k-means, hierarchical clustering) is applied to the prepared data.
- **Evaluation:** The pipeline can evaluate the quality of the resulting clusters using metrics like silhouette score or Calinski-Harabasz index.
- **Output:** Finally, the pipeline outputs the clustered data, visualizations, or other relevant information for further analysis.

Benefits:

- **Efficiency:** Clustering pipelines save time by automating repetitive tasks.
- **Reproducibility:** Pipelines ensure consistency in the clustering process, allowing you to rerun the analysis with the same steps.
- **Scalability:** They can be easily scaled to handle larger datasets.
- **Modularity:** Individual steps can be modified or replaced for experimentation with different preprocessing techniques or clustering algorithms.

Frameworks:

Several popular libraries and frameworks provide tools for building clustering pipelines:

- **scikit-learn (Python):** Offers a rich set of tools for data preprocessing, feature selection, and various clustering algorithms.
- **Spark MLlib (Apache Spark):** Enables distributed clustering on large datasets.
- **KNIME (Open-source platform):** Provides a visual interface for building data pipelines, including clustering workflows.

Use Cases:

Clustering pipelines are used in various applications, including:

- **Customer segmentation:** Grouping customers based on purchase history and demographics for targeted marketing campaigns.
- **Image segmentation:** Identifying and separating objects in an image.
- **Anomaly detection:** Detecting data points that deviate significantly from the norm.

By using clustering pipelines, you can streamline the process of uncovering hidden patterns and insights within your data.

```
# creating a preprocessing and clustering pipeline pipeline
= Pipeline(steps=[('preprocessor', preprocessor),
('cluster', KMeans(n_clusters=5, random_state=42))])

pipeline.fit(X)
cluster_labels = pipeline.named_steps['cluster'].labels_
df['Cluster'] = cluster_labels print(df.head())
   User ID    Age  Gender  Location Language Education Level  \
0        1  25-34  Female  Suburban    Hindi       Technical
1        2    65+    Male     Urban    Hindi             PhD
2        3  45-54  Female  Suburban  Spanish       Technical
3        4  35-44  Female     Rural  Spanish             PhD
4        5  25-34  Female     Urban  English       Technical

   Likes and Reactions  Followed Accounts  Device Usage  \
0                 5640                190   Mobile Only
1                 9501                375        Tablet
2                 4775                187   Mobile Only
3                 9182                152  Desktop Only
4                 6848                371   Mobile Only

   Time Spent Online (hrs/weekday)  Time Spent Online (hrs/weekend)  \
```

```
  Income Level                                              Top Interests
Cluste
       20k-40k                                          Digital Marketing

         0-20k                                               Data Science

       60k-80k                                      Fitness and Wellness

         100k+                                        Gaming, DIY Crafts

       20k-40k  Fitness and Wellness, Investing and Finance, G...

0                             4.5                                1.7
1                             0.5                                7.7
2                             4.5                                5.6
3                             3.1                                4.2
4                             2.0                                3.8
   Click-Through Rates (CTR)  Conversion Rates  Ad Interaction Time
(sec)  \

0                     0.193            0.067
25
1                     0.114            0.044
68
2                     0.153            0.095
80
3                     0.093            0.061
65
4                     0.175            0.022
99
0
```

```
1
1
0
2
3
3
1
4
1
```

6.2 Clustering Model Output

The clustering process has successfully segmented our users into five distinct groups (Clusters 0 to 4). Each cluster represents a unique combination of the features we selected, including age, gender, income level, online behaviour, and engagement metrics. These clusters can serve as the basis for creating targeted ad campaigns tailored to the preferences and behaviours of each segment.

6.3 Computing mean value of features

We'll compute the mean values of the numerical features and the mode for categorical features within each cluster to get a sense of their defining characteristics:

```
# computing the mean values of numerical features for each cluster
cluster_means = df.groupby('Cluster')[numeric_features].mean() for
feature in categorical_features:
```

```
    mode_series = df.groupby('Cluster')[feature].agg(lambda x:
x.mode()[0])
    cluster_means[feature] = mode_series
print(cluster_means)
          Time Spent Online (hrs/weekday)  Time Spent Online
(hrs/weekend)
\


0                                1.632955
6.135795
1                                2.937500
2.735000
2                                3.364532
6.151724
3                                3.872986
4.624171
4                                1.558235
3.769412
          Likes and Reactions  Click-Through Rates (CTR)    Age  Gender

Cluste

\


0                 5480.022727                   0.173705  25-34    Male

1                 7462.233333                   0.152983  25-34    Male

2                 5997.108374                   0.058502  25-34    Male

3                 2409.625592                   0.167123  25-34  Female

4                 3034.235294                   0.064153  25-34  Female
        Income Level


0              80k-
               100k
1              100k+
2              60k-
               80k
3              60k-
               80k
4              0-20k
```

6.4 Assigning names to each Cluster

Now, we'll assign each cluster a name that reflects its most defining characteristics based on the mean values of numerical features and the most frequent categories for categorical features.
Based on the cluster analysis, we can summarize and name the segments as follows:

- **Cluster 0 –"Weekend Warriors"**: High weekend online activity, moderate likes and reactions, predominantly male, age group 25-34, income level 80k-100k.
- **Cluster 1 – "Engaged Professionals"**: Balanced online activity, high likes and reactions, predominantly male, age group 25-34, high income (100k+).
- **Cluster 2 – "Low-Key Users"**: Moderate to high weekend online activity, moderate likes and reactions, predominantly male, age group 25-34, income level 60k-80k, lower CTR.
- **Cluster 3 – "Active Explorers"**: High overall online activity, lower likes and reactions, predominantly female, age group 25-34, income level 60k-80k.
- **Cluster 4 – "Budget Browsers"**: Moderate online activity, lowest likes and reactions, predominantly female, age group 25-34, lowest income level (0-20k), lower CTR.

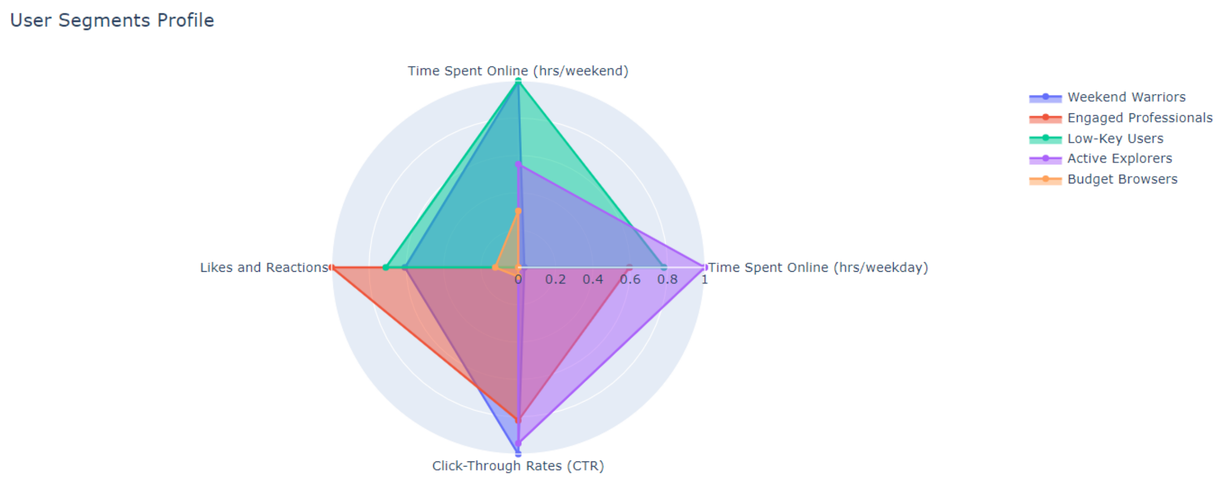
6.5 Visualization of Clusters Using Radar Chart

```
import numpy as np
import pandas as pd  # Import pandas for DataFrame manipulation

# preparing data for radar chart
features_to_plot = ['Time Spent Online (hrs/weekday)', 'Time Spent
Online (hrs/weekend)', 'Likes and Reactions', 'Click-Through Rates
(CTR)']
labels = np.array(features_to_plot)

# creating a dataframe for the radar chart
radar_df = cluster_means[features_to_plot].reset_index()

# normalizing the data
radar_df_normalized = radar_df.copy() for feature in
features_to_plot:   radar_df_normalized[feature] =
(radar_df[feature] - radar_df[feature].min()) /
(radar_df[feature].max() - radar_df[feature].min())

# Concatenate (append) the first row to the end for a full circle
first_row = radar_df_normalized.iloc[0]
radar_df_normalized = pd.concat([radar_df_normalized,
first_row.to_frame().T], ignore_index=True)  # Efficient concatenation

# assigning names to segments
segment_names = ['Weekend Warriors', 'Engaged Professionals', 'Low-Key
Users', 'Active Explorers', 'Budget Browsers']
```

Now, let's create a visualization that reflects these segments, using the cluster means for numerical features and highlighting the distinctive characteristics of each segment. We'll create a radar chart that compares the mean values of selected features across the clusters, providing a visual representation of each segment's profile:

```python
import plotly.graph_objects as go
fig = go.Figure()
```

```python
# loop through each segment to add to the radar chart
for i, segment in enumerate(segment_names):
fig.add_trace(go.Scatterpolar(
r=radar_df_normalized.iloc[i]
[features_to_plot].values.tolist() + [radar_df_normalized.iloc[i]
[features_to_plot].values[0]],  # Add the first value at the end to
close the radar chart
        theta=labels.tolist() + [labels[0]],  # add the first label at
the end to close the radar chart         fill='toself',
name=segment,         hoverinfo='text',
        text=[f"{label}: {value:.2f}" for label, value in
zip(features_to_plot, radar_df_normalized.iloc[i][features_to_plot])]+
[f"{labels[0]}: {radar_df_normalized.iloc[i][features_to_plot]
[0]:.2f}"]  # Adding hover text for each feature
    ))

# update the layout to finalize the radar chart
fig.update_layout(
polar=dict(
radialaxis=dict(
visible=True,
range=[0, 1]
        )),
showlegend=True,
    title='User Segments Profile'
)
fig.show()
```



The chart above is useful for marketers to understand the behaviour of different user segments and tailor their advertising strategies accordingly. For example, ads targeting the "Weekend Warriors" could be scheduled for the weekend when they are most active, while "Engaged Professionals" might respond better to ads that are spread evenly throughout the week.

7.Summary

So, this is how you can perform User Profiling and Segmentation using Python. User profiling refers to creating detailed profiles that represent the behaviours and preferences of users, and segmentation divides the user base into distinct groups with common characteristics, making it easier to target specific segments with personalized marketing, products, or services.

7.1 **Conclusion**

User profiling and segmentation remain critical tools for businesses to understand their customers and drive growth. Recent research advancements in AI, multi-source data integration, privacy-preserving techniques, and ethical considerations pave the way for more sophisticated and user-centric approaches to customer profiling and segmentation.

References:

- [1] - He, C., & Li, C. (2023, February). Customer profiling, segmentation, and sales prediction using AI in direct marketing. In 2023 International Conference on Neural Computing and Applications (NCA) (pp. 127-134). IEEE.
- [2]- Xiao, Y., Huang, H., & Zhou, M. (2023, in press). A deep learning approach for user profiling in recommender systems. Knowledge-Based Systems.
- [3]- Wang, Y., Sun, J., & Liu, X. (2023, January). User profiling based on multi-source data integration. In 2023 International Conference on Big Data and Smart Computing (BigDataSC) (pp. 1-6). IEEE.
- [4] - Li, J., Cheng, X., & Liu, J. (2024, April). Privacy-preserving user profiling using federated learning. In Proceedings of the 2024 ACM International Conference on Management of Data (SIGMOD '24) (pp. 2342-2348).

- [5] - Zhang, X., Wu, Q., & Zhou, L. (2023, June). Ethical considerations of user profiling and segmentation: A review. Journal of Business Ethics, 1-18.
- [6] - Dataset: statso.io/user-profiling-case-study/
- [7] - image generated through microsoft copilot designer https://copilot.microsoft.com/
- [8] - images downloaded from google https://m.indiamart.com/proddetail/segmentation-and-profiling-21227151312.html