aerofit

March 20, 2024

Defining Problem Statement and Analysing basic metrics

About Case Study:

About Aerofit: Aerofit is a leading brand in the field of fitness equipment. Aerofit provides a product range including machines such as treadmills, exercise bikes, gym equipment, and fitness accessories to cater to the needs of all categories of people.

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.

1. Perform descriptive analytics to create a customer profile for each AeroFit treadmill product by developing appropriate tables and charts.
2. 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.

Dataset Information:

The company collected the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months.

The dataset has the following features: 1. Product Purchased: KP281, KP481, or KP781 2. Age: In years 3. Gender: Male/Female 4. Education: In years 5. MaritalStatus: Single or partnered 6. Usage: The average number of times the customer plans to use the treadmill each week. 7. Income: Annual income (in $) 8. Fitness: Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 is the excellent 9. Miles: The average number of miles the customer expects to walk/run each week

Product Portfolio: 1. The KP281 is an entry-level treadmill that sells for $1500.

2. The KP481 is for mid-level runners that sell for $1750.

3. The KP781 treadmill is having advanced features that sell for $2500.

[ ]:
```
# importing required libraries

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[ ]:
```
# downloading the dataset

!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/
  ↪original/aerofit_treadmill.csv
```

```
Downloading…
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/ori
ginal/aerofit_treadmill.csv
To: /content/aerofit_treadmill.csv
100% 7.28k/7.28k [00:00<00:00, 35.3MB/s]
```

[ ]:
```
# loading the dataset

df = pd.read_csv('aerofit_treadmill.csv')
```

[ ]:
```
# exploring dataset

df
```

[ ]:

	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
..	…	…	…	…	…	…	…	…	…
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

[180 rows x 9 columns]

[ ]:
```
# checking shape of dataset

df.shape
```

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

The Dataset has 180 rows and 9 columns.

[ ]:
```
# checking data type of all attributes

df.dtypes
```

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

Product, Gender, MaritalStatus are object (string)

Age, Education, Usage, Fitness, Income, Miles are int64 (integer)

[ ]:
```
# checking statistical summary of data

df.describe(include="all")
```

[ ]:
```
        Product         Age Gender   Education MaritalStatus       Usage  \
count       180  180.000000    180  180.000000            180  180.000000
unique        3         NaN      2         NaN              2         NaN
top       KP281         NaN   Male         NaN     Partnered         NaN
freq         80         NaN    104         NaN            107         NaN
mean        NaN   28.788889    NaN   15.572222            NaN    3.455556
std         NaN    6.943498    NaN    1.617055            NaN    1.084797
min         NaN   18.000000    NaN   12.000000            NaN    2.000000
25%         NaN   24.000000    NaN   14.000000            NaN    3.000000
50%         NaN   26.000000    NaN   16.000000            NaN    3.000000
75%         NaN   33.000000    NaN   16.000000            NaN    4.000000
max         NaN   50.000000    NaN   21.000000            NaN    7.000000

            Fitness         Income       Miles
count    180.000000     180.000000  180.000000
unique          NaN            NaN         NaN
top             NaN            NaN         NaN
```

freq	NaN	NaN	NaN
mean	3.311111	53719.577778	103.194444
std	0.958869	16506.684226	51.863605
min	1.000000	29562.000000	21.000000
25%	3.000000	44058.750000	66.000000
50%	3.000000	50596.500000	94.000000
75%	4.000000	58668.000000	114.750000
max	5.000000	104581.000000	360.000000

Observations:

- Total count of all columns is 180.
- There are 3 types of Product.
- There are 2 types of Gender.
- There are 2 types of Marital Status.
- The most sold treadmill is KP281.
- In the dataset, number of Males is greater than no of Females.
- Partnered individuals outnumber singles.
- 80 units of KP281 have been sold.
- There are 104 Males. Therefore, there are 76 Females.
- There are 107 people with Marital Status : Partnered. Therefore, there are 73 people with Marital Status : Single
- Age: The mean age of the customers is approx 29 years. Half of the customer's mean age is 26 years. With maximum age as 50 years and minimum age as 18 years.
- Education : Mean education is approx 15.6 years, with maximum 21 years and minimum 12 years. Half of the customer's mean education is 16 years.
- Usage: Mean usage per week is 3.45 times, with maximum usage of 7 times per week and minimum usage of 2 times per week. Half of the customer's mean usage is 3 times per week.
- Fitness: Average rating is 3.3 on a scale of 1 to 5.
- Income: Mean income is 53.7k, with maximum income of 104.5k and minimum income of 29.5k.
- Miles: Average number of miles the customer walk/run each week is 103, with maximum of 360 and minimum of 21.

Non-Graphical Analysis: Value counts and unique attributes

[ ]:
```
# total number of unique product ids

df['Product'].nunique()
```

[ ]:
```
3
```

There are 3 unique product ids.

[ ]:
```
# unique list of product ids

df['Product'].unique().tolist()
```

[ ]:
```
['KP281', 'KP481', 'KP781']
```

Products are KP281, KP481 & KP781

[ ]:
```
# total number of unique ages

df['Age'].nunique()
```

[ ]:
```
32
```

There are total 32 unique ages.

[ ]:
```
# list of unique ages
df['Age'].unique()
```

[ ]:
```
array([18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 43, 44, 46, 47, 50, 45, 48, 42])
```

[ ]:
```
# number of people having same age

df['Age'].value_counts().sort_index()

# left-side represents age, right-side represents no of people of that age
```

[ ]:
```
18     1
19     4
20     5
21     7
22     7
23    18
24    12
25    25
26    12
27     7
28     9
29     6
30     7
31     6
32     4
33     8
34     6
35     8
36     1
37     2
38     7
39     1
40     5
41     1
42     1
43     1
```

```
44     1
45     2
46     1
47     2
48     2
50     1
Name: Age, dtype: int64
```

[ ]:
```
# Count of male and female customers

df['Gender'].value_counts()
```

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

There are 104 males and 76 females.

[ ]:
```
# total number of unique education(in years)

df['Education'].nunique()
```

[ ]:
```
8
```

[ ]:
```
# list of unique education(in years)

df['Education'].unique().tolist()
```

[ ]:
```
[14, 15, 12, 13, 16, 18, 20, 21]
```

[ ]:
```
# number of people having same education (in years)
df['Education'].value_counts().sort_index()

# left-side represent education.
# right-side represent no of people.
```

[ ]:
```
12     3
13     5
14    55
15     5
16    85
18    23
20     1
21     3
Name: Education, dtype: int64
```

[ ]:
```
# Count of people based on Marital Status

df['MaritalStatus'].value_counts()
```

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

There are 107 partnered individuals and 73 single individuals.

[ ]:
```
# Number of customers based on Usage count

df['Usage'].value_counts().sort_index()

# Left-Side : Usage per week
# Right-Side : No of customers having same usage
```

[ ]:
```
2    33
3    69
4    52
5    17
6     7
7     2
Name: Usage, dtype: int64
```

[ ]:
```
# Number of people who rate them same on a scale of 1-to-5.

df['Fitness'].value_counts().sort_index()

# Left-Side represent scale from 1-to-5.
# Right-side represent no of people.
```

[ ]:
```
1     2
2    26
3    97
4    24
5    31
Name: Fitness, dtype: int64
```

[ ]:
```
# Count of different product types.

df['Product'].value_counts().sort_index()
```

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

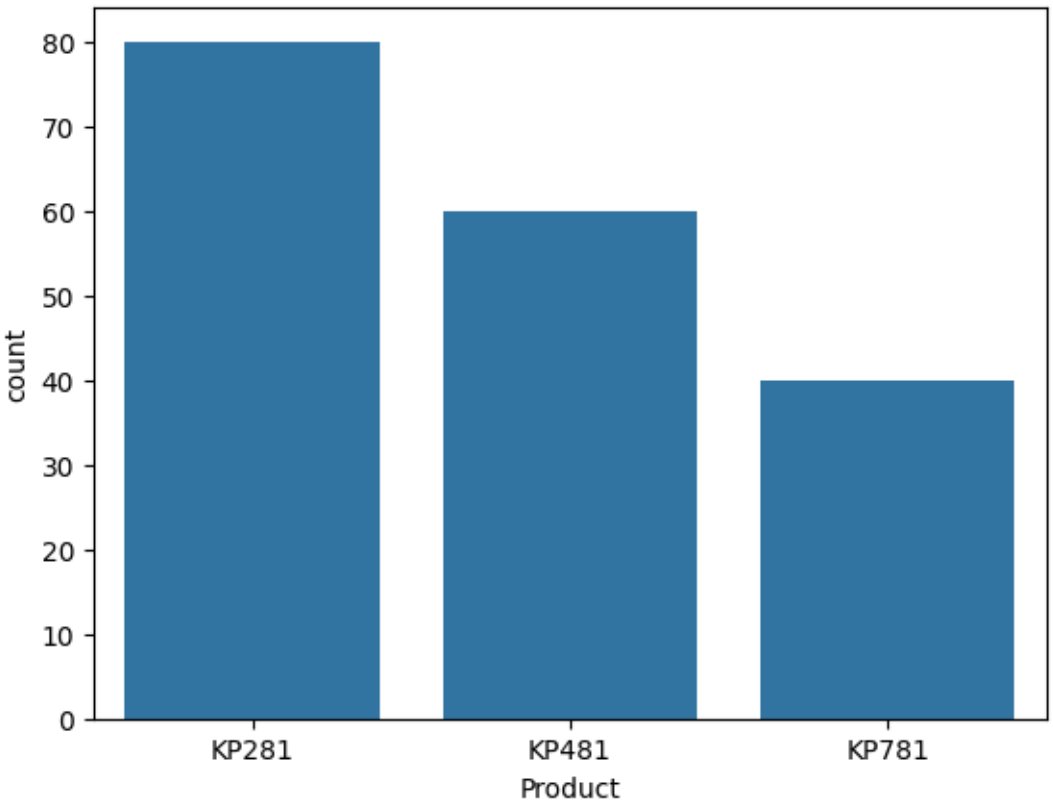
Summary: 1. KP281, KP481, KP781 are the 3 different products 2. Most commonly purchased treadmill product type is KP281. 3. Least commonly purchased treadmill product type is KP781. 4. There are 32 unique ages. 5. There are 104 males and 76 females. 6. 8 unique set of Educations (14, 15, 12, 13, 16, 18, 20, 21). 7. Most number of people have rated them 3 in fitness (on a scale of 1-to-5) 8. Majority of customers are having Marital Status: Partnered. 9. Most of the customers used treadmill 3 times a week.

Visual Analysis - Univariate & Bivariate

Univariate Analysis

For continuous variable(s): Distplot, countplot, histogram for univariate analysis

[ ]:
```
# Product Analysis : Using countplot

sns.countplot(data=df, x='Product')
plt.show()
```

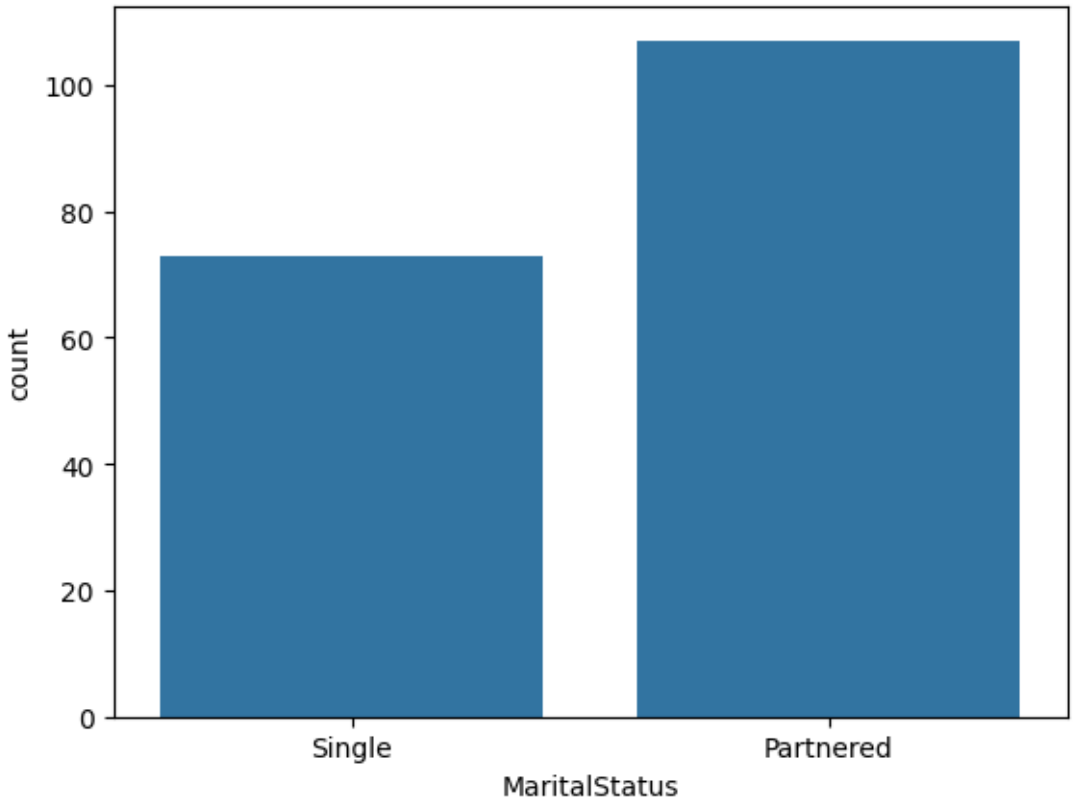


KP281 is most sold product.

KP481 is the second most sold product.

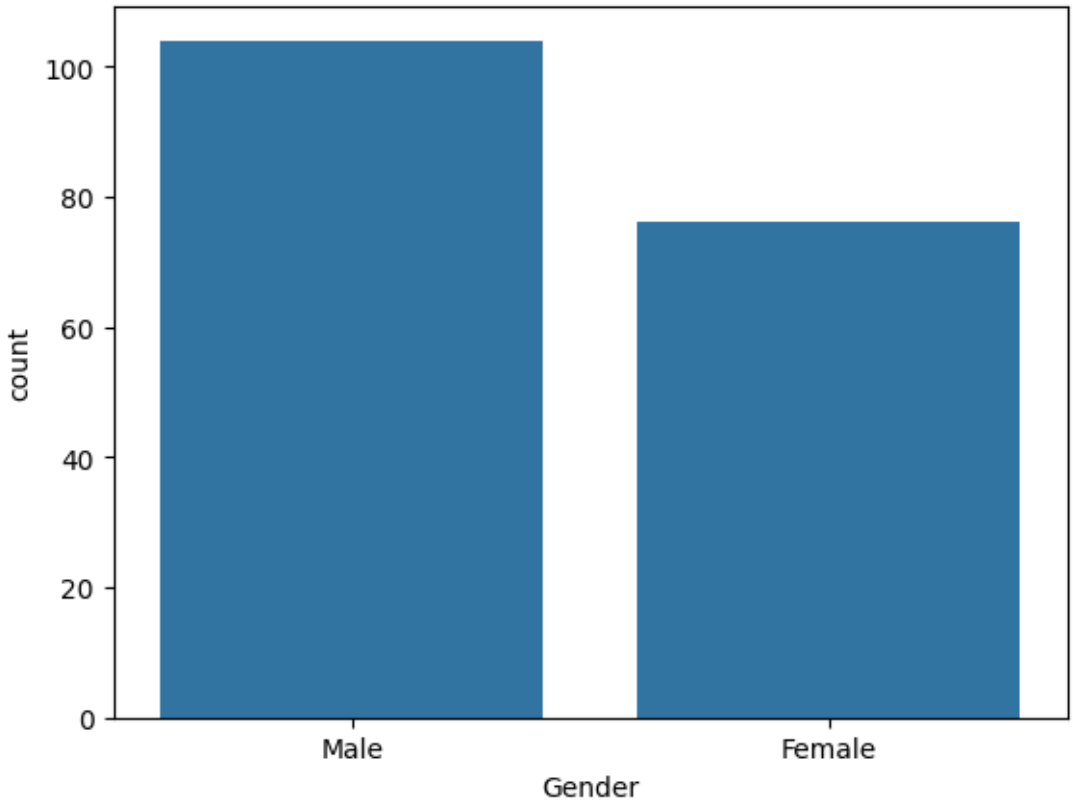
KP781 is the least sold product.

[ ]:
```
# Marital Status Analysis : Using countplot

sns.countplot(data=df, x='MaritalStatus')
plt.show()
```



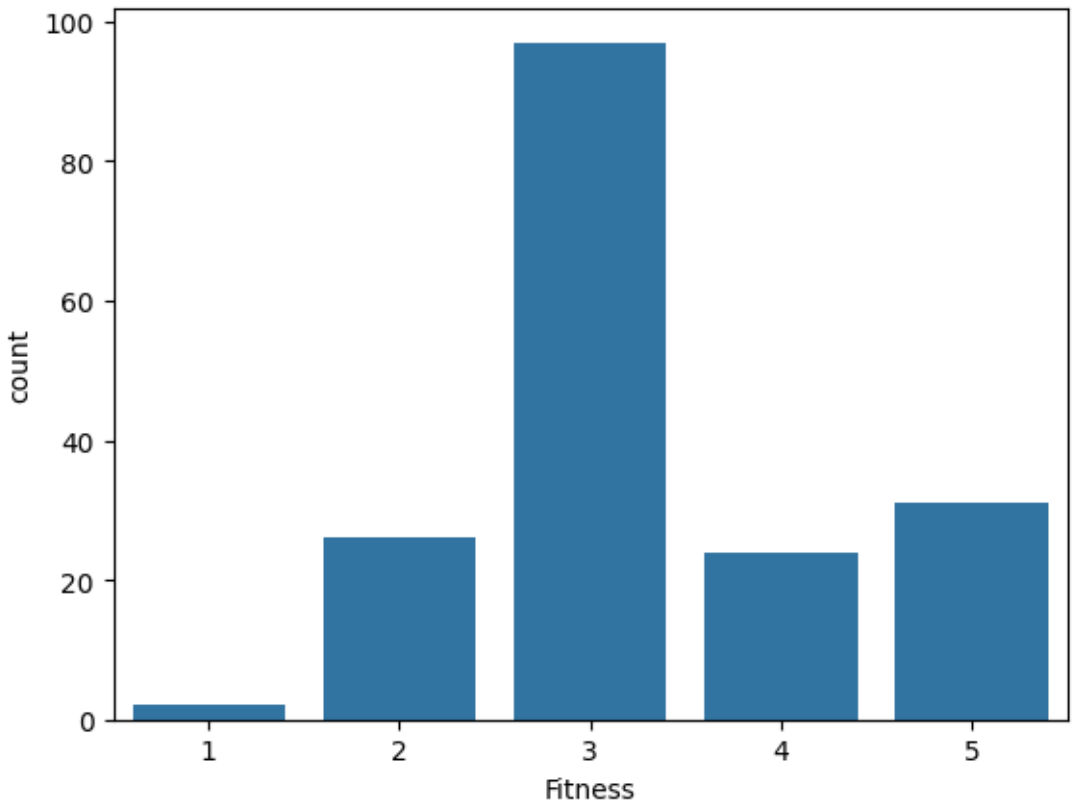
Partnered individuals purchased most products.

[ ]:
```
# Gender Analysis: Using countplot

sns.countplot(data=df, x='Gender')
plt.show()
```



Males have purchased more products as compared to Females.

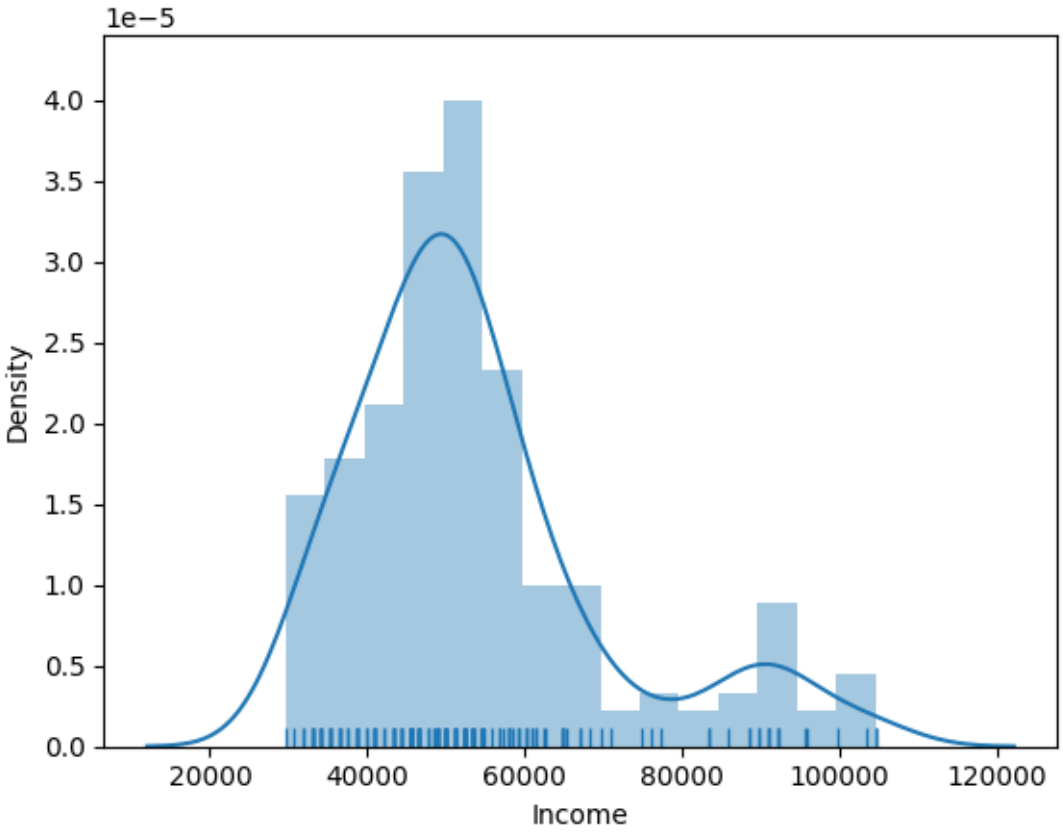
[ ]:
```
# Fitness Rating Analysis: Using Countplot
sns.countplot(data=df, x='Fitness')
plt.show()
```



More than 90 customers have rated their physical fitness as average.

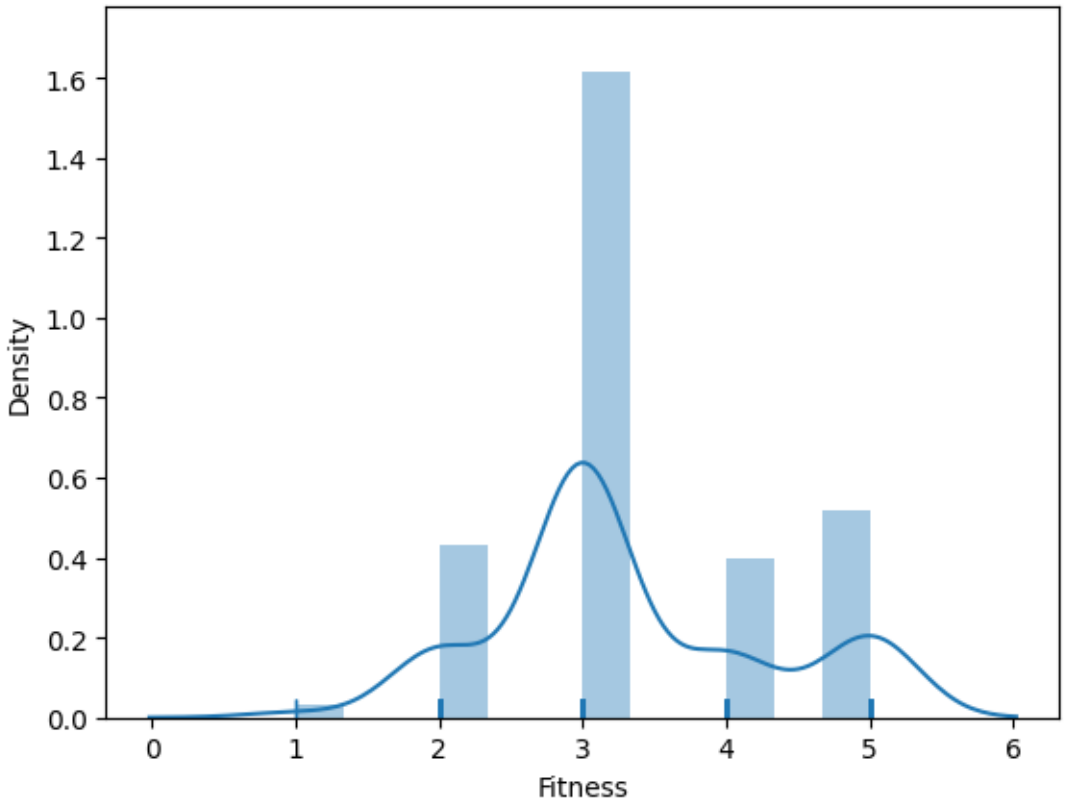
[ ]:
```
# Income Analysis : Using Distplot
sns.distplot(df.Income, rug=True)
plt.show()
```

```
<ipython-input-66-d29d76a21d69>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Income, rug=True)
```

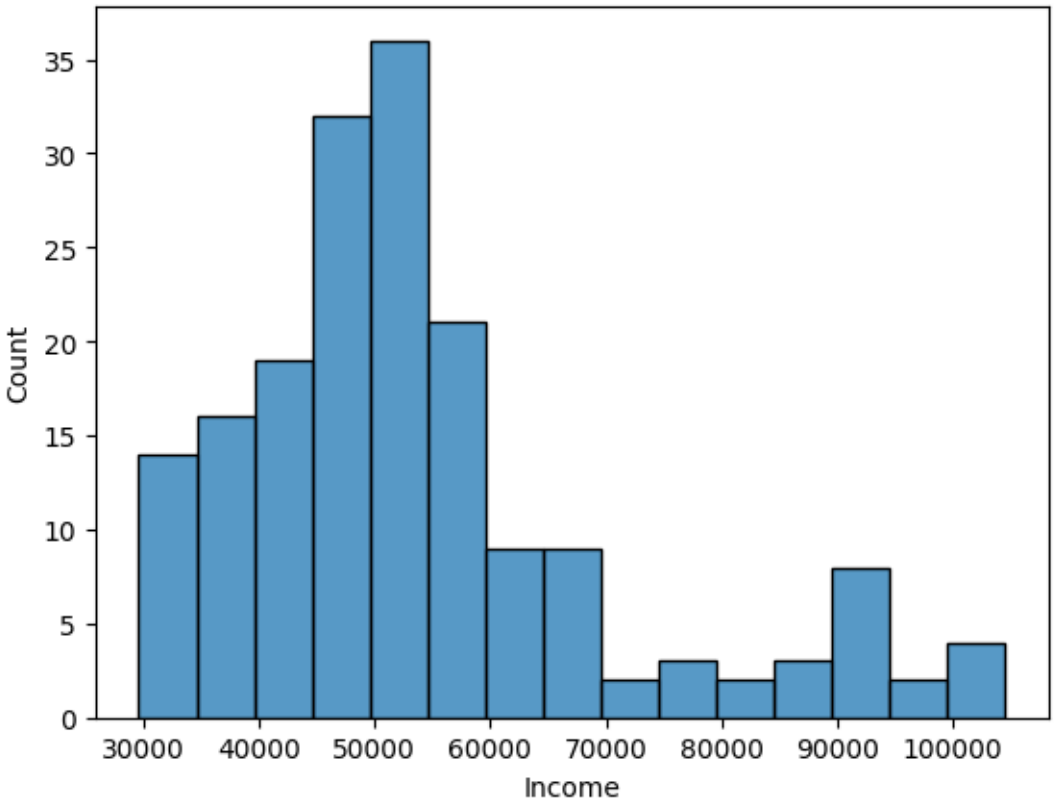


[ ]:
```
# Fitness Rating Analysis : Using Distplot
sns.distplot(df.Fitness, rug=True)
plt.show()
```

```
<ipython-input-68-1f2d4f0ec0d0>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Fitness, rug=True)
```



[ ]:
```
# Income Analysis : Using Histogram
sns.histplot(data=df,x='Income')
plt.show()
```

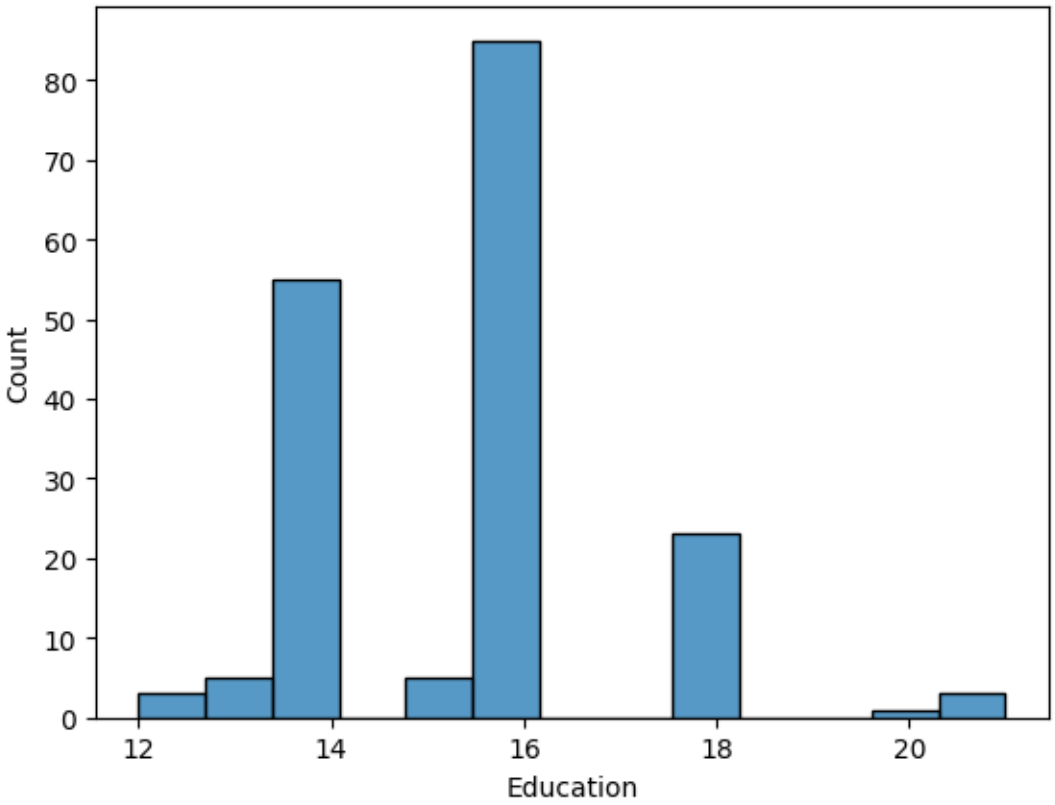


More than 35 customers earn 50-55K per year

More than 30 customers earn 45-50K per year

More than 20 customers earn 55-60K per year

[ ]:
```
# Education Analysis : Using Histogram
sns.histplot(data=df, x ='Education')
plt.show()
```

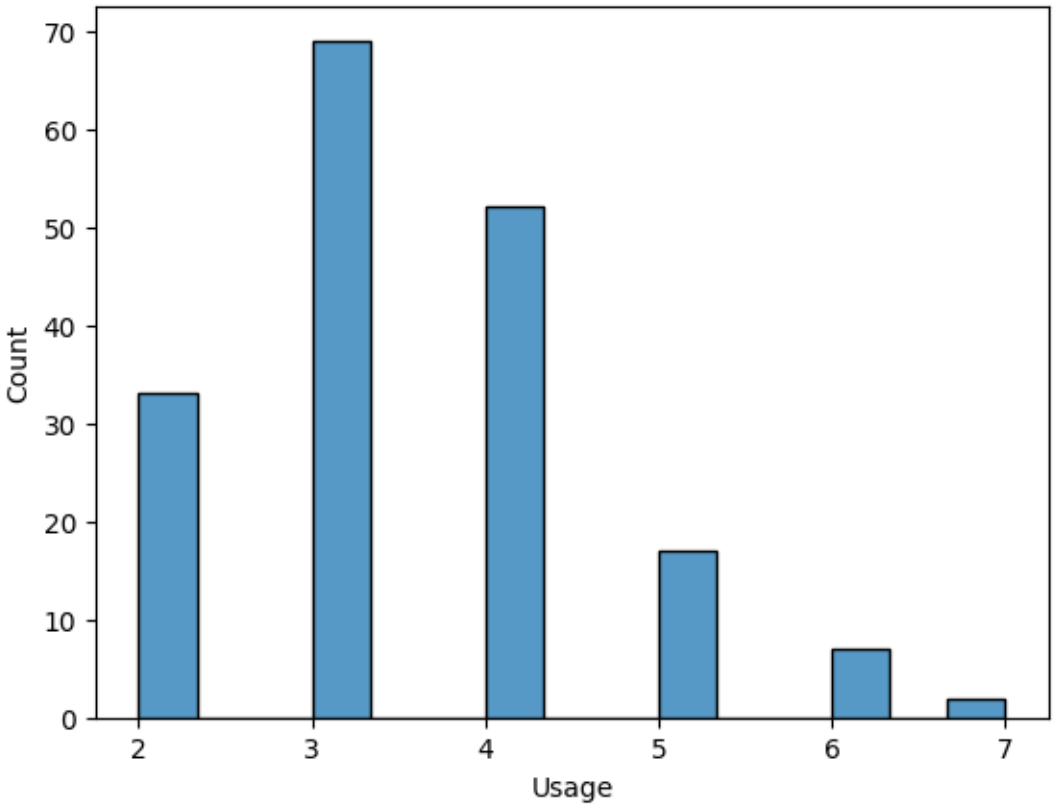


Highest number of customers have 16 as their Education

14 is the second highest education among the customers

20 is the least education among the customers

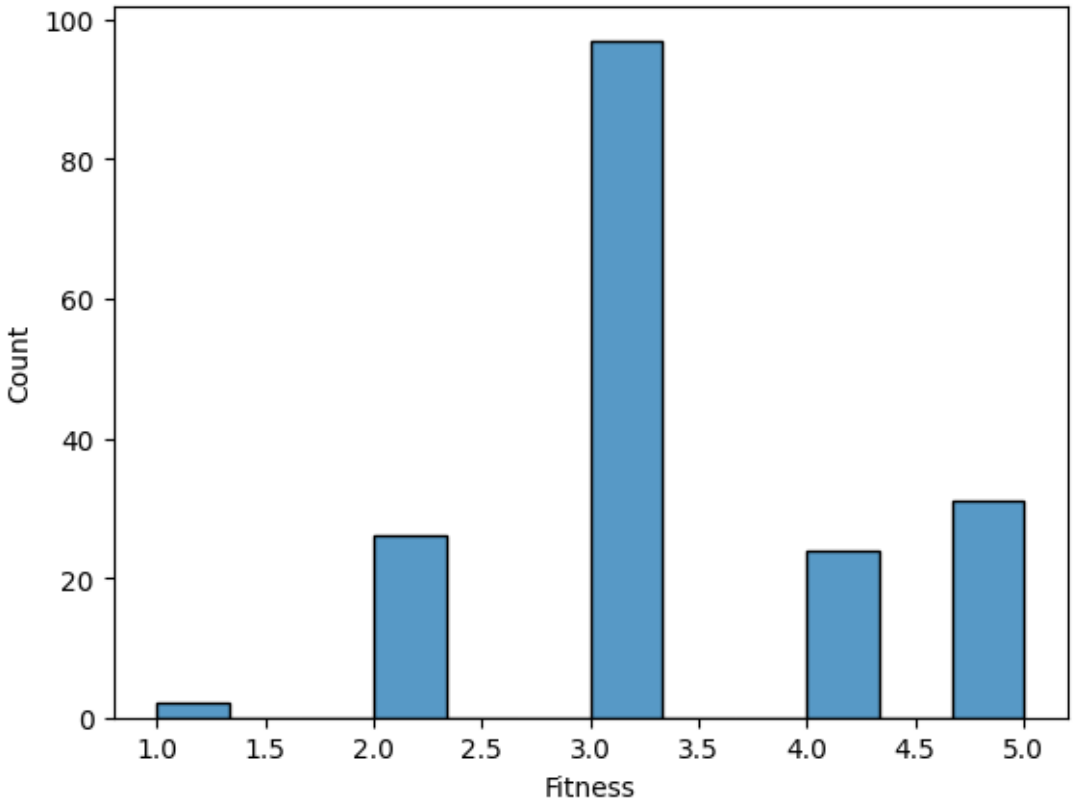
[ ]:
```
# Usage Analysis - Using Histogram

sns.histplot(data=df, x='Usage')
plt.show()
```



3 Days per week is most common usage.

4 days is second highest usage, 2 days is third highest usage among the customers.

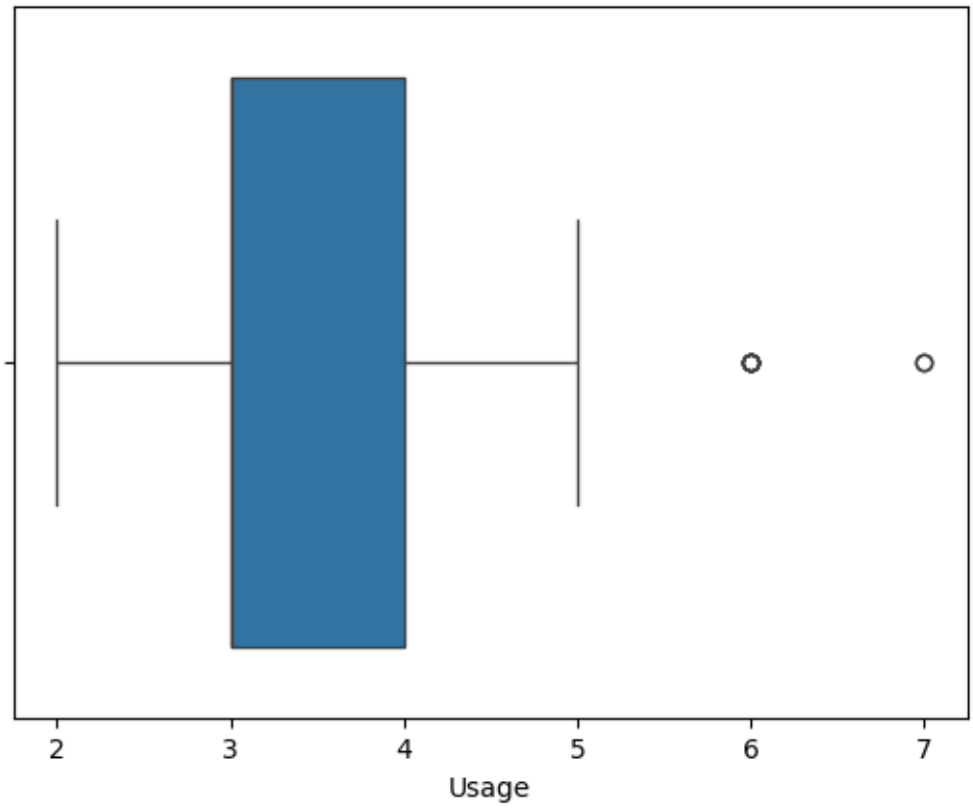
[ ]:
```
# Fitness Analysis : Using Histogram

sns.histplot(data=df, x='Fitness')
plt.show()
```



Most people rated them Average (3.0) on Fitness.

For categorical variable(s): Boxplot

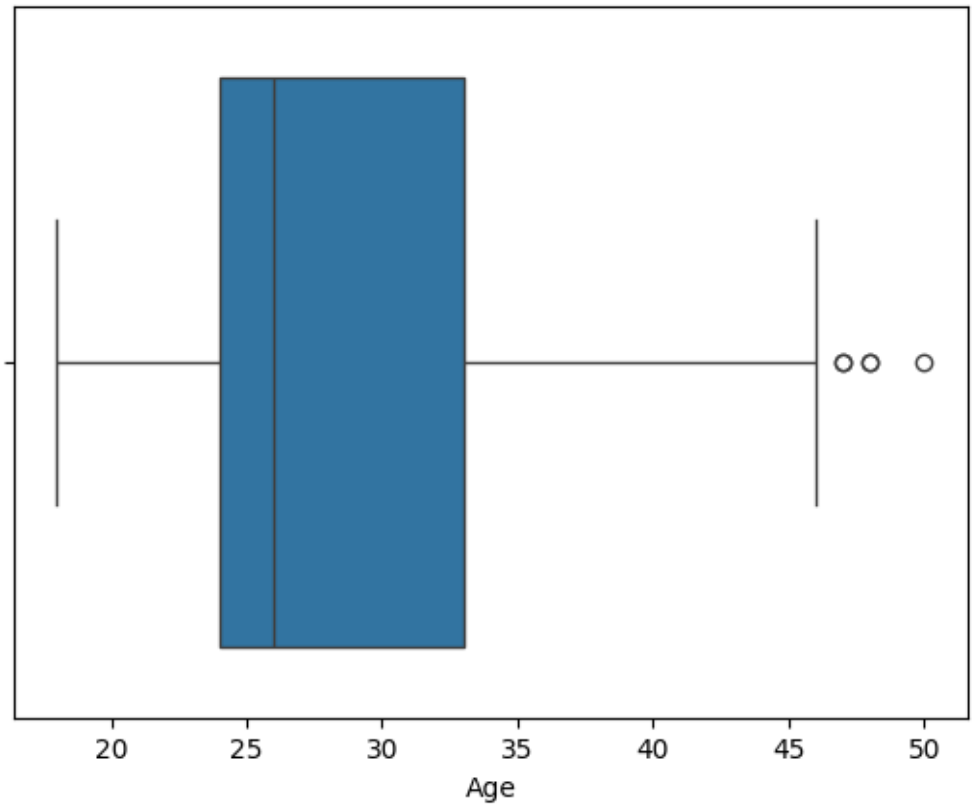
[ ]:
```
# Usage analysis : using boxplot

sns.boxplot(data=df,x='Usage')
plt.show()
```



3 to 4 times per week usage is most preferred among customers.

6 and 7 times per week is only for a few customers (Outliers).

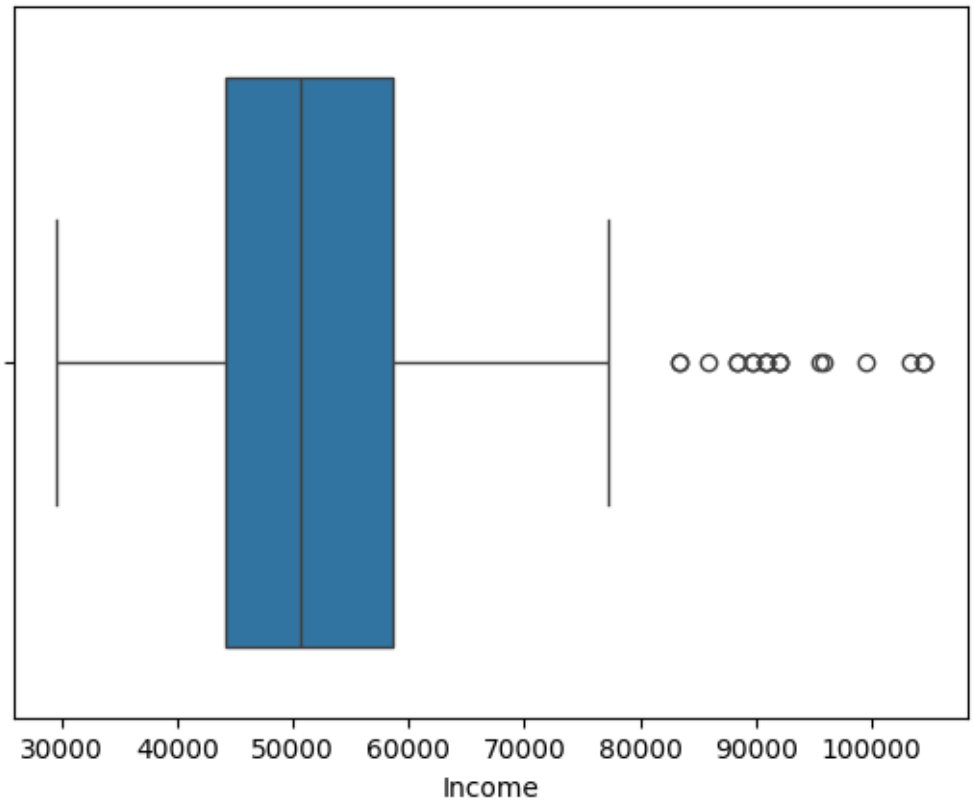
[ ]:
```
# Age analysis : using boxplot

sns.boxplot(data=df, x='Age')
plt.show()
```



Above 45 years old customers are very few. (Outliers)

Most customers lie between the age group 23 to 34.

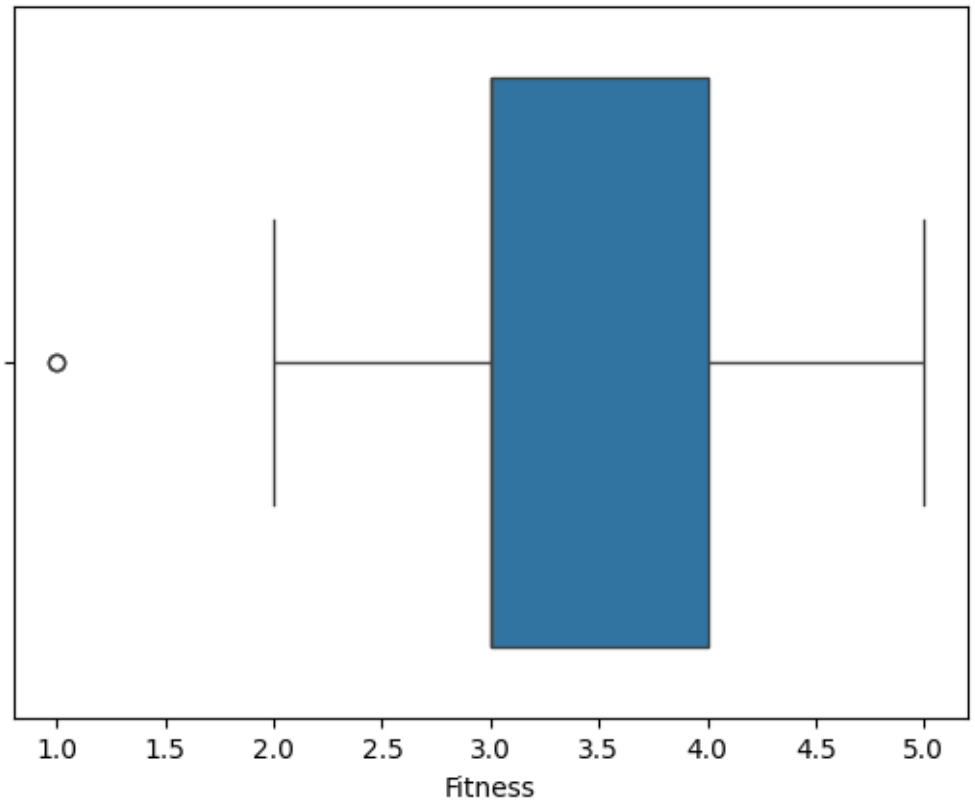
[ ]:
```
# Income analysis : using boxplot

sns.boxplot(data=df, x='Income')
plt.show()
```



Only a few customers have income above 77k (Outliers).

Most customers earn between 45k to 60k.

[ ]:
```
# Fitness Rating Analysis - Using Boxplot

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



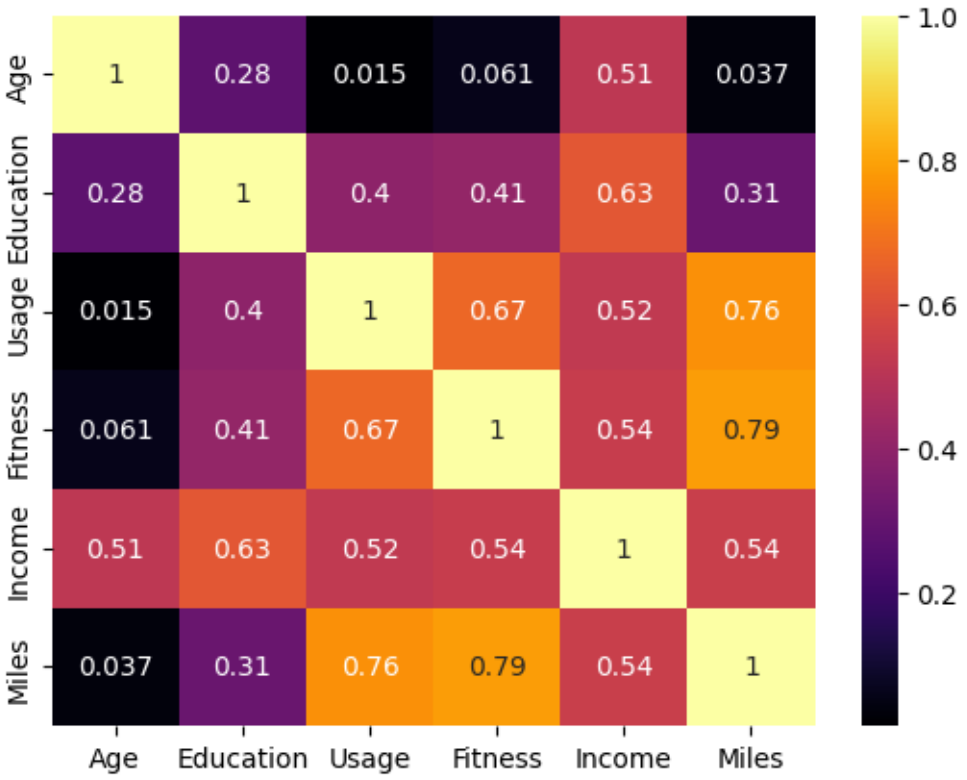
Few Customers have rated their fitness as 1 (Outliers)

Most customers have rated their fitness between 3 to 4.

For correlation: Heatmaps, Pairplots

[ ]:
```
# Correlation Heatmap

sns.heatmap(df.corr(), annot=True, cmap='inferno')
plt.show()
```

```
<ipython-input-81-78dd8ea8baa7>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df.corr(), annot=True, cmap='inferno')
```



Correlation between Age & Income is 0.51.

Correlation between Education & Income is 0.63.

Correlation between Usage & Fitness is 0.67.

Correlation between Usage & Income is 0.52.

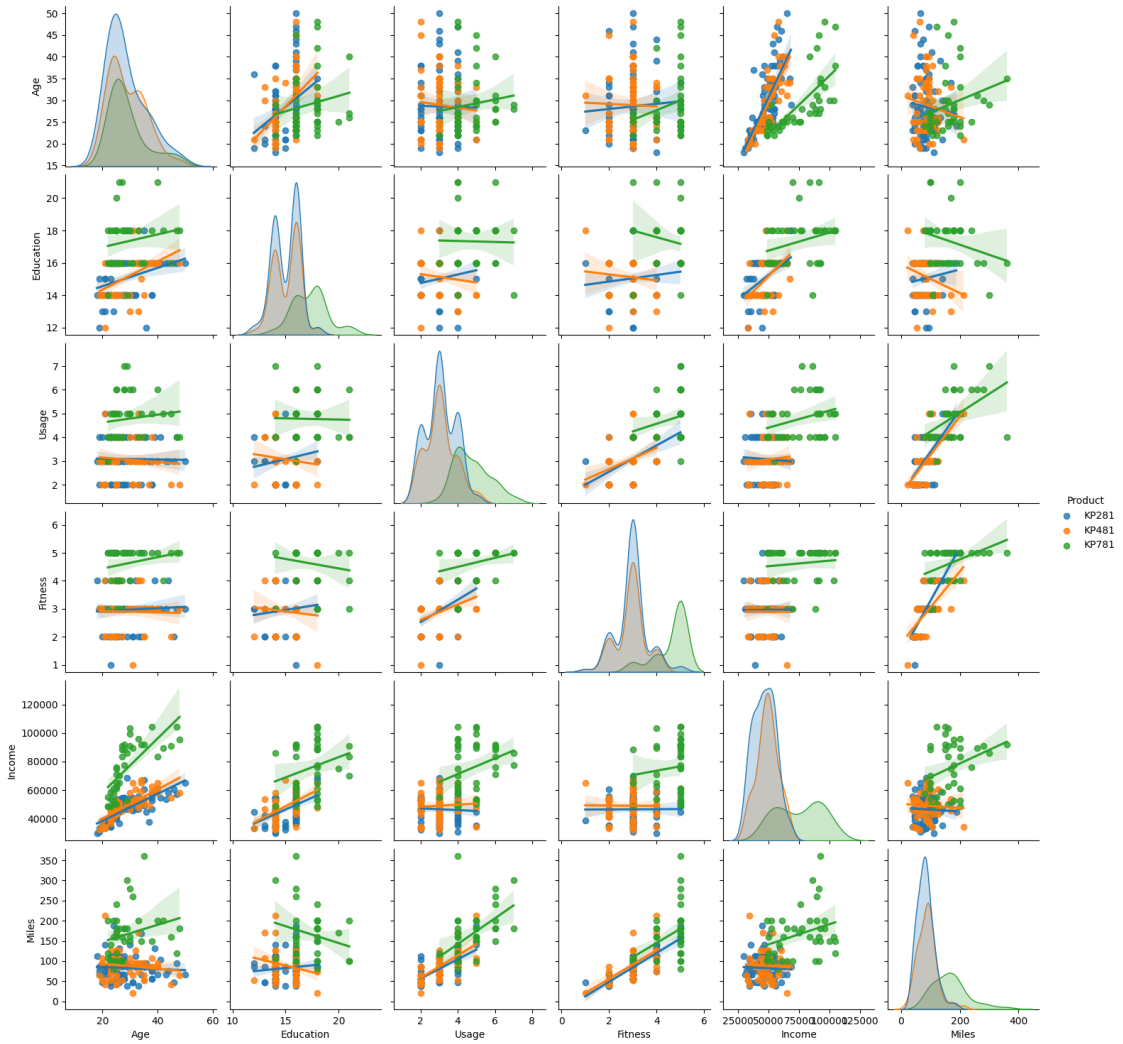
Correlation between Usage & Miles is 0.76.

Correlation between Fitness & Income is 0.54.

Correlation between Fitness & Miles is 0.79.

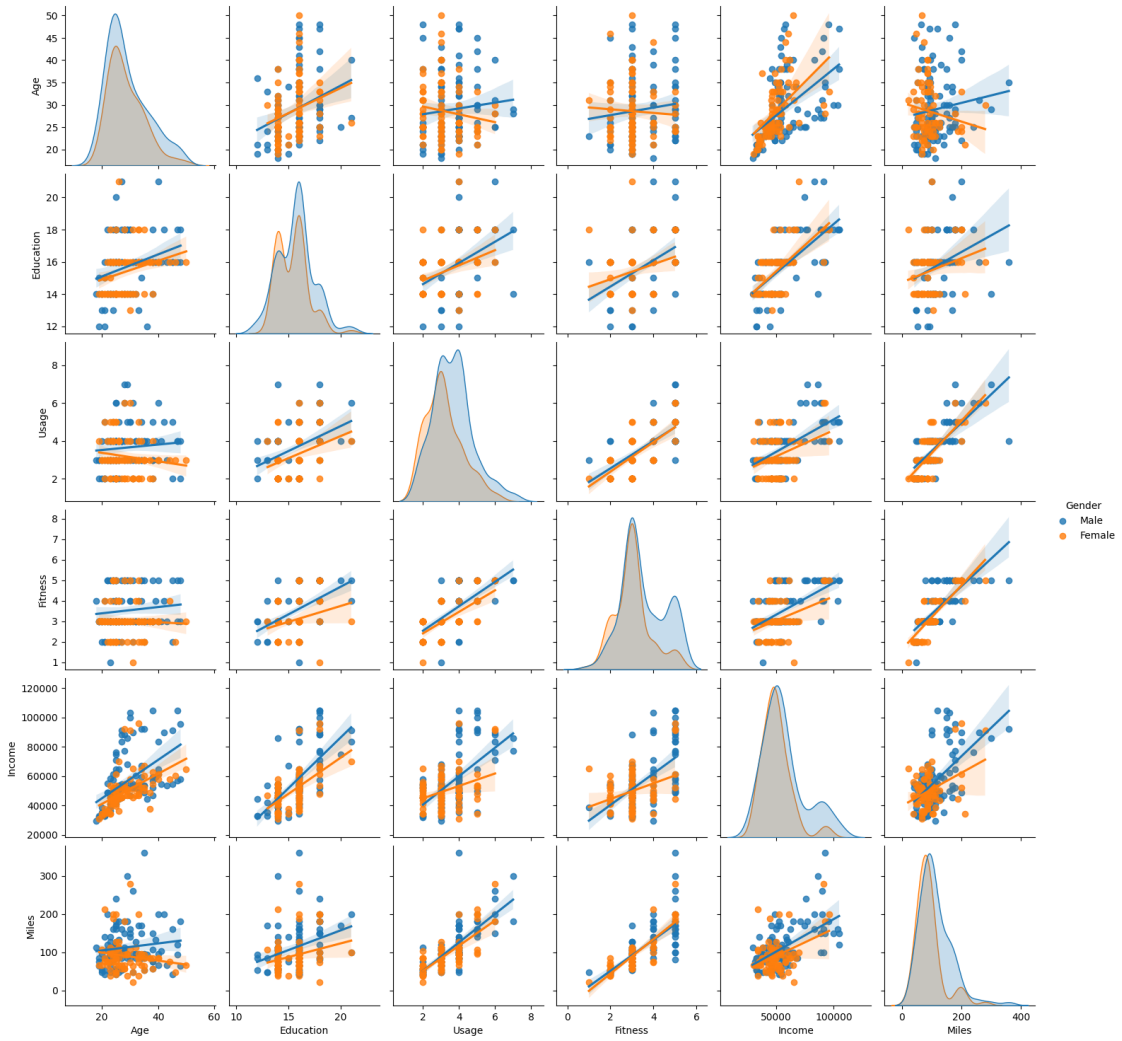
Correlation between Income & Miles is 0.54.

[ ]:
```
# Product Analysis - Using pairplot

sns.pairplot(df, hue='Product', kind='reg')
plt.show()
```



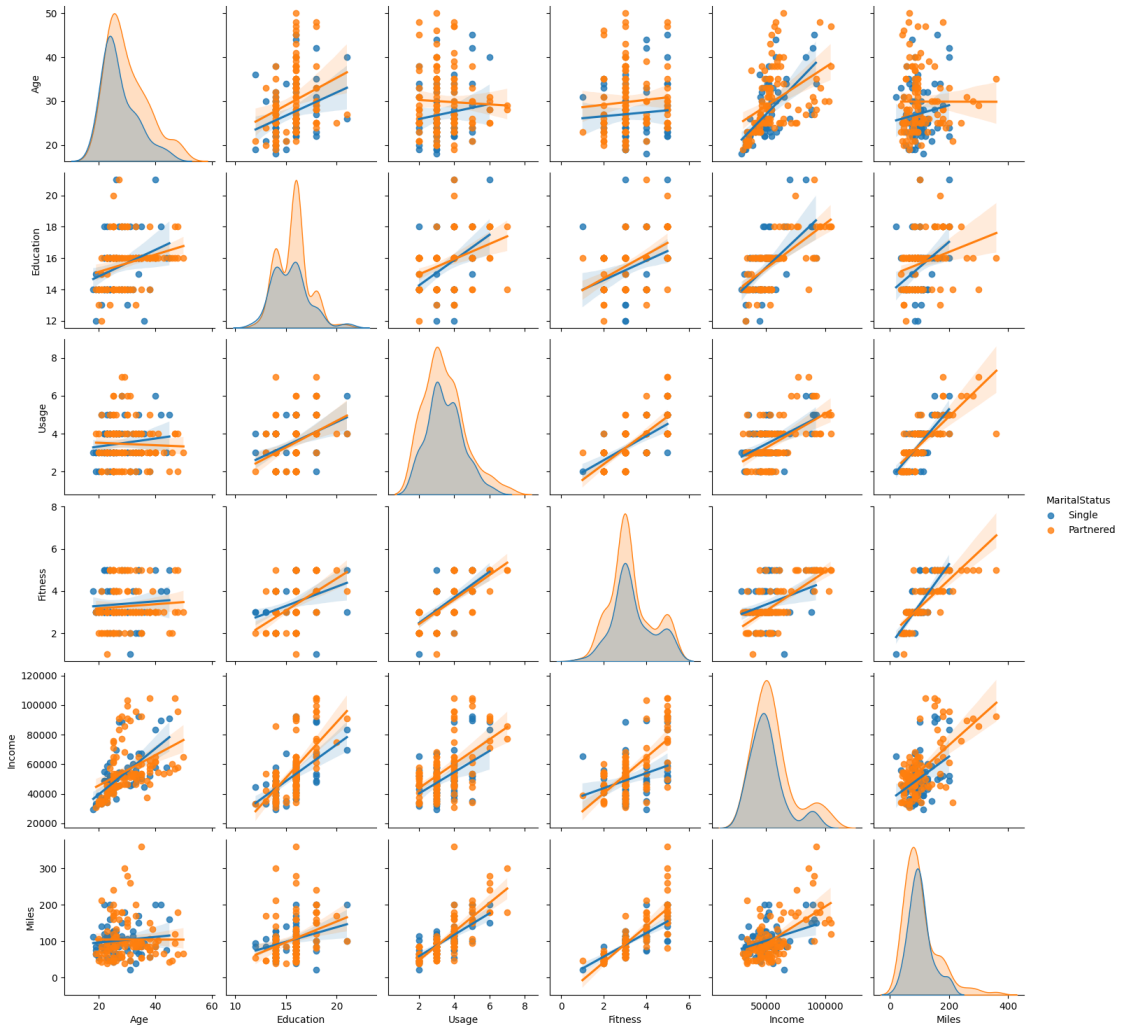
Similar correlation as heatmap.

[ ]:
```
# Gender Analysis - Using pairplot
sns.pairplot(df,hue='Gender',kind='reg')
plt.show()
```



Similar correlation as heatmap.

[ ]:
```
# Marital Status - Using pairplot
sns.pairplot(df,hue='MaritalStatus',kind='reg')
plt.show()
```



Similar correlation as heatmap.

Bivariate Analysis

[ ]:
```
# Average Income of customer buying each product type

df.groupby('Product')['Income'].mean()
```

[ ]:
```
Product
KP281    46418.025
KP481    48973.650
KP781    75441.575
Name: Income, dtype: float64
```

Mean Income of the customer who purchased product KP281 is 46.4k

Mean Income of the customer who purchased product KP481 is 48.9k

Mean Income of the customer who purchased product KP781 is 75.4k

[ ]:
```
# Average Age of customer buying each product type

df.groupby('Product')['Age'].mean()
```

[ ]:
```
Product
KP281    28.55
KP481    28.90
KP781    29.10
Name: Age, dtype: float64
```

Mean Age of the customer who purchased product KP281 is 28.55

Mean Age of the customer who purchased product KP481 is 28.90

Mean Age of the customer who purchased product KP781 is 29.10

[ ]:
```
# Average usage of each product type bought by customer

df.groupby('Product')['Usage'].mean()
```

[ ]:
```
Product
KP281    3.087500
KP481    3.066667
KP781    4.775000
Name: Usage, dtype: float64
```

[ ]:
```
# Average Fitness of customer buying each model

df.groupby('Product')['Fitness'].mean()
```

[ ]:
```
Product
KP281    2.9625
KP481    2.9000
KP781    4.6250
Name: Fitness, dtype: float64
```

Customer fitness mean for product KP281 is 2.96

Customer fitness mean for product KP481 is 2.90

Customer fitness mean for product KP781 is 4.62

[ ]:
```
# Average Education of customer using each product

df.groupby('Product')['Education'].mean()
```

[ ]:
```
Product
KP281    15.037500
```

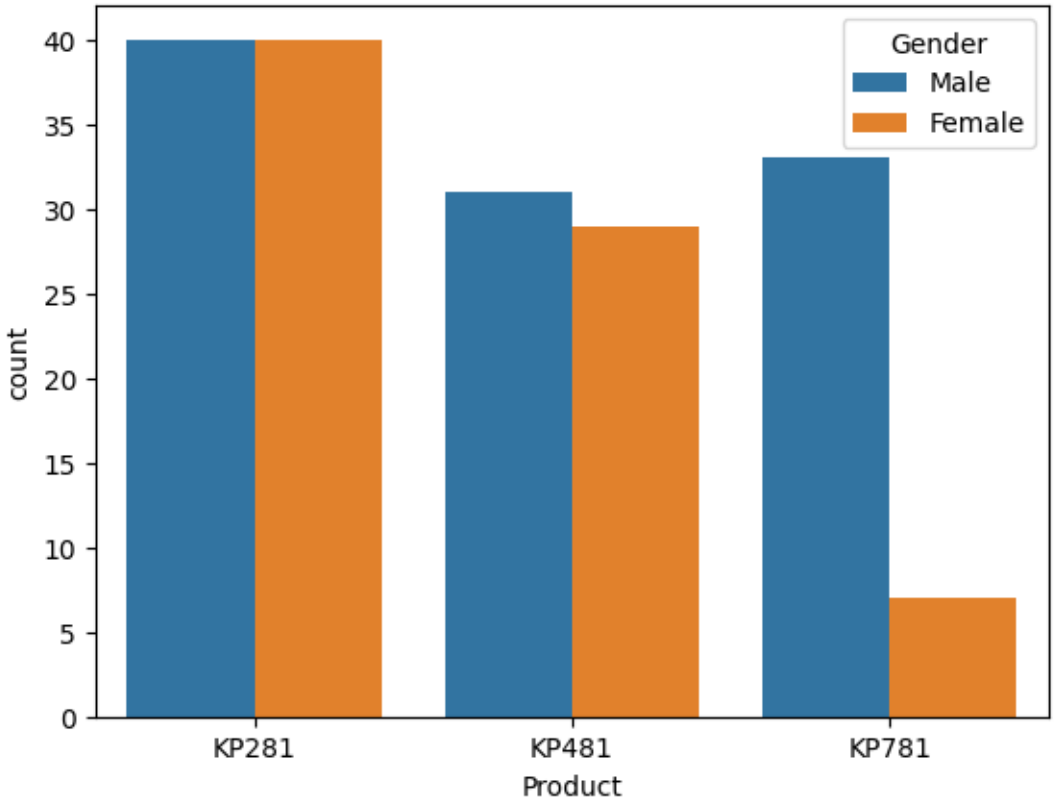
```
KP481    15.116667
KP781    17.325000
Name: Education, dtype: float64
```

Mean Education of the customer who purchased product KP281 is 15.03

Mean Education of the customer who purchased product KP481 is 15.11

Mean Education of the customer who purchased product KP781 is 17.32

[ ]:
```
# Gender & Product

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



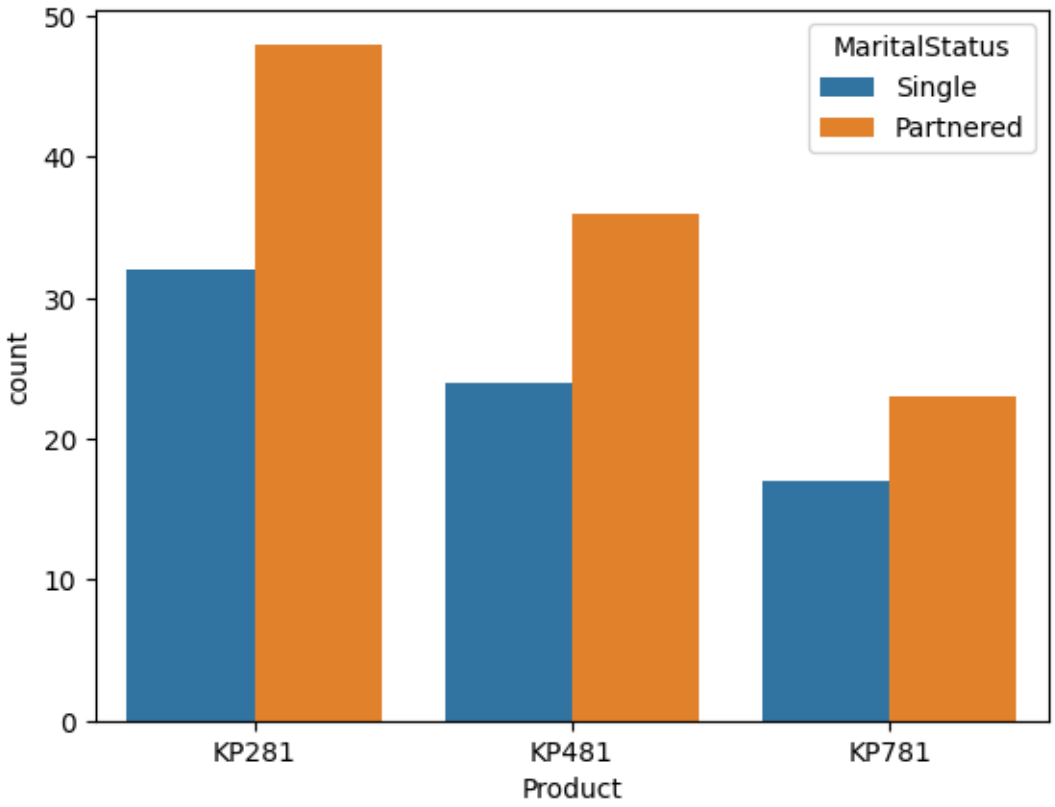
Most common preference for both gender is KP281.

Ratio of Male/Female customers is huge in KP781.

Distribution of Male & Female is roughly same for KP481.

Males have bought more KP781 compare to KP481.

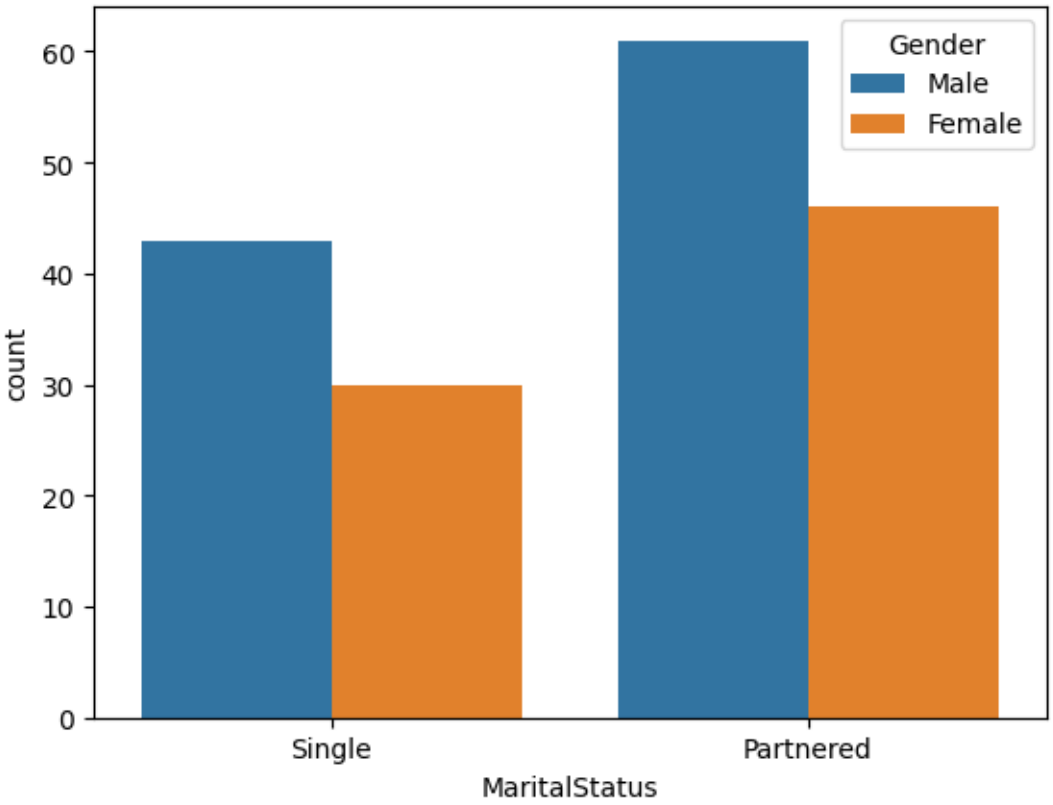
[ ]:
```
# Marital Status & Product

sns.countplot(data=df, x='Product', hue='MaritalStatus')
plt.show()
```



The count for singles is lower across all products, with the most significant difference observed in product "KP281". This suggests that singles show a lower interest in these products compared to partnered individuals.

Partnered customers are the major product purchasers.

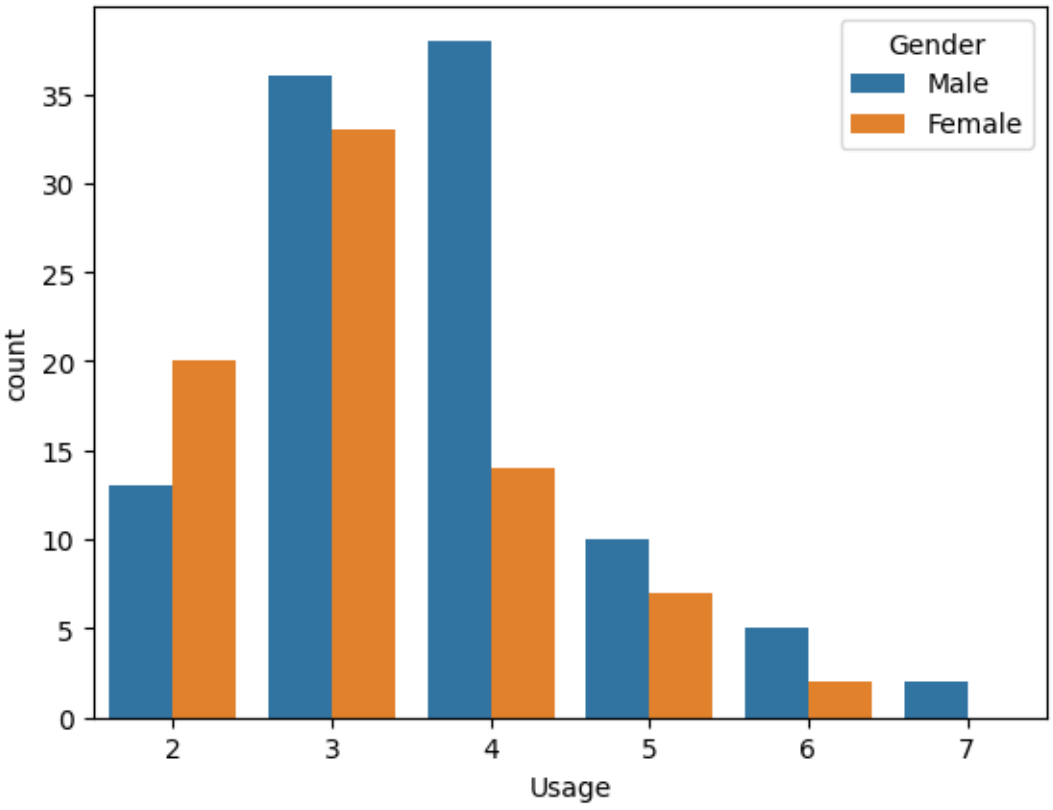
[ ]:
```
# Gender & Marital Status

sns.countplot(data=df,x='MaritalStatus',hue='Gender')
plt.show()
```



Out of both single & partnered customers, male customers are significantly high.

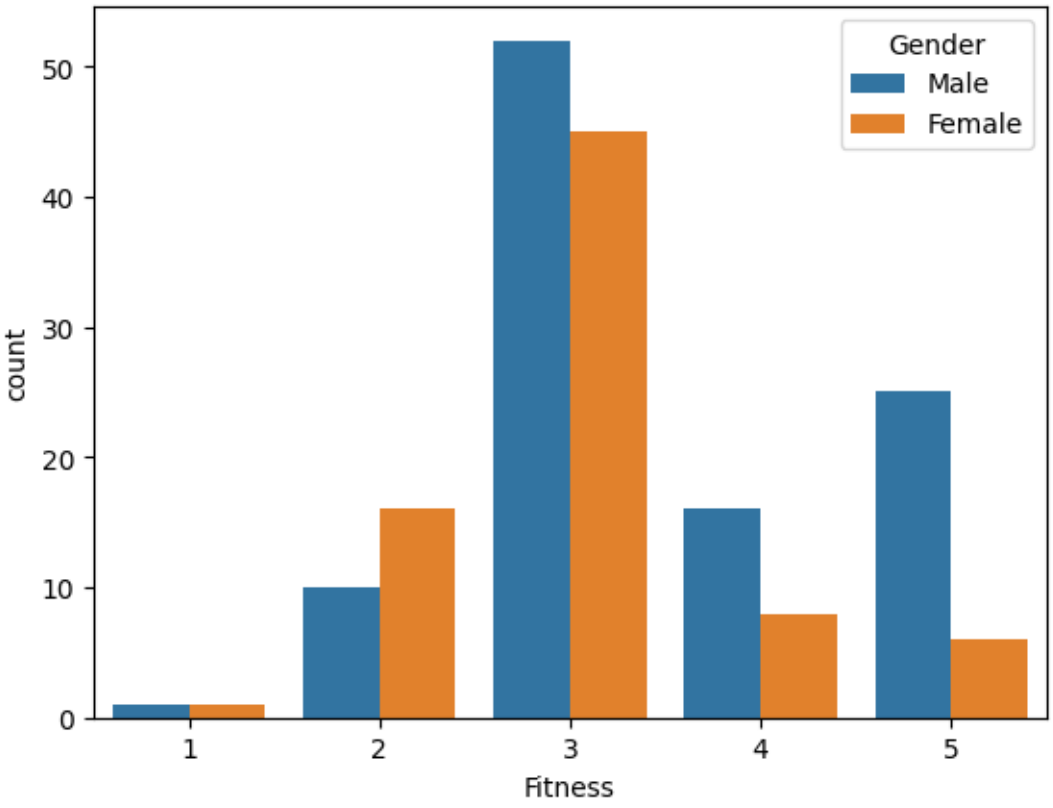
Female customers are low as compared to Male Customers.

[ ]:
```
# Product purchased among gender based on usage

sns.countplot(data=df,x='Usage',hue='Gender')
plt.show()
```



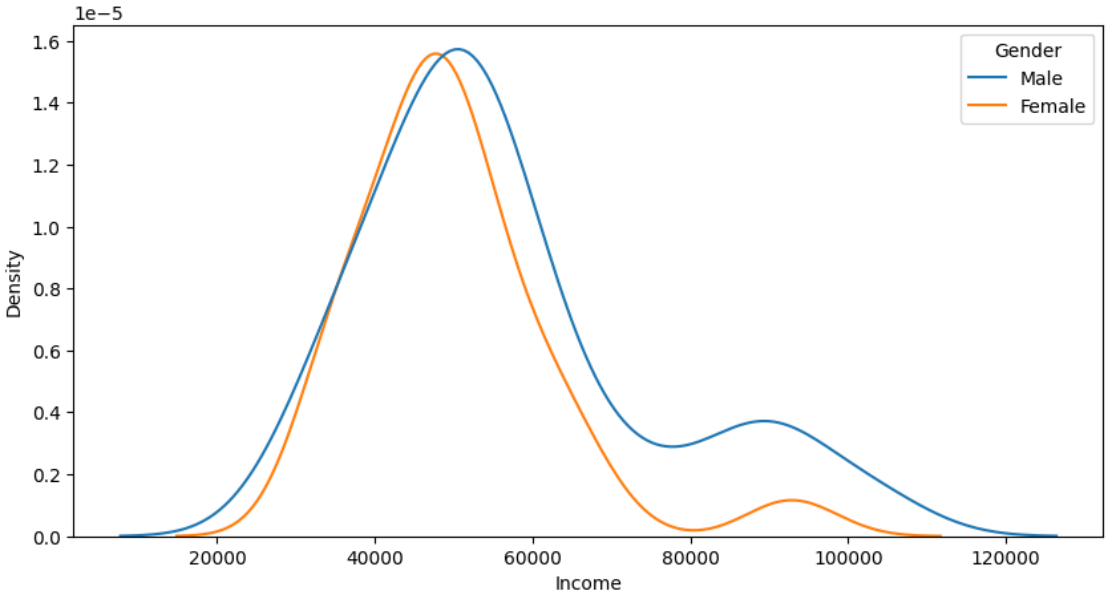
Only few Male customers use 7 times per week whereas female customer's maximum usage is only 6 times per week

[ ]:
```
# Fitness rating among the customers categorised by Gender

sns.countplot(data=df,x='Fitness',hue='Gender')
plt.show()
```



Among the fitness rating both Male and Female most of them have rated themselves as average

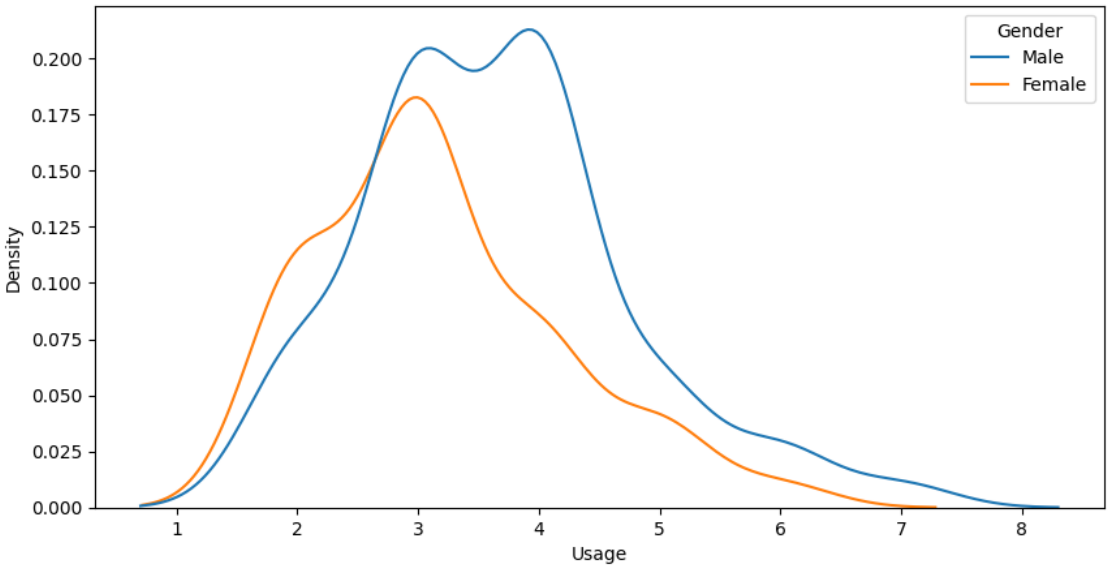
[ ]:
```
# Customers Income and Gender

plt.figure(figsize=(10,5))
sns.kdeplot(data=df, x='Income', hue='Gender')
plt.show()
```



Distribution of individuals earning less than 30k exhibits a similar pattern across both gender.

A large number of male customers have income exceeding 75k as compared to female customers.

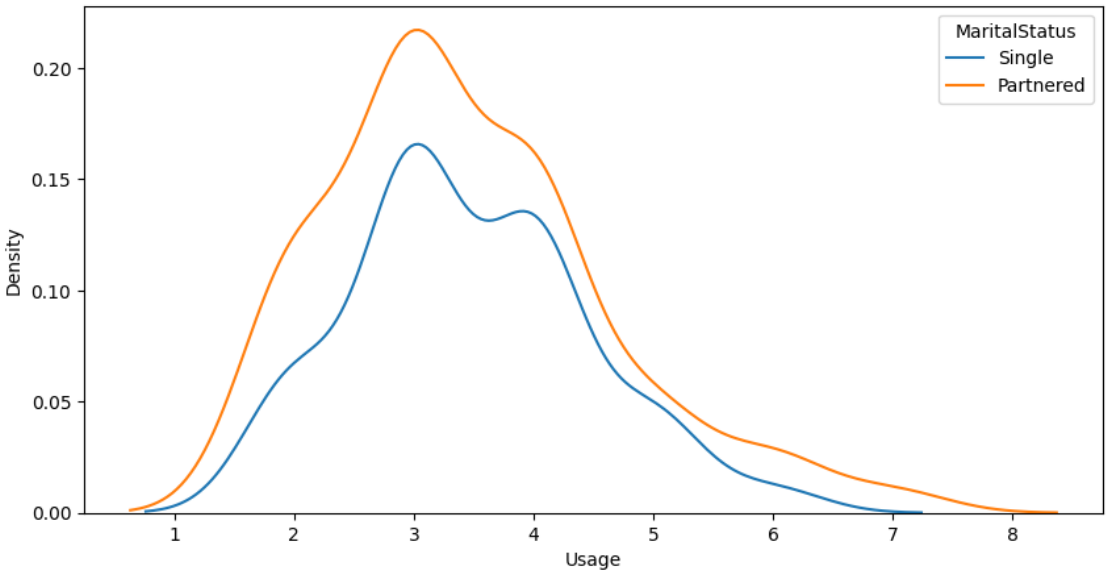
[ ]:
```
# Customers Usage per week and their Gender

plt.figure(figsize=(10,5))
sns.kdeplot(data=df,x='Usage',hue='Gender')
plt.show()
```



Male customer's usage is significantly higher than female customers.

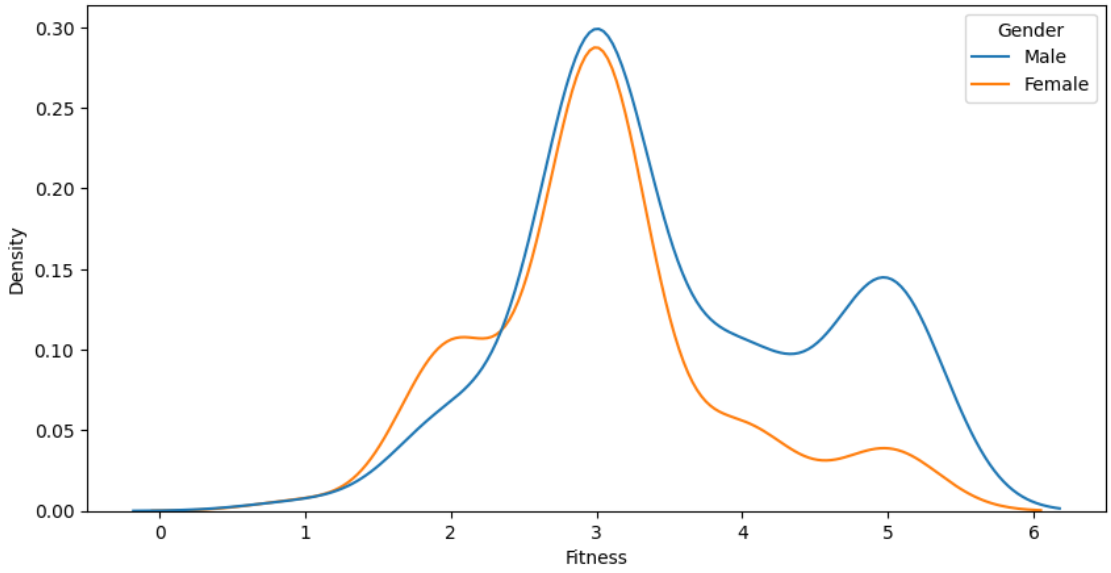
Female customer's usage drops down after 3 times per week.

[ ]:
```
# Customers Usage per week and their Marital Status

plt.figure(figsize=(10,5))
sns.kdeplot(data=df,x='Usage',hue='MaritalStatus')
plt.show()
```



Usage of customers having marital status : partnered is higher than single customers.

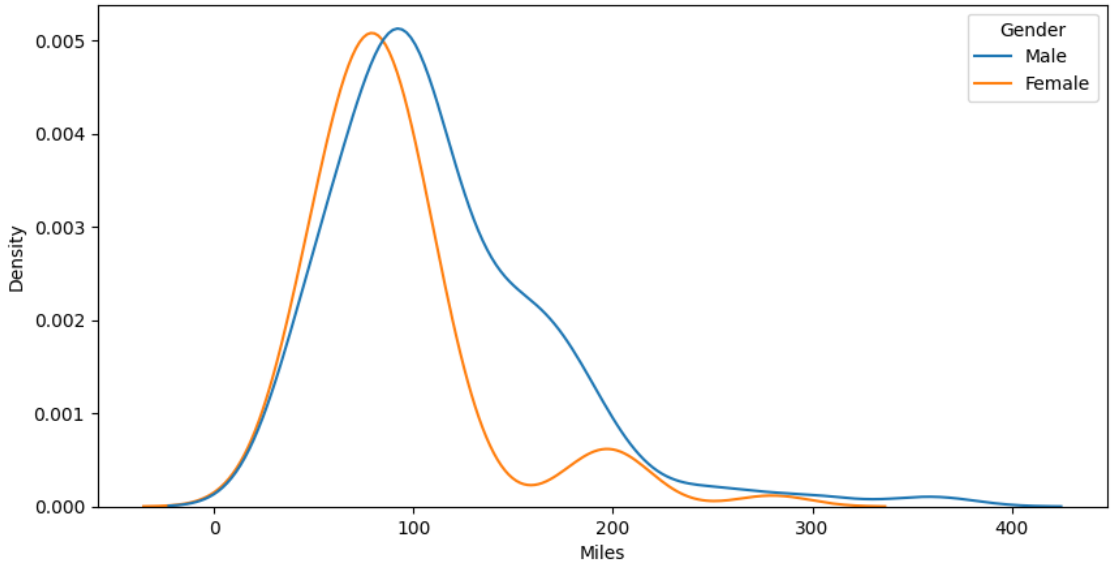
[ ]:
```
# Customers Fitness Rating and their Gender

plt.figure(figsize=(10,5))
sns.kdeplot(data=df,x='Fitness',hue='Gender')
plt.show()
```



Male customers are in better shape than female customers.

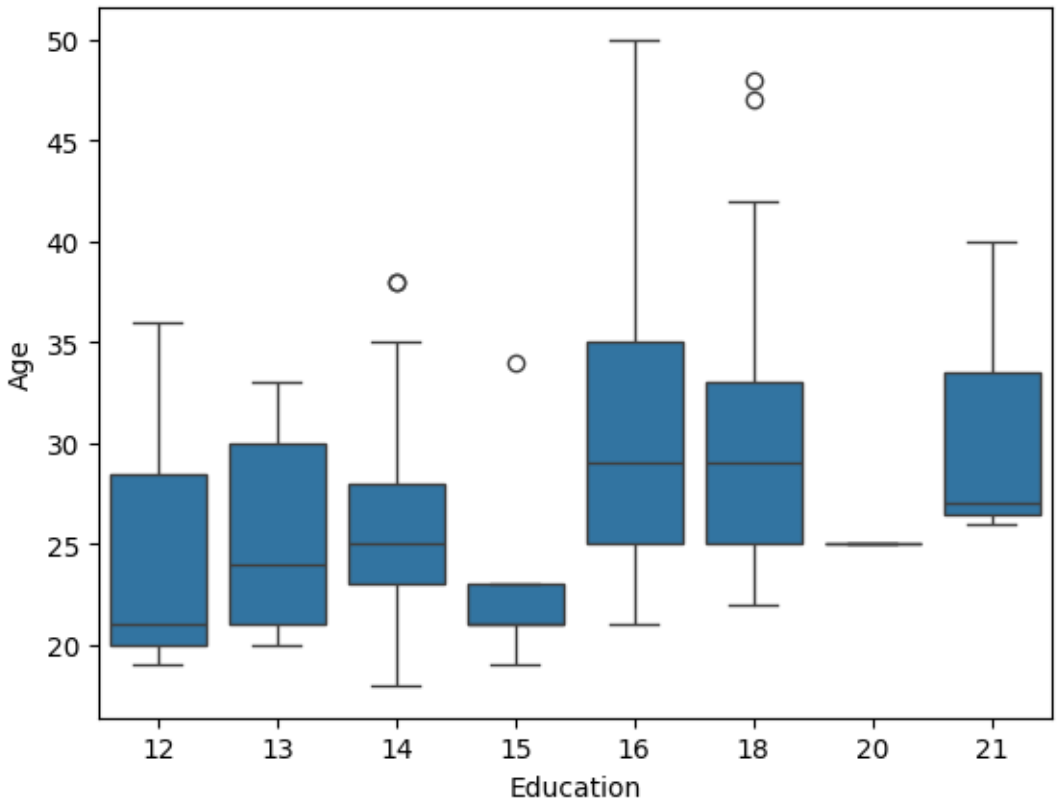
There are more male customers in excellent shape as compared to female customers.

[ ]:
```
# Miles covered by each Gender among the customers

plt.figure(figsize=(10,5))
sns.kdeplot(data=df,x='Miles',hue='Gender')
plt.show()
```

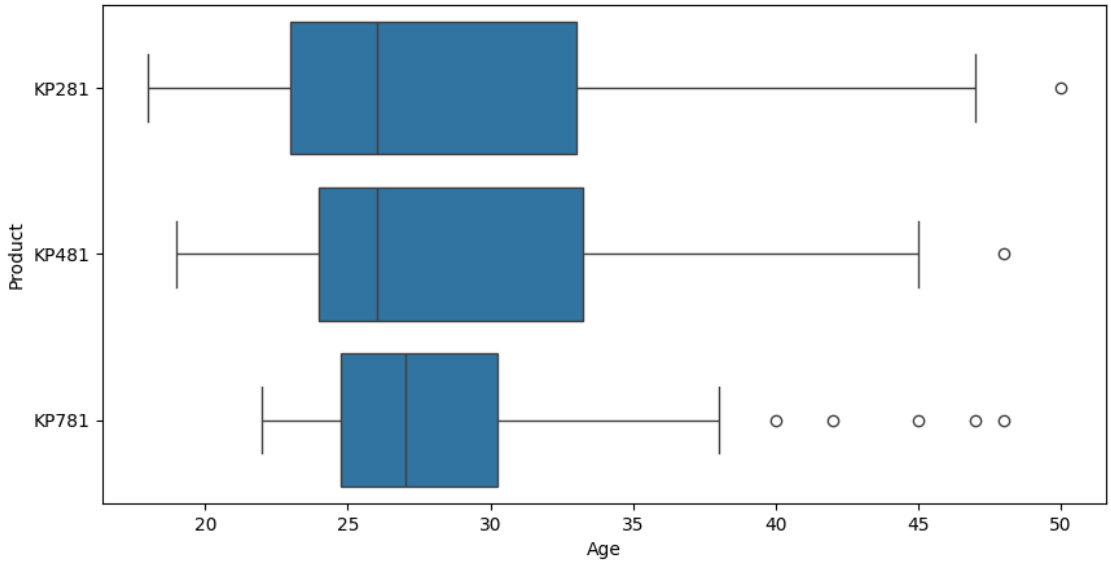


Female customers have max distance covered as just over 300 miles, whereas Male customers have a little over 400 miles.

[ ]:
```
# Education data against Age of the customer : Using boxplot

sns.boxplot(x='Education',y='Age',data=df)
plt.show()
```



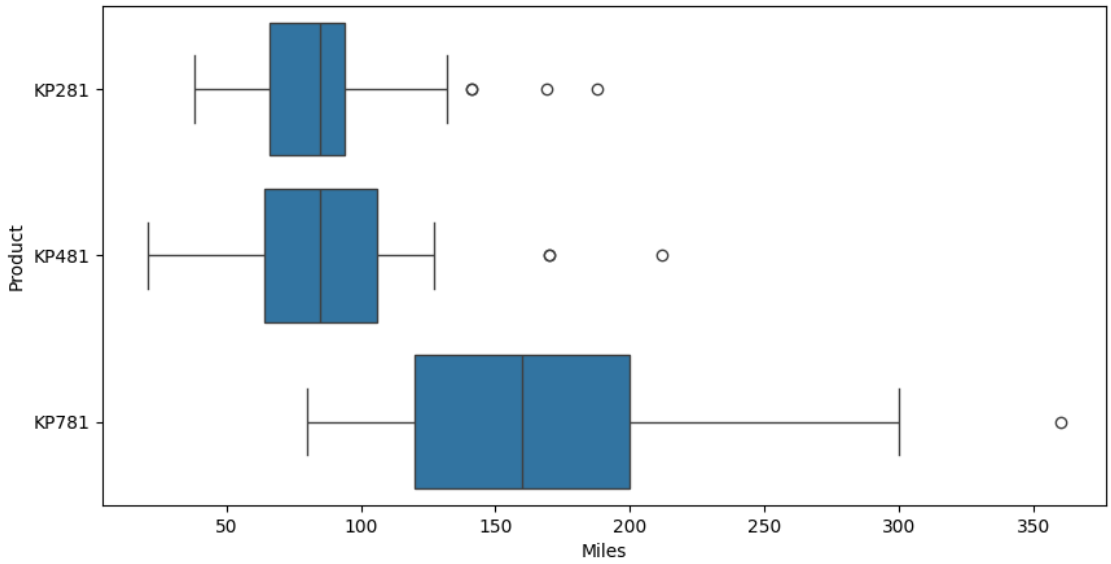
[ ]:
```
# Age data against product used by the customer : Using boxplot

plt.figure(figsize=(10,5))
sns.boxplot(x='Age',y='Product',data=df)
plt.show()
```



Only a few customers with age above 40 use KP781.

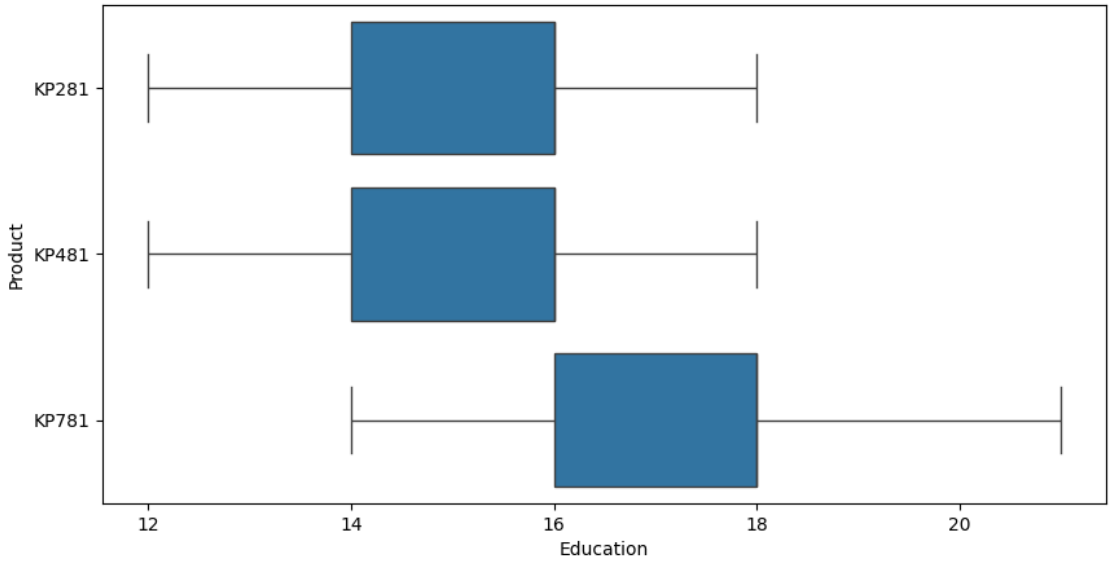
Most of the customers are comfortable with KP281.

[ ]:
```
# Miles data against product used by the customer : Using boxplot

plt.figure(figsize=(10,5))
sns.boxplot(x='Miles',y='Product',data=df)
plt.show()
```



Customers having KP781 cover more miles than other two product types, followed by KP481 & KP281.

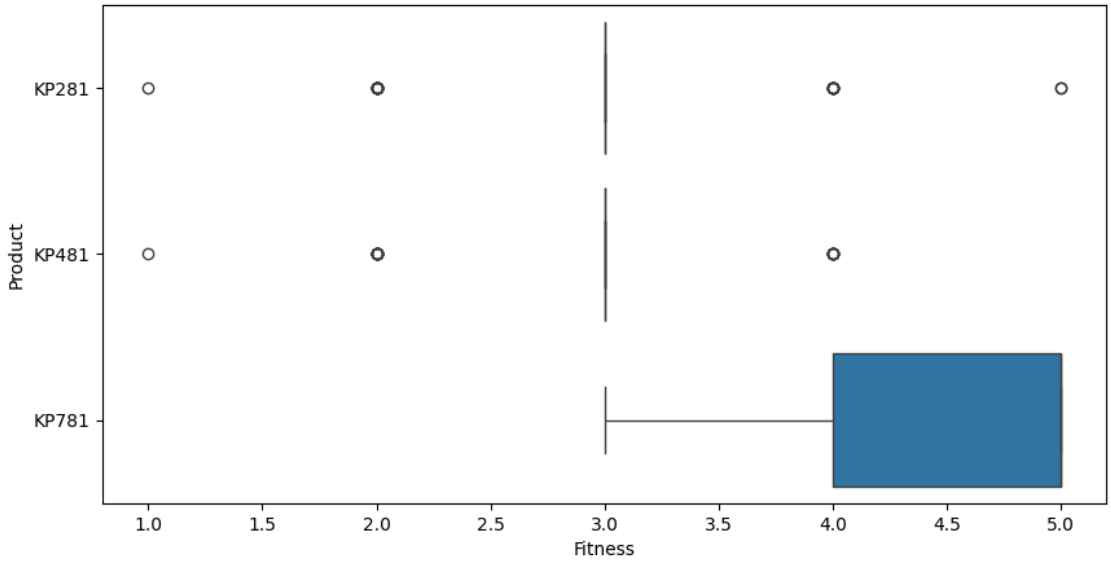
[ ]:
```
# Education data against Product of the customer : Using boxplot

plt.figure(figsize=(10,5))
sns.boxplot(x='Education',y='Product',data=df)
plt.show()
```



Customers with education between 14 to 16 prefers KP281 & KP481 equally.

Customers with education between 16 to 18 prefers KP781.

[ ]:
```
# Fitness data against Product of the customer : Using boxplot

plt.figure(figsize=(10,5))
sns.boxplot(x='Fitness',y='Product',data=df)
plt.show()
```



Customers who rated themselves between 4 to 5 are mostly using KP781.

KP281 & KP481 are scattered across the ratings.

Missing Value & Outlier Detection

Missing Value

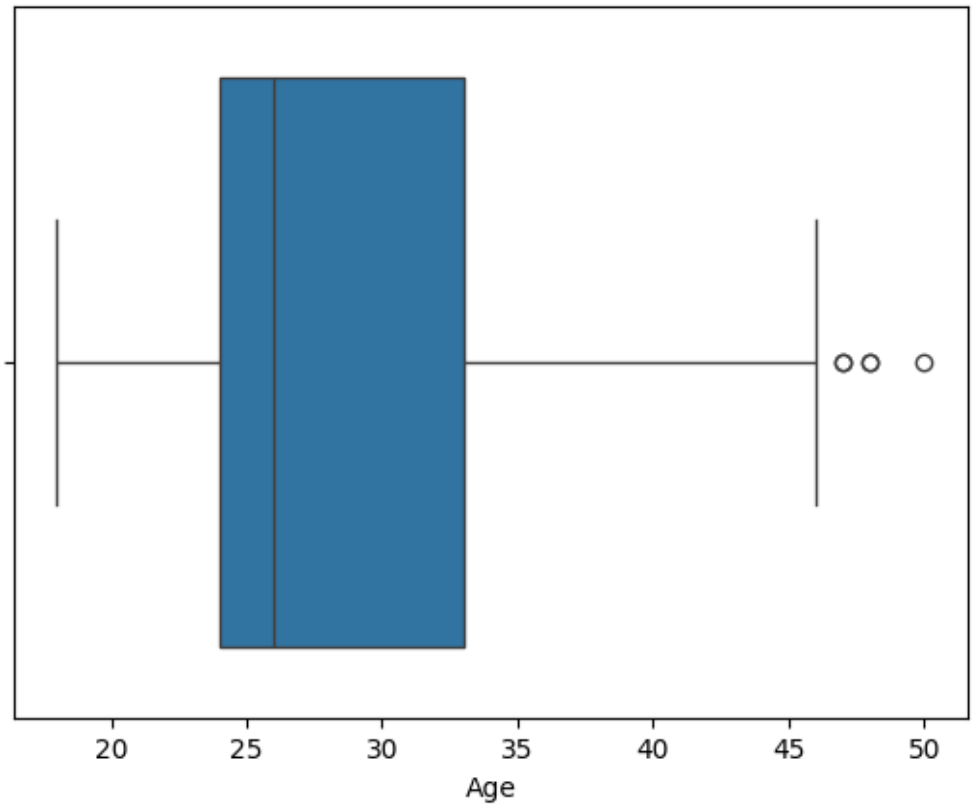
[ ]:
```
# checking for missing values
missing_values = df.isnull().sum()
print(missing_values)
```

```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

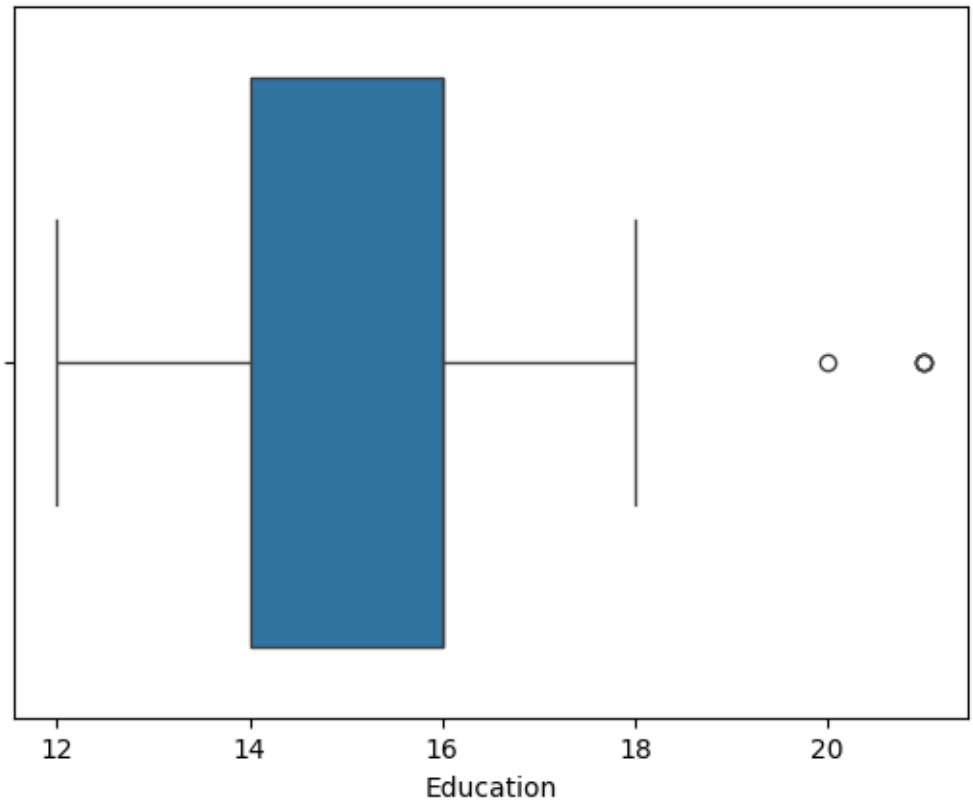
No missing values have been found in any of the columns.

Outlier Detection

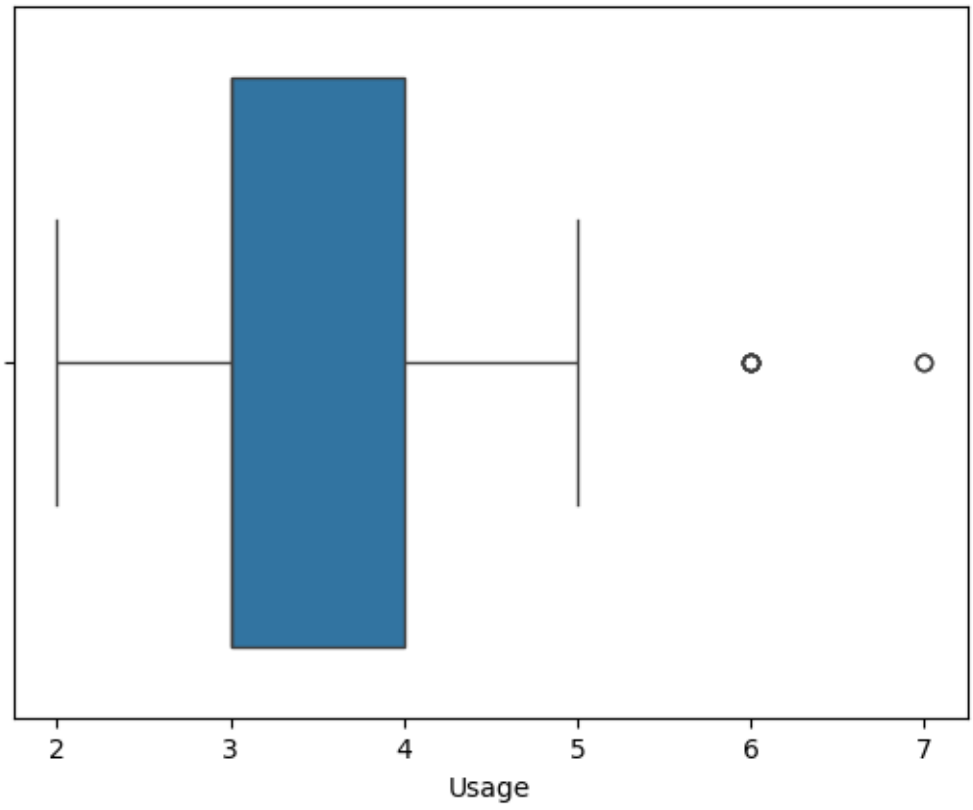
[ ]:
```
# Outlier Detection Using Boxplot

sns.boxplot(data=df, x='Age')
plt.show()
```



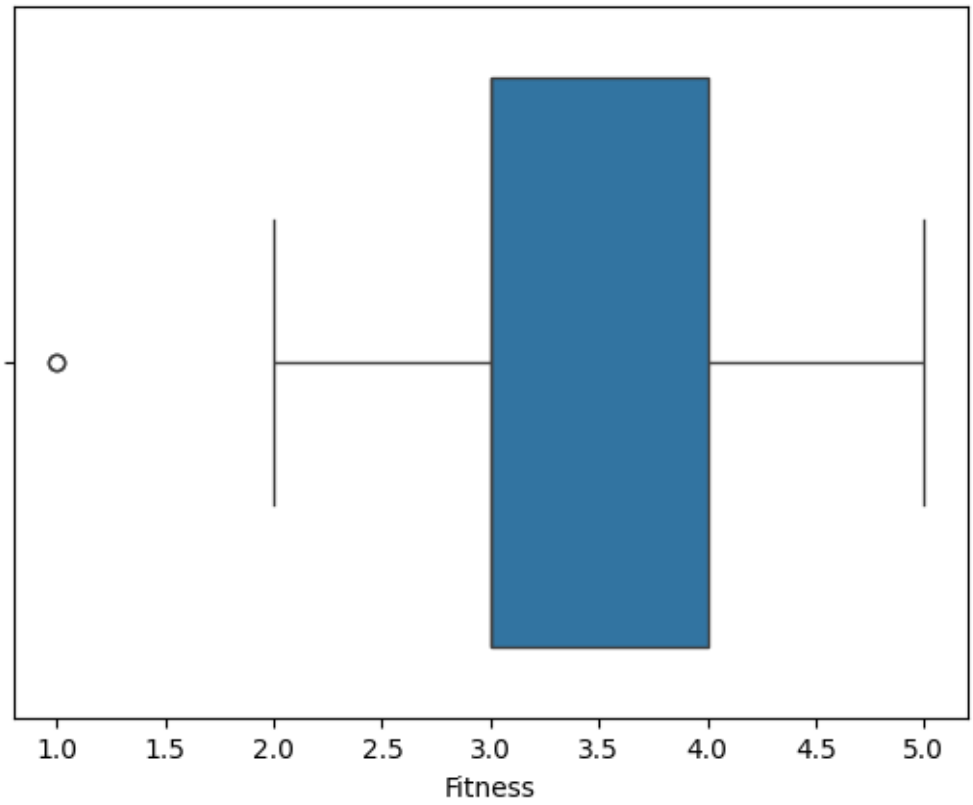
[ ]:
```
sns.boxplot(data=df, x='Education')
plt.show()
```



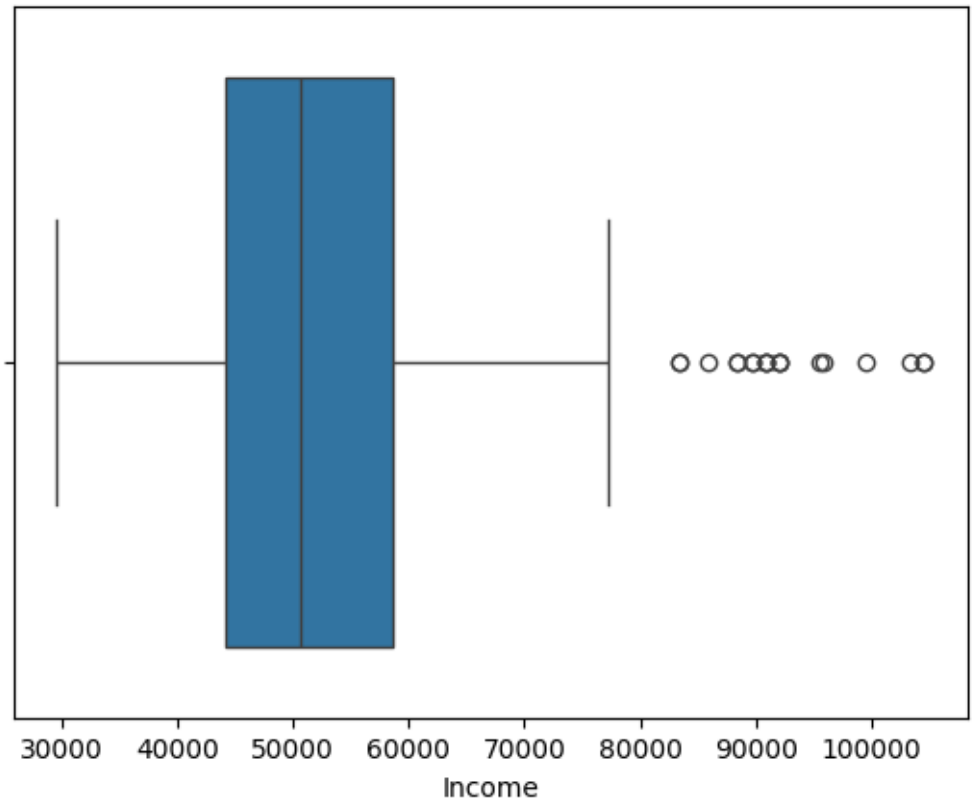
[ ]:
```
sns.boxplot(data=df, x='Usage')
plt.show()
```



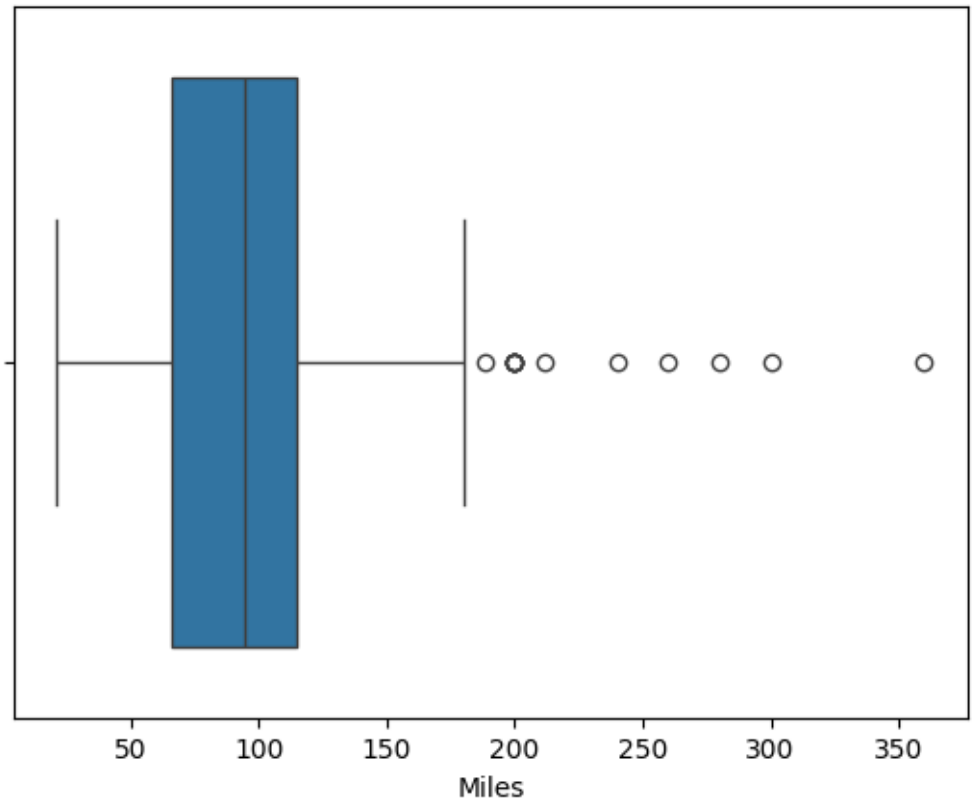
[ ]:
```
sns.boxplot(data=df, x='Fitness')
plt.show()
```



[ ]:
```
sns.boxplot(data=df, x='Income')
plt.show()
```



[ ]:
```
sns.boxplot(data=df, x='Miles')
plt.show()
```



Observation:

Age, Education, Fitness & Usage are having very few outliers.

While Income & Miles are having more outliers.

Business Insights based on Non-Graphical and Visual Analysis

[ ]:
```
df.Product.value_counts(normalize=True)
```

[ ]:
```
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: Product, dtype: float64
```

Probability of buying KP281, KP481 & KP781 are 0.44, 0.33 & 0.22 respectively

[ ]:
```
df.Gender.value_counts(normalize=True)
```

[ ]:
```
Male      0.577778
Female    0.422222
Name: Gender, dtype: float64
```

Probability of Male customer is 0.578, Probability of Female customer is 0.422.

[ ]:
```
df.MaritalStatus.value_counts(normalize=True)
```

[ ]:
```
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
```

Probability of Partnered Individual is 0.595, Probability of Single is 0.405

Conditional Probability, P (Product | Gender)

[ ]:
```
# Probability of buying KP281 given that the customer is male
pd.crosstab(df['Product'], df['Gender'], normalize='index').loc['KP281', 'Male']
```

[ ]:
```
0.5
```

[ ]:
```
# Probability of buying KP281 given that the customer is female
pd.crosstab(df['Product'], df['Gender'], normalize='index').loc['KP281',␣
↪'Female']
```

[ ]:
```
0.5
```

[ ]:
```
# Probability of buying KP481 given that the customer is male
pd.crosstab(df['Product'], df['Gender'], normalize='index').loc['KP481', 'Male']
```

[ ]:
```
0.5166666666666667
```

[ ]:
```
# Probability of buying KP481 given that the customer is Female
pd.crosstab(df['Product'], df['Gender'], normalize='index').loc['KP481',␣
↪'Female']
```

[ ]:
```
0.48333333333333334
```

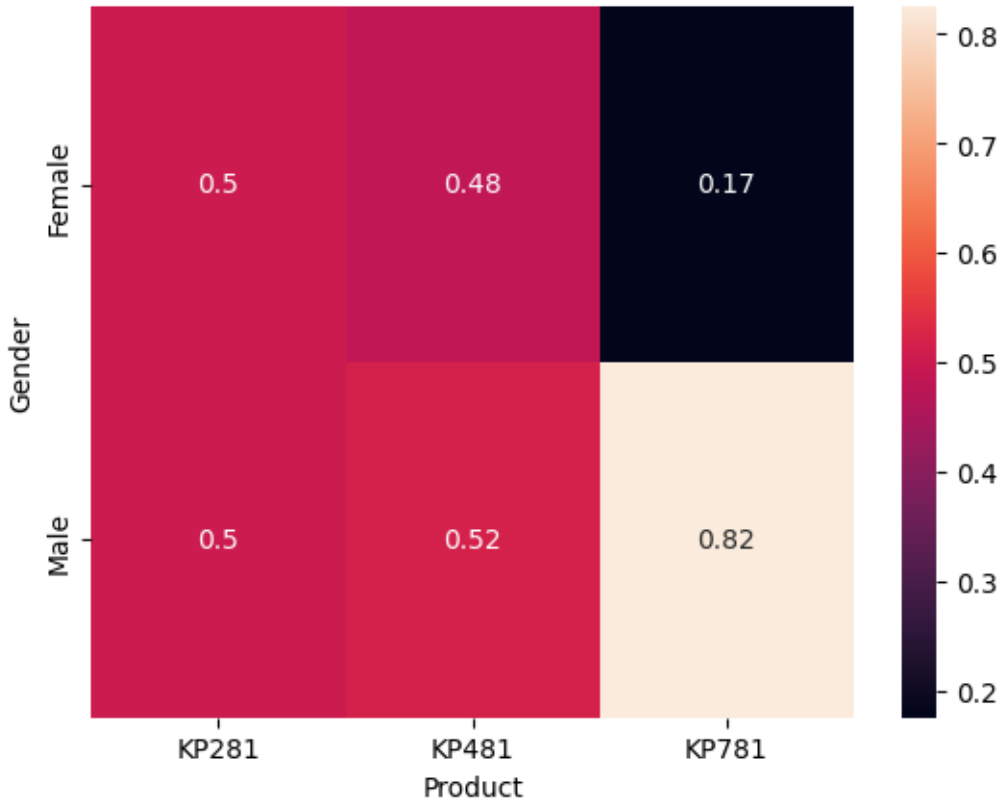
[ ]:
```
# Probability of buying KP781 given that the customer is male
pd.crosstab(df['Product'], df['Gender'], normalize='index').loc['KP781', 'Male']
```

[ ]:
```
0.825
```

[ ]:
```
# Probability of buying KP781 given that the customer is Female
pd.crosstab(df['Product'], df['Gender'], normalize='index').loc['KP781',␣
↪'Female']
```

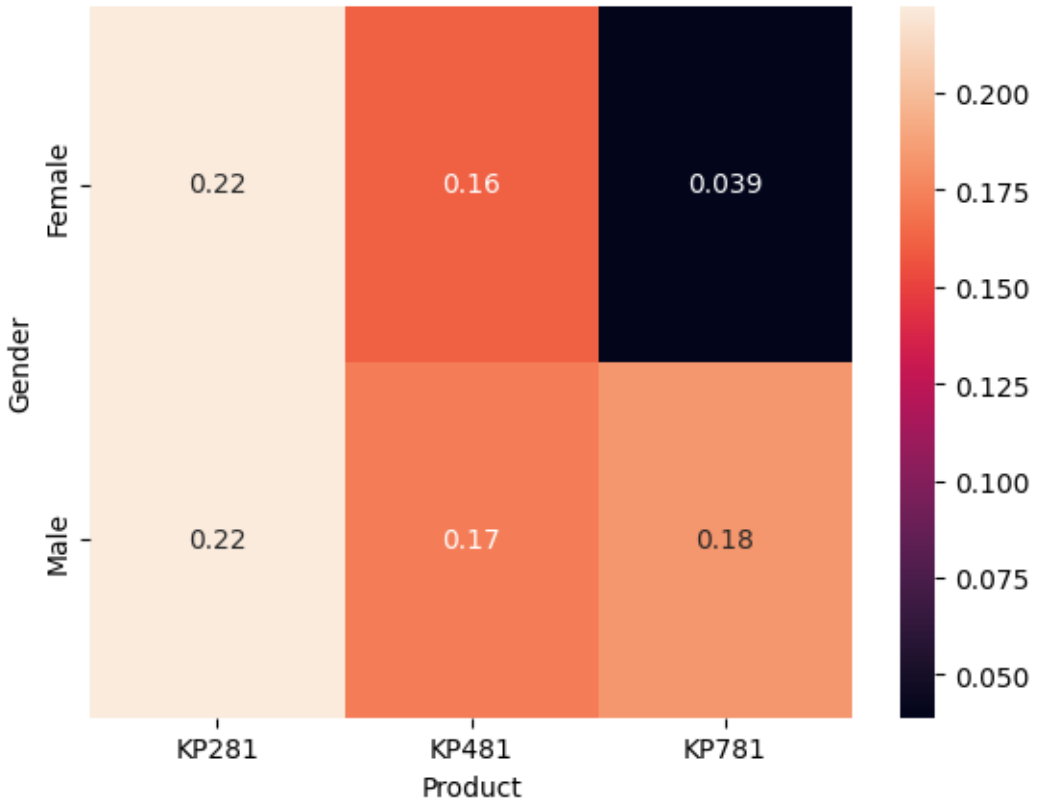
[ ]:
```
0.175
```

[ ]:
```
# all above probabilies shown with the help of heatmap.

sns.heatmap(pd.crosstab(df['Gender'], df['Product'], normalize='columns'),␣
↪annot=True)
plt.show()
```



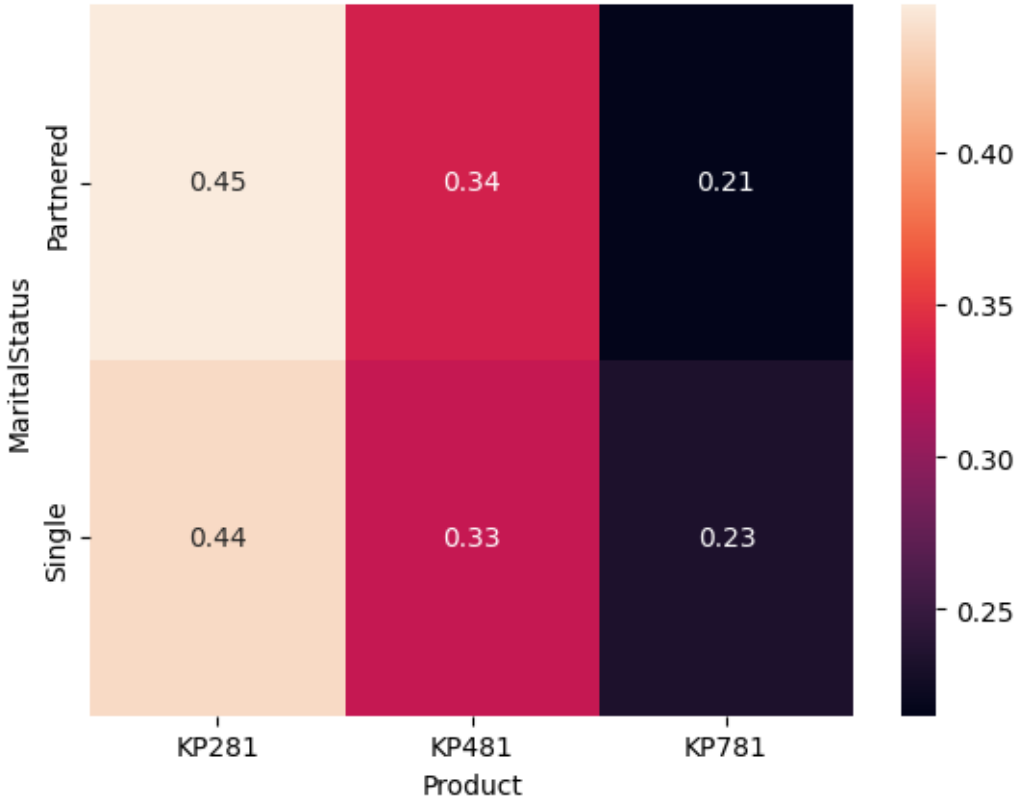
Conditional Probability, P(Gender | Product)

[ ]:
```
sns.heatmap(pd.crosstab(df['Gender'], df['Product'], normalize=True),␣
↪annot=True)
plt.show()
```



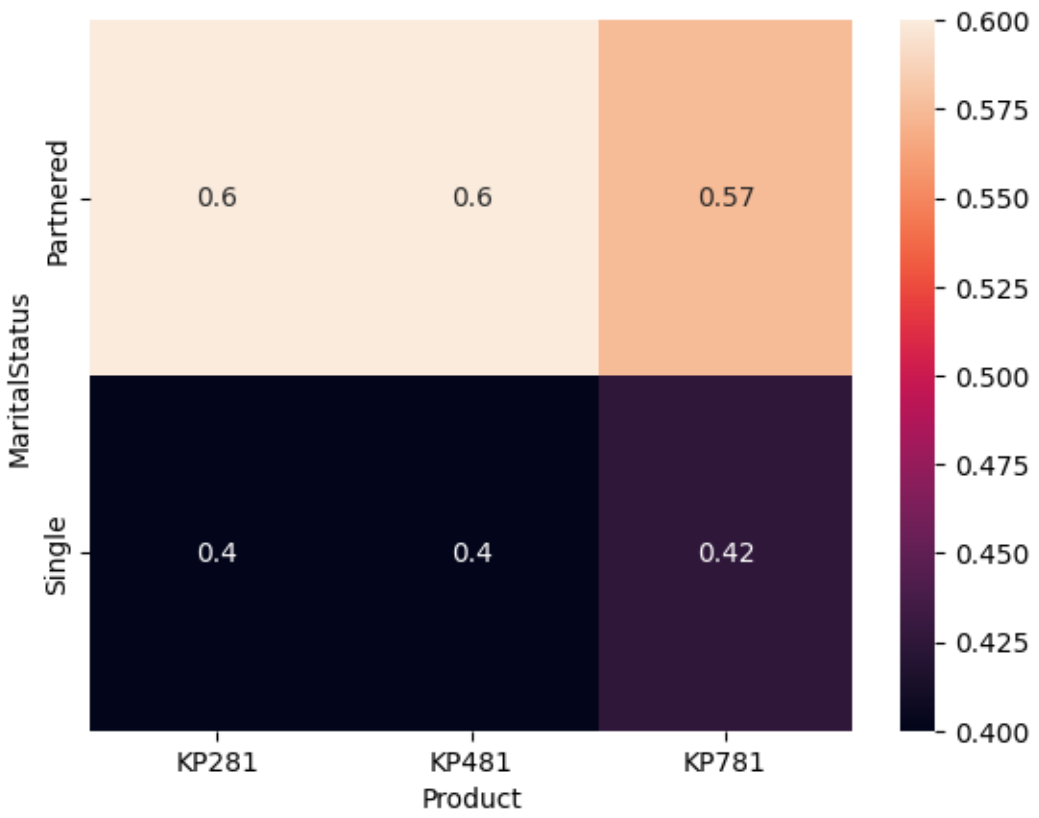
Joint Probability, P(Product Intersection Gender)

[ ]:
```
sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'], normalize='index'),␣
↪annot=True)
plt.show()
```



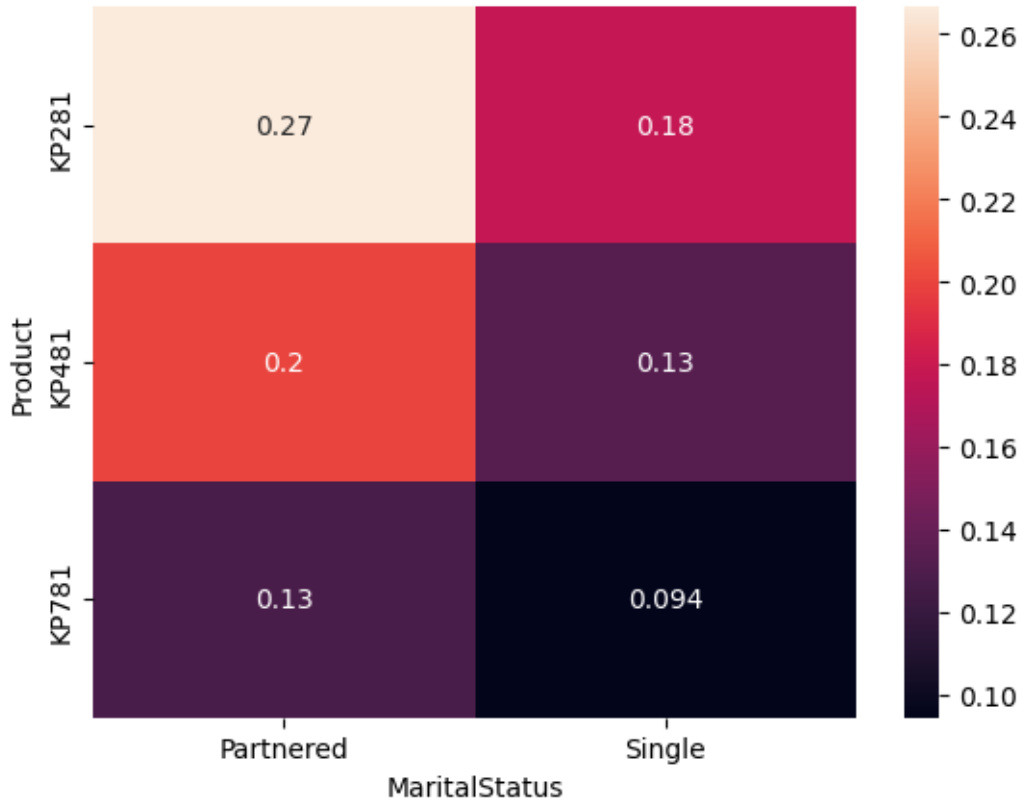
Conditional Probability, P(Product | MaritalStatus)

[ ]:
```
sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'],␣
 ↪normalize='columns'), annot=True)
plt.show()
```



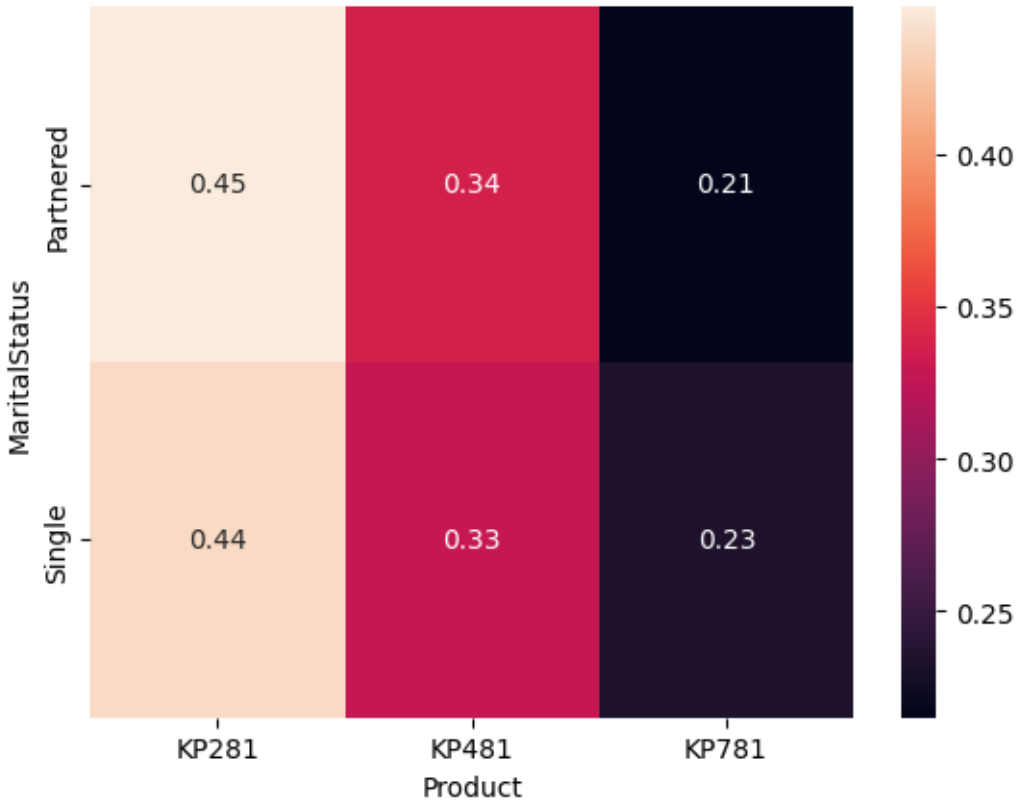
Conditional Probability P(MaritalStatus | Product)

[ ]:
```
sns.heatmap(pd.crosstab(df['Product'], df['MaritalStatus'], normalize=True),␣
↪annot=True)
plt.show()
```



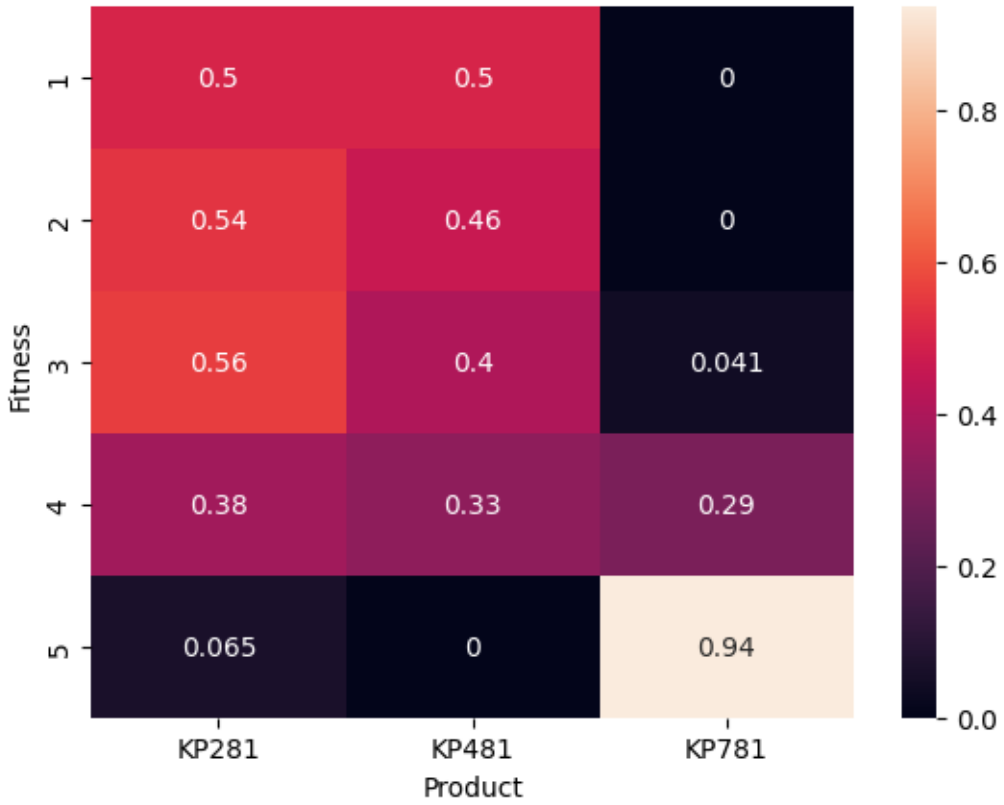
Joint Probability, P(Product Intersection MartialStatus)

[ ]:
```
sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'], normalize='index'),␣
↪annot=True)
plt.show()
```



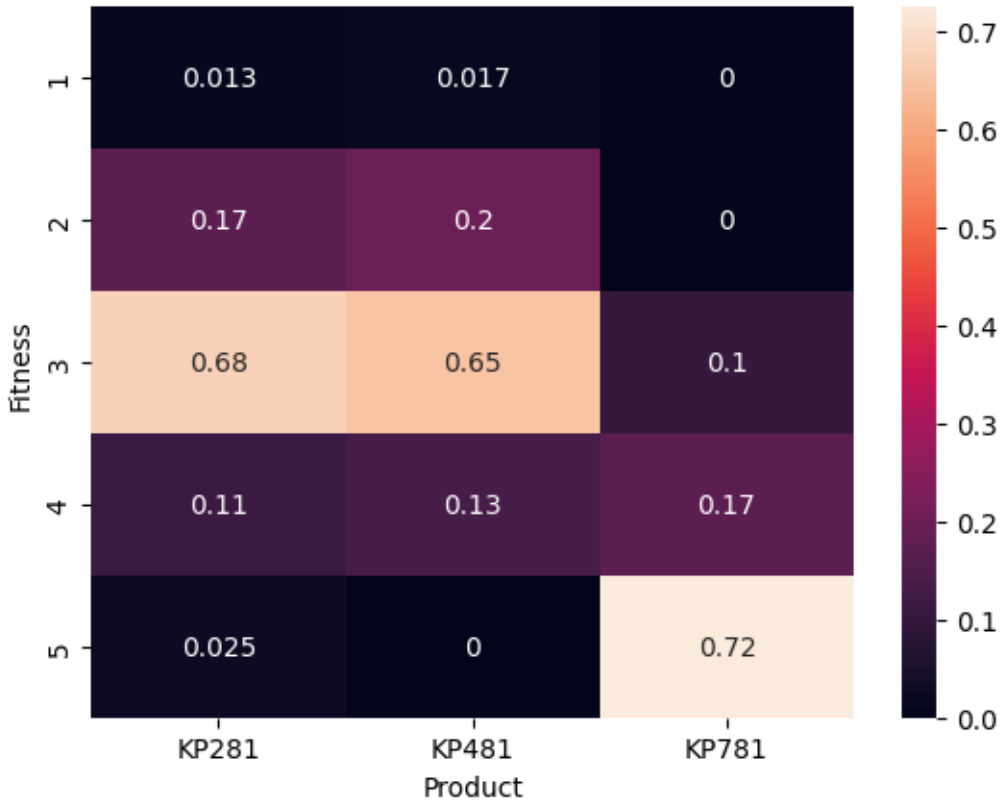
Conditional Probability P(Product | Fitness)

[ ]:
```
sns.heatmap(pd.crosstab(df['Fitness'], df['Product'], normalize='index'),␣
↪annot=True)
plt.show()
```



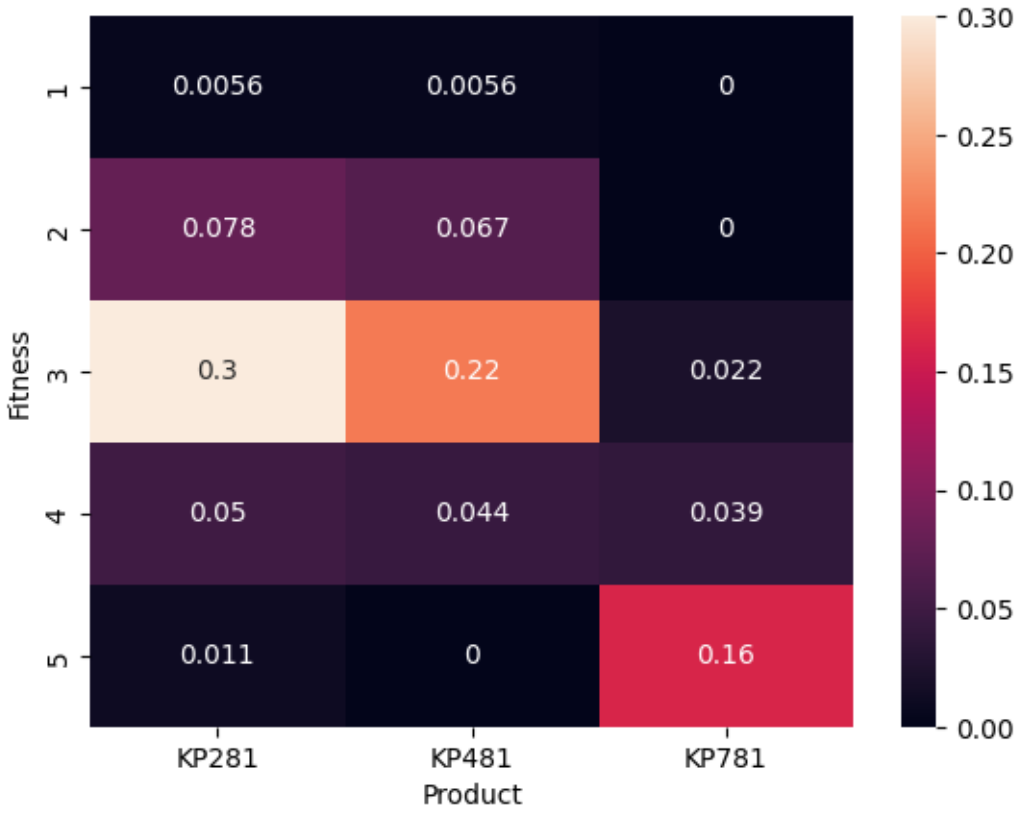
Conditional Probability P(Fitness | Product)

[ ]:
```
sns.heatmap(pd.crosstab(df['Fitness'], df['Product'], normalize='columns'),␣
↪annot=True)
plt.show()
```



Joint Probability P(Product Intersection Fitness)

[ ]:
```
sns.heatmap(pd.crosstab(df['Fitness'], df['Product'], normalize=True),␣
↪annot=True)
plt.show()
```



Marginal Probability P (Product)

[ ]:
```
df['Product'].value_counts(normalize=True)
```

[ ]:
```
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: Product, dtype: float64
```

Probability of buying KP281 treadmill, P(Product=KP281) = 0.44.

Probability of buying KP481 treadmill, P(Product=KP481) = 0.33.

Probability of buying KP781 treadmill, P(Product=KP781) = 0.22.

Marginal Probability P (Gender)

[ ]:
```
df['Gender'].value_counts(normalize=True)
```

[ ]:
```
Male      0.577778
Female    0.422222
Name: Gender, dtype: float64
```

Probability of customer gender is Male, P(Gender=Male) = 0.58

Probability of customer gender is Female, P(Gender=Female) = 0.42

Marginal Probability P (MaritalStatus)

[ ]:
```
df['MaritalStatus'].value_counts(normalize=True)
```

[ ]:
```
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
```

Probability of customer's MaritalStatus is Partnered, P(MaritalStatus=Partnered) = 0.60.

Probability of customer's MaritalStatus is Single, P(MaritalStatus=Single) = 0.40.

Marginal Probability P (Fitness)

[ ]:
```
df['Fitness'].value_counts(normalize=True).sort_index()
```

[ ]:
```
1    0.011111
2    0.144444
3    0.538889
4    0.133333
5    0.172222
Name: Fitness, dtype: float64
```

Probability of customer having fitness rating of 1 is P(Fitness=1) = 0.01.

Probability of customer having fitness rating of 2 is P(Fitness=2) = 0.14.

Probability of customer having fitness rating of 3 is P(Fitness=3) = 0.53.

Probability of customer having fitness rating of 4 is P(Fitness=4) = 0.13.

Probability of customer having fitness rating of 5 is P(Fitness=5) = 0.17.

Marginal Probability P (Usage)

[ ]:
```
df['Usage'].value_counts(normalize=True).sort_index()
```

[ ]:
```
2    0.183333
3    0.383333
4    0.288889
5    0.094444
6    0.038889
7    0.011111
Name: Usage, dtype: float64
```

Probability of customer using 2 times per week is P(Usage=2) = 0.18.

Probability of customer using 3 times per week is P(Usage=3) = 0.38.

Probability of customer using 4 times per week is P(Usage=4) = 0.29.

Probability of customer using 5 times per week is P(Usage=5) = 0.09.

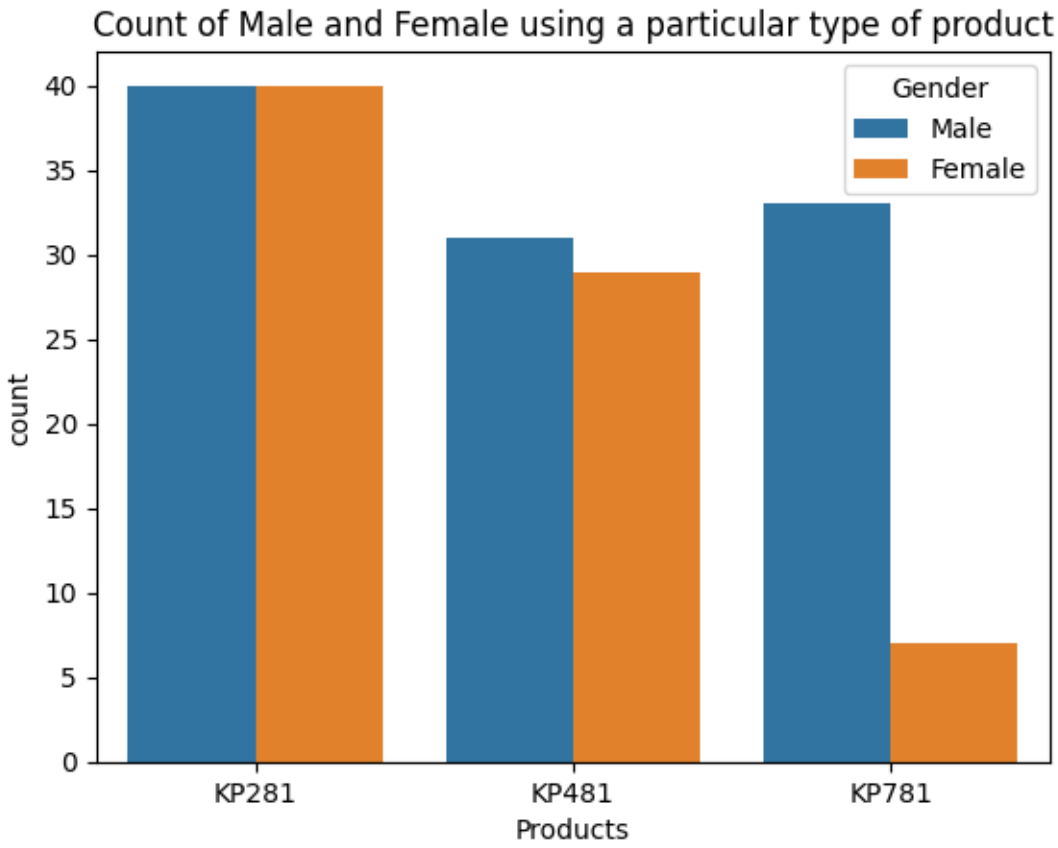
Probability of customer using 6 times per week is P(Usage=6) = 0.03.

Probability of customer using 7 times per week is P(Usage=7) = 0.01.

Using Two-Way Contingency Table:

Marginal Probability

[ ]:
```
sns.countplot(x = "Product", data= df, hue = "Gender")
plt.xlabel("Products")
plt.title("Count of Male and Female using a particular type of product")
plt.show()
```



[ ]:
```
pd.crosstab([df.Product],df.Gender,margins=True)
```

[ ]:
```
Gender   Female  Male  All
Product
KP281        40    40   80
KP481        29    31   60
KP781         7    33   40
```

```
All          76   104  180
```

[ ]:
```
np.round(((pd.crosstab(df.Product,df.Gender,margins=True))/180)*100,2)
```

[ ]:
```
Gender   Female   Male     All
Product
KP281     22.22  22.22   44.44
KP481     16.11  17.22   33.33
KP781      3.89  18.33   22.22
All       42.22  57.78  100.00
```

Probability of Male Customer Purchasing any product is : 57.77 %

Probability of Female Customer Purchasing any product is : 42.22 %

Marginal Probability of any customer buying:

product KP281 is : 44.44 %

product KP481 is : 33.33 %

product KP781 is : 22.22 %

Conditional Probabilies

[ ]:
```
np.round((pd.crosstab([df.Product],df.
  ↪Gender,margins=True,normalize="columns"))*100,2)
```

[ ]:
```
Gender   Female   Male    All
Product
KP281     52.63  38.46  44.44
KP481     38.16  29.81  33.33
KP781      9.21  31.73  22.22
```

Probability of Selling Product:

- KP281 | Female = 52 %
- KP281 | Male = 38 %
- KP481 | Female = 38 %
- KP481 | Male = 30 %
- KP781 | Female = 9 %
- KP781 | Male = 32 %

Customer Profiling for Each Product

Customer profiling based on the 3 product categories provided

KP281

Easily affordable entry level product, which is also the maximum selling product.

KP281 is the most popular product among the entry level customers.

This product is easily afforded by both Male and Female customers.

Average distance covered in this model is around 70 to 90 miles.

Product is used 3 to 4 times a week.

Most of the customer who have purchased the product have rated Average shape as the fitness rating.

Younger to Elder beginner level customers prefer this product.

Single female & Partnered male customers bought this product more than single male customers.

Income range between 39K to 53K have preferred this product.

KP481

This is an Intermediate level Product.

KP481 is the second most popular product among the customers.

Fitness Level of this product users varies from Bad to Average Shape depending on their usage.

Customers Prefer this product mostly to cover more miles than fitness.

Average distance covered in this product is from 70 to 130 miles per week.

More Female customers prefer this product than males.

Probability of Female customer buying KP481 is significantly higher than male.

KP481 product is specifically recommended for Female customers who are intermediate user.

Three different age groups prefer this product - Teen, Adult and middle aged.

Average Income of the customer who buys KP481 is 49K.

Average Usage of this product is 3 days per week.

More Partnered customers prefer this product.

There are slightly more male buyers of the KP481.

The distance travelled on the KP481 treadmill is roughly between 75 - 100 Miles. It is also the 2nd most distance travelled model.

The buyers of KP481 in Single & Partnered, Male & Female are same.

The age range of KP481 treadmill customers is roughly between 24-34 years.

KP781

Due to the High Price & being the advanced type, customer prefers less of this product.

Customers use this product mainly to cover more distance.

Customers who use this product have rated excelled shape as fitness rating.

Customer walk/run average 120 to 200 or more miles per week on his product.

Customers use 4 to 5 times a week at least.

Female Customers who are running average 180 miles (extensive exercise) , are using product KP781, which is higher than Male average using same product.

Probability of Male customer buying Product KP781(31.73%) is way more than female(9.21%).

Probability of a single person buying KP781 is higher than Married customers. So , KP781 is also recommended for people who are single and exercises more.

Middle aged to higher age customers tend to use this model to cover more distance.

Average Income of KP781 buyers are over 75K per annum

Partnered Female bought KP781 treadmill compared to Partnered Male.

Customers who have more experience with previous aerofit products tend to buy this product

This product is preferred by the customer where the correlation between Education and Income is High.

Recommendations

• Position KP281 and KP481 treadmills as cost-effective options, targeting customers with annual incomes in the $39,000 - $53,000 range.

• Market the KP781 treadmill as a high-end product with advanced features, catering to professionals and fitness enthusiasts.

• Enhance the marketing strategy for KP781 by collaborating with renowned athletes like Neeraj Chopra, leveraging their achievements for better outreach and brand association.

• Implement targeted marketing campaigns on occasions such as Women's Day and Mother's Day to encourage more women to adopt an active lifestyle, highlighting the benefits of using the company's treadmills.

• Conduct market research to expand the customer base beyond the age of 50. Offer basic treadmill models (KP281/KP481) as suitable options for beginners in this age group.

• Encourage existing customers to upgrade their treadmills to higher-end models as their usage and fitness levels increase over time, leading to increased revenue for the business.