Aerofit Exploratory data analysis

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.

Importing libraries

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

Uploading data

```
data= pd.read_csv('/Users/divyabansal/Downloads/aerofit_treadmill.csv')
```

```
data.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

Basic Observations

```
data.shape
```

```
(180, 9)
```

```
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
```

The info shows there are no null/missing values in the data

```
data.columns
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

```
data.describe()
```

	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

```
data.describe(include ='object') # description of string columns
```

	Product	Gender	MaritalStatus
count	180	180	180
unique	3	2	2
top	KP281	Male	Partnered
freq	80	104	107

Insights -

1. Men have bought the most Treadmills

2. The most bought Treadmill is KP281

```
data.dtypes
```

```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
```

```
Miles            int64
dtype: object
```

Correcting the data type of categorical columns

```
category_columns= ['Product','Gender','MaritalStatus']
data[category_columns]= data[category_columns].astype('category')
```

```
data.dtypes
```

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

Non-Graphical analysis

```
data.nunique()
```

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

```
for col in category_columns:
    value_count= data[col].value_counts()
    print(value_count)
    print('-------------------')
```

```
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
-------------------
Gender
Male      104
Female     76
Name: count, dtype: int64
-------------------
MaritalStatus
Partnered    107
Single        73
Name: count, dtype: int64
-------------------
```

Missing value and Outlier detection

```
data.isna().sum()
```

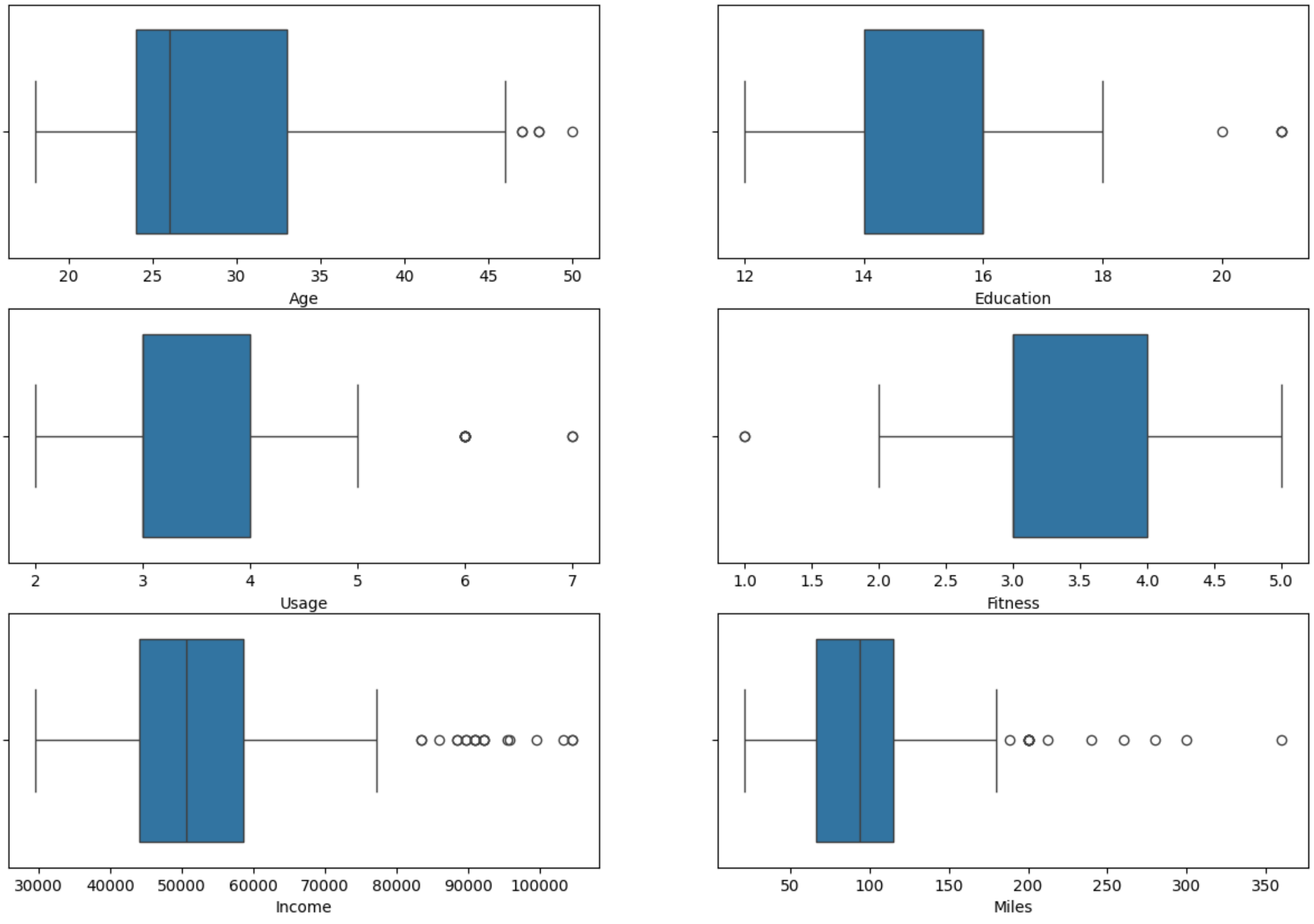
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

Insights- no missing values found in the data

```
# outlier detection using boxplot for all the numerical columns in the data
```

```
fig, ax = plt.subplots(3, 2, figsize = (15, 10))
sns.boxplot(x='Age', data=data,ax= ax[0,0])
sns.boxplot(x='Education', data=data, ax=ax[0,1])
sns.boxplot(x='Usage', data=data,ax=ax[1,0])
sns.boxplot(x='Fitness', data=data,ax=ax[1,1])
sns.boxplot(x='Income', data=data,ax=ax[2,0])
sns.boxplot(x='Miles', data=data,ax=ax[2,1])
```

```
<Axes: xlabel='Miles'>
```



Insights-

outliers detected-

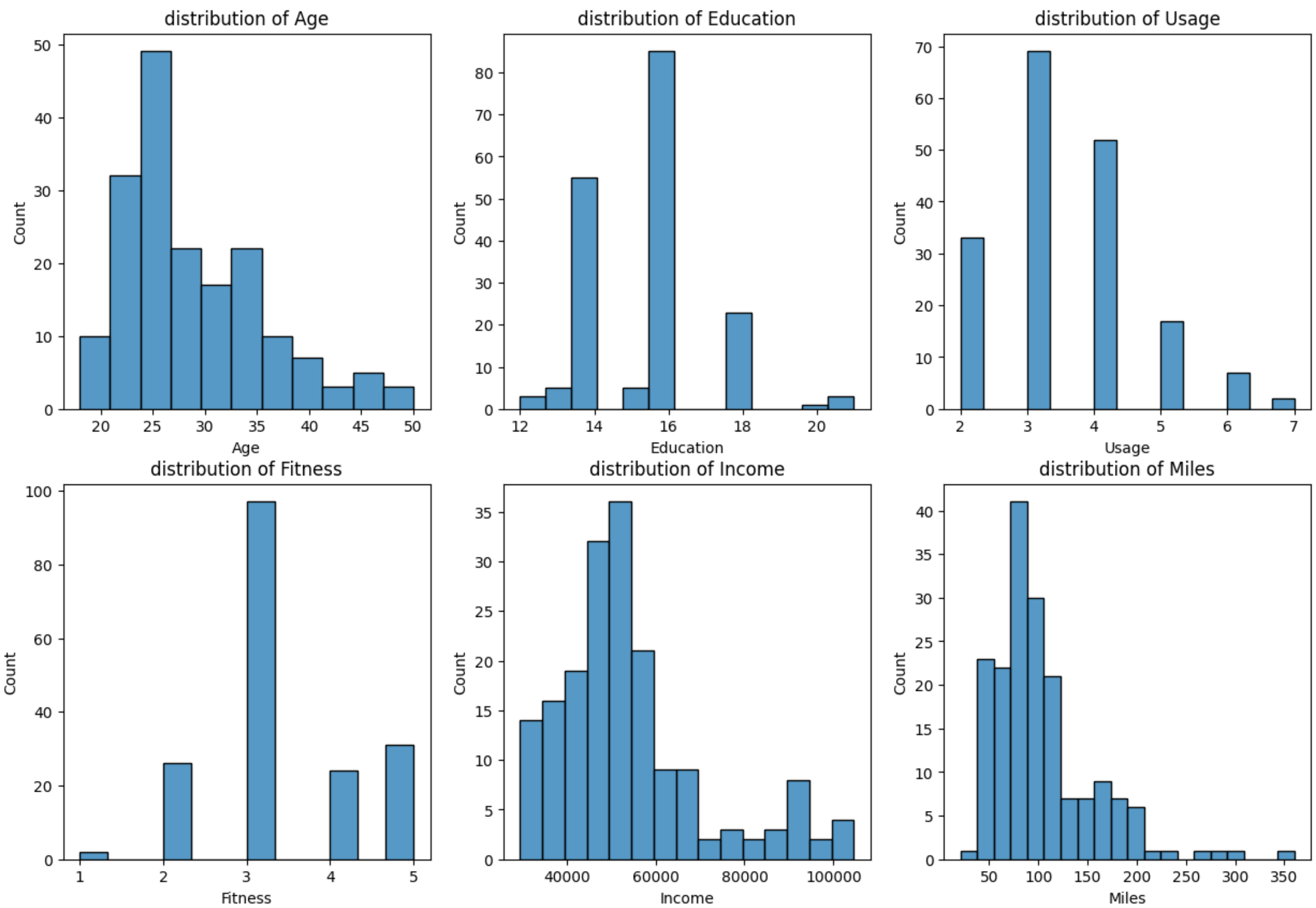
Age, Education, Usage and fitness columns have no considerable outliers while 'Income' and 'Miles' columns have some outliers

Visual Analysis- Univariate and Bivariate

```
df=data
df.head()
```

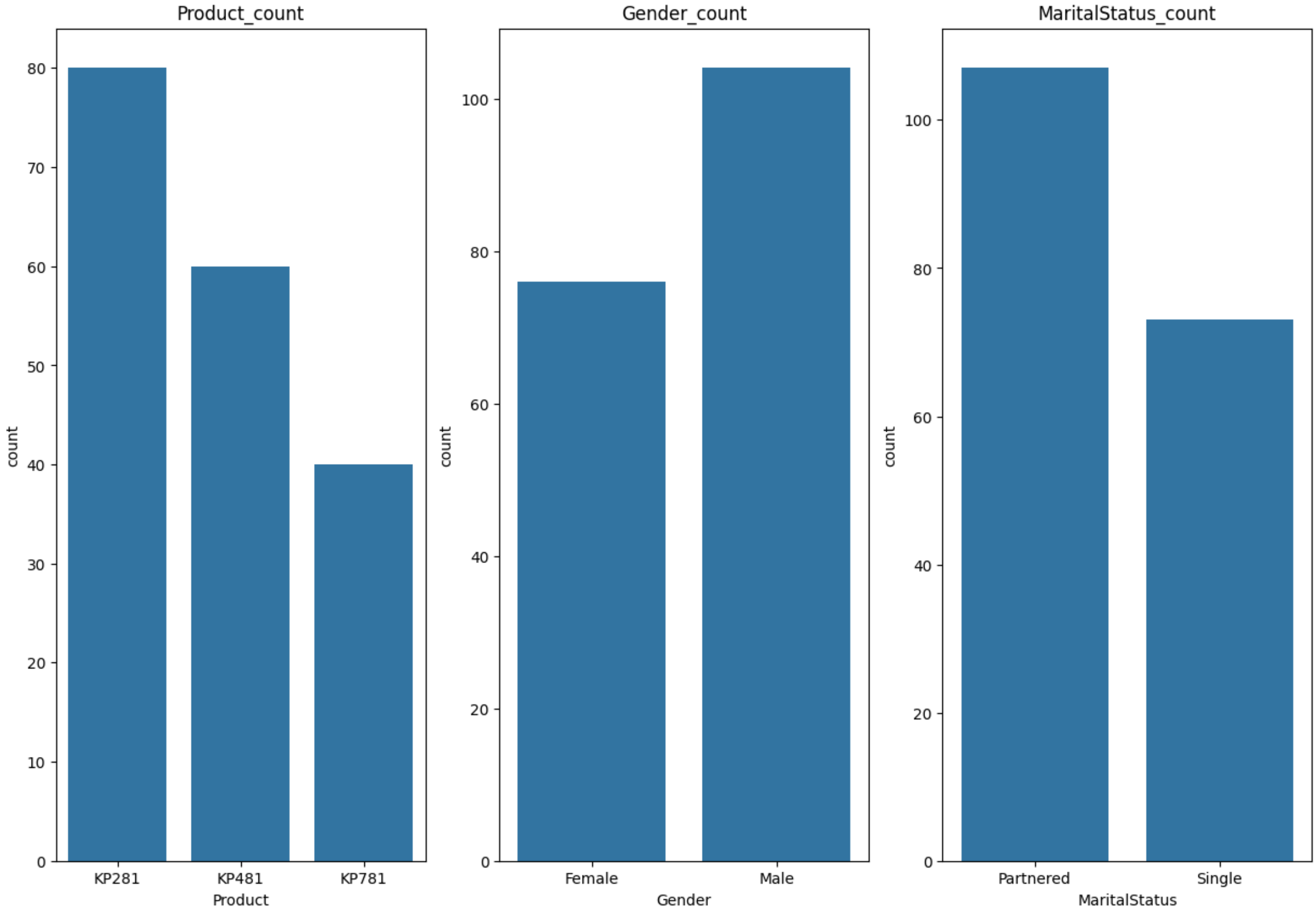
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

```
# Histplot of Numerical/Continuous attributes
plt.figure(figsize=(15,10))
num_col=['Age','Education','Usage','Fitness','Income','Miles']
for i,col in enumerate(num_col, 1):
    plt.subplot(2, 3, i)
    sns.histplot(df[col])
    plt.title(f'distribution of {col}')


plt.show()
```



```
# countplot of categorical attributes

fig, ax=plt.subplots(1,3, figsize=(15,10))
sns.countplot(x='Product',data=df, ax= ax[0])
ax[0].set_title('Product_count')
sns.countplot(x='Gender',data=df, ax= ax[1])
ax[1].set_title('Gender_count')
sns.countplot(x='MaritalStatus',data=df, ax= ax[2])
ax[2].set_title('MaritalStatus_count')
```

```
Text(0.5, 1.0, 'MaritalStatus_count')
```



insights-

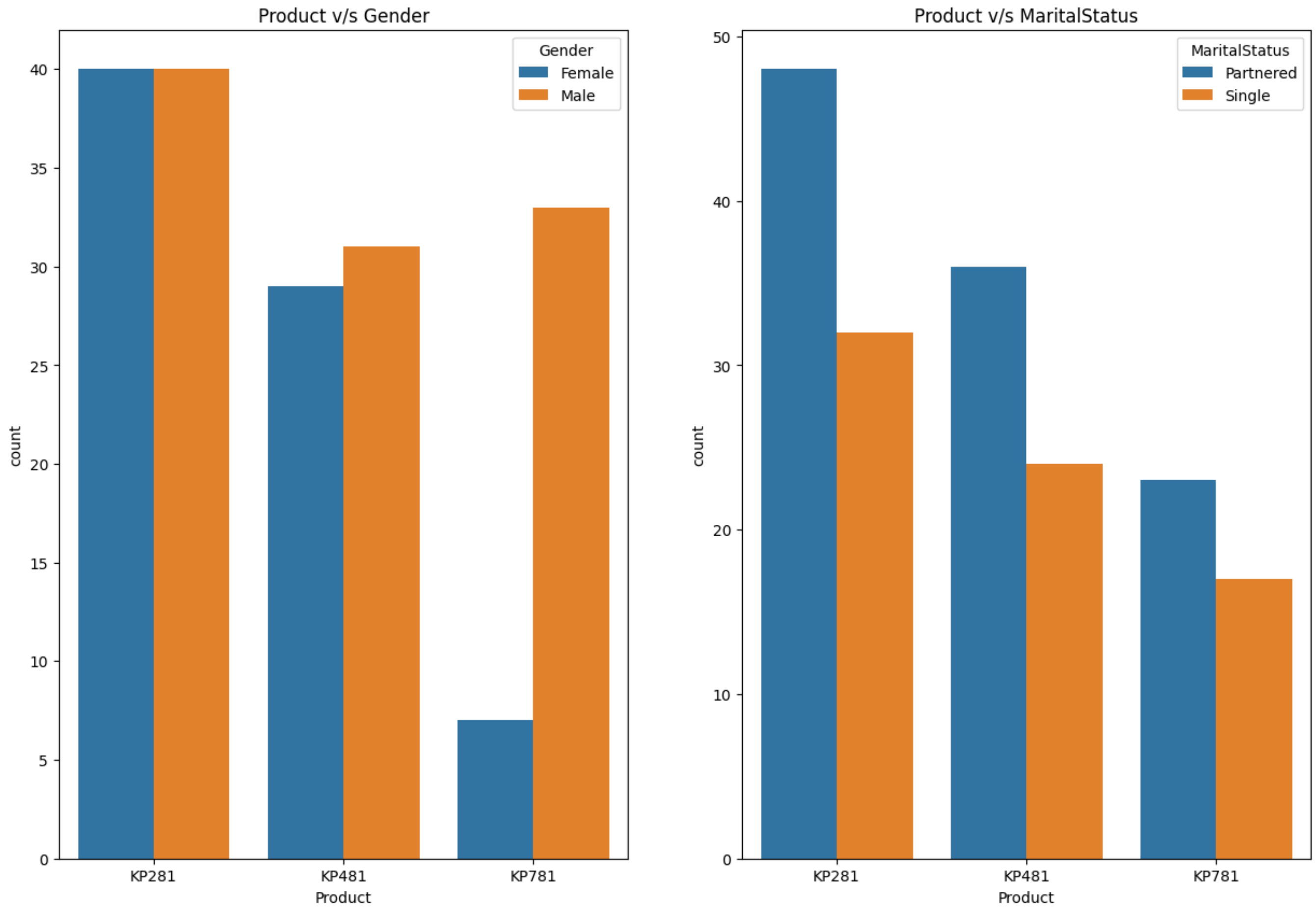
1. Most sold product= KP281

2. Most customer gender= Male

3. Most Customer MaritalStatus= Partnered

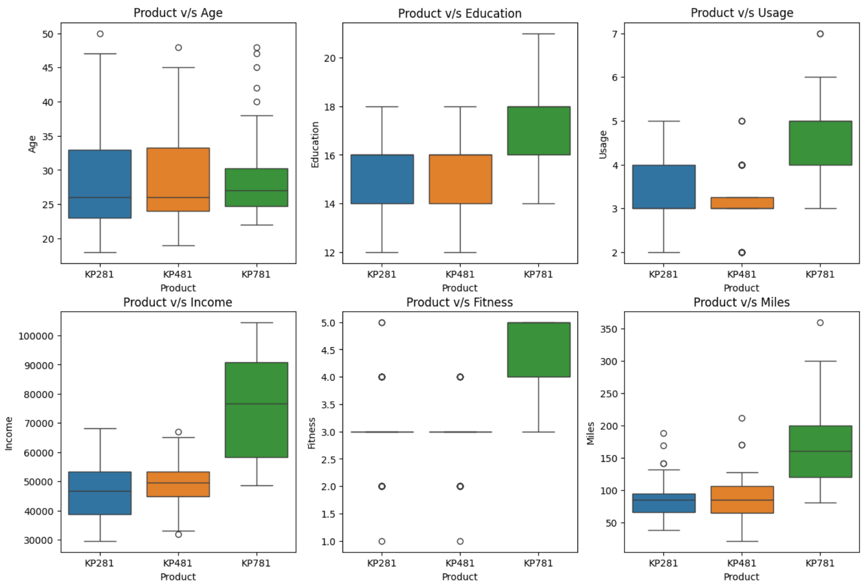
Bivariate analysis

```
# countplot of two categorical variables
fig, ax=plt.subplots(1,2,figsize=(15,10))
sns.countplot(x='Product',hue='Gender', data=df, ax= ax[0])
ax[0].set_title('Product v/s Gender')
sns.countplot(x='Product',hue='MaritalStatus',data=df, ax= ax[1])
ax[1].set_title('Product v/s MaritalStatus')

plt.show()
```




```
# Insights-
# 1. Partnered customers have bought all the products more compared to the single customers
# 2. KP281 is equally bought by Male and Female customers
# 3. KP481 is bought by Male customers more compared to female customers
# 4. KP781 is bought the most by Male customers
```

```
fig, ax=plt.subplots(2,3,figsize=(15,10))

sns.boxplot(x='Product',y='Age', data=df, ax= ax[0,0],hue='Product', legend=False)
ax[0,0].set_title('Product v/s Age')

sns.boxplot(x='Product',y='Education', data=df, ax= ax[0,1],hue='Product', legend=False)
ax[0,1].set_title('Product v/s Education')

sns.boxplot(x='Product',y='Usage', data=df, ax= ax[0,2],hue='Product', legend=False)
ax[0,2].set_title('Product v/s Usage')

sns.boxplot(x='Product',y='Income', data=df, ax= ax[1,0],hue='Product', legend=False)
ax[1,0].set_title('Product v/s Income')

sns.boxplot(x='Product',y='Fitness', data=df, ax= ax[1,1],hue='Product', legend=False)
ax[1,1].set_title('Product v/s Fitness')

sns.boxplot(x='Product',y='Miles', data=df, ax= ax[1,2],hue='Product', legend=False)
ax[1,2].set_title('Product v/s Miles')

plt.show()
```



Insights-

1. Product vs Age

Customers whose age lies between 25-30, are more likely to buy KP781 product while the others buy KP281 & KP481 equally

2. 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.

3. 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.

4. Product vs Income

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

5.Product vs Fitness

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

6.Product vs Miles

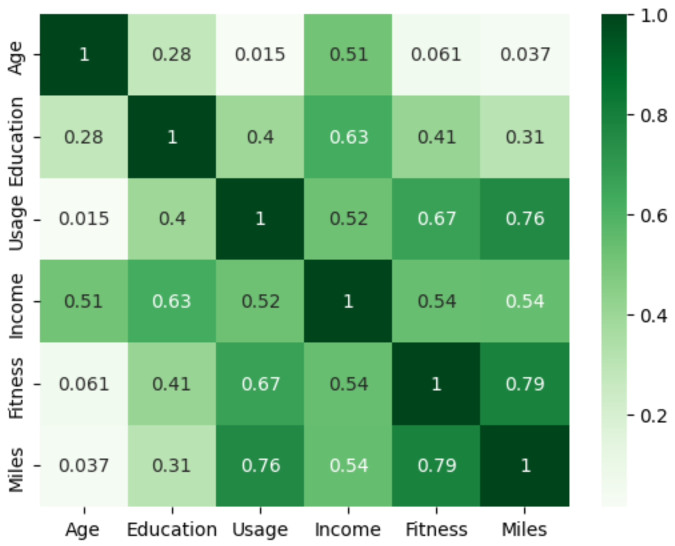
If the customer expects to walk/run greater than 120 Miles per week, it is more likely that the customer will buy KP781 product.

```
# Correlation analysis
num_data= df[['Age','Education','Usage','Income','Fitness','Miles']]
num_data.corr()
```

	Age	Education	Usage	Income	Fitness	Miles
Age	1.000000	0.280496	0.015064	0.513414	0.061105	0.036618
Education	0.280496	1.000000	0.395155	0.625827	0.410581	0.307284
Usage	0.015064	0.395155	1.000000	0.519537	0.668606	0.759130
Income	0.513414	0.625827	0.519537	1.000000	0.535005	0.543473
Fitness	0.061105	0.410581	0.668606	0.535005	1.000000	0.785702
Miles	0.036618	0.307284	0.759130	0.543473	0.785702	1.000000

```
sns.heatmap(num_data.corr(),cmap='Greens', annot=True)
```

<Axes: >



Marginal & Conditional Probablility

```
# Marginal Probability
```

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

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

```
# Conditional Probability of buying each product given Gender
```

```
def prob_product_given_gender(gender, print_marginal=False):

    df1 = pd.crosstab(index=df['Gender'], columns=[df['Product']])
    p_KP781 = df1['KP781'][gender] / df1.loc[gender].sum()
    p_KP481 = df1['KP481'][gender] / df1.loc[gender].sum()
    p_KP281 = df1['KP281'][gender] / df1.loc[gender].sum()

    if print_marginal:
        print(f"P(Male): {df1.loc['Male'].sum()/len(df):.2f}")
        print(f"P(Female): {df1.loc['Female'].sum()/len(df):.2f}\n")

    print(f"P(KP781/{gender}): {p_KP781:.2f}")
    print(f"P(KP481/{gender}): {p_KP481:.2f}")
    print(f"P(KP281/{gender}): {p_KP281:.2f}\n")

prob_product_given_gender('Male', True)
prob_product_given_gender('Female')
```

```
P(Male): 0.58
P(Female): 0.42

P(KP781/Male): 0.32
```

Recommendations-

After analyzing the given data, it can be said-

1. KP781 is the most expensive product among all the products. The Male customers who are fit and whose income is high,

are more likely to buy this product. so the marketing team should aim at this segment of customers

2. Customers whose age lies between 25-30, are more likely to buy KP781 products while the others buy KP281 & KP481 equally

3. Men are more likely to buy Treadmills compared to women. These products should be marketed to the female customer segment

by using some tempting offers, women fitness aware campaigns, etc

4. Married people are more likely to buy the products hence they should be targeted accordingly