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

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

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 s
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.

1. Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset.

```
Aerofit = pd.read_csv("Aerofit.txt")
```

```
print(Aerofit.head())
```

```
  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
```

```
print(Aerofit.isna().sum())
```

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

```
print(Aerofit.info())
```

```
 #   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
None
```

```
print(Aerofit.dtypes)
```

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

```
print(Aerofit.shape)
```

```
(180, 9)
```

```
print(Aerofit.Product.value_counts())
```

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

```
print(Aerofit.Product.unique())
```

```
['KP281' 'KP481' 'KP781']
```

Business Insights :

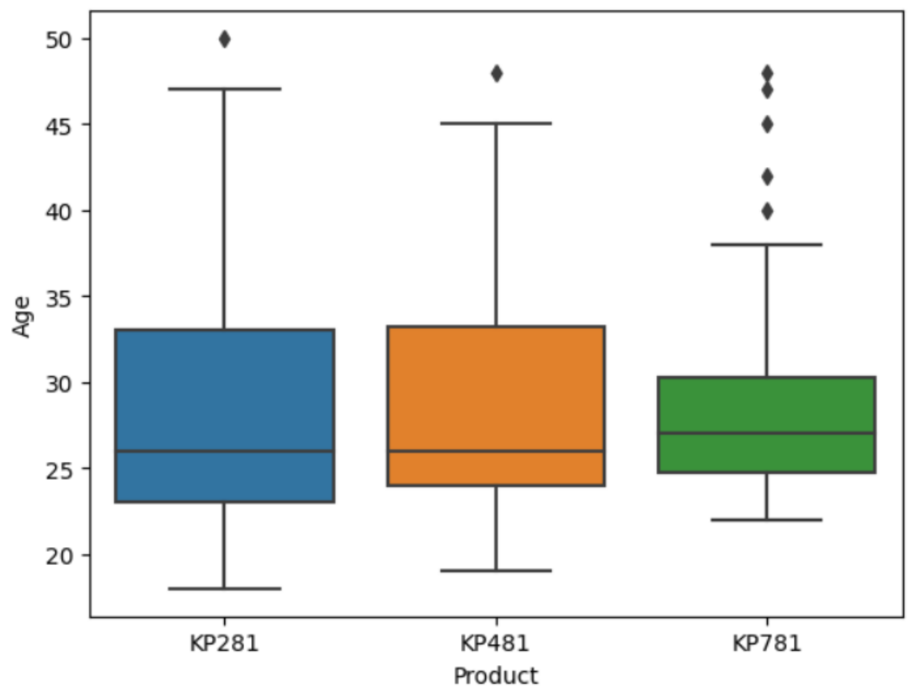
1. DataFrame has 180 rows and 9 columns.
2. There is no missing and Null values present in the dataframe.
3. Data has 3 unique products.

2. Detect Outliers (using boxplot, “describe” method by checking the difference between mean and median).

```
print(Aerofit.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
```

```
sns.boxplot(x = "Product", y = "Age", data = Aerofit )
```



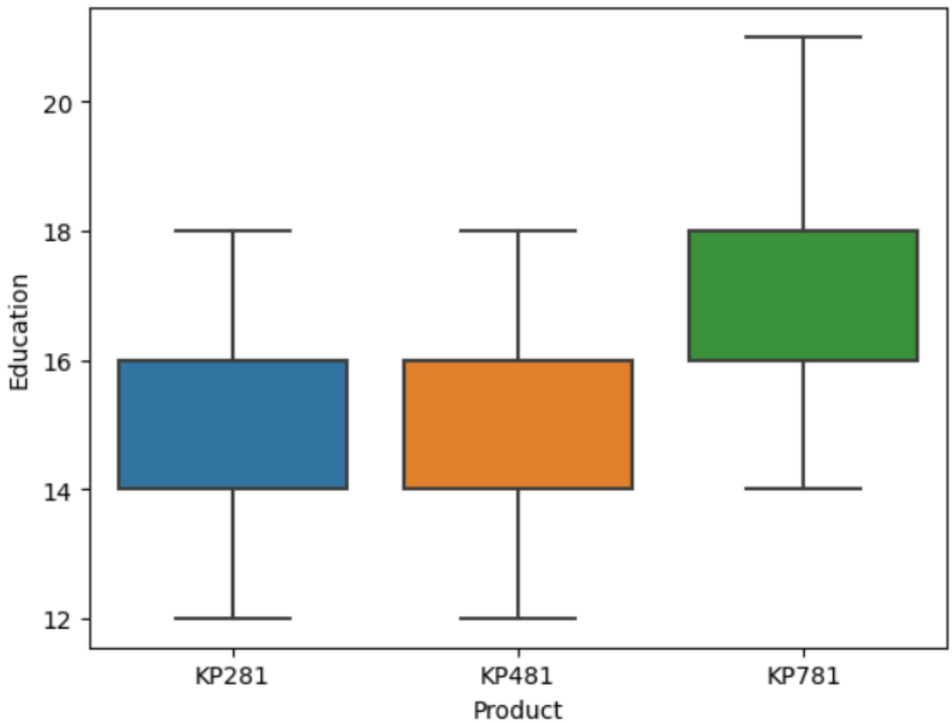

```
KP281_BoxPlot_Age_Max = min(KP281_Product.Age.max() ,
(KP281_Product.Age.quantile(0.75)) + (1.5 *
((KP281_Product.Age.quantile(0.75)) -
KP281_Product.Age.quantile(0.25))))
KP481_BoxPlot_Age_Max = min(KP481_Product.Age.max() ,
(KP481_Product.Age.quantile(0.75)) + (1.5 *
((KP481_Product.Age.quantile(0.75)) -
KP481_Product.Age.quantile(0.25))))
KP781_BoxPlot_Age_Max = min(KP781_Product.Age.max() ,
(KP781_Product.Age.quantile(0.75)) + (1.5 *
((KP781_Product.Age.quantile(0.75)) -
KP781_Product.Age.quantile(0.25))))

KP281_outliar_Age_count = KP281_Product.Age[KP281_Product.Age
> KP281_BoxPlot_Age_Max].count()
KP481_outliar_Age_count = KP481_Product.Age[KP481_Product.Age
> KP481_BoxPlot_Age_Max].count()
KP781_outliar_Age_count = KP781_Product.Age[KP781_Product.Age
> KP781_BoxPlot_Age_Max].count()


print(f"KP281_BoxPlot_Age_Max : {KP281_BoxPlot_Age_Max}")
print(f"KP481_BoxPlot_Age_Max : {KP481_BoxPlot_Age_Max}")
print(f"KP781_BoxPlot_Age_Max : {KP781_BoxPlot_Age_Max}")
print()
print(f"KP281_outliar_Age_count : {KP281_outliar_Age_count}")
print(f"KP481_outliar_Age_count : {KP481_outliar_Age_count}")
print(f"KP781_outliar_Age_count : {KP781_outliar_Age_count}")
```

```
KP281_BoxPlot_Age_Max : 48.0
KP481_BoxPlot_Age_Max : 47.125
KP781_BoxPlot_Age_Max : 38.5

KP281_outliar_Age_count : 1
KP481_outliar_Age_count : 1
KP781_outliar_Age_count : 5
```

Business Insights :

1. Median age of KP281 is same as median of KP481. median age of KP781 is higher than KP281 and KP481.
2. KP281 and KP481 has 1 outlier age and KP781 has 5 outlier age.
3. Standard distribution in income and miles is very high.
4. More than 75% of people have education less than or equal to 16 years as per data.
5. customers who's age is between 25-30 are more likely to buy KP781 product.

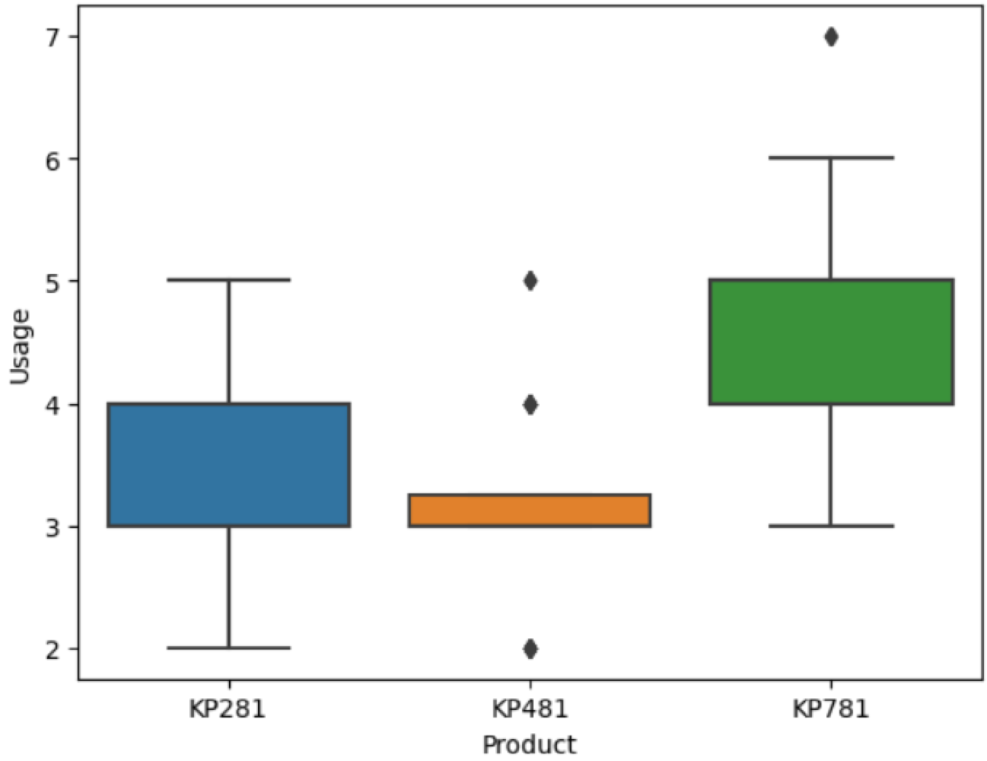
```
sns.boxplot(x = "Product" , y = "Education" , data = Aerofit )
```



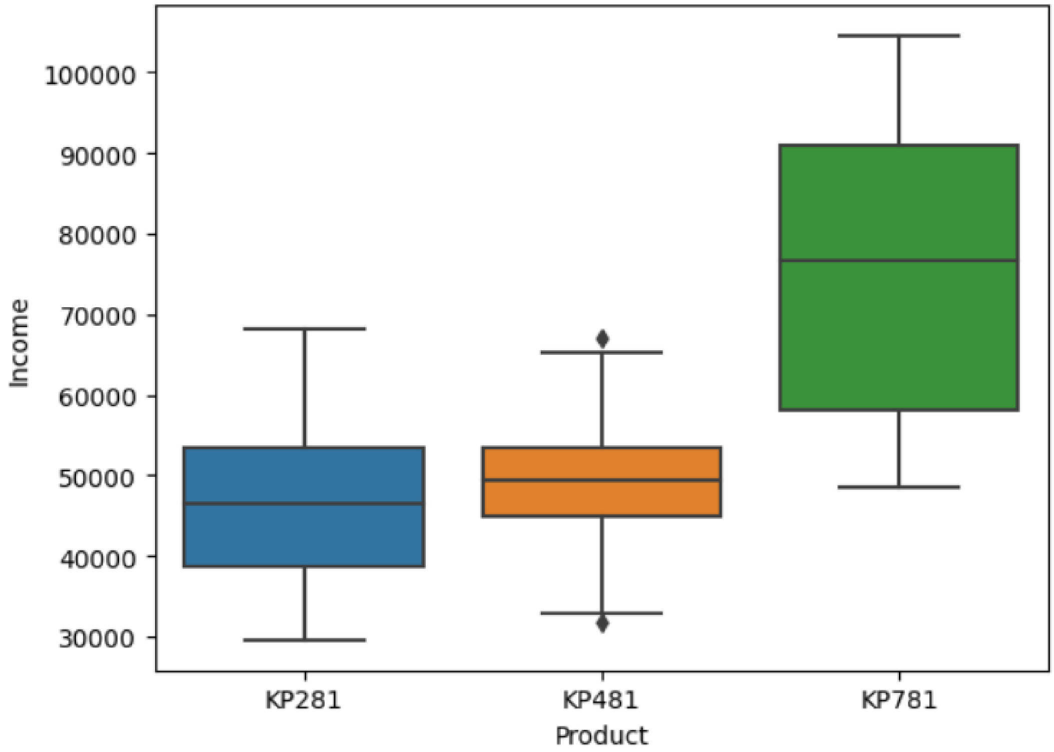
Business Insights :

1. KP781 is mainly preferred by highly educated customers.

```
sns.boxplot(x = "Product" , y = "Usage" , data = Aerofit)
```



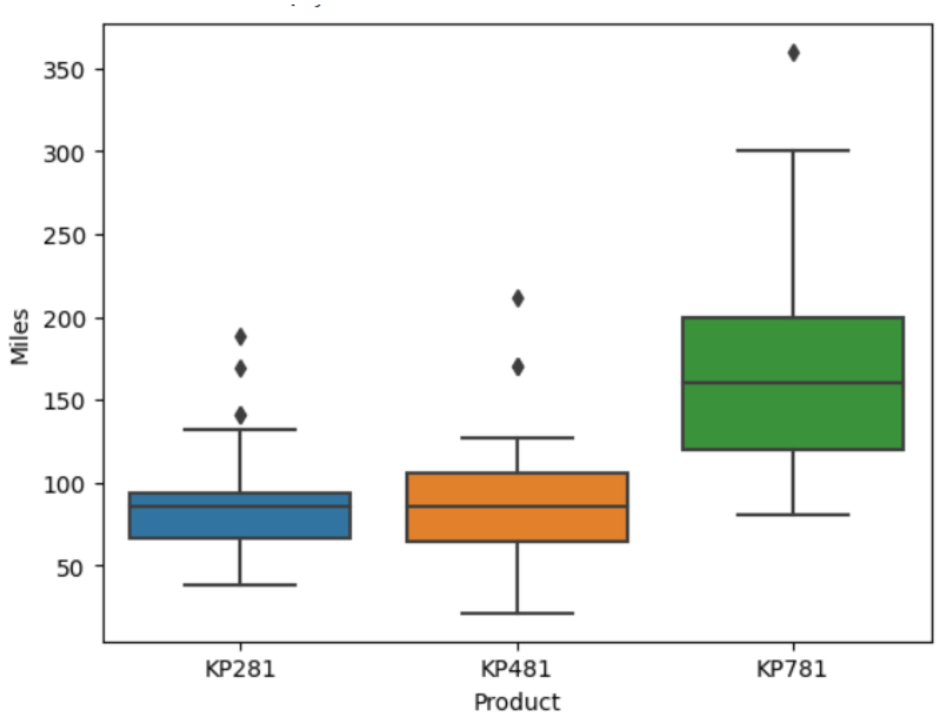

```
sns.boxplot(x = "Product" , y = "Income" , data = Aerofit)
```



Business Insights :

1. There is very less customers with salary less than 4000.
2. The median salary of KP781 customers is higher than maximum salary of KP281 and KP481 customers.

```
sns.boxplot(x = "Product" , y = "Miles" , data = Aerofit)
```




```
KP281_BoxPlot_Miles_Max = min(KP281_Product.Miles.max() ,
(KP281_Product.Miles.quantile(0.75)) + (1.5 *
((KP281_Product.Miles.quantile(0.75)) -
KP281_Product.Miles.quantile(0.25))))
KP481_BoxPlot_Miles_Max = min(KP481_Product.Miles.max() ,
(KP481_Product.Miles.quantile(0.75)) + (1.5 *
((KP481_Product.Miles.quantile(0.75)) -
KP481_Product.Miles.quantile(0.25))))
KP781_BoxPlot_Miles_Max = min(KP781_Product.Miles.max() ,
(KP781_Product.Miles.quantile(0.75)) + (1.5 *
((KP781_Product.Miles.quantile(0.75)) -
KP781_Product.Miles.quantile(0.25))))

KP281_outliar_Miles_count =
KP281_Product.Miles[KP281_Product.Miles >
KP281_BoxPlot_Miles_Max].count()
KP481_outliar_Miles_count =
KP481_Product.Miles[KP481_Product.Miles >
KP481_BoxPlot_Miles_Max].count()
KP781_outliar_Miles_count =
KP781_Product.Miles[KP781_Product.Miles >
KP781_BoxPlot_Miles_Max].count()

print(f"KP281_BoxPlot_Miles_Max : {KP281_BoxPlot_Miles_Max}")
print(f"KP481_BoxPlot_Miles_Max : {KP481_BoxPlot_Miles_Max}")
print(f"KP781_BoxPlot_Miles_Max : {KP781_BoxPlot_Miles_Max}")
print()
```

```
print(f"KP281_outliar_Miles_count :
{KP281_outliar_Miles_count}")
print(f"KP481_outliar_Miles_count :
{KP481_outliar_Miles_count}")
print(f"KP781_outliar_Miles_count :
{KP781_outliar_Miles_count}")
```

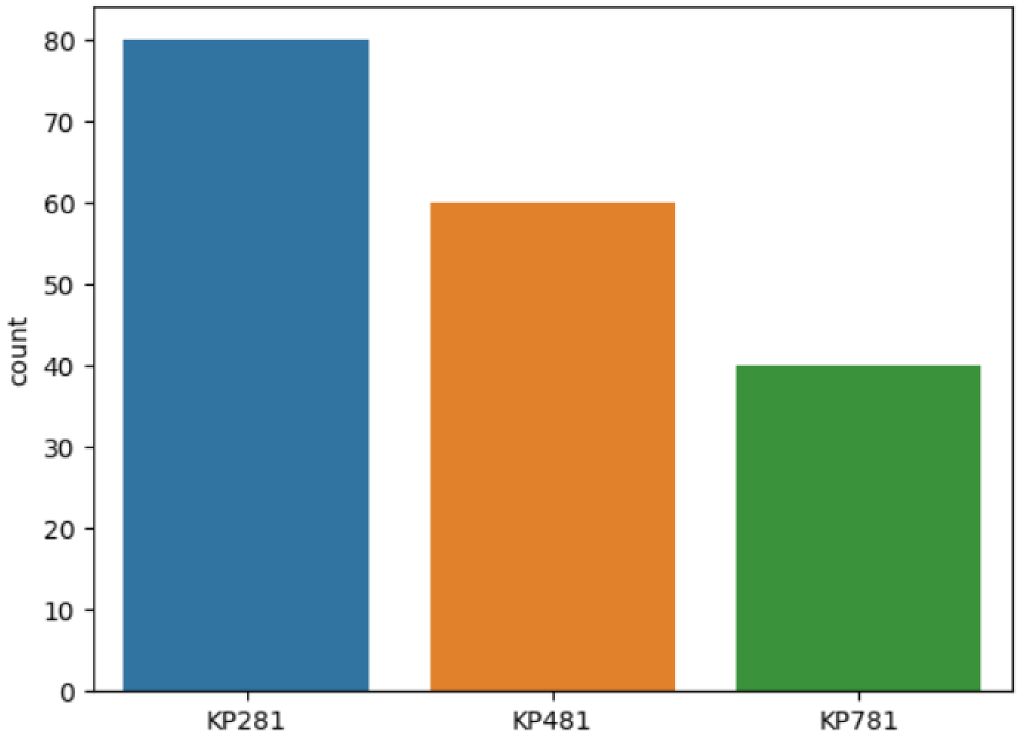
```
KP281_BoxPlot_Miles_Max : 136.0
KP481_BoxPlot_Miles_Max : 169.0
KP781_BoxPlot_Miles_Max : 320.0

KP281_outliar_Miles_count : 4
KP481_outliar_Miles_count : 3
KP781_outliar_Miles_count : 1
```

Business Insights :

1. Median salary of KP281 is same as median salary of KP481. Median salary of KP781 is higher than KP281 and KP481.
2. KP281 has 4 salary which is laying outside of the boxplot while KP481 has 3 salaries of customers and KP781 has only 1 outlier salary.

3. Check if features like marital status, age have any effect on the product purchased (using countplot, histplots, boxplots etc).

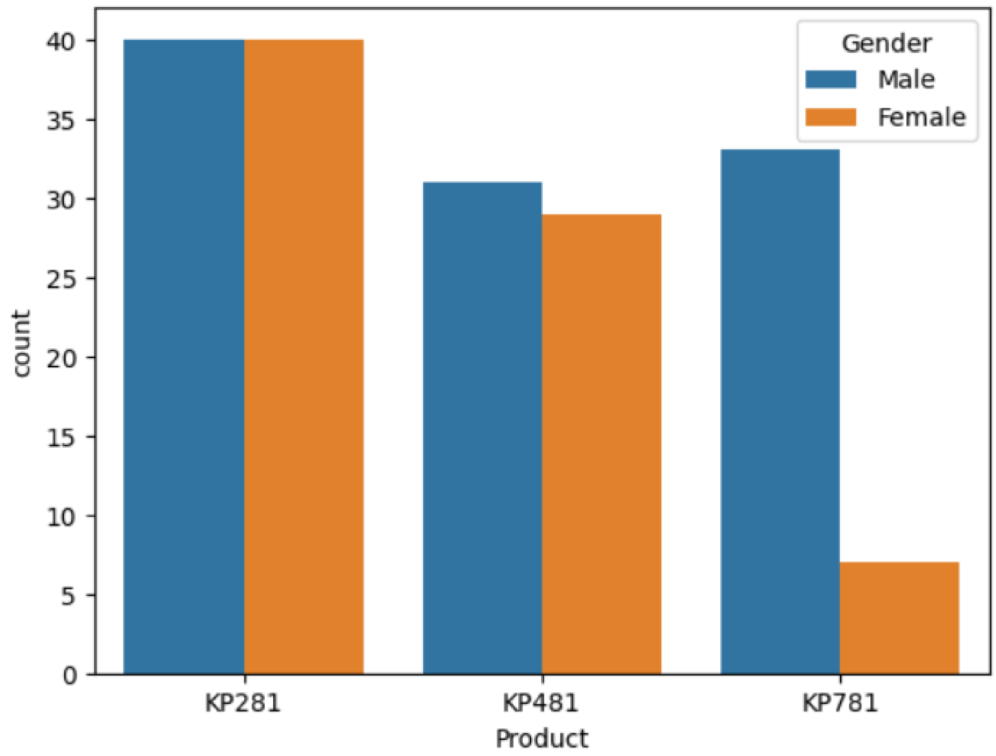
```
sns.countplot(x = Aerofit.Product , data = Aerofit)
```



Business Insights :

1. KP281 is the most sold product out of the given products one most important reason for this is because KP281 is cheaper than KP481 and KP781.

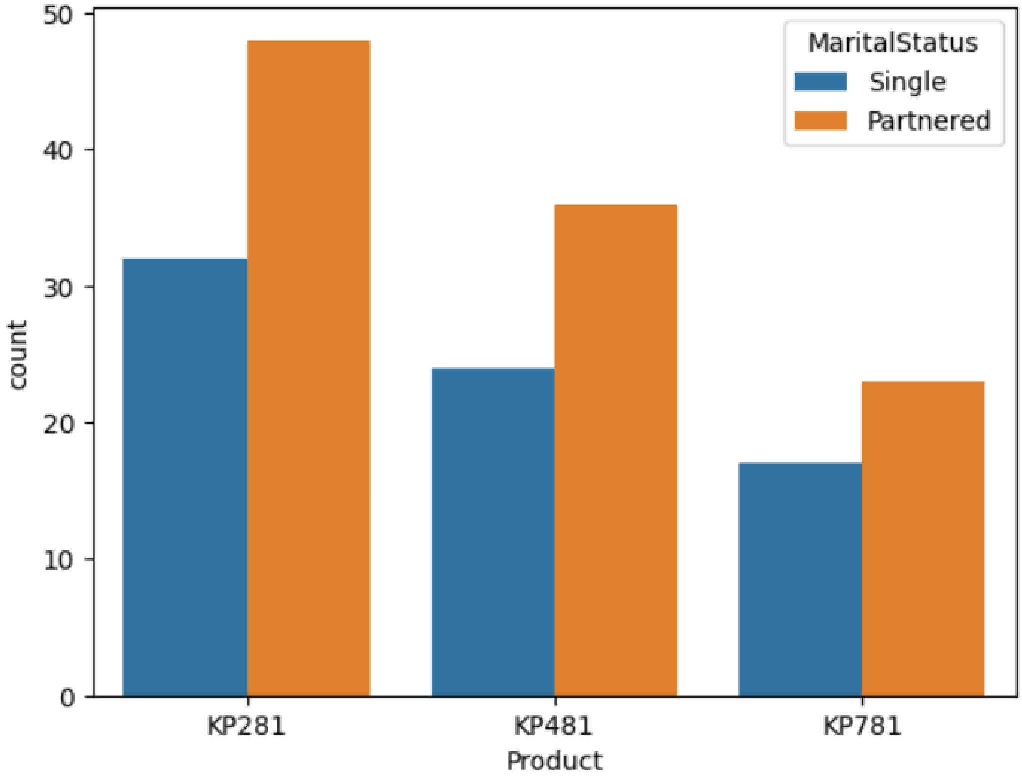
```
sns.countplot(x = Aerofit.Product , data = Aerofit , hue =
"Gender")
```



Business Insights :

1. Both male and female have purchased KP281 equally.
2. More males have purchased KP481 than females.
3. KP781 is mainly prefer by males than female as there is very less purchase of KP781 by females than males.

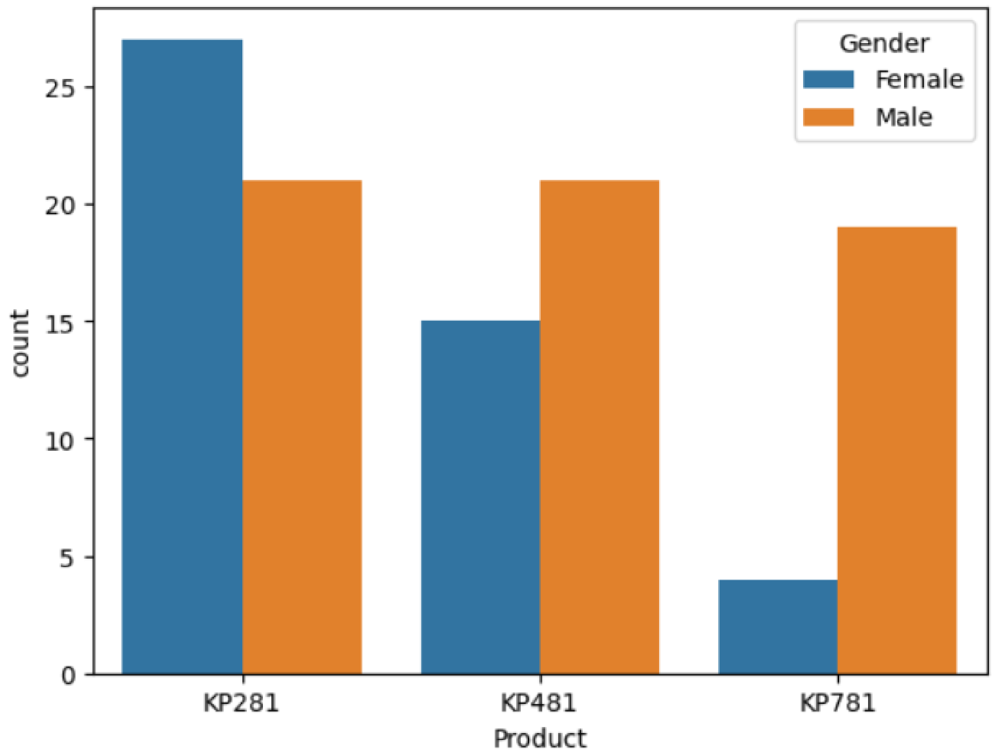
```
sns.countplot(x = Aerofit.Product , data = Aerofit , hue =
"MaritalStatus")
```



Business Insights :

1. As we can infer from the above plot partnered people have more purchase products than single people.

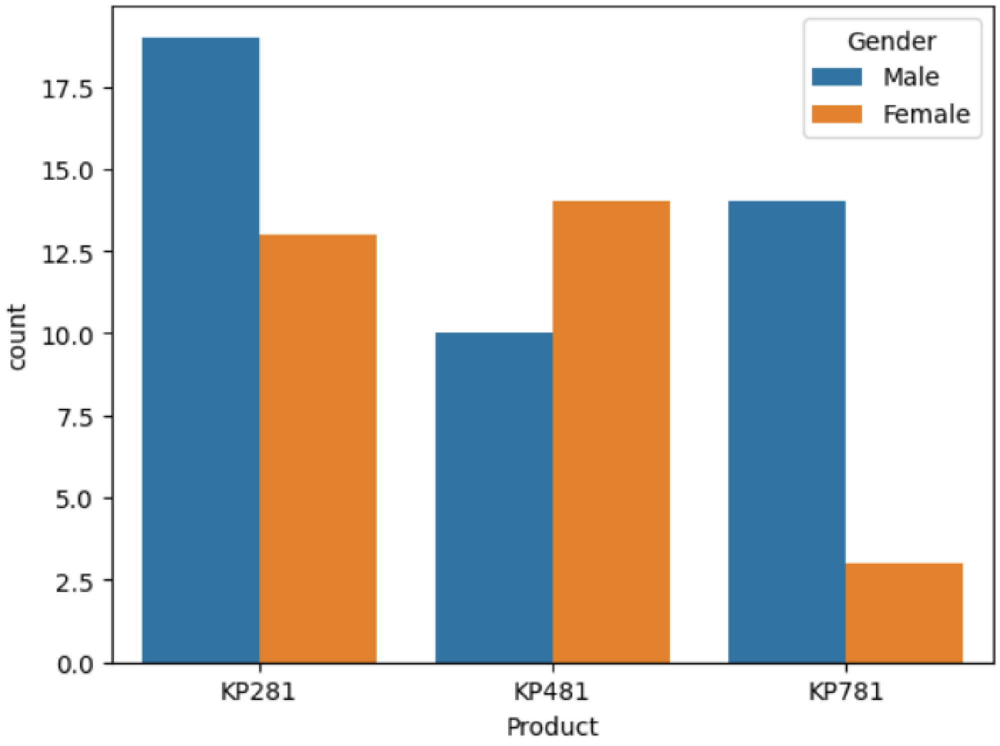
```
Married_data = Aerofit[Aerofit.MaritalStatus == "Partnered"]
Single_data = Aerofit[Aerofit.MaritalStatus == "Single"]


sns.countplot(x = Married_data.Product , data = Married_data,
hue = "Gender")
```



Business Insights :

1. Female married people have purchased more KP281 products than male married people while KP481 is purchased by less female married than male married people.
2. Mainly male married people have purchased KP781 treadmill.

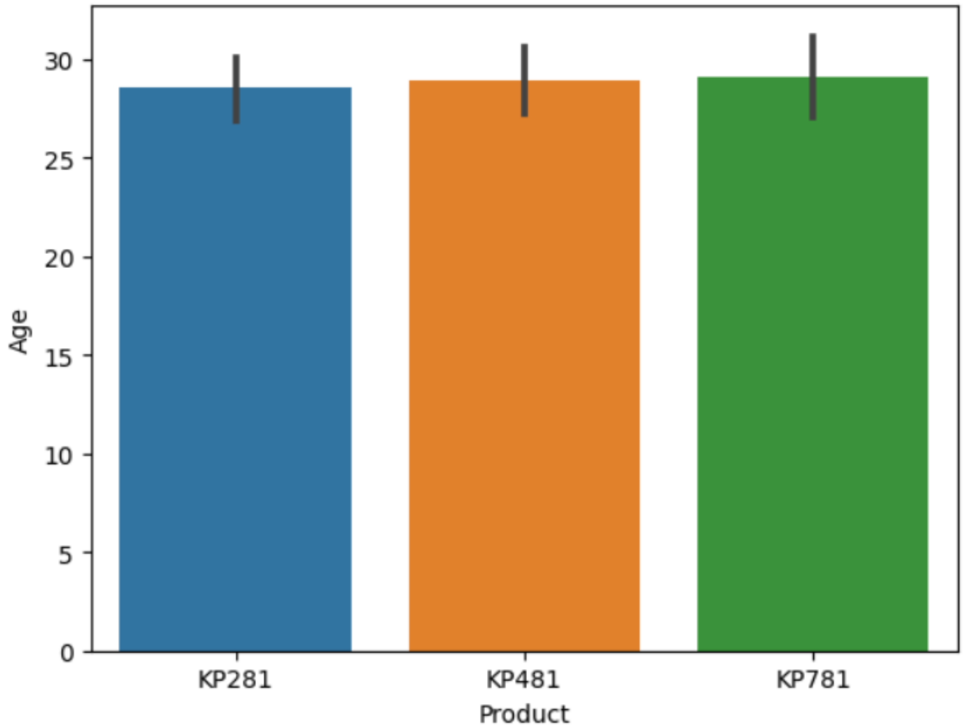
```
sns.countplot(x = Single_data.Product , data = Single_data,
hue = "Gender")
```



Business Insights :

1. Female single people have purchased more KP281 products than male married people while KP481 is purchased by less single female than single male people.
2. Mainly single male people have purchased KP781 treadmill.

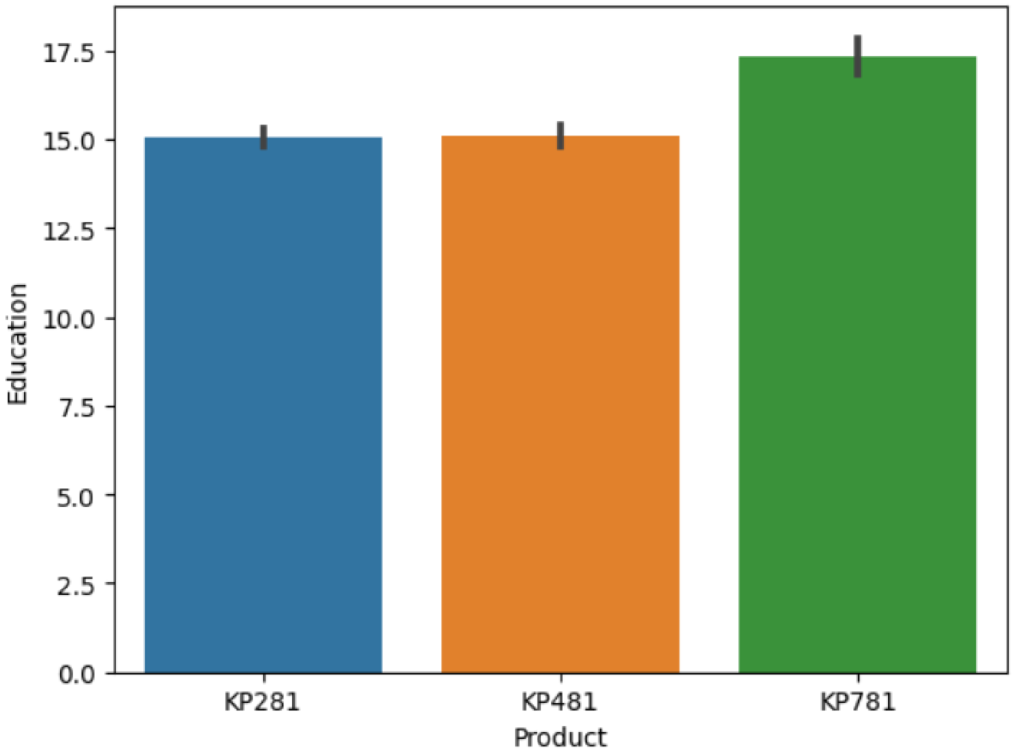
```
sns.barplot(x = "Product" , y = "Age" , data = Aerofit)
```



Business Insights :

1. Mean age for all the three products are similar.

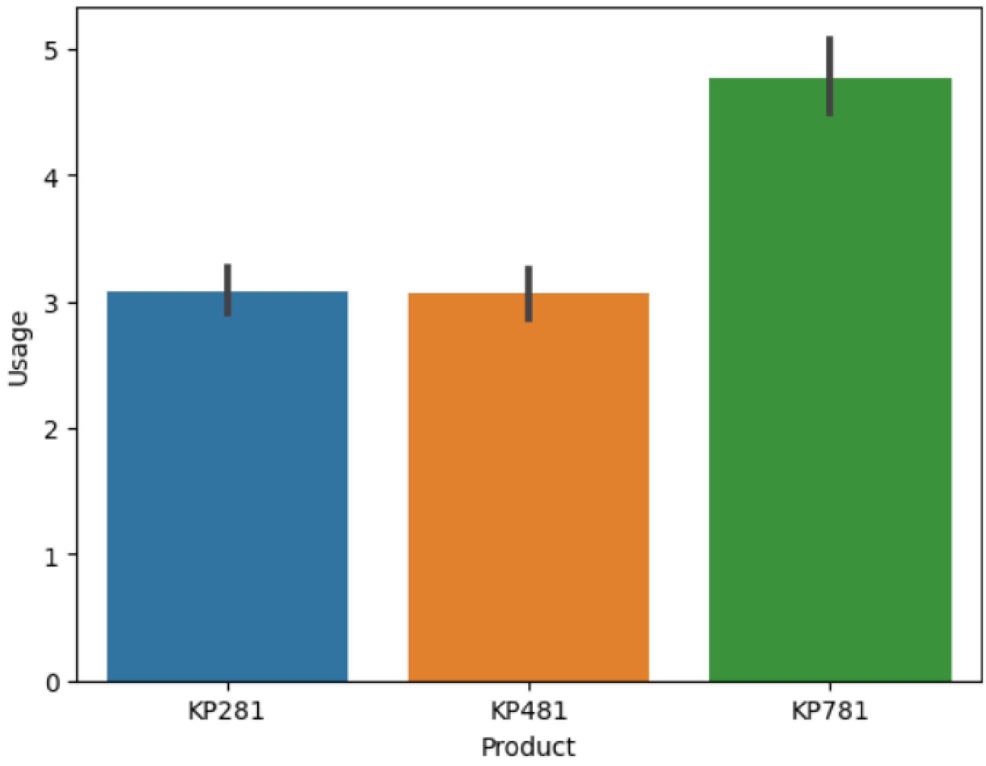
```
sns.barplot(x = "Product" , y = "Education" , data = Aerofit)
```



Business Insights :

1. Mean education for KP281 and KP481 treadmill is same while KP781 has higher mean education.
2. We can infer from the above plot that education for KP281 and KP481 is same but KP781 treadmill is mainly purchased by more educated people.

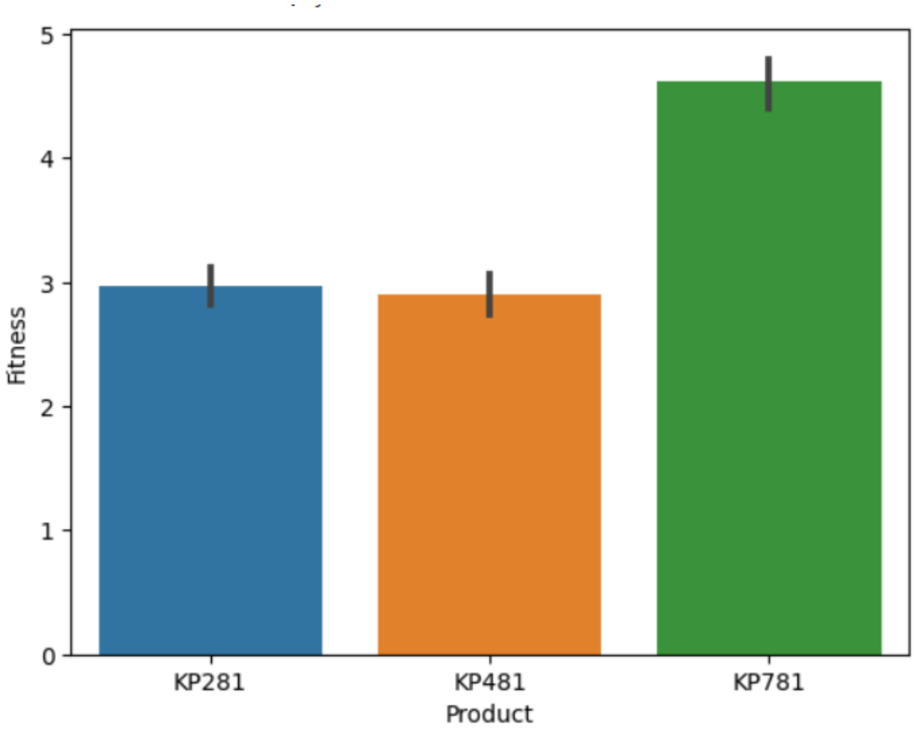
```
sns.barplot(x = "Product" , y = "Usage" , data = Aerofit)
```



Business Insights :

1. Customers who are willing to use the treadmill for 3 days a week are going for KP281 or KP481.
2. Customers who wants to use the treadmill more around 4 or 5 days a week are preferring KP781 as it has more advanced features.

```
sns.barplot(x = "Product" , y = "Fitness" , data = Aerofit)
```

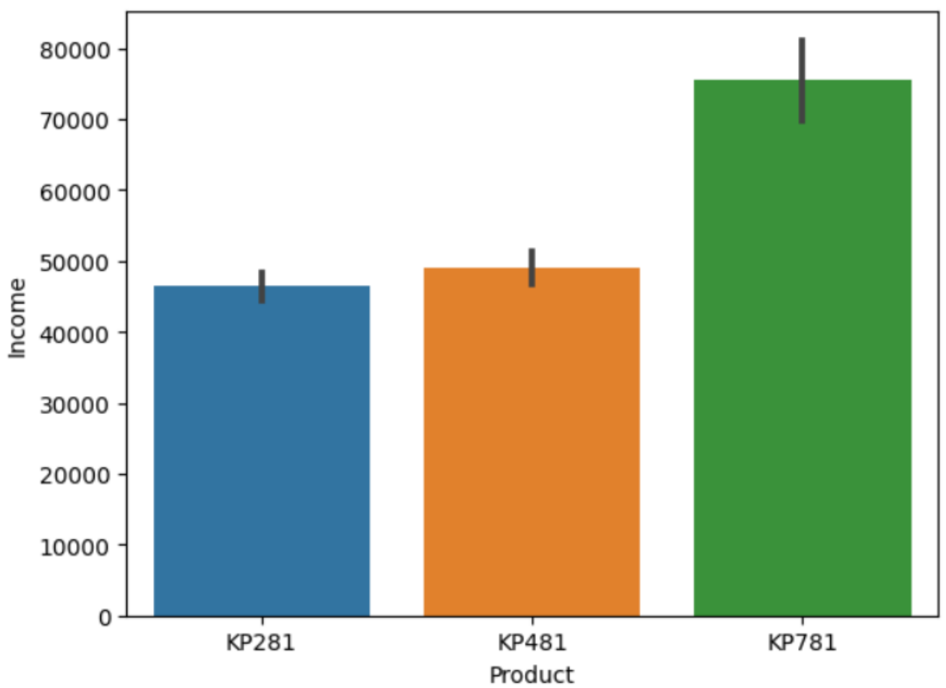


Business Insights :

1. Customers who have rated their fitness around 3 out of 5 are mainly buying KP281 or KP481.

2. Customers who are more fit are mainly buying KP781 treadmill.

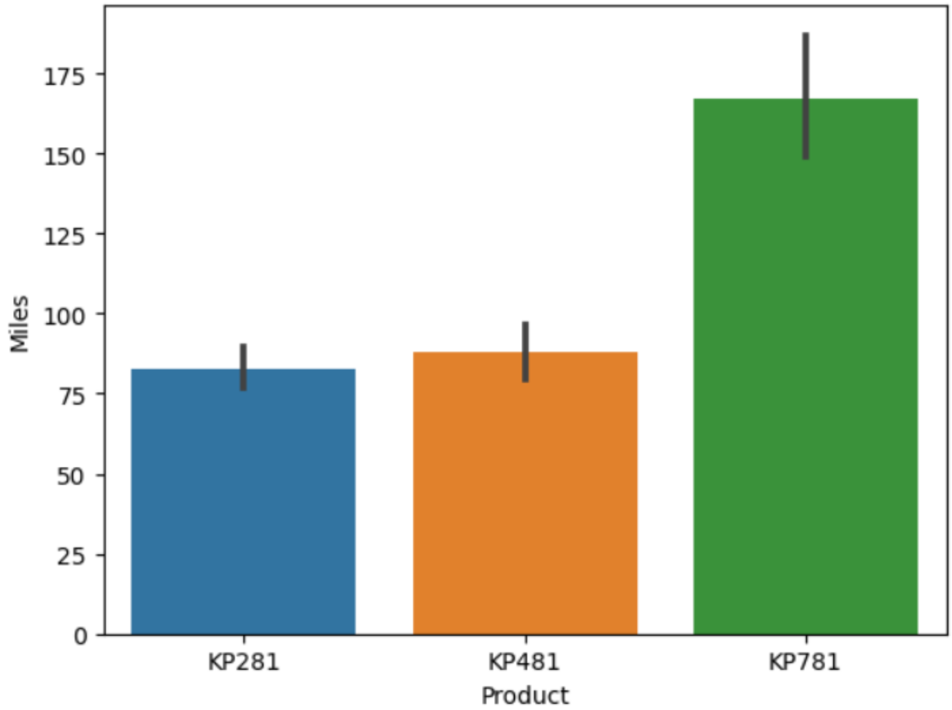
```
sns.barplot(x = "Product" , y = "Income" , data = Aerofit)
```



Business Insights :

1. Customers who's salaries are between 45000 and 50000 are buying KP281 or KP481 treadmill.
2. Customers with high salary are preferring KP781 treadmill.

```
sns.barplot(x = "Product" , y = "Income" , data = Aerofit)
```



Business Insights :

1. Customers who are expecting to walk or run 80 to 90 miles per week are buying KP281 or KP481 treadmill.
2. Customers who are expecting to walk or run higher miles are purchasing KP781 treadmill.

4. Representing the marginal probability like - what percent of customers have purchased KP281, KP481, or KP781 in a table *(can use pandas.crosstab here).*

```
Product_count = Aerofit["Product"].value_counts()
Product_count_per = np.round((Product_count / len(Aerofit)) * 100 , 2)
print(Product_count)
print(Product_count_per)
```

```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
KP281    44.44
KP481    33.33
KP781    22.22
Name: Product, dtype: float64
```

```python
GenderWise_Probability = pd.crosstab([Aerofit.Product] ,
[Aerofit.Gender] , normalize = True)
print(GenderWise_Probability)
```

```
Gender      Female      Male
Product
KP281     0.222222  0.222222
KP481     0.161111  0.172222
KP781     0.038889  0.183333
```

```python
MaritalStatus_Probability = pd.crosstab([Aerofit.Product] ,
[Aerofit.MaritalStatus] , normalize = True)
print(MaritalStatus_Probability)
```

```
MaritalStatus  Partnered    Single
Product
KP281           0.266667  0.177778
KP481           0.200000  0.133333
KP781           0.127778  0.094444
```

```python
GenderWise_MaritalStatus_Probability =
pd.crosstab([Aerofit.Product , Aerofit.Gender] ,
[Aerofit.MaritalStatus] , normalize = True)
print(GenderWise_MaritalStatus_Probability)
```

```
MaritalStatus   Partnered    Single
Product Gender
KP281   Female   0.150000  0.072222
        Male     0.116667  0.105556
KP481   Female   0.083333  0.077778
        Male     0.116667  0.055556
KP781   Female   0.022222  0.016667
        Male     0.105556  0.077778
```

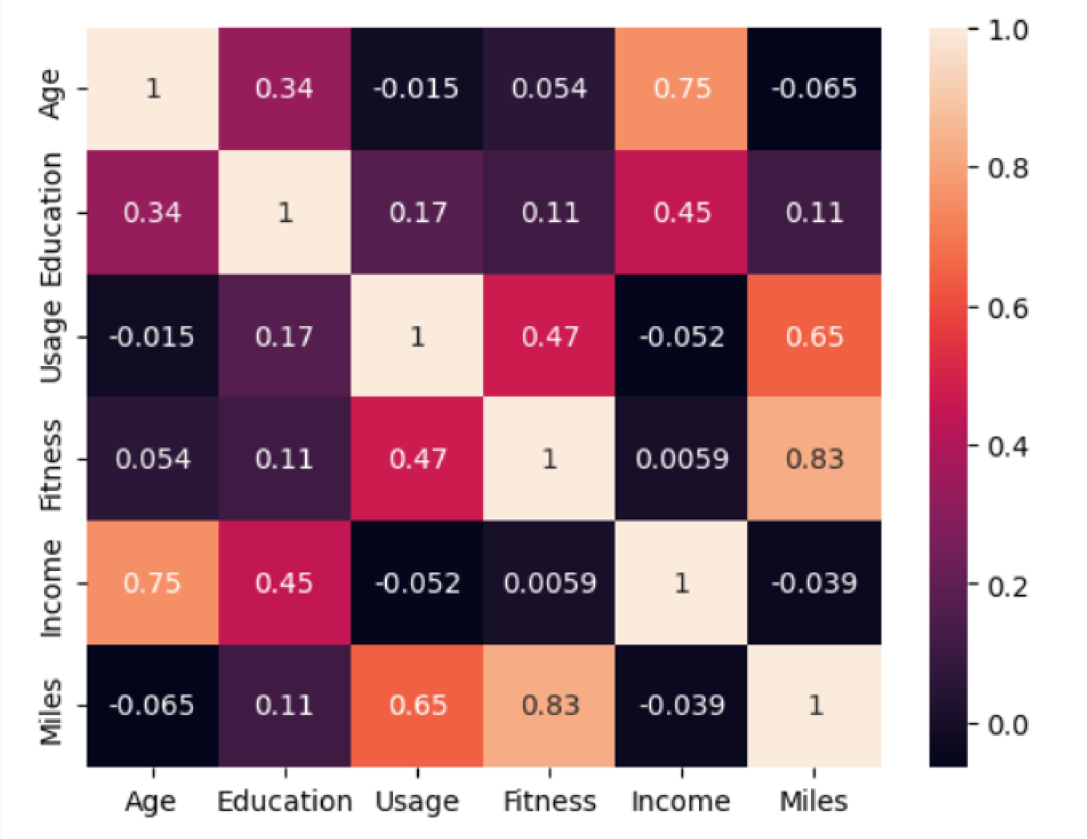
Business Insights :

1. **We can infer different combinations of purchasing different products from the above outputs.**
2. **KP281 is purchased maximum times with 44.44% of probability.**
3. **All the three products are mostly purchased by partnered customers than single.**
4. **In partnered customers females have purchased KP281 with 15% probability which is highest.**
5. **Male partnered have almost equal purchases of all the three products.**

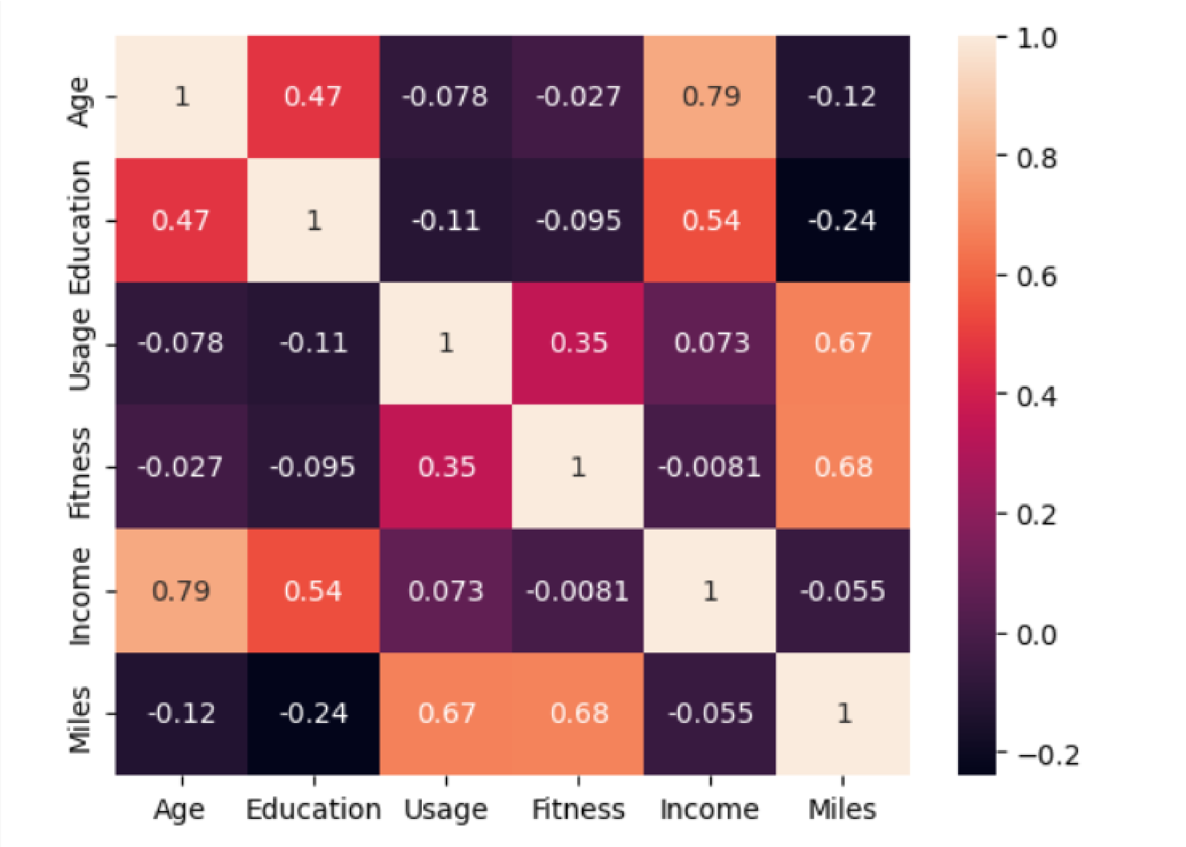
5. **Check correlation among different factors using heat maps or pair plots.**

```
KP281_heatmap = KP281_Product.corr()
KP481_heatmap = KP481_Product.corr()
KP781_heatmap = KP781_Product.corr()
```

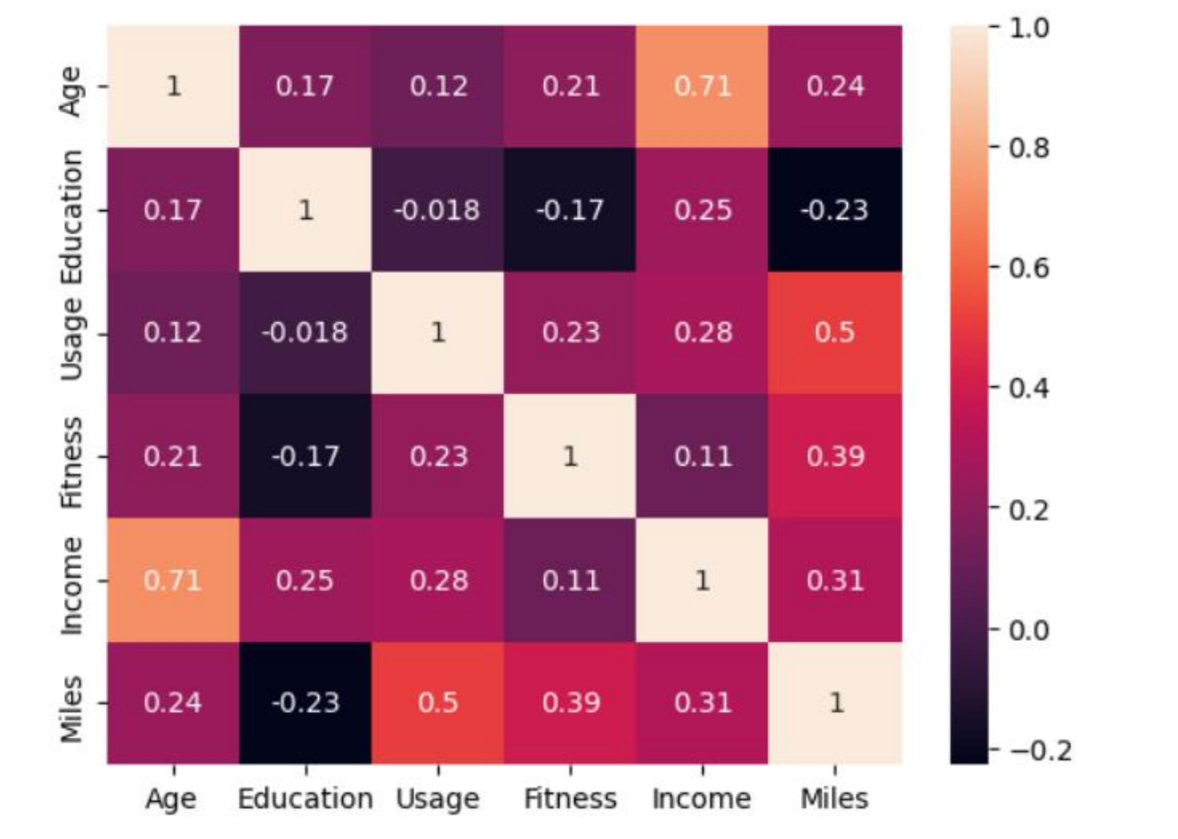
```
sns.heatmap(data = KP281_heatmap , annot = True )
```




```
sns.heatmap(data = KP481_heatmap , annot = True)
```




```
sns.heatmap(data = KP781_heatmap, annot = True)
```



Business Insights :

1. In above heatmaps we can see correlation of different variables with each other for different products.

2. Age and income is highly correlated for all the products and Age and usage or miles are negatively correlated.

3. Fitness is positively correlated with usage and miles.

4. Income is positively correlated with age and education.

6. Marginal and Conditional probability.

6.1 Marginal probability

```
Product_count = Aerofit["Product"].value_counts()
Product_count_per = np.round((Product_count / len(Aerofit)) *
100 , 2)
print(Product_count)
print(Product_count_per)
```

```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
KP281    44.44
KP481    33.33
KP781    22.22
Name: Product, dtype: float64
```

6.2 Conditional Probability

```
GenderWise_Probability = pd.crosstab([Aerofit.Product] ,
[Aerofit.Gender] , normalize = "index")
print(GenderWise_Probability)
```

```
Gender      Female      Male
Product
KP281     0.222222  0.222222
KP481     0.161111  0.172222
KP781     0.038889  0.183333
```

```
MaritalStatus_Probability = pd.crosstab([Aerofit.Product] ,
[Aerofit.MaritalStatus] , normalize = "index")
print(MaritalStatus_Probability)
```

```
MaritalStatus  Partnered    Single
Product
KP281           0.266667  0.177778
KP481           0.200000  0.133333
KP781           0.127778  0.094444
```

```
GenderWise_MaritalStatus_Probability =
pd.crosstab([Aerofit.Product , Aerofit.Gender] ,
[Aerofit.MaritalStatus] , normalize = "index")
print(GenderWise_MaritalStatus_Probability)
```

```
MaritalStatus   Partnered    Single
Product Gender
KP281   Female   0.150000  0.072222
        Male     0.116667  0.105556
KP481   Female   0.083333  0.077778
        Male     0.116667  0.055556
KP781   Female   0.022222  0.016667
        Male     0.105556  0.077778
```

Recommendations:

1. KP781 should be marked as premium product and marketing it to high income groups and educational over 20 years market segments could result in more sales.

2. Aerofit should conduct a market research to determine if it can attract customers with income under 40,000 to expand it's customer base.

3. The KP781 is a premium model, so it ideally suited for the sporty people who have a high average weekly mileage.