Better By Far Report

Team Number: 9
Team Name: Nexus
DVP Team Members: Vaasanthi Ethakota, Praveen Kumar Kanuri
Date: November 15, 2023

Client Name: Better By Far

Offering: Social Media Platform

Brand: Better By Far

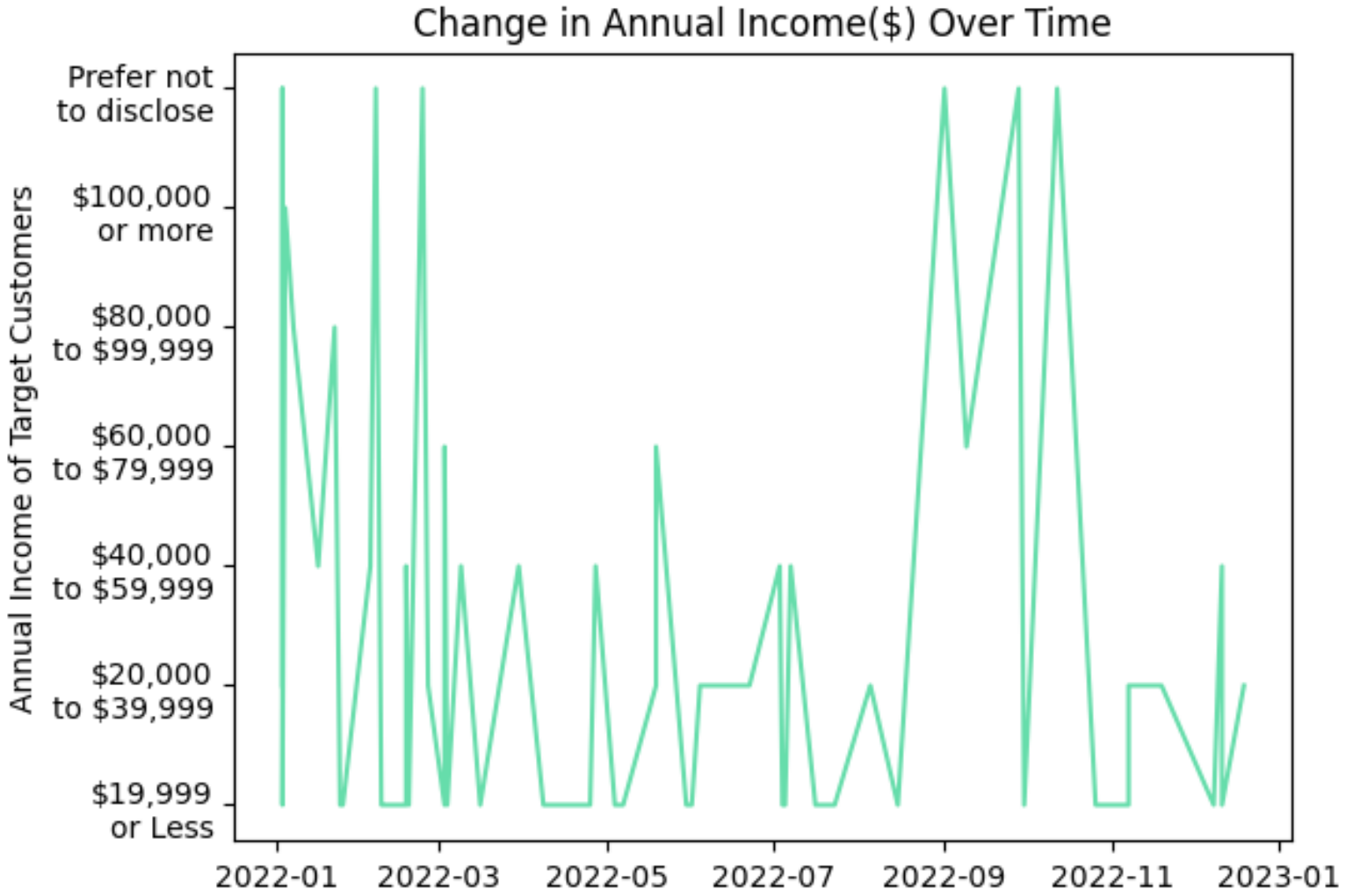
Contents

1. Brief description of the Use Case

Better By Far aims to launch a cutting-edge Social Media Platform, "Better By Far." Our Data Visualization Programming team, under Dr. Mead's guidance, is tasked with decoding seven pivotal datasets. Each module, aligned with Better By Far's objectives, unfolds a unique facet of user behavior, demographics, and intentions, paving the way for data-driven decisions.

1. **Data Exploration Journey:** Embark on a modular analysis journey, unraveling insights from the initial dataset to subsequent ones.
2. **Strategic Objectives:** Address Better By Far's seven objectives (M1 to M7), encompassing user behavior, demographics, and geographical representations.
3. **Actionable Insights:** Illuminate how the data insights will guide Better By Far's strategic decisions for a successful social media platform launch.

2 M2) User Behavior and Opinions Over Time.



Income Diversity Trends:

Identified gaps in income disclosure during specific months, emphasizing the need for targeted engagement strategies.

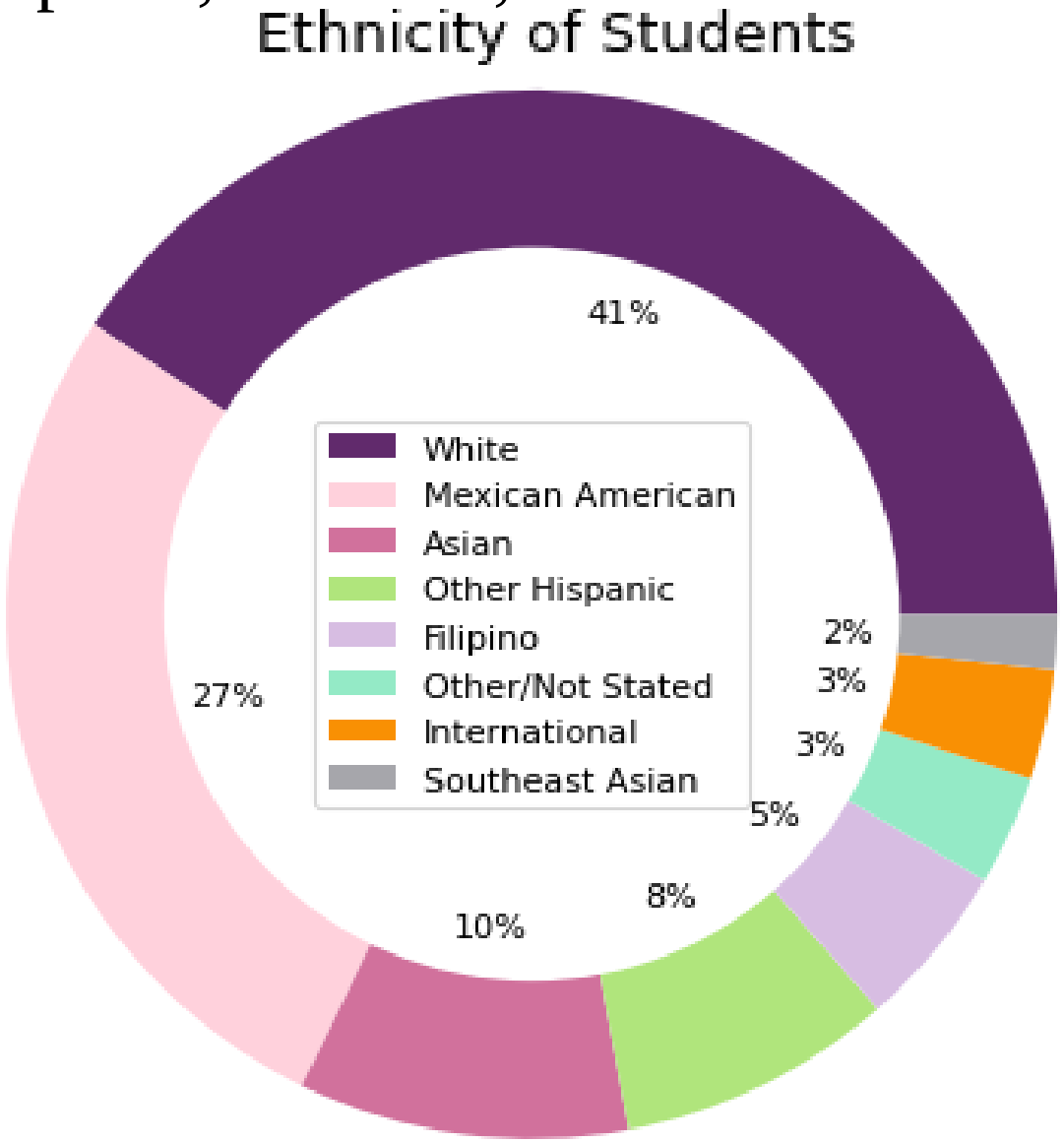
Discovered a significant portion of users reporting annual incomes over **$100,000**, indicating a high-purchasing-power user segment.

Proactive Engagement:

Implement targeted campaigns to address income disclosure gaps and encourage users to willingly share financial information.

Devise **exclusive features** and **personalized content** for high-income users to maximize engagement and satisfaction.

3. M3) Composition of User Demographics, Interests, and Intentions.



Diverse User Base:

- Uncovered a diverse user composition with **White (41%) and Mexican American (27%)** target customers as the majority, alongside **Southeast Asian** and **International** users (both at **2%**).

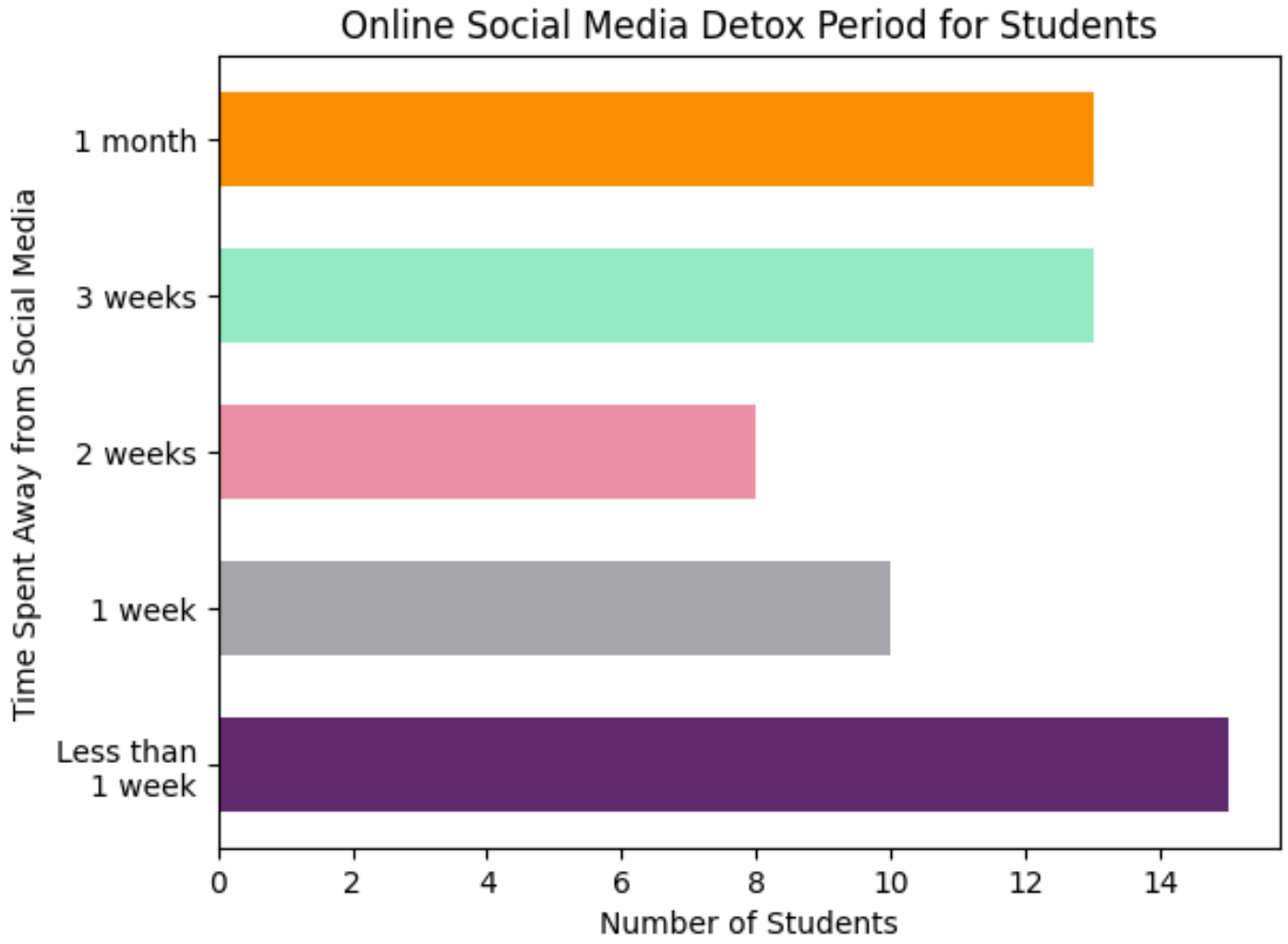
- Noted a significant opportunity for engagement with Mexican American users and the potential for international user outreach.

Culturally Inclusive Platform:

- Foster platform inclusivity by creating tailored content for the prominent Mexican American demographic.

- **Expand international appeal** through multilingual support, respecting user privacy, and implementing personalized marketing strategies for diverse user segments.

3. M3) Composition of User Demographics, Interests, and Intentions.



Diverse Break Preferences:

Uncovered diverse user preferences, with **15** students favoring very short social media breaks, **13** each opting for **3 weeks** and **1 month** off, and **10** students for a **one-week break**, indicating a balanced range of user choices.

Mid-Range Break Preferences:

Identified a mid-range preference among **10** students for a one-week break and **8** students for two weeks, suggesting a potential target segment for tailored features.

Tailored Break Features:

Implement platform features like **daily highlights** for users taking short breaks, fostering more frequent engagement.

Provide content such as productivity tips and personal development for those on one or two-week breaks, enhancing overall user experience and potentially **increasing retention**.

4.M4) Distributions of User Age, Income, Usage Intent, and Purchase Intent.

Diverse Work Hour Engagement:

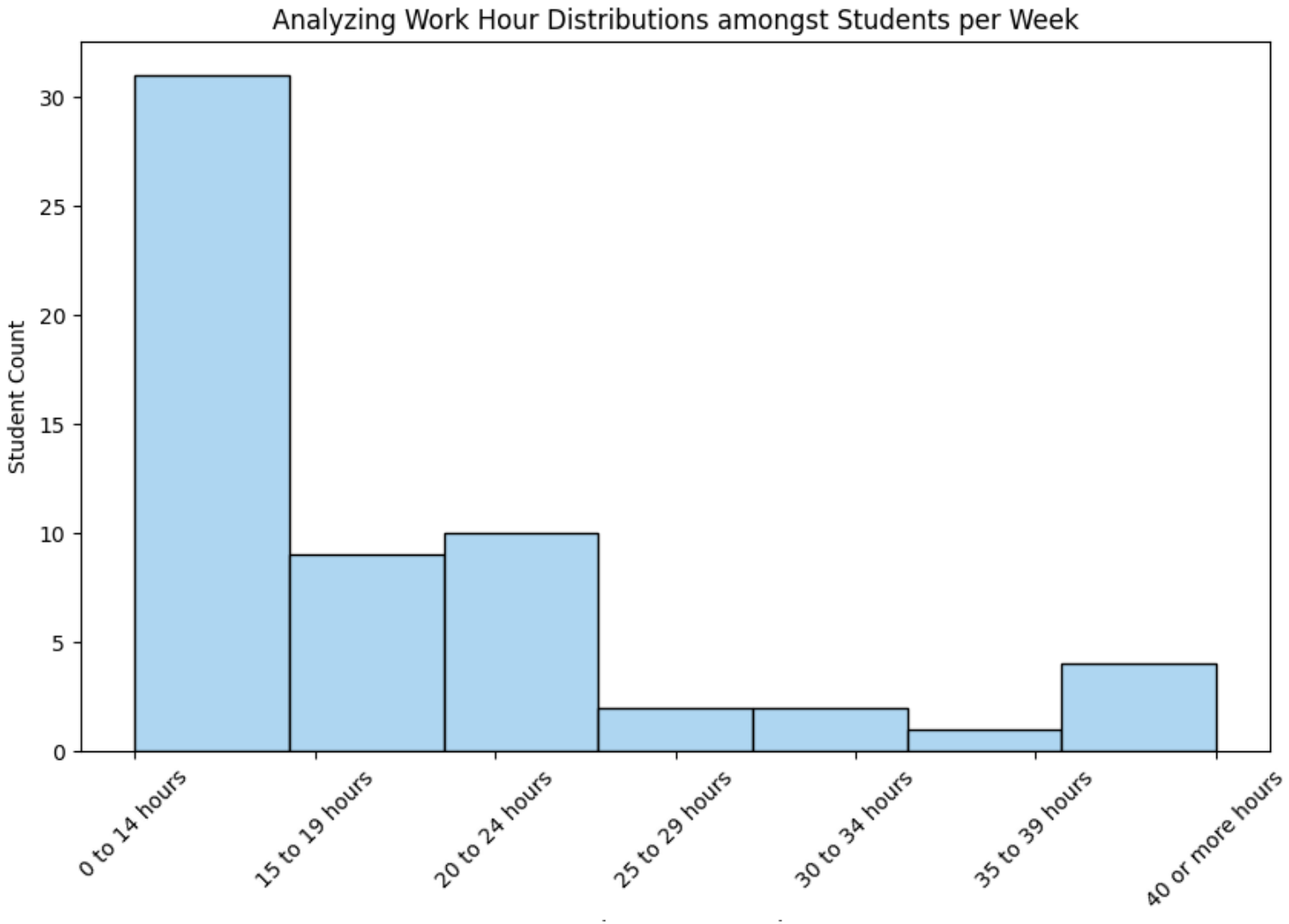
- A diverse user base with **31** members working **0-14 hours**, **10** members engaged for **20-24 hours**, and **4** members working **40 or more hours** per week reflects varying work commitments.

Engagement Segmentation:

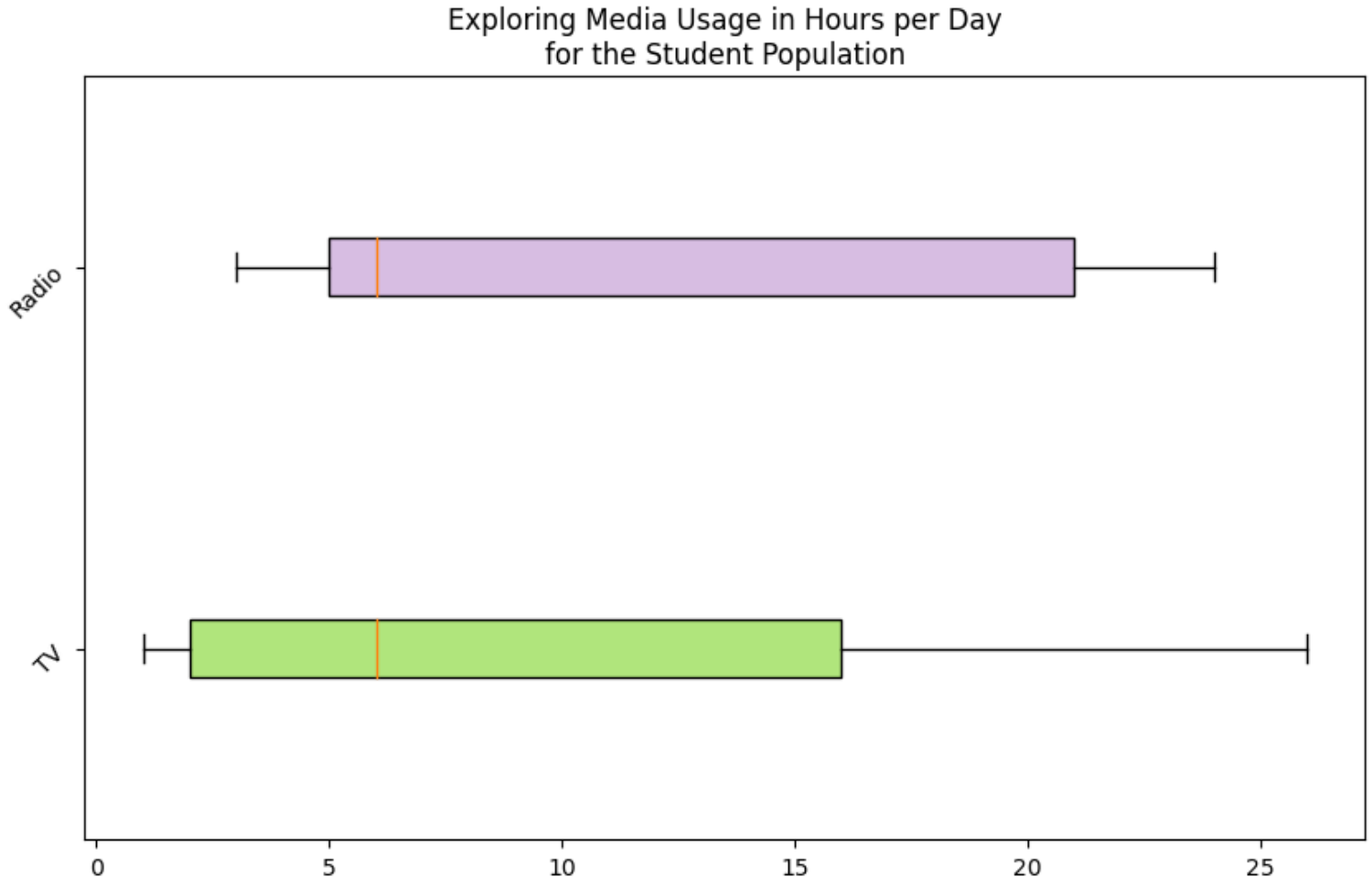
- Identified a moderately engaged group (20-24 hours) and a **highly engaged segment** (40 or more hours) within the user base.

Personalized Engagement Strategies:

- **Tailor engagement initiatives** based on users' work hours, providing a personalized experience for diverse engagement levels and ensuring sustained platform usage.



4.M4) Distributions of User Age, Income, Usage Intent, and Purchase Intent.



TV and Radio Dominance:

Majority of students prefer spending **1-3 hours** daily on **TV** and **Radio**, indicating their popularity as primary media choices.

Diverse Engagement Levels:

Some users display high engagement, investing **3-5 hours** or **5-7 hours** daily on TV and Radio, showcasing varied media consumption habits.

Targeted Ad Campaigns:

- Focus advertising efforts on users with higher TV and Radio engagement (3-5 hours, 5-7 hours) for **effective targeting**.

- Explore **alternative advertising channels** for users with lower media usage to ensure comprehensive outreach and engagement.

4.M4) Distributions of User Age, Income, Usage Intent, and Purchase Intent.

Limited Print Media Engagement:

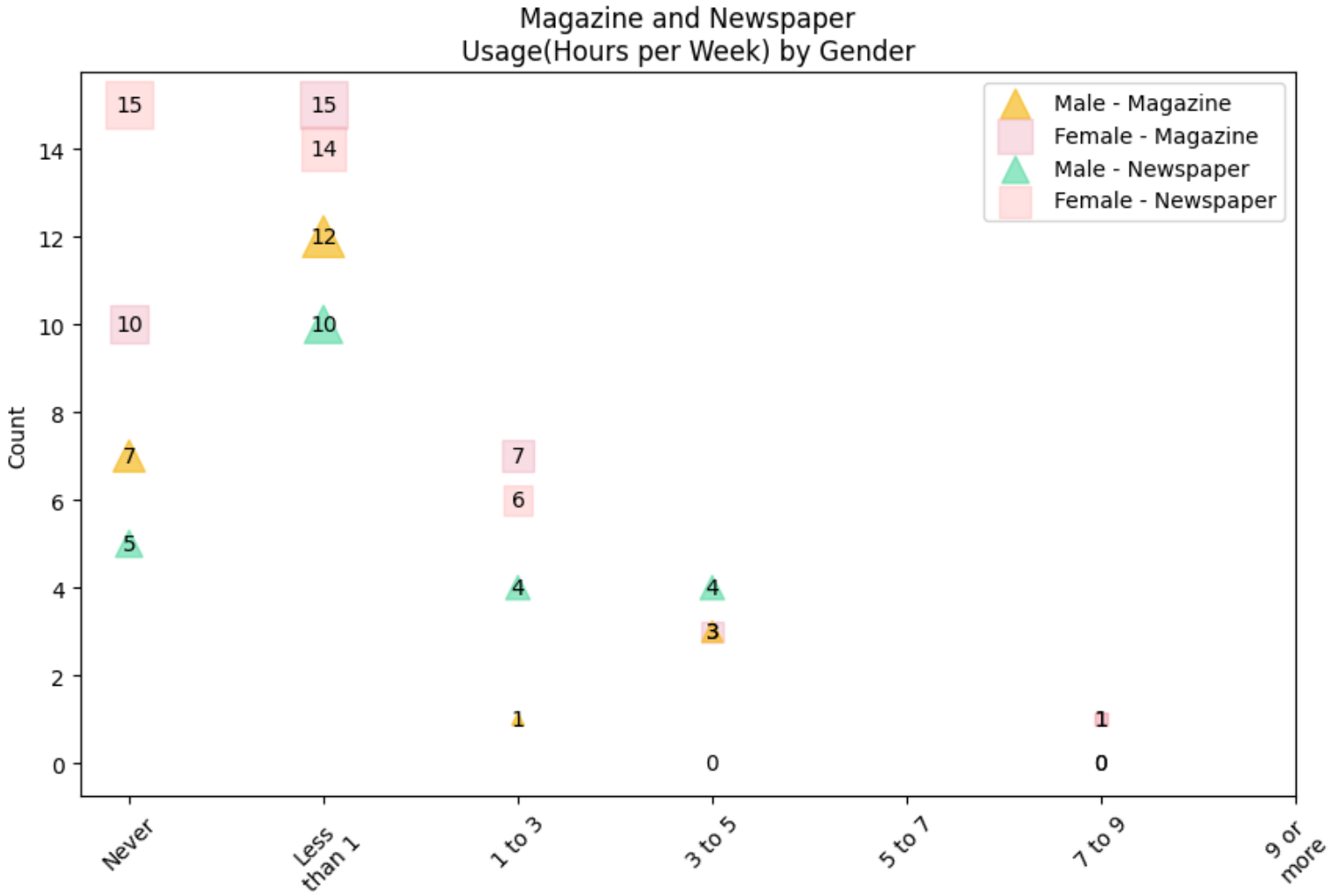
Magazine and Newspaper consumption is notably low, with a substantial number of users spending **less than 1 hour per week** or showing no engagement.

Modest Interest Segment:

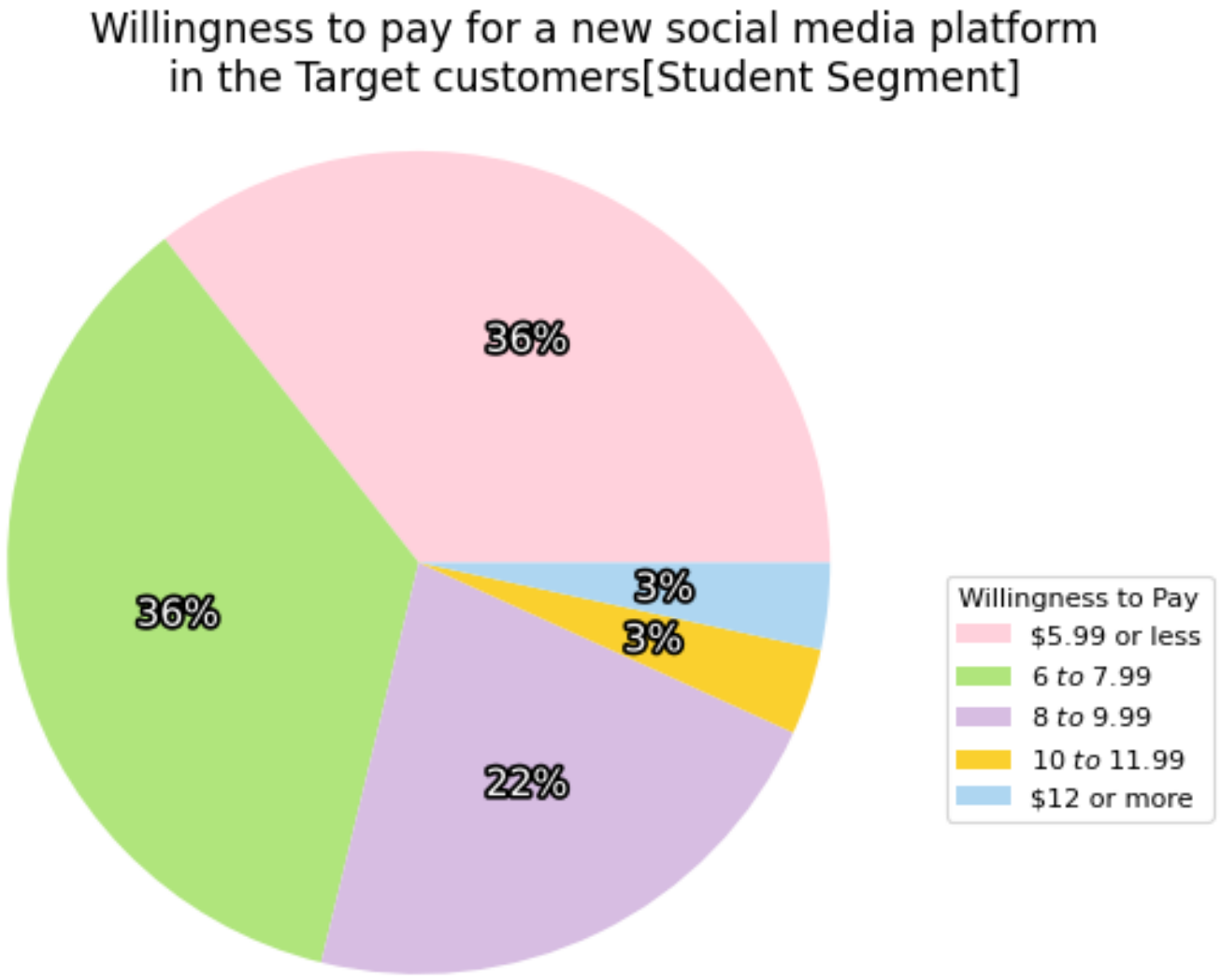
Few users exhibit moderate interest, spending 1-3 hours or 3-5 hours per week on Magazine and Newspaper.

Digital Transition:

Considering the limited interest, explore **digital alternatives** or collaborations with online news platforms to revitalize print media consumption.



5.M5) Demographic and other user-characteristic profile comparison of a) most viable, b) unsure, and c) most disinterested users.



Pricing Sweet Spot Identified:

The pie chart highlights a significant user preference, with 58% comfortable paying between $6 and $9.99, establishing a distinct "Pricing Sweet Spot" for the student segment.

Budget-Conscious Majority:

A noteworthy 36% of students are comfortable with a subscription cost of $5.99 or less, indicating a sizable budget-conscious user base.

Strategic Pricing Blueprint

Optimize Pricing Strategy for Maximal Engagement: Implement competitive tiered pricing ($5.99 to $9.99) for broad user appeal, introduce a budget-friendly entry point, and selectively market premium features for enhanced user engagement.

5.M5) Demographic and other user-characteristic profile comparison of a) most viable, b) unsure, and c) most disinterested users.

Privacy Dominance:

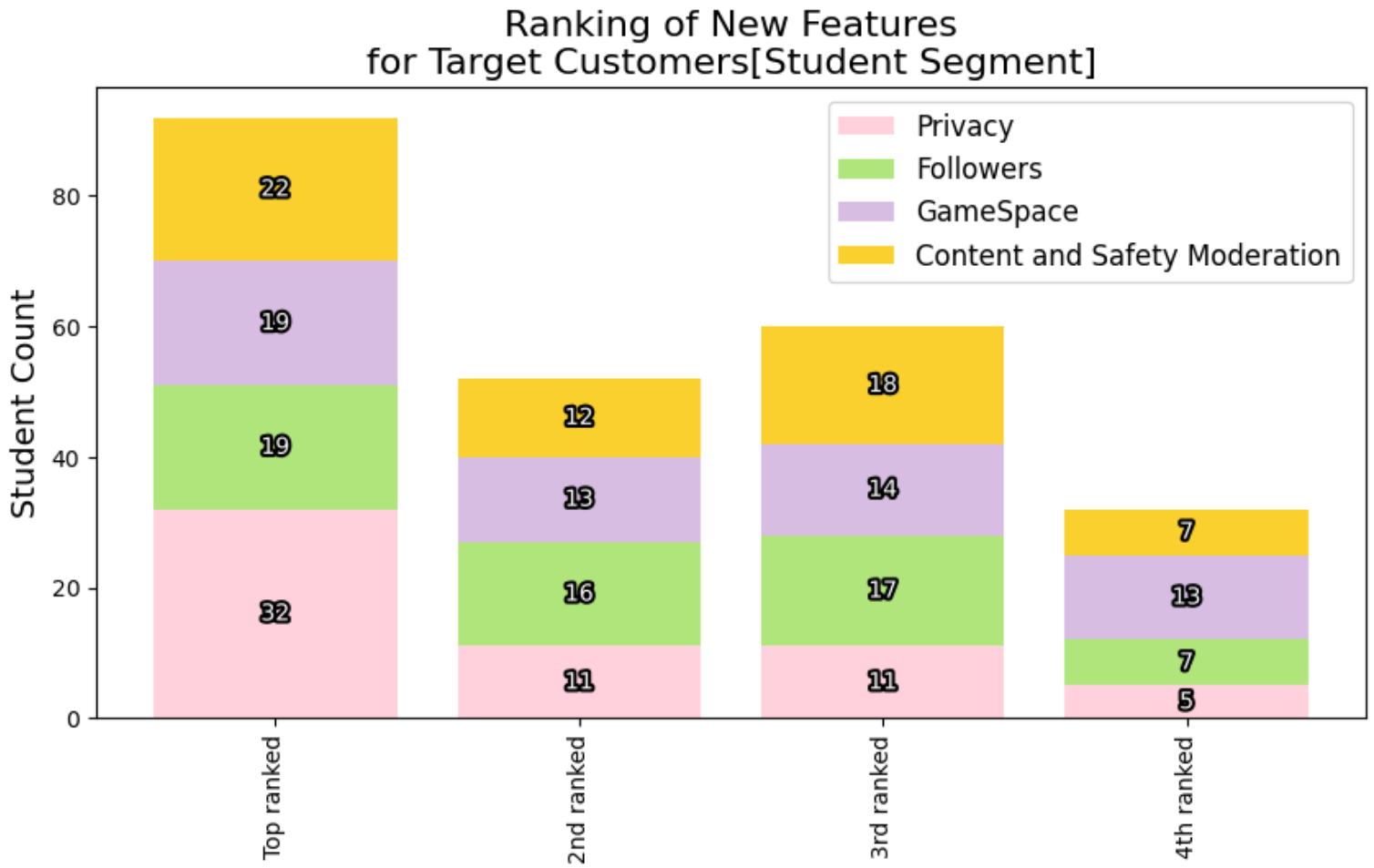
The Stacked Bar Chart reveals that "**Privacy**" is the top preference for **57%** of student users, emphasizing the paramount importance of **data security**.

Neck-and-Neck Preferences:

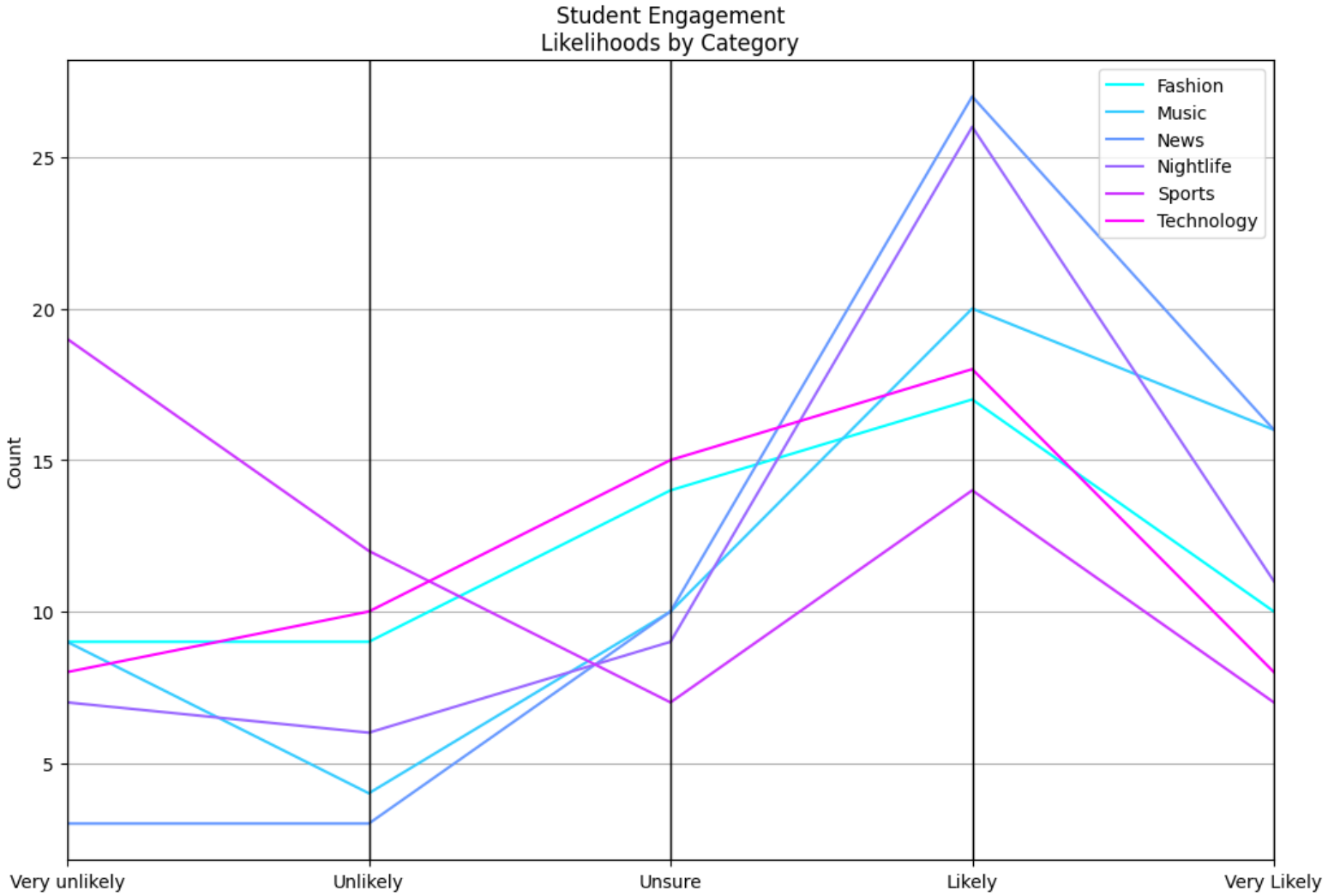
"**Followers**" and "**GameSpace**" compete closely for second place (**21%** each), indicating a significant user segment valuing both social interaction and gaming experiences.

Strategic Enhancement for Optimal User Experience:

Prioritize data security, maintain feature balance, and improve content moderation for a safer and more inclusive "**Better By Far**" platform.



6. M6) Revelatory relationships between user characteristics and intended behaviors.



Target Customer Engagement:

- **Fashion** and **Music** Categories hold broad appeal, with **60%** engagement. Optimize content for a larger market share through personalized strategies.

- **News** and **Technology** are highly attractive, with over **70%** interest. Allocate resources for content creation, aiming for **75%** engagement in these categories.

Strategic Engagement for Growth: Expand the customer database by **20%**, align content with **75%** focus on key categories, and enhance engagement by **25%** in "**Sports**" and "**Nightlife**" for a tailored platform experience.

6. M6) Revelatory relationships between user characteristics and intended behaviors.

Weekend Hurdles:

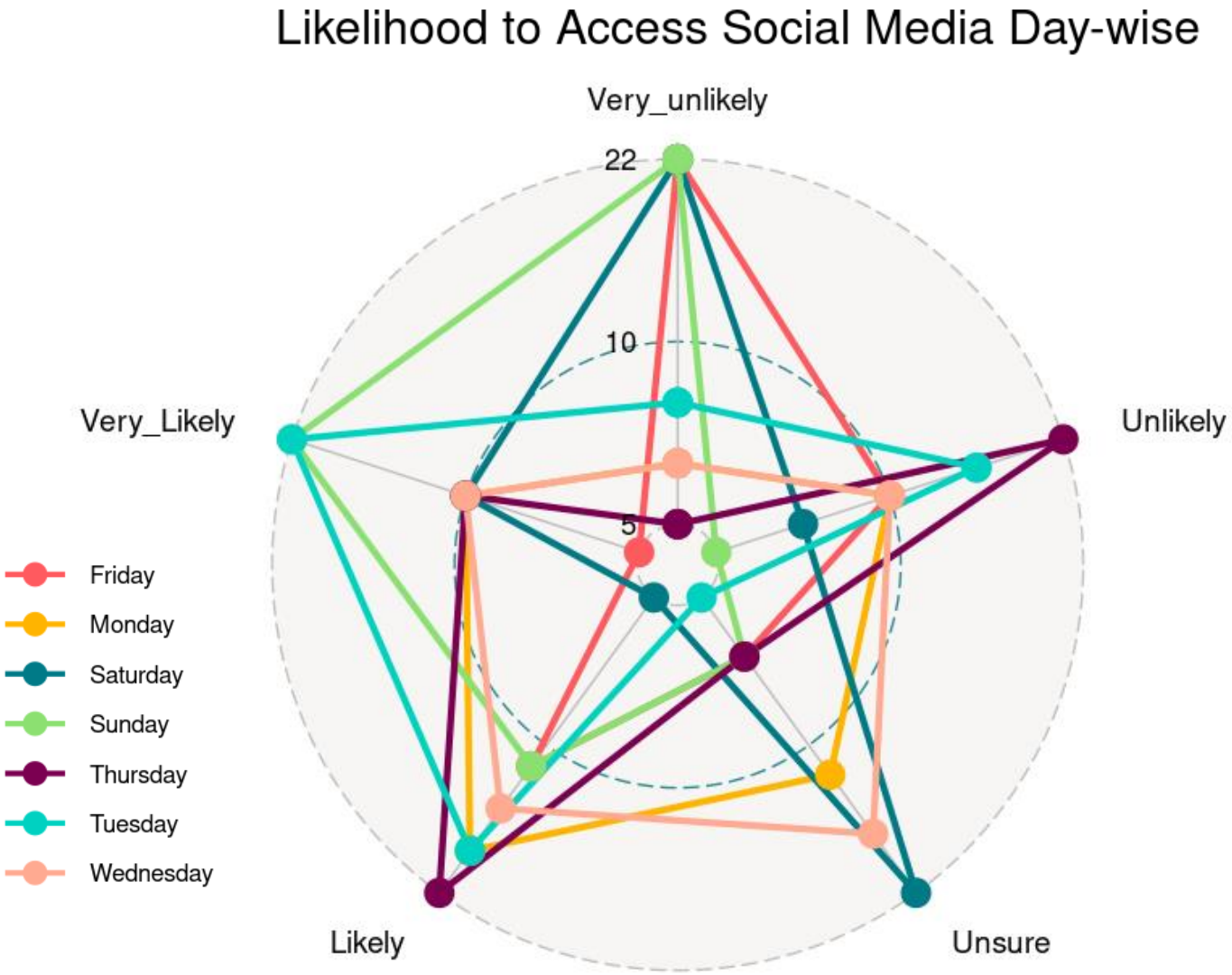
Fridays and **Sundays** pose significant challenges, with around **17%** and **18%** of customers unlikely to engage, necessitating tailored content and promotions.

Steady Weekdays:

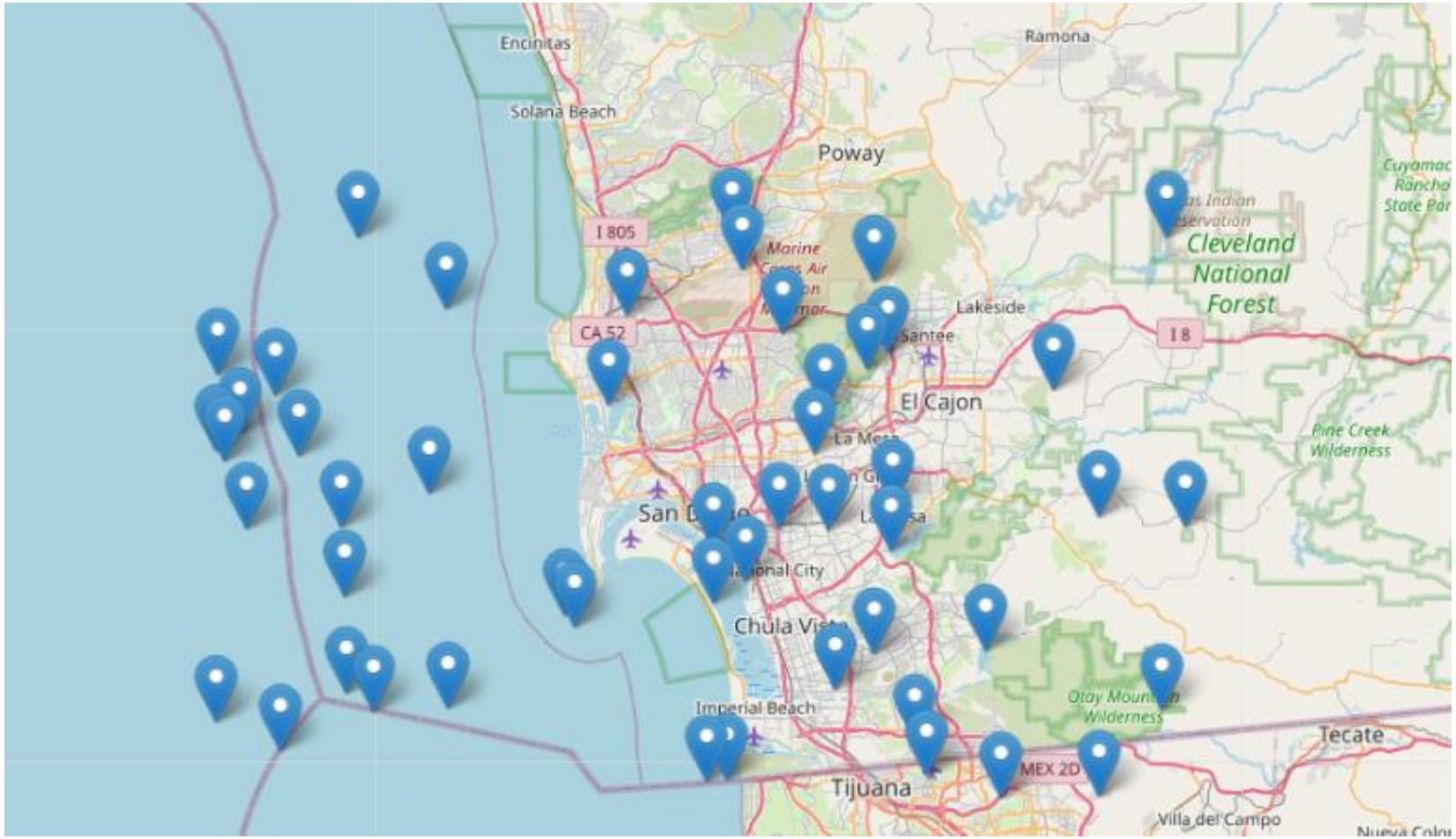
Weekdays (**Monday to Thursday**) showcase stability, with approximately **35%** of users likely to engage, forming a foundation for consistent promotional efforts.

Weekend Revamp Strategy:

Refine **Friday** and **Sunday** content strategies, implement weekend-specific campaigns, and target a **20%** increase in engagement for enhanced user participation.



7. M7) Possible data insights from geographical representations.



We find the sample population for the Target customers are mostly located in and around in **San Diego.**

8. Summary of Data Insights most effective for decision making

We deduced the following insights from the given data:

- **Diverse Engagement:** Fashion, Music, News, and Tech categories attract **60-70% engagement**.
- **Engagement Challenges: Friday** and **Sunday** face **17-18%** disengagement, and **Saturdays** and **Wednesdays** have **21-22% uncertainty**.
- **Demographic Diversity:** User base is multicultural with **privacy** as a top choice, **57%**.

By aligning strategies with these insights, we recommend, Better By Far can enhance and optimize platform performance for sustained success using the following:

- **Optimize Content:** Focus on Fashion, Music, News, and Tech, targeting **75% content alignment**.
- **Address Engagement Challenges:** Refine Friday and Sunday strategies, aim for **20% improvement**, maintain weekday consistency, and **reduce Saturday** and **Wednesday** uncertainty by **15%**.
- **Demographic Targeting:** Expand customer database by **20%**, invest significantly in key categories, and develop strategies for the **40%** uncertain in "**Sports**" and "**Nightlife**," targeting a **25%** improvement in **engagement**.

End of the part of the Use Case Report wherein you are "speaking to the client".

Next comes the part of the Use Case Report wherein you are "speaking among us as a group of programmers".

1. Discuss and showcase the Python code required for using the JupyterLite web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
import piplite # When using the JupyterLite web-based environment, this is needed for installing specific session-based libraries inside of a "wrapper" calle
await piplite.install("seaborn")# An "await" statement is used to install the "seaborn" library within a piplite wrapper,
await piplite.install("openpyxl") # An "await" statement is used for installing (when/if later called on) a library for working with xlsx files called "openp
await piplite.install("folium") # An "await" statement is used for installing (when/if later called on) a library for visualizing geospatial data called "fol
```

```
# Import required libraries
import pandas as pd
# Importing the pandas library and aliasing it as "pd" for convenient use.
# Pandas is commonly used for data processing and manipulation.

import numpy as np
# Importing the NumPy library and aliasing it as "np" for convenient use.
# NumPy is widely used for numerical operations on arrays and matrices.

import matplotlib.pyplot as plt
# Importing the pyplot module from the Matplotlib library and aliasing it as "plt" for convenient use.
# Matplotlib is a popular library for creating static, animated, and interactive visualizations in Python.

import seaborn as sns
# Importing the seaborn library and aliasing it as "sns" for enhanced data visualization and plotting.
# Seaborn is built on top of Matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics.

import pandas.testing as tm
# Importing the "testing" module from the pandas library and aliasing it as "tm".
# This is used for comparing groups, specifically for testing that dataframes match appropriately.

import matplotlib.patheffects as path_effects
# Importing the "patheffects" module from the Matplotlib library.
# Path effects are used to apply various visual effects to the text in Matplotlib plots.

import folium # An "import" statement is used for importing a library for visualizing geospatial data called "folium".

from pandas.plotting import parallel_coordinates
# Importing the "parallel_coordinates" module from the "plotting" submodule of the pandas library.
# This module is used for creating parallel coordinates plots.

from matplotlib import patheffects
# Importing the "patheffects" module from the Matplotlib library.
# This is an alternative import style for the "patheffects" module.

from matplotlib.colors import LinearSegmentedColormap
# Importing the "LinearSegmentedColormap" class from the "colors" submodule of the Matplotlib library.
```

1. Discuss and showcase the Python code required for using the JupyterLite web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
# Load the Excel files into separate data frames
# The read_excel() function is used to read data from Excel files.

# Load the first Excel file "M1Initial dataset.xlsx" into a data frame called df1
df1 = pd.read_excel("M1Initial dataset.xlsx", sheet_name=0)

# Load the second Excel file "M2Change Over Time.xlsx" into a data frame called df2
df2 = pd.read_excel("M2Change Over Time.xlsx", sheet_name=0)

# Load the third Excel file "M3Data Composition.xlsx" into a data frame called df3
df3 = pd.read_excel("M3Data Composition.xlsx", sheet_name=0)

# Load the fourth Excel file "M4Data Distributions.xlsx" into a data frame called df4
df4 = pd.read_excel("M4Data Distributions.xlsx", sheet_name=0)

# Load the fifth Excel file "M5Group Comparisons.xlsx" into a data frame called df5
df5 = pd.read_excel("M5Group Comparisons.xlsx", sheet_name=0)

# Load the sixth Excel file "M6Relationships.xlsx" into a data frame called df6
df6 = pd.read_excel("M6Relationships.xlsx", sheet_name=0)

# Load the seventh Excel file "M7Geographical Data.xlsx" into a data frame called df7
df7 = pd.read_excel("M7Geographical Data.xlsx", sheet_name=0)
```

```
# Establishing connections between multiple data frames (M1, M2, M3, M4, M5, M6, and M7) using the "merge" function.

# Merge data frames M1 and M2 based on the common "ID" column, resulting in a new data frame df12.
df12 = pd.merge(df1, df2, on="ID")

# Merge data frames M3 and M4 based on the common "ID" column, resulting in a new data frame df34.
df34 = pd.merge(df3, df4, on="ID")

# Merge data frames M5 and M6 based on the common "ID" column, resulting in a new data frame df56.
df56 = pd.merge(df5, df6, on="ID")

# Merge the data frames df12 and df34 based on the common "ID" column, resulting in a new data frame df1234.
df1234 = pd.merge(df12, df34, on="ID")

# Merge the resulting data frame df1234 with data frame df56 based on the common "ID" column, resulting in the final merged data frame df.
df = pd.merge(df1234, df56, on="ID")

# Display the columns in the merged data frame.
df.columns
```

1. Discuss and showcase the Python code required for using the JupyterLite web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
# Create a separate data frame specifically for students by filtering rows where the 'Q2' column contains "Student," "student," or "STUDENT."
df_student = df[df['Q2'].isin(["Student", "student", "STUDENT"])]

# Display the resulting data frame containing only student data.
df_student
```

```
# Create a copy of the student data frame to preserve the original data.
df_student_new = df_student.copy()

# Standardize values in the 'Q2' column by converting them to lowercase and replacing variations with "student."
df_student_new['Q2'] = df_student_new['Q2'].str.lower().replace(["student", "student1", "student2"], "student")

# Display the updated student data frame with standardized values in the 'Q2' column.
df_student_new
```

```
# Use the rename() function to rename variables of interest in the student data frame for better readability and data visualization.
df_student_new.rename(columns={
    "Q1": "Number of people living in household",
    "Q2": "Occupation",
    "Q3": "How many hours per week do you work?",
    "Q4": "Gender",
    "Q5": "Age",
    "Q6": "Ethnicity",
    "Q7": "Education Level",
    "Q8": "Annual Income",
    "Q9": "Hours away from social media platform?",
    "Q10": "New online social media platform?",
    "Q11": "New online social media platform - beneficial to me",
    "Q12": "Interested in New Social Media Platform",
    "Q13": "Internet usage per day",
    "Q14": "Television usage per day",
    "Q15": "Radio usage per day",
    "Q16": "Magazines read per week",
    "Q17": "Newspapers read per week",
    "Q18": "Willingness to pay",
    "Q19": "Privacy controls",
    "Q20": "Quickly gain followers",
    "Q21": "Multi-player real-time game spaces",
    "Q22": "Sophisticated content and safety moderation",
    "Q23": "Sports",
    "Q24": "Movies",
    "Q25": "News",
    "Q26": "Fashion",
    "Q27": "Nightlife",
    "Q28": "Technology",
    "Q29": "Music",
    "Q30": "Safe & Friendly environment",
```

1. Discuss and showcase the Python code required for using the JupyterLite web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
    "Q31": "Outgoing",
    "Q32": "Sunday",
    "Q33": "Monday",
    "Q34": "Tuesday",
    "Q35": "Wednesday",
    "Q36": "Thursday",
    "Q37": "Friday",
    "Q38": "Saturday",
    "Q39": "Livestream",
    "Q40": "Games",
    "Q41": "Monthly prize drawings",
    "Q42": "Regular celebrity features",
    "Q43": "Live virtual parties",
    "Q44": "Music streams",
    "Q45": "Newsletter email",
    "Q46": "Early Morning 4:01 am to 5:00 am",
    "Q47": "Morning 5:01 am to 11:00 am",
    "Q48": "Midday 11:01 am to 6:00 pm",
    "Q49": "Evening 6:01 pm to 9:00 pm",
    "Q50": "Night 9:01 pm to 11:59 pm",
    "Q51": "Late Night 12:00 am to 4:00 am",
    "Q52": "Talkative"
}, inplace=True)

# Display the updated column names.
print(df_student_new.columns)
```

```
#=======================SECTION SEPARATOR
```

```
# Create a copy of the student data frame for further processing
df_student_new1 = df_student_new.copy()

# Sort the data by date using the sort_values() function and the "Date" column
df_student_new1 = df_student_new1.sort_values("Date")

# Display the resulting data frame with sorted values
df_student_new1
```

1. Discuss and showcase the Python code required for using the JupyterLite web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
# Define a dictionary mapping numerical codes to corresponding annual income categories
Annual_Income = {
    1: '$19,999\nor Less',
    2: '$20,000\nto $39,999',
    3: '$40,000\nto $59,999',
    4: '$60,000\nto $79,999',
    5: '$80,000\nto $99,999',
    6: '$100,000\nor more',
    7: 'Prefer not\nto disclose'
}

# Extract data from the DataFrame for x (date) and y (annual income) variables for plotting
x = df_student_new1["Date"]
y = df_student_new1["Annual Income"]

# Choose a custom color for the line plot
custom_color = ['#66DDAB']  # You can choose any color from the list

# Create a line plot with the chosen custom color
plt.plot(x, y, color=custom_color[0])

# Customize y-axis labels using the defined Annual_Income dictionary
plt.yticks(list(Annual_Income.keys()), list(Annual_Income.values()))

# Set the title and y-axis label for the plot
plt.title("Change in Annual Income($) Over Time")
plt.ylabel("Annual Income of Target Customers")

# Save the line plot as an image file
plt.savefig("TCP-Better_By_Far-PythonDV1.png", bbox_inches='tight')

# Display the plot
plt.show()
```

2. Discuss and showcase the R code required for using the Posit Cloud RStudio Cloud web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
# Install packages if required
# Check if the "tidyverse" package is already installed, If not installed, install the "tidyverse" package
if(!require("tidyverse")) install.packages("tidyverse")
# Check if the "fs" package is already installed, If not installed, install the "fs" package
if(!require("fs")) install.packages("fs")
# Check if the "scales" package is already installed, if not installed, install the "scales" package
if(!require("scales")) install.packages("scales")
# Check if the "dplyr" package is already installed, If not installed, install the "dplyr" package
if(!require("dplyr")) install.packages("dplyr")
# Check if the "ggplot2" package is already installed, If not installed, install the "ggplot2" package
if (!require("ggplot2"))install.packages("ggplot2")

# Requires libraries
install.packages("ggiraphExtra")
# This line installs the "ggiraphExtra" package using the install.packages() function.
# The package provides extensions to ggplot2 for creating interactive and animated graphics using HTML and JavaScript.
install.packages("ggradar")
# This line installs the "ggradar" package using the install.packages() function.
# The package likely provides functionality for creating radar (spider) charts using ggplot2, allowing users to visualize multivariate data on a two-dimensional plot.
library(ggiraphExtra)
# This line loads the "ggiraphExtra" library, which provides extensions to ggplot2 for creating interactive and animated graphics using HTML and JavaScript.
# It allows for the creation of graphs that can be embedded in web pages and support interactive features.
library(ggradar)
# This line loads the "ggradar" library, which likely provides functions for creating radar (spider) charts using ggplot2.
# Radar charts are useful for visualizing multivariate data on a two-dimensional plot, with each variable represented as a spoke.
library(ggplot2)
# This line loads the "ggplot2" library, a powerful and popular data visualization package in R.
# ggplot2 is known for its declarative syntax and ability to create a wide range of static and dynamic plots.
library(tidyverse)
# This line loads the "tidyverse" library, which is a collection of packages for data manipulation and visualization.
# It includes packages like dplyr, ggplot2, tidyr, and others, providing a consistent and integrated set of tools for data analysis.
library(fs)
# This line loads the "fs" library, which provides a set of file system functions for working with files and directories in R.
# It includes functions for file manipulation, path construction, and directory navigation.
library(scales)
# This line loads the "scales" library, which is often used in conjunction with ggplot2 to customize axis scales and breaks in plots.
library(readxl)
# This line loads the "readxl" library, which is used for reading Excel files into R.
# It provides functions to import data from Excel worksheets into data frames.
library(dplyr)
# This line loads the "dplyr" library, a key package in the tidyverse ecosystem, providing a set of functions for data manipulation.
# It includes functions like filter, mutate, and summarize for efficient and expressive data wrangling.
library(tidyr)
# This line loads the "tidyr" library, which is part of the tidyverse and focuses on data tidying and reshaping.
# It includes functions like gather and spread for converting data between wide and long formats.
```

2. Discuss and showcase the R code required for using the Posit Cloud RStudio Cloud web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
# Use the read_excel() function to load the excel dataset file that has the filename and assign them to corresponding variables.
data0 <- read_excel("M1Initial dataset.xlsx", 1)
data1 <- read_excel("M2Change Over Time.xlsx", 1)
data2 <- read_excel("M3Data Composition.xlsx", 1)
data3 <- read_excel("M4Data Distributions.xlsx", 1)
data4 <- read_excel("M5Group Comparisons.xlsx", 1)
data5 <- read_excel("M6Relationships.xlsx", 1)

# merge all the files by ID into a single dataframe called "Merge"
Merge <- merge(data0, data1, by = "ID")
Merge <- merge(Merge, data2, by = "ID")
Merge <- merge(Merge, data3, by = "ID")
Merge <- merge(Merge, data4, by = "ID")
Merge <- merge(Merge, data5, by = "ID")

#=======================SECTION SEPARATOR

#Replacing multiple variations of "student" with "student" in the Q2 column
Merge$Q2 <- ifelse(tolower(Merge$Q2) %in% c("student", "Student", "STUDENT"), "student", Merge$Q2)

# Create a separate data frame for students
df_student = Merge %>% filter(Q2=="student")
# Display the resulting data frame containing student data
head(df_student)

# Create a copy of the student data frame and standardize values in the 'Q2' column
df_student_new = df_student

#=======================SECTION SEPARATOR

# Create a named vector to map old column names to new column names.
column_mapping <- c(
  "Q1" = "Number of people living in household",
  "Q2" = "Occupation",
  "Q3" = "How many hours per week do you work?",
  "Q4" = "Gender",
  "Q5" = "Age",
  "Q23" = "Sports",
  "Q24" = "Movies",
  "Q25" = "News",
  "Q26" = "Fashion",
  "Q27"= "Nightlife",
  "Q28" = "Technology",
  "Q29" = "Music",
  "Q32" = "Sunday",
  "Q33" = "Monday",
  "Q34" = "Tuesday",
```

2. Discuss and showcase the R code required for using the Posit Cloud RStudio Cloud web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
  "Q32" = "Sunday",
  "Q33" = "Monday",
  "Q34" = "Tuesday",
  "Q35" = "Wednesday",
  "Q36" = "Thursday",
  "Q37" = "Friday",
  "Q38" = "Saturday"
)

# Extract the current column names of the data frame.
current_column_names <- names(df_student_new)

# Identify the old column names that are present in the data frame.
existing_old_column_names <- intersect(names(column_mapping), current_column_names)

# Rename the columns using the names() function for only the existing old column names.
for (old_col_name in existing_old_column_names) {
  new_col_name <- column_mapping[old_col_name]
  names(df_student_new)[current_column_names == old_col_name] <- new_col_name
}

# To print the updated column names
names(df_student_new)

#======================SECTION SEPARATOR


# Working on a Radar chart
# Make a new data frame with only selected variables of interest from the dataset.
df_student_new_access = df_student_new %>%
                                select(Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday)

# View how the first five rows of the dataset looks like
head(df_student_new_access)

# Define the columns of interest
columns_of_interest <- c('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',"Saturday")

# Create an empty data frame to store the results
df_student_new_access_engagement <- data.frame()

# Iterate through the columns of interest
for (column in columns_of_interest) {
  # Iterate through the likelihood categories
  for (likelihood in c(-2, -1, 0, 1, 2)) {
    # Filter the data frame based on the category and likelihood
    filtered_data <- df_student_new_access %>%
      filter(.data[[column]] == likelihood)
```

2. Discuss and showcase the R code required for using the Posit Cloud RStudio Cloud web-based environment to use your raw Use Case datasets to create a DV that shows how target customers for your use case vary with regard to their reported interest in or intentions for the client's offering based on some demographic variable.

```
    # Calculate the count
    count <- nrow(filtered_data)

    # Create a data frame with the results
    result <- data.frame(Category = column, Likelihood = likelihood, Count = count)

    # Append the result to the counts data frame
    df_student_new_access_engagement <- rbind(df_student_new_access_engagement, result)
  }
}

# Print the resulting data frame
print(df_student_new_access_engagement)

#=========================SECTION SEPARATOR

# Pivot the data frame
pivoted_df1 <- df_student_new_access_engagement %>%
  pivot_wider(names_from = Likelihood, values_from = Count, values_fill = 0)

#  Rename columns
colnames(pivoted_df1)[colnames(pivoted_df1) == '-2'] <- 'Very_unlikely'
colnames(pivoted_df1)[colnames(pivoted_df1) == '-1'] <- 'Unlikely'
colnames(pivoted_df1)[colnames(pivoted_df1) == '0'] <- 'Unsure'
colnames(pivoted_df1)[colnames(pivoted_df1) == '1'] <- 'Likely'
colnames(pivoted_df1)[colnames(pivoted_df1) == '2'] <- 'Very_Likely'

# Display the reshaped DataFrame
print(pivoted_df1)

#=========================SECTION SEPARATOR

# Radar plot
pivoted_df1 %>%
  mutate_each(funs(rescale), -Category) %>%
  ggradar(values.radar = c("5","10","22"))+
  ggtitle("Likelihood to Access Social Media Day-wise")
# Save the plot to a file
ggsave("TCP-Better_By_Far-PythonDV10.png")

#=========================SECTION SEPARATOR
```

End of file.