Q1. Business Case: Aerofit - Descriptive Statistics & Probability

Mindset:

Evaluation will be kept lenient, so make sure you attempt this case study. It is understandable that you might struggle with getting started on this. Just brainstorm, discuss with peers, or get help from TAs. There is no right or wrong answer. We have to get used to dealing with uncertainty in business. This is exactly the skill we want to develop. 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.

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. Dataset

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:

Dataset link: Aerofit_treadmill.csv

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

The KP281 is an entry-level treadmill that sells for $1,500. The KP481 is for mid-level runners that sell for$1,750. The KP781 treadmill is having advanced features that sell for $2,500. What good looks like?

Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset Detect Outliers (using boxplot, “describe” method by checking the difference between mean and median)

Check if features like marital status, age have any effect on the product purchased (using countplot, histplots, boxplots etc) Representing the marginal probability like - what percent of customers have purchased KP281, KP481, or KP781 in a table (can use pandas.crosstab here)

Check correlation among different factors using heat maps or pair plots. With all the above steps you can answer questions like: What is the probability of a male customer buying a KP781 treadmill? Customer Profiling - Categorization of users. Probability- marginal, conditional probability.

Some recommendations and actionable insights, based on the inferences. Later on, we will see more ways to do “customer segmentation”, but this case study in itself is relevant in some real-world scenarios.

Evaluation Criteria

Defining Problem Statement and Analysing basic metrics (10 Points)

1. Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), statistical summary

Non-Graphical Analysis: Value counts and unique attributes (10 Points)

Visual Analysis - Univariate & Bivariate (30 Points)

1. For continuous variable(s): Distplot, countplot, histogram for univariate analysis (10 Points)
2. For categorical variable(s): Boxplot (10 Points)
3. For correlation: Heatmaps, Pairplots(10 Points)

Missing Value & Outlier Detection (10 Points)

Business Insights based on Non-Graphical and Visual Analysis (10 Points) Comments on the range of attributes Comments on the distribution of the variables and relationship between them Comments for each univariate and bivariate plot

Recommendations (10 Points) - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand

Objective:

To perform basic Exploratory Data Analysis using descriptive Statistics and Probabilities on Aerofit data, derive basic insights from it

1. Importing libraries

In [1]:
```
import os
import pandas as pd
import numpy as np
from tqdm import tqdm
import plotly.express as px
import plotly.graph_objects as go
#from plotly.offline import init_notebook_mode, iplot
#init_notebook_mode(connected=True)

import warnings
warnings.filterwarnings("ignore")
```

2. Data Overview

In [2]:
```
read_path = 'aerofit_treadmill.csv'
data = pd.read_csv(read_path)
data.shape
```

Out[2]:
```
(180, 9)
```

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Product        180 non-null    object
 1   Age            180 non-null    int64 
 2   Gender         180 non-null    object
 3   Education      180 non-null    int64 
 4   MaritalStatus  180 non-null    object
 5   Usage          180 non-null    int64 
 6   Fitness        180 non-null    int64 
 7   Income         180 non-null    int64 
 8   Miles          180 non-null    int64 
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

Product, Gender and Marital Status are object(string)

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

There are no null values for any features hence no actions required.

In [4]:
```
data.head()
```

Out[4]:

	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

In [5]:
```
data.describe()
```

Out[5]:

	Age	Education	Usage	Fitness	Income	Miles
count	180.000000	180.000000	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	3.455556	3.311111	53719.577778	103.194444
std	6.943498	1.617055	1.084797	0.958869	16506.684226	51.863605
min	18.000000	12.000000	2.000000	1.000000	29562.000000	21.000000
25%	24.000000	14.000000	3.000000	3.000000	44058.750000	66.000000
50%	26.000000	16.000000	3.000000	3.000000	50596.500000	94.000000
75%	33.000000	16.000000	4.000000	4.000000	58668.000000	114.750000
max	50.000000	21.000000	7.000000	5.000000	104581.000000	360.000000

Descriptive Analysis Total count of all columns is 180

Age: Mean age of the customer is 28.7 years, half of the customer's age is less than or equal to 26.

Education: Mean Education is 15.57 years with maximum as 21 and minimum as 12.

Usage: Mean Usage per week is 3.4, with maximum as 7 and minimum as 2.

Fitness: Average rating is 3.3 on a scale of 1 to 5.

Miles: Average number of miles the customer walks is 103 with maximum distance travelled by most people is 114.75 and minimum is 21.

Income (in $): Most customer earns around 50.5K annually, with maximum of 104.5K and minimum almost 30K

In [6]:
```
!pip install skimpy -q
```

In [7]:
```
from skimpy import skim
skim(data)
```

skimpy summary

Data Summary

dataframe	Values
Number of rows	180
Number of columns	9

Data Types

Column Type	Count
int32	6
string	3

number

column_name	NA	NA %	mean	sd	p0	p25
Age	0	0	29	6.9	18	
Education	0	0	16	1.6	12	
Usage	0	0	3.5	1.1	2	
Fitness	0	0	3.3	0.96	1	
Income	0	0	54000	17000	30000	4400
Miles	0	0	100	52	21	

string

column_name	NA	NA %	words per row
Product	0	0	
Gender	0	0	
MaritalStatus	0	0	

End

Mean Age of the given customer dataset is 28.78

Minimum Age of the customer starts from 18 and maximum age is 50

25% of the customers age is 24

75% of the customer age is 33

Maximum Education qualification is 21, with most frequent education as 16

Average usage per week for a customer is 3 days

Average Fitness rating is 3 with most common fitness rating is 4

Average Income of the purchased customer is around 54K per year

Highest salary recorded for the customer is around 104K per year

Maximum distance covered by the customer in treadmill is 360 miles

Most of the customers cover a distance of 114 miles with an average of 103 miles

Around 25% of the customer cover an average of 66 miles

2.1 Data description

Product: tells us which of the three product categories it belongs to: KP281, KP481, KP781

Age: age of the customer

Gender: gender of the customer

Education: education duration of the customer in years

MaritalStatus: status of the customer whether he/she is married or single

Usage: average treadmill usage

Fitness: fitness level of the customer

Income: annual income of the customer(in dollars)

Miles: miles tracked by the customer

In [8]:
```
for col in data.columns.tolist():
    print("Feature: ", col)
    if data[col].nunique()>10:
        print(data[col].describe())
    else:
        print(data[col].value_counts(normalize=True))
    print("**"*30)
```

```
Feature:  Product
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: Product, dtype: float64
************************************************************
Feature:  Age
count    180.000000
mean      28.788889
std        6.943498
min       18.000000
25%       24.000000
50%       26.000000
75%       33.000000
max       50.000000
Name: Age, dtype: float64
************************************************************
Feature:  Gender
Male      0.577778
Female    0.422222
Name: Gender, dtype: float64
************************************************************
Feature:  Education
16    0.472222
14    0.305556
18    0.127778
15    0.027778
13    0.027778
12    0.016667
21    0.016667
20    0.005556
Name: Education, dtype: float64
************************************************************
Feature:  MaritalStatus
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
************************************************************
Feature:  Usage
3    0.383333
4    0.288889
2    0.183333
5    0.094444
6    0.038889
7    0.011111
Name: Usage, dtype: float64
************************************************************
Feature:  Fitness
3    0.538889
5    0.172222
2    0.144444
4    0.133333
1    0.011111
Name: Fitness, dtype: float64
************************************************************
Feature:  Income
count      180.000000
mean     53719.577778
std      16506.684226
min      29562.000000
```

```
25%         44058.750000
50%         50596.500000
75%         58668.000000
max        104581.000000
Name: Income, dtype: float64
************************************************************
Feature:  Miles
count     180.000000
mean      103.194444
std        51.863605
min        21.000000
25%        66.000000
50%        94.000000
75%       114.750000
max       360.000000
Name: Miles, dtype: float64
************************************************************
```

In [9]:

```
df1 = data[['Product', 'Gender', 'MaritalStatus']].melt()
df1.groupby(['variable', 'value'])[['value']].count() / len(data)
```

Out[9]:

variable	value	value
Gender	**Female**	0.422222
	Male	0.577778
MaritalStatus	**Partnered**	0.594444
	Single	0.405556
Product	**KP281**	0.444444
	KP481	0.333333
	KP781	0.222222

44.44% of customers bought KP281 product type

33.33% of customers bought KP481 product type

22.22% of customers bought KP781 product type

57.78% of customers are Male and 42.22% customers are Female

59.44% of customers are Married/Partnered

40.56% of customers are Single

Around 39% of customers use 3 days per week

Less than 2% of customers use 7 days per week

More than 53% of customers have rated themselves as average in fitness (rated 3)

14% of customers have rated their fitness less than average

Over 17% of customers have peak fitness ratings

2.2 Product Portfolio:

The KP281 is an entry-level treadmill that sells for $1,500.

The KP481 is for mid-level runners that sell for $1,750.

The KP781 treadmill is having advanced features that sell for $2,500.

Revenues from each of the products

In [10]:
```
product_data  = data['Product'].value_counts().to_frame().reset_index().rename(c
product_data
```

Out[10]:

	Product	Products sold
0	KP281	80
1	KP481	60
2	KP781	40

In [11]:
```
product_data['Revenue'] = pd.Series( np.array( [80*1500, 60*1750, 40*2500] ) )
product_data
```

Out[11]:

	Product	Products sold	Revenue
0	KP281	80	120000
1	KP481	60	105000
2	KP781	40	100000

3. Univariate Analysis

In [12]:
```
data.columns.tolist()
```

Out[12]:
```
['Product',
 'Age',
 'Gender',
 'Education',
 'MaritalStatus',
 'Usage',
 'Fitness',
 'Income',
 'Miles']
```

In [13]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
```

```python
import seaborn as sns

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

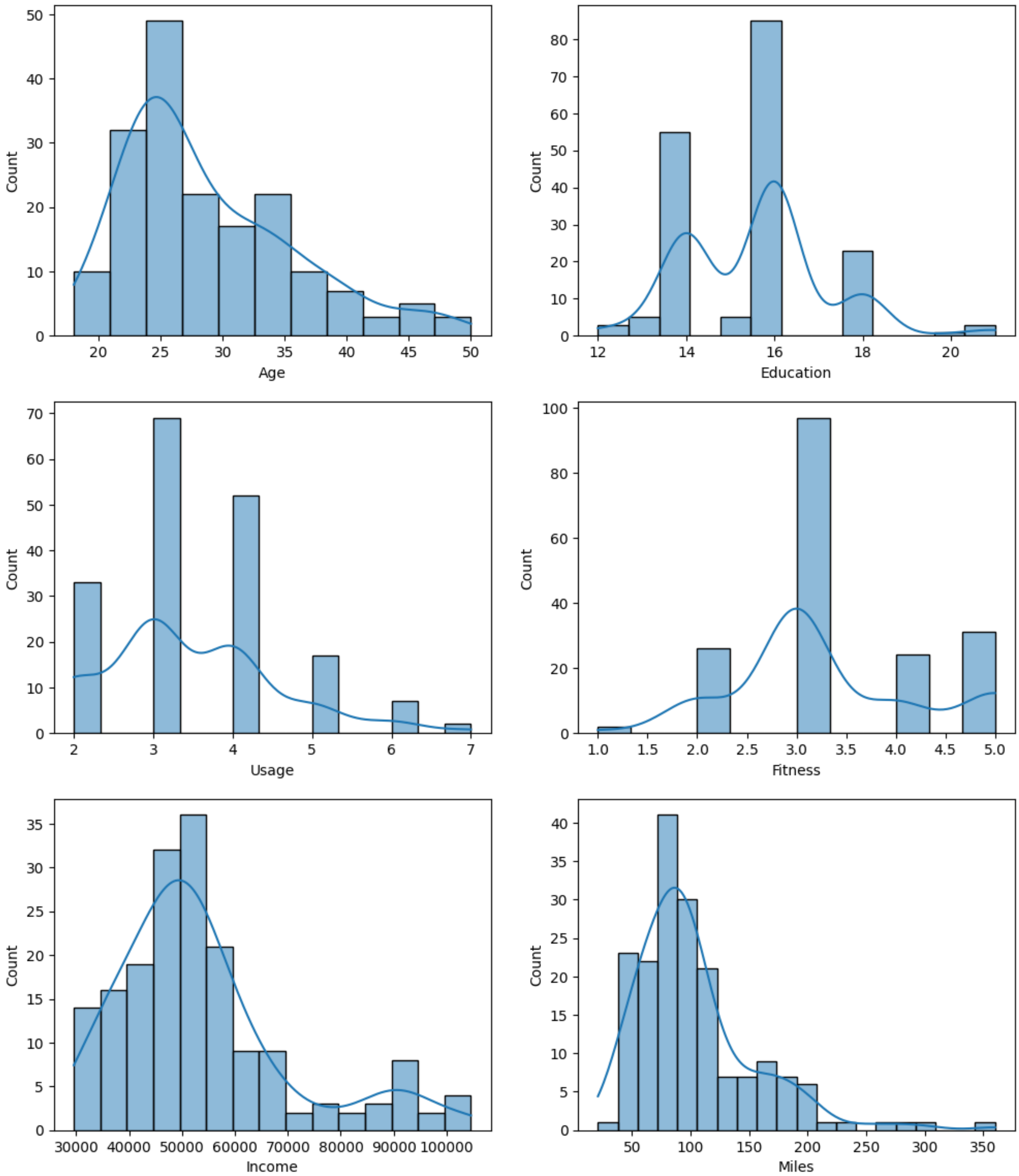
Out[13]:

	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 × 9 columns

In [14]:

```python
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))
fig.subplots_adjust(top=1.2)

sns.histplot(data=df, x="Age", kde=True, ax=axis[0,0])
sns.histplot(data=df, x="Education", kde=True, ax=axis[0,1])
sns.histplot(data=df, x="Usage", kde=True, ax=axis[1,0])
sns.histplot(data=df, x="Fitness", kde=True, ax=axis[1,1])
sns.histplot(data=df, x="Income", kde=True, ax=axis[2,0])
sns.histplot(data=df, x="Miles", kde=True, ax=axis[2,1])
plt.show()
```



4. Fixing data

In [15]:
```
#
data[data['Miles']>225]
```

Out[15]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
155	KP781	25	Male	18	Partnered	6	5	75946	240
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
170	KP781	31	Male	16	Partnered	6	5	89641	260
173	KP781	35	Male	16	Partnered	4	5	92131	360

In [16]:
```
df1 = df[['Product', 'Gender', 'MaritalStatus']].melt()
df1.groupby(['variable', 'value'])[['value']].count() / len(df)
```

Out[16]:

variable	value	value
Gender	**Female**	0.422222
	Male	0.577778
MaritalStatus	**Partnered**	0.594444
	Single	0.405556
Product	**KP281**	0.444444
	KP481	0.333333
	KP781	0.222222

Obervations

Product 44.44% of the customers have purchased KP2821 product. 33.33% of the customers have purchased KP481 product. 22.22% of the customers have purchased KP781 product.

Gender 57.78% of the customers are Male.

MaritalStatus 59.44% of the customers are Partnered.

In [17]:
```
# difference in Age and Edu-years to determine any invalid entry
df['Age-Edu-Diff']=df['Age'] - df['Education']
#print(df['Age-Edu-Diff'].value_counts())
df['Age-Edu-Diff'].min()
```

Out[17]: 4

The Minimum age-Education difference is 4 years which satisfies the normal scenarios and hence no invalid data

In [18]:
```
# Miles per usage calculation to check impractical/over-ambitious targets
df['MilesPerUsage']=df['Miles']/df['Usage']
#print(df['Age-Edu-Diff'].value_counts())
df['MilesPerUsage'].max()
# As any value above 60 miles/usage would be impractical based on standard data
df[df['MilesPerUsage']>=50]
```

Out[18]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
16	KP281	23	Female	14	Single	2	3	34110	103
24	KP281	24	Male	14	Single	2	3	45480	113
142	KP781	22	Male	18	Single	4	5	48556	200
171	KP781	33	Female	18	Partnered	4	5	95866	200
173	KP781	35	Male	16	Partnered	4	5	92131	360

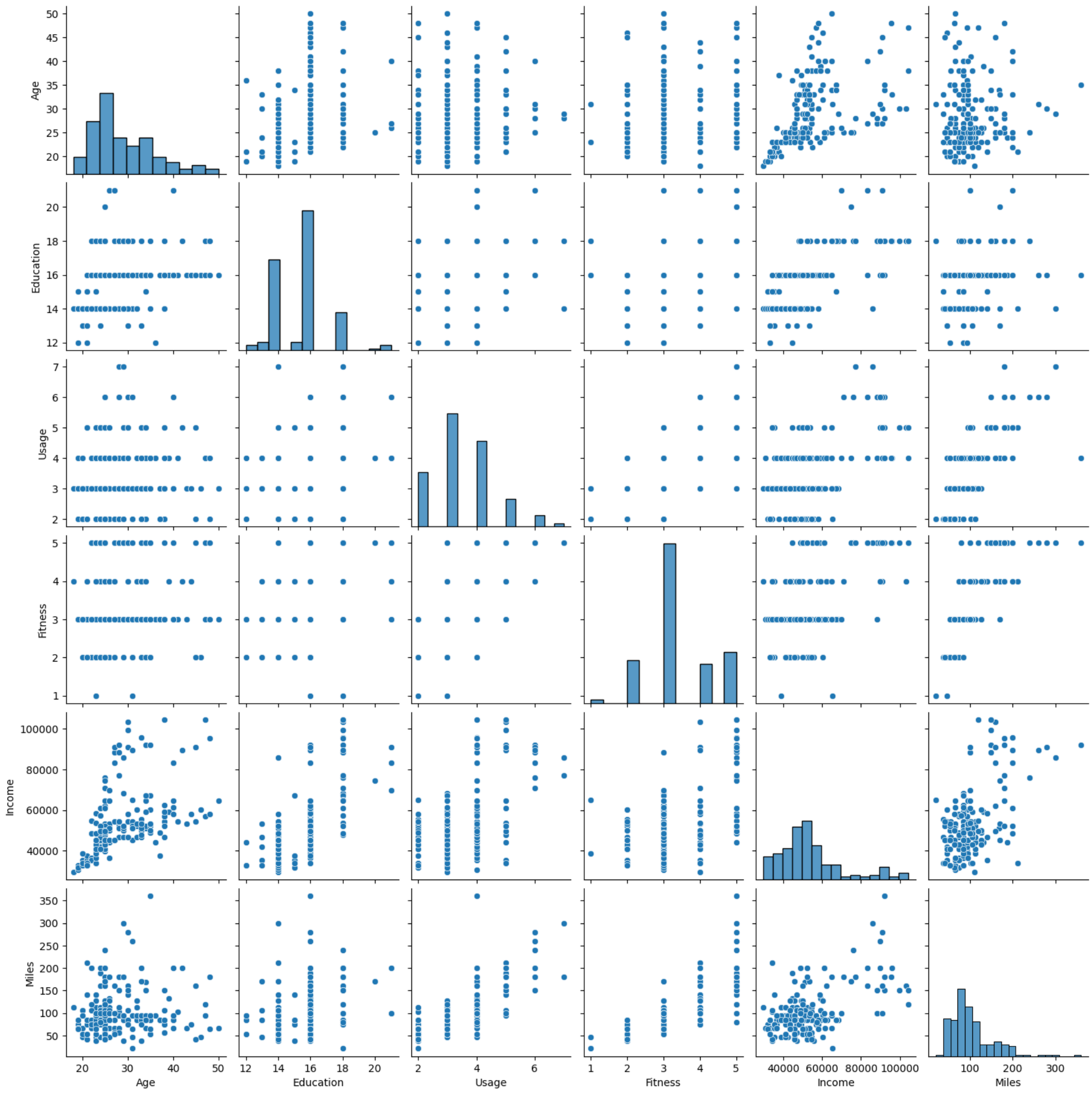
We can try to verify the data above and consult the customers to set practical targets

5. Bivariate Analysis

In [19]:

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

plt.figure(figsize=(20,10))
sns.pairplot(data)
plt.show()
```

```
<Figure size 2000x1000 with 0 Axes>
```

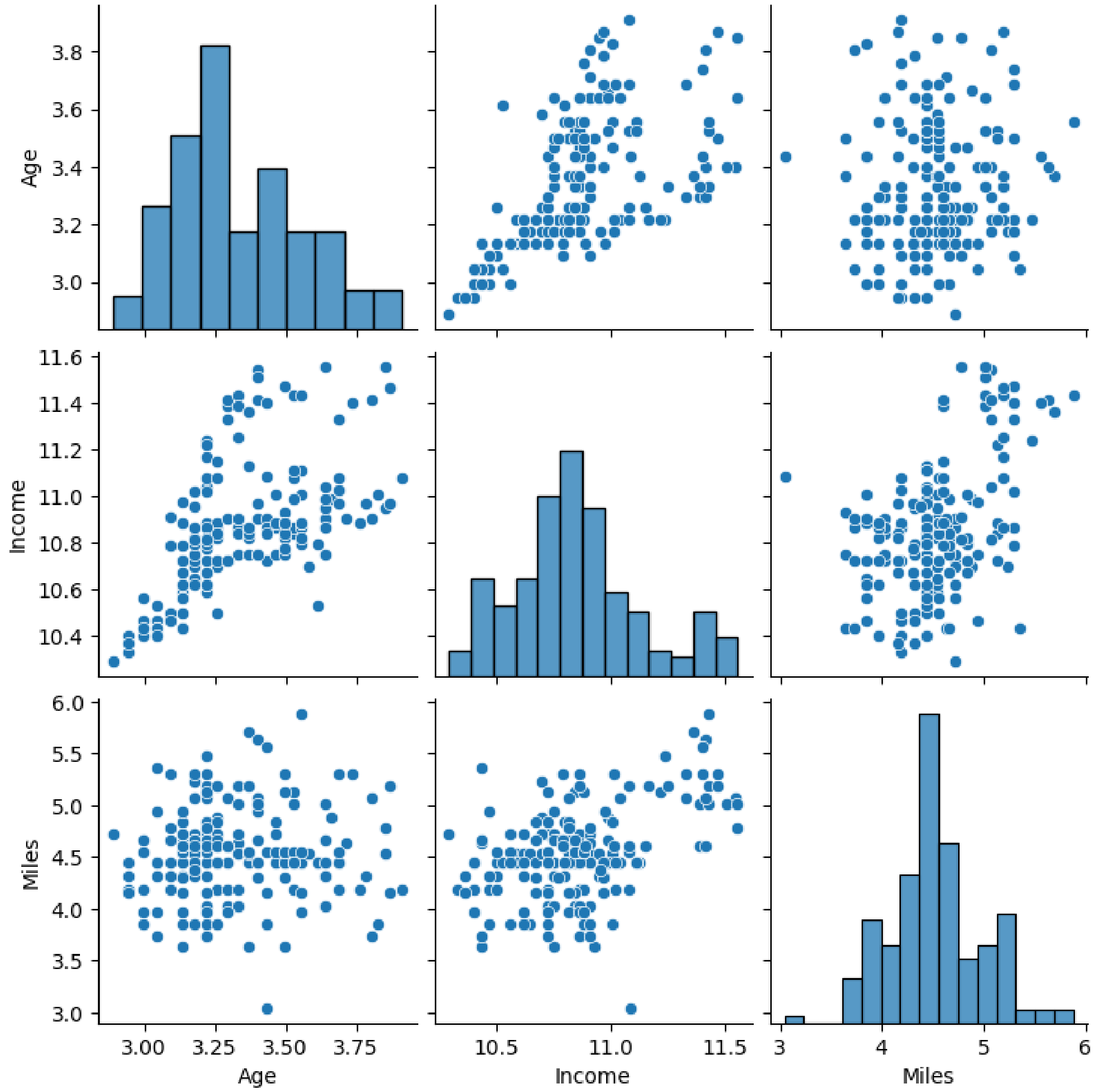


Observations: Age, Income and Miles show Heteroscedasticity.

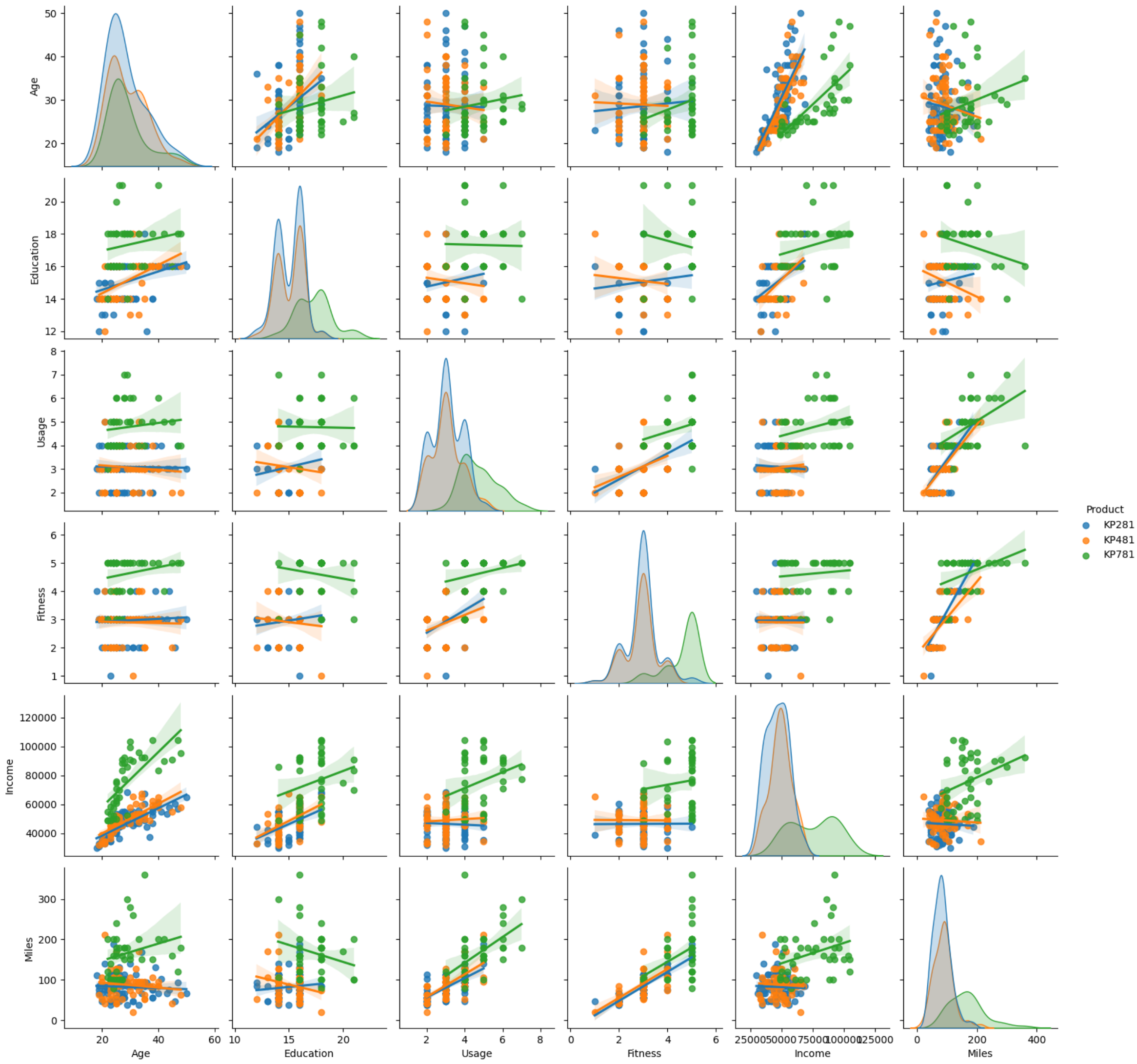
In [20]:
```
transformed_data = pd.DataFrame()

transformed_data['Age'] = np.log(data['Age'])
transformed_data['Income'] = np.log(data['Income'])
transformed_data['Miles'] = np.log(data['Miles'])
```

In [21]:
```
plt.figure(figsize=(20,10))
sns.pairplot(transformed_data)
plt.show()
```

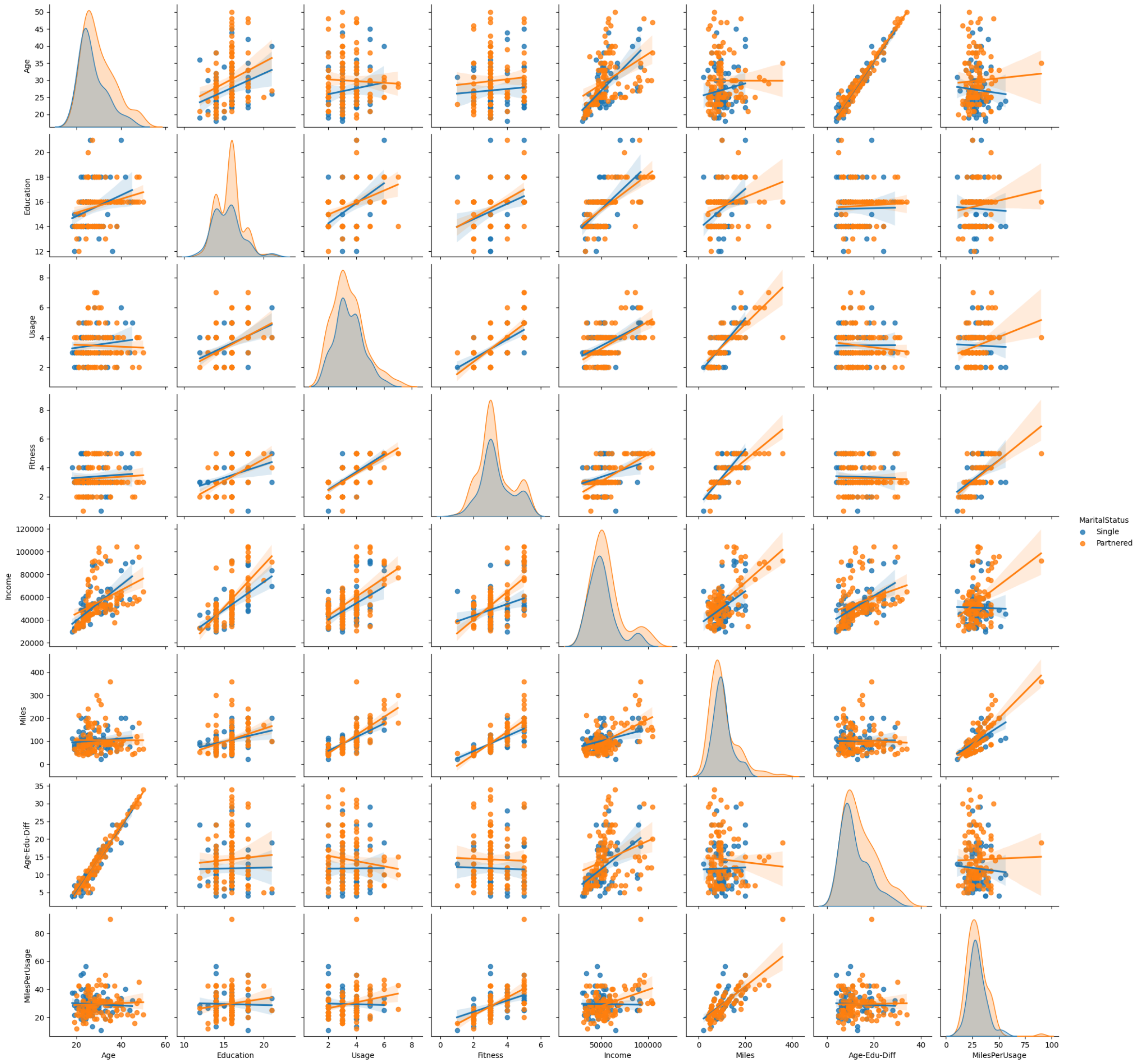
```
<Figure size 2000x1000 with 0 Axes>
```



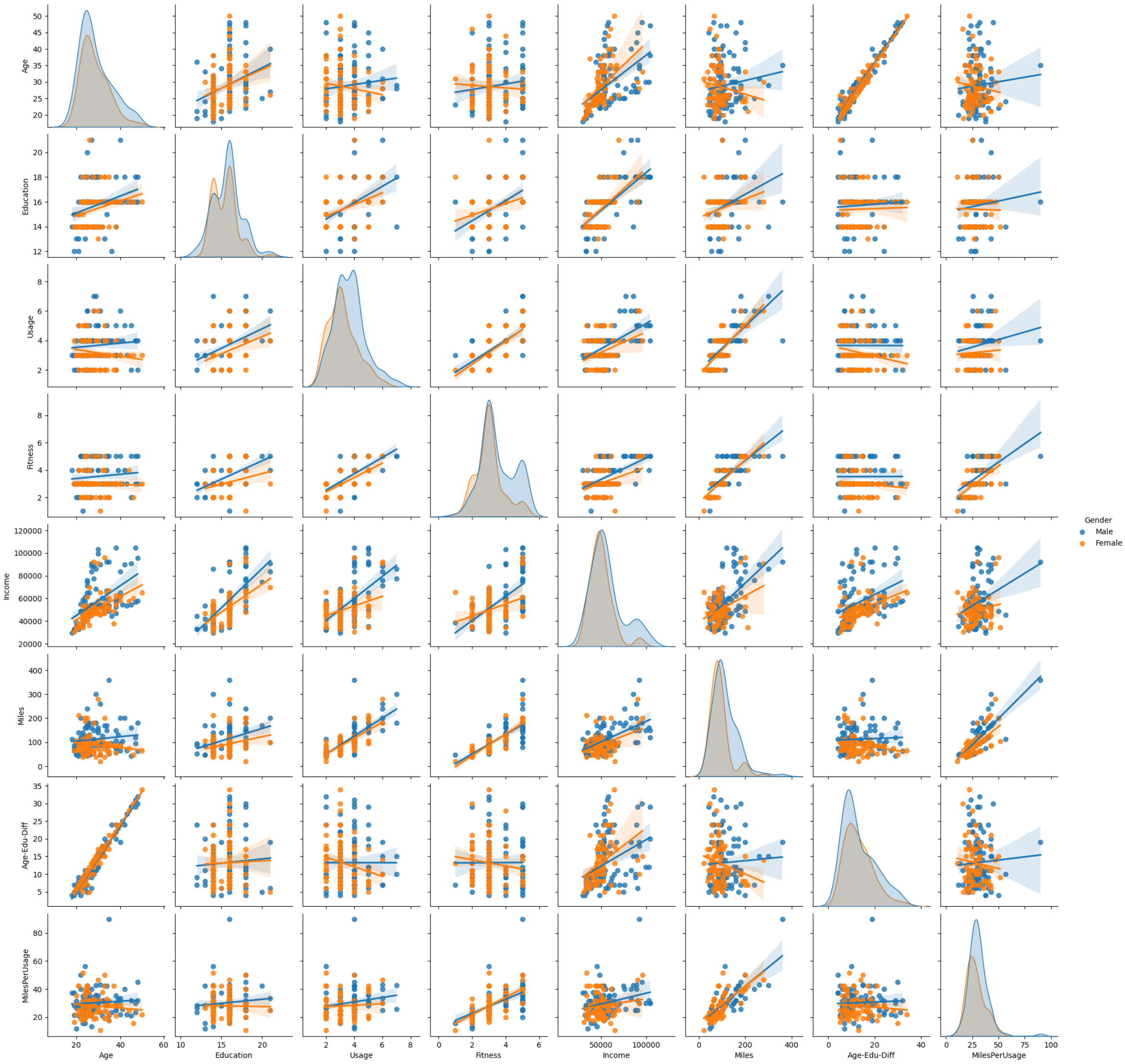
In [22]:
```
# Product Analysis - Pair Plot
sns.pairplot(data,hue='Product',kind='reg')
plt.show()
```



In [23]:
```
# Marital Status - pair plot
sns.pairplot(df,hue='MaritalStatus',kind='reg')
plt.show()
```

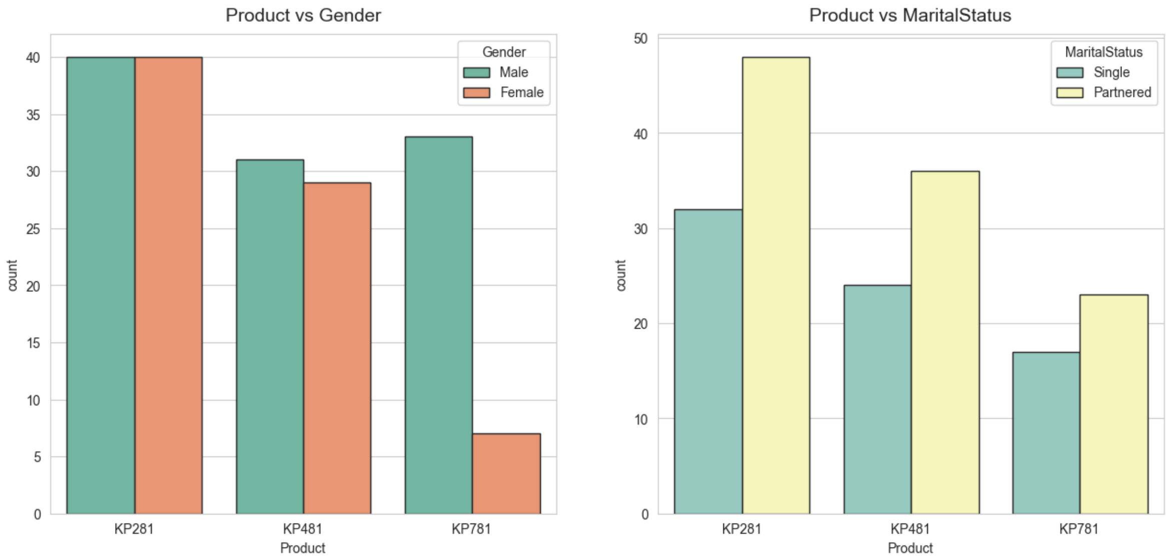


In [24]:
```python
# Gender Analysis - Pair Plot
sns.pairplot(df,hue='Gender',kind='reg')
plt.show()
```

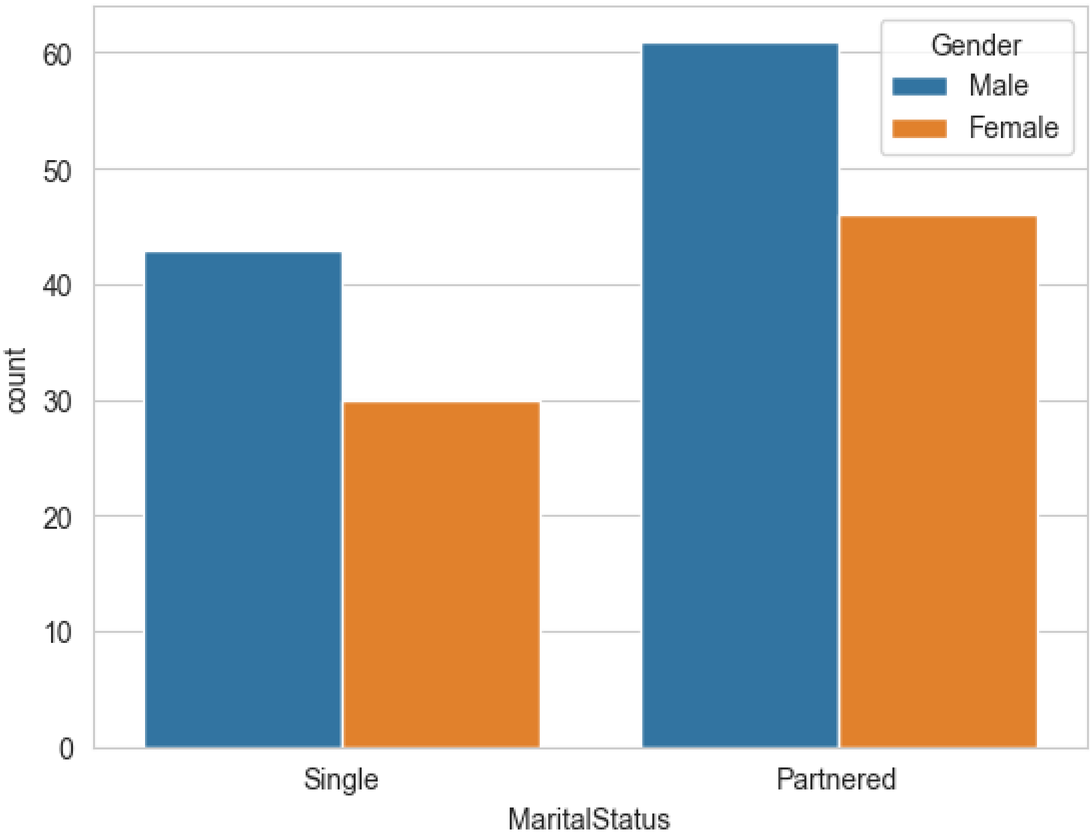


Checking if features - **Gender** or **MaritalStatus** have any effect on the product purchased.

In [25]:
```
sns.set_style(style='whitegrid')
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15, 6.5))
sns.countplot(data=df, x='Product', hue='Gender', edgecolor="0.15", palette='Set
sns.countplot(data=df, x='Product', hue='MaritalStatus', edgecolor="0.15", palet
axs[0].set_title("Product vs Gender", pad=10, fontsize=14)
axs[1].set_title("Product vs MaritalStatus", pad=10, fontsize=14)
plt.show()
```



In [26]:
```
# Count among Gender and their Marital Status
sns.countplot(data=data,x='MaritalStatus',hue='Gender')
plt.show()
```

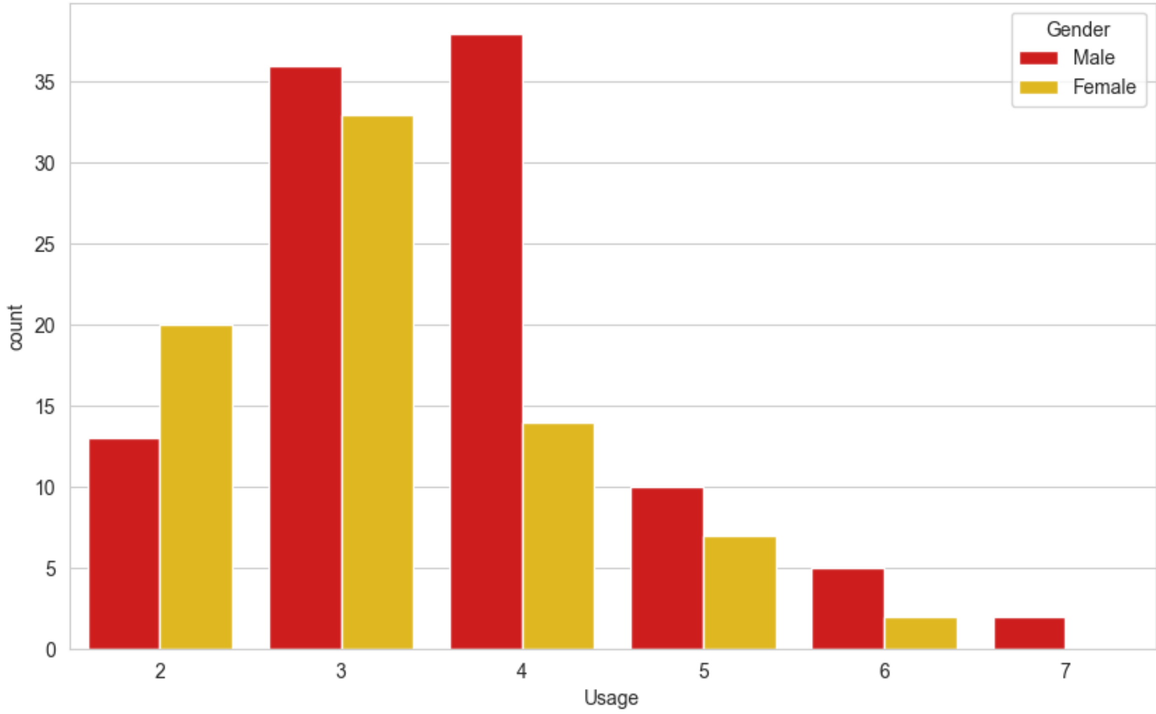


Partnered customers are the most buyers of aerofit product

Out of both Single and Partnered customers, Male customers are significantly high

Female customers are considerably low compared to Male customers

In [27]:
```
# Purchased product usage among Gender
plt.figure(figsize=(10,6))
sns.countplot(data=data,x='Usage',hue='Gender',palette='hot')
plt.show()
```



Among Male and Female genders, Male's usage is 3 and 4 days per week majorly

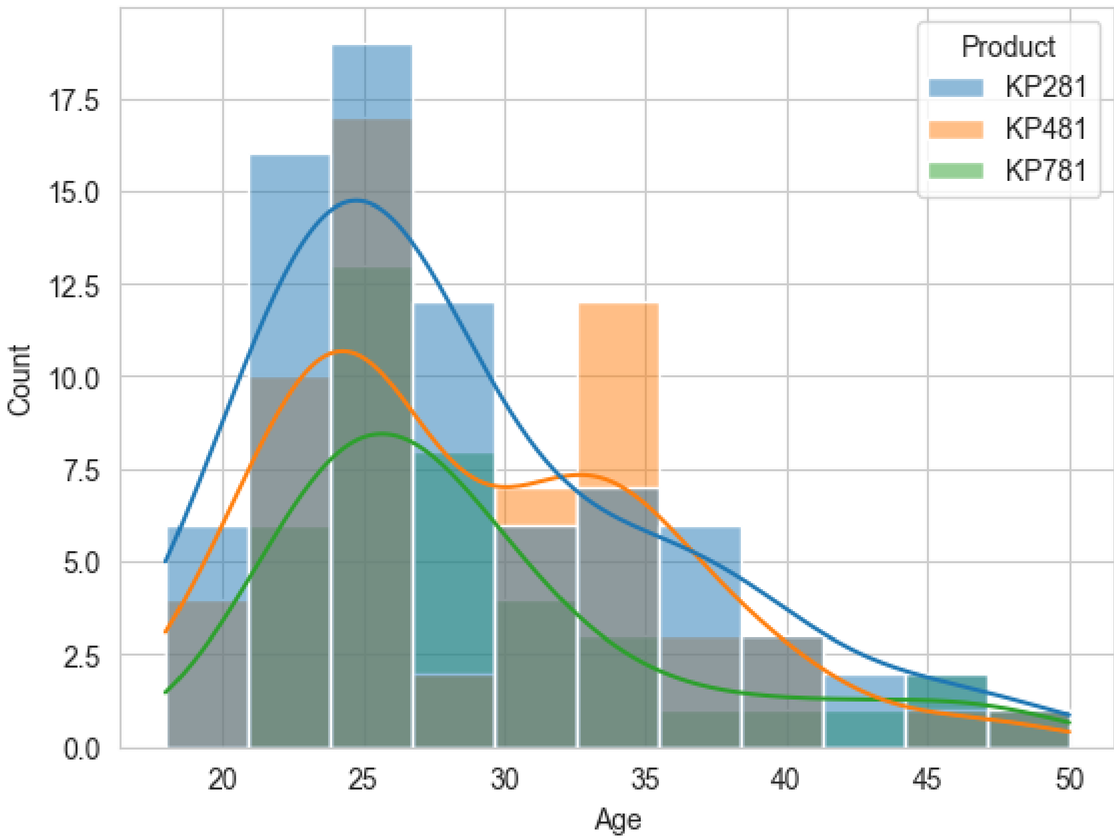
Female customers mostly use 3 days per week

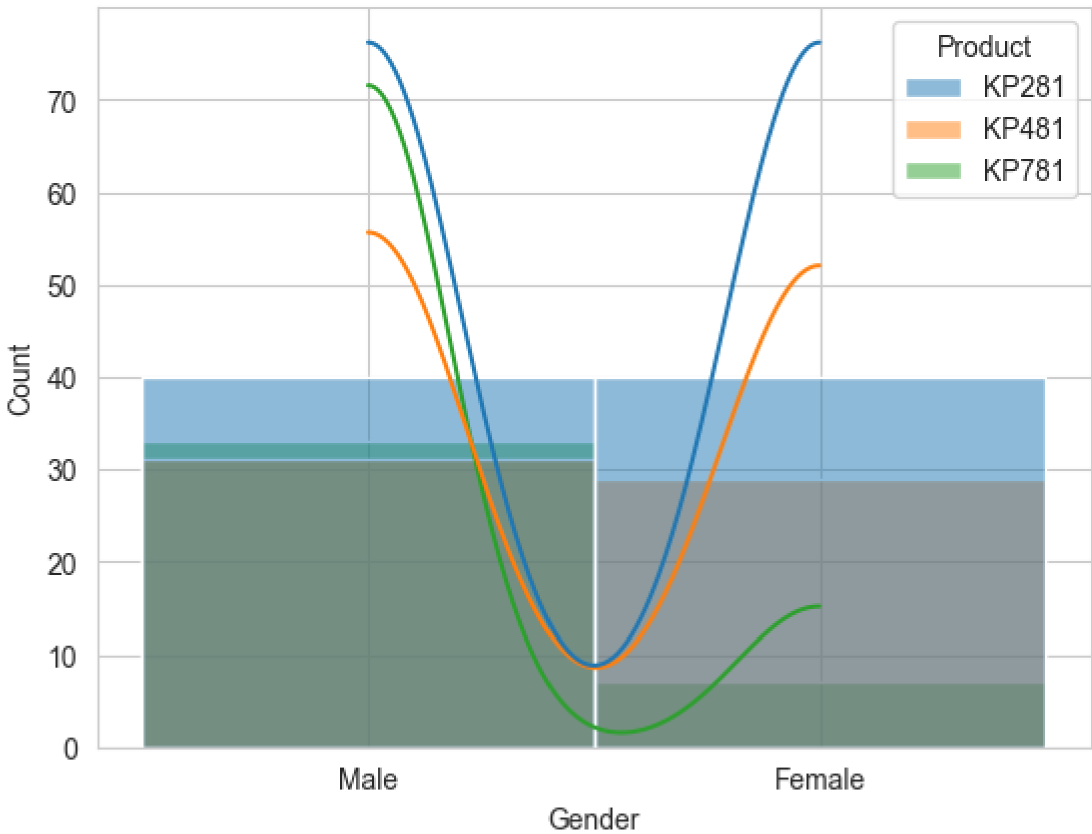
Only few Male customers use 7 days per week whereas female customer's maximum usage is only 6 days per week

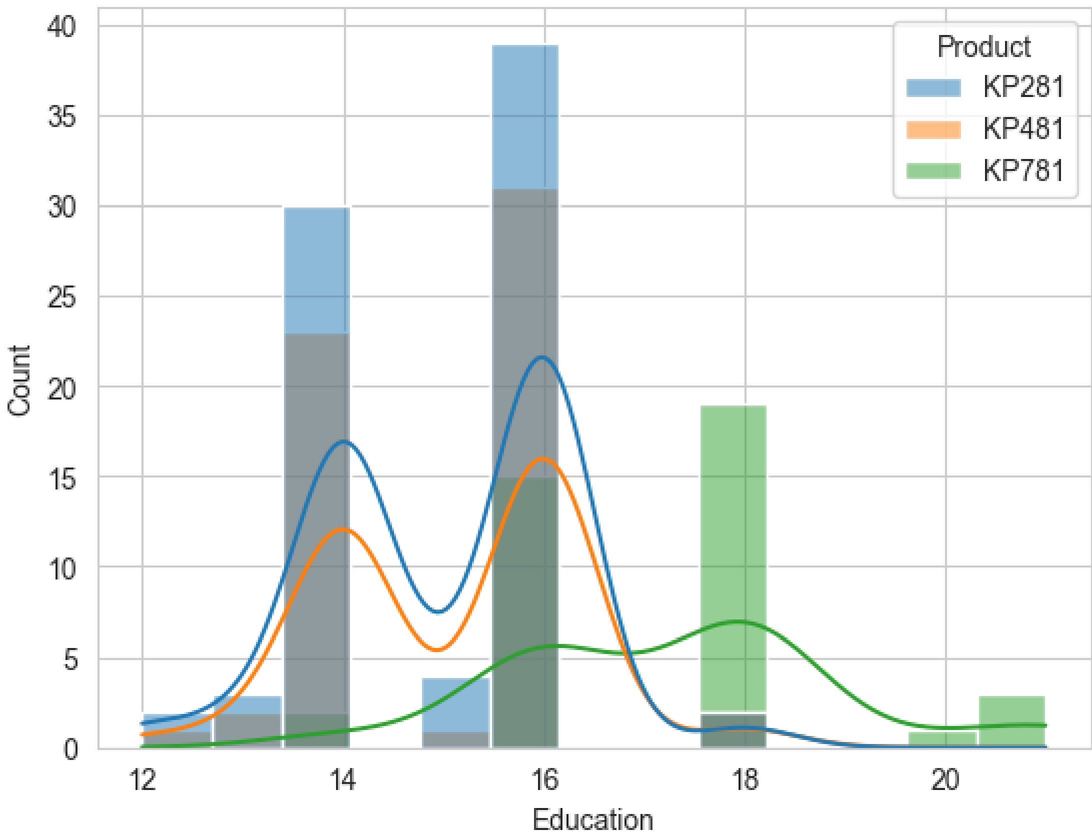
6. Product-based EDA

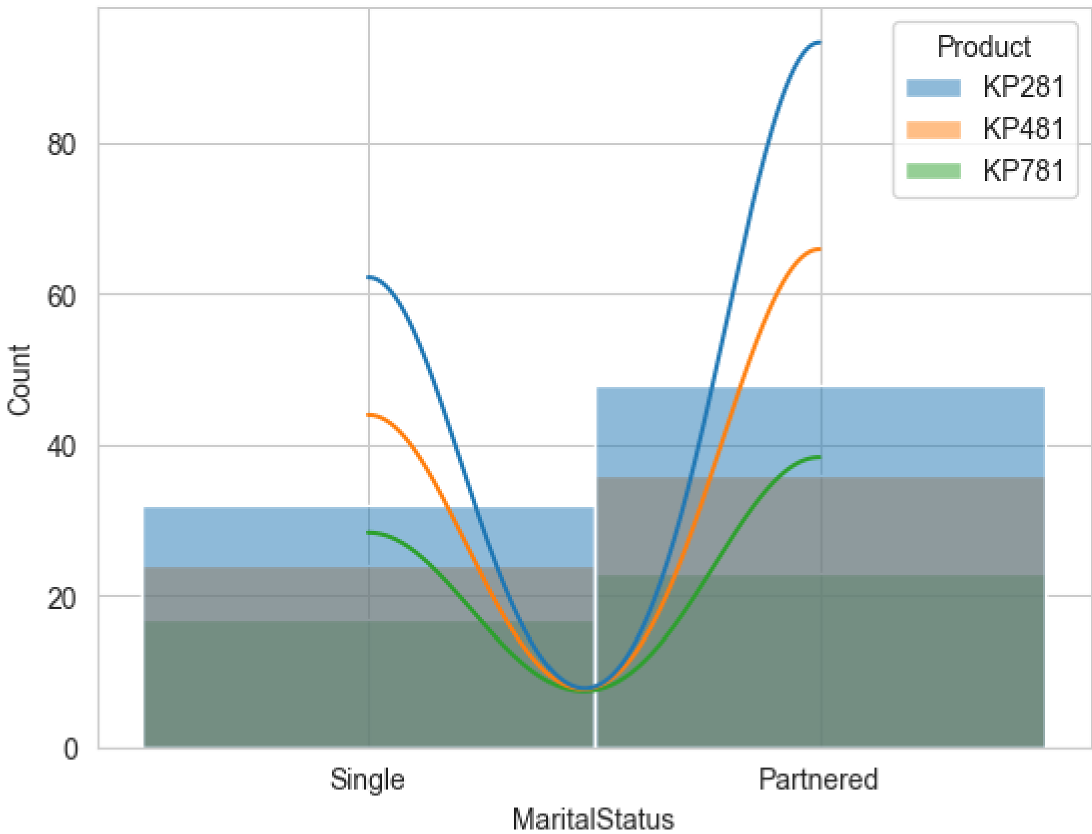
Obervations *Product vs Gender Equal number of males and females have purchased KP281 product and Almost same for the product KP481 Most of the Male customers have purchased the KP781 product. Product vs MaritalStatus Customer who is Partnered, is more likely to purchase the product.

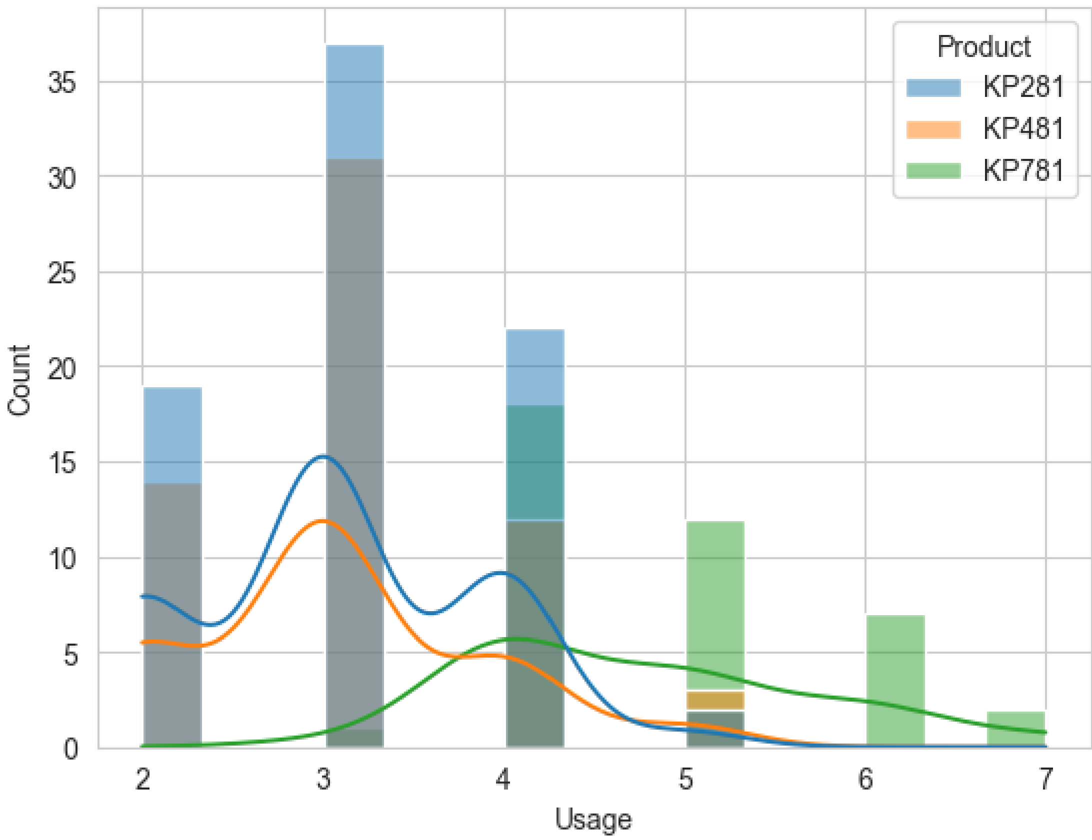
In [28]:
```
#fig, axis = plt.subplots(nrows=len(data.columns.tolist()), ncols=1, figsize=(12
#fig.subplots_adjust(top=1.2)
for col in data.columns.tolist()[1:]:
#     print(col)
    #fig = px.histogram(data, x=col, color="Product", barmode='group')
        sns.histplot(data=df, x=col, kde=True,hue="Product")
        #sns.set_title(f"Distribution of the feature: {col}", pad=10, fontsize=1
        #axs[1].set_title("Product vs MaritalStatus", pad=10, fontsize=14)
        plt.show()
```

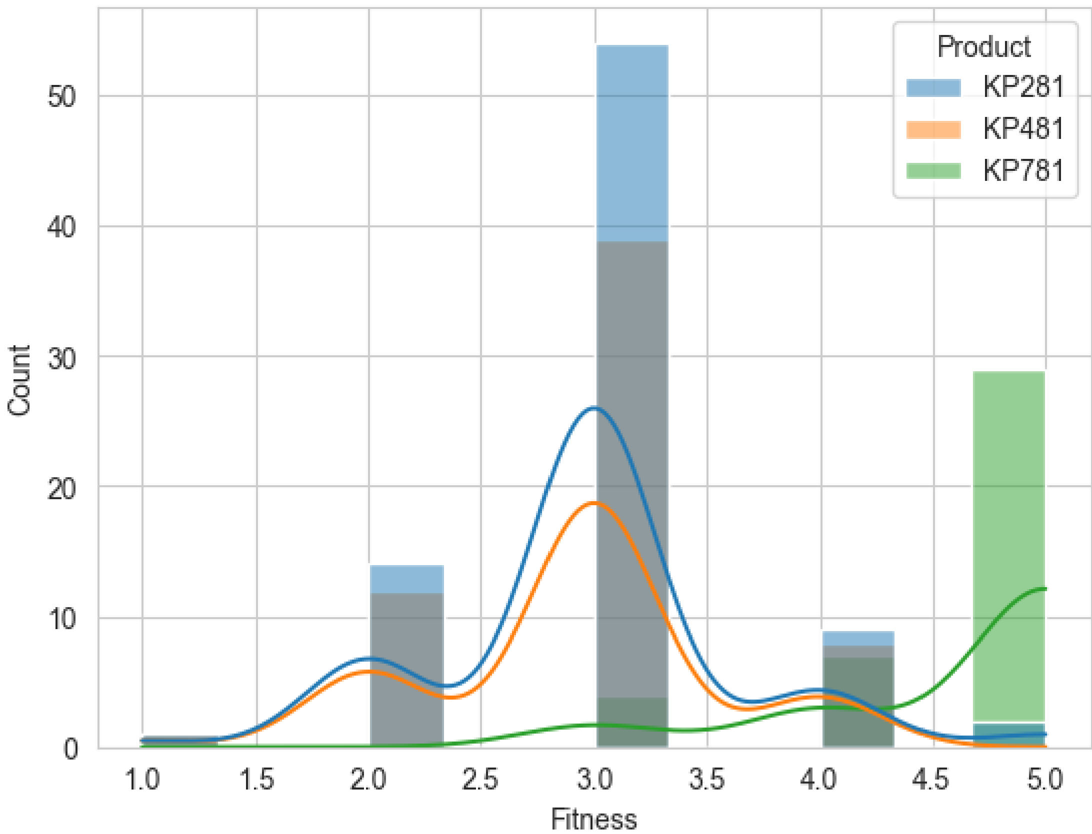

Product
KP281
KP481
KP781
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Count
20
25
30
35
40
45
50
Age

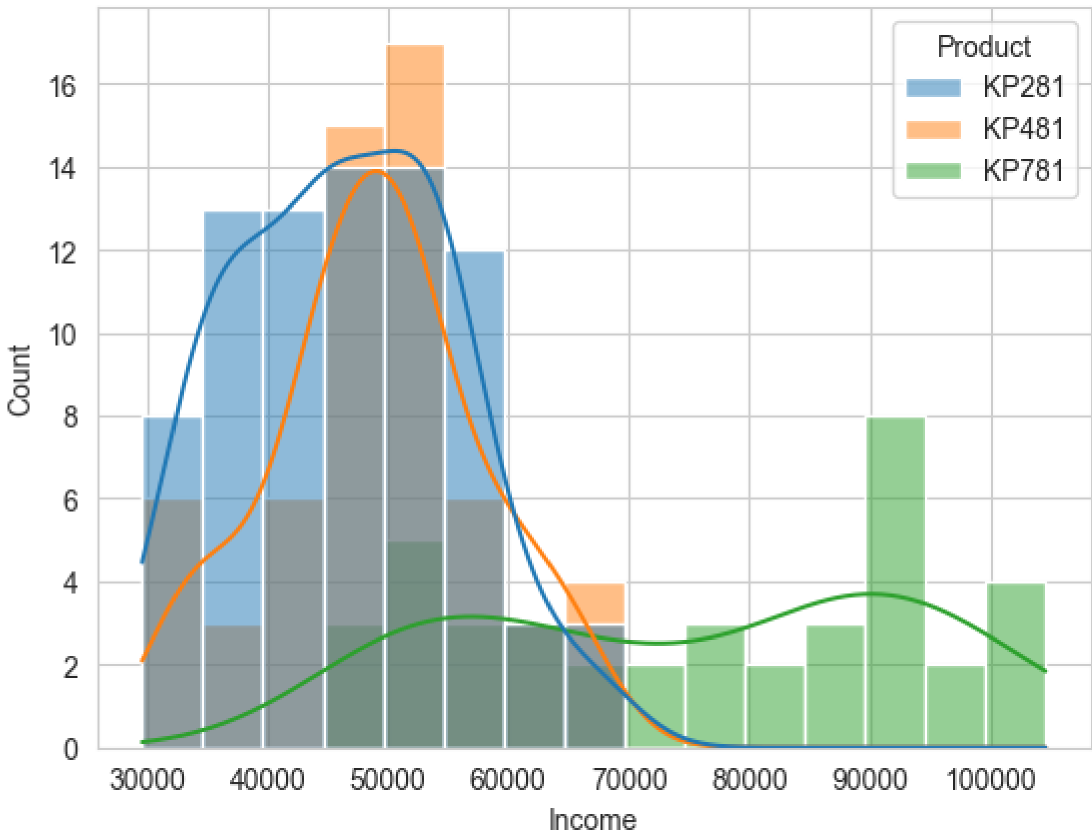

Product
KP281
KP481
KP781
70
60
50
40
30
20
10
0
Count
Male
Female
Gender

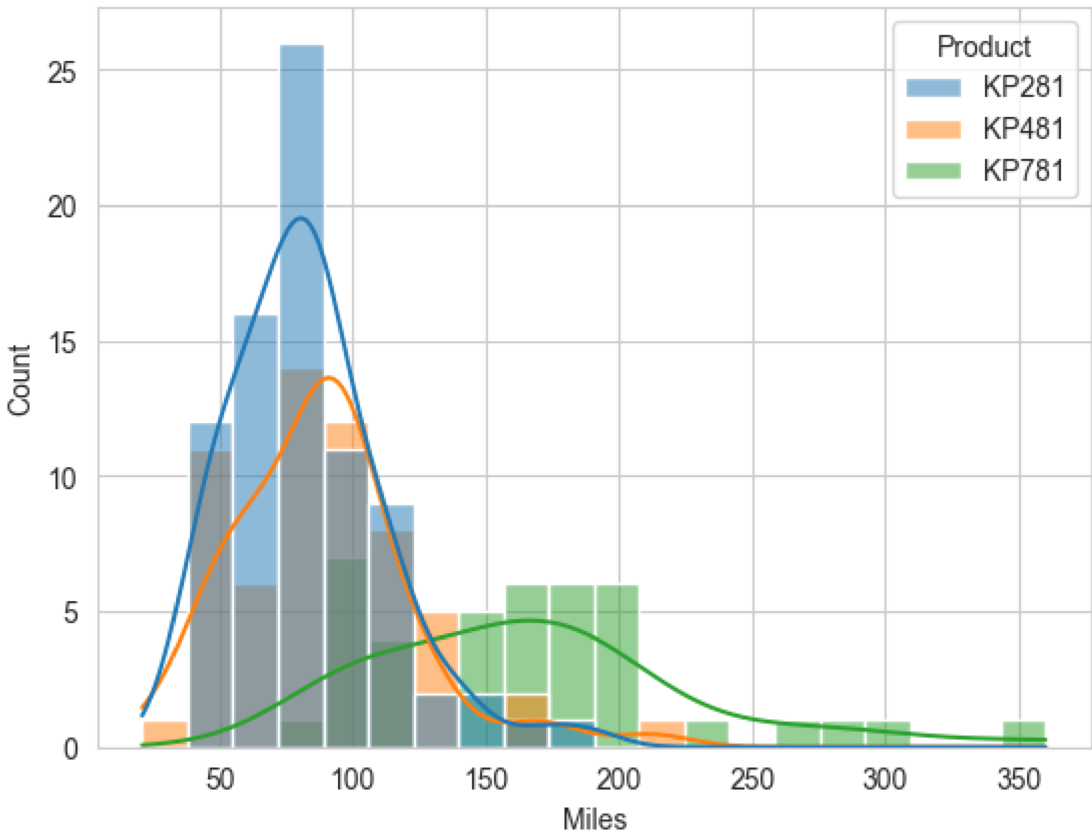

Product
KP281
KP481
KP781
Count
Education
0
5
10
15
20
25
30
35
40
12
14
16
18
20


Product
KP281
KP481
KP781
Count
MaritalStatus
0
20
40
60
80
Single
Partnered


Product
KP281
KP481
KP781
Count
Usage
0
5
10
15
20
25
30
35
2
3
4
5
6
7

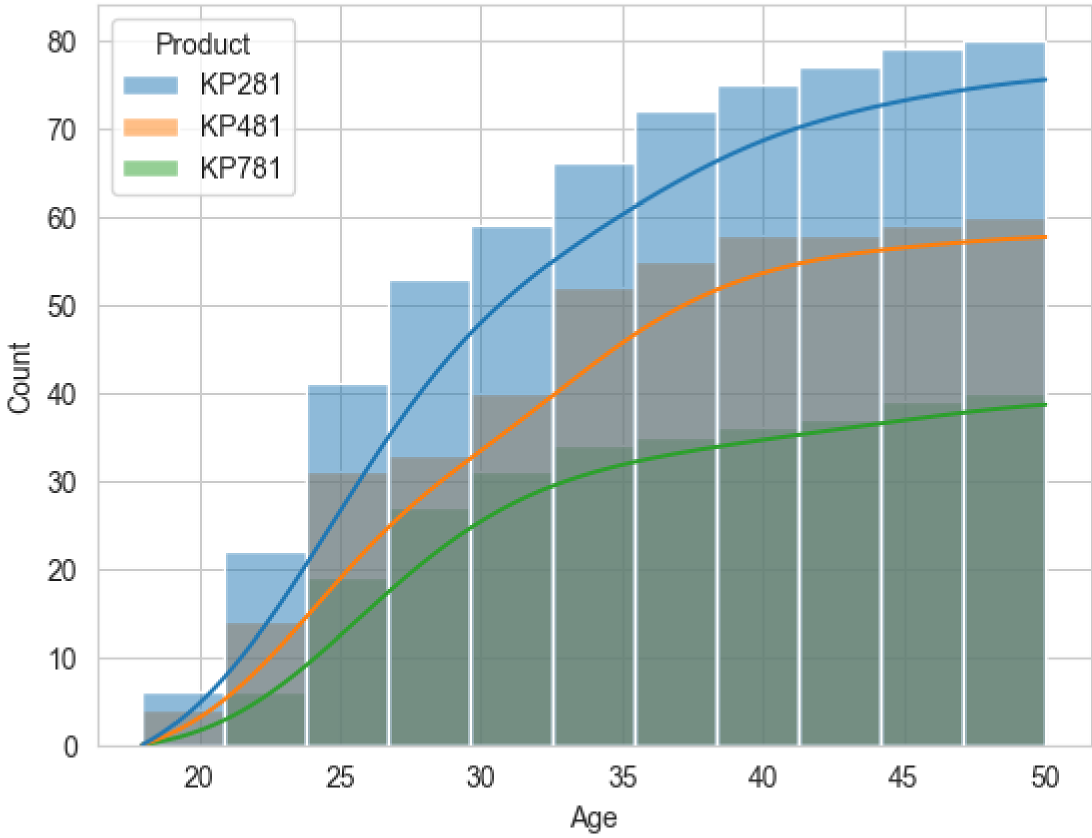

Product
KP281
KP481
KP781
Count
Fitness
0
10
20
30
40
50
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

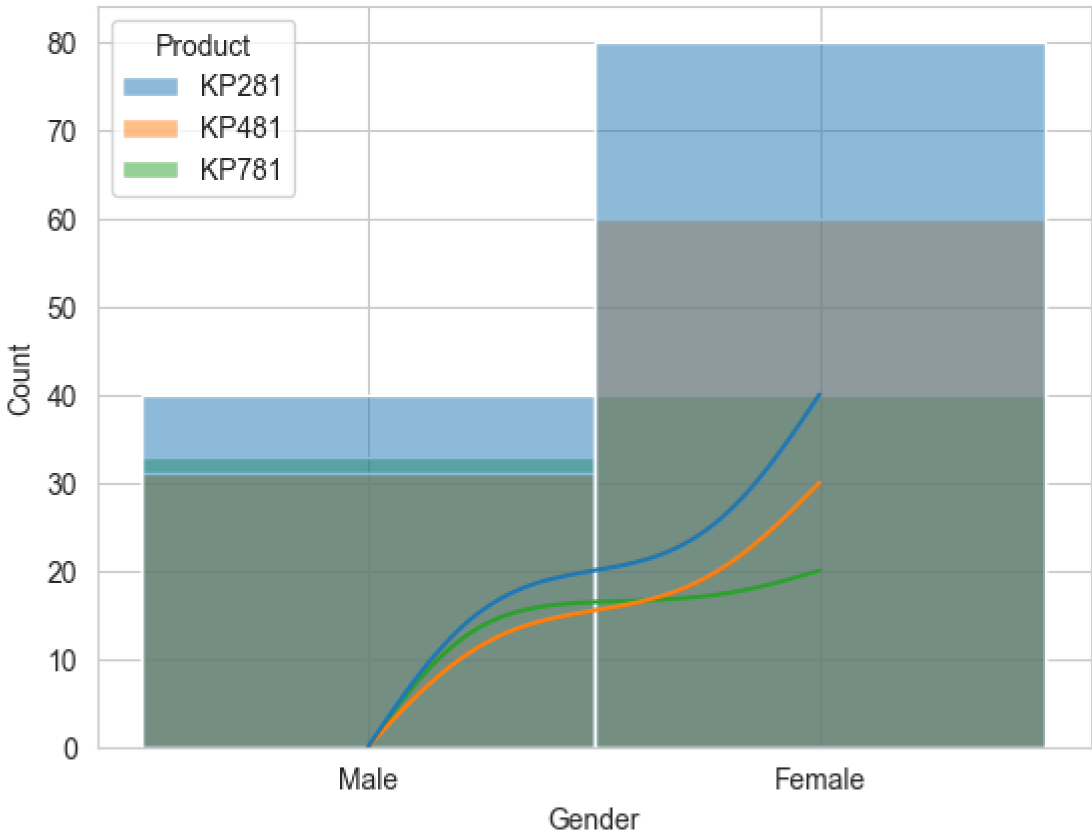


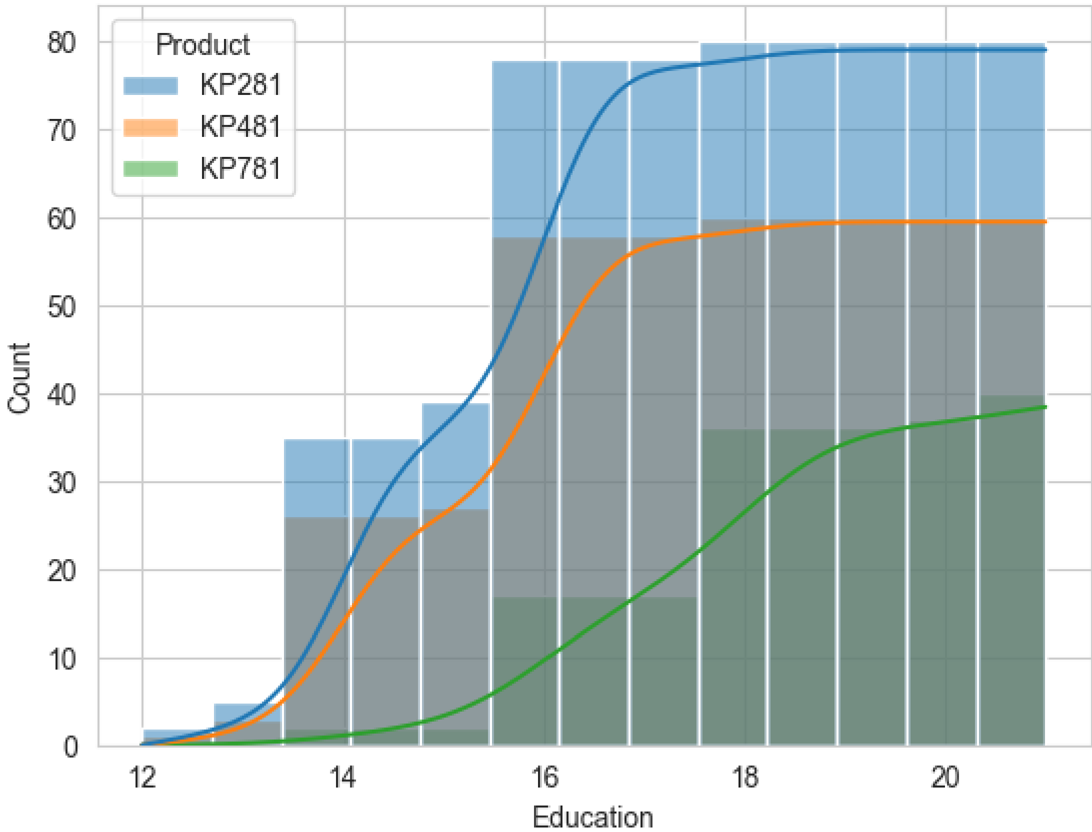


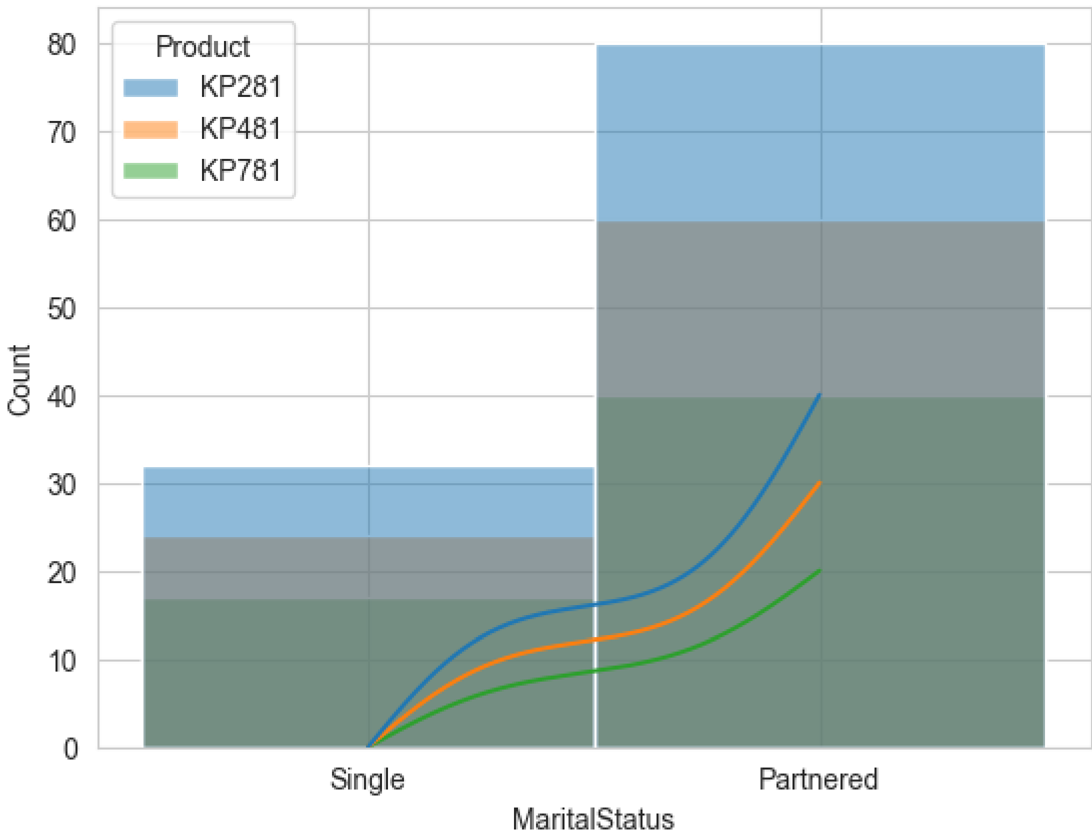
In [29]:
```
#import scipy
#from scipy import stats
from scipy.stats import gaussian_kde
for col in data.columns.tolist()[1:]:
#     print(col)
        sns.histplot(data=df, x=col, kde=True,hue="Product", cumulative=True)
        #sns.set_title(f"Distribution of the feature: {col}", pad=10, fontsize=1
```

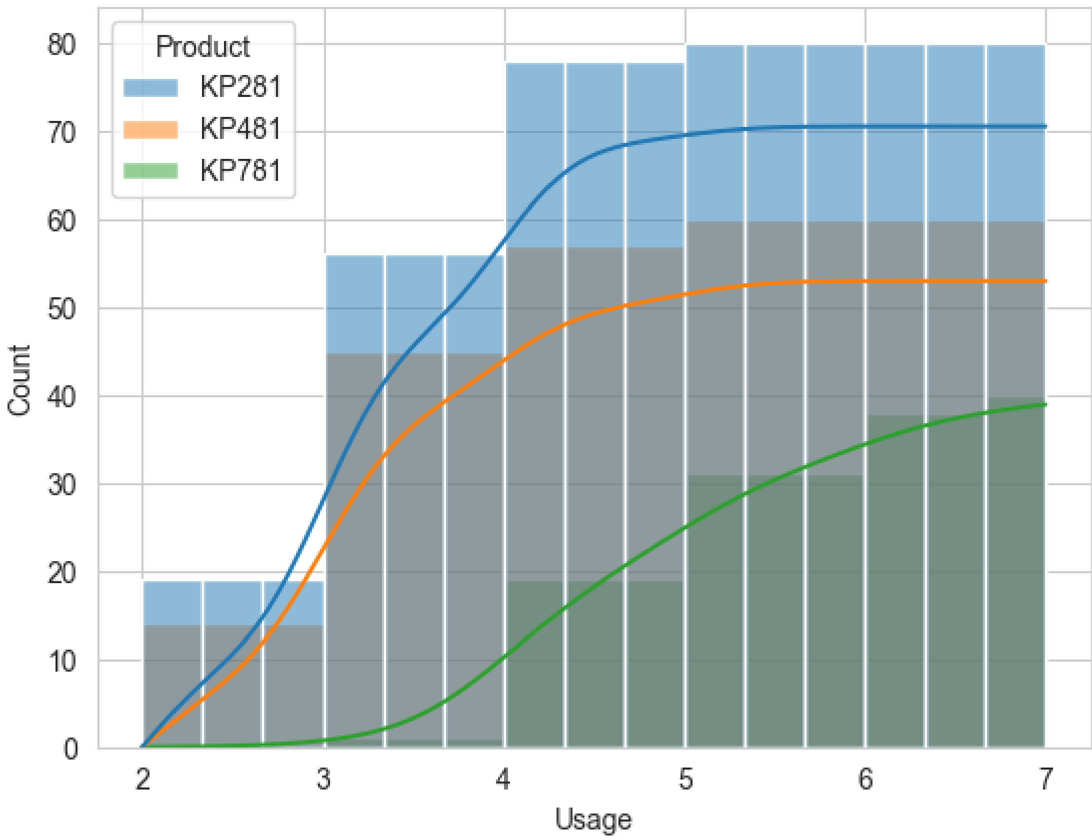
```
#axs[1].set_title("Product vs MaritalStatus", pad=10, fontsize=14)
plt.show()
```

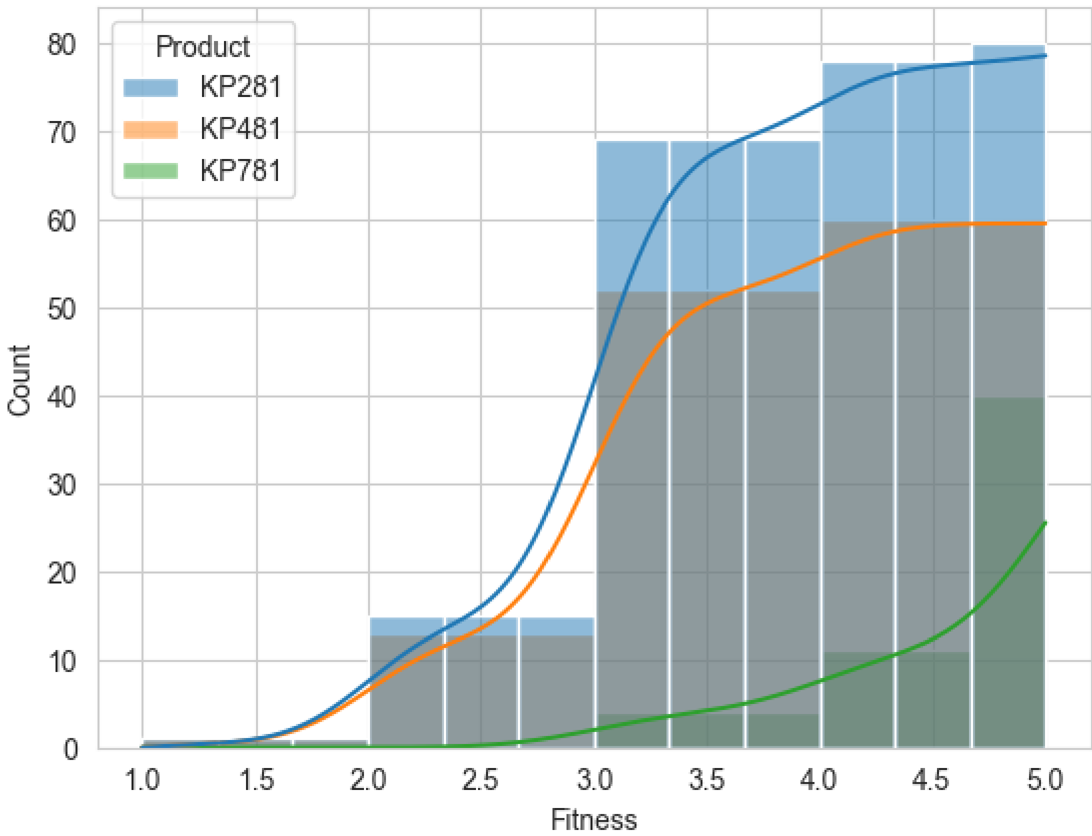


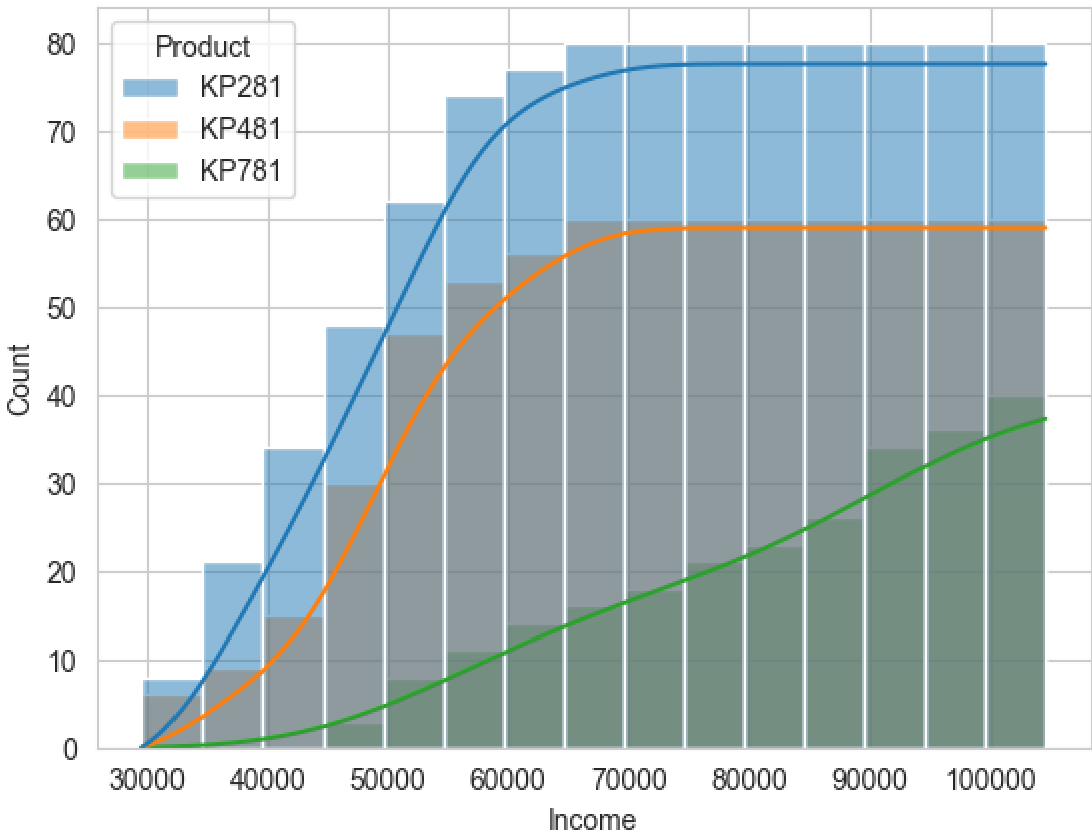


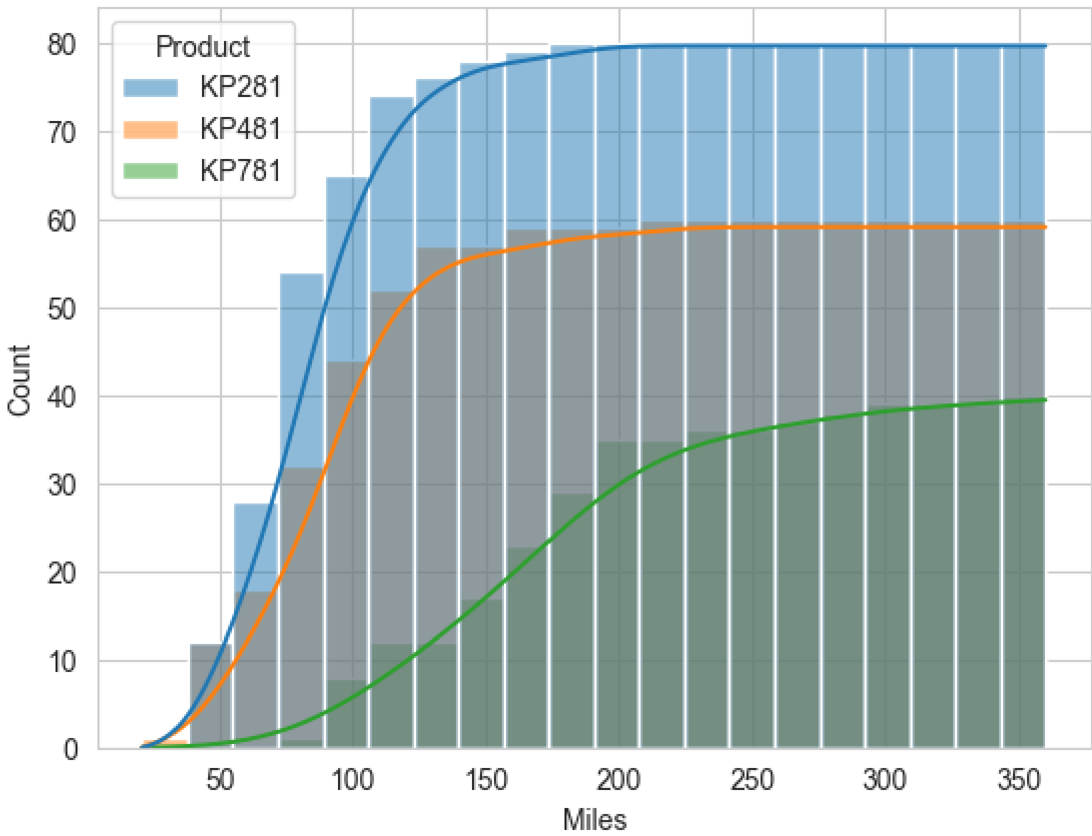

Product
KP281
KP481
KP781
Count
Education
0
10
20
30
40
50
60
70
80
12
14
16
18
20


Product
KP281
KP481
KP781
Count
MaritalStatus
Single
Partnered
0
10
20
30
40
50
60
70
80


Product
KP281
KP481
KP781
Count
Usage
0
10
20
30
40
50
60
70
80
2
3
4
5
6
7


Product
KP281
KP481
KP781
Count
Fitness
0
10
20
30
40
50
60
70
80
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0



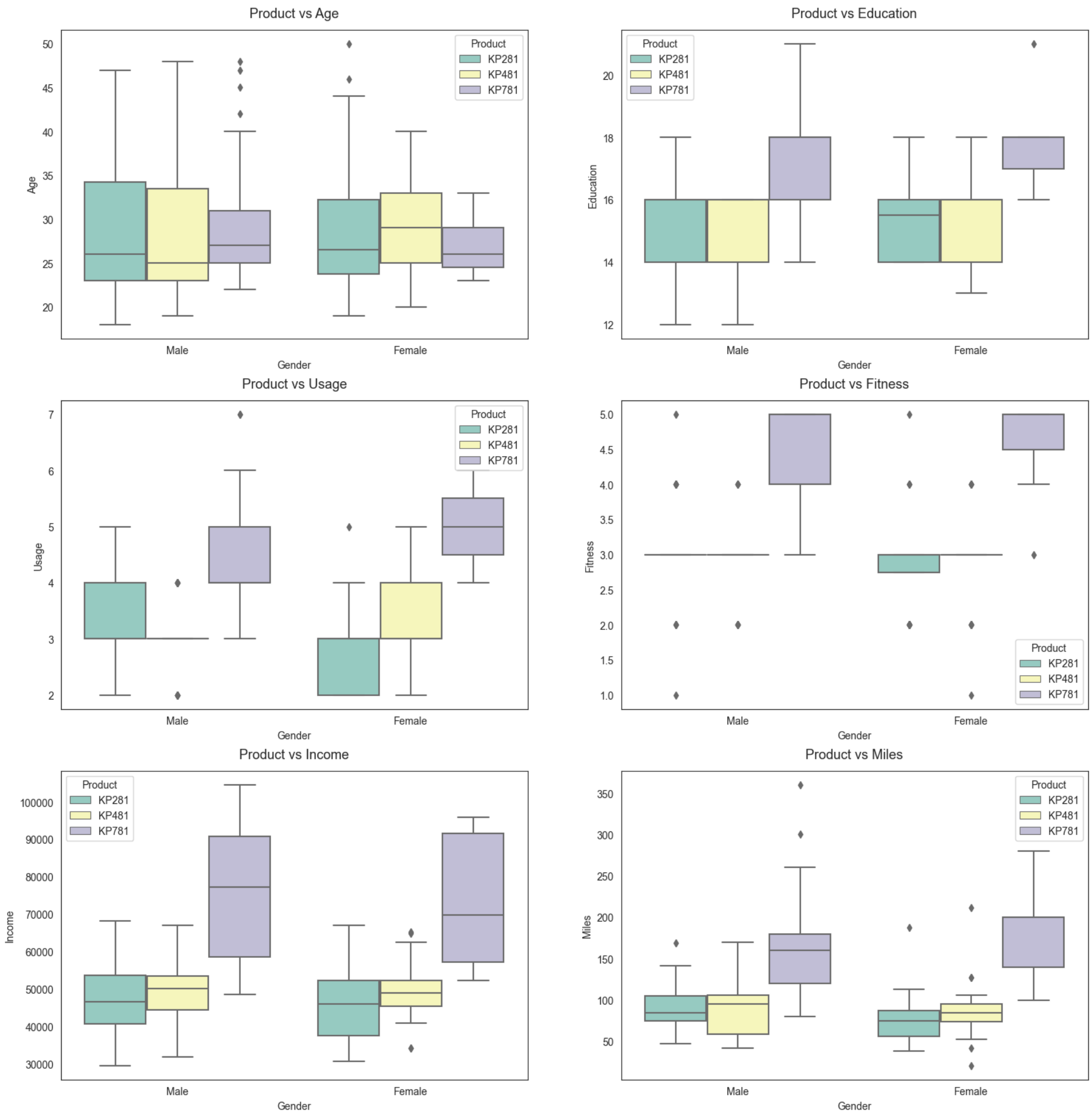


In [ ]:

Multivariate Analysis and Heatmaps

In [43]:
```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(18, 12))
```

```
fig.subplots_adjust(top=1.3)
count = 0
for i in range(3):
    for j in range(2):
        sns.boxplot(data=df, x='Gender', y=attrs[count], hue='Product', ax=axs[i
        axs[i,j].set_title(f"Product vs {attrs[count]}", pad=12, fontsize=13)
        count += 1
```



Roughly few customers with age above 40 use product KP781

Most of the customers are comfortable with KP281 product type

KP481 is the second highest popular product among the younger side of the customer

Customers with product KP781, has been able to cover more miles than other two product types

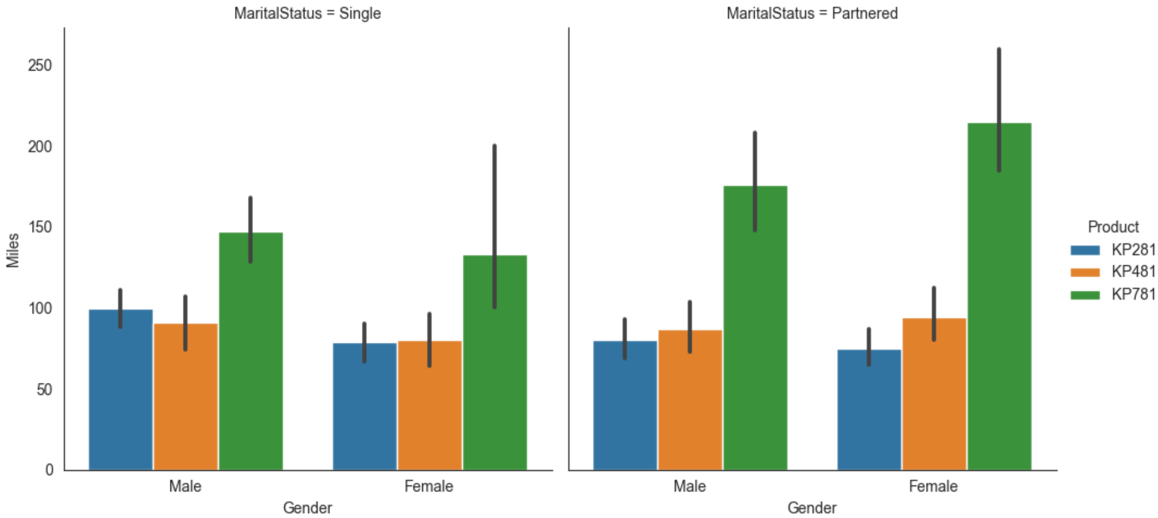
KP481 product is the second most highest miles covering product among the customers

KP281 product customer had covered less distance compared with other two product types

Customers with excellent shape are significantly using KP781 product type

KP481 and KP281 product type are scattered across the fitness rating

In [44]:
```
# Miles covered in each product by gender and their marital status
sns.catplot(x='Gender',y='Miles',hue='Product',col='MaritalStatus',data=data,kir
plt.show()
```



KP781 is more populat among the single and Partnered customers

Among the both marital statuses, Single female does not prefer much of the products.

Partnered Female bought KP781 treadmill compared to Partnered Male.

Single Female customers bought KP281 treadmill slightly more compared to Single Male customers.

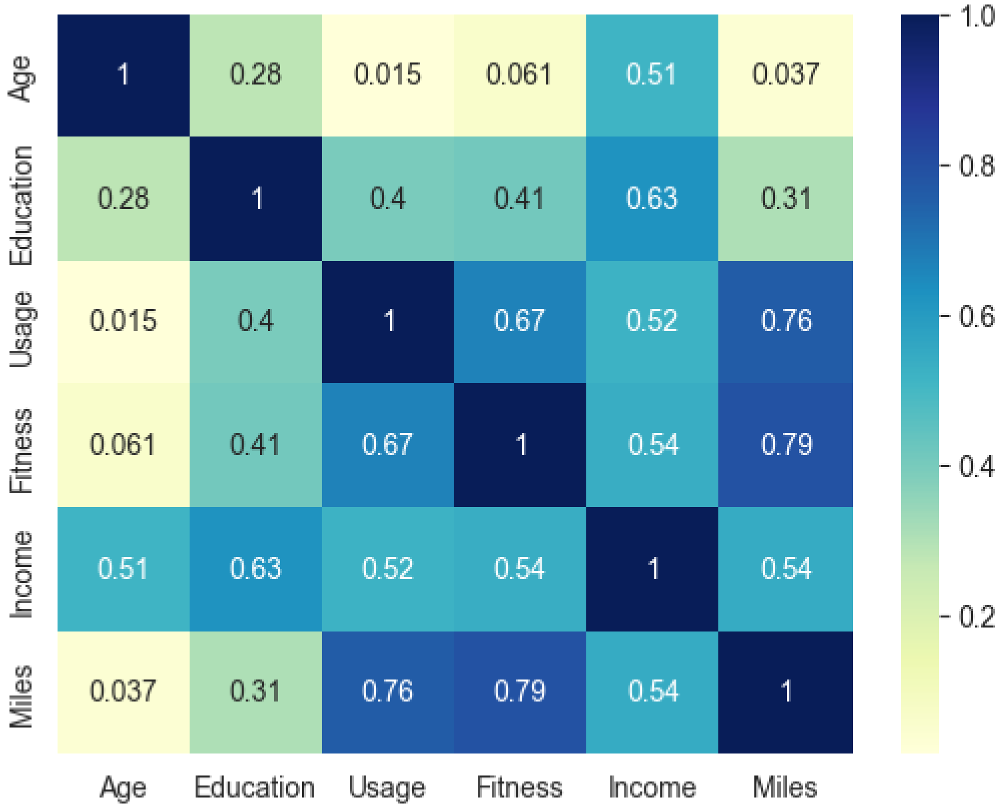
Partnered Male customers bought KP281 treadmill slightly more than Single Male customers.

There are more single Males buying treadmill than single Females.

Single Male customers bought KP781 treadmill compared to single Female.

Partnered customers are more than Single customers.

In [45]:
```
## Heatmaps

# import modules
import matplotlib.pyplot as mp
import pandas as pd
import seaborn as sns

# plotting correlation heatmap
dataplot = sns.heatmap(data.corr(), cmap="YlGnBu", annot=True)

# displaying heatmap
mp.show()
```

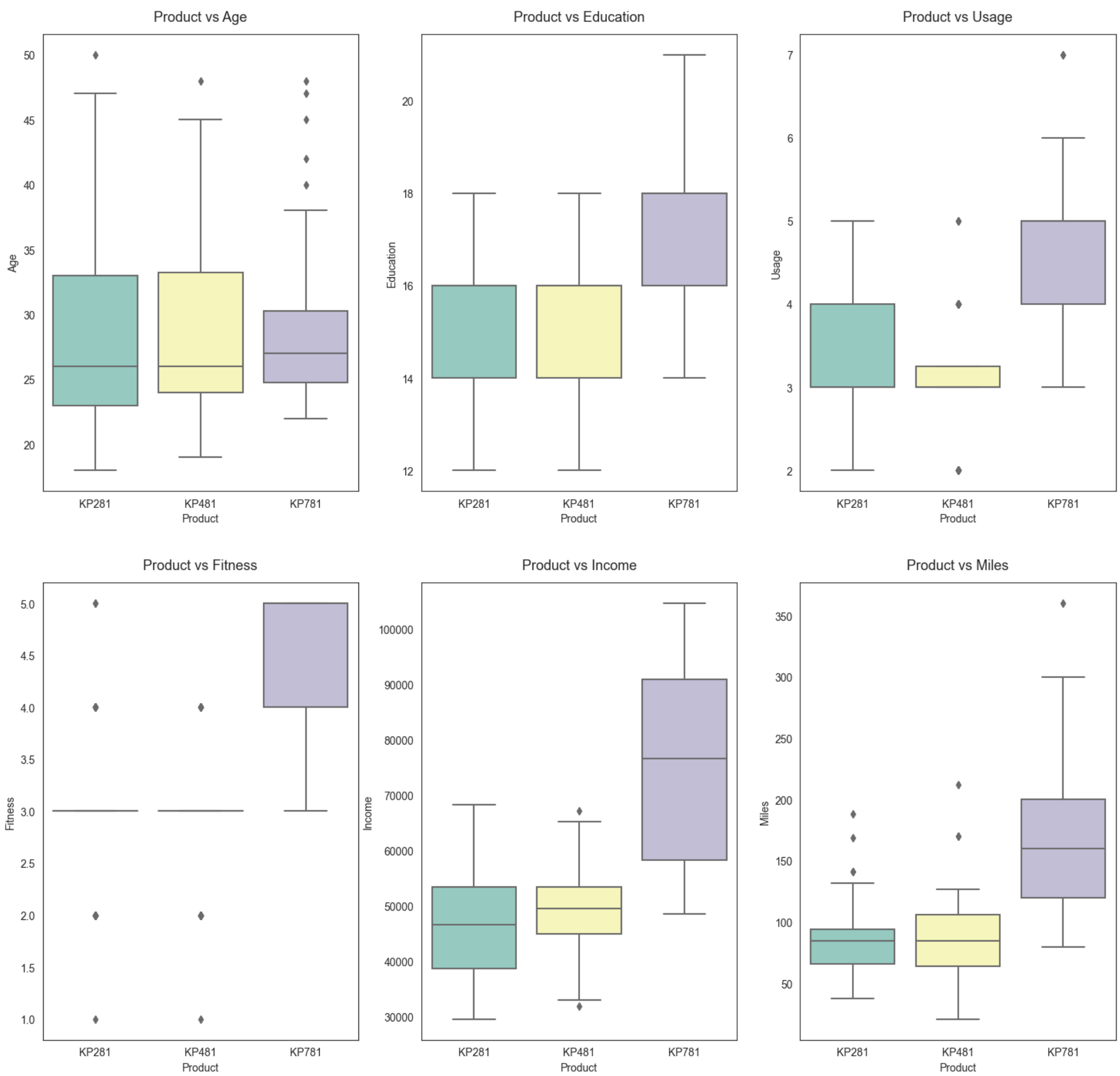


The heatmap indicates below:

1. The Miles targeted by the cutomers is highly related to the Usage, Fitness and Income with positive relation and is moderately correlated with Age and Education of the customer.
2. The Income of the Customer is Moderately directly proportional with the age of the customer. Also the Income depends highly on Education of the customer.
3. High value for Fitness-Miles correlation indicates that the customers who are fit tend to pledge more Miles
4. Usage and Miles have high correlation which is obvious and logical indicating direct proportionality between these two features.

Outlier Analysis

In [46]:
```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(18, 12))
fig.subplots_adjust(top=1.2)
count = 0
for i in range(2):
    for j in range(3):
        sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i,j], palette='
        axs[i,j].set_title(f"Product vs {attrs[count]}", pad=12, fontsize=13)
        count += 1
```



No Missing values detected based on the dataframe describe/info analysis ALso few outliers detected and considering the small amount of data these records are kept as it is. The records indicating the very high miles indicates impractical targets, but as it is customer defined targets the data can not be termerd as invalid. If needed we can consult with the customer and understand if it can be corrected/updated.

In [ ]:

7. Conditional and Marginal probabilities using pd.crosstab()

Two-way contingency tables for each product

In [47]:

```
#two-way contingency tables for each product
import pprint

for col in data.columns.tolist()[1:]:
    print("Feature: ",col)
    print()
    print("Absolute numbers: ")
    pprint.pprint(pd.crosstab(index=data[col],columns=data['Product'],margins=Tr
```

```
    print()
    print("Normalized numbers: ")
    pprint.pprint(pd.crosstab(index=data[col],columns=data['Product'],margins=Tr
    print()
    print(f"Marginal probs by {col}(normalized): ")
    pprint.pprint(pd.crosstab(index=data[col],columns=data['Product'],margins=Tr
    print()
    print("Marginal probs by product(normalized): ")
    pprint.pprint(pd.crosstab(index=data[col],columns=data['Product'],margins=Tr
    print("--"*50)
```

```
Feature:  Age

Absolute numbers:
Product  KP281  KP481  KP781  All
Age
18           1      0      0    1
19           3      1      0    4
20           2      3      0    5
21           4      3      0    7
22           4      0      3    7
23           8      7      3   18
24           5      3      4   12
25           7     11      7   25
26           7      3      2   12
27           3      1      3    7
28           6      0      3    9
29           3      1      2    6
30           2      2      3    7
31           2      3      1    6
32           2      2      0    4
33           2      5      1    8
34           2      3      1    6
35           3      4      1    8
36           1      0      0    1
37           1      1      0    2
38           4      2      1    7
39           1      0      0    1
40           1      3      1    5
41           1      0      0    1
42           0      0      1    1
43           1      0      0    1
44           1      0      0    1
45           0      1      1    2
46           1      0      0    1
47           1      0      1    2
48           0      1      1    2
50           1      0      0    1
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Age
18       0.005556  0.000000  0.000000  0.005556
19       0.016667  0.005556  0.000000  0.022222
20       0.011111  0.016667  0.000000  0.027778
21       0.022222  0.016667  0.000000  0.038889
22       0.022222  0.000000  0.016667  0.038889
23       0.044444  0.038889  0.016667  0.100000
24       0.027778  0.016667  0.022222  0.066667
25       0.038889  0.061111  0.038889  0.138889
26       0.038889  0.016667  0.011111  0.066667
27       0.016667  0.005556  0.016667  0.038889
28       0.033333  0.000000  0.016667  0.050000
29       0.016667  0.005556  0.011111  0.033333
30       0.011111  0.011111  0.016667  0.038889
31       0.011111  0.016667  0.005556  0.033333
32       0.011111  0.011111  0.000000  0.022222
33       0.011111  0.027778  0.005556  0.044444
34       0.011111  0.016667  0.005556  0.033333
35       0.016667  0.022222  0.005556  0.044444
```

```
36       0.005556  0.000000  0.000000  0.005556
37       0.005556  0.005556  0.000000  0.011111
38       0.022222  0.011111  0.005556  0.038889
39       0.005556  0.000000  0.000000  0.005556
40       0.005556  0.016667  0.005556  0.027778
41       0.005556  0.000000  0.000000  0.005556
42       0.000000  0.000000  0.005556  0.005556
43       0.005556  0.000000  0.000000  0.005556
44       0.005556  0.000000  0.000000  0.005556
45       0.000000  0.005556  0.005556  0.011111
46       0.005556  0.000000  0.000000  0.005556
47       0.005556  0.000000  0.005556  0.011111
48       0.000000  0.005556  0.005556  0.011111
50       0.005556  0.000000  0.000000  0.005556
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by Age(normalized):
Product     KP281     KP481     KP781
Age
18       1.000000  0.000000  0.000000
19       0.750000  0.250000  0.000000
20       0.400000  0.600000  0.000000
21       0.571429  0.428571  0.000000
22       0.571429  0.000000  0.428571
23       0.444444  0.388889  0.166667
24       0.416667  0.250000  0.333333
25       0.280000  0.440000  0.280000
26       0.583333  0.250000  0.166667
27       0.428571  0.142857  0.428571
28       0.666667  0.000000  0.333333
29       0.500000  0.166667  0.333333
30       0.285714  0.285714  0.428571
31       0.333333  0.500000  0.166667
32       0.500000  0.500000  0.000000
33       0.250000  0.625000  0.125000
34       0.333333  0.500000  0.166667
35       0.375000  0.500000  0.125000
36       1.000000  0.000000  0.000000
37       0.500000  0.500000  0.000000
38       0.571429  0.285714  0.142857
39       1.000000  0.000000  0.000000
40       0.200000  0.600000  0.200000
41       1.000000  0.000000  0.000000
42       0.000000  0.000000  1.000000
43       1.000000  0.000000  0.000000
44       1.000000  0.000000  0.000000
45       0.000000  0.500000  0.500000
46       1.000000  0.000000  0.000000
47       0.500000  0.000000  0.500000
48       0.000000  0.500000  0.500000
50       1.000000  0.000000  0.000000
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product   KP281     KP481  KP781       All
Age
18       0.0125  0.000000  0.000  0.005556
19       0.0375  0.016667  0.000  0.022222
20       0.0250  0.050000  0.000  0.027778
21       0.0500  0.050000  0.000  0.038889
```

```
22       0.0500  0.000000  0.075  0.038889
23       0.1000  0.116667  0.075  0.100000
24       0.0625  0.050000  0.100  0.066667
25       0.0875  0.183333  0.175  0.138889
26       0.0875  0.050000  0.050  0.066667
27       0.0375  0.016667  0.075  0.038889
28       0.0750  0.000000  0.075  0.050000
29       0.0375  0.016667  0.050  0.033333
30       0.0250  0.033333  0.075  0.038889
31       0.0250  0.050000  0.025  0.033333
32       0.0250  0.033333  0.000  0.022222
33       0.0250  0.083333  0.025  0.044444
34       0.0250  0.050000  0.025  0.033333
35       0.0375  0.066667  0.025  0.044444
36       0.0125  0.000000  0.000  0.005556
37       0.0125  0.016667  0.000  0.011111
38       0.0500  0.033333  0.025  0.038889
39       0.0125  0.000000  0.000  0.005556
40       0.0125  0.050000  0.025  0.027778
41       0.0125  0.000000  0.000  0.005556
42       0.0000  0.000000  0.025  0.005556
43       0.0125  0.000000  0.000  0.005556
44       0.0125  0.000000  0.000  0.005556
45       0.0000  0.016667  0.025  0.011111
46       0.0125  0.000000  0.000  0.005556
47       0.0125  0.000000  0.025  0.011111
48       0.0000  0.016667  0.025  0.011111
50       0.0125  0.000000  0.000  0.005556
-----------------------------------------------------------------------------------
-------------------
Feature:  Gender

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by Gender(normalized):
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
-----------------------------------------------------------------------------------
-------------------
```

```
Feature:  Education

Absolute numbers:
Product    KP281  KP481  KP781  All
Education
12             2      1      0    3
13             3      2      0    5
14            30     23      2   55
15             4      1      0    5
16            39     31     15   85
18             2      2     19   23
20             0      0      1    1
21             0      0      3    3
All           80     60     40  180

Normalized numbers:
Product       KP281     KP481     KP781       All
Education
12         0.011111  0.005556  0.000000  0.016667
13         0.016667  0.011111  0.000000  0.027778
14         0.166667  0.127778  0.011111  0.305556
15         0.022222  0.005556  0.000000  0.027778
16         0.216667  0.172222  0.083333  0.472222
18         0.011111  0.011111  0.105556  0.127778
20         0.000000  0.000000  0.005556  0.005556
21         0.000000  0.000000  0.016667  0.016667
All        0.444444  0.333333  0.222222  1.000000

Marginal probs by Education(normalized):
Product       KP281     KP481     KP781
Education
12         0.666667  0.333333  0.000000
13         0.600000  0.400000  0.000000
14         0.545455  0.418182  0.036364
15         0.800000  0.200000  0.000000
16         0.458824  0.364706  0.176471
18         0.086957  0.086957  0.826087
20         0.000000  0.000000  1.000000
21         0.000000  0.000000  1.000000
All        0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product     KP281     KP481  KP781       All
Education
12         0.0250  0.016667  0.000  0.016667
13         0.0375  0.033333  0.000  0.027778
14         0.3750  0.383333  0.050  0.305556
15         0.0500  0.016667  0.000  0.027778
16         0.4875  0.516667  0.375  0.472222
18         0.0250  0.033333  0.475  0.127778
20         0.0000  0.000000  0.025  0.005556
21         0.0000  0.000000  0.075  0.016667
----------------------------------------------------------------------------------
-------------------
Feature:  MaritalStatus

Absolute numbers:
Product        KP281  KP481  KP781  All
MaritalStatus
Partnered         48     36     23  107
```

```
Single            32     24     17   73
All               80     60     40  180

Normalized numbers:
Product           KP281     KP481     KP781       All
MaritalStatus
Partnered      0.266667  0.200000  0.127778  0.594444
Single         0.177778  0.133333  0.094444  0.405556
All            0.444444  0.333333  0.222222  1.000000

Marginal probs by MaritalStatus(normalized):
Product           KP281     KP481     KP781
MaritalStatus
Partnered      0.448598  0.336449  0.214953
Single         0.438356  0.328767  0.232877
All            0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product        KP281  KP481  KP781       All
MaritalStatus
Partnered        0.6    0.6  0.575  0.594444
Single           0.4    0.4  0.425  0.405556
-----------------------------------------------------------------------------------
-------------------
Feature:  Usage

Absolute numbers:
Product  KP281  KP481  KP781  All
Usage
2           19     14      0   33
3           37     31      1   69
4           22     12     18   52
5            2      3     12   17
6            0      0      7    7
7            0      0      2    2
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Usage
2        0.105556  0.077778  0.000000  0.183333
3        0.205556  0.172222  0.005556  0.383333
4        0.122222  0.066667  0.100000  0.288889
5        0.011111  0.016667  0.066667  0.094444
6        0.000000  0.000000  0.038889  0.038889
7        0.000000  0.000000  0.011111  0.011111
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by Usage(normalized):
Product     KP281     KP481     KP781
Usage
2        0.575758  0.424242  0.000000
3        0.536232  0.449275  0.014493
4        0.423077  0.230769  0.346154
5        0.117647  0.176471  0.705882
6        0.000000  0.000000  1.000000
7        0.000000  0.000000  1.000000
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
```

```
Product   KP281     KP481  KP781       All
Usage
2        0.2375  0.233333  0.000  0.183333
3        0.4625  0.516667  0.025  0.383333
4        0.2750  0.200000  0.450  0.288889
5        0.0250  0.050000  0.300  0.094444
6        0.0000  0.000000  0.175  0.038889
7        0.0000  0.000000  0.050  0.011111
--------------------------------------------------------------------------------
-------------------
Feature:  Fitness

Absolute numbers:
Product  KP281  KP481  KP781  All
Fitness
1            1      1      0    2
2           14     12      0   26
3           54     39      4   97
4            9      8      7   24
5            2      0     29   31
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Fitness
1        0.005556  0.005556  0.000000  0.011111
2        0.077778  0.066667  0.000000  0.144444
3        0.300000  0.216667  0.022222  0.538889
4        0.050000  0.044444  0.038889  0.133333
5        0.011111  0.000000  0.161111  0.172222
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by Fitness(normalized):
Product     KP281     KP481     KP781
Fitness
1        0.500000  0.500000  0.000000
2        0.538462  0.461538  0.000000
3        0.556701  0.402062  0.041237
4        0.375000  0.333333  0.291667
5        0.064516  0.000000  0.935484
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product   KP281     KP481  KP781       All
Fitness
1        0.0125  0.016667  0.000  0.011111
2        0.1750  0.200000  0.000  0.144444
3        0.6750  0.650000  0.100  0.538889
4        0.1125  0.133333  0.175  0.133333
5        0.0250  0.000000  0.725  0.172222
--------------------------------------------------------------------------------
-------------------
Feature:  Income

Absolute numbers:
Product  KP281  KP481  KP781  All
Income
29562        1      0      0    1
30699        1      0      0    1
31836        1      1      0    2
```

```
32973        3      2      0    5
34110        2      3      0    5
...        ...    ...    ...  ...
95866        0      0      1    1
99601        0      0      1    1
103336       0      0      1    1
104581       0      0      2    2
All         80     60     40  180

[63 rows x 4 columns]

Normalized numbers:
Product     KP281     KP481     KP781       All
Income
29562    0.005556  0.000000  0.000000  0.005556
30699    0.005556  0.000000  0.000000  0.005556
31836    0.005556  0.005556  0.000000  0.011111
32973    0.016667  0.011111  0.000000  0.027778
34110    0.011111  0.016667  0.000000  0.027778
...           ...       ...       ...       ...
95866    0.000000  0.000000  0.005556  0.005556
99601    0.000000  0.000000  0.005556  0.005556
103336   0.000000  0.000000  0.005556  0.005556
104581   0.000000  0.000000  0.011111  0.011111
All      0.444444  0.333333  0.222222  1.000000

[63 rows x 4 columns]

Marginal probs by Income(normalized):
Product     KP281     KP481     KP781
Income
29562    1.000000  0.000000  0.000000
30699    1.000000  0.000000  0.000000
31836    0.500000  0.500000  0.000000
32973    0.600000  0.400000  0.000000
34110    0.400000  0.600000  0.000000
...           ...       ...       ...
95866    0.000000  0.000000  1.000000
99601    0.000000  0.000000  1.000000
103336   0.000000  0.000000  1.000000
104581   0.000000  0.000000  1.000000
All      0.444444  0.333333  0.222222

[63 rows x 3 columns]

Marginal probs by product(normalized):
Product   KP281     KP481  KP781       All
Income
29562    0.0125  0.000000  0.000  0.005556
30699    0.0125  0.000000  0.000  0.005556
31836    0.0125  0.016667  0.000  0.011111
32973    0.0375  0.033333  0.000  0.027778
34110    0.0250  0.050000  0.000  0.027778
...         ...       ...    ...       ...
95508    0.0000  0.000000  0.025  0.005556
95866    0.0000  0.000000  0.025  0.005556
99601    0.0000  0.000000  0.025  0.005556
103336   0.0000  0.000000  0.025  0.005556
104581   0.0000  0.000000  0.050  0.011111
```

```
[62 rows x 4 columns]
--------------------------------------------------------------------------------
-------------------
Feature:  Miles

Absolute numbers:
Product  KP281  KP481  KP781  All
Miles
21           0      1      0    1
38           3      0      0    3
42           0      4      0    4
47           9      0      0    9
53           0      7      0    7
56           6      0      0    6
64           0      6      0    6
66          10      0      0   10
74           0      3      0    3
75          10      0      0   10
80           0      0      1    1
85          16     11      0   27
94           8      0      0    8
95           0     12      0   12
100          0      0      7    7
103          3      0      0    3
106          0      8      1    9
112          1      0      0    1
113          8      0      0    8
120          0      0      3    3
127          0      5      0    5
132          2      0      0    2
140          0      0      1    1
141          2      0      0    2
150          0      0      4    4
160          0      0      5    5
169          1      0      0    1
170          0      2      1    3
180          0      0      6    6
188          1      0      0    1
200          0      0      6    6
212          0      1      0    1
240          0      0      1    1
260          0      0      1    1
280          0      0      1    1
300          0      0      1    1
360          0      0      1    1
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Miles
21       0.000000  0.005556  0.000000  0.005556
38       0.016667  0.000000  0.000000  0.016667
42       0.000000  0.022222  0.000000  0.022222
47       0.050000  0.000000  0.000000  0.050000
53       0.000000  0.038889  0.000000  0.038889
56       0.033333  0.000000  0.000000  0.033333
64       0.000000  0.033333  0.000000  0.033333
66       0.055556  0.000000  0.000000  0.055556
74       0.000000  0.016667  0.000000  0.016667
75       0.055556  0.000000  0.000000  0.055556
```

```
80       0.000000  0.000000  0.005556  0.005556
85       0.088889  0.061111  0.000000  0.150000
94       0.044444  0.000000  0.000000  0.044444
95       0.000000  0.066667  0.000000  0.066667
100      0.000000  0.000000  0.038889  0.038889
103      0.016667  0.000000  0.000000  0.016667
106      0.000000  0.044444  0.005556  0.050000
112      0.005556  0.000000  0.000000  0.005556
113      0.044444  0.000000  0.000000  0.044444
120      0.000000  0.000000  0.016667  0.016667
127      0.000000  0.027778  0.000000  0.027778
132      0.011111  0.000000  0.000000  0.011111
140      0.000000  0.000000  0.005556  0.005556
141      0.011111  0.000000  0.000000  0.011111
150      0.000000  0.000000  0.022222  0.022222
160      0.000000  0.000000  0.027778  0.027778
169      0.005556  0.000000  0.000000  0.005556
170      0.000000  0.011111  0.005556  0.016667
180      0.000000  0.000000  0.033333  0.033333
188      0.005556  0.000000  0.000000  0.005556
200      0.000000  0.000000  0.033333  0.033333
212      0.000000  0.005556  0.000000  0.005556
240      0.000000  0.000000  0.005556  0.005556
260      0.000000  0.000000  0.005556  0.005556
280      0.000000  0.000000  0.005556  0.005556
300      0.000000  0.000000  0.005556  0.005556
360      0.000000  0.000000  0.005556  0.005556
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by Miles(normalized):
Product     KP281     KP481     KP781
Miles
21       0.000000  1.000000  0.000000
38       1.000000  0.000000  0.000000
42       0.000000  1.000000  0.000000
47       1.000000  0.000000  0.000000
53       0.000000  1.000000  0.000000
56       1.000000  0.000000  0.000000
64       0.000000  1.000000  0.000000
66       1.000000  0.000000  0.000000
74       0.000000  1.000000  0.000000
75       1.000000  0.000000  0.000000
80       0.000000  0.000000  1.000000
85       0.592593  0.407407  0.000000
94       1.000000  0.000000  0.000000
95       0.000000  1.000000  0.000000
100      0.000000  0.000000  1.000000
103      1.000000  0.000000  0.000000
106      0.000000  0.888889  0.111111
112      1.000000  0.000000  0.000000
113      1.000000  0.000000  0.000000
120      0.000000  0.000000  1.000000
127      0.000000  1.000000  0.000000
132      1.000000  0.000000  0.000000
140      0.000000  0.000000  1.000000
141      1.000000  0.000000  0.000000
150      0.000000  0.000000  1.000000
160      0.000000  0.000000  1.000000
169      1.000000  0.000000  0.000000
170      0.000000  0.666667  0.333333
```

```
180      0.000000  0.000000  1.000000
188      1.000000  0.000000  0.000000
200      0.000000  0.000000  1.000000
212      0.000000  1.000000  0.000000
240      0.000000  0.000000  1.000000
260      0.000000  0.000000  1.000000
280      0.000000  0.000000  1.000000
300      0.000000  0.000000  1.000000
360      0.000000  0.000000  1.000000
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product   KP281     KP481  KP781       All
Miles
21       0.0000  0.016667  0.000  0.005556
38       0.0375  0.000000  0.000  0.016667
42       0.0000  0.066667  0.000  0.022222
47       0.1125  0.000000  0.000  0.050000
53       0.0000  0.116667  0.000  0.038889
56       0.0750  0.000000  0.000  0.033333
64       0.0000  0.100000  0.000  0.033333
66       0.1250  0.000000  0.000  0.055556
74       0.0000  0.050000  0.000  0.016667
75       0.1250  0.000000  0.000  0.055556
80       0.0000  0.000000  0.025  0.005556
85       0.2000  0.183333  0.000  0.150000
94       0.1000  0.000000  0.000  0.044444
95       0.0000  0.200000  0.000  0.066667
100      0.0000  0.000000  0.175  0.038889
103      0.0375  0.000000  0.000  0.016667
106      0.0000  0.133333  0.025  0.050000
112      0.0125  0.000000  0.000  0.005556
113      0.1000  0.000000  0.000  0.044444
120      0.0000  0.000000  0.075  0.016667
127      0.0000  0.083333  0.000  0.027778
132      0.0250  0.000000  0.000  0.011111
140      0.0000  0.000000  0.025  0.005556
141      0.0250  0.000000  0.000  0.011111
150      0.0000  0.000000  0.100  0.022222
160      0.0000  0.000000  0.125  0.027778
169      0.0125  0.000000  0.000  0.005556
170      0.0000  0.033333  0.025  0.016667
180      0.0000  0.000000  0.150  0.033333
188      0.0125  0.000000  0.000  0.005556
200      0.0000  0.000000  0.150  0.033333
212      0.0000  0.016667  0.000  0.005556
240      0.0000  0.000000  0.025  0.005556
260      0.0000  0.000000  0.025  0.005556
280      0.0000  0.000000  0.025  0.005556
300      0.0000  0.000000  0.025  0.005556
360      0.0000  0.000000  0.025  0.005556
--------------------------------------------------------------------------------
-------------------
```

In [48]:
```
round(pd.crosstab(index=[data.Product,data.Fitness],columns=data.Gender,normaliz
```

Out[48]:

Product	Gender / Fitness	Female	Male
KP281	1	0.00	0.56
	2	5.56	2.22
	3	14.44	15.56
	4	1.67	3.33
	5	0.56	0.56
KP481	1	0.56	0.00
	2	3.33	3.33
	3	10.00	11.67
	4	2.22	2.22
KP781	3	0.56	1.67
	4	0.56	3.33
	5	2.78	13.33

In [49]:
```
np.round(((pd.crosstab(data.Product,data.Gender,margins=True))/180)*100,2)
```

Out[49]:

Gender / Product	Female	Male	All
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

In [50]:
```
# 0-21 -> Teen
# 22-35 -> Adult
# 36-45 -> Middle Age
# 46-60 -> Elder Age
data.age_group = pd.cut(data.Age,bins=[0,21,35,45,60],labels=['Teen','Adult','Mi
```

In [51]:
```
# Conditional and Marginal Probabilities with product type and age group
np.round(pd.crosstab(index=data.Product,columns=data.age_group,normalize=True,ma
```

Out[51]:

Age / Product	Teen	Adult	Middle Aged	Elder	All
KP281	5.56	31.11	6.11	1.67	44.44
KP481	3.89	25.00	3.89	0.56	33.33
KP781	0.00	18.89	2.22	1.11	22.22
All	9.44	75.00	12.22	3.33	100.00

Marginal Probability

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 % (cheapest / entry level product)

product KP481 is : 33.33 % (intermediate user level product)

product KP781 is : 22.22 % (Advanced product with ease of use that help in covering longer distance)

Conditional Probabilities

In [52]:

```
np.round((pd.crosstab([data.Product],data.Gender,margins=True,normalize="columns
```

Out[52]:

Gender / Product	Female	Male	All
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 %

KP481 | Female = 38 %

KP781 | Female = 10 %

KP281 | male = 38 %

KP481 | male = 30 %

KP781 | male = 32 %

Probability of Female customer buying KP281(52.63%) is more than male(38.46%).

KP281 is more recommended for female customers.

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

Probability of Female customer buying Product KP481(38.15%) is significantly higher than male (29.80%.)

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

8. Insights

```
In [ ]:
```

Observations

Product vs Age Customers purchasing products KP281 & KP481 are having same Age median value. Customers whose age lies between 25-30, are more likely to buy KP781 product

Product vs Education Customers whose Education is greater than 16, have more chances to purchase the KP781 product. While the customers with Education less than 16 have equal chances of purchasing KP281 or KP481.

Product vs Usage Customers who are planning to use the treadmill greater than 4 times a week, are more likely to purchase the KP781 product. While the other customers are likely to purchasing KP281 or KP481.

Product vs Fitness The more the customer is fit (fitness >= 3), higher the chances of the customer to purchase the KP781 product.

Product vs Income Higher the Income of the customer (Income >= 60000), higher the chances of the customer to purchase the KP781 product.

Product vs Miles If the customer expects to walk/run greater than 120 Miles per week, it is more likely that the customer will buy KP781 product.

1. Product KP281 brings in the highest revenue, KP481 and KP781 come next in line respectively
2. Majority of the customers are in the age group of 22-33 years

~60-40% distribution of the male and female product buyers 3. Majority of the buyers spend 14, 16, 18 years on their education ~60-40% distribution of the single and partnered product buyers 4. Most of the users use the treadmill 3-4 times a week 5. Most of the users rate themselves average in terms of their fitness levels 6. Majority of the users earn between USD35000 and USD60000 annually 7. Majority of the users set target miles expected to be walked/ran between 53 and 132 miles

Insights from product-based study:

1. Relationship with Gender:

Very few female customers buy KP781 product(priced at 2500 dollats); could be cost-related reasons 2. Relationship with Education: Highly educated customers prefer product KP781; they could be more aware of the product's typical features and its usage 3. Relationship with MaritalStatus: no major insights 4. Relationship with Usage: product KP781 is used more compared to others products KP281 and KP481 this product is also prefered by highly-educated customers; this means highly-educated customers tend to exercise more 5. Relationship with Fitness: since highly-educated customer prefer product KP781 because they exercise more; their fitness levels are generally on high scale 6. Relationship with Income and Miles: product KP781 is prefered by high-income earning individuals. Since highly-educated customer prefer product KP781 because they exercise more; their fitness levels are generally on high scale, the number of target miles they set are also higher

Objective: Customer Profiling for Each Product

Customer profiling based on the 3 product categories provided

KP281

Due to affordability this is the mostly sought after product, specially by the entry level consumers.

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

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

This product is used for 70-90 Miles a week

Product is majorly used for 3-4 times a week.

Majority of the average rated fitness enthusiast have opted for this product.

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 USD 35K to USD 55K have preferred this product.

KP481

This is an Intermediate level Product from the features and price pov.

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, indicating no specific group favouring the product.

Customers using this product have targeted 70 to 130 miles per week.

The female customers have preffered this product more frequently compared to male consumers so we can safely conclude that the 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.

Customers with partners are found to be the major group preffering this product.

Male customers have favoured this product slightly more compared to female customers.

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 indicating no major correlation with the marital status and gender bias.

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

KP781

As expected, the customer base is small considering the high proce for this product.

Majority of the customers of this product have great fitness ratings indicating the product to be favourite in the fitness oriented customers.

Customer walk/run average 120 to 200 or more miles per week using this product.

Customers use 4 to 5 times a week at least indicating higher frequency of usage compared to other products.

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

This product is preferred by the customer where the correlation between Education and Income is High ie highly educated high income customer group have preferred this product.

```
In [ ]:
```

9. Recommendations

1. Campaigns to promote KP781 product for females specially
2. Since KP281 and KP481 also brings in significant revenue and is prefered by young & learnings individuals, added features and specialized discounts could help boost sales
3. Based on the Education and income level marketing strategy can be improvised to target the customers accordingly to promote the relevant Products and increase the sales/profit
4. A better, high-end, premium product for highly-educated, high income and active customers to increase revenue.

5. Female who prefer exercising equipments are very low here. Hence, we should run a marketing campaign on to encourage women to exercise more

6. KP281 & KP481 treadmills are preferred by the customers whose annual income lies in the range of 39K - 53K Dollars. These models can be promoted as budget treadmills to increase the reach further.

7. As KP781 provides more features and functionalities, the treadmill should be marketed for professionals and athletes as it has higher chances.

8. Research required for expanding market beyond 50 years of age considering health pros and cons.

```
In [ ]:
```