
▼ Aerofit Business Case Analysis

This Aerofit Case analysis wants us to identify the type of customers for each type of treadmill the company provides. This would also help in recommendation for new customers willing to buy treadmill.



This case study includes descriptive and statistical analysis with appropriate tables and charts.

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

Firstly we need to import python libraries into work notebook to access different functions and methods to work with Aerofit data. These functions help in data manipulation, data analysis and helps to perform statistical and mathematical operations.

▼ Reading Data File

```
data = pd.read_csv('Aerofit_BusinessCase.csv')
data
```

	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	

▼ Shape of the Data

```
178    KP781    40    Male    21    Partnered    6    5    104581    120
```

```
data.shape
```

```
(180, 9)
```

The given dataset has 180 rows and 9 columns

▼ Information of Data

```
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 given dataset has 8 columns with no NULL values. the different datatypes are int and object.



▼ Data Top Signifies

```
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	

▼ Data Bottom Signifies



```
data.tail()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	
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	85500	100	

- The data statement above gives us the glimpse of small portion of dataset. Head() return top 5 rows and Tail() return bottom 5 row data.

▼ Statistical description of data

```
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	

▼ Statistical description of data with String/object format conclusion

```
data.describe(include = 'all')
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000



▼ Data cleaning and featurizing

```
data.isnull().any()
```

Product	False
Age	False
Gender	False

```
Education      False
MaritalStatus  False
Usage          False
Fitness        False
Income         False
Miles          False
dtype: bool
```

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

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

Observations :

- There are no missing values in the data.
- There are 3 unique products in the dataset.
- KP281 is the most frequent product.
- Minimum & Maximum age of the person is 18 & 50, mean is 28.79 and 75% of persons have age less than or equal to 33.
- Most of the people are having 16 years of education i.e., 75% of persons are having education ≤ 16 years.
- Out of 180 data points, 104's gender is Male and rest are the female.
- Standard deviation for Income & Miles is very high. These variables might have the outliers in it.

▼ Column Names

```
data.columns
```

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

▼ Display number of Treadmills in each product category

```
data['Product'].value_counts()
```

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

▼ Displaying the number of people who have rated the machines based on the 'Fitness' rating

```
data[['Product', 'Fitness']].value_counts()
```

Product	Fitness	
KP281	3	54
KP481	3	39
KP781	5	29
KP281	2	14
KP481	2	12
KP281	4	9
KP481	4	8
KP781	4	7
	3	4
KP281	5	2

```
      1      1
KP481  1      1
dtype: int64
```

▼ Fetching uniques attributes

```
data['Product'].unique()

array(['KP281', 'KP481', 'KP781'], dtype=object)
```

```
data['MaritalStatus'].unique()

array(['Single', 'Partnered'], dtype=object)
```

▼ Corelation between all features

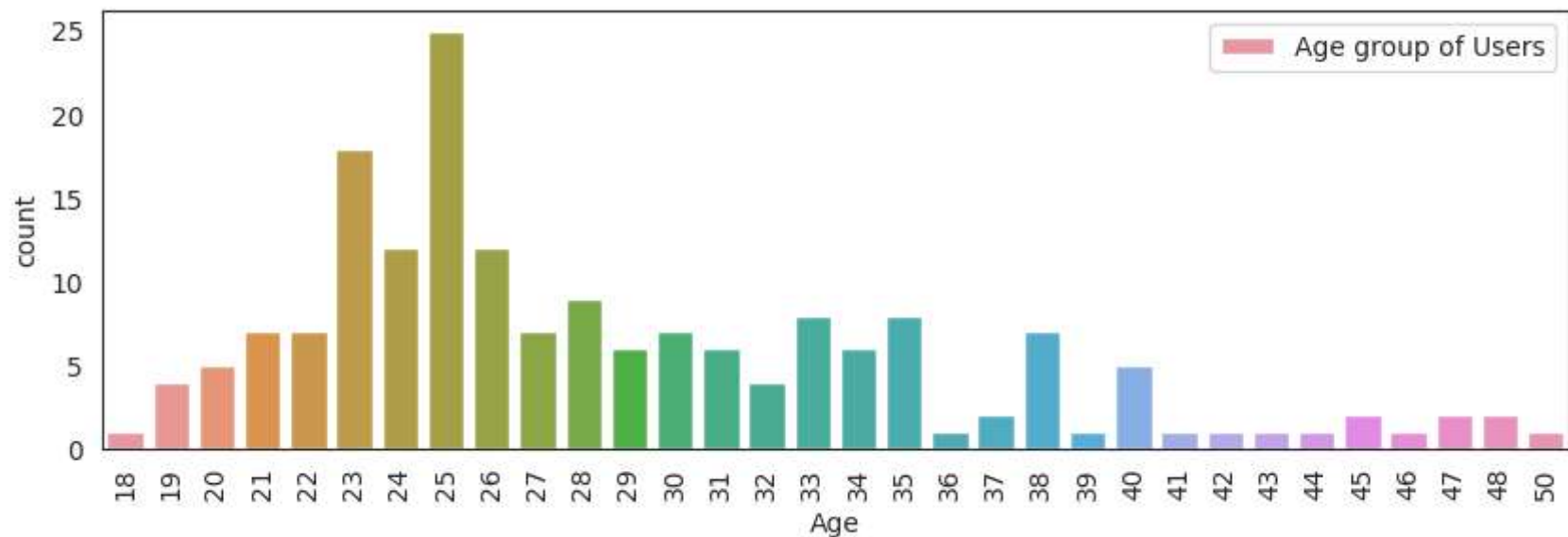
```
data.corr()
```


▼ Count of people using treadmill based on the Age category

Age 1.000000 0.200450 0.010004 0.001100 0.010414 0.000010



```
plt.figure(figsize=(10,3))
sns.countplot(data, x = "Age", label = 'Age group of Users')
plt.xticks(rotation = 90)
plt.legend()
plt.show()
```



```
data['Age'].value_counts()
```

25	25
23	18
24	12
26	12
28	9
35	8
33	8
30	7

38	7
21	7
22	7
27	7
31	6
34	6
29	6
20	5
40	5
32	4
19	4
48	2
37	2
45	2
47	2
46	1
50	1
18	1
44	1
43	1
41	1
39	1
36	1
42	1

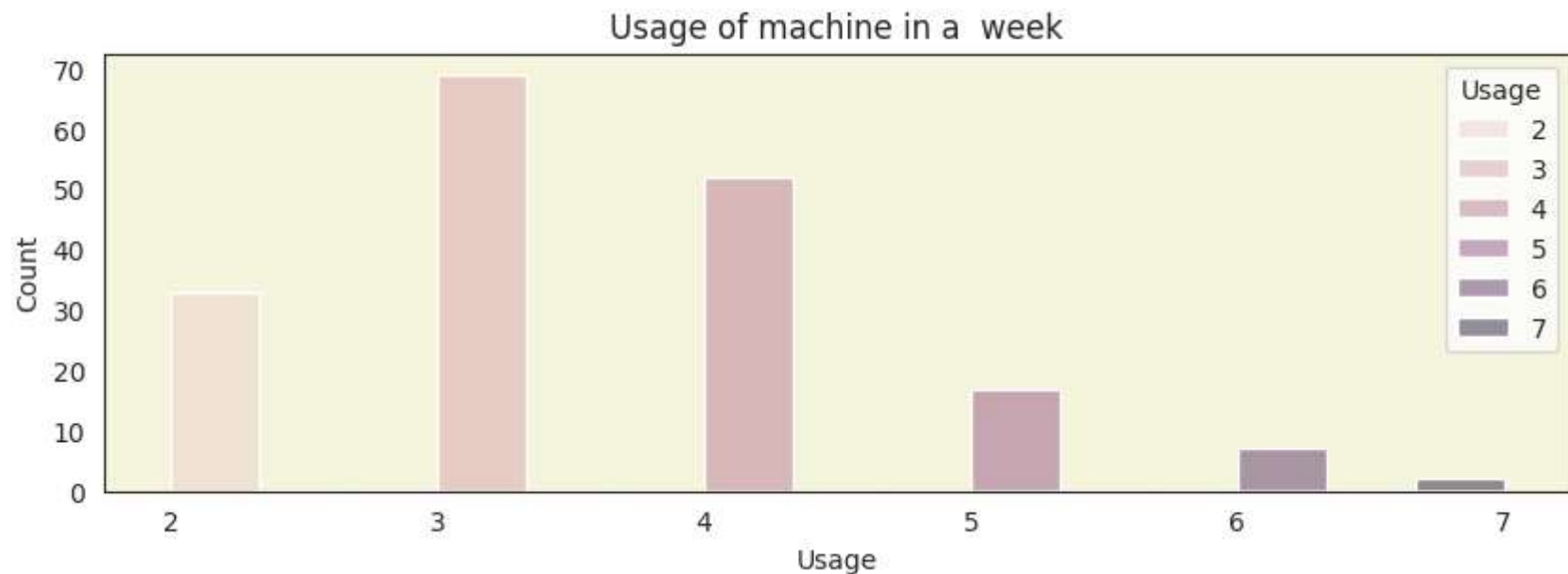
Name: Age, dtype: int64

- The above of graph displays the number of people falling under specific age group using the machine.
- We can say people of age between 23 - 36 purchased the machines more than any other age group

▾ Displaying the usage count of the machines

```
plt.figure(figsize = (10,3))
sns.histplot(data=data, x="Usage", kde=True, hue = 'Usage')
ax = plt.gca()
ax.set_facecolor('beige')
```

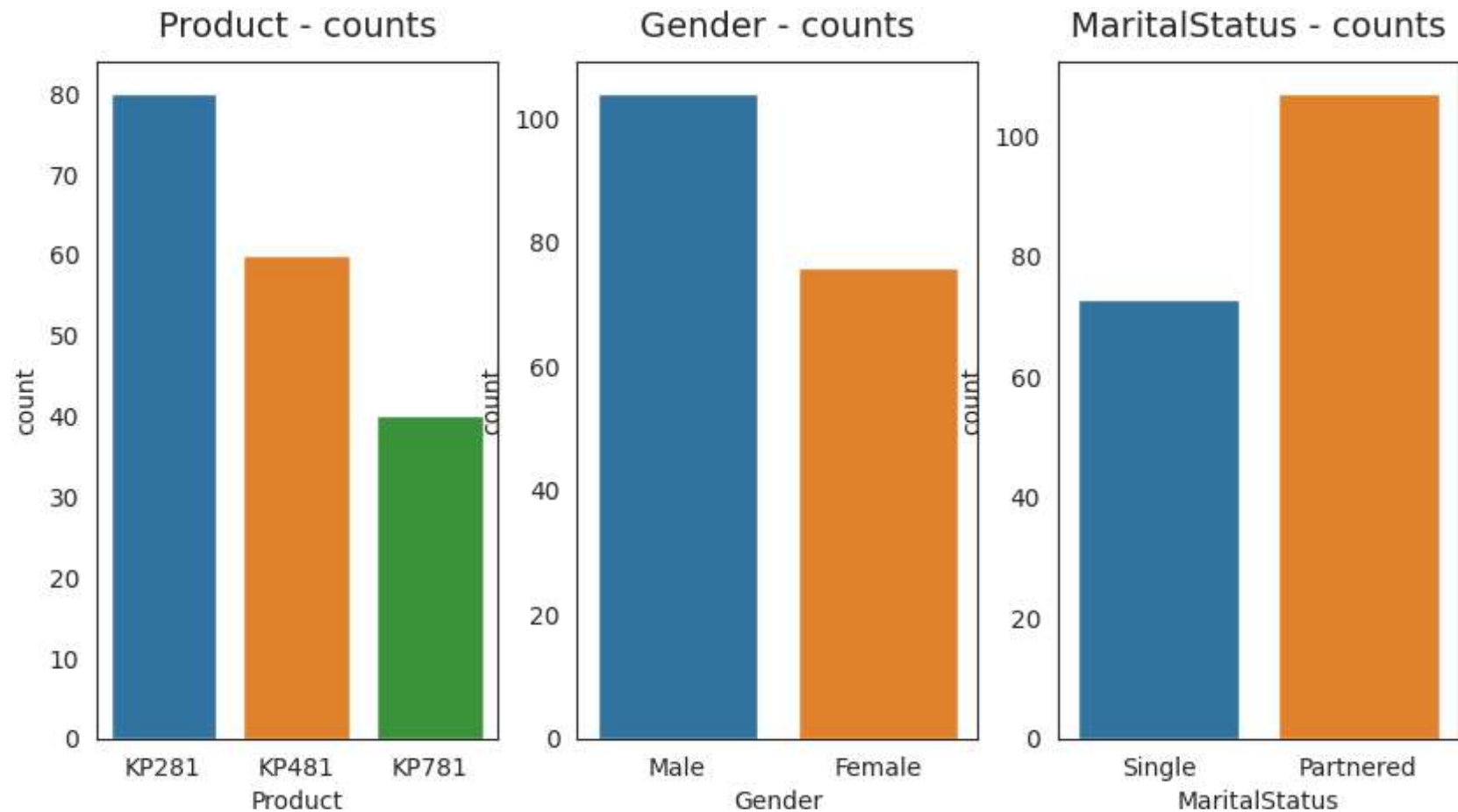
```
plt.title('Usage of machine in a week')
plt.show()
```



- Based on the above observation we can say majority of the people use the machine 3 times a week.

▼ Understanding the distribution of the data for the categorical attributes:

```
fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(10,5))
sns.countplot(data, x = 'Product', ax=axs[0])
sns.countplot(data, x = 'Gender', ax=axs[1])
sns.countplot(data, x = 'MaritalStatus', ax=axs[2])
axs[0].set_title("Product - counts", pad=10, fontsize=14)
axs[1].set_title("Gender - counts", pad=10, fontsize=14)
axs[2].set_title("MaritalStatus - counts", pad=10, fontsize=14)
plt.show()
```





Observations

1. KP281 is the most frequent product.
2. There are more Males in the data than Females.
3. More Partnered persons are there in the data.

▼ To be precise - normalized count for each variable is shown below:

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

		value	
variable	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	

Observation -

1. Product

- 44.44% of the customers have purchased KP28 product
- 33.33% of the customers have purchased KP481 product
- 22.22% of the customers have purchased KP781 product

2. Gender

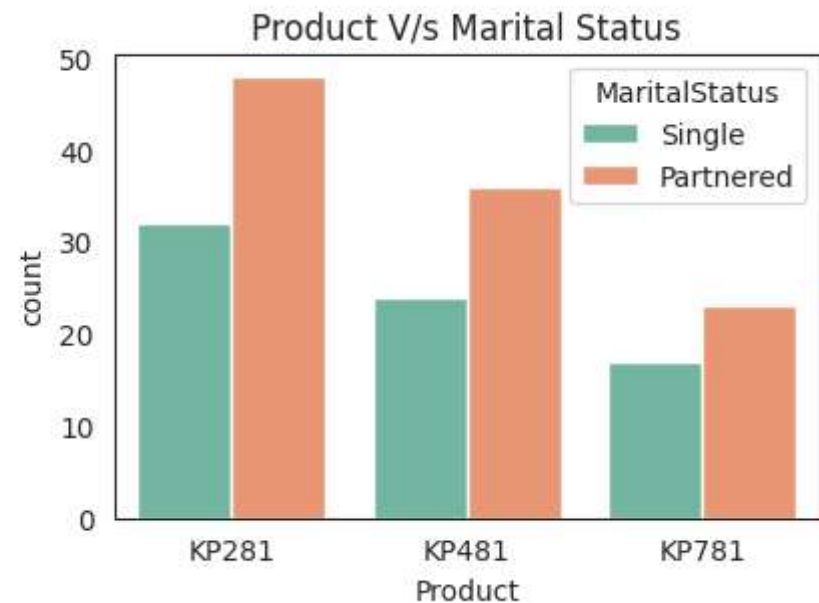
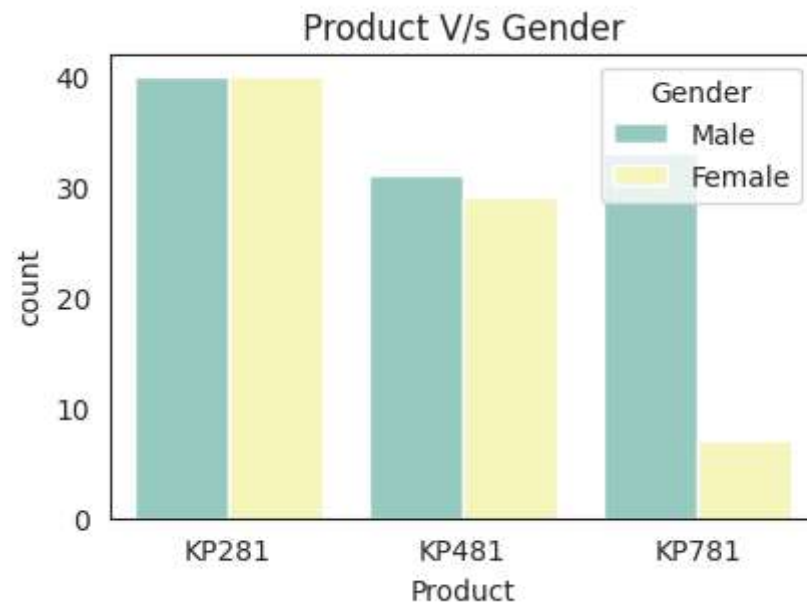
- 57.78% of the customers are Male.
- List item

3. MaritalStatus

- 59.44% of the customers are Partnered.

▼ Bivariate Analysis

```
fig, axs = plt.subplots(nrows = 1, ncols = 2, figsize=(10,3))
sns.countplot(data, x = 'Product', hue = 'Gender',palette = 'Set3', ax = axs[0])
sns.countplot(data, x = 'Product', hue = 'MaritalStatus',palette = 'Set2', ax = axs[1])
axs[0].set_title('Product V/s Gender')
axs[1].set_title('Product V/s Marital Status')
plt.show()
```



Observations :

Product vs Gender

1. Same number of males and females have purchased KP281 product and Almost same for the product KP481
2. The KP781 product is purchased by most men

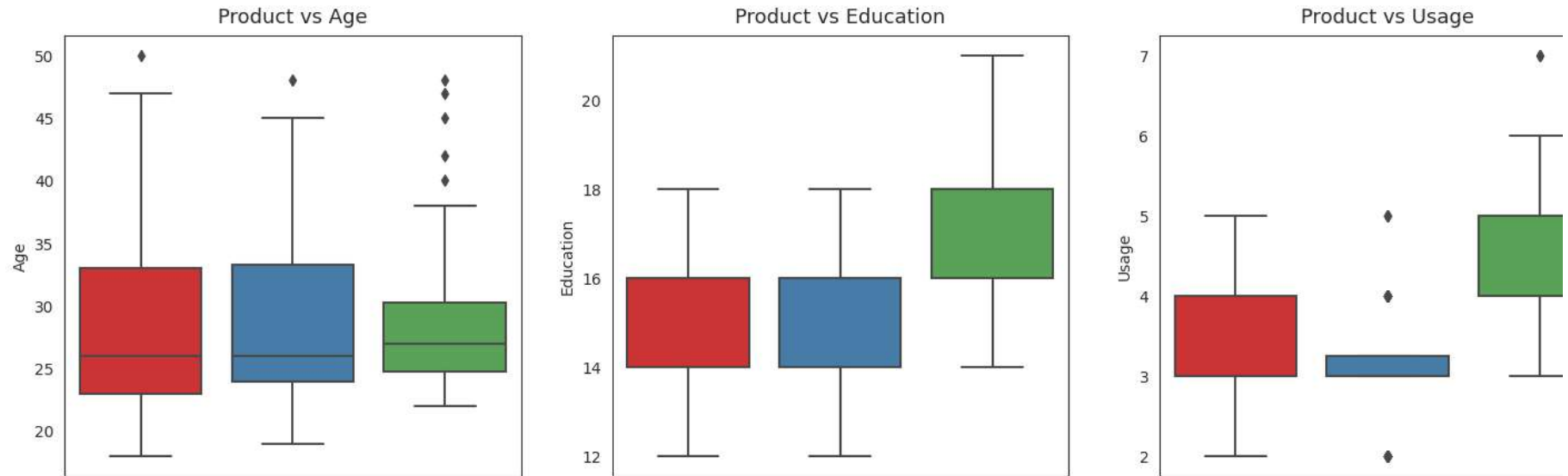
Product vs MaritalStatus

1. Customer who is Partnered, is more likely to purchase the product.

Checking the impact of fields('Age','Education','Usage','Fitness','Income','Miles') on product

```
fields = ['Age','Education','Usage','Fitness','Income','Miles']

count = 0
fig, axs = plt.subplots(nrows = 2, ncols = 3, figsize=(18,8))
fig.subplots_adjust(top=1.2)
for i in range(2):
    for j in range(3):
        sns.boxplot(data, x = 'Product', y = fields[count], ax = axs[i,j] ,palette='Set1')
        axs[i,j].set_title(f"Product vs {fields[count]}",pad=8, fontsize=13 )
        count += 1
plt.show()
```



Observations :

1. 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

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 Fitness

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

5. Product vs Income

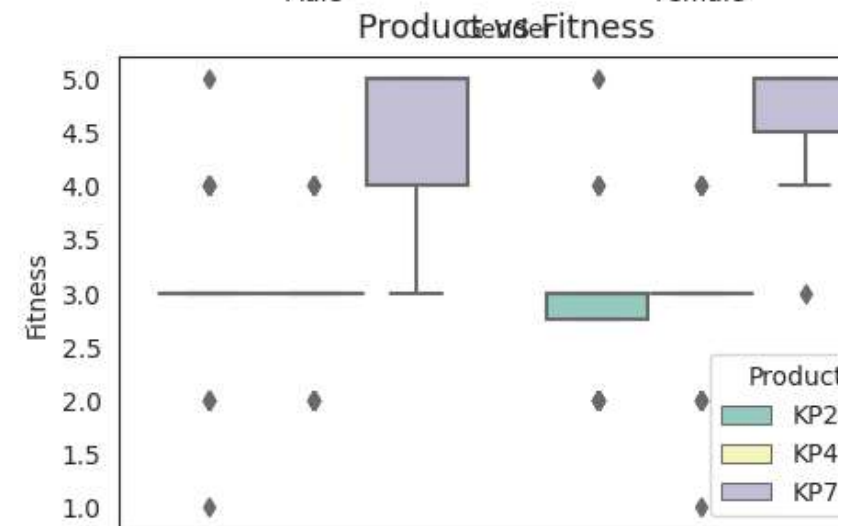
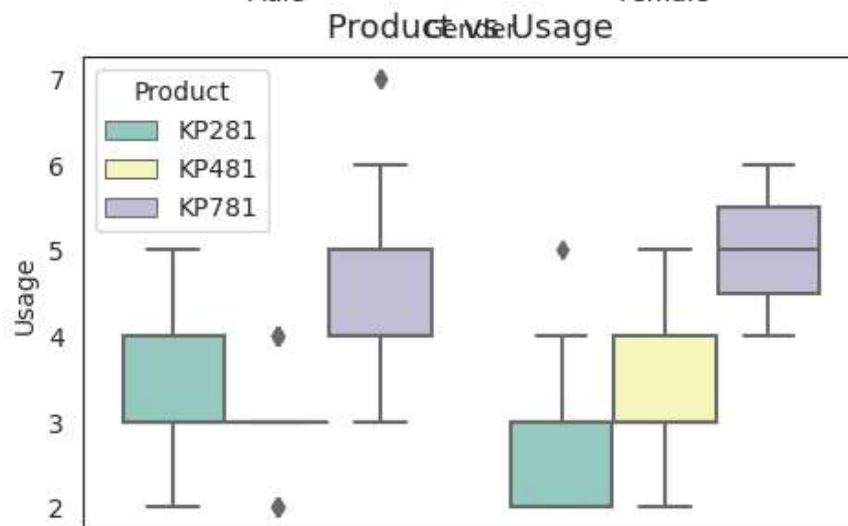
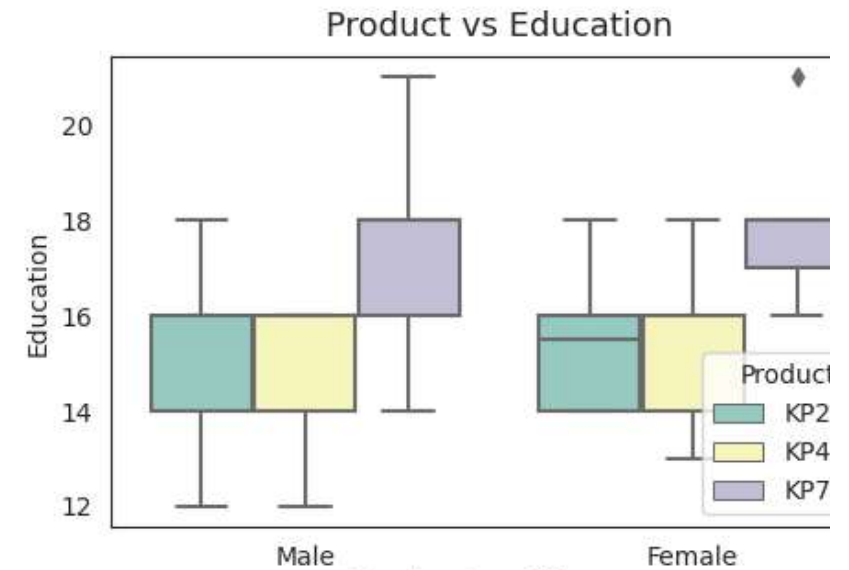
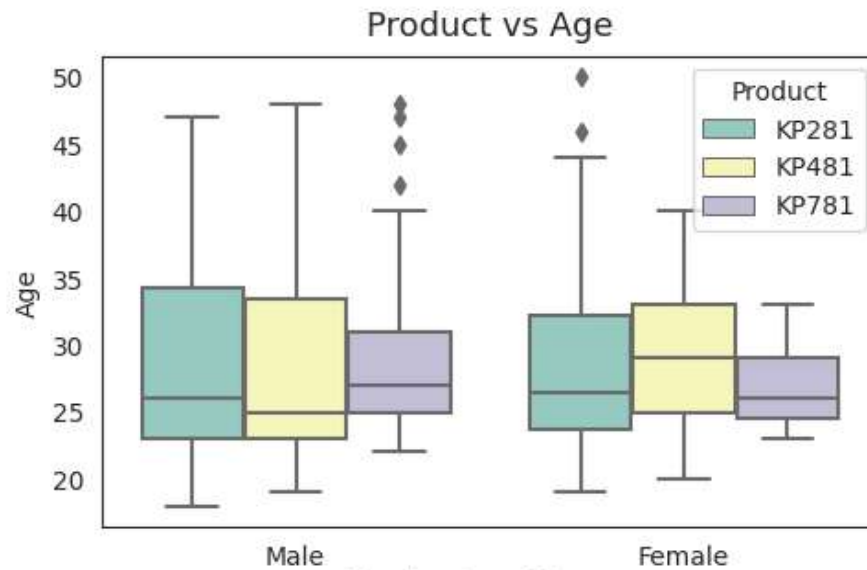
- Higher the Income of the customer (Income \geq 60000), higher the chances of the customer to purchase the KP781 product.

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.

▼ Multivariate Analysis

```
fields = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(12, 8))
fig.subplots_adjust(top=1.2)
count = 0
for i in range(3):
    for j in range(2):
        sns.boxplot(data=data, x='Gender', y=fields[count], hue='Product',
                    ax=axs[i,j], palette='Set3')
        axs[i,j].set_title(f"Product vs {fields[count]}", pad=8, fontsize=13)
        count += 1
```



Here we are plotting boxplot graph for gender v/s all the continuous variables.

Observations :

1. Men falling in age group 24-35 tend to buy qual number of KP281 and KP481
2. Females planning to use treadmill 3-4 times a week, are more likely to buy KP481 product
3. Males who are fit (fitness between 4-5) tend to buy KP781

4. Both male and female who are earning big end up buying KP781



▼ Computing Marginal & Conditional Probabilities:

- Marginal Probability

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

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

▼ - Conditional Probabilities

Probability of each product for specific gender:

```
#data['Product'].groupby('Gender').value_counts()
df = data.groupby('Gender')['Product'].value_counts()/len(data)
df
```

```
Gender  Product
Female  KP281    0.222222
        KP481    0.161111
        KP781    0.038889
Male    KP281    0.222222
        KP781    0.183333
        KP481    0.172222
Name: Product, dtype: float64
```

customer profile for particular products

KP281:

- Most affordable and entry-level machine and Maximum Selling Product.
- This model is popular amongst both Male and Female customers
- Same number of Male and Female customers.
- Customers walk/run an average of 70 to 90 miles on this product.
- Customers use it 3 to 4 times a week
- Fitness Level of this product users is Average
- Used by all age groups and fitness levels.

KP481:

- Intermediate Price Range
- Fitness Level of this product users varies from Bad to Average Shape depending on their usage.
- Customers prefer the KP481 model to use less frequently but to run more miles per week on this.
- Customer walks/runs an average of 70 to 130 or more miles per week on his product.
- Probability of Female customers buying KP481 is significantly higher than male.
- customers are from the adult, teen, and mid-age categories.

KP781:

- least sold product
- high price and preferred by customers who do exercises more extensively and run more miles.
- Customer walks/runs an average of 120 to 200 or more miles per week on his product.
- Customers use 4 to 5 times a week at least.
- If a person is in Excellent Shape, the probability that he is using KP781 is more than 90%.
- Female Customers who are running an average of 180 miles (extensive exercise), are using product KP781, which is higher than the Male average using the same product.
- KP781 can be recommended for Female customers who exercise extensively.

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

Recommendations :

- Probability of Men using KP481 is too low, Aerofit should work on it to be more attractive
- most of the partnered customers tend to use treadmills. Keeping this in mind Aerofit should offer some coupons/ discounts for them to gain more popularity and sales
- Aerofit should work on ideas to attract people above 30 age. As most of the youngsters are using treadmills frequently
- We see Both male and female who are earning big end up buying KP781. Aerofit should try to make it affordable too for less/ moderate earning individuals

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 3:48 PM

