Fitness Consumer Analysis

Objectives

To evaluate your understanding and ability to apply data analysis and visualization techniques to a real-world dataset. This assessment will test your proficiency in handling data, using various libraries, applying statistical methods, and presenting insights through visualizations.

Task Description

You are provided with a dataset containing user interactions with a digital platform over the last year. The dataset `(Fitness Consumer Analysis.csv)` includes various details.

Project Overview

The dataset contains attributes such as:

- `Timestamp` refers to the date and time when the survey response was submitted. This attribute is crucial for tracking the recency and seasonality of the responses, allowing analysts to understand trends over time. It ensures that the data is relevant and up-to-date, helping to capture the dynamics of fitness wearable usage and its impact over different periods.
- `What is your age?` captures the respondent's age. This attribute is essential for demographic analysis, revealing age-related trends and preferences in fitness behavior. By segmenting data based on age groups, we can make fitness recommendations and understand how different age cohorts engage with fitness wearables.
- `What is your gender?` identifies the respondent's gender. This attribute allows for gender-based analysis of fitness habits and the impact of wearables. Understanding gender-specific trends helps in addressing the unique needs of different genders and can guide the development of more inclusive fitness products and marketing strategies.
- `What is your highest level of education?` indicates the respondent's highest educational attainment. This attribute provides insight into the correlation between education levels and fitness behavior. It helps in understanding how education influences fitness awareness, wearable usage, and the effectiveness of fitness interventions.
- `What is your current occupation?` describes the respondent's current job or professional status. This attribute helps in analyzing how occupation impacts fitness habits and wearable usage. Identifying trends among different occupational groups can assist in customizing fitness solutions to fit various work-life dynamics.
- `How often do you exercise in a week?` quantifies the frequency of the respondent's weekly exercise routine. This attribute is a direct indicator of the respondent's fitness activity level. It is essential for evaluating the overall fitness engagement of the respondents and understanding how often they incorporate physical activity into their routine.
- `How long have you been using a fitness wearable?` asks about the duration for which the respondent has been using a fitness wearable. This attribute helps in understanding long-term engagement and the adoption patterns of fitness wearables. It is important for assessing the sustained impact of wearables over time.
- `How frequently do you use your fitness wearable?` measures the regularity with which the respondent uses their fitness wearable. This attribute indicates the level of dependency and routine integration of the wearable. It is useful for measuring user engagement and consistency in using the wearable's features.
- `How often do you track fitness data using wearable?` examines the frequency of tracking fitness data through the wearable. This attribute reflects the respondent's commitment to monitoring their fitness progress. It is crucial for understanding how actively users utilize the data tracking features of wearables.
- `How has the fitness wearable impacted your fitness routine?` seeks to understand the perceived effect of the fitness wearable on the respondent's exercise habits. This attribute provides qualitative insights into behavior changes prompted by the wearable, indicating its effectiveness in altering fitness routines.
- `Has the fitness wearable helped you stay motivated to exercise?` explores whether the wearable has positively influenced the respondent's motivation to exercise. This attribute is key for assessing the psychological benefits of fitness wearables and their role in encouraging consistent physical activity.

- `Do you think that the fitness wearable has made exercising more enjoyable?` asks if the respondent perceives the wearable as enhancing the enjoyment of exercising. This attribute reflects the wearable's role in improving the exercise experience, which is important for understanding user satisfaction.
- `How engaged do you feel with your fitness wearable?` gauges the level of engagement the respondent feels with their fitness wearable. This attribute measures the depth of user interaction and involvement with the device, which is useful for understanding overall user engagement.
- `Does using a fitness wearable make you feel more connected to the fitness community?` inquires whether the wearable helps the respondent feel a sense of belonging to the fitness community. This attribute indicates the social impact of using fitness wearables and their role in fostering community connections.
- `How has the fitness wearable helped you achieve your fitness goals?` examines the extent to which the wearable has assisted the respondent in reaching their fitness objectives. This attribute provides direct feedback on the effectiveness of the wearable in goal attainment, essential for evaluating its practical benefits.
- `How has the fitness wearable impacted your overall health?` looks at the perceived impact of the wearable on the respondent's general health. This attribute reflects broader health benefits beyond just fitness, which is important for understanding the holistic health effects of fitness wearables.
- `Has the fitness wearable improved your sleep patterns?` assesses whether the wearable has had a positive effect on the respondent's sleep. This attribute indicates the impact of wearables on sleep quality and habits, useful for evaluating their role in promoting better sleep health.
- `Do you feel that the fitness wearable has improved your overall well-being?` asks for the respondent's perception of the wearable's effect on their overall well-being. This attribute provides a comprehensive view of the wearable's impact on quality of life, important for understanding broader implications.
- `Has using a fitness wearable influenced your decision to exercise more?` investigates whether the wearable has motivated the respondent to increase their exercise frequency. This attribute measures the wearable's influence on exercise habits, crucial for evaluating behavioral changes prompted by the device.
- `Has using a fitness wearable influenced your decision to purchase other fitness-related products?` explores whether the wearable has led the respondent to buy additional fitness products. This attribute indicates the wearable's impact on consumer behavior and spending, useful for understanding its market influence.
- `Has using a fitness wearable influenced your decision to join a gym or fitness class?` inquires whether the wearable has influenced the respondent to join fitness facilities or classes. This attribute reflects the wearable's effect on social and group fitness activities, important for understanding its role in fitness engagement.
- `Has using a fitness wearable influenced your decision to change your diet?` asks whether the wearable has prompted dietary changes in the respondent. This attribute indicates the wearable's impact on nutritional habits and health choices, crucial for evaluating its influence on health and lifestyle changes.

Libraries and Data Handling

Libraries Used

- `numpy as np` - Used for numerical operations.
- `pandas as pd` - Used for data manipulation.
- `matplotlib.pyplot as plt` - Used for plotting.
- `seaborn as sns` - Used for advanced plotting.

Data Handling Functions Used

- `pd.read_csv()` - Load the dataset into the environment for analysis.
- `df.shape` - Returns the dimensions of the DataFrame to understand the size of the dataset. In this analysis, it returns, (30, 22) which meant that the dataset contains 30 rows and 22 columns.
- `df.head()` - Displays the first five rows of the DataFrame for a quick preview.
- `df.info()` - Provides a concise summary of the DataFrame by giving the overview of data types and non-null counts. In this data, there are 30 non-null counts out of 30 rows of the DataFrame. It means that all entries contain valid (non-missing) data.
- `df.column` - Lists all column names.
- `df.dtypes` - Lists the data types of each column. All data types as of now are object.
- `df.isnull().sum()` - Summarizes the number of missing values per column but since there are no missing values, there's no need for data cleaning and imputation in this case.
- `df.nunique()` - Returns the number of unique values in each column to understand the diversity of the data in each column.
- `df.set_index()` - Sets a column as the index of the DataFrame. Since the 'Timestamp' column is not necessary for the analysis, I set it as an index to manage the data better.
- `df.rename(columns={}, inplace=True)` - Renames columns to shorter, more manageable names. The column names are long questions, making them cumbersome. Renaming them to shorter names improves readability.

In [ ]:

```python
# Importing necessary libraries for numerical operations, data manipulation, and visualization
import numpy as np  # For numerical operations
import pandas as pd  # For data manipulation and analysis
import matplotlib.pyplot as plt  # For plotting and visualization
import seaborn as sns  # For advanced visualization
import statsmodels.api as sm  # For statistical modeling
import pprint  # For pretty-printing data structures

# Importing specific statistical functions from scipy
from scipy import stats  # For various statistical functions
from scipy.stats import chi2_contingency  # For chi-square test
from scipy.stats import f_oneway  # For one-way ANOVA test
from statsmodels.formula.api import ols  # For ordinary least squares regression

# Importing preprocessing tools from sklearn
from sklearn.preprocessing import LabelEncoder, OneHotEncoder  # For encoding categorical features
from sklearn.preprocessing import StandardScaler  # For feature scaling

# Importing tools for model selection and evaluation
from sklearn.model_selection import train_test_split  # For splitting data into training and test sets
from sklearn.linear_model import LogisticRegression  # For logistic regression model
from sklearn.naive_bayes import GaussianNB  # For Naive Bayes classifier
from sklearn.neighbors import KNeighborsClassifier  # For K-Nearest Neighbors classifier
from sklearn.tree import DecisionTreeClassifier  # For Decision Tree classifier
from sklearn.svm import SVC  # For Support Vector Classifier
from sklearn.ensemble import RandomForestClassifier  # For Random Forest classifier
from sklearn.ensemble import GradientBoostingClassifier  # For Gradient Boosting classifier
from sklearn.inspection import permutation_importance  # For computing permutation importance
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score  # For evaluation metrics
```

In [ ]:

```python
# Reading the dataset from a CSV file into a pandas DataFrame
df = pd.read_csv("Fitness Consumer Analysis.csv")
```

In [ ]:

```python
# Displaying the shape of the DataFrame (number of rows and columns)
df.shape
```

Out[ ]:

```
(30, 22)
```

In [ ]:
```
# Displaying the first five rows of the DataFrame to get an initial look at the data
df.head()
```
Out[ ]:

	Timestamp	What is your age?	What is your gender?	What is your highest level of education?	What is your current occupation?	How often do you exercise in a week?	How long have you been using a fitness wearable?	How frequently do you use your fitness wearable?
0	2023/03/30 9:43:19 PM GMT+5:30	18-24	Male	Some college or associate degree	Student	5 or more times a week	Less than 6 months	Daily
1	2023/03/31 5:07:46 PM GMT+5:30	Under 18	Male	Bachelor's degree	Student	5 or more times a week	Less than 6 months	3-4 times a week
2	2023/03/31 7:44:46 PM GMT+5:30	18-24	Female	Bachelor's degree	Student	Less than once a week	Less than 6 months	Rarely
3	2023/03/31 9:36:07 PM GMT+5:30	25-34	Female	Some college or associate degree	Employed part-time	3-4 times a week	6-12 months	3-4 times a week
4	2023/03/31 9:37:32 PM GMT+5:30	18-24	Male	Bachelor's degree	Student	1-2 times a week	Less than 6 months	Daily

5 rows × 22 columns

In [ ]:
```
# Displaying a summary of the DataFrame, including the number of non-null e
ntries and data types
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 22 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Timestamp
30 non-null     object
 1   What is your age?
30 non-null     object
 2   What is your gender?
30 non-null     object
 3   What is your highest level of education?
30 non-null     object
 4   What is your current occupation?
30 non-null     object
 5   How often do you exercise in a week?
30 non-null     object
 6   How long have you been using a fitness wearable?
30 non-null     object
 7   How frequently do you use your fitness wearable?
30 non-null     object
 8   How often do you track fitness data using wearable?
30 non-null     object
 9   How has the fitness wearable impacted your fitness routine?
30 non-null     object
 10  Has the fitness wearable helped you stay motivated to exercise?
30 non-null     object
 11  Do you think that the fitness wearable has made exercising more enjoy
able?                                 30 non-null     object
 12  How engaged do you feel with your fitness wearable?
30 non-null     object
 13  Does using a fitness wearable make you feel more connected to the fit
ness community?                       30 non-null     object
 14  How has the fitness wearable helped you achieve your fitness goals?
30 non-null     object
 15  How has the fitness wearable impacted your overall health?
30 non-null     object
 16  Has the fitness wearable improved your sleep patterns?
30 non-null     object
 17  Do you feel that the fitness wearable has improved your overall well-
being?                                30 non-null     object
 18  Has using a fitness wearable influenced your decision? [To exercise m
ore?]                                 30 non-null     object
 19  Has using a fitness wearable influenced your decision? [To purchase o
ther fitness-related products?]  30 non-null     object
 20  Has using a fitness wearable influenced your decision? [To join a gym
or fitness class?]                  30 non-null     object
 21  Has using a fitness wearable influenced your decision? [To change you
r diet?]                              30 non-null     object
dtypes: object(22)
memory usage: 5.3+ KB
```

In [ ]:
```
# Displaying the column names of the DataFrame
df.columns
```

Out[ ]:
```
Index(['Timestamp', 'What is your age?', 'What is your gender?',
       'What is your highest level of education?',
       'What is your current occupation?',
       'How often do you exercise in a week?',
       'How long have you been using a fitness wearable?',
       'How frequently do you use your fitness wearable?',
       'How often do you track fitness data using wearable?',
       'How has the fitness wearable impacted your fitness routine?',
       'Has the fitness wearable helped you stay motivated to exercise?',
       'Do you think that the fitness wearable has made exercising more en
joyable?',
       'How engaged do you feel with your fitness wearable?',
       'Does using a fitness wearable make you feel more connected to the
fitness community?',
       'How has the fitness wearable helped you achieve your fitness goal
s?',
       'How has the fitness wearable impacted your overall health?',
       'Has the fitness wearable improved your sleep patterns?',
       'Do you feel that the fitness wearable has improved your overall we
ll-being?',
       'Has using a fitness wearable influenced your decision? [To exercis
e more?]',
       'Has using a fitness wearable influenced your decision? [To purchas
e other fitness-related products?]',
       'Has using a fitness wearable influenced your decision? [To join a
gym or fitness class?]',
       'Has using a fitness wearable influenced your decision? [To change
your diet?]'],
      dtype='object')
```

In [ ]:
```
# Displaying the data types of each column in the DataFrame
df.dtypes
```

Out[ ]:
```
Timestamp
object
What is your age?
object
What is your gender?
object
What is your highest level of education?
object
What is your current occupation?
object
How often do you exercise in a week?
object
How long have you been using a fitness wearable?
object
How frequently do you use your fitness wearable?
object
How often do you track fitness data using wearable?
object
How has the fitness wearable impacted your fitness routine?
object
Has the fitness wearable helped you stay motivated to exercise?
object
Do you think that the fitness wearable has made exercising more enjoyable?
object
How engaged do you feel with your fitness wearable?
object
Does using a fitness wearable make you feel more connected to the fitness
community?                              object
How has the fitness wearable helped you achieve your fitness goals?
object
How has the fitness wearable impacted your overall health?
object
Has the fitness wearable improved your sleep patterns?
object
Do you feel that the fitness wearable has improved your overall well-bein
g?                                       object
Has using a fitness wearable influenced your decision? [To exercise more?]
object
Has using a fitness wearable influenced your decision? [To purchase other
fitness-related products?]    object
Has using a fitness wearable influenced your decision? [To join a gym or f
itness class?]                object
Has using a fitness wearable influenced your decision? [To change your die
t?]                           object
dtype: object
```

In [ ]:
```
# Displaying the number of missing values in each column of the DataFrame
df.isnull().sum()
```

Out[ ]:
```
Timestamp
0
What is your age?
0
What is your gender?
0
What is your highest level of education?
0
What is your current occupation?
0
How often do you exercise in a week?
0
How long have you been using a fitness wearable?
0
How frequently do you use your fitness wearable?
0
How often do you track fitness data using wearable?
0
How has the fitness wearable impacted your fitness routine?
0
Has the fitness wearable helped you stay motivated to exercise?
0
Do you think that the fitness wearable has made exercising more enjoyable?
0
How engaged do you feel with your fitness wearable?
0
Does using a fitness wearable make you feel more connected to the fitness
community?                              0
How has the fitness wearable helped you achieve your fitness goals?
0
How has the fitness wearable impacted your overall health?
0
Has the fitness wearable improved your sleep patterns?
0
Do you feel that the fitness wearable has improved your overall well-bein
g?                                       0
Has using a fitness wearable influenced your decision? [To exercise more?]
0
Has using a fitness wearable influenced your decision? [To purchase other
fitness-related products?]    0
Has using a fitness wearable influenced your decision? [To join a gym or f
itness class?]                0
Has using a fitness wearable influenced your decision? [To change your die
t?]                           0
dtype: int64
```

In [ ]:
```
# Displaying the number of unique values in each column of the DataFrame
df.nunique()
```

Out[ ]:
```
Timestamp
30
What is your age?
6
What is your gender?
3
What is your highest level of education?
6
What is your current occupation?
6
How often do you exercise in a week?
4
How long have you been using a fitness wearable?
4
How frequently do you use your fitness wearable?
4
How often do you track fitness data using wearable?
5
How has the fitness wearable impacted your fitness routine?
4
Has the fitness wearable helped you stay motivated to exercise?
4
Do you think that the fitness wearable has made exercising more enjoyable?
4
How engaged do you feel with your fitness wearable?
4
Does using a fitness wearable make you feel more connected to the fitness
community?                              3
How has the fitness wearable helped you achieve your fitness goals?
3
How has the fitness wearable impacted your overall health?
4
Has the fitness wearable improved your sleep patterns?
4
Do you feel that the fitness wearable has improved your overall well-bein
g?                                       4
Has using a fitness wearable influenced your decision? [To exercise more?]
3
Has using a fitness wearable influenced your decision? [To purchase other
fitness-related products?]      4
Has using a fitness wearable influenced your decision? [To join a gym or f
itness class?]                          3
Has using a fitness wearable influenced your decision? [To change your die
t?]                                     3
dtype: int64
```

In [ ]:
```
# Setting the "Timestamp" column as the index of the DataFrame
df.set_index("Timestamp", inplace=True)
```

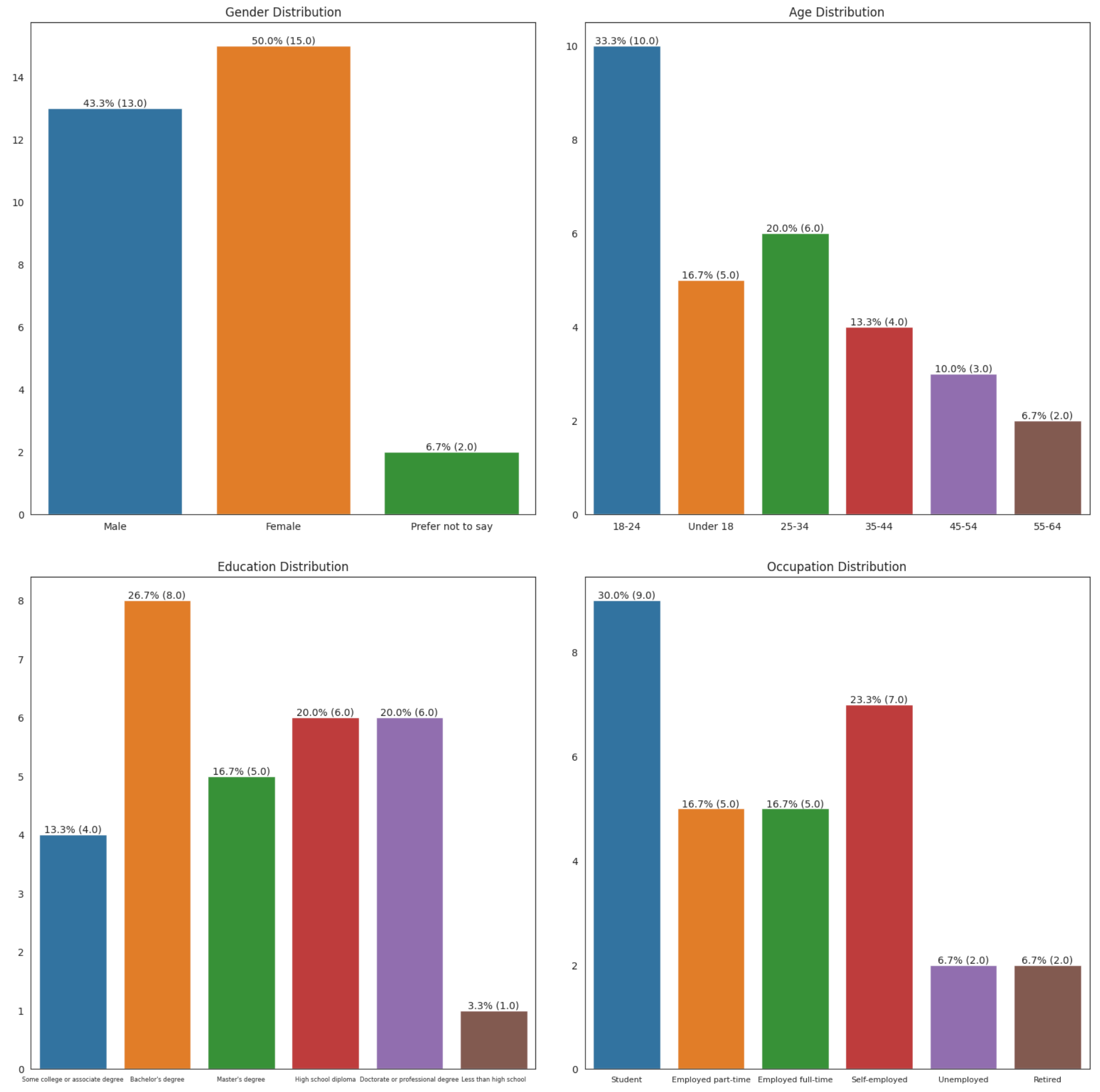
In [ ]:

```python
# Renaming the columns of the DataFrame for easier reference and better readability
df.rename(columns={
    'What is your age?': 'Age',
    'What is your gender?': 'Gender',
    'What is your highest level of education?': 'Education',
    'What is your current occupation?': 'Occupation',
    'How often do you exercise in a week?': 'ExerciseFreq',
    'How long have you been using a fitness wearable?': 'WearableDuration',
    'How frequently do you use your fitness wearable?': 'WearableFreq',
    'How often do you track fitness data using wearable?': 'TrackDataFreq',
    'How has the fitness wearable impacted your fitness routine?': 'RoutineImpact',
    'Has the fitness wearable helped you stay motivated to exercise?': 'MotivationImpact',
    'Do you think that the fitness wearable has made exercising more enjoyable?': 'EnjoymentImpact',
    'How engaged do you feel with your fitness wearable?': 'Engagement',
    'Does using a fitness wearable make you feel more connected to the fitness community?': 'CommunityConnection',
    'How has the fitness wearable helped you achieve your fitness goals?': 'GoalImpact',
    'How has the fitness wearable impacted your overall health?': 'HealthImpact',
    'Has the fitness wearable improved your sleep patterns?': 'SleepImpact',
    'Do you feel that the fitness wearable has improved your overall well-being?': 'WellbeingImpact',
    'Has using a fitness wearable influenced your decision? [To exercise more?]': 'DecisionExerciseMore',
    'Has using a fitness wearable influenced your decision? [To purchase other fitness-related products?]': 'DecisionBuyProducts',
    'Has using a fitness wearable influenced your decision? [To join a gym or fitness class?]': 'DecisionJoinGym',
    'Has using a fitness wearable influenced your decision? [To change your diet?]': 'DecisionChangeDiet'
}, inplace=True)
```

Visual Insights

```
In [ ]: # Set the figure size and style
plt.figure(figsize=(16, 16))
sns.set_style("white")

# Create subplots
plt.subplot(2, 2, 1)
sns.countplot(data=df, x='Gender', hue='Gender', palette='tab10')
plt.title('Gender Distribution')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['Gender'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 2)
sns.countplot(data=df, x='Age', hue='Age', palette='tab10')
plt.title('Age Distribution')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['Age'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 3)
sns.countplot(data=df, x='Education', hue='Education', palette='tab10')
plt.title('Education Distribution')
plt.xticks(fontsize=6)
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['Education'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 4)
sns.countplot(data=df, x='Occupation', hue='Occupation', palette='tab10')
plt.title('Occupation Distribution')
plt.xticks(fontsize=8)
plt.xlabel('')
```

```python
plt.ylabel('')
# Add percentage labels
total = len(df['Occupation'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.tight_layout(pad=3.0)
plt.show()
```



Demographic

The following analysis focuses on the distribution of four key demographic categories: gender, age, education, and occupation.

Gender Distribution

The majority of respondents in the dataset are **female**, accounting for **50.0% (15 individuals)**. **Males** constitute **43.3% (13 individuals)**, while a smaller portion, **6.7% (2 individuals), preferred not to disclose their gender**.

The nearly balanced gender distribution highlights the importance of creating gender-inclusive fitness programs and marketing campaigns that appeal to both men and women.

Age Distribution

Among the age groups, the **18-24 age** bracket is the largest, representing **33.3% (10 individuals)** of the respondents. This is followed by the **under 18** group at **16.7% (5 individuals)**. The **25-34 age** group comprises 20.0% (6 individuals), and the **35-44 age** group is **13.3% (4 individuals)**. The **45-54 age** group makes up 10.0% (3 individuals), and the smallest age category, **55-64**, represents **6.7% (2 individuals)**.

This indicates that young adults and students are highly engaged in fitness-related activities or products. This suggests a significant market potential within educational institutions and among young professionals.

Education Distribution

In terms of educational attainment, **26.7% (8 individuals)** of the respondents hold a **bachelor's degree**, making it the most common educational level. **High school diploma holders** and those with a **doctorate or professional degree** each make up 20.0% (6 individuals). Respondents with a **master's degree** constitute **16.7% (5 individuals)**, while those with **some college or an associate degree** represent **13.3% (4 individuals)**. The least represented group, at **3.3% (1 individual)**, has **less than a high school education**.

This indicates that fitness consumers are likely to be well-educated, potentially valuing scientifically backed fitness programs and products that emphasize health benefits and innovative approaches.

Occupation Distribution

The dataset reveals that **30.0% (9 individuals)** of the respondents are **students**, making it the largest occupational category. **Self-employed individuals** follow at **23.3% (7 individuals)**. Both **employed part-time** and **employed full-time** respondents constitute **16.7% (5 individuals each)**. Those who are **unemployed** and **retired** each make up **6.7% (2 individuals)**.

The presence of various employment statuses suggests that fitness solutions should accommodate different lifestyles, including flexible schedules and varying levels of disposable income.

Conclusion

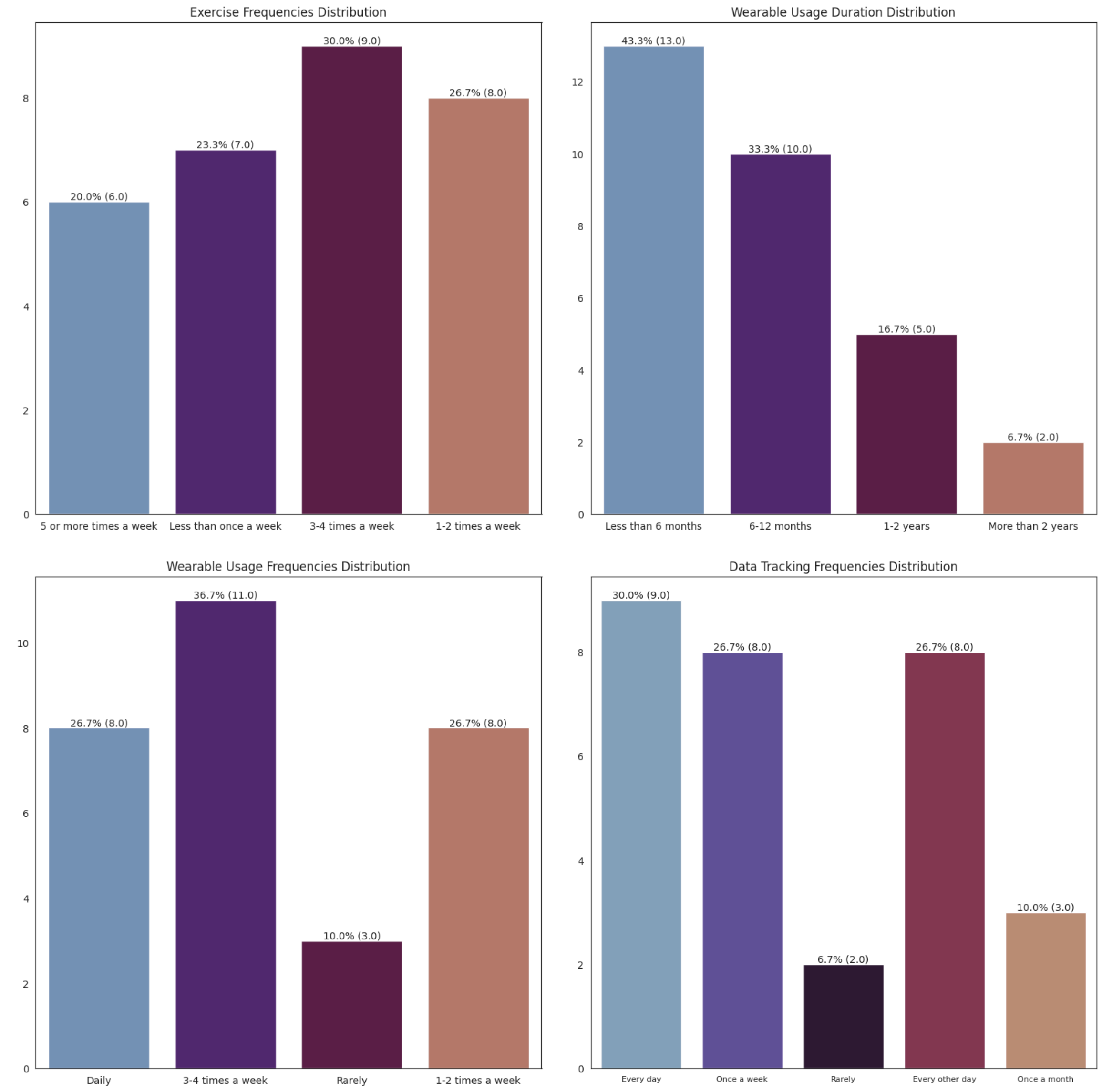
The results indicate that the fitness product or service is most likely to resonate with young, educated individuals, particularly those still in school or early in their careers. Marketing efforts should leverage platforms and channels popular among this demographic, such as social media, online fitness communities, and campus events.

By targeting the young, well-educated, and diverse occupational demographic, fitness providers can better meet the expectations and preferences of their primary consumers.

In [ ]:

```
# Set the figure size and style
plt.figure(figsize=(16, 16))
sns.set_style("white")

# Create subplots
plt.subplot(2, 2, 1)
sns.countplot(data=df, x='ExerciseFreq', hue='ExerciseFreq', palette='twilight')
plt.title('Exercise Frequencies Distribution')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['ExerciseFreq'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 2)
sns.countplot(data=df, x='WearableDuration', hue='WearableDuration', palette='twilight')
plt.title('Wearable Usage Duration Distribution')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['WearableDuration'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 3)
sns.countplot(data=df, x='WearableFreq', hue='WearableFreq', palette='twilight')
plt.title('Wearable Usage Frequencies Distribution')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['WearableFreq'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 4)
sns.countplot(data=df, x='TrackDataFreq', hue='TrackDataFreq', palette='twilight')
plt.title('Data Tracking Frequencies Distribution')
plt.xticks(fontsize=8)
plt.xlabel('')
plt.ylabel('')
```

```
# Add percentage labels
total = len(df['TrackDataFreq'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.tight_layout(pad=3.0)
plt.show()
```



Fitness Habits and Tracking Behaviour

The following analysis fall into the broader category of Fitness Habits and Tracking Behavior. It encompass the regular activities and behaviors of individuals related to their fitness routines and their interaction with fitness tracking technologies.

Exercise Frequencies Distribution

Starting with exercise frequencies, the most common exercise pattern is **3-4 times a week**, accounting for **30.0% of the respondents (9 individuals)**. This is followed by those who exercise **1-2 times a week**, representing **26.7% (8 individuals)**. **Less than once a week** and **5 or more times a week** are reported by **23.3% (7 individuals)** and **20.0% (6 individuals)**, respectively.

This suggests a fairly active audience that might benefit from fitness programs designed to maintain or slightly increase their current activity levels.

Wearable Usage Duration Distribution

Regarding wearable usage duration, the largest group of users has been using their devices for **less than 6 months**, which constitutes **43.3% (13 individuals)**. Those who have used wearables for **6-12 months** come next at **33.3% (10 individuals)**. Users with **1-2 years of experience** make up **16.7% (5 individuals)**, while those using wearables for **more than 2 years** are the smallest group at **6.7% (2 individuals)**.

This indicates a growing interest in fitness tracking technologies and presents an opportunity for marketing efforts to focus on educating new users about the benefits and features of wearables.

Wearable Usage Frequencies

When examining wearable usage frequencies, **36.7% of the participants (11 individuals)** use their wearables **3-4 times a week**, making it the most common frequency. Both **daily usage** and **1-2 times a week** usage are equally prevalent, each at **26.7% (8 individuals)**. A small fraction, **10.0% (3 individuals)**, uses their wearables **rarely**.

This suggests a clear demand for reliable and user-friendly wearable devices. Products should cater to frequent usage with features such as long battery life and comprehensive data analytics.

Data Tracking Frequencies Distribution

In terms of data tracking frequencies, **every day** is the most frequent tracking interval, reported by **30.0% (9 individuals)**. This is closely followed by tracking **every other day** and **once a week**, each at **26.7% (8 individuals)**. Monthly data tracking is noted by **10.0% (3 individuals)**, and **rarely** tracking data is observed in **6.7% (2 individuals)**.

This underscores the need for fitness apps and devices to offer personalized insights and recommendations based on users' data to enhance their fitness journey. This level of personalization can help users make informed decisions about their health and fitness, thereby increasing their satisfaction and loyalty to the product.

Conclusion

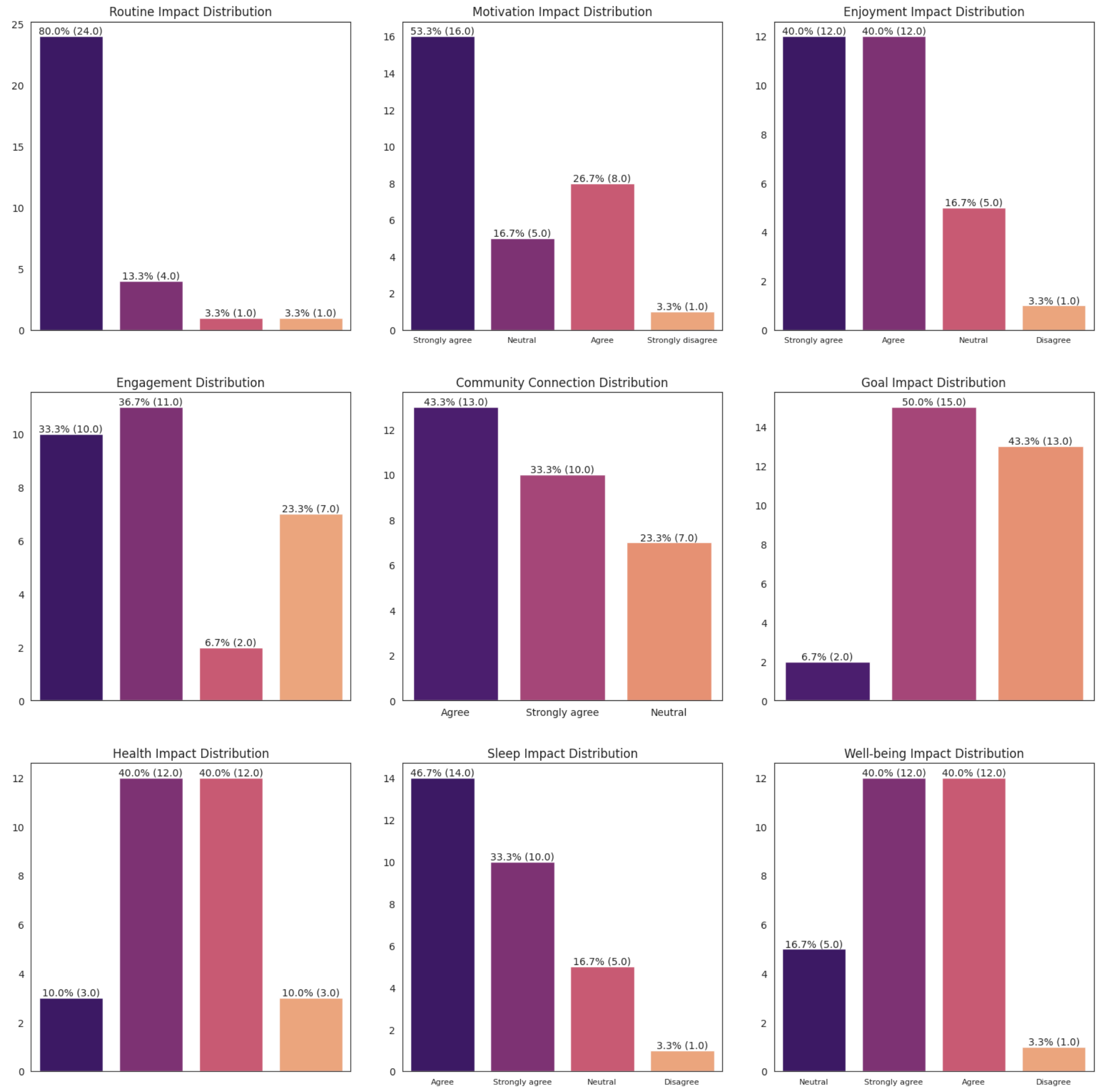
The results suggest that fitness products and services should focus on supporting an active lifestyle and providing useful tracking features. There is a significant market for educational content to assist new users in maximizing their use of wearables and data tracking. Additionally, emphasizing the long-term benefits of consistent exercise and wearable usage can help retain and grow the customer base.

By addressing the specific needs and preferences of this audience, fitness products and services can enhance user satisfaction, retention, and overall market success.

In [ ]:
```
# Set the figure size and style
plt.figure(figsize=(16, 16))
sns.set_style("white")

# Create subplots
plt.subplot(3, 3, 1)
sns.countplot(data=df, x='RoutineImpact', hue='RoutineImpact', palette='magma')
plt.title('Routine Impact Distribution')
plt.xticks([])
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['RoutineImpact'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(3, 3, 2)
sns.countplot(data=df, x='MotivationImpact', hue='MotivationImpact', palette='magma')
plt.title('Motivation Impact Distribution')
plt.xticks(fontsize=8)
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['MotivationImpact'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(3, 3, 3)
sns.countplot(data=df, x='EnjoymentImpact', hue='EnjoymentImpact', palette='magma')
plt.title('Enjoyment Impact Distribution')
plt.xticks(fontsize=8)
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['EnjoymentImpact'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(3, 3, 4)
sns.countplot(data=df, x='Engagement', hue='Engagement', palette='magma')
plt.title('Engagement Distribution')
plt.xticks([])
```

```
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['Engagement'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(3, 3, 5)
sns.countplot(data=df, x='CommunityConnection', hue='CommunityConnection',
palette='magma')
plt.title('Community Connection Distribution')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['CommunityConnection'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(3, 3, 6)
sns.countplot(data=df, x='GoalImpact', hue='GoalImpact', palette='magma')
plt.title('Goal Impact Distribution')
plt.xticks([])
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['GoalImpact'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(3, 3, 7)
sns.countplot(data=df, x='HealthImpact', hue='HealthImpact', palette='magm
a')
plt.title('Health Impact Distribution')
plt.xticks([])
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['HealthImpact'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')
```

```
plt.subplot(3, 3, 8)
sns.countplot(data=df, x='SleepImpact', hue='SleepImpact', palette='magma')
plt.title('Sleep Impact Distribution')
plt.xticks(fontsize=8)
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['SleepImpact'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(3, 3, 9)
sns.countplot(data=df, x='WellbeingImpact', hue='WellbeingImpact', palette
='magma')
plt.title('Well-being Impact Distribution')
plt.xticks(fontsize=8)
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['WellbeingImpact'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.tight_layout(pad=3.0)
plt.show()
```



Impact Distribution

The following analysis focus on Impact Distributions. These illustrates how various factors influence different aspects of fitness routines and experiences for consumers. These distributions provide insights into the extent and nature of these impacts which highlights trends and patterns in user responses.

Routine Impact Distribution

In terms of the impact on fitness routines, **80.0% (24 individuals)** reported that their routines were **positively impacted**, while **13.3% (4 individuals)** were uncertain, responding with **I don't know**. Only **3.3% (1 individual)** felt that there was **no impact**, and another **3.3% (1 individual)** felt **negatively impacted**.

This indicates that the factors analyzed significantly enhance fitness routines for most respondents, with very few reporting no or negative impact.

Motivation Impact Distribution

Regarding motivation, **53.3% (16 individuals) strongly agreed** that their motivation was positively impacted, and **26.7% (8 individuals) agreed**. **16.7% (5 individuals)** were **neutral**, and **3.3% (1 individual) strongly disagreed**.

This suggests that the majority of respondents feel an increased motivation due to the analyzed factors, showing their effectiveness in boosting users' drive to maintain their fitness routines.

Enjoyment Impact Distribution

When it comes to enjoyment, both **40.0% (12 individuals) strongly agreed** and **agreed** that their enjoyment increased, while **16.7% (5 individuals)** were **neutral** and **3.3% (1 individual) disagreed**.

This demonstrates that a significant portion of respondents find their fitness activities more enjoyable, which is crucial for sustaining long-term engagement.

Engagement Distribution

In terms of engagement, **36.7% (11 individuals)** reported being **somewhat engaged**, while **33.3% (10 individuals)** were **very engaged**. **23.3% (7 individuals)** were **neutral**, and **6.7% (2 individuals)** were **not very engaged**.

This indicates a high level of interaction with fitness routines for over 70% of participants, suggesting that the analyzed factors effectively foster engagement.

Community Connection Distribution

For community connection, **43.3% (13 individuals) agreed**, and **33.3% (10 individuals) strongly agreed** that they felt a sense of community, while **23.3% (7 individuals)** were **neutral**.

This implies that the community aspects of fitness routines are important for many users, fostering a sense of belonging and support.

Goal Impact Distribution

In terms of goal achievement, **50.0% (15 individuals)** felt that the factors **helped them achieve their goals somewhat more quickly**, and **43.3% (13 individuals) helped them achieve their goals much more quickly**, while only **6.7% (2 individuals)** reported **no impact**.

This indicates a significant efficacy in goal achievement support, with most respondents experiencing quicker progress.

Health Impact Distribution

Regarding health impact, **40.0% (12 individuals)** reported that their health was **significantly improved**, and another **40.0% (12 individuals)** said it was **somewhat improved**. **10.0% (3 individuals)** felt there was **no impact**, and another **10.0% (3 individuals)** responded with **I don't know**.

This highlights a positive impact on overall health for the majority of respondents, reinforcing the benefits of the analyzed factors.

Sleep Impact Distribution

When considering sleep impact, **46.7% (14 individuals) agreed** and **33.3% (10 individuals) strongly agreed** that their sleep improved, while **16.7% (5 individuals)** were **neutral** and **3.3% (1 individual) disagreed**.

This suggests that the factors positively contribute to better sleep quality for most respondents.

Well-being Distribution

In terms of overall well-being, **40.0% (12 individuals) strongly agreed**, and another **40.0% (12 individuals) agreed** that their well-being improved, while **16.7% (5 individuals)** were **neutral** and **3.3% (1 individual) disagreed**.

This indicates that the analyzed factors significantly enhance users' mental and physical well-being, contributing to a better quality of life.

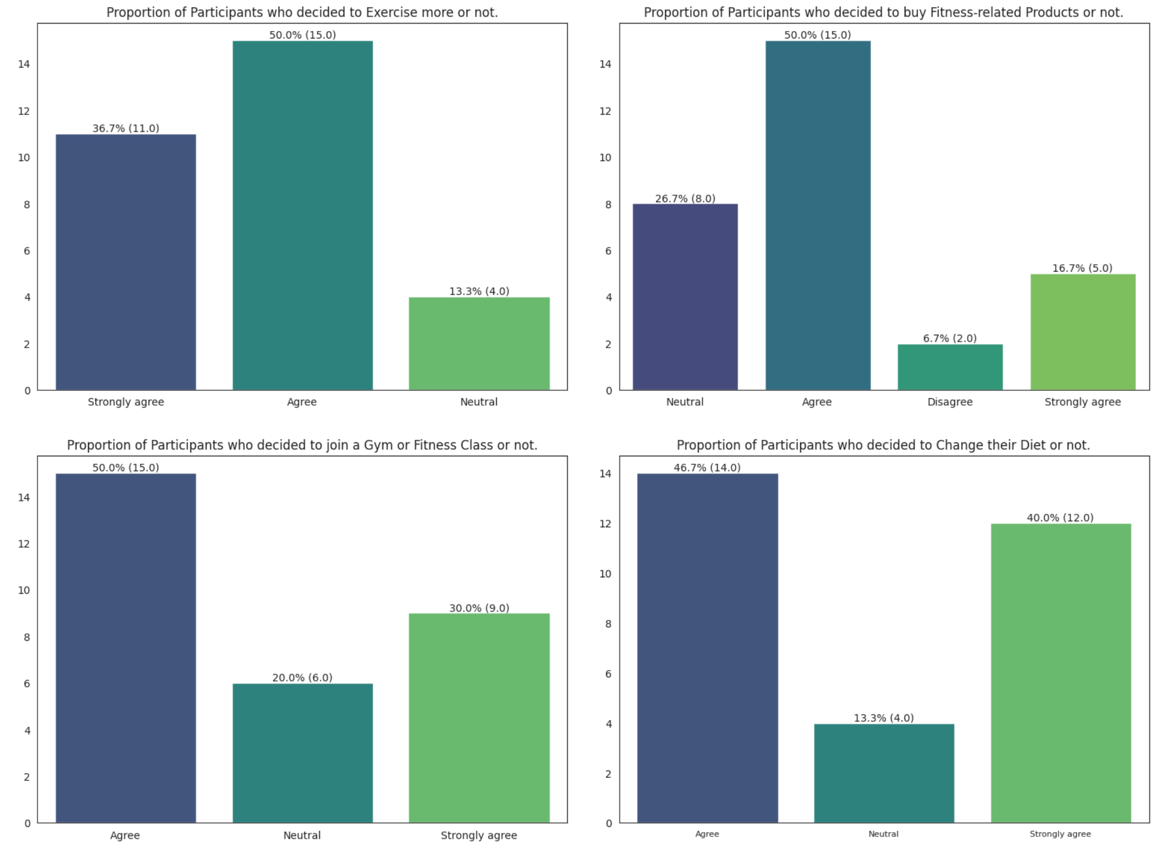
Conclusion

The findings suggest that the analyzed factors are effective in enhancing users' fitness experiences, contributing to more consistent and enjoyable routines, better health outcomes, and a stronger sense of community. This comprehensive positive impact underscores the importance of considering these factors in designing fitness programs and interventions to maximize user satisfaction and effectiveness.

In [ ]:

```
# Set the figure size and style
plt.figure(figsize=(16, 12))
sns.set_style("white")

# Create subplots
plt.subplot(2, 2, 1)
sns.countplot(data=df, x='DecisionExerciseMore', hue='DecisionExerciseMor
e', palette='viridis')
plt.title('Proportion of Participants who decided to Exercise more or no
t.')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['DecisionExerciseMore'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 2)
sns.countplot(data=df, x='DecisionBuyProducts', hue='DecisionBuyProducts',
palette='viridis')
plt.title('Proportion of Participants who decided to buy Fitness-related Pr
oducts or not.')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['DecisionBuyProducts'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 3)
sns.countplot(data=df, x='DecisionJoinGym', hue='DecisionJoinGym', palette
='viridis')
plt.title('Proportion of Participants who decided to join a Gym or Fitness
Class or not.')
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['DecisionJoinGym'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.subplot(2, 2, 4)
sns.countplot(data=df, x='DecisionChangeDiet', hue='DecisionChangeDiet', pa
lette='viridis')
plt.title('Proportion of Participants who decided to Change their Diet or n
```

```
ot.')
plt.xticks(fontsize=8)
plt.xlabel('')
plt.ylabel('')
# Add percentage labels
total = len(df['DecisionChangeDiet'])
for p in plt.gca().patches:
    count = p.get_height()
    percentage = '{:.1f}%'.format(100 * count / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    label = f'{percentage} ({count})'
    plt.annotate(label, (x, y), ha='center', va='bottom')

plt.tight_layout(pad=3.0)
plt.show()
```



Consumer Behavior

The following analysis focuses on Consumer Behavior. These illustrates how various factors influence participants' choices regarding their fitness behaviors. These distributions provide insights into the extent and nature of these behavioral changes, highlighting trends and patterns in user responses.

Proportion of Participants who decided to Exercise more

A significant **50.0% (15 individuals)** of respondents **agree** that they decided to exercise more, with an additional **36.7% (11 individuals) strongly agreeing**. Only **13.3% (4 individuals)** remained **neutral**.

This demonstrates a strong inclination towards increasing physical activity among the majority of the participants.

Proportion of Participants who decided to Change their diet

In terms of dietary changes, **46.7% (14 individuals)** of respondents **agreed**, and **40.0% (12 individuals) strongly agreed** that they decided to change their diet, while only **13.3% (4 individuals)** remained **neutral**.

This indicates a considerable interest in making dietary improvements alongside physical activity.

Proportion of Participants who decided to join a Gym or Fitness Class

Half of the respondents **50.0% (15 individuals) agreed** that they decided to join a gym or fitness class, with **30.0% (9 individuals) strongly agreeing**. **20.0% (6 individuals)** remained **neutral**.

These highlights a strong tendency towards engaging in structured fitness programs.

Proportion of Participants who decided to buy Fitness-related Products

Regarding the purchase of fitness-related products, **50.0% (15 individuals) agreed** and **16.7% (5 individuals) strongly agreed** to having made such purchases. Meanwhile, **26.7% (8 individuals)** remained **neutral** and a small percentage, **6.7% (2 individuals), disagreed**.

This shows a considerable market potential for fitness-related products among the participants.

Conclusion

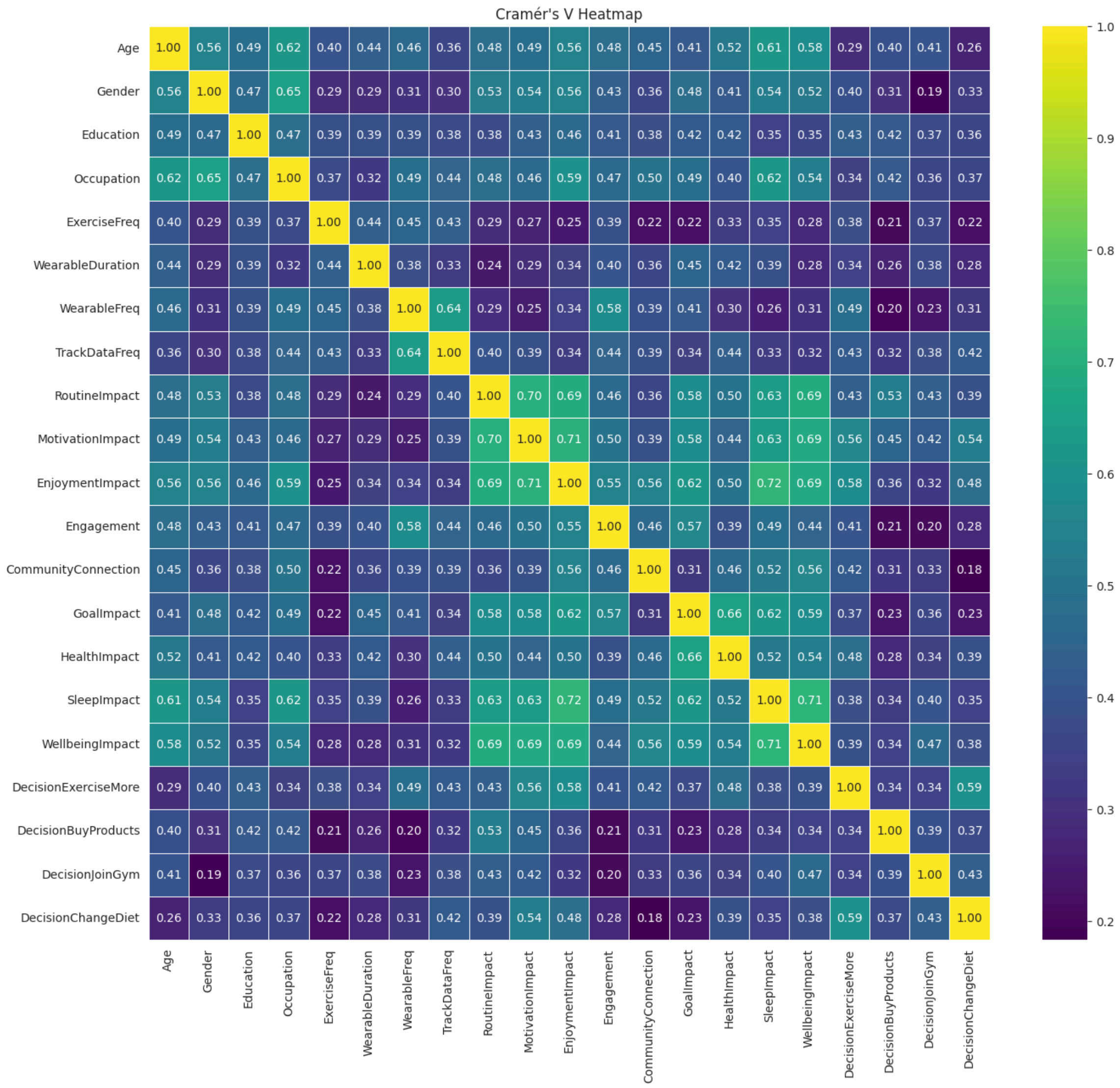
These insights suggest that the factors analyzed are highly effective in motivating participants to adopt healthier lifestyles and commit to their fitness goals. This comprehensive positive impact underscores the importance of addressing multiple aspects of fitness, including exercise, nutrition, and community support, to maximize user satisfaction and effectiveness in fitness programs and interventions.

Key Findings and Business Impact

- `Cramér's V` is a measure of association between two nominal (categorical) variables. It is based on the chi-squared statistic and ranges from 0 (no association) to 1 (perfect association).

In [ ]:

```
# Defining a function to calculate Cramér's V, which measures the associati
on between two categorical variables
def cramers_v(x, y):
  # Creating a confusion matrix (contingency table) for the two variables
  confusion_matrix = pd.crosstab(x, y)

  # Calculating the chi-square statistic for the confusion matrix
  chi2 = chi2_contingency(confusion_matrix)[0]

  # Getting the total number of observations
  n = confusion_matrix.sum().sum()

  # Getting the shape of the confusion matrix (number of rows and columns)
  r, k = confusion_matrix.shape

  # Calculating and returning Cramér's V
  return np.sqrt(chi2 / (n * (min(r, k) - 1)))

# List of categorical columns in the DataFrame
categorical_cols = ['Age', 'Gender', 'Education', 'Occupation', 'ExerciseFr
eq',
                    'WearableDuration', 'WearableFreq', 'TrackDataFreq', 'R
outineImpact',
                    'MotivationImpact', 'EnjoymentImpact', 'Engagement',
                    'CommunityConnection', 'GoalImpact', 'HealthImpact', 'S
leepImpact',
                    'WellbeingImpact', 'DecisionExerciseMore', 'DecisionBuy
Products',
                    'DecisionJoinGym', 'DecisionChangeDiet']

# Creating an empty DataFrame to store the Cramér's V values for pairs of c
ategorical variables
cramers_v_matrix = pd.DataFrame(index=categorical_cols, columns=categorical
_cols)

# Calculating Cramér's V for each pair of categorical variables and store i
n the DataFrame
for col1 in categorical_cols:
  for col2 in categorical_cols:
    cramers_v_matrix.loc[col1, col2] = cramers_v(df[col1], df[col2])

# Converting the Cramér's V values to float type
cramers_v_matrix = cramers_v_matrix.astype(float)

# Plotting a heatmap of the Cramér's V matrix to visualize the associations
between categorical variables
plt.figure(figsize=(16, 14))  # Setting the size of the plot
sns.heatmap(cramers_v_matrix, annot=True, cmap='viridis', fmt='.2f', linewi
dths=.5)  # Creating the heatmap
plt.title('Cramér\'s V Heatmap')  # Adding a title to the heatmap
plt.show()  # Displaying the plot
```



Data Analysis Techniques

- `df.describe()` is a method in pandas that provides descriptive statistics of a DataFrame. It includes metrics like count, mean, standard deviation, min, and max values, and the 25th, 50th (median), and 75th percentiles for numerical columns.
- `ANOVA` is a statistical method used to compare the means of three or more samples to see if at least one of the sample means is significantly different from the others.

In [ ]:

```python
# Displaying summary statistics for the numerical columns in the DataFrame
df.describe()

df.to_csv('Fitness Consumer Describe.csv')
```

In [ ]:

```python
# List of independent variables (predictors) to be used in the analysis
independent_var = [
    'Age', 'Gender', 'Education', 'Occupation', 'ExerciseFreq',
    'WearableDuration', 'WearableFreq', 'TrackDataFreq', 'RoutineImpact',
    'MotivationImpact', 'EnjoymentImpact', 'Engagement', 'CommunityConnecti
on',
    'GoalImpact', 'HealthImpact', 'SleepImpact', 'WellbeingImpact'
]

# Converting each independent variable to categorical codes
for col in independent_var:
    df[col] = df[col].astype('category').cat.codes

# Mapping for converting Likert scale responses to numerical codes
decision_map = {
    'Strongly disagree': 0,
    'Disagree': 1,
    'Neutral': 2,
    'Agree': 3,
    'Strongly agree': 4
}

# List of dependent variables (outcomes) to be used in the analysis
dependent_var = [
    'DecisionExerciseMore', 'DecisionBuyProducts', 'DecisionJoinGym', 'Deci
sionChangeDiet'
]

# Converting each dependent variable to numerical codes using the defined m
apping
for col in dependent_var:
    df[col] = df[col].map(decision_map)

# List to store ANOVA results
anova_results = []

# Performing ANOVA for each combination of dependent and independent variab
les
for dep in dependent_var:
    for ind in independent_var:
        # Fitting an Ordinary Least Squares (OLS) model
        model = ols(f'{dep} ~ C({ind})', data=df).fit()
        # Performing ANOVA on the fitted model
        anova_table = sm.stats.anova_lm(model, typ=2)
        # Extracting the p-value from the ANOVA table
        p_value = anova_table["PR(>F)"][0]
        # Appending the results to the anova_results list
        anova_results.append({
            'Independent Variable': ind,
            'Dependent Variable': dep,
            'p-value': p_value
        })

# Converting the list of ANOVA results to a DataFrame for easier viewing
pd.DataFrame(anova_results)
```

Out[ ]:

	Independent Variable	Dependent Variable	p-value
0	Age	DecisionExerciseMore	0.931187
1	Gender	DecisionExerciseMore	0.880298
2	Education	DecisionExerciseMore	0.366345
3	Occupation	DecisionExerciseMore	0.734031
4	ExerciseFreq	DecisionExerciseMore	0.104605
...	...	...	...
63	CommunityConnection	DecisionChangeDiet	0.502188
64	GoalImpact	DecisionChangeDiet	0.249156
65	HealthImpact	DecisionChangeDiet	0.494087
66	SleepImpact	DecisionChangeDiet	0.310736
67	WellbeingImpact	DecisionChangeDiet	0.272125

68 rows × 3 columns

Implementation of Machine Learning

For implementing Machine Learning, I used the following features: `['Age', 'Gender', 'Education', 'Occupation', 'ExerciseFreq', 'WearableDuration', 'WearableFreq', 'TrackDataFreq', 'RoutineImpact', 'MotivationImpact', 'EnjoymentImpact', 'Engagement', 'CommunityConnection', 'GoalImpact', 'HealthImpact', 'SleepImpact', 'WellbeingImpact']`.

These features were used to predict the target variables: `['DecisionExerciseMore', 'DecisionBuyProducts', 'DecisionJoinGym', and 'DecisionChangeDiet']`. Multiple Machine Learning models were applied to determine the most effective one.

- `Logistic Regression` is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). It estimates the probability that a given input point belongs to a certain class.
- `Random Forest` is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- `Gradient Boosting` is an ensemble technique that builds models sequentially. Each new model attempts to correct the errors made by the previous model. Models are added until no further improvements can be made.
- `Support Vector Machine` is a supervised learning model that analyzes data for classification and regression analysis. It finds the hyperplane that best separates the data into classes with the maximum margin.
- `Gaussian Naive Bayes` is a probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features. When dealing with continuous data, it assumes that the continuous values associated with each class are distributed according to a Gaussian distribution.
- `K-Nearest Neighbor` is a non-parametric, lazy learning algorithm that classifies a data point based on how its neighbors are classified. It assigns the class most common among its k nearest neighbors.
- `Decision Tree` is a non-parametric supervised learning algorithm used for classification and regression. It splits the data into subsets based on the value of input features, creating a tree-like model of decisions.

In [ ]:

```python
# Defining feature and target columns
features = ['Age', 'Gender', 'Education', 'Occupation', 'ExerciseFreq', 'We
arableDuration',
            'WearableFreq', 'TrackDataFreq', 'RoutineImpact', 'MotivationIm
pact', 'EnjoymentImpact',
            'Engagement', 'CommunityConnection', 'GoalImpact', 'HealthImpac
t', 'SleepImpact', 'WellbeingImpact']
targets = ['DecisionExerciseMore', 'DecisionBuyProducts', 'DecisionJoinGy
m', 'DecisionChangeDiet']

# Encoding categorical features
label_encoders = {}
for col in features:
    # Initializing label encoder
    le = LabelEncoder()
    # Fitting and transforming the feature column
    df[col] = le.fit_transform(df[col])
    # Storing the label encoder for future use
    label_encoders[col] = le

# Encoding target variables
for target in targets:
    # Initializing label encoder
    le = LabelEncoder()
    # Fitting and transform the target column
    df[target] = le.fit_transform(df[target])
    # Storing the label encoder for future use
    label_encoders[target] = le

# Defining a function to train and evaluate models
def train_evaluate_model(X, y):
    results = {}
    # Defining different models to evaluate
    models = {
        'Logistic Regression': LogisticRegression(max_iter=1000),
        'Random Forest': RandomForestClassifier(),
        'Gradient Boosting': GradientBoostingClassifier(),
        'Support Vector Machine': SVC(),
        'Gaussian Naive Bayes': GaussianNB(),
        'K-Nearest Neighbor': KNeighborsClassifier(),
        'Decision Tree': DecisionTreeClassifier(random_state=42)
    }
    # Splitting the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.
2, random_state=42)
    # Training and evaluating each model
    for model_name, model in models.items():
        # Training the model
        model.fit(X_train, y_train)
        # Predicting on the test set
        y_pred = model.predict(X_test)
        # Storing the evaluation metrics for the model
        results[model_name] = {
            'Accuracy': accuracy_score(y_test, y_pred),
            'Precision': precision_score(y_test, y_pred, average='weighte
d', zero_division=0),
            'Recall': recall_score(y_test, y_pred, average='weighted', zero
_division=0),
            'F1 Score': f1_score(y_test, y_pred, average='weighted', zero_d
ivision=0)
```

```
        }
    return results

# Defining a function to print the results in a formatted manner
def print_formatted_results(results):
    for target, model_results in results.items():
        print(target)
        for model_name, metrics in model_results.items():
            print(f"* {model_name}:")
            print(f"   * Accuracy: {metrics['Accuracy']:.2%}")
            print(f"   * Precision: {metrics['Precision']:.2%}")
            print(f"   * Recall: {metrics['Recall']:.2%}")
            print(f"   * F1 Score: {metrics['F1 Score']:.2%}")
        print()

# Performing analysis for each target variable
analysis_results = {}
for target in targets:
    X = df[features]
    y = df[target]
    # Training and evaluating models for the target variable
    analysis_results[target] = train_evaluate_model(X, y)

# Printing the formatted results
print_formatted_results(analysis_results)
```

DecisionExerciseMore
* Logistic Regression:
  * Accuracy: 50.00%
  * Precision: 70.83%
  * Recall: 50.00%
  * F1 Score: 51.11%
* Random Forest:
  * Accuracy: 50.00%
  * Precision: 70.83%
  * Recall: 50.00%
  * F1 Score: 51.11%
* Gradient Boosting:
  * Accuracy: 50.00%
  * Precision: 70.83%
  * Recall: 50.00%
  * F1 Score: 51.11%
* Support Vector Machine:
  * Accuracy: 50.00%
  * Precision: 70.83%
  * Recall: 50.00%
  * F1 Score: 51.11%
* Gaussian Naive Bayes:
  * Accuracy: 50.00%
  * Precision: 70.83%
  * Recall: 50.00%
  * F1 Score: 51.11%
* K-Nearest Neighbor:
  * Accuracy: 66.67%
  * Precision: 72.22%
  * Recall: 66.67%
  * F1 Score: 65.48%
* Decision Tree:
  * Accuracy: 50.00%
  * Precision: 70.83%
  * Recall: 50.00%
  * F1 Score: 51.11%

DecisionBuyProducts
* Logistic Regression:
  * Accuracy: 66.67%
  * Precision: 66.67%
  * Recall: 66.67%
  * F1 Score: 66.67%
* Random Forest:
  * Accuracy: 66.67%
  * Precision: 66.67%
  * Recall: 66.67%
  * F1 Score: 66.67%
* Gradient Boosting:
  * Accuracy: 50.00%
  * Precision: 83.33%
  * Recall: 50.00%
  * F1 Score: 62.50%
* Support Vector Machine:
  * Accuracy: 83.33%
  * Precision: 69.44%
  * Recall: 83.33%
  * F1 Score: 75.76%
* Gaussian Naive Bayes:
  * Accuracy: 83.33%
  * Precision: 91.67%

* Recall: 83.33%
  * F1 Score: 85.19%
* K-Nearest Neighbor:
  * Accuracy: 50.00%
  * Precision: 62.50%
  * Recall: 50.00%
  * F1 Score: 55.56%
* Decision Tree:
  * Accuracy: 83.33%
  * Precision: 83.33%
  * Recall: 83.33%
  * F1 Score: 83.33%

DecisionJoinGym
* Logistic Regression:
  * Accuracy: 66.67%
  * Precision: 100.00%
  * Recall: 66.67%
  * F1 Score: 79.37%
* Random Forest:
  * Accuracy: 83.33%
  * Precision: 100.00%
  * Recall: 83.33%
  * F1 Score: 90.48%
* Gradient Boosting:
  * Accuracy: 83.33%
  * Precision: 100.00%
  * Recall: 83.33%
  * F1 Score: 88.89%
* Support Vector Machine:
  * Accuracy: 83.33%
  * Precision: 100.00%
  * Recall: 83.33%
  * F1 Score: 88.89%
* Gaussian Naive Bayes:
  * Accuracy: 33.33%
  * Precision: 11.11%
  * Recall: 33.33%
  * F1 Score: 16.67%
* K-Nearest Neighbor:
  * Accuracy: 66.67%
  * Precision: 100.00%
  * Recall: 66.67%
  * F1 Score: 79.37%
* Decision Tree:
  * Accuracy: 66.67%
  * Precision: 100.00%
  * Recall: 66.67%
  * F1 Score: 79.37%

DecisionChangeDiet
* Logistic Regression:
  * Accuracy: 83.33%
  * Precision: 87.50%
  * Recall: 83.33%
  * F1 Score: 82.86%
* Random Forest:
  * Accuracy: 50.00%
  * Precision: 50.00%
  * Recall: 50.00%
  * F1 Score: 48.57%

```
* Gradient Boosting:
   * Accuracy: 50.00%
   * Precision: 58.33%
   * Recall: 50.00%
   * F1 Score: 53.33%
* Support Vector Machine:
   * Accuracy: 66.67%
   * Precision: 80.00%
   * Recall: 66.67%
   * F1 Score: 62.50%
* Gaussian Naive Bayes:
   * Accuracy: 50.00%
   * Precision: 50.00%
   * Recall: 50.00%
   * F1 Score: 48.57%
* K-Nearest Neighbor:
   * Accuracy: 83.33%
   * Precision: 87.50%
   * Recall: 83.33%
   * F1 Score: 82.86%
* Decision Tree:
   * Accuracy: 50.00%
   * Precision: 58.33%
   * Recall: 50.00%
   * F1 Score: 53.33%
```

The models are predicting various decisions related to fitness and lifestyle changes. Here's what each target represents and what the model predicts:

- `DecisionExerciseMore` : This target predicts whether an individual will decide to exercise more based on the provided features.
- `DecisionBuyProducts` : This target predicts whether an individual will decide to buy fitness-related products (e.g., wearables, supplements, equipment) based on the provided features.
- `DecisionJoinGym` : This target predicts whether an individual will decide to join a gym based on the provided features.
- `DecisionChangeDiet` : This target predicts whether an individual will decide to change their diet based on the provided features.

Guide to the metrics:

- **Accuracy** - proportion of correctly classified instances out of the total instances. Higher accuracy means the model correctly predicts more instances overall, indicating better overall performance.
- **Precision** - proportion of true positive predictions out of the total predicted positives. igher precision means the model has fewer false positive errors, indicating that when the model predicts a positive instance, it is more likely to be correct.
- **Recall** - proportion of true positive predictions out of the actual positives. Higher recall means the model has fewer false negative errors, indicating that the model can identify more of the actual positive instances.
- **F1 Score** - the harmonic mean of precision and recall. A higher F1 score indicates a better balance between precision and recall, which is particularly useful when there is an uneven class distribution.

Based on the provided results, the best model for the `DecisionExerciseMore` prediction task is the K-Nearest Neighbor (KNN) model.

- **Accuracy**: KNN had the highest accuracy at 66.67%, compared to 50.00% for all other models.
- **Precision**: KNN had a precision of 72.22%, which is slightly higher than the 70.83% for the other models.
- **Recall**: KNN had the highest recall at 66.67%, while the other models had a recall of 50.00%.
- **F1 Score**: KNN had the highest F1 score at 65.48%, compared to 51.11% for the other models.

Based on the provided results, the best model for the `DecisionBuyProducts` prediction task is the Gaussian Naive Bayes model.

- **Accuracy**: Gaussian Naive Bayes, Random Forest, Support Vector Machine, and Decision Tree all have an accuracy of 83.33%, which is the highest among the models.
- **Precision**: Gaussian Naive Bayes has the highest precision at 91.67%.
- **Recall**: All top-performing models (Gaussian Naive Bayes, Random Forest, Support Vector Machine, and Decision Tree) have a recall of 83.33%.
- **F1 Score**: Gaussian Naive Bayes has the highest F1 score at 85.19%.

Based on the provided results, the best model for the `DecisionJoinGym` prediction task is any of the following three models, as they have identical metrics:

- Random Forest
- Gradient Boosting
- Support Vector Machine

Here are the key metrics for these models:

- **Accuracy**: 83.33% (highest among the models)
- **Precision**: 100.00% (highest among the models)
- **Recall**: 83.33% (highest among the models)
- **F1 Score**: 88.89% (highest among the models)

Based on the provided results, the best models for the `DecisionChangeDiet` prediction task are:

- Logistic Regression
- K-Nearest Neighbor (KNN)

Both models have identical metrics:

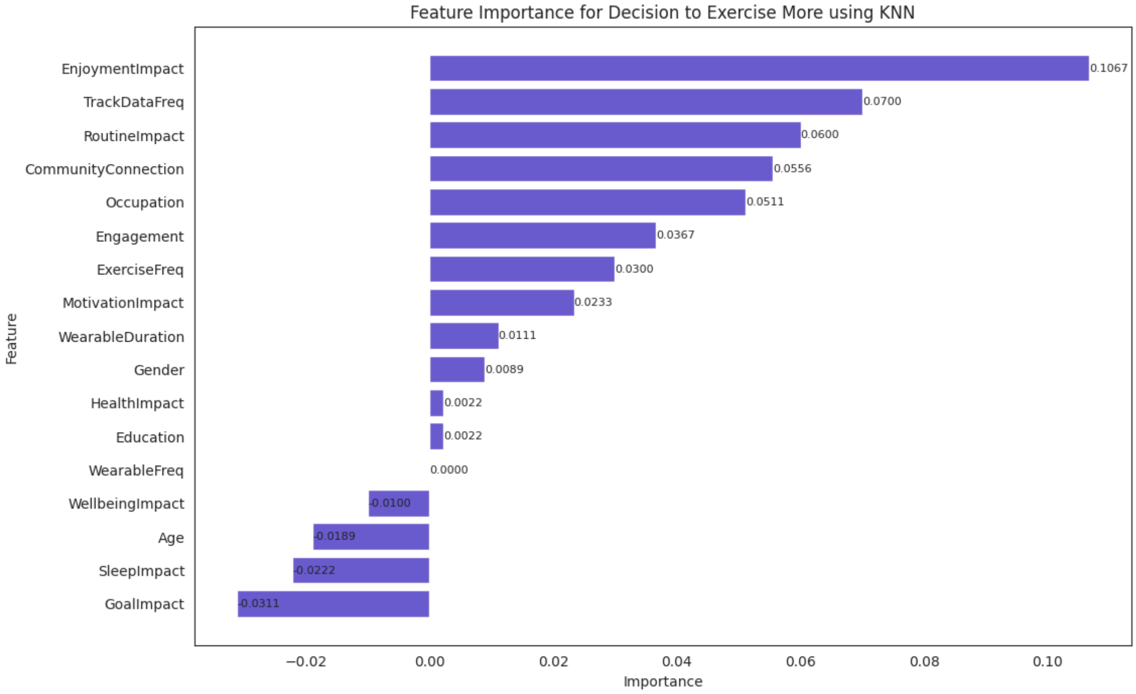
- **Accuracy**: 83.33% (highest among the models)
- **Precision**: 87.50% (highest among the models)
- **Recall**: 83.33% (highest among the models)
- **F1 Score**: 82.86% (highest among the models)

Advanced Analysis

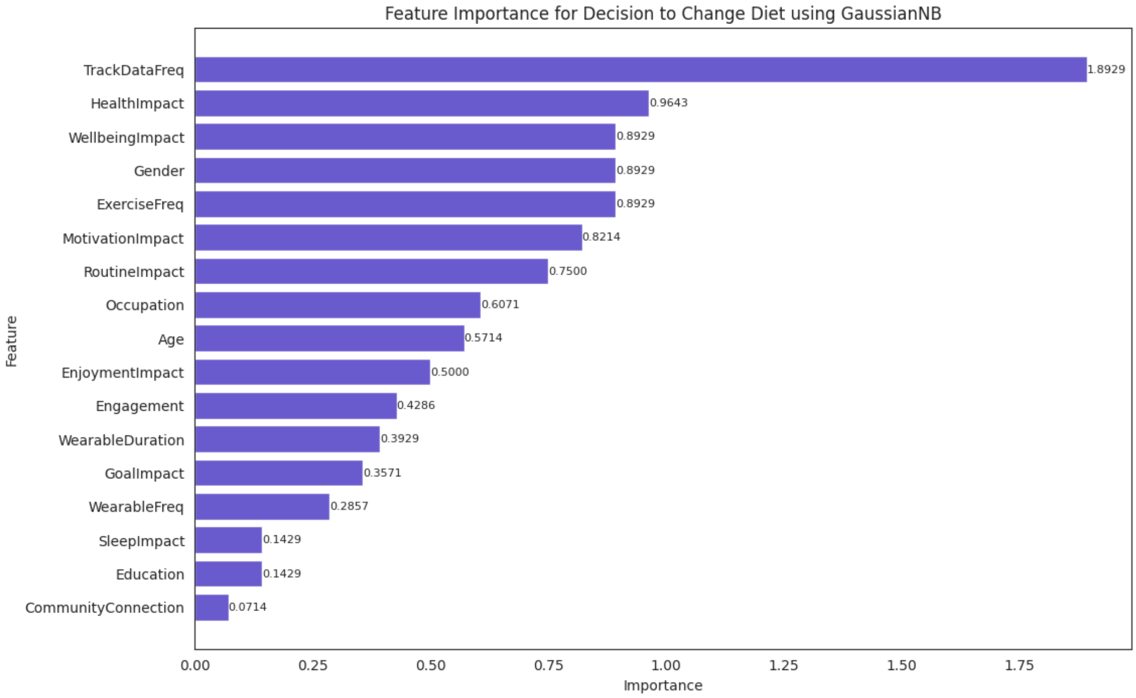
For this part, I identified the feature importance for each target variable using the top-performing models based on evaluation metrics. Feature importance refers to a technique used to assign scores to input features based on their significance in predicting a target variable. It helps in understanding which features contribute the most to the model's predictions.

In [ ]:
```
# Feature Importance for DecisionExerciseMore using KNN
X = df[features]
y = df['DecisionExerciseMore']
# Initializing and training the KNN model
knn = KNeighborsClassifier()
knn.fit(X, y)
# Calculating permutation importance
perm_importance_knn = permutation_importance(knn, X, y, n_repeats=30, rando
m_state=42, n_jobs=-1)
# Getting mean importance and sort features by importance
knn_importances = perm_importance_knn.importances_mean
knn_indices = np.argsort(knn_importances)[::-1]
# Creating a DataFrame to store feature importance
knn_importance_df = pd.DataFrame({
    'Feature': [features[i] for i in knn_indices],
    'Importance': knn_importances[knn_indices]
})

# Feature Importance for DecisionChangeDiet using GaussianNB
X = df[features]
y = df['DecisionChangeDiet']
# Initializing and training the Gaussian Naive Bayes model
gnb = GaussianNB()
gnb.fit(X, y)
# Calculating feature importance based on the difference in means of the cl
asses
gnb_importances = np.abs(gnb.theta_[1] - gnb.theta_[0])
# Sorting features by importance
gnb_indices = np.argsort(gnb_importances)[::-1]
# Creating a DataFrame to store feature importance
gnb_importance_df = pd.DataFrame({
    'Feature': [features[i] for i in gnb_indices],
    'Importance': gnb_importances[gnb_indices]
})

# Feature Importance for DecisionJoinGym using Random Forest
X = df[features]
y = df['DecisionJoinGym']
# Initializing and training the Random Forest model
rf = RandomForestClassifier(random_state=42)
rf.fit(X, y)
# Getting feature importance from the model
rf_importances = rf.feature_importances_
# Sorting features by importance
rf_indices = np.argsort(rf_importances)[::-1]
# Creating a DataFrame to store feature importance
rf_importance_df = pd.DataFrame({
    'Feature': [features[i] for i in rf_indices],
    'Importance': rf_importances[rf_indices]
})

# Feature Importance for DecisionChangeDiet using Logistic Regression
X = df[features]
y = df['DecisionChangeDiet']
# Initializing and training the Logistic Regression model
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X, y)
# Calculating feature importance based on the absolute value of coefficient
s
log_reg_importances = np.abs(log_reg.coef_[0])
```

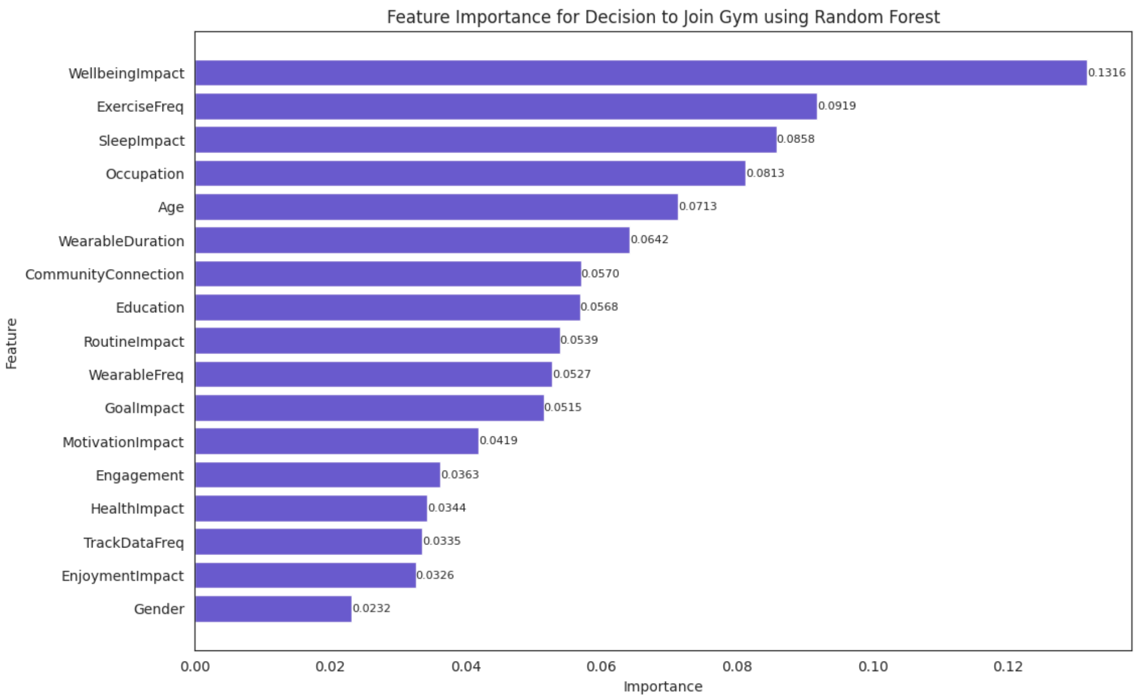
```
# Sorting features by importance
log_reg_indices = np.argsort(log_reg_importances)[::-1]
# Creating a DataFrame to store feature importance
log_reg_importance_df = pd.DataFrame({
    'Feature': [features[i] for i in log_reg_indices],
    'Importance': log_reg_importances[log_reg_indices]
})

# Function to plot feature importances with numerical values at the end of the bars
def plot_feature_importances(df, title):
    plt.figure(figsize=(12, 8))
    plt.title(title)
    bars = plt.barh(df['Feature'], df['Importance'], color='slateblue')
    plt.xlabel('Importance')
    plt.ylabel('Feature')
    plt.gca().invert_yaxis()

    # Adding the numerical values at the end of the bars
    for bar in bars:
        plt.text(bar.get_width(), bar.get_y() + bar.get_height()/2,
                 f'{bar.get_width():.4f}', va='center', fontsize=8)

    plt.show()

# Plotting feature importances with numbers for each model
plot_feature_importances(knn_importance_df, 'Feature Importance for Decision to Exercise More using KNN')
plot_feature_importances(gnb_importance_df, 'Feature Importance for Decision to Change Diet using GaussianNB')
plot_feature_importances(rf_importance_df, 'Feature Importance for Decision to Join Gym using Random Forest')
plot_feature_importances(log_reg_importance_df, 'Feature Importance for Decision to Change Diet using Logistic Regression')
```

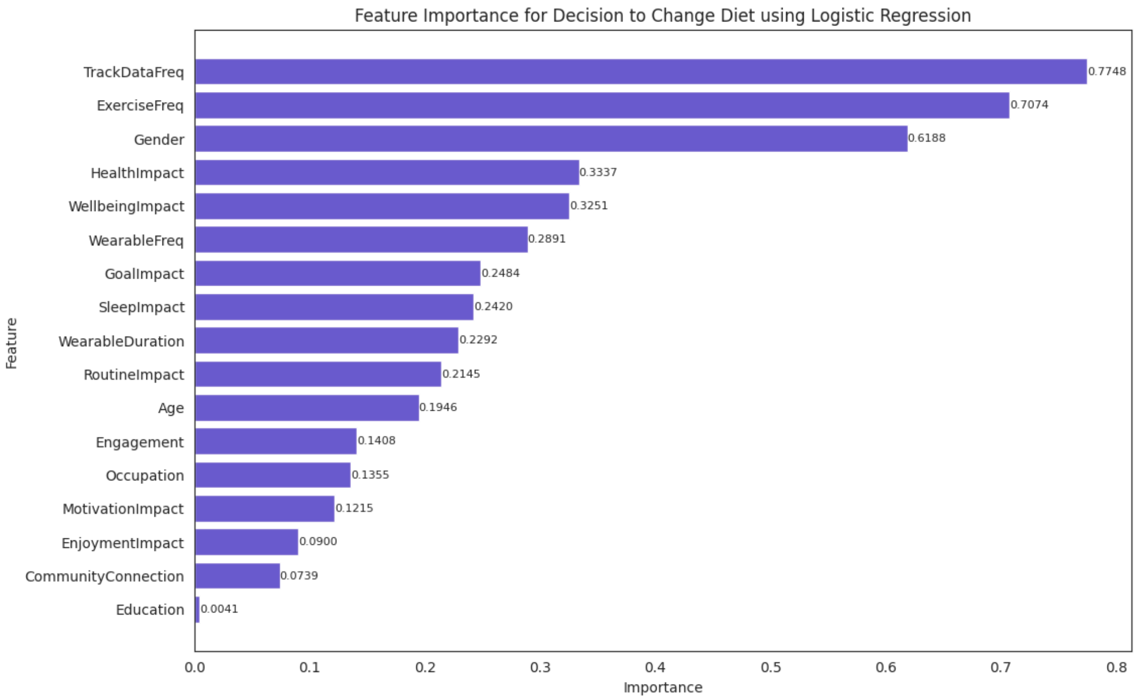


Feature Importance for Decision to Exercise More using KNN
Feature
Importance
EnjoymentImpact 0.1067
TrackDataFreq 0.0700
RoutineImpact 0.0600
CommunityConnection 0.0556
Occupation 0.0511
Engagement 0.0367
ExerciseFreq 0.0300
MotivationImpact 0.0233
WearableDuration 0.0111
Gender 0.0089
HealthImpact 0.0022
Education 0.0022
WearableFreq 0.0000
WellbeingImpact -0.0100
Age -0.0189
SleepImpact -0.0222
GoalImpact -0.0311
−0.02 0.00 0.02 0.04 0.06 0.08 0.10



Feature Importance for Decision to Change Diet using GaussianNB
Feature
Importance
TrackDataFreq 1.8929
HealthImpact 0.9643
WellbeingImpact 0.8929
Gender 0.8929
ExerciseFreq 0.8929
MotivationImpact 0.8214
RoutineImpact 0.7500
Occupation 0.6071
Age 0.5714
EnjoymentImpact 0.5000
Engagement 0.4286
WearableDuration 0.3929
GoalImpact 0.3571
WearableFreq 0.2857
SleepImpact 0.1429
Education 0.1429
CommunityConnection 0.0714
0.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75


Feature Importance for Decision to Join Gym using Random Forest
WellbeingImpact 0.1316
ExerciseFreq 0.0919
SleepImpact 0.0858
Occupation 0.0813
Age 0.0713
WearableDuration 0.0642
CommunityConnection 0.0570
Education 0.0568
RoutineImpact 0.0539
WearableFreq 0.0527
GoalImpact 0.0515
MotivationImpact 0.0419
Engagement 0.0363
HealthImpact 0.0344
TrackDataFreq 0.0335
EnjoymentImpact 0.0326
Gender 0.0232
Feature
Importance
0.00 0.02 0.04 0.06 0.08 0.10 0.12


Feature Importance for Decision to Change Diet using Logistic Regression
TrackDataFreq 0.7748
ExerciseFreq 0.7074
Gender 0.6188
HealthImpact 0.3337
WellbeingImpact 0.3251
WearableFreq 0.2891
GoalImpact 0.2484
SleepImpact 0.2420
WearableDuration 0.2292
RoutineImpact 0.2145
Age 0.1946
Engagement 0.1408
Occupation 0.1355
MotivationImpact 0.1215
EnjoymentImpact 0.0900
CommunityConnection 0.0739
Education 0.0041
Feature
Importance
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8

Top 5 Feature Importance for Decision to Exercise More using KNN

- `EnjoymentImpact` - 0.1067
- `TrackDataFreq` - 0.0700
- `RoutineImpact` - 0.0600
- `CommunityConnection` - 0.0556
- `Occupation` - 0.0511

Top 5 Feature Importance for Decision to Change Diet using GaussianNB

- `TrackDataFreq` - 1.8929
- `HealthImpact` - 0.9643
- `WellBeingImpact` - 0.8929
- `Gender` - 0.8929
- `ExerciseFreq` - 0.8929

Top 5 Feature Importance for Decision to Join Gym using Random Forest

- `WellBeingImpact` - 0.1316
- `ExerciseFreq` - 0.0919
- `SleepImpact` - 0.0858
- `Occupation` - 0.0813
- `Age` - 0.0713

Top 5 Feature Importance for Decision to Change Diet using Logistic Regression

- `TrackDataFreq` - 0.7748
- `ExerciseFreq` - 0.7074
- `Gender` - 0.6188
- `HealthImpact` - 0.3337
- `WellbeingImpact` - 0.3251

Conclusion

Visual Insights

The **demographic analysis** reveals that fitness products and services are particularly appealing to young, educated individuals, especially those who are still in school or at the early stages of their careers. To effectively reach this demographic, marketing efforts should focus on popular platforms such as social media, online fitness communities, and campus events. By doing so, fitness providers can align with the preferences and expectations of their primary consumers.

In terms of **fitness habits and tracking behavior**, the results emphasize the importance of supporting an active lifestyle and offering robust tracking features. There is a notable demand for educational content that helps new users maximize their use of wearables and data tracking. Highlighting the long-term benefits of consistent exercise and wearable usage can enhance customer retention and growth.

The **impact distribution analysis** indicates that the examined factors significantly enhance users' fitness experiences, leading to more consistent and enjoyable routines, improved health outcomes, and a stronger sense of community. These positive impacts highlight the necessity of incorporating these factors into the design of fitness programs and interventions to ensure maximum user satisfaction and effectiveness.

Lastly, the **consumer behavior analysis** suggest that addressing various aspects of fitness, including exercise, nutrition, and community support, is crucial for motivating participants to adopt healthier lifestyles and commit to their fitness goals. By focusing on these elements, fitness products and services can achieve greater user satisfaction, retention, and market success.

Key Findings and Business Impact

The Cramér's V heatmap reveals fascinating insights into user behavior and preferences. Strong associations between `GoalImpact`, `HealthImpact`, `SleepImpact`, and `WellbeingImpact` underscore the importance of a diverse approach in wellness industries. Similarly, the interplay between `RoutineImpact`, `MotivationImpact`, and `EnjoymentImpact` suggests that businesses focusing on user engagement, like fitness apps or e-learning platforms, should prioritize creating impactful routines to boost motivation and enjoyment. Interestingly, the weak associations with decision-related variables and `ExerciseFreq` highlight the complexity of user decisions and habits which calls for deeper data analysis and personalized strategies. The moderate association between occupation and gender could inform targeted marketing efforts.

These insights exemplify the power of data-driven decision-making. By using such data, businesses can move beyond guesswork, optimize resources, adopt a customer-centric approach, and gain a competitive edge. In today's data-rich environment, companies that harness these insights can craft more resonant products and services, driving growth and user satisfaction. This heatmap underscores that success in various domains, from wellness to user engagement, is multifaceted and best navigated with the compass of data analytics. By understanding the multifaceted nature of user behavior and preferences, businesses can craft more effective strategies, optimize their resources, and ultimately achieve better outcomes in terms of growth, user satisfaction, and competitive advantage.

Data Analysis Techniques

Based on the analysis of the Fitness Consumer Analysis dataset using statistical analysis `df.describe()`, several critical insights for businesses in the fitness industry has been revealed.

The data indicates a diverse customer base across age groups and genders, with varying exercise frequencies and moderate engagement with wearable fitness technology. Notably, the impact of fitness data on consumers' routines is generally positive, suggesting that data-driven insights can significantly influence behavior.

Moreover, fitness activities show a moderate to high impact on motivation, enjoyment, health, and overall wellbeing, underscoring the need for engaging and holistic fitness experiences. These findings have substantial business implications, from product development opportunities in wearable tech to refining marketing strategies that emphasize holistic benefits.

The data also points to potential for improved customer retention through personalized insights, market segmentation based on diverse fitness habits, and cross-selling opportunities in fitness gear and nutrition. Interestingly, despite low community connection scores, high engagement and motivation metrics suggest untapped potential in community-building features.

Most importantly, this dataset underscores the indispensable value of data-driven decision-making in the fitness industry. By leveraging such data, businesses can optimize resource allocation, personalize user experiences, identify growth opportunities, measure and improve initiatives, and gain a competitive edge.

In an industry as personal and varied as fitness, the ability to make data-driven decisions can be the difference between a good business and a great one, driving growth, retention, and profitability.

Based on the analysis of the Fitness Consumer Analysis dataset using `Analysis of Variance (ANOVA)`, several key insights have been derived.

The dataset's p-values indicate that routine and motivation consistently show significant or near-significant impacts across various fitness-related decisions which highlights their critical importance. Community connection and enjoyment significantly influence the decision to exercise more, emphasizing the value of a supportive and enjoyable fitness environment. Additionally, the frequency of wearable technology usage and data tracking demonstrates significant impacts which suggests that integrating technology and data tracking into fitness programs can positively affect fitness decisions.

These insights can significantly impact decision-making processes within a fitness-related business or organization. By understanding the limited influence of demographic factors like age and gender, businesses can create more targeted and effective marketing campaigns that focus on motivational and technological aspects instead. Insights into the importance of wearable technology and exercise frequency can guide product development, leading to the creation of fitness solutions that integrate with users' lifestyles. Emphasizing community connection and enjoyment in fitness programs can enhance user engagement which lead to higher customer satisfaction and loyalty.

Using these insights for strategic planning ensures decisions are based on empirical evidence, leading to more effective outcomes. Understanding the key factors influencing fitness decisions helps create offerings to meet customer needs, improving overall business performance. In conclusion, the p-value analysis of the Fitness Consumer Analysis dataset underscores the importance of data-driven decision-making in the fitness industry. By focusing on factors that truly influence consumer behavior, businesses can develop strategies that enhance user experience, drive engagement, and ultimately achieve better business outcomes.

Implementation of Machine Learning

The implementation of machine learning reveals that different machine learning models perform best for various prediction tasks related to fitness decisions.

For the **decision to exercise more**, the `K-Nearest Neighbor (KNN)` model stands out with an accuracy of 66.67%, precision of 72.22%, recall of 66.67%, and an F1 score of 65.48%, surpassing other models. Higher accuracy means the model correctly predicts more instances overall, higher precision indicates fewer false positive errors, higher recall means the model identifies more actual positive instances, and a higher F1 score shows a better balance between precision and recall.

In predicting the **decision to buy products**, the `Gaussian Naive Bayes` model excels, achieving the highest precision of 91.67% and an F1 score of 85.19%, with an accuracy of 83.33%, alongside other top-performing models. These metrics indicate the model's effectiveness in making correct positive predictions and maintaining a strong balance between precision and recall.

For the **decision to join a gym**, `Random Forest`, `Gradient Boosting`, and `Support Vector Machine` models perform equally well, each achieving an accuracy of 83.33%, precision of 100.00%, recall of 83.33%, and an F1 score of 88.89%. These high metrics suggest the models are reliable, make accurate positive predictions, and effectively identify positive instances.

Lastly, for the **decision to change diet**, both `Logistic Regression` and `K-Nearest Neighbor (KNN)` models are highly effective, each with an accuracy of 83.33%, precision of 87.50%, recall of 83.33%, and an F1 score of 82.86%, indicating great overall performance and balance between precision and recall.

These insights show the importance of selecting the appropriate machine learning model for specific prediction tasks to achieve optimal performance. By using these data-driven insights, fitness providers can make informed decisions to modify their products and services more effectively, enhancing user satisfaction and engagement. For instance, understanding which model best predicts the decision to buy products can help in targeted marketing strategies, while insights into gym membership decisions can inform membership retention programs.

Advanced Analysis

The advanced analysis highlights the critical features influencing various fitness-related decisions, with each model revealing specific insights. For the **decision to exercise more using the K-Nearest Neighbor (KNN)** model, the most important features are EnjoymentImpact (0.1067), TrackDataFreq (0.0700), RoutineImpact (0.0600), CommunityConnection (0.0556), and Occupation (0.0511). These insights suggest that enhancing the enjoyment of fitness activities, providing frequent data tracking, positively impacting routines, fostering community connections, and considering the user's occupation are crucial in encouraging more exercise.

For the **decision to change diet using the Gaussian Naive Bayes** model, the key features are TrackDataFreq (1.8929), HealthImpact (0.9643), WellBeingImpact (0.8929), Gender (0.8929), and ExerciseFreq (0.8929). These results indicate that frequent tracking of data, emphasizing health and well-being impacts, considering gender differences, and encouraging regular exercise are significant factors in influencing dietary changes.

The **decision to join a gym using the Random Forest** model is primarily influenced by WellBeingImpact (0.1316), ExerciseFreq (0.0919), SleepImpact (0.0858), Occupation (0.0813), and Age (0.0713). This suggests that promoting the well-being benefits of gym membership, regular exercise, improved sleep, and targeting specific occupations and age groups can effectively drive gym membership.

Lastly, for the **decision to change diet using the Logistic Regression** model, the important features are TrackDataFreq (0.7748), ExerciseFreq (0.7074), Gender (0.6188), HealthImpact (0.3337), and WellBeingImpact (0.3251). These insights emphasize the importance of data tracking, regular exercise, considering gender-specific approaches, and highlighting health and well-being impacts in dietary decisions.

By understanding which features significantly influence fitness-related decisions, businesses can design more targeted marketing strategies, develop personalized fitness programs, and enhance user satisfaction and retention. For instance, promoting the enjoyment and routine impacts of exercise can encourage more people to exercise, while emphasizing the health benefits and data tracking features can influence dietary changes.