Aerofit Case study

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.

In [51]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [52]:
```
df = pd.read_csv('aerofit.csv')
df.head()
```

Out[52]:

	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 [53]:
```
#finding no. of rows and columns in dataset
print('Number of rows in dataset :',df.shape[0])
print('Number of columns in dataset :',df.shape[1])
```

```
Number of rows in dataset : 180
Number of columns in dataset : 9
```

In [54]:
```
#checking data types of columns in given dataset
df.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
```

In [55]:
```
df.describe()
```

Out[55]:

	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

In [56]:
```
#Finding Unique Values
df.nunique()
```

Out[56]:
```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

In [57]:
```
#Finding Missing values in DataFrame
df.isnull().any()
```

Out[57]:
```
Product          False
Age              False
Gender           False
Education        False
MaritalStatus    False
Usage            False
Fitness          False
Income           False
Miles            False
dtype: bool
```

In [58]:
```
df['Product'].value_counts()
```

Out[58]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

In [59]:
```
df['Usage'].value_counts()
```

Out[59]:
```
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

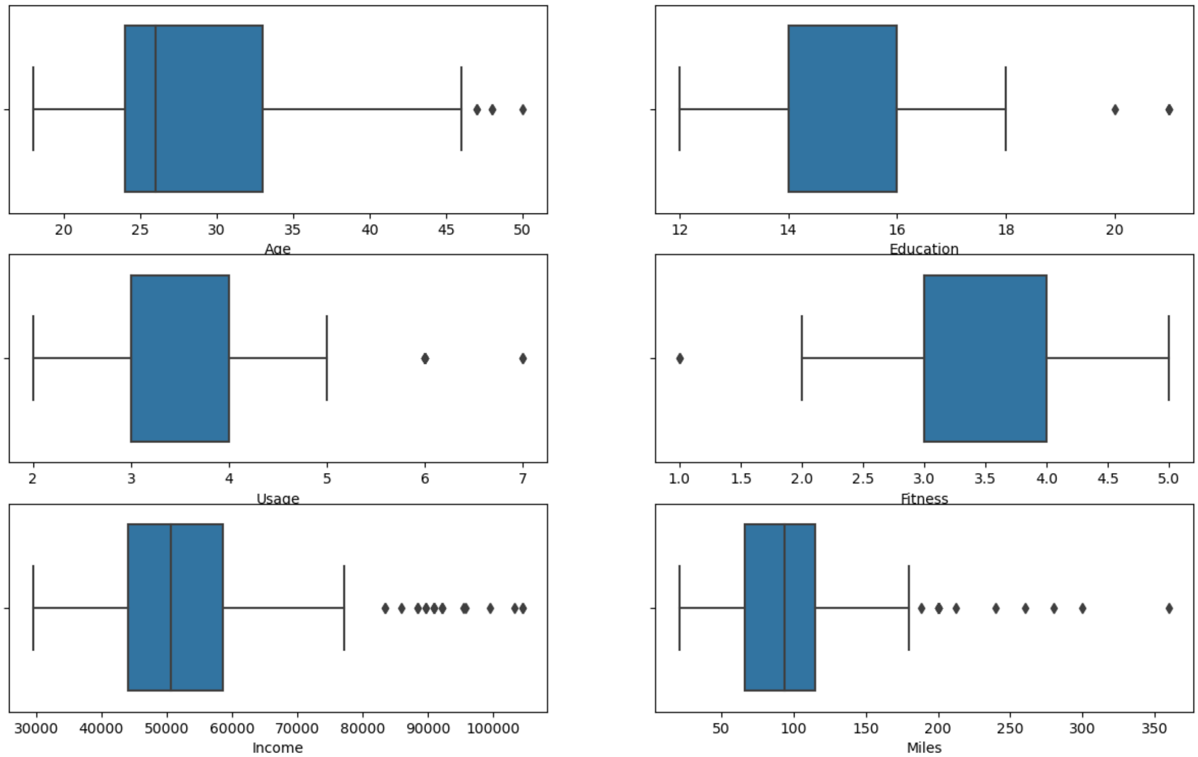
In [60]:
```
df.groupby('MaritalStatus')['Product'].count()
```

Out[60]:
```
MaritalStatus
Partnered    107
Single        73
Name: Product, dtype: int64
```

In [61]:
```
#Detecting Outliers
fig, axis = plt.subplots(3,2,figsize = (15,9))

sns.boxplot(data=df,x='Age',ax = axis[0,0],orient = 'h')
sns.boxplot(data=df,x='Education',ax = axis[0,1],orient = 'h')
sns.boxplot(data=df,x='Usage',ax = axis[1,0],orient = 'h')
```

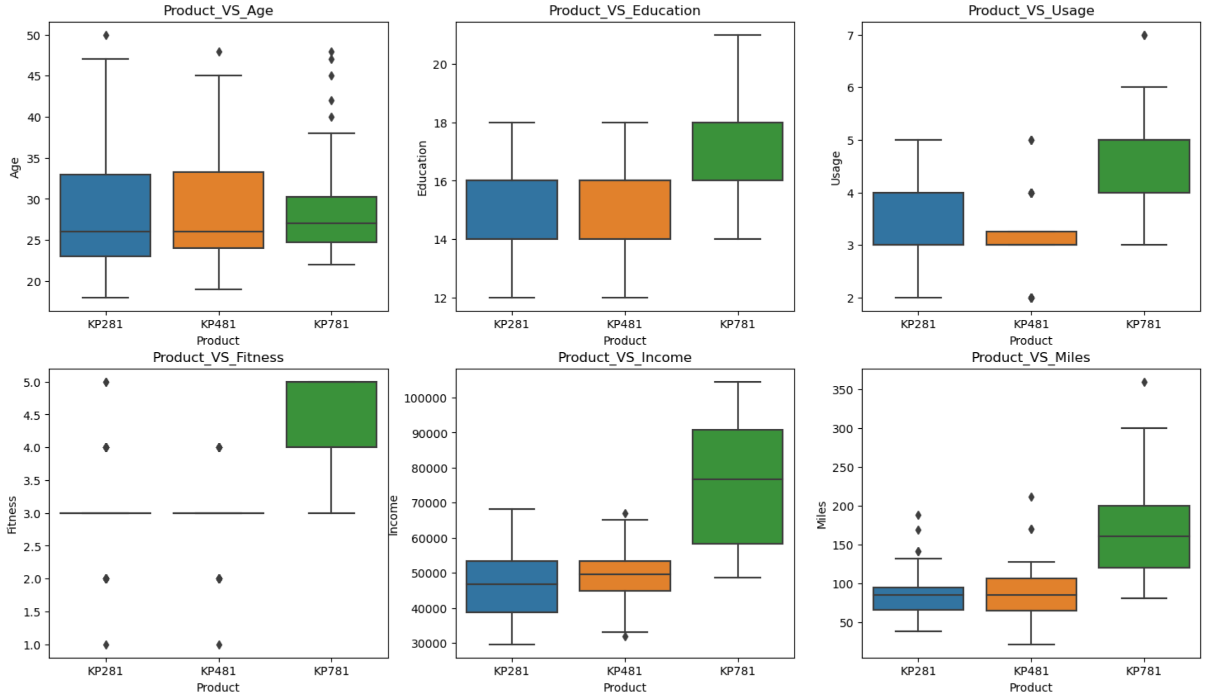
```
sns.boxplot(data=df,x='Fitness',ax = axis[1,1],orient = 'h')
sns.boxplot(data=df,x='Income',ax = axis[2,0],orient = 'h')
sns.boxplot(data=df,x='Miles',ax = axis[2,1],orient = 'h')
plt.show()
```



The boxplots above make it evident which Income and Miles have more outliers than others.

In [62]:
```
#checking whether given features have any effect on product purchase
fig, axis = plt.subplots(2,3,figsize = (18,10))

sns.boxplot(data=df,x='Product',y='Age',ax = axis[0,0])
axis[0,0].set_title('Product_VS_Age')
sns.boxplot(data=df,x='Product',y='Education',ax = axis[0,1])
axis[0,1].set_title('Product_VS_Education')
sns.boxplot(data=df,x='Product',y='Usage',ax = axis[0,2])
axis[0,2].set_title('Product_VS_Usage')
sns.boxplot(data=df,x='Product',y='Fitness',ax = axis[1,0])
axis[1,0].set_title('Product_VS_Fitness')
sns.boxplot(data=df,x='Product',y='Income',ax = axis[1,1])
axis[1,1].set_title('Product_VS_Income')
sns.boxplot(data=df,x='Product',y='Miles',ax = axis[1,2])
axis[1,2].set_title('Product_VS_Miles')
plt.show()
```



Observations:

1. Product Vs Age :
   - With very few outliers, the customers purchasing KP281 and KP481 are in the 24-34 age range. Their median age is the same.
   - The median age of customers buying KP781 is slightly higher than in other categories, and their maximum number of people lies in the age group 25–30.

1. Product Vs Education:
   - Buyers of KP281 and KP481 treadmills are more likely to be between the ages of 14 and 16.
   - Constomers who purchased the KP781 treadmill had between 16 and 18 years of education.

1. Product Vs Usage:

   - The KP781 model is typically purchased by customers that use the tredmill four to five times per week.
   - While others whose usage is less than 4 times a week are likely to purchase rest of the models.
2. Product Vs Fitness:

   - Customers having KP781 model have high fitness levels (typically between 4 and 5).

- while rest of the buyer having KP281 and KP481 have average fitness level 3.

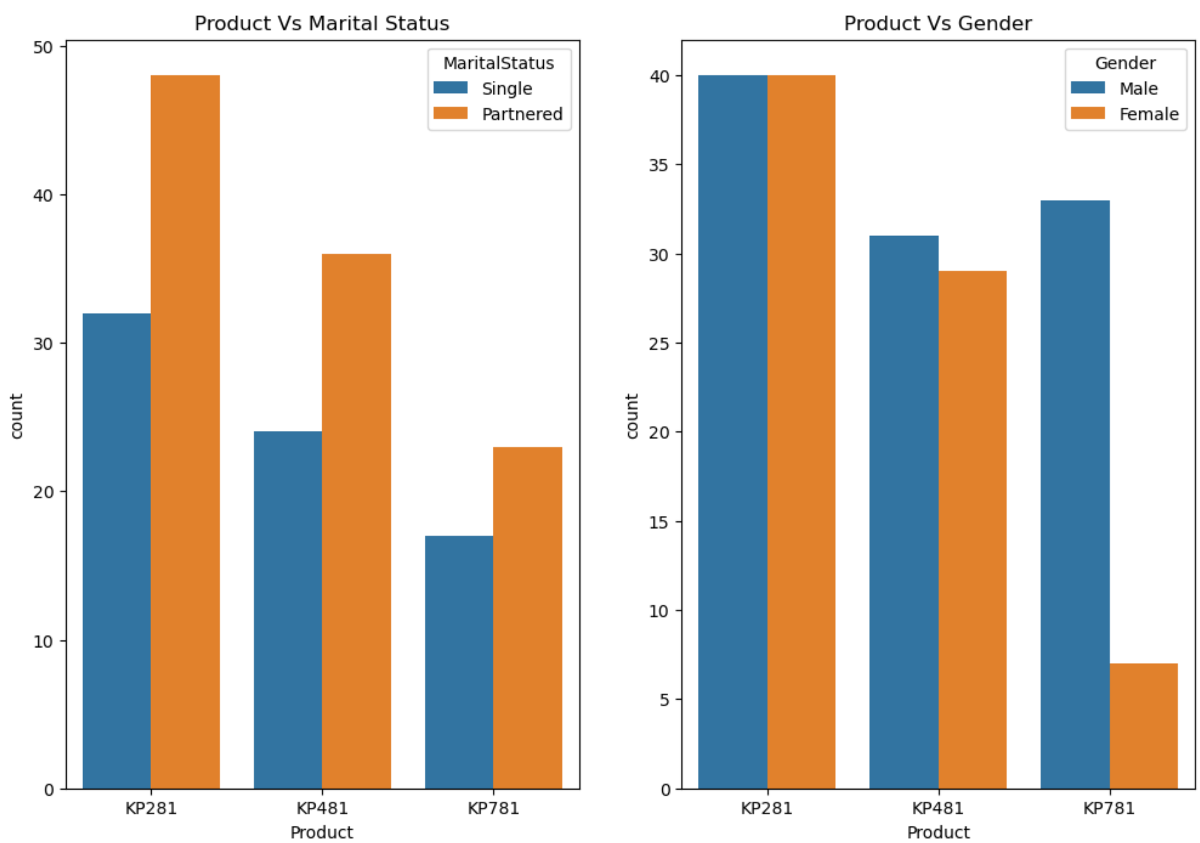
3. Product Vs Income:

- High income individuals buy the KP781 model.
- While low and mid-level income customers have bought rests of the models

4. Product Vs Mile:

- Those who typically log over 120 miles a week of walking or running have bought the KP781 product.
- Conversely, people who run or walk 50–100 miles per week are more likely to purchase KP281 and KP481 items.

In [63]:
```
fig, axs = plt.subplots(1,2,figsize = (12,8))
sns.countplot(data=df,x = 'Product',hue='MaritalStatus',ax=axs[0])
axs[0].set_title('Product Vs Marital Status')
sns.countplot(data=df,x = 'Product',hue='Gender',ax=axs[1])
axs[1].set_title('Product Vs Gender')
plt.show()
```

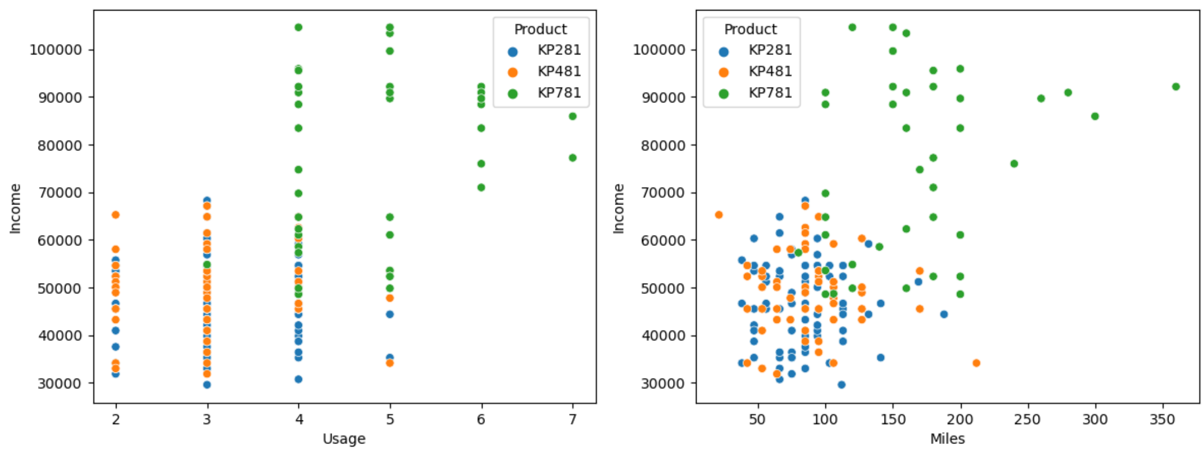


Observations :

- Product Vs Marital Status

    - Customers who are partenered have a higher likelihood of making purchases.
    - The most popular product among consumers is KP281, which is followed by KP481 and KP781.
- Product Vs Gender
    - Given that the product is purchased by an equal number of male and female consumers, KP281 is the most popular item purchased by both genders.
    - KP781 is more popular in male customers.
    - Men tend to be more fitness conscious than women, as seen by the overall higher percentage of male clients.

In [64]:
```
#Multivariate Analysis
fig, axs = plt.subplots(1,2,figsize = (14,5))
sns.scatterplot(data=df ,x='Usage',y='Income',hue='Product',ax=axs[0])
sns.scatterplot(data=df ,x='Miles',y='Income',hue='Product',ax=axs[1])
plt.show()
```

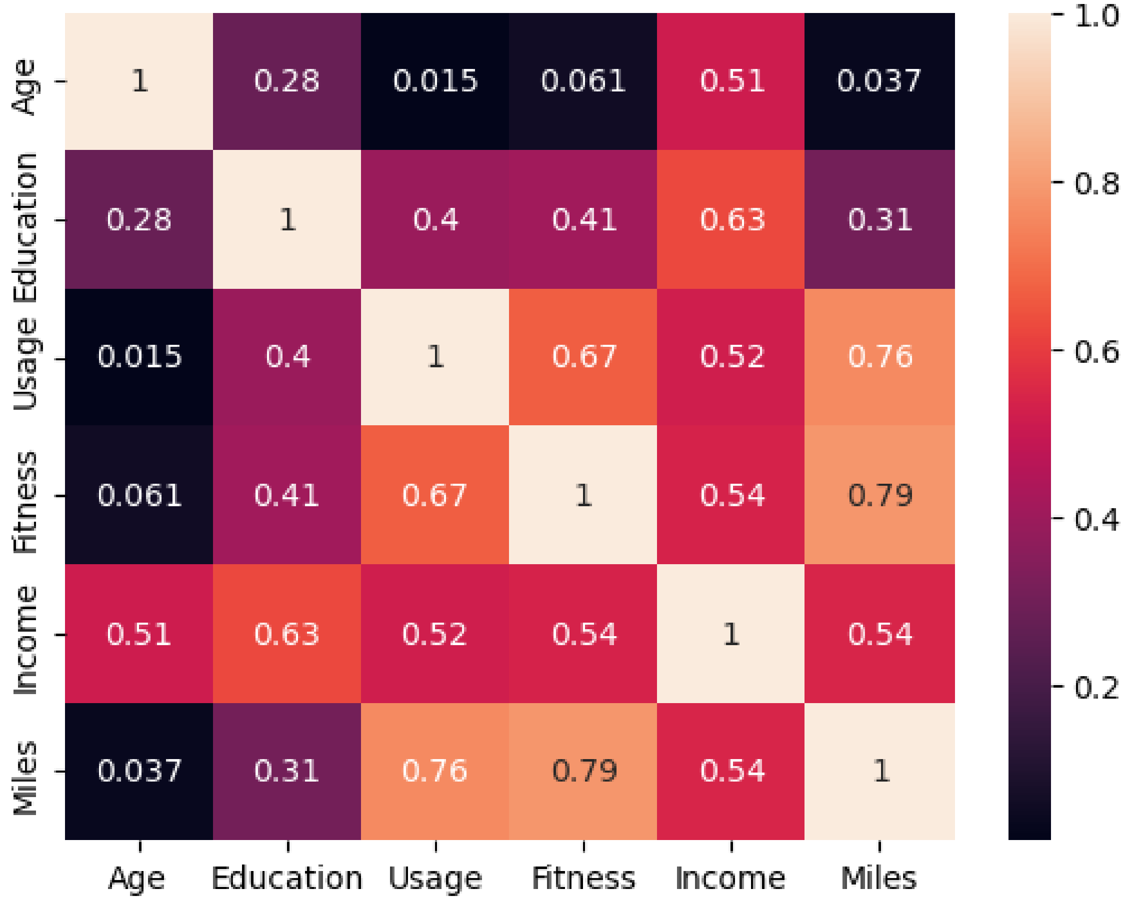


Obeservations:

- It's clear that those with high earnings use the KP781 treadmill the most, whereas people with low and intermediate incomes prefer the KP281 and KP481 versions.

In [65]:
```
#Correlation between different factors
sns.heatmap(df.corr(),annot = True)
```

Out[65]: <AxesSubplot:>



Customer Profiling

In [66]:
```
dff = df[['Product','Gender','MaritalStatus']].melt()
dff.groupby(['variable','value'])[['value']].count()/len(df) *100
```

Out[66]:

		value
variable	**value**	
Gender	**Female**	42.222222
	Male	57.777778
MaritalStatus	**Partnered**	59.444444
	Single	40.555556
Product	**KP281**	44.444444
	KP481	33.333333
	KP781	22.222222

In [67]:
```
#Percentage of male and female
df['Gender'].value_counts(normalize = True)*100
```

Out[67]:
```
Male      57.777778
Female    42.222222
Name: Gender, dtype: float64
```

In [68]:
```
#Categorization of Products
df['Product'].value_counts(normalize=True)*100
```

Out[68]:
```
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: Product, dtype: float64
```

Calculation of Marginal & Conditional Probabilities

In [69]:
```
pd.crosstab(index = df['Gender'],columns = df['Product'],margins = True,nor
```

Out[69]:

Product Gender	KP281	KP481	KP781	All
Female	22.222222	16.111111	3.888889	42.222222
Male	22.222222	17.222222	18.333333	57.777778
All	44.444444	33.333333	22.222222	100.000000

Probability :

- P(Male) = 57.78%
- P(Female) = 42.22%
- P(KP281) = 44.44%
- P(KP481) = 33.33%
- P(KP781) = 22.22%
- P(KP281/Male) = 22.22%
- P(KP481/Male) = 17.22%
- P(KP781/Male) = 18.33%
- P(KP281/Female) = 22.22%
- P(KP481/Female) = 16.11%
- P(KP781/Female) = 3.88%

Insights & Recommendations:

- Since the majority of our customers are in the 25–35 age range, we should focus on reaching more of them in order to boost sales.
- In order to boost sales of the KP781 product, we ought to focus on individuals with higher incomes and those with more than 16 years of schooling.

- We should target married customers to enhance sales because they are more likely to make purchases of all kinds.
- To increase sales, we should target more men as they are more inclined to make purchases.

In [ ]: