Business Problem

The market research team at AeroFit wants to identify the characteristics of the target audience for each type of treadmill offered by the company, to provide a better recommendation of the treadmills to the new customers. The team decides to investigate whether there are differences across the product with respect to customer characteristics.

Perform descriptive analytics to create a customer profile for each AeroFit treadmill product by developing appropriate tables and charts. For each AeroFit treadmill product, construct two-way contingency tables and compute all conditional and marginal probabilities along with their insights/impact on the business.

Importing the dataset

In [84]:
```
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749"
```

```
--2024-04-14 17:43:46--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.238.92.21, 18.238.92.63, 18.238.92.172, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.238.92.21|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7279 (7.1K) [text/plain]
Saving to: 'aerofit_treadmill.csv?1639992749.1'

aerofit_treadmill.c 100%[===================>]   7.11K  --.-KB/s    in 0s

2024-04-14 17:43:46 (1.79 GB/s) - 'aerofit_treadmill.csv?1639992749.1' saved [7279/7279]
```

Importing the necessary libraries

In [85]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1. Import the dataset and doing usual Exploratory analysis on aerofit_treadmill.csv

Reading the dataset

In [86]:
```
df = pd.read_csv("aerofit_treadmill.csv?1639992749")
df.head()
```

Out[86]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

Checking for null values

In [87]:
```
df.isnull().sum()
```

Out[87]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

Finding the number of rows and columns in a given dataset

In [88]:
```
df.shape
```

Out[88]:
```
(180, 9)
```

Observing the datatype of each column

In [89]:
```
df.dtypes
```

Out[89]:
```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

Finding value counts on categorical data

In [90]:
```
df.Product.value_counts()
```

Out[90]:
```
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
```

In [91]:
```
df.Gender.value_counts()
```

Out[91]:
```
Gender
Male      104
Female     76
Name: count, dtype: int64
```

In [92]:
```
df.MaritalStatus.value_counts()
```

Out[92]:
```
MaritalStatus
Partnered    107
Single        73
Name: count, dtype: int64
```

Finding the Unique values

In [93]:
```
df.Fitness.unique()
```

Out[93]:
```
array([4, 3, 2, 1, 5])
```

Observation

- Customers have rated there fitness ranging from 1 to 5 where 1 being poor and 5 being good.

In [94]:
```
df.Education.unique()
```

Out[94]:
```
array([14, 15, 12, 13, 16, 18, 20, 21])
```

Observation

- Customers and their years of education
- We have wide range of customers with years of education ranginng from 12 to 21

In [95]:
```
df.Usage.unique()
```

Out[95]:
```
array([3, 2, 4, 5, 6, 7])
```

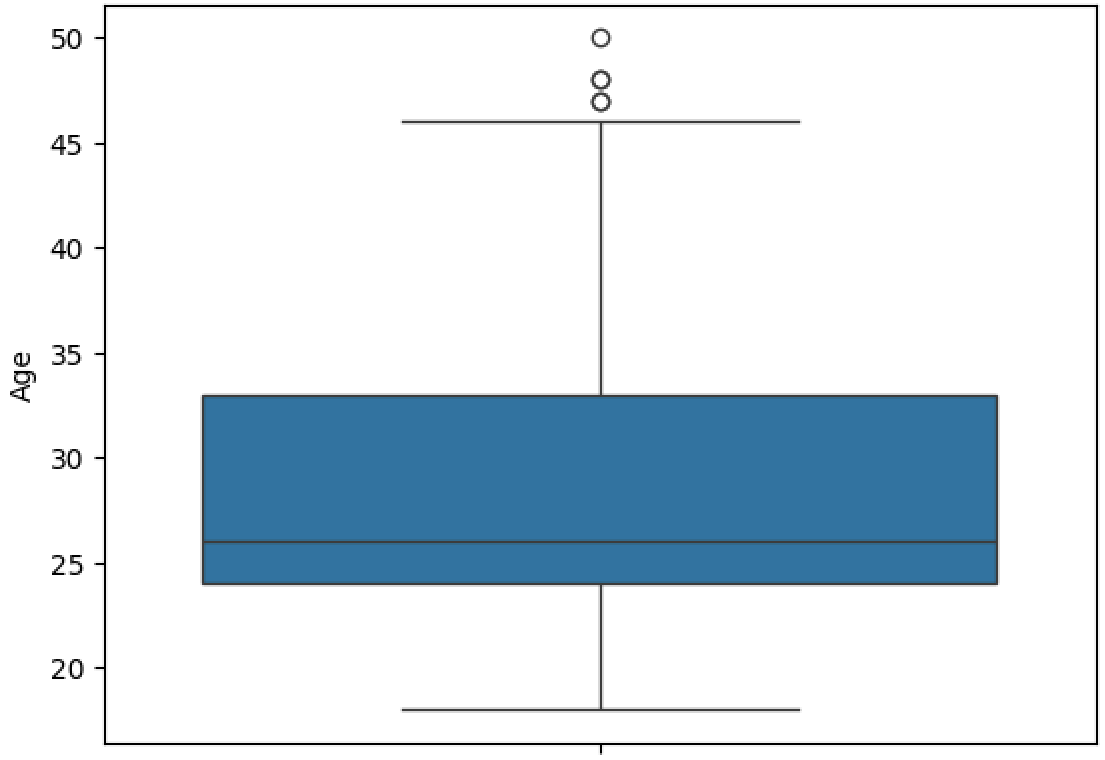
Observation

- We have Customers who plan to use treadmill from 2 times a week to 7 times a week

2. Identifying the Outliers for every continuous data

Observing Age feature

In [96]:
```
sns.boxplot(data = df,y = "Age")
plt.show()
```

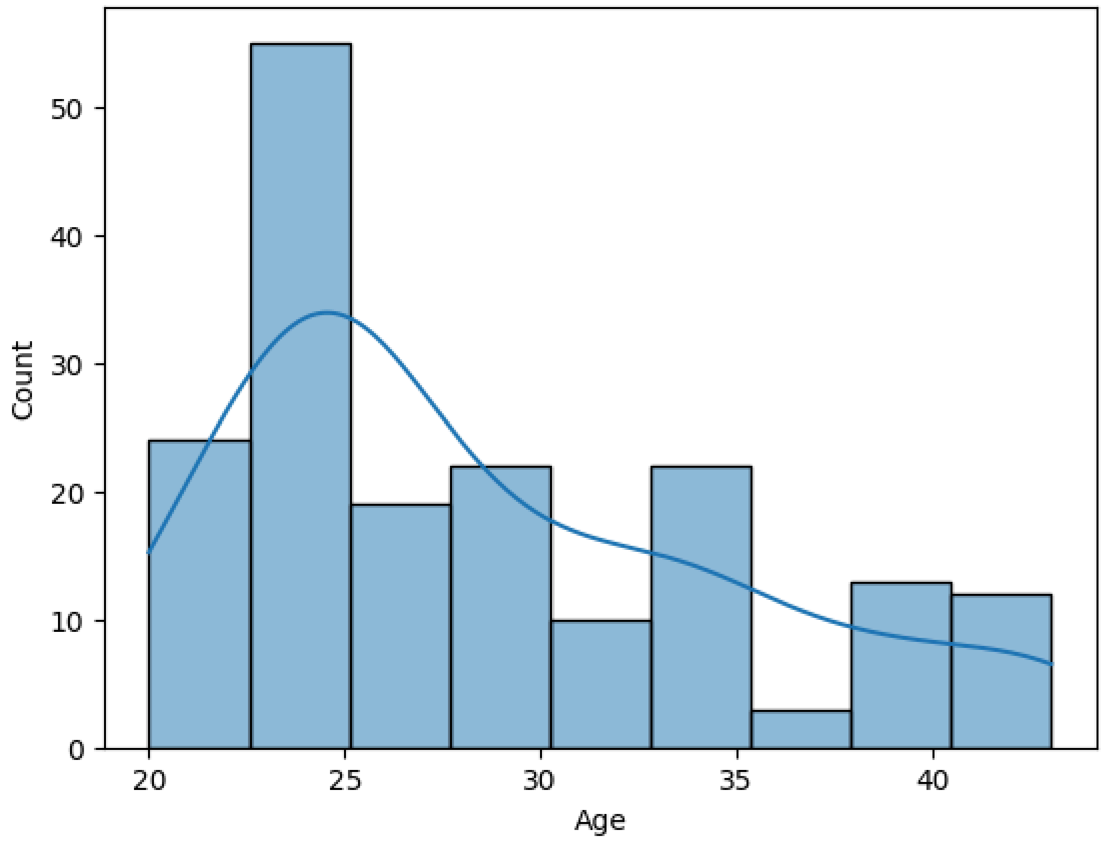


Observation

- Customers of age around **46** and above are **outliers**.

Clipping the age between 5 th and 95 th percentile and describing

In [97]:
```
# values at 5th and 95th percentiles
percentile_5 = df['Age'].quantile(0.05)
percentile_95 = df['Age'].quantile(0.95)
# clipping the values in between 5 th and 95 th percentile
clipped_values = np.clip(df['Age'], percentile_5, percentile_95)
clipped_Age = pd.DataFrame(clipped_values,columns = ['Age'])

sns.histplot(data = clipped_Age ,x = "Age",kde = True)
plt.show()
```



In [98]:
```
clipped_Age['Age'].describe()
```

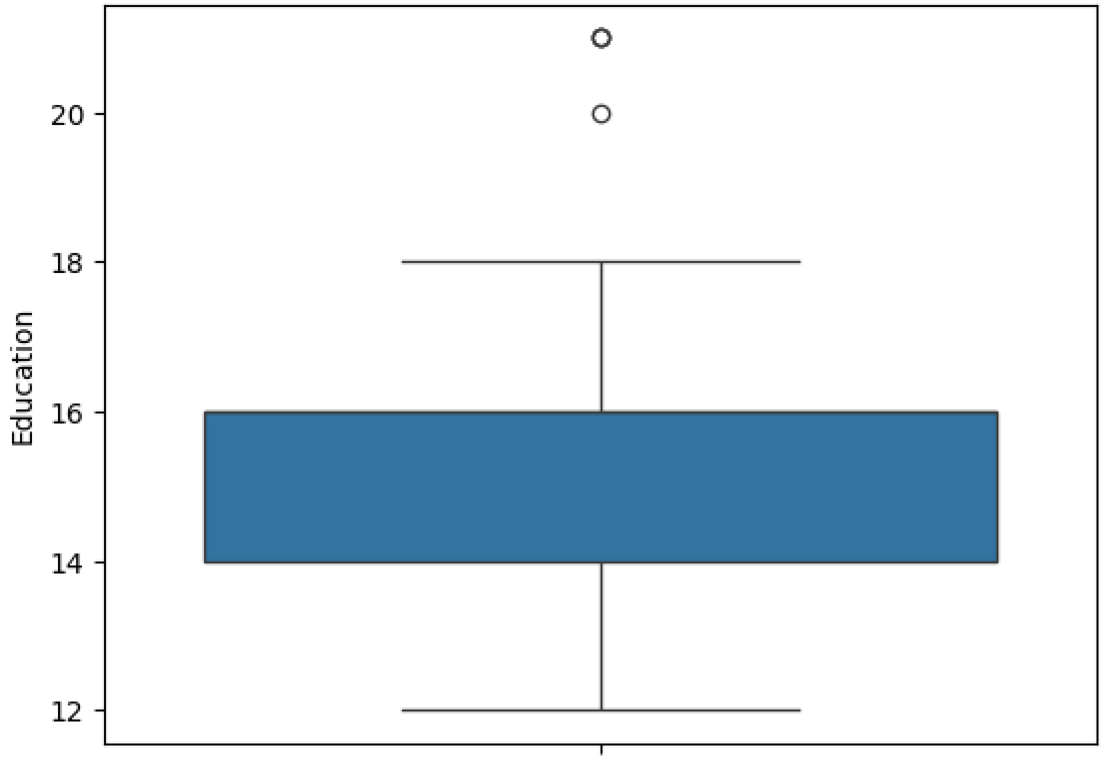
Out[98]:
```
count    180.000000
mean      28.641389
std        6.446373
min       20.000000
25%       24.000000
50%       26.000000
75%       33.000000
max       43.050000
Name: Age, dtype: float64
```

Observation

- In a sample of 180 customers **mean age** of customers is **28.641389** with **+/- 6.446373** of **standard deviation**

Observing Education feature

In [99]:
```
sns.boxplot(data = df,y="Education")
plt.show()
```



Observation

- We can notice that in and around **20** years we have few **outliers**

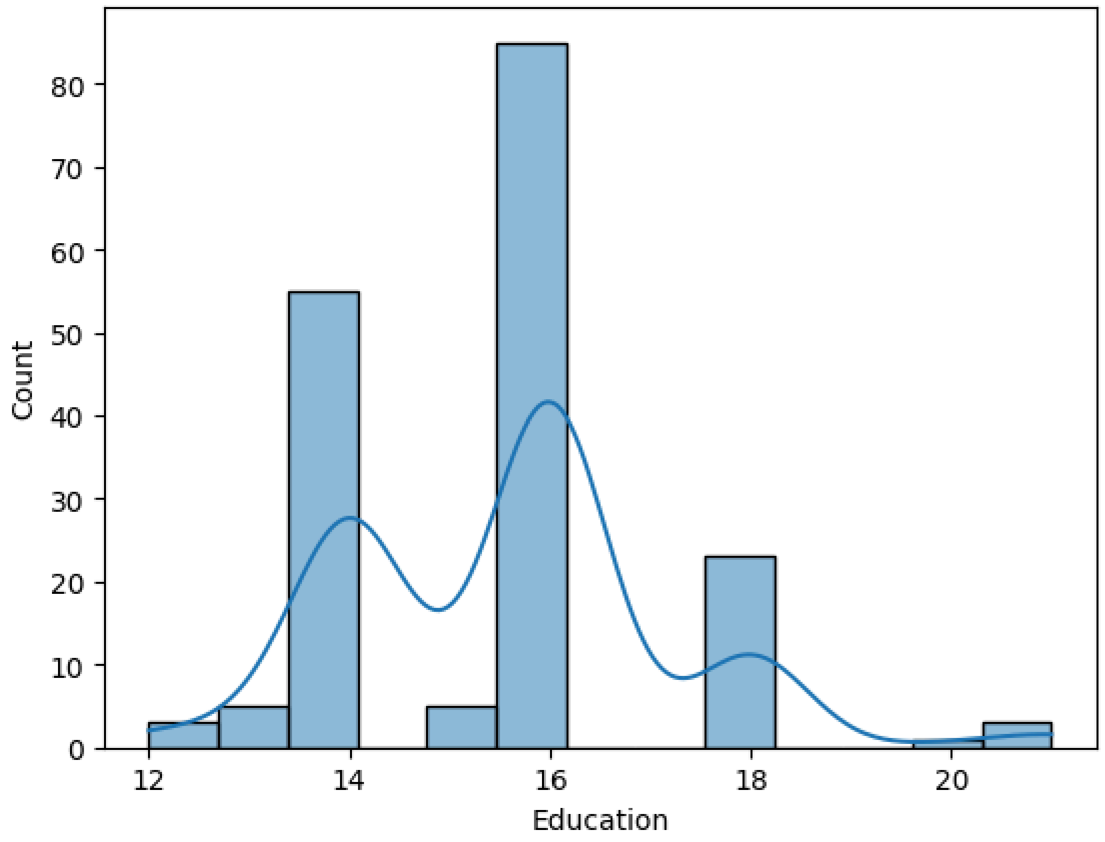
Clipping the data between 5 th and 95 th percentile and describing

In [100…
```
percentile_5 = df['Education'].quantile(0.05)
percentile_95 = df['Education'].quantile(0.95)

# Clip the DataFrame to keep values within the 5th and 95th percentiles
clipped_values = np.clip(df['Education'], percentile_5, percentile_95)

clipped_edu = pd.DataFrame(clipped_values, columns=['Education'])
```

In [101…
```
clipped_edu['Education'].describe()
```

Out[101]:
```
count    180.000000
mean      15.572222
std        1.362017
min       14.000000
25%       14.000000
50%       16.000000
75%       16.000000
max       18.000000
Name: Education, dtype: float64
```

In [102…
```
sns.histplot(data = df,x = "Education",kde =True)
plt.show()
```



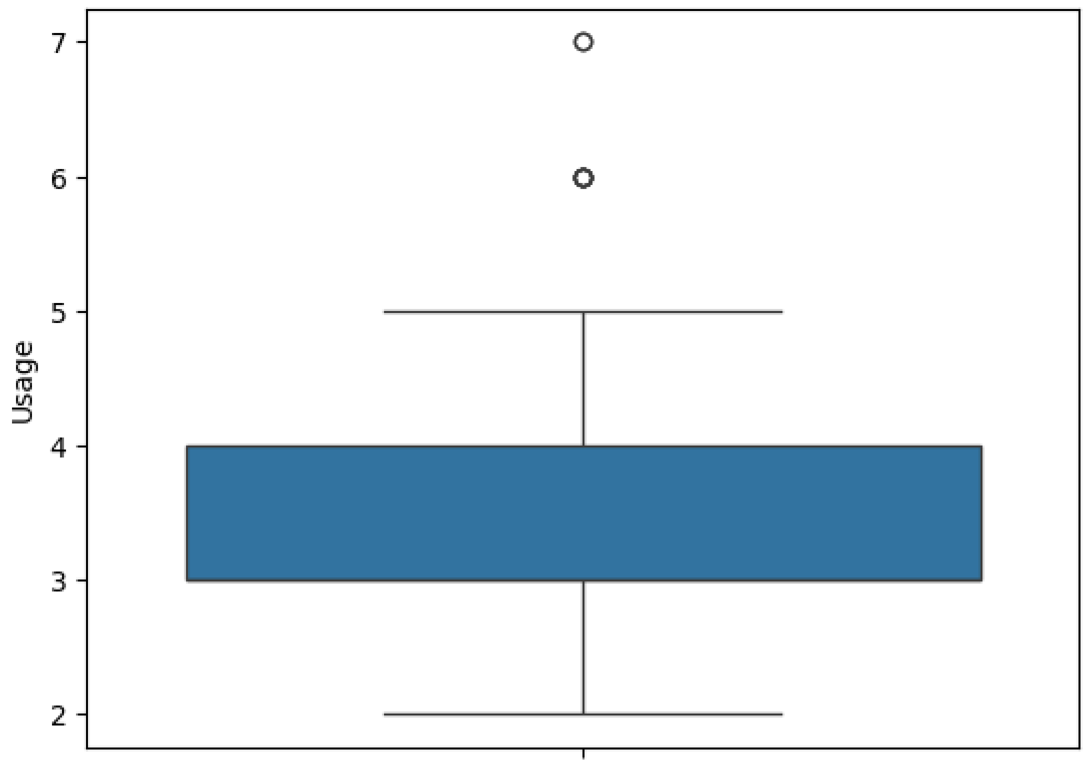
Observation

- Among 180 customers **mean of years of education** is around **15.572222** with a standard deviation of around +/- **1.362017**

Observing Usage feature - Tells us about the average number of times the customer plans to use the treadmill each week.

In [103…

```
sns.boxplot(data = df, y = "Usage")
plt.show()
```



Observation

- Users that use treadmill for an average of 6 or 7 times in a week are considered outliers

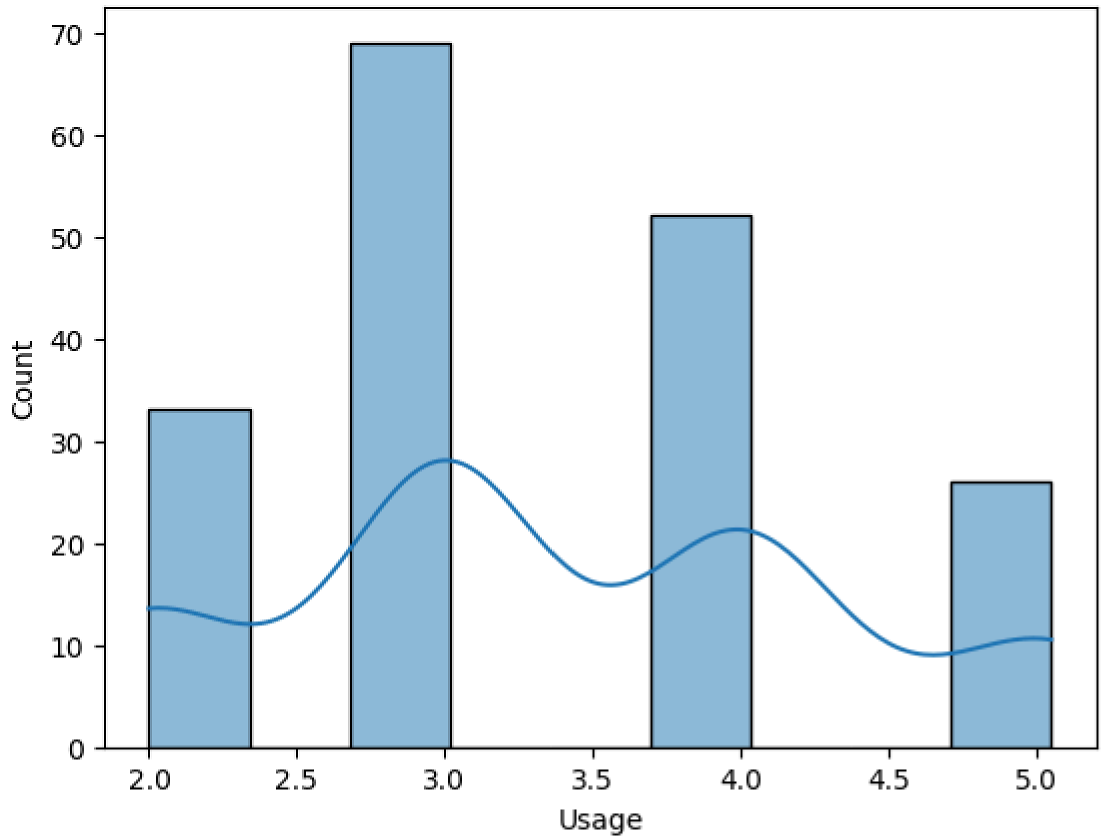
Clipping the data between 5 th and 95 th percentile and describing

In [104…
```
percentile_5 = df['Usage'].quantile(0.05)
percentile_95 = df['Usage'].quantile(0.95)

# Clip the DataFrame to keep values within the 5th and 95th percentiles
clipped_values = np.clip(df['Usage'], percentile_5, percentile_95)

clipped_usage = pd.DataFrame(clipped_values, columns=['Usage'])
```

In [105…
```
clipped_usage['Usage'].describe()
```

Out[105]:
```
count    180.000000
mean       3.396944
std        0.952682
min        2.000000
25%        3.000000
50%        3.000000
75%        4.000000
max        5.050000
Name: Usage, dtype: float64
```

In [106…
```
sns.histplot(data = clipped_usage, x = "Usage",kde =True)
plt.show()
```



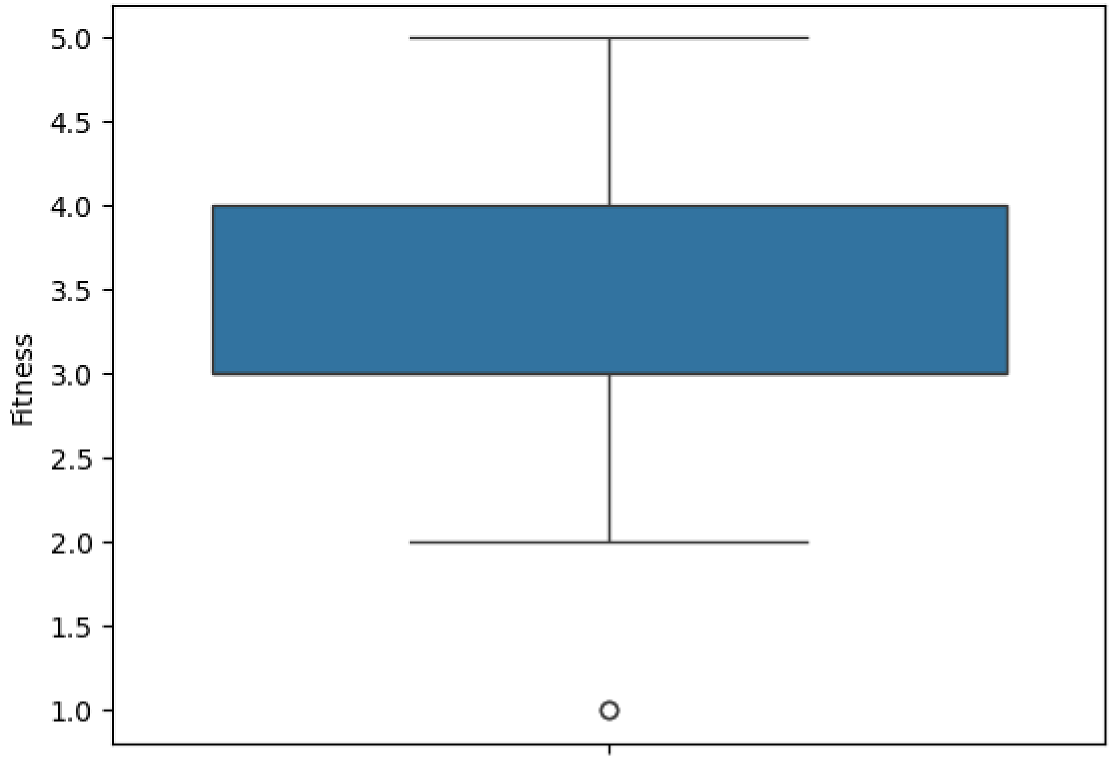
Observation

- Out of 180 customers most of them plan to use treadmill on an average 3 times in a week
- Max and min will be around 5 and 2 times per week respectively

Observing Fitness feature

In [107…

```
sns.boxplot(data = df, y = 'Fitness')
plt.show()
```



Observation

- Out of all the 180 customers there is only one customer with extremely poor fitness level that is around 1 and its considered as an outlier

Clipping the data between 5 th and 95 th percentile and describing the fitness level

In [108…

```
percentile_5 = df['Fitness'].quantile(0.05)
percentile_95 = df['Fitness'].quantile(0.95)

# Clip the DataFrame to keep values within the 5th and 95th percentiles
clipped_values = np.clip(df['Fitness'], percentile_5, percentile_95)

clipped_fitness = pd.DataFrame(clipped_values, columns=['Fitness'])
```

In [109…

```
clipped_fitness['Fitness'].describe()
```

Out[109]:

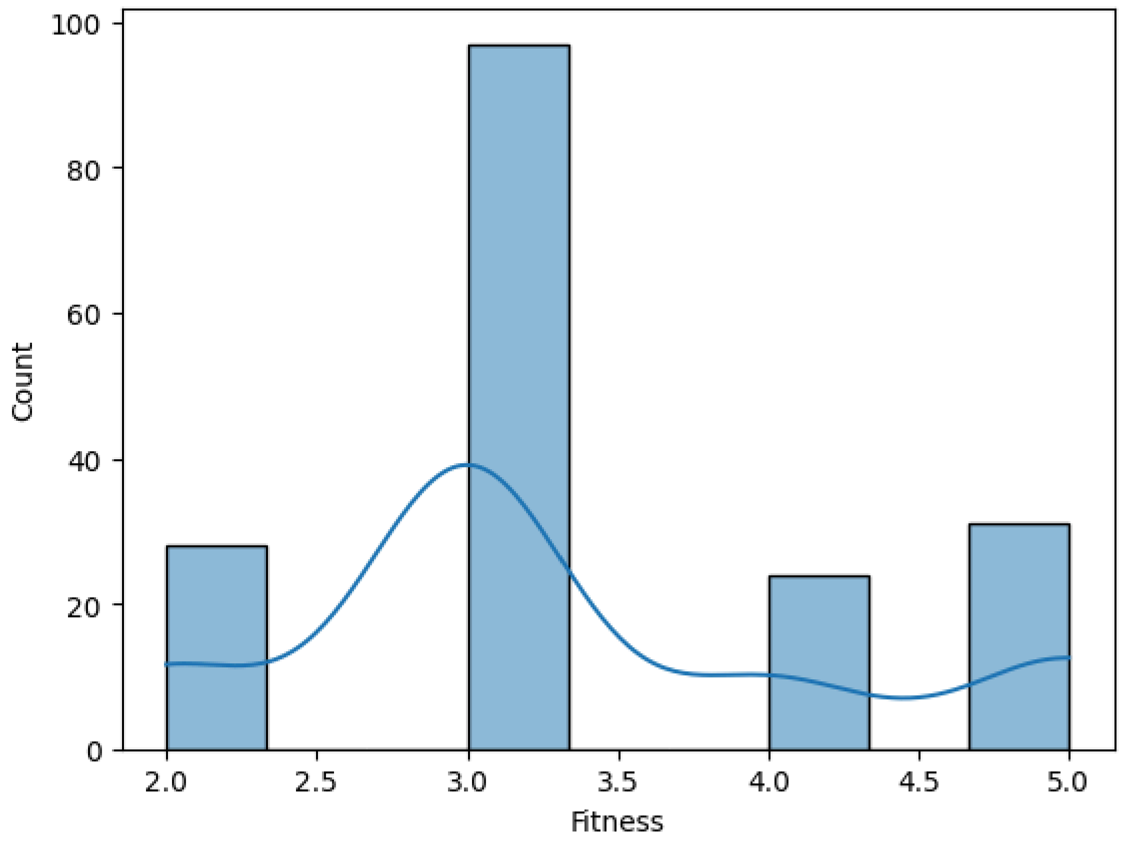
```
count    180.000000
mean       3.322222
std        0.937461
min        2.000000
25%        3.000000
50%        3.000000
75%        4.000000
max        5.000000
Name: Fitness, dtype: float64
```

Observation

- Mean fitness level of 180 customers is around **3.322222** with a **std of +/- 0.937461**
- Majority of customers fall within the range of moderate to good fitness levels, with **25%** of customers having a fitness level of **3** or below

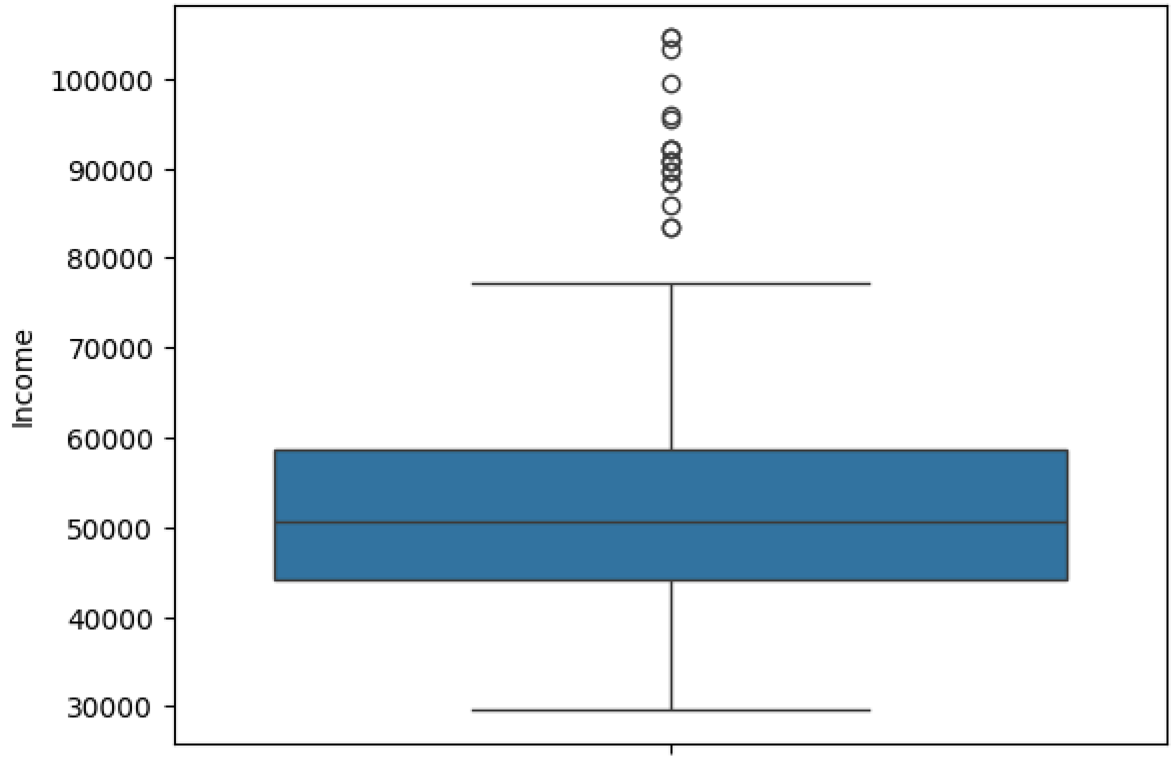
- **75%** having a fitness level of **4** or above

In [110…
```
sns.histplot(data = clipped_fitness, x = 'Fitness',kde = True)
plt.show()
```



Observing the Income feature

In [111…
```
sns.boxplot(data = df, y = 'Income')
plt.show()
```



Observation

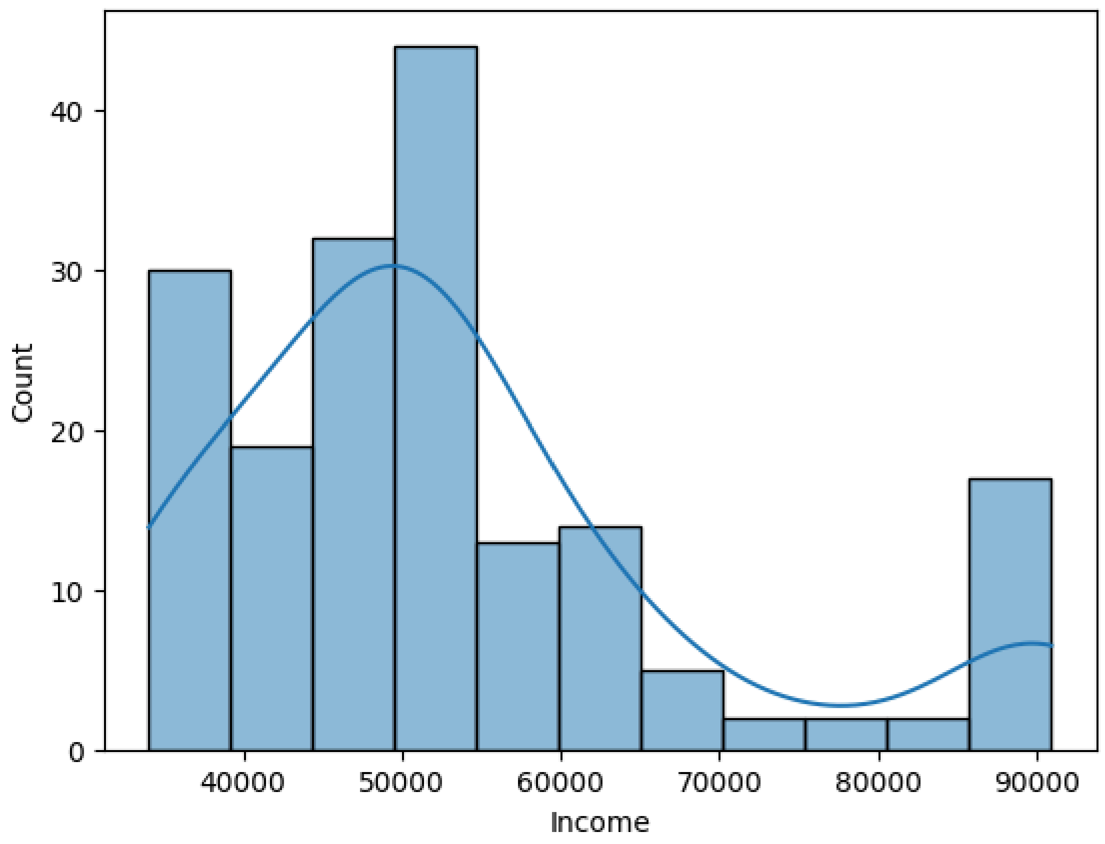
- We have noticeable amount of outliers whose salary is greater than 78000

Clipping the income between 5 th and 95 th percentile and describing

In [112…
```
percentile_5 = df['Income'].quantile(0.05)
percentile_95 = df['Income'].quantile(0.95)


# Clip the DataFrame to keep values within the 5th and 95th percentiles
clipped_values = np.clip(df['Income'], percentile_5, percentile_95)

clipped_income = pd.DataFrame(clipped_values, columns=['Income'])
clipped_income['Income'].describe()
```

Out[112]:
```
count      180.000000
mean     53477.070000
std      15463.662523
min      34053.150000
25%      44058.750000
50%      50596.500000
75%      58668.000000
max      90948.250000
Name: Income, dtype: float64
```

In [113…
```
sns.histplot(data = clipped_income, x = 'Income',kde = True)
plt.show()
```



Observation

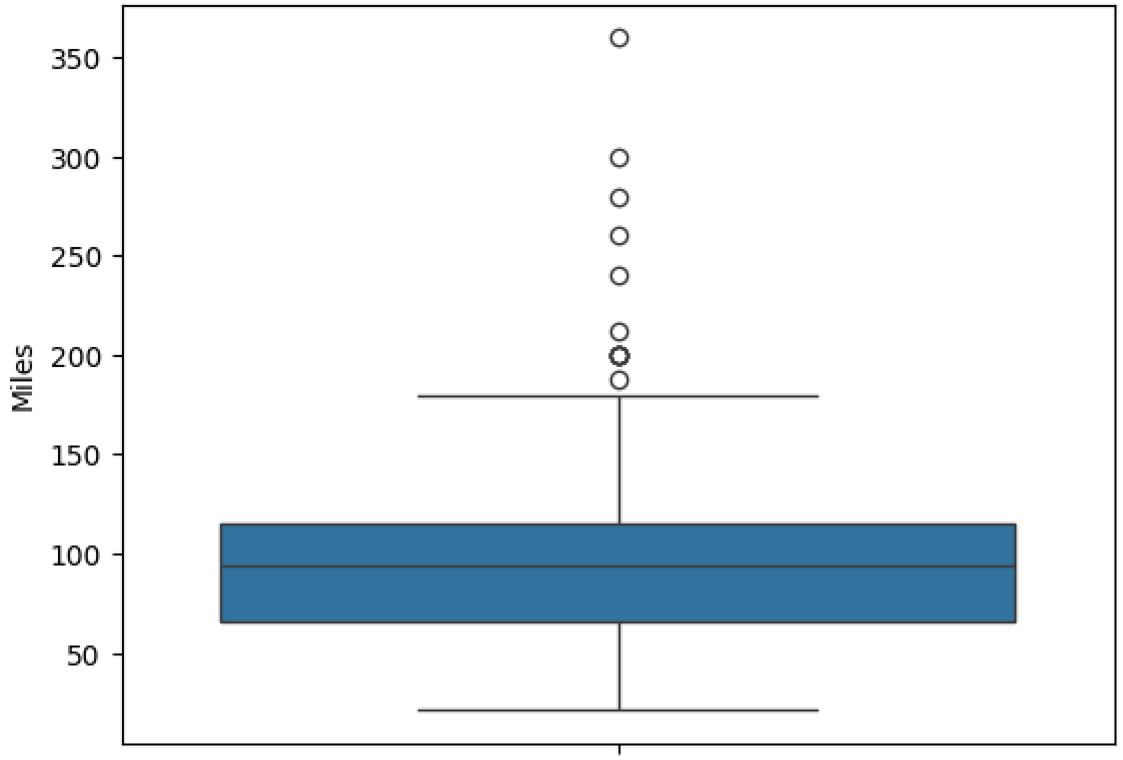
- **Mean** salary is around **53477.07** with a **std of +/- 15463.66**
- 25% of customers have income of around **44058.75** and below.
- 75 % of customers have income of around **58668** and above,
- Customers whose income is greater than **78000** are considered **outliers**
- Upon clipping there is a subset of customers with above-average incomes within the dataset, and while the clipping reduced the presence of extreme outliers, it did not eliminate the presence of relatively high-income individuals.

Observing Miles feature

- This feature talks about the average number of miles the customer expects to walk/run each week

In [114…

```
sns.boxplot(data = df, y = 'Miles')
plt.show()
```



Observation

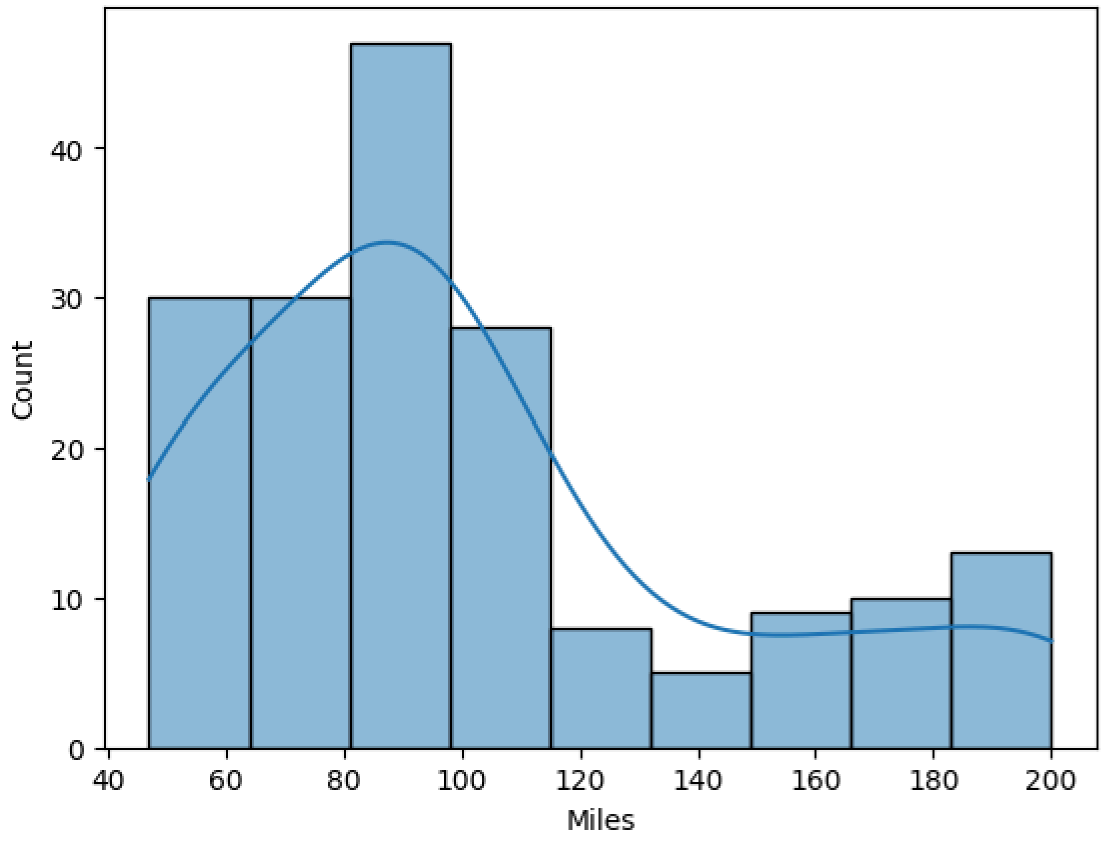
- Customers expecting to walk above **180 miles** per week are considered outliers

Clipping the Miles data between 5 th and 95 th percentile and describing

In [115…
```
percentile_5 = df['Miles'].quantile(0.05)
percentile_95 = df['Miles'].quantile(0.95)

# Clip the DataFrame to keep values within the 5th and 95th percentiles
clipped_values = np.clip(df['Miles'], percentile_5, percentile_95)

clipped_miles = pd.DataFrame(clipped_values, columns=['Miles'])
clipped_miles['Miles'].describe()
```

Out[115]:
```
count    180.000000
mean     101.088889
std       43.364286
min       47.000000
25%       66.000000
50%       94.000000
75%      114.750000
max      200.000000
Name: Miles, dtype: float64
```

In [116…
```
sns.histplot(data = clipped_miles, x = 'Miles',kde = True)
plt.show()
```



Observation

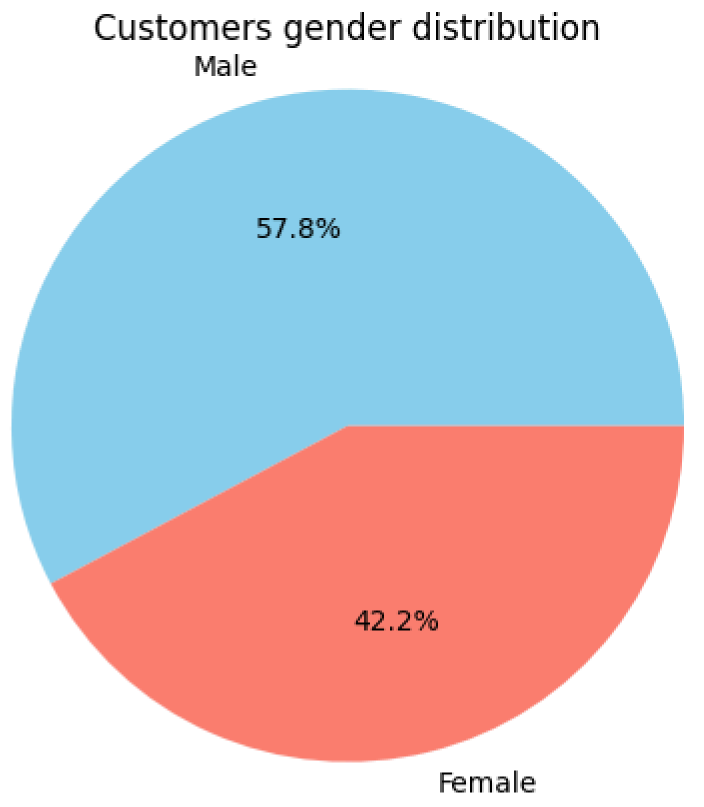
- So **50%** of the customers expect to walk **94 miles** per week
- **25%** of the customers expect to walk **66 miles** per week
- **75%** of customers expect to walk **114.75** miles per week
- On an **average** 180 customers expect to walk **101 miles** per week
- **Min** and **max** miles are **47** and **200** miles respectively

3. Check if features like Gender, Marital status and Age have any effect on the product purchased

Understanding the customers profile

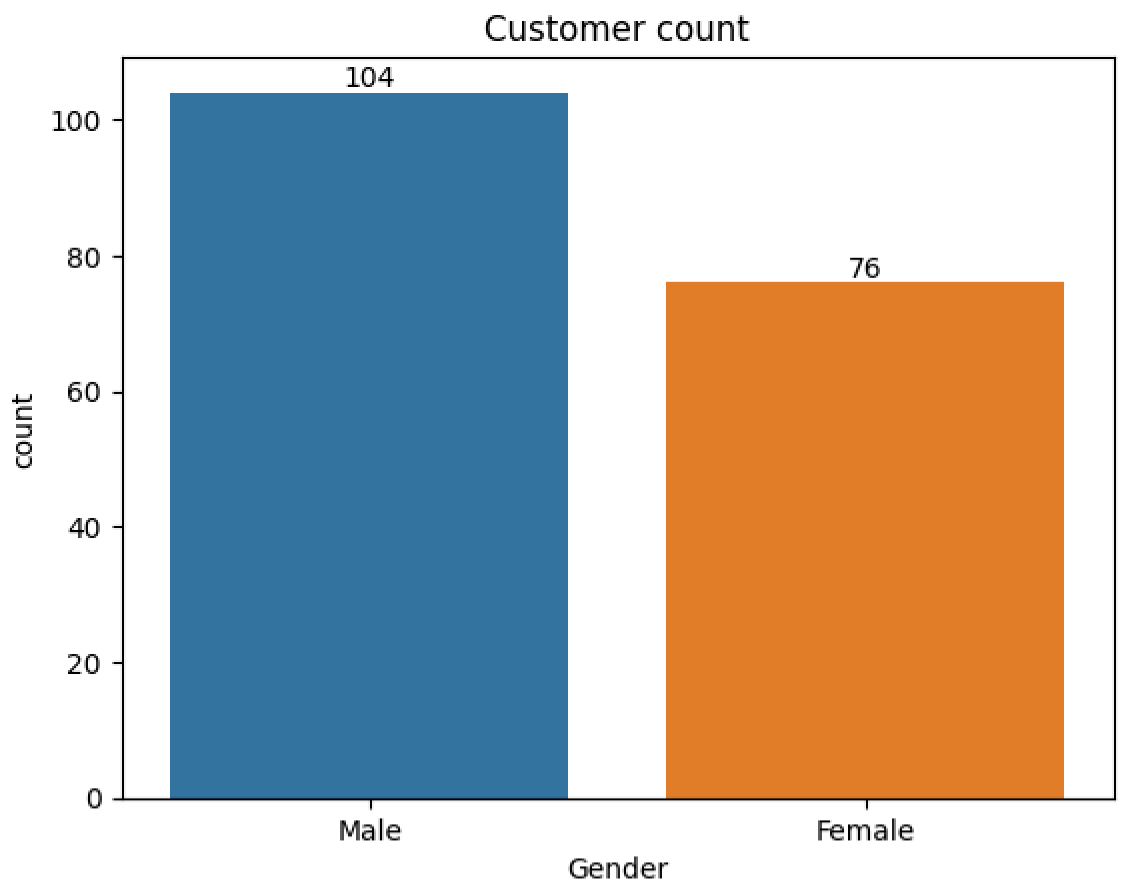
In [117…

```
gender_counts = df['Gender'].value_counts()
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', colors=['skyblue', 'salmon'])
plt.title('Customers gender distribution')
plt.axis('equal')
plt.show()
```



In [118…

```
ax1 = sns.countplot(data = df , x = 'Gender', hue = 'Gender' , legend = False)
# Loop through each container (bar group) in the countplot and Add count labels to each bar within the current container
for container in ax1.containers:
  ax1.bar_label(container)
plt.title("Customer count")
plt.show()
```



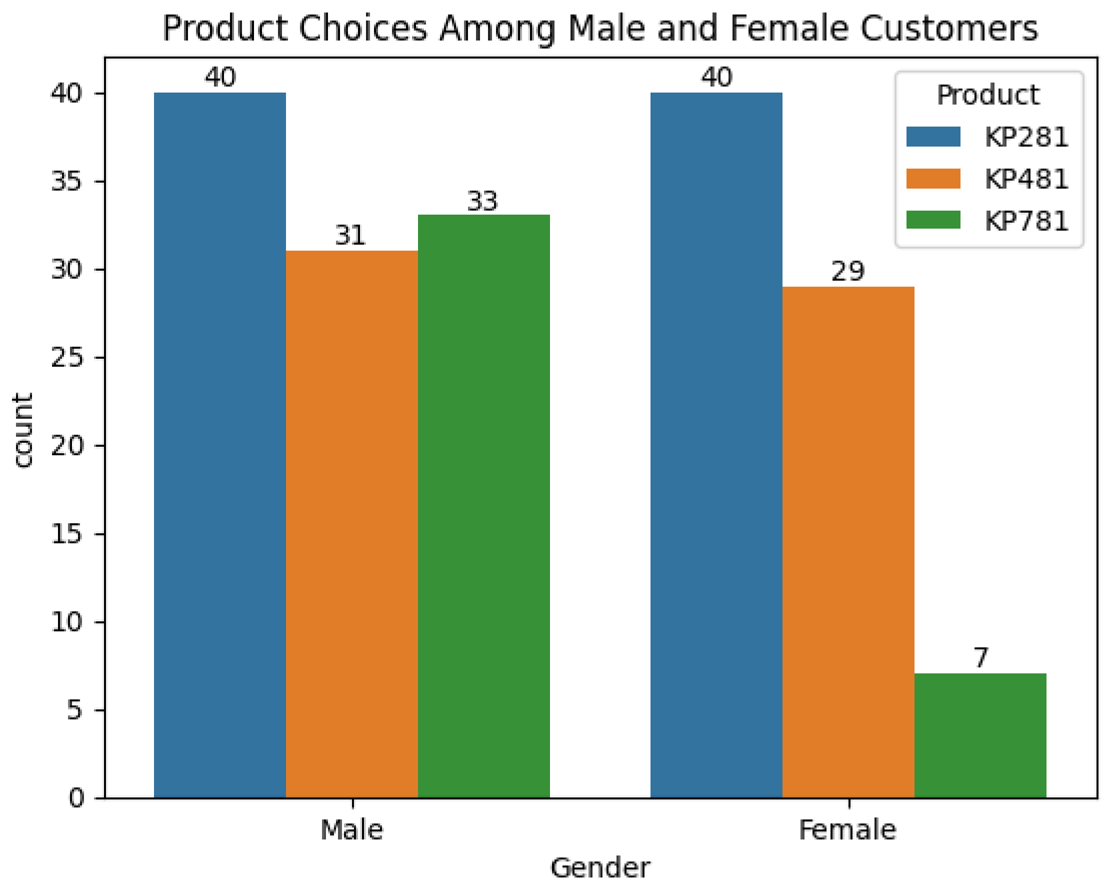
Observation

- Looking at the above two plot we can notice there are around **104** Male customers and **76** Female customers
- From pie plot we can infer that female percentage is comparitively less than males **57.8% and 42.2%** respectively

Understanding their product choices

In [119…

```
ax1  = sns.countplot(data = df , x='Gender', hue = 'Product')

# Loop through each container (bar group) in the countplot and Add count labels to each bar within the current container
for container in ax1.containers:
  ax1.bar_label(container)

plt.title('Product Choices Among Male and Female Customers')
plt.show()
```

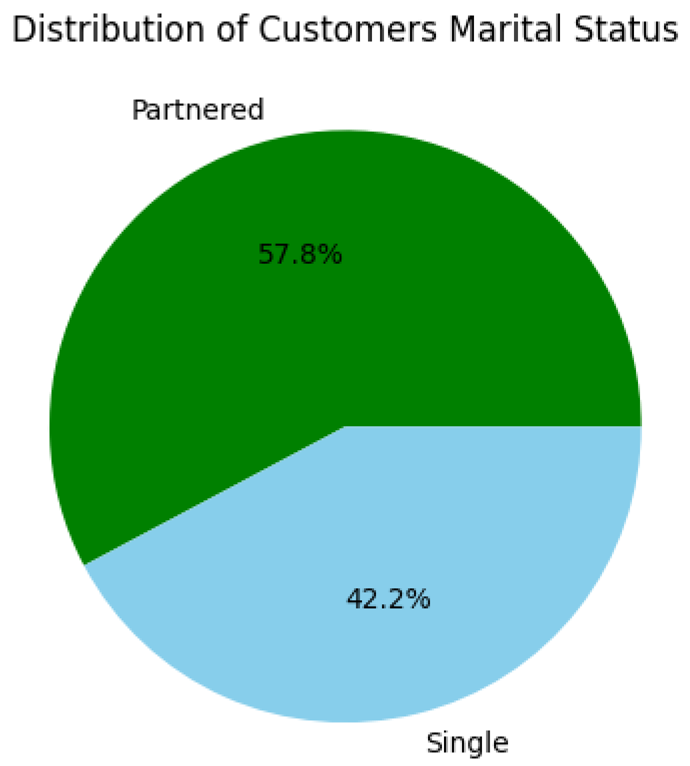


Observation

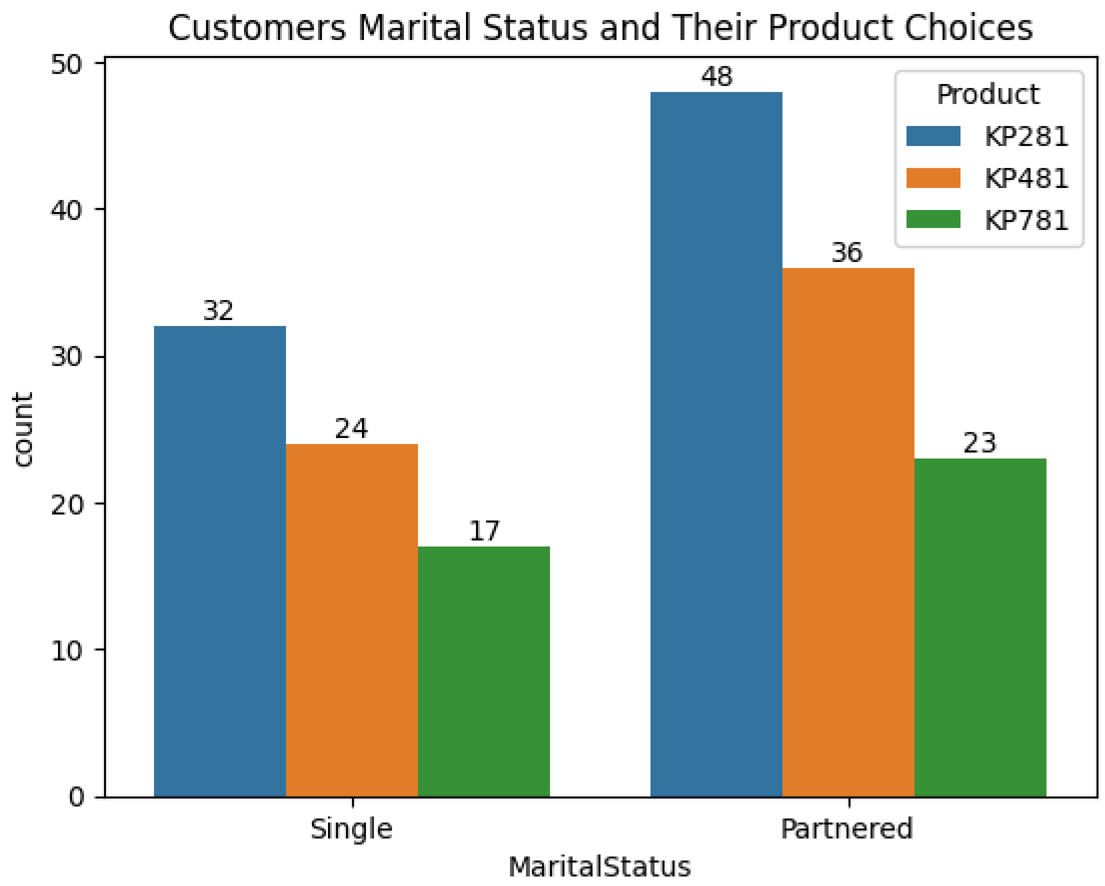
- It's evident that irrespective of gender both customers are more likely to buy KP281
- Likeliness to buy KP481 is mostly same
- Whereas Male customers tend to buy more KP781 than Females

Understanding the Marital status of customers

In [120…
```
marital_status_counts = df['MaritalStatus'].value_counts()
plt.pie(gender_counts, labels=marital_status_counts.index, autopct='%1.1f%%', colors=['green', 'skyblue'])
plt.title('Distribution of Customers Marital Status')
plt.show()
```



In [121…
```
ax2 = sns.countplot(data = df , x = 'MaritalStatus', hue = 'Product')

for container in ax2.containers:
  ax2.bar_label(container)

plt.title("Customers Marital Status and Their Product Choices")
plt.show()
```

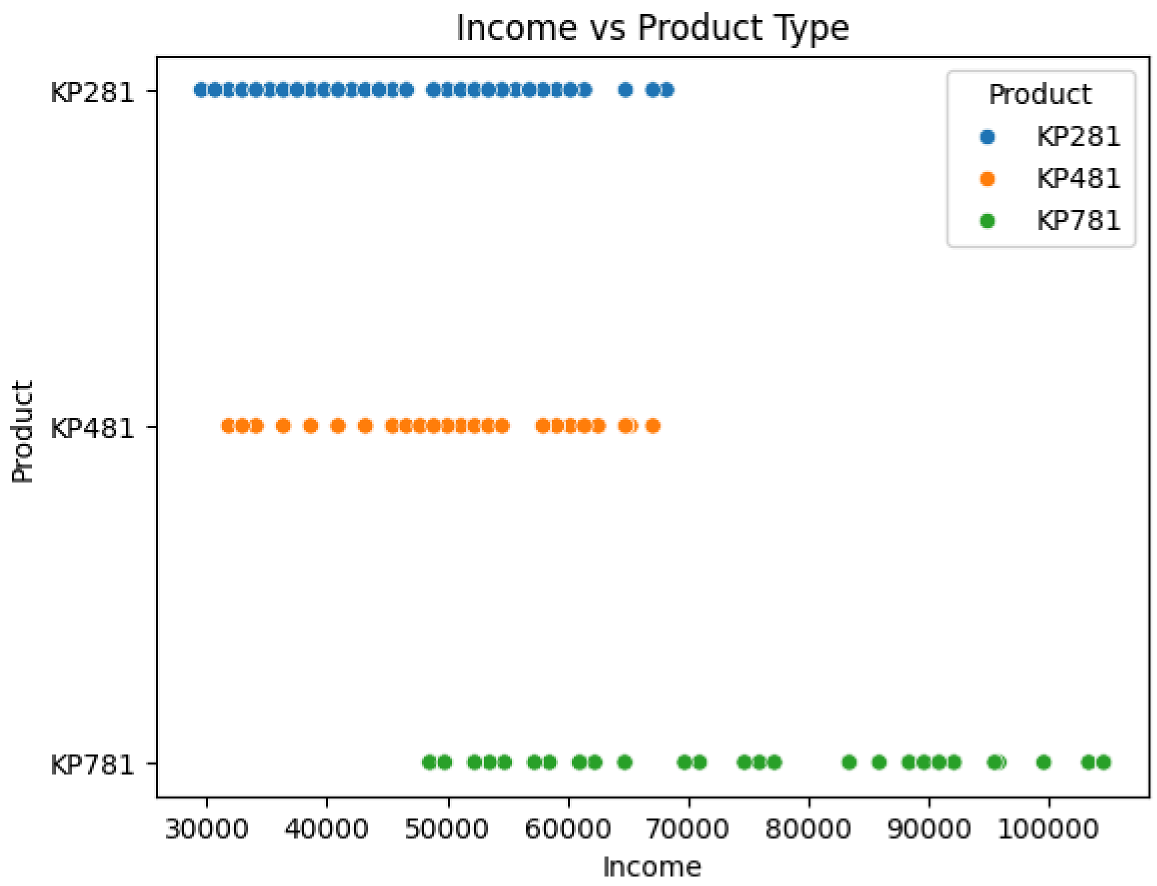


Observation

- Partnered customers are likely to purchase more treadmills than singles irrespective of type
- If we see ndividually singles purchase KP281 more
- Partnered customers are also likely to purchse KP281

In [122…

```
sns.scatterplot(data = df,x = 'Income',y = 'Product',hue = 'Product')
plt.title('Income vs Product Type')
plt.show()
```



Observation

- Income in range of 25000 to below 70000 most likely to buy KP281 and some section among the same range even buy KP481 as well
- Income of greater than 70K till 100000 or more buy KP781

4. Representing the Probability

Understanding the product distribution

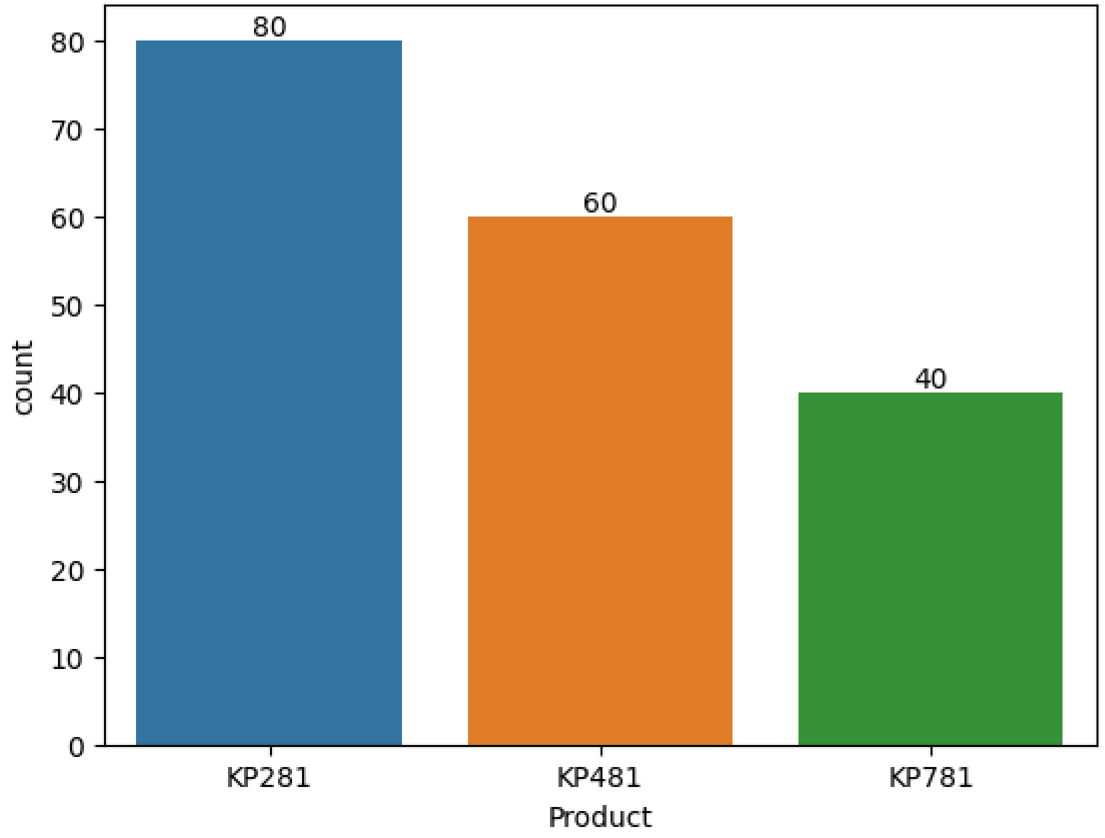
In [123…
```
product_count =  pd.DataFrame(df['Product'].value_counts()).reset_index()
product_count
```

Out[123]:

	Product	count
0	KP281	80
1	KP481	60
2	KP781	40

In [124…
```
ax3 = sns.barplot(data = product_count,x='Product',y='count',hue = 'Product' )

for container in ax3.containers:
  ax3.bar_label(container)
```

```
plt.show()
```



Find the marginal probability (what percent of customers have purchased KP281, KP481, or KP781)

In [125…

```
product_marginal_probability = pd.crosstab(df['Product'],columns = 'marginal probability',normalize =True)
product_marginal_probability
```

Out[125]:

col_0	marginal probability
Product	
KP281	0.444444
KP481	0.333333
KP781	0.222222

Observation

- Among all the products KP281 has highest probability for purchasing around **0.44**
- Followed by KP481 **0.33**
- Least would be KP781 **0.22**

Find the conditional probability that an event occurs given that another event has occurred.

Given the gender what is the probability that they'll buy a particular product type

In [126…
```
probability_gender_product = pd.crosstab(df['Product'],df['Gender'],margins = True,margins_name ='Total')
probability_gender_product
# below is the contingency table showing Product type and Gender
```

Out[126]:

Gender Product	Female	Male	Total
KP281	40	40	80
KP481	29	31	60
KP781	7	33	40
Total	76	104	180

In [127…
```
conditional_probability_given_gender = pd.crosstab(df['Product'], df['Gender'],normalize='columns')
conditional_probability_given_gender
```

Out[127]:

Gender Product	Female	Male
KP281	0.526316	0.384615
KP481	0.381579	0.298077
KP781	0.092105	0.317308

Observation

- P(buying a partiuclar product|Females)
  - Among *females* probability of buying **KP281** is more around **0.526316**
  - Least is **KP781** around **0.092105**
- P(buying a particular product|Males)
  - Even in *males* probability of buying **KP281** is more around **0.384615**
  - Least is **KP481** around **0.298077**

Given the Marital Status what is the probability that they'll buy a particular product type

In [128…
```
conditional_probability_given_marital_Status = pd.crosstab(df['Product'],df['MaritalStatus'],normalize = 'columns')
conditional_probability_given_marital_Status
```

Out[128]:

MaritalStatus Product	Partnered	Single
KP281	0.448598	0.438356
KP481	0.336449	0.328767
KP781	0.214953	0.232877

Observation

- P(buying a partiuclar product|Partnered)
  - Among *Partnered* probability of buying **KP281** is more around **0.44**
  - Least is **KP781** around **0.21**
- P(buying a particular product|Single)
  - Even in *Single* probability of buying **KP281** is more around **0.43**
  - Least is **KP781** around **0.23**

Find the probability that the customer buys a product based on few other features like Fitness and Usage

Given the Fitness level what is the probability that they'll buy a particular product type

In [129…
```
conditional_probability_given_fitness = pd.crosstab(df['Product'],df['Fitness'],normalize = 'columns')
conditional_probability_given_fitness
```

Out[129]:

Fitness	1	2	3	4	5
Product					
KP281	0.5	0.538462	0.556701	0.375000	0.064516
KP481	0.5	0.461538	0.402062	0.333333	0.000000
KP781	0.0	0.000000	0.041237	0.291667	0.935484

Observation

- Customers who rated their fitness levels **5** have higher probability of buying KP781 around **0.93**
- Customers with ratings between **1-4** tend to buy **KP281** more rather than **KP481**

Given the Usage frequencies what is the probability that they'll buy a particular product type

Excluding the outliers like those who plan to use treadmill for about 6 to 7 times in a week

In [130…
```
conditional_probability_given_Usage = pd.crosstab(df['Product'],df['Usage'][(df['Usage']!=6) & (df['Usage']!=7)],normalize = 'columns')
conditional_probability_given_Usage
```

Out[130]:

Usage	2	3	4	5
Product				
KP281	0.575758	0.536232	0.423077	0.117647
KP481	0.424242	0.449275	0.230769	0.176471
KP781	0.000000	0.014493	0.346154	0.705882

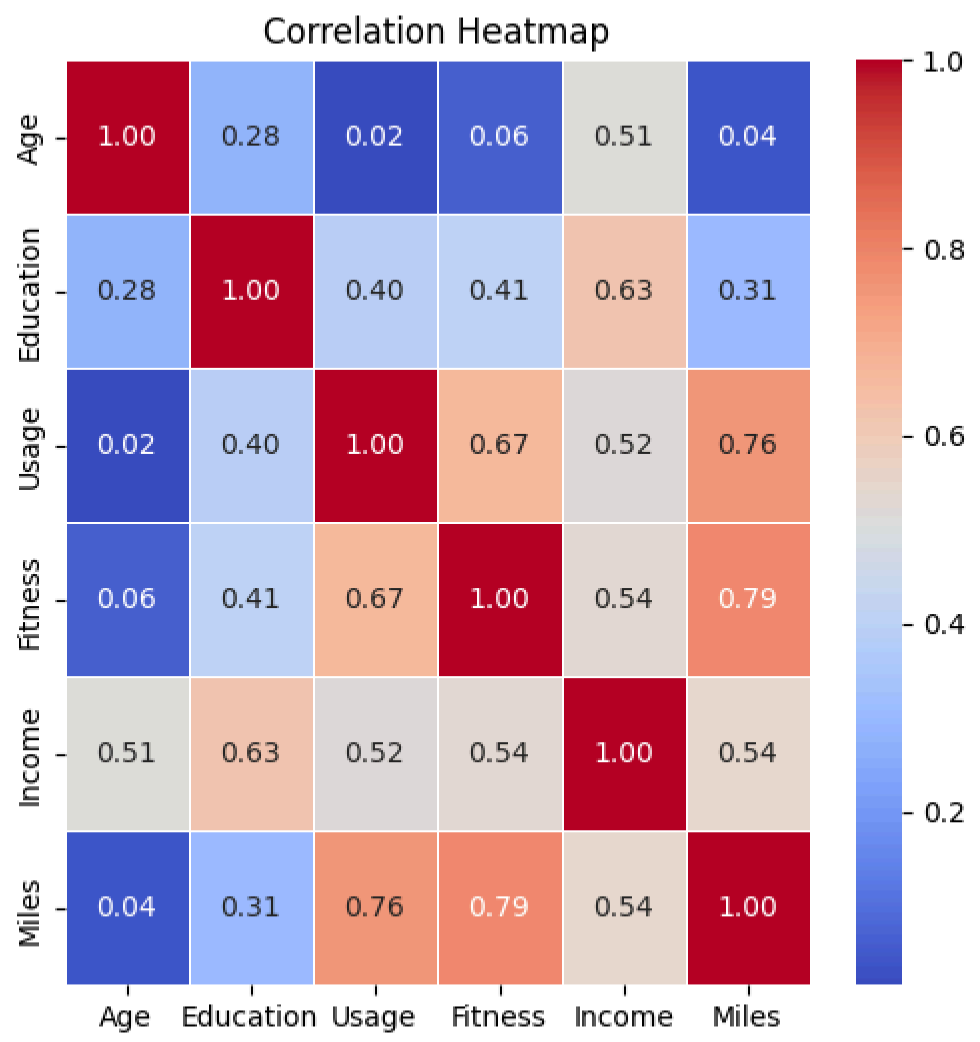
Observation

- Customers who plan to use treadmill for about **5** times in a week tend to buy **KP781** with a probability of **0.70**
- Rest of the customers tend to buy **KP281**

5. Check the correlation among different factors

In [131…

```
# Selecting numerical features
numerical_features = ['Age','Education','Usage','Fitness','Income','Miles']

# Creating a correlation matrix
correlation_matrix = df[numerical_features].corr()

# Plotting the heatmap
plt.figure(figsize=(6, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



Observation

- All the features in the given data has moderate to strong positive correlation.
- Fitness and Miles have strong positive correlation **0.79**
- Similarly Fitness and Usage **0.67**
- Miles and Usage too has strong positive correlation **0.76**

- Income and Education also has strong positive correlation **0.63**

6. Customer profiling

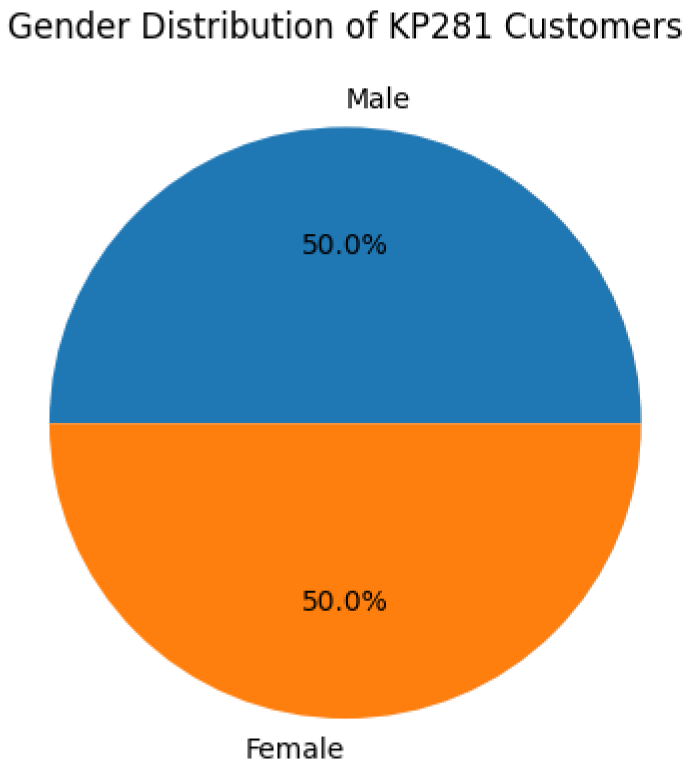
Treadmill product type KP281 price $1,500

In [132…
```
KP281_data = df[['Age','Gender','Income']][df['Product']=='KP281']
```

In [133…
```
print('Customers of KP281 product type and their Age ranges between', KP281_data['Age'].min(),'and',KP281_data['Age'].max())
print('Income ranges between',KP281_data['Income'].min(),'and',KP281_data['Income'].max(),'$')
```

```
Customers of KP281 product type and their Age ranges between 18 and 50
Income ranges between 29562 and 68220 $
```

In [134…
```
gender_counts = KP281_data['Gender'].value_counts()
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%')
plt.title('Gender Distribution of KP281 Customers')
plt.show()
```



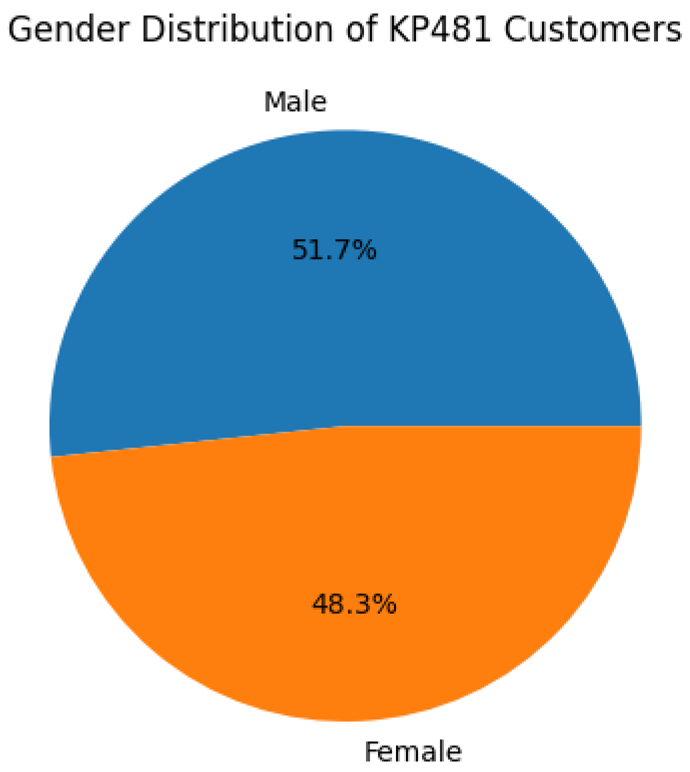
Treadmill product type KP481 price $ 1,750

In [135…
```
KP481_data = df[['Age','Gender','Income']][df['Product']=='KP481']
```

In [136…
```
print('Customers of KP481 product type and their Age ranges between', KP481_data['Age'].min(),'and',KP481_data['Age'].max())
print('Income ranges between',KP481_data['Income'].min(),'and',KP481_data['Income'].max(),'$')
```

```
Customers of KP481 product type and their Age ranges between 19 and 48
Income ranges between 31836 and 67083 $
```

In [137...
```
gender_counts = KP481_data['Gender'].value_counts()
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%')
plt.title('Gender Distribution of KP481 Customers')
plt.show()
```



Observation

- Here customer gender distribution in almost striking balance with **51.7%** being **males** and **48.3% being females**

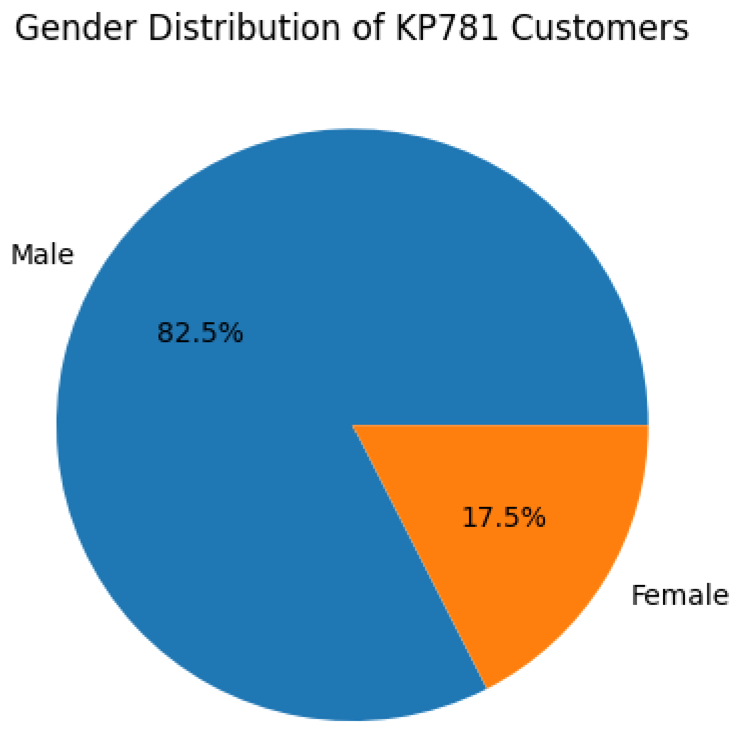
Treadmill product type KP781 price $2,500

In [138...
```
KP781_data = df[['Age','Gender','Income']][df['Product']=='KP781']
```

In [139...
```
print('Customers of KP781 product type and their Age ranges between', KP781_data['Age'].min(),'and',KP781_data['Age'].max())
print('Income ranges between',KP781_data['Income'].min(),'and',KP781_data['Income'].max(),'$')
```

```
Customers of KP781 product type and their Age ranges between 22 and 48
Income ranges between 48556 and 104581 $
```

In [140...
```
gender_counts = KP781_data['Gender'].value_counts()
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%')
plt.title('Gender Distribution of KP781 Customers')
plt.show()
```



Observation

- Customers who buy this product type are mostly **males** with **82.5%** distribution and females very less with **17.5%**

Insights:

- Customers of all product types span a relatively wide age range.
- Both males and females are almost equally fitness driven
- KP281 is positioned as a versatile and inclusive option that meets the needs and preferences of a diverse customer base.
- It also suggests customers that buy KP781 tend to have higher incomes compared to customers of KP281 and KP481.
- Count of Male customers purchasing KP781 is high,suggesting they would buy premium products to enhance there fitness game.
- Customers with longer education years have greater incomes.
- Most of the customers about 50% have **94 miles** per week as their goal,they plan to use treadmill for **3 times** a week and they rate themselves with **3 rating**,indicating moderate to strong motivation towards fitness.
- KP281 is the most preferred product, with a probability of purchase around **44%**
- KP481 follows with a probability of purchase around **33%**.
- KP781 is the least preferred, with a probability of purchase around **22%**.
- Among females, the probability of purchasing KP281 is approximately **52.6%**, while for males, it's around **38.5%**.
- The probability of purchasing KP481 is higher among males around **29.8%** compared to females around **9.2%**.
- KP781 is least preferred by both genders, with probabilities of around **9.2%** for females and **30.8%** for males.
- Partnered customers show a preference for KP281 with a probability of around **44%**, while single customers' probability is around **43%**.
- KP781 is least preferred by both groups, with probabilities around **21%** for partnered and **23%** for single customers.
- Customers with a fitness level of **5** have a high probability of purchasing **KP781**, around **93%**.
- Those with fitness levels between 1-4 prefer KP281 more, with probabilities ranging from around **38% to 56%**.
- Customers planning to use the treadmill five times a week have a high probability around **70%** of purchasing KP781.

- For customers with lower usage frequency, the probability of purchasing KP281 is higher, ranging from around **42% to 58%**.
- Fitness level strongly correlates with miles walked and treadmill usage.
- Higher income tends to be associated with higher education levels.
- Customers who walk more miles also tend to use the treadmill more frequently.
- Fitness level is positively correlated with treadmill usage frequency.

Recommendations

- **Targeted Marketing** Tailor marketing to highlight KP281's versatility for females and emphasize KP781's premium features for males.
- **Educational Content** Create educational materials to demonstrate how KP781 can help customers achieve fitness goals.
- **Bundle Deals and referral offers** Offer bundle deals with KP281 and accessories to encourage repeat purchases among partnered customers and referral offers among singles
- **Loyalty Programs** Implement loyalty programs to reward frequent purchasers, especially those with moderate fitness motivation.
- **Personalized Recommendations** Provide personalized product recommendations based on individual fitness goals and usage patterns.
- **Community Engagement** Foster a sense of community through events where customers can share experiences.
- **Feedback Loop** Gather regular feedback to continuously improve products and services.