jsxh5cdyl

December 17, 2024

1 Aerofit Business Case Study

2 Problem Statement:

In the competitive fitness industry, understanding customer behavior and preferences is vital for strategic decision-making. Our goal is to analyze the customer data collected from Aerofit Fitness and gain actionable insights into various aspects of customer behavior.

This analysis will assist in tailoring marketing strategies, product offerings, and customer experiences to better align with customer preferences. The analysis involves investigating the relationships between different variables as mentioned below. Basic metrics: gender, marital status, education, age, income, fitness level, and product purchases.

3 Libraries

Below are the libraries required for **analysing and visualizing data.**

[ ]:
```
# Libraries to analyze data
import numpy as np
import pandas as pd

# Libraries to visualize data
import matplotlib.pyplot as plt
import seaborn as sns
```

4 Data loading and initial analysis

Loading the data into Pandas dataframe for easily handling of data

[ ]:
```
filepath = 'https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/
  ↪125/original/aerofit_treadmill.csv?1639992749'
df = pd.read_csv(filepath)
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
```

5 Analysis

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

5.1.1 a. The data type of all columns in the "customers" table.

Hint: We want you to display the data type of each column present in the 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
```

5.1.2 b. You can find the number of rows and columns given in the dataset.

Hint: We want you to find the shape of the dataset.

[ ]:
```
df.shape
```

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

Insight:

So our data has 180 rows and 9 columns.

[ ]:
```
df.columns
```

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

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

[ ]:
```
                Age   Education       Usage     Fitness         Income  \
count  180.000000  180.000000  180.000000  180.000000     180.000000
mean    28.788889   15.572222    3.455556    3.311111   53719.577778
std      6.943498    1.617055    1.084797    0.958869   16506.684226
min     18.000000   12.000000    2.000000    1.000000   29562.000000
25%     24.000000   14.000000    3.000000    3.000000   44058.750000
50%     26.000000   16.000000    3.000000    3.000000   50596.500000
75%     33.000000   16.000000    4.000000    4.000000   58668.000000
max     50.000000   21.000000    7.000000    5.000000  104581.000000

            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
```

[ ]:
```
df['Product'].unique()
```

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

5.1.3 c. Check for the missing values and find the number of missing values in each column

[ ]:
```
## Checking for missing values

df.isna().sum()
```

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

Insight:

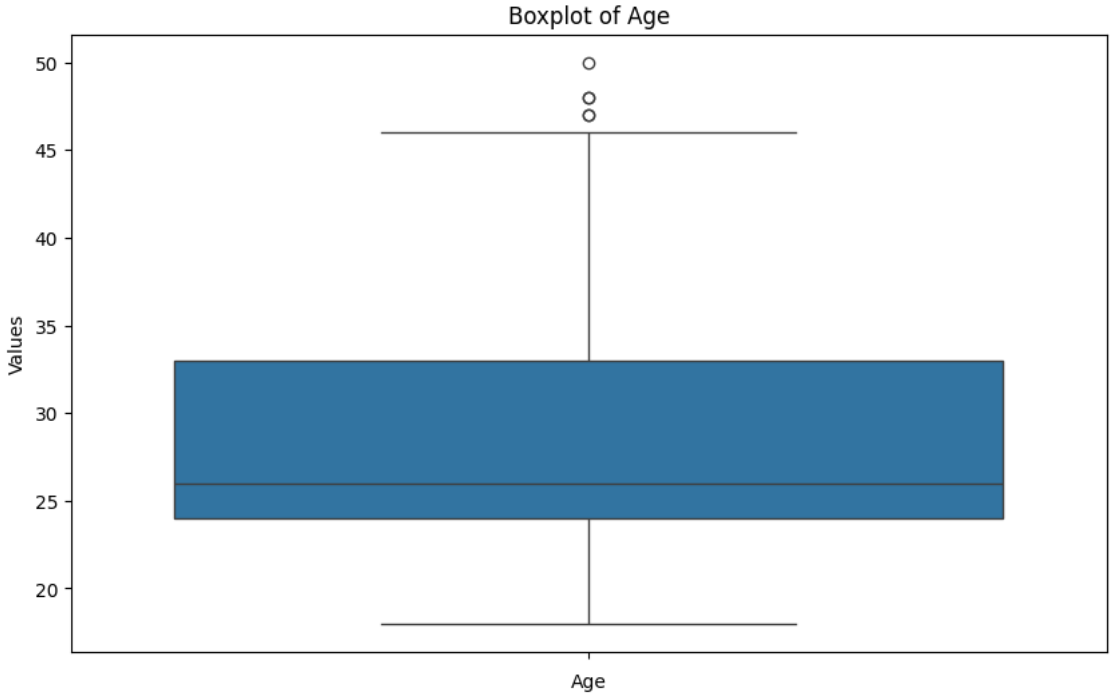
There is no missing value present in the dataset.

5.2 Detect Outliers

5.2.1 a. Find the outliers for every continuous variable in the dataset

Hint: We want you to use boxplots to find the outliers in the given dataset.

[ ]:
```
#Checking the outliers of continuous Variables.

# Visualize boxplots for Age variable
plt.figure(figsize=(10, 6))
sns.boxplot(data=df,y='Age')
plt.title('Boxplot of Age')
plt.xlabel('Age')
plt.ylabel('Values')
plt.xticks(rotation=90)
plt.show()
```

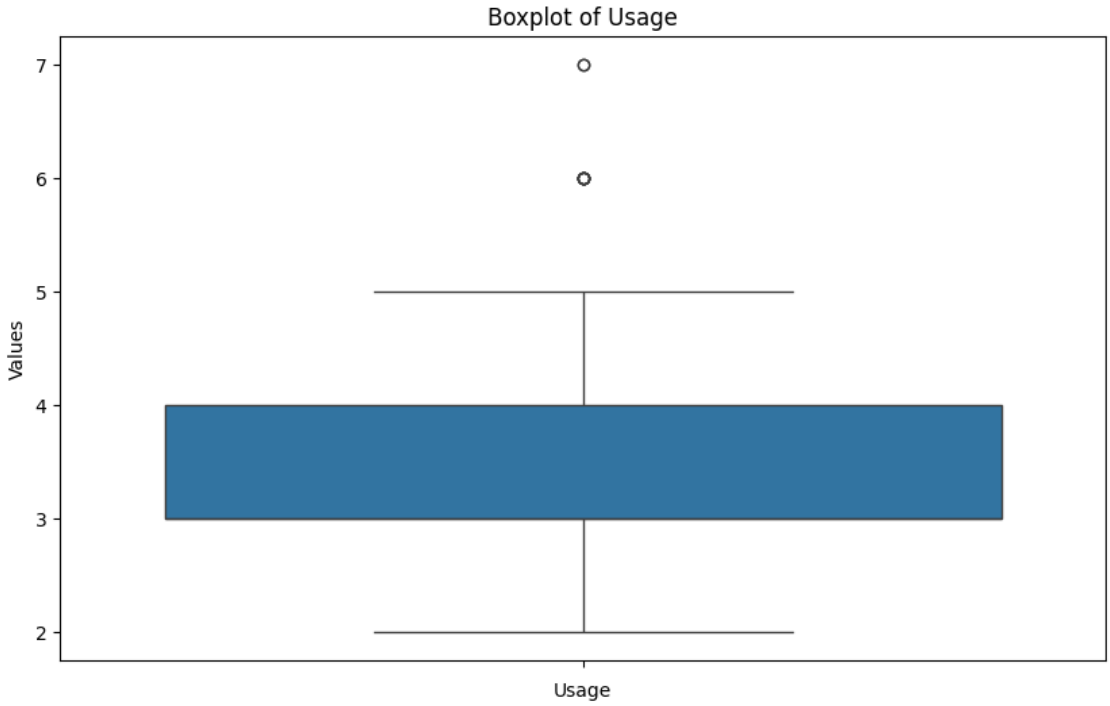


Insights:

1. From the boxplot we can see there are 3 outliers present in the dataset.
2. The median age is 26 & most of the cuistomers belong in the range of 24-33 years.

[ ]:
```
# Visualize boxplots for Usage variable
plt.figure(figsize=(10, 6))
sns.boxplot(df['Usage'])
plt.title('Boxplot of Usage')
```

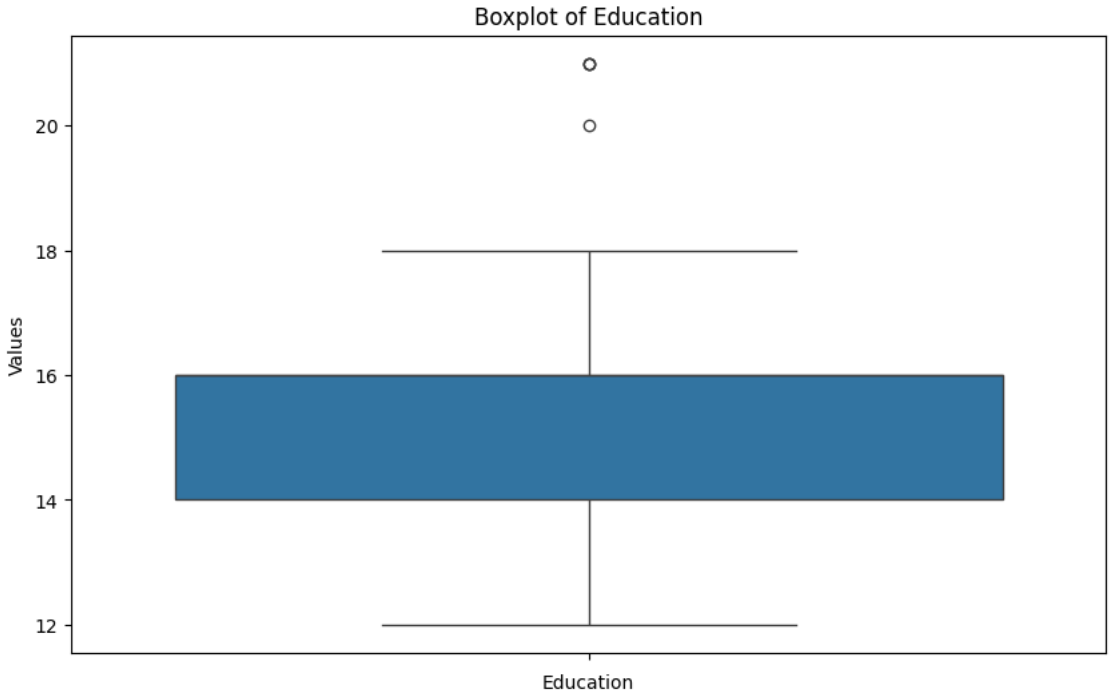
```
plt.xlabel('Usage')
plt.ylabel('Values')
plt.xticks(rotation=45)
plt.show()
```



Insights:

1. From the boxplot we can see there are 2 outliers present in the Usage column. It's interesting to note that there is only 1 customer who uses the treadmill 7 days per week and one customer who use the product 6 days per week.
2. The median Usage is 3 days per week & most of the customers use the instrument in the range of 3-4 days per week .

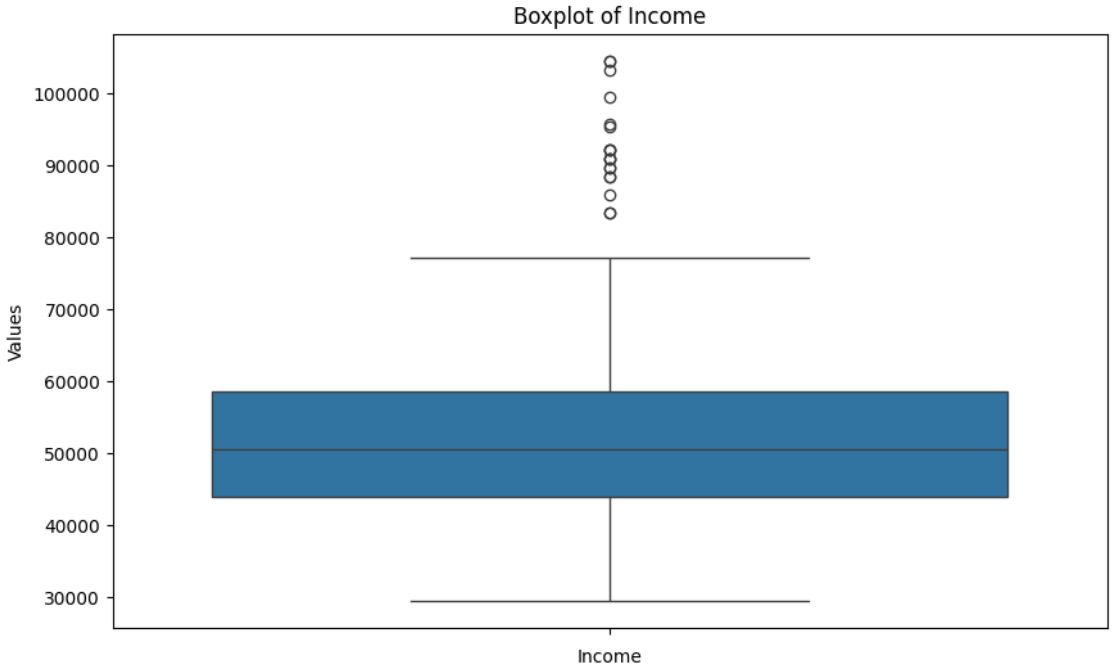
[ ]:
```
# Visualize boxplots for Education variable
plt.figure(figsize=(10, 6))
sns.boxplot(df['Education'])
plt.title('Boxplot of Education')
plt.xlabel('Education')
plt.ylabel('Values')
plt.xticks(rotation=45)
plt.show()
```



Insights:

1. From the boxplot we can see there are 2 outliers present in the Education variable.
2. The median education is of 16 Years & most of the customers recieved education in the range of 14-16 years.

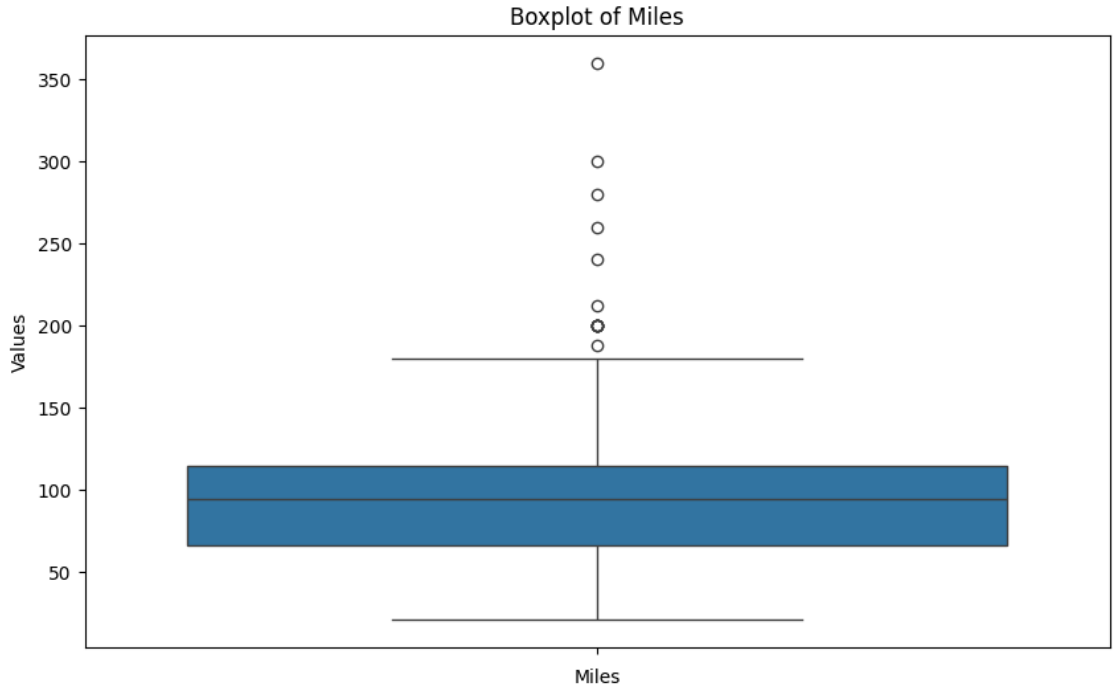
[ ]:
```
# Visualize boxplots for Income variable
plt.figure(figsize=(10, 6))
sns.boxplot(df['Income'])
plt.title('Boxplot of Income')
plt.xlabel('Income')
plt.ylabel('Values')
plt.xticks(rotation=45)
plt.show()
```



Insights:

1. From the boxplot we can see there are many outliers present in the Income Variable. This indicates that aerofit caters to some rich customers who have annual income of more than $80000.
2. The median income of customers is 51k & most of the cuistomers belong in the range of 44k-59k per year.

[ ]:
```
# Visualize boxplots for Miles variable
plt.figure(figsize=(10, 6))
sns.boxplot(df['Miles'])
plt.title('Boxplot of Miles')
plt.xlabel('Miles')
plt.ylabel('Values')
plt.xticks(rotation=45)
plt.show()
```



Insights:

1. From the boxplot we can see there are many outliers present in the Miles Variable. This indicates that aerofit caters to some health concious/athlete customers who run more than 180 miles per week.
2. The median miles per week of customers is 94 miles & most of the cuistomers belong in the range of 66-115 miles per week.

5.2.2 b. Remove/clip the data between the 5 percentile and 95 percentile

Hint: We want You to use np.clip() for clipping the data

[ ]:
```
# Clipping the values for education column
fifth_percentile=np.percentile(df['Education'],5)
max_percentile=np.percentile(df['Education'],95)
df['Education_clipped']=np.clip(df['Education'],fifth_percentile,max_percentile)
df
```

[ ]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	\
0	KP281	18	Male	14	Single	3	4	29562	
1	KP281	19	Male	15	Single	2	3	31836	
2	KP281	19	Female	14	Partnered	4	3	30699	
3	KP281	19	Male	12	Single	3	3	32973	
4	KP281	20	Male	13	Partnered	4	2	35247	
..	…	…	…	…	…	…	…	…	

```
175   KP781   40    Male          21         Single      6         5   83416
176   KP781   42    Male          18         Single      5         4   89641
177   KP781   45    Male          16         Single      5         5   90886
178   KP781   47    Male          18      Partnered      4         5  104581
179   KP781   48    Male          18      Partnered      4         5   95508

     Miles  Education_clipped
0      112                14
1       75                15
2       66                14
3       85                14
4       47                14
..     …                 …
175    200                18
176    200                18
177    160                16
178    120                18
179    180                18

[180 rows x 10 columns]
```

[ ]:
```
# Clipping the values for age column
fifth_percentile=np.percentile(df['Age'],5)
max_percentile=np.percentile(df['Age'],95)
df['Age_clipped']=np.clip(df['Age'],fifth_percentile,max_percentile)
df
```

[ ]:
```
     Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
0     KP281   18    Male          14         Single      3         4   29562
1     KP281   19    Male          15         Single      2         3   31836
2     KP281   19  Female          14      Partnered      4         3   30699
3     KP281   19    Male          12         Single      3         3   32973
4     KP281   20    Male          13      Partnered      4         2   35247
..      …   …     …         …            …    …       …      …
175   KP781   40    Male          21         Single      6         5   83416
176   KP781   42    Male          18         Single      5         4   89641
177   KP781   45    Male          16         Single      5         5   90886
178   KP781   47    Male          18      Partnered      4         5  104581
179   KP781   48    Male          18      Partnered      4         5   95508

     Miles  Education_clipped  Age_clipped
0      112                14        20.00
1       75                15        20.00
2       66                14        20.00
3       85                14        20.00
4       47                14        20.00
..     …                 …           …
```

```
175    200                 18        40.00
176    200                 18        42.00
177    160                 16        43.05
178    120                 18        43.05
179    180                 18        43.05

[180 rows x 11 columns]
```

[ ]:
```
# Clipping the values for Income column
fifth_percentile=np.percentile(df['Income'],5)
max_percentile=np.percentile(df['Income'],95)
df['Income_clipped']=np.clip(df['Income'],fifth_percentile,max_percentile)
df
```

[ ]:
```
     Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
0      KP281   18    Male         14        Single      3        4   29562
1      KP281   19    Male         15        Single      2        3   31836
2      KP281   19  Female         14     Partnered      4        3   30699
3      KP281   19    Male         12        Single      3        3   32973
4      KP281   20    Male         13     Partnered      4        2   35247
..       …   …     …          …             …     …      …      …
175    KP781   40    Male         21        Single      6        5   83416
176    KP781   42    Male         18        Single      5        4   89641
177    KP781   45    Male         16        Single      5        5   90886
178    KP781   47    Male         18     Partnered      4        5  104581
179    KP781   48    Male         18     Partnered      4        5   95508

     Miles  Education_clipped  Age_clipped  Income_clipped
0      112                 14        20.00        34053.15
1       75                 15        20.00        34053.15
2       66                 14        20.00        34053.15
3       85                 14        20.00        34053.15
4       47                 14        20.00        35247.00
..     …                 …          …             …
175    200                 18        40.00        83416.00
176    200                 18        42.00        89641.00
177    160                 16        43.05        90886.00
178    120                 18        43.05        90948.25
179    180                 18        43.05        90948.25

[180 rows x 12 columns]
```

[ ]:
```
# Clipping the values for Miles column
fifth_percentile=np.percentile(df['Miles'],5)
max_percentile=np.percentile(df['Miles'],95)
df['Miles_clipped']=np.clip(df['Miles'],fifth_percentile,max_percentile)
df
```

[ ]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	\
0	KP281	18	Male	14	Single	3	4	29562	
1	KP281	19	Male	15	Single	2	3	31836	
2	KP281	19	Female	14	Partnered	4	3	30699	
3	KP281	19	Male	12	Single	3	3	32973	
4	KP281	20	Male	13	Partnered	4	2	35247	
..	…	…	…	…	…	…	…	…	
175	KP781	40	Male	21	Single	6	5	83416	
176	KP781	42	Male	18	Single	5	4	89641	
177	KP781	45	Male	16	Single	5	5	90886	
178	KP781	47	Male	18	Partnered	4	5	104581	
179	KP781	48	Male	18	Partnered	4	5	95508	

	Miles	Education_clipped	Age_clipped	Income_clipped	Miles_clipped
0	112	14	20.00	34053.15	112
1	75	15	20.00	34053.15	75
2	66	14	20.00	34053.15	66
3	85	14	20.00	34053.15	85
4	47	14	20.00	35247.00	47
..	…	…	…	…	…
175	200	18	40.00	83416.00	200
176	200	18	42.00	89641.00	200
177	160	16	43.05	90886.00	160
178	120	18	43.05	90948.25	120
179	180	18	43.05	90948.25	180

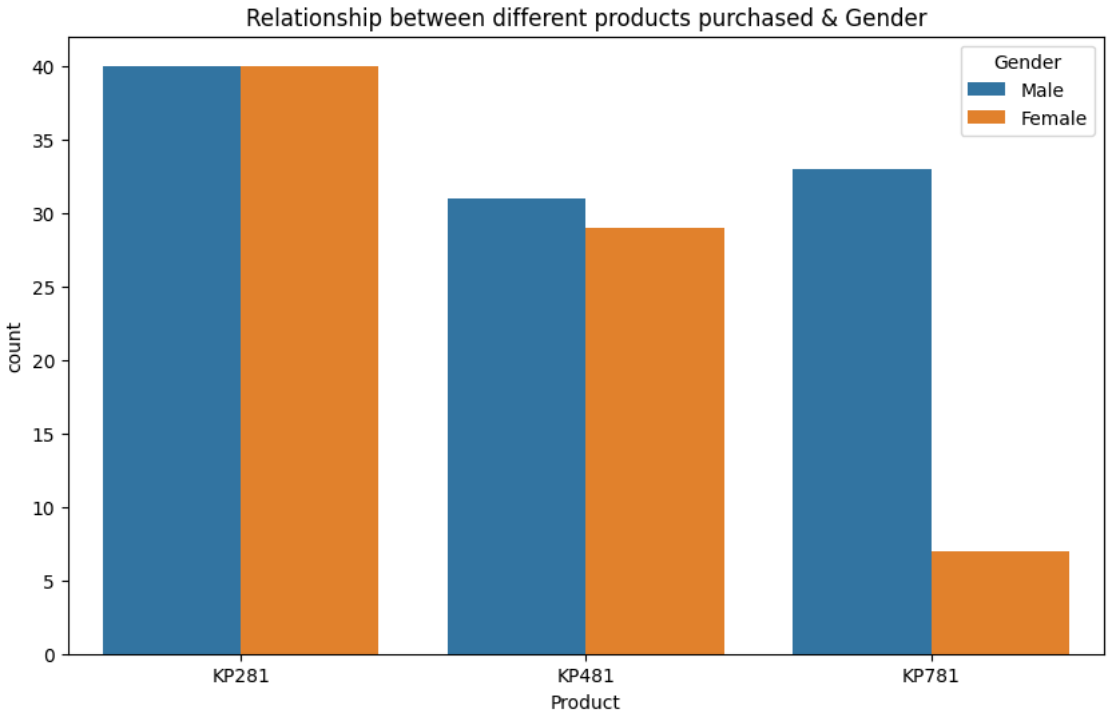
[180 rows x 13 columns]

5.3 3. Check if features like marital status, Gender, and age have any effect on the product purchased.

5.3.1 a. Find if there is any relationship between the categorical variables and the output variable in the data.

Hint: We want you to use the count plot to find the relationship between categorical variables and output variables.

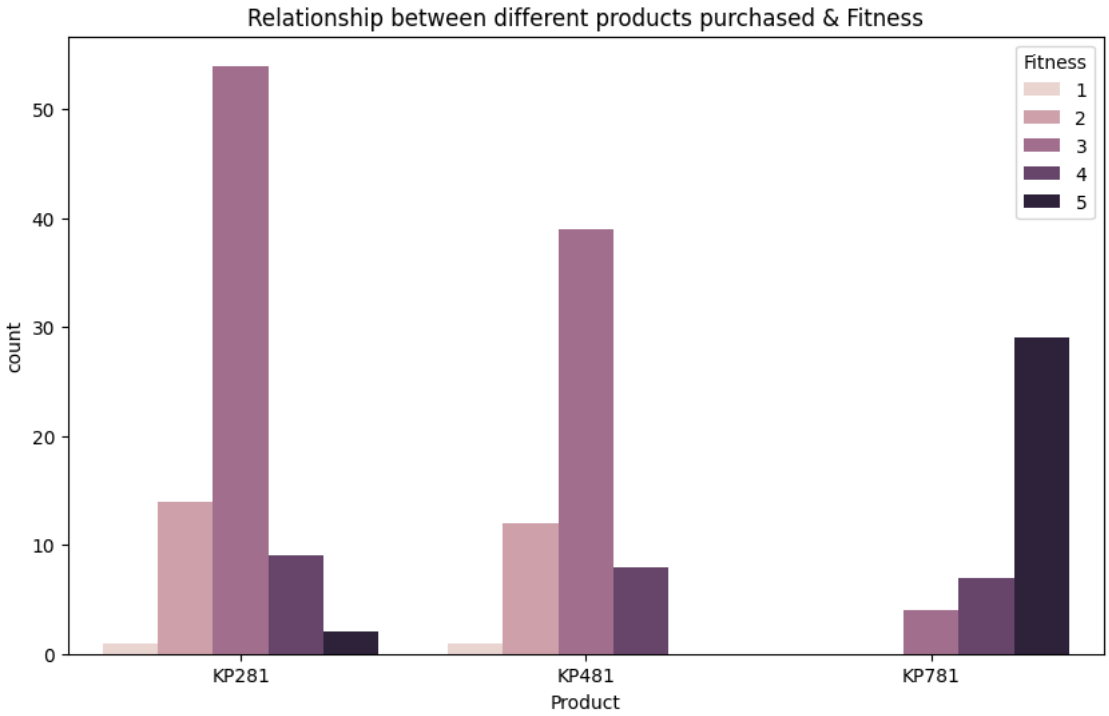
[ ]:
```
plt.figure(figsize=(10,6))
sns.countplot(data=df,x='Product',hue='Gender')
plt.title('Relationship between different products purchased & Gender')
plt.show()
```



Insights:

1. KP781 which is an advanced instrument, is preferred mostly by the male users.
2. Kp281 being a beginner's trademill , has the highest number of purchases. It's equally preferred by both males and females.

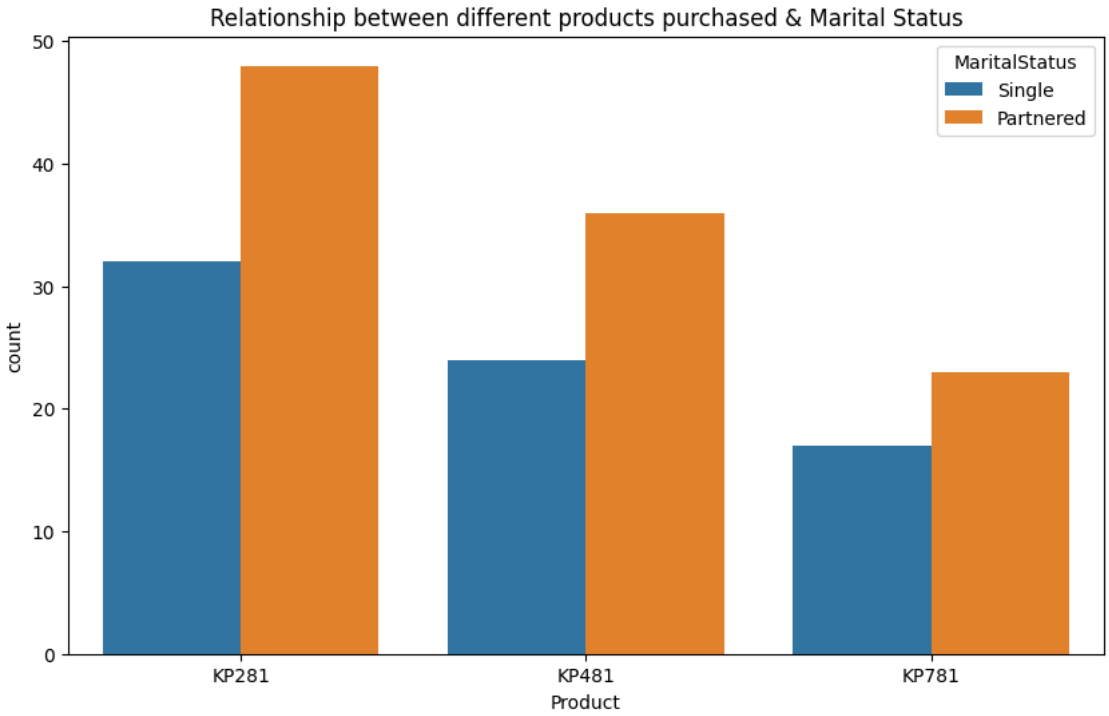
[ ]:
```
plt.figure(figsize=(10,6))
sns.countplot(data=df,x='Product',hue='Fitness')
plt.title('Relationship between different products purchased & Fitness')
plt.show()
```



Insights:

1. Customers with highest fitness level (Level-5) have preferred the most advanced trademill KP781.
2. Customers with moderate fitness level (LEVEL:2-4) has preferred KP281 & KP481.

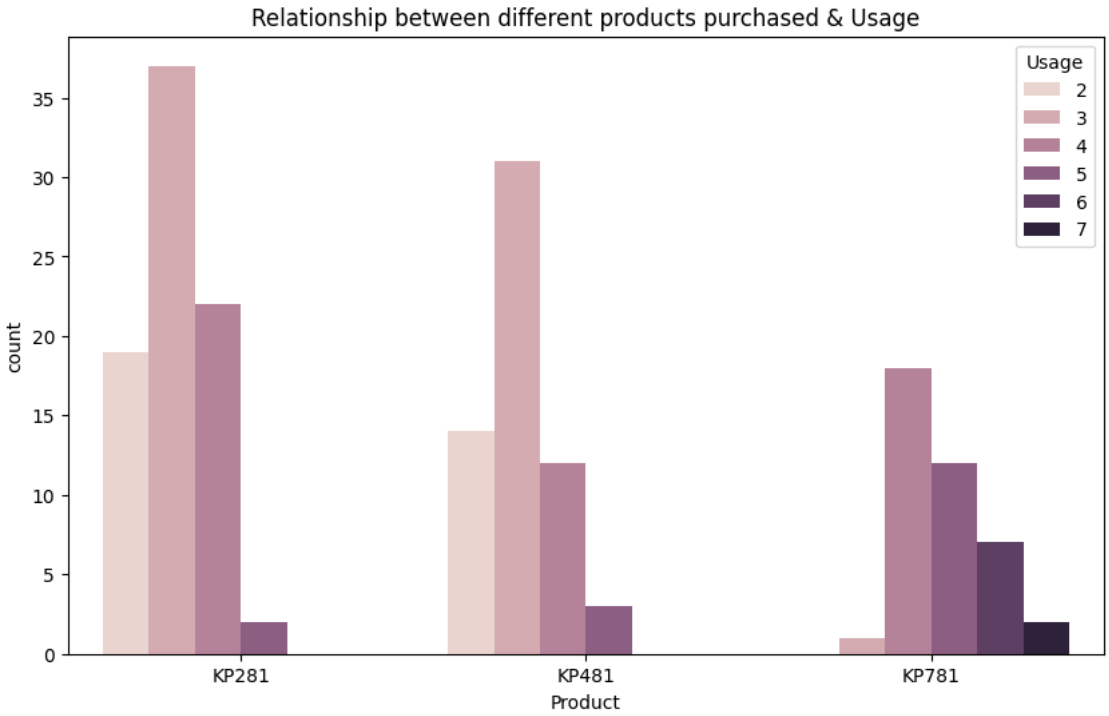
[ ]:
```
plt.figure(figsize=(10,6))
sns.countplot(data=df,x='Product',hue='MaritalStatus')
plt.title('Relationship between different products purchased & Marital Status')
plt.show()
```



Insights:

1. Married people are more likely to purchase fitness equipment rather than their single counterparts.

[ ]:
```
plt.figure(figsize=(10,6))
sns.countplot(data=df,x='Product',hue='Usage')
plt.title('Relationship between different products purchased & Usage')
plt.show()
```



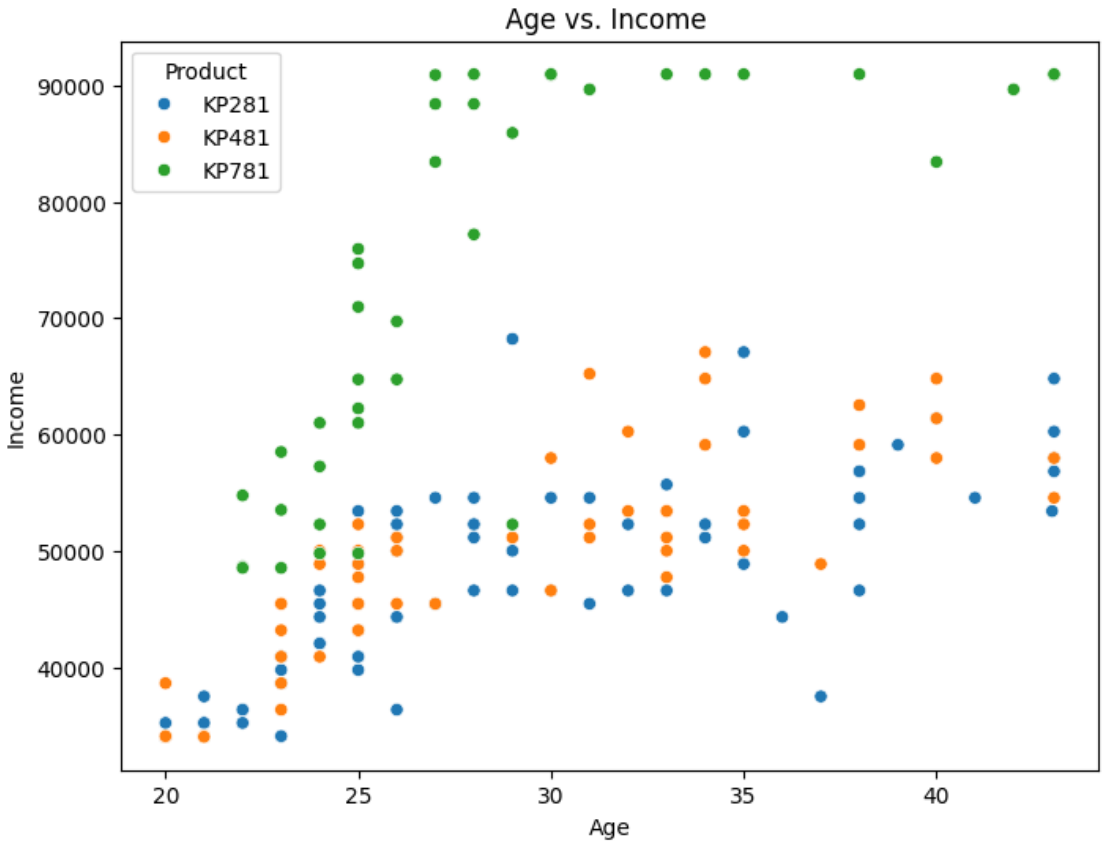
Insights:

1. People with moderate usgege (less than 4 days per week) , are preferring KP281 & KP481.
2. Whereas, people with avg. usage of more than 4 days per week prefer the advanced trademill KP781.

5.3.2 b. Find if there is any relationship between the continuous variables and the output variable in the data.

Hint: We want you to use a scatter plot to find the relationship between continuous variables and output variables.

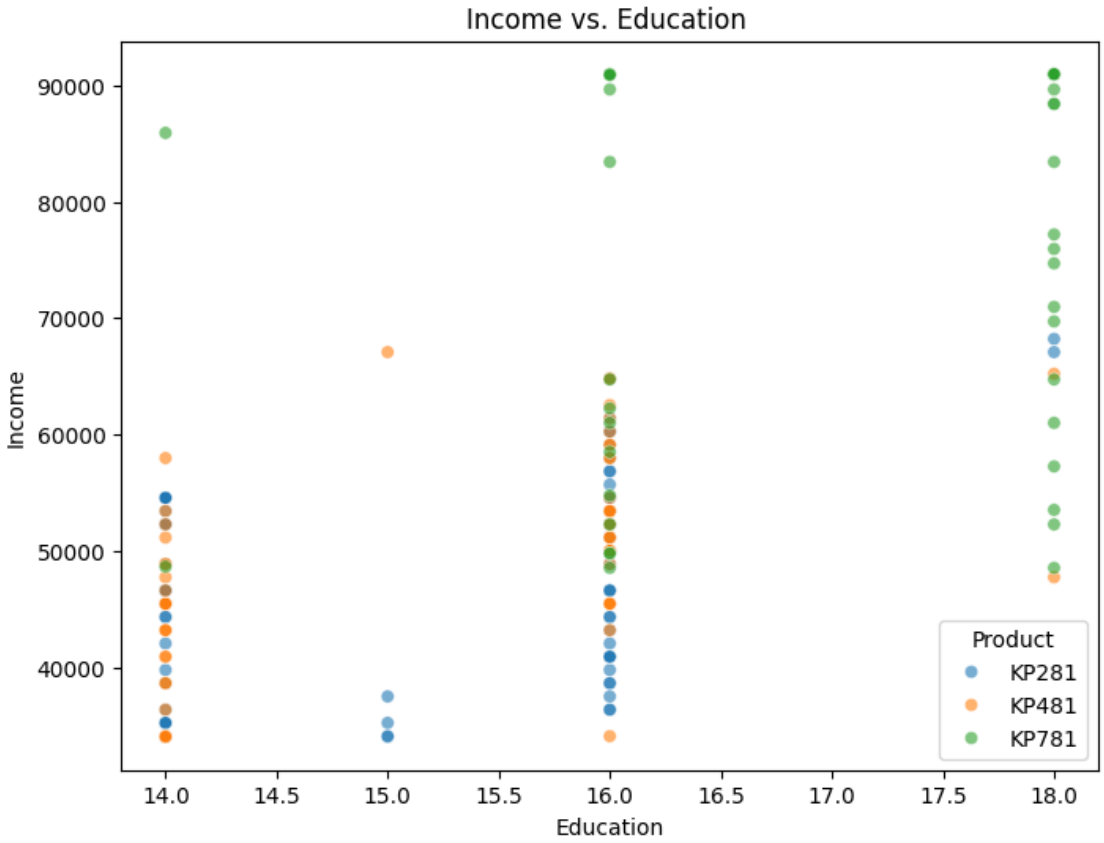
[ ]:
```
# Scatter plot for 'Age' vs. 'Income' for different products
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age_clipped', y='Income_clipped', data=df,hue='Product')
plt.title('Age vs. Income ')
plt.xlabel('Age')
plt.ylabel('Income')
plt.show()
```



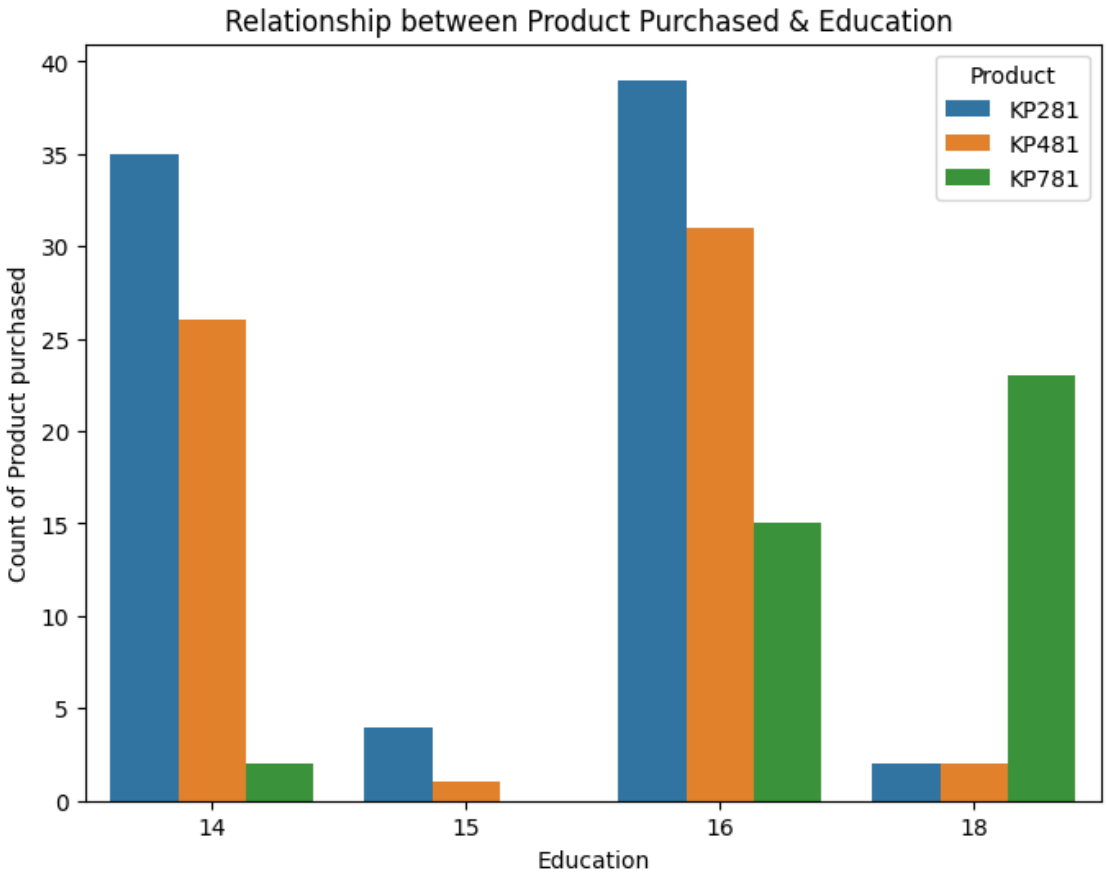
Insights:

- People with higher income are more likely to buy the premium KP781 trademill.
- People below the age 30 are more likely to buy the advanced trademill.
- People above the age of 30 and income below are more likely to buy the KP281 and KP781.
- Rich customers who earn over $70,000 may choose KP781 regardless of their age, fitness, or gender. This is because they view it as a luxury item and are likely to purchase it despite the high price to maintain their social status and avoid losing prestige.

[ ]:
```
# Scatter plot for 'Age' vs. 'Income' for different products
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Education_clipped', y='Income_clipped',␣
 ↪data=df,hue='Product',alpha=0.6)
plt.title('Income vs. Education ')
plt.xlabel('Education')
plt.ylabel('Income')
plt.show()
```



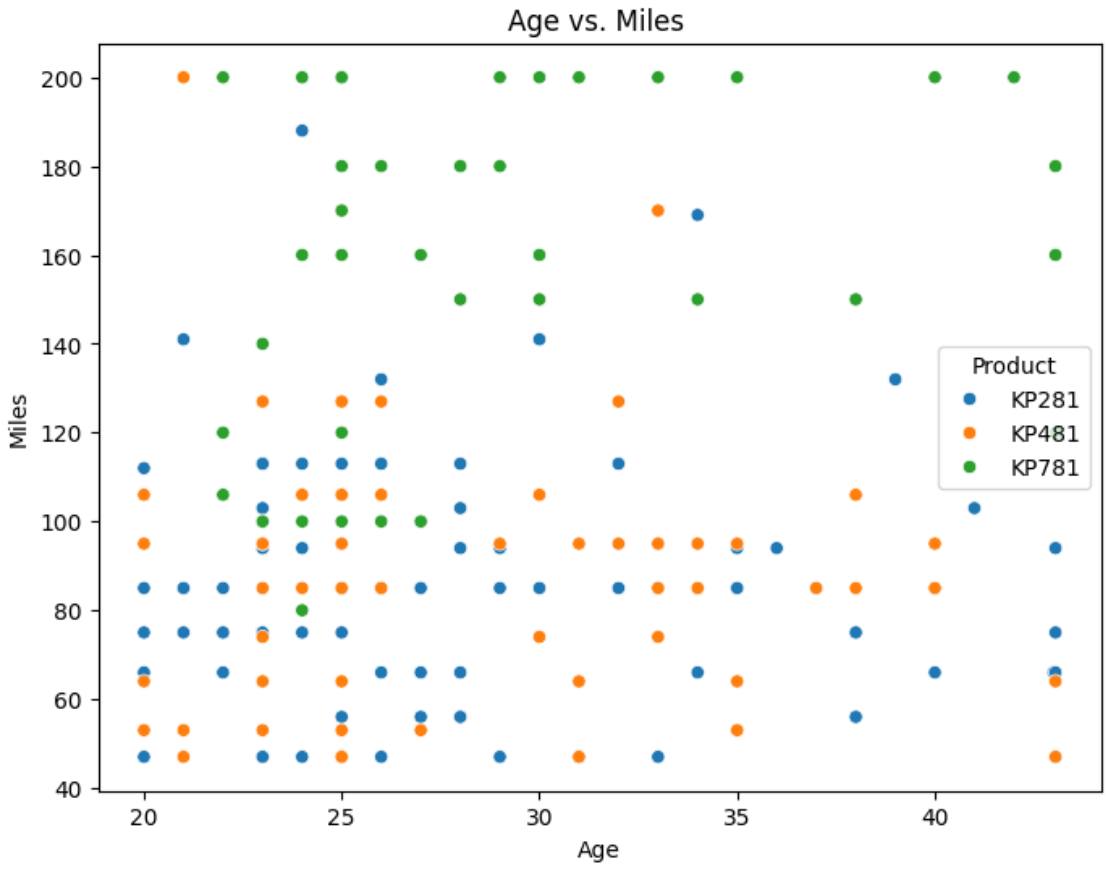
[ ]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='Education_clipped', data=df,hue='Product')
plt.title('Relationship between Product Purchased & Education ')
plt.xlabel('Education')
plt.ylabel('Count of Product purchased')
plt.show()
```



Insights:

- People with 18 years of education are more likely to prefer advanced KP781 trademill.
- Customers with eduction less than 18 years more likely to prefer kp281 and KP481.

[ ]:
```
# Scatter plot for 'Age' vs. 'Miles' for different products
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age_clipped', y='Miles_clipped', data=df,hue='Product')
plt.title('Age vs. Miles ')
plt.xlabel('Age')
plt.ylabel('Miles')
plt.show()
```



Insights:

- Customers with more than 140 miles per week will prefer KP781.
- Customers with lower average miles per week are more likely to buy KP-281 & KP-481.

5.4 4. Representing the Probability

5.4.1 a. Find the marginal probability (what percent of customers have purchased KP281, KP481, or KP781)

Hint: We want you to use the pandas crosstab to find the marginal probability of each product.

[ ]:
```
pd.crosstab(df['Product'], df['Product'], normalize=True)
```

[ ]:
```
Product      KP281     KP481     KP781
Product
KP281     0.444444  0.000000  0.000000
KP481     0.000000  0.333333  0.000000
KP781     0.000000  0.000000  0.222222
```

Insight:

- The probability that a customer will purchase KP281 is 44.4%.
- The probability that a customer will purchase KP481 is 33.3%.
- The probability that a customer will purchase KP781 is 22.3%.

5.4.2 b. Find the probability that the customer buys a product based on each column.

Hint: Based on previous crosstab values you find the probability.

[ ]:
```
pd.crosstab(df['Product'], df['Age'], margins=True)
```

[ ]:
```
Age      18  19  20  21  22  23  24  25  26  27  …  41  42  43  44  45  46  \
Product                                          …
KP281     1   3   2   4   4   8   5   7   7   3  …   1   0   1   1   0   1
KP481     0   1   3   3   0   7   3  11   3   1  …   0   0   0   0   1   0
KP781     0   0   0   0   3   3   4   7   2   3  …   0   1   0   0   1   0
All       1   4   5   7   7  18  12  25  12   7  …   1   1   1   1   2   1

Age      47  48  50  All
Product
KP281     1   0   1   80
KP481     0   1   0   60
KP781     1   1   0   40
All       2   2   1  180

[4 rows x 33 columns]
```

Insight:

- KP281 is the most purchased product (44.4%), with strong sales among customers aged 23–26.
- KP481 (33.3%) peaks at age 25 with 18.3% of its sales.
- KP781 (22.2%) sees consistent sales in the 24–27 age range, with smaller contributions from older customers.
- Ages 23–26 account for the majority of purchases (37.2%, 67 out of 180).
- Sales drop significantly beyond age 30, with minimal engagement from older customers (5.6%, 10 out of 180).

[ ]:
```
pd.crosstab(df['Product'], df['Education'], margins=True)
```

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

Insights:

- KP281 is the top product, with 44.4% of total purchases.
- Most purchases (47.2%) are from customers with education level 16.

- Education level 14 has high engagement (30.6%), while levels 20 and 21 have minimal purchases.

[ ]:
```
pd.crosstab(df['Product'], df['MaritalStatus'], margins=True) #'Product',␣
↪'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage','Fitness', 'Income',␣
↪'Miles']
```

[ ]:
```
MaritalStatus  Partnered  Single  All
Product
KP281                48      32   80
KP481                36      24   60
KP781                23      17   40
All                 107      73  180
```

Insights:

- KP281 is the most purchased product, with 44.4% of total purchases.
- Partnered customers account for 59.4% of purchases (107 out of 180).
- Single customers make up the remaining 40.6% (73 out of 180).
- Across all products, partnered customers consistently buy more than single customers.

[ ]:
```
pd.crosstab(df['Product'], df['Usage'], margins=True)
```

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

Insights:

- KP281 is the most purchased product (44.4%) with peak usage at 3 times (37 purchases).
- KP481 follows (33.3%) with the highest usage also at 3 times (31 purchases).
- KP781 (22.2%) stands out for higher usage levels, with 18 purchases at 4 times and 12 purchases at 5 times.
- Overall, 3 times is the most common usage frequency (38.3%, 69 out of 180).
- Usage drops significantly beyond 5 times, with only 5% of purchases (9 out of 180).

[ ]:
```
pd.crosstab(df['Product'], df['Fitness'], margins=True)
```

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

Insights:

- KP281 has the highest sales (44.4%) and peaks at fitness level 3 (67.5%, 54 out of 80).
- KP481 contributes 33.3% of total sales, also peaking at fitness level 3 (65.0%, 39 out of 60).
- KP781 (22.2%) dominates at higher fitness levels, with 72.5% of its sales at levels 4 and 5 (7 and 29 purchases, respectively).
- Overall, fitness level 3 has the most purchases (53.9%, 97 out of 180).
- Fitness levels 1 and 5 account for the least engagement (18.3%, 33 out of 180 combined).

[ ]:
```
pd.crosstab(df['Product'], df['Income'], margins=True)
```

[ ]:
```
Income   29562  30699  31836  32973  34110  35247  36384  37521  38658  39795  \
Product
KP281        1      1      1      3      2      5      3      2      3      2
KP481        0      0      1      2      3      0      1      0      2      0
KP781        0      0      0      0      0      0      0      0      0      0
All          1      1      2      5      5      5      4      2      5      2

Income   …  88396  89641  90886  92131  95508  95866  99601  103336  104581  \
Product  …
KP281    …      0      0      0      0      0      0      0       0       0
KP481    …      0      0      0      0      0      0      0       0       0
KP781    …      2      2      3      3      1      1      1       1       2
All      …      2      2      3      3      1      1      1       1       2

Income   All
Product
KP281     80
KP481     60
KP781     40
All      180

[4 rows x 63 columns]
```

Insights:

- KP281 has the widest income range but peaks at mid-range income levels (32973–37521).
- KP481 also shows a mid-range focus, with purchases concentrated around 32973–38658.
- KP781 dominates higher income levels, with most purchases (**15%) from incomes 88396–104581.
- Purchases are sparse at the extremes, with minimal activity below 31836 or above 104581.
- Income levels 32973–38658 see the most overall purchases (9.4% of total).

[ ]:
```
pd.crosstab(df['Product'], df['Miles'], margins=True)
```

[ ]:
```
Miles    21  38  42  47  53  56  64  66  74  75  …  180  188  200  212  240  \
Product                                          …
KP281     0   3   0   9   0   6   0  10   0  10  …    0    1    0    0    0
KP481     1   0   4   0   7   0   6   0   3   0  …    0    0    0    1    0
KP781     0   0   0   0   0   0   0   0   0   0  …    6    0    6    0    1
```

```
All       1   3   4   9   7   6   6  10   3  10  …    6    1    6    1    1

Miles    260  280  300  360  All
Product
KP281      0    0    0    0   80
KP481      0    0    0    0   60
KP781      1    1    1    1   40
All        1    1    1    1  180

[4 rows x 38 columns]
```

Insights:

- KP281 is most purchased at mid-range distances like 47, 66, and 75 miles, contributing significantly to its total sales (44.4%).
- KP481 peaks at distances 53 and 64 miles, accounting for a sizable portion of its sales (33.3%).
- KP781 dominates longer distances, with most purchases (**40%) at 180–360 miles.
- Mid-range distances (47–75 miles) see the most overall purchases, while extreme distances (both short and long) have fewer sales.

[ ]:
```
pd.crosstab(df['Product'], df['Gender'], margins=True)
```

[ ]:
```
Gender   Female  Male  All
Product
KP281        40    40   80
KP481        29    31   60
KP781         7    33   40
All          76   104  180
```

Insights:

- KP281 is equally popular among females and males, with 50% of its sales from each gender.
- KP481 shows a slight male preference, with 51.7% of its sales from males.
- KP781 is predominantly purchased by males, contributing 82.5% of its sales.
- Overall, males account for 57.8% of total purchases, while females account for 42.2%.

5.4.3 c. Find the conditional probability that an event occurs given that another event has occurred.

(Example: given that a customer is female, what is the probability she'll purchase a KP481)

Hint: Based on previous crosstab values you find the probability.

[41]:
```
## Conditional Probabilities : Given Gender = Female
# Conditional Probabilty: P(KP281/Female)
print(f'P(KP281/Female):',(round((40/76)*100,2)))

# Conditional Probabilty: P(KP481/Female)
print(f'P(KP481/Female):',(round((29/76)*100,2)))
```

```
# Conditional Probabilty: P(KP781/Female)
print(f'P(KP781/Female):',(round((7/76)*100,2)))
```

```
P(KP281/Female): 52.63
P(KP481/Female): 38.16
P(KP781/Female): 9.21
```

[42]:
```
## Conditional Probabilities : Given Gender = Male
# Conditional Probabilty: P(KP281/Male)
print(f'P(KP281/Male):',(round((40/104)*100,2)))

# Conditional Probabilty: P(KP481/Male)
print(f'P(KP481/Male):',(round((31/104)*100,2)))

# Conditional Probabilty: P(KP781/Male)
print(f'P(KP781/Male):',(round((33/104)*100,2)))
```

```
P(KP281/Male): 38.46
P(KP481/Male): 29.81
P(KP781/Male): 31.73
```

[43]:
```
## Conditional Probabilities : Given customer is partnered
# Conditional Probabilty: P(KP281/customer is partnered )
print(f'P(KP281/customer is partnered ):',(round((48/107)*100,2)))

# Conditional Probabilty: P(KP481/customer is partnered )
print(f'P(KP481/customer is partnered ):',(round((36/107)*100,2)))

# Conditional Probabilty: P(KP781/customer is partnered )
print(f'P(KP781/customer is partnered ):',(round((23/107)*100,2)))
```

```
P(KP281/customer is partnered ): 44.86
P(KP481/customer is partnered ): 33.64
P(KP781/customer is partnered ): 21.5
```

[44]:
```
## Conditional Probabilities : Given customer is single
# Conditional Probabilty: P(KP281/customer is single )
print(f'P(KP281/customer is single):',(round((32/73)*100,2)))

# Conditional Probabilty: P(KP481/customer is single )
print(f'P(KP481/customer is single ):',(round((24/73)*100,2)))

# Conditional Probabilty: P(KP781/customer is single )
print(f'P(KP781/customer is single ):',(round((17/73)*100,2)))
```

```
P(KP281/customer is single): 43.84
P(KP481/customer is single ): 32.88
P(KP781/customer is single ): 23.29
```

[46]:
```
## Conditional Probabilities : Given customer is moderately fit
# Conditional Probabilty: P(KP281/customer is moderately fit )
print(f'P(KP281/customer is moderately fit ):',(round((54/97)*100,2)))
```

```
P(KP281/customer is moderately fit ): 55.67
```

[45]:
```
## Conditional Probabilities : customer is extremely fit
# Conditional Probabilty: P(KP281/customer is extremely fit )
print(f'P(KP281/customer is extremely fit):',(round((29/31)*100,2)))
```

```
P(KP281/customer is extremely fit): 93.55
```

Insights:

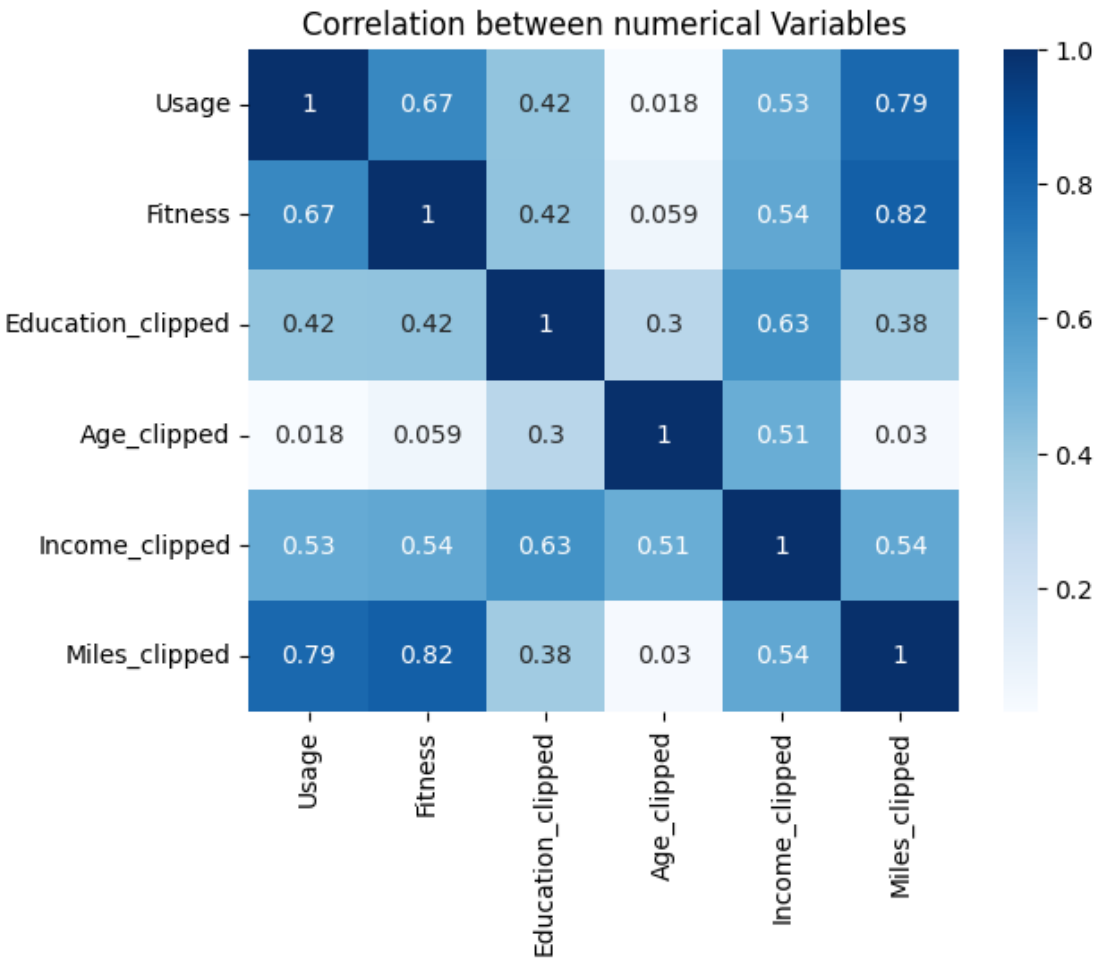
- Given that a customer is female, the probability that she will buy KP281 is higher, 52.6% (40/76), than the probability of her buying KP781, 9.2% (7/76).
- Given that a customer is male, the probability that he will buy KP281, 38.5% (40/104), is little higher compared to KP481 or KP781 which is almost same, 29.8% (31/104) and 31.7% (33/104) respectively.
- Given that a customer is partnered, the probability of he/she buying KP281 is 44.9% (48/107), KP481 is 33.6% (36/107)) and KP781 is 21.5% (23/107).
- Given that a customer is single, the probability of he/she buying KP281 is 43.8% (32/73), KP481 is 32.9% (24/73)) and KP781 is 23.3% (17/73).
- Given that a customer is moderately fit, the probability of he/she buying KP281 is higher, 55.7% (54/97).
- Given that a customer is extremely fit, the probability of he/she buying KP781 is higher, 93.5% (29/31).

5.5 5. Check the correlation among different factors

5.5.1 a. Find the correlation between the given features in the table.

Hint: We want you can use the heatmap and corr function to find the correlation between the variables

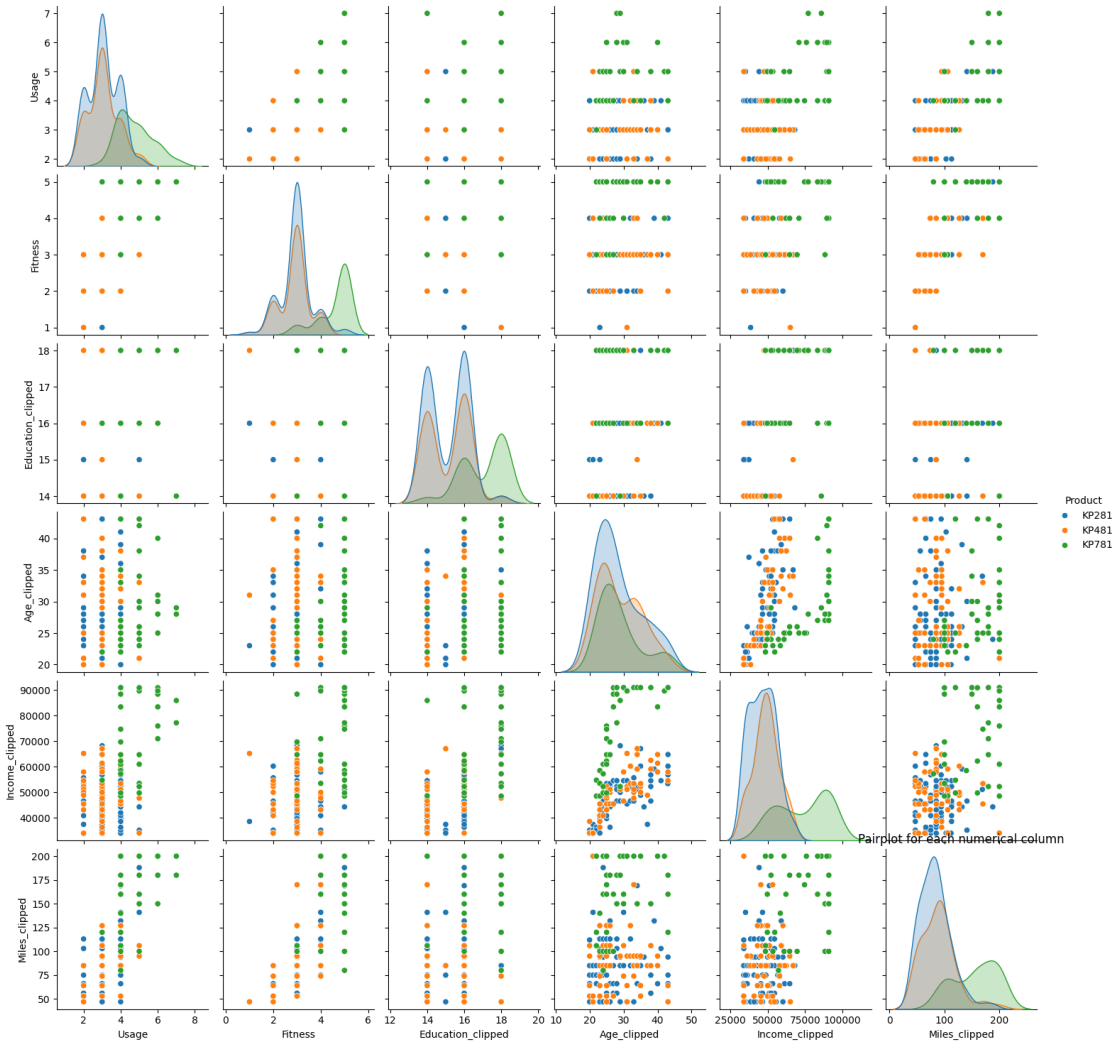
[47]:
```
df=df.drop(['Age','Education','Income','Miles'],axis=1)

corr=df.corr(numeric_only=True)
sns.heatmap(corr,cmap="Blues",annot=True)
plt.title('Correlation between numerical Variables')
plt.show()
```



Insights:

- The correlation between avg miles of running per week and fitness is 0.82
- The correlation between avg miles of running per week and avg no of times the customer use the trademill per week is 0.79.
- From the given dataset, it can be observed that Fitness and Miles are highly correlated followed by Usage and Miles. This is expected as fit people tend to use the treadmill more often and run more miles. On the other hand, Age seems to be unrelated to Usage, Miles and Fitness and therby we can conclude that fitness can be achieved at any age

[48]:
```
sns.pairplot(data=df,hue='Product')
plt.title('Pairplot for each numerical column')
plt.show()
```



5.6 6. Customer profiling and recommendation

5.6.1 a. Make customer profilings for each and every product.

Hint: We want you to find at What age, gender, and income group but product the KP281

[50]:
```
kp281_df = df[df['Product']=='KP281']
kp481_df = df[df['Product']=='KP481']
kp781_df = df[df['Product']=='KP781']
print('Mean of KP281 features :\n', kp281_df.describe().loc['mean'])
print('\nMean of KP481 features :\n', kp481_df.describe().loc['mean'])
print('\nMean of KP781 features :\n', kp781_df.describe().loc['mean'])
```

```
Mean of KP281 features :
 Usage                     3.08750
Fitness                   2.96250
```

```
Education_clipped       15.12500
Age_clipped             28.42750
Income_clipped       46584.31125
Miles_clipped           83.12500
Name: mean, dtype: float64

Mean of KP481 features :
 Usage                   3.066667
Fitness                  2.900000
Education_clipped       15.183333
Age_clipped             28.801667
Income_clipped       49046.607500
Miles_clipped           88.500000
Name: mean, dtype: float64

Mean of KP781 features :
 Usage                   4.77500
Fitness                  4.62500
Education_clipped       17.05000
Age_clipped             28.82875
Income_clipped       73908.28125
Miles_clipped          155.90000
Name: mean, dtype: float64
```

Insights:

For KP281:

```
Age: Prefered by customers of all age.
Gender: Prefered by both male and female customers equally.
Education: Mostly prefered by customers who have completed less than 16 years of education.
MaritalStatus: Mostly Prefered by partnered customers than single customers.
Usage: Prefered by customers who would use the treadmill for less than 4 times/week
Income: Prefered by low income(46,000 dollars average income) customers.
Fitness: Mostly prefered by customers with fitness level less than 3.
Miles: Mostly prefered by customers who expect to walk/run 82 miles/week on average.
```

For KP481:

```
Age: Prefered by customers of all age.
Gender: Prefered by both male and female customers equally.
Education: Mostly prefered by customers who have completed less than 16 years of education.
MaritalStatus: Mostly Prefered by partnered customers than single customers.
Usage: Prefered by customers who would use the treadmill for less than 4 times/week
Income: Prefered by low income(49,000 dollars average income) customers.
Fitness: Mostly prefered by customers with fitness level less than 3.
Miles: Mostly prefered by customers who expect to walk/run 88 miles/week on average.
```

For KP781:

```
Age: Prefered by customers of all age.
```

```
Gender: Mostly prefered by male customers.
Education: Mostly prefered by customers who have completed greater than 16 years of education.
MaritalStatus: Mostly Prefered by partnered customers than single customers.
Usage: Prefered by customers who would use the treadmill for greater than 4 times/week
Income: Mostly prefered by high income(75,000 dollars average income) customers.
Fitness: Mostly prefered by customers with fitness level 3 and above.
Miles: Mostly prefered by customers who expect to walk/run 167 miles/week on average.
```

5.6.2 b. Write a detailed recommendation from the analysis that you have done.

Insights and Recommendations:

1. **KP281 & KP481**:
   - **Target Audience**: Customers across all **ages, genders, marital statuses**, and those with **low to medium fitness levels** and **low to medium incomes**.
   - **Strategy**:
     - Maintain availability for the general audience.
     - **Upsell opportunity**: Target **high-income customers with low to medium fitness levels**. Use fitness incentives to transition them toward **KP781** as they improve their fitness levels and overcome cost concerns.
2. **KP781**:
   - **Target Audience**:
     - **High fitness, high income males** (current primary buyers).
     - **High-income females** (underutilized segment).
     - **High fitness, low-income individuals** (untapped potential).
   - **Strategy**:
     - **For high-income females**: Create gender-inclusive marketing campaigns highlighting advanced features, performance, and luxury.
     - **For high fitness, low-income customers**: Provide **easy financing options**, such as **0% EMI** or a **subscription-based model**, to make the product more accessible.
     - Reinforce the product's image as aspirational and worth investing in.

6 Recommendations:

- Product KP281 has the highest purchase frequency among customers, followed by KP481 and KP781. Consider focusing on promoting these products further to maximize revenue.
- Product KP781 has a significant customer base in terms of total revenue. Invest in strategies to maintain and enhance its popularity.
- Targeted marketing efforts should be directed towards males and partnered customers, as they are more likely to purchase fitness products.
- Focus on tailoring products and campaigns to different age groups, as preferences and purchasing patterns vary across age categories.
- Customers with Education level 14 and 16 have the highest purchase frequencies. Create marketing content that resonates with these education levels and addresses their specific needs.
- Fitness level 3 is the most common among customers and correlates with higher purchase rates. Develop products that cater to customers with fitness level 3.

- Customers using fitness products 3 times a week show the highest purchase frequency. Consider offering incentives or discounts to encourage consistent product usage.
- Products are purchased across different income levels, indicating a diverse customer base. However, consider adjusting pricing strategies based on income brackets to cater to different customer segments.
- Utilize the insights gained from bivariate analyses and pair plots to create targeted marketing campaigns for specific customer segments. This personalized approach can improve customer engagement.
- The majority of customers fall into the lower income brackets. Offer a variety of price points and consider introducing entry-level products to cater to this segment.
- Consider introducing more products that appeal to both genders. While there are differences in preferences, there's an opportunity to expand product offerings for greater inclusivity.
- Marital status and age influence purchasing behavior. Leverage these insights to design products and marketing campaigns that align with the preferences of partnered and single customers across different age groups.