aerofit-analysis

July 19, 2024

Business Problem

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.

The dataset has the following features:

- 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 shape.
- Miles: The average number of miles the customer expects to walk/run each week

Objectives of the Project

- Perform EDA on the given dataset and find insights.
- Provide Useful Insights and Business recommendations that can help the business to grow.

Importing libraries

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import gdown as gd
```

[ ]:
```
!gdown 1XGWPoolMLRbptqwjtPwEQ3wvPMsMBkSO
```

```
Downloading…
From: https://drive.google.com/uc?id=1XGWPoolMLRbptqwjtPwEQ3wvPMsMBkSO
To: /content/aerofit_treadmill (2).csv
100% 7.28k/7.28k [00:00<00:00, 17.6MB/s]
```

[ ]:
```
df = pd.read_csv('aerofit_treadmill (2).csv')
```

Basic Obervation

[ ]:
```
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

These are the first 5 rows of the dataset.

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

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

Aerofit dataset, there are 180 rows and 9 columns.

[ ]:
```
df.ndim
```

[ ]:
```
2
```

[ ]:
```
df.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	95508	180

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

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

[ ]:
```
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
```

The data types include object (for text/string data) and int64 (for integer data).

[ ]:
```
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

[ ]:
```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
outliers = (df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))
difference = df.mean() - df.median()
print("Difference between mean and median: \n" )
print(difference)
```

```
Difference between mean and median:

Age             2.788889
Education      -0.427778
Usage           0.455556
Fitness         0.311111
Income       3123.077778
Miles           9.194444
dtype: float64
```

[ ]:
```
df.describe(include=object)
```

[ ]:

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

2.Data Cleaning

[ ]:
```
df.isnull().sum()
```

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

There are no missing values in this dataset.

3. Non-Graphical Analysis

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

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

These numbers represent the quantities sold of each product (e.g., 80 units of KP281, 60 units of KP481, and 40 units of KP781).

[ ]:
```
KP281=df.loc[df['Product']=='KP281']
KP481=df.loc[df['Product']=='KP481']
KP781=df.loc[df['Product']=='KP781']
```

[ ]:
```
df['Gender'].value_counts()
```

[ ]:
```
Male      104
Female     76
Name: Gender, dtype: int64
```

There are more Males in the data than Females.

[ ]:
```
male_customers = df[df['Gender'] == 'Male']
total_male_customers = len(male_customers)
kp781_purchased_by_male = len(male_customers[male_customers['Product'] ==␣
 ↪'KP781'])
probability = (kp781_purchased_by_male / total_male_customers)*100
probability
```

[ ]:
```
31.73076923076923
```

The probability of a male customer purchasing the product KP781 is approximately 31.73%.

[ ]:
```
df['MaritalStatus'].value_counts()
```

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

More Partnered persons are there in the data.

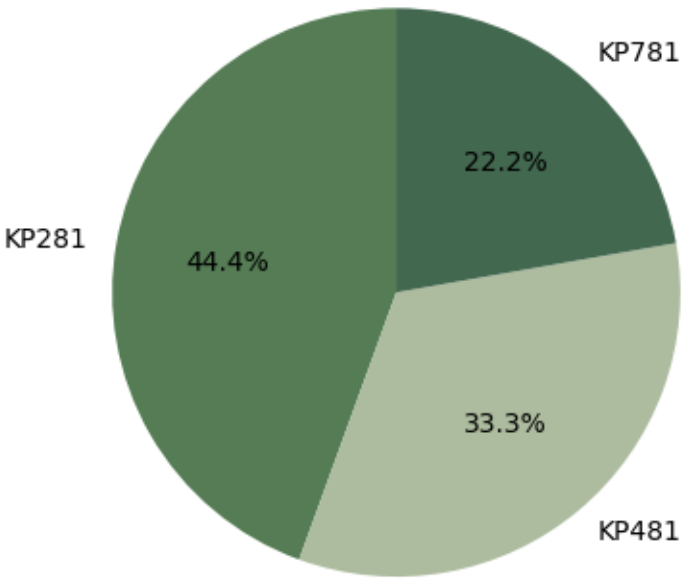
[ ]:
```
marginal_prob = pd.crosstab(df['Product'], columns='count', normalize=True) *␣
 ↪100
marginal_prob
```

[ ]:
```
col_0        count
Product
KP281    44.444444
KP481    33.333333
KP781    22.222222
```

4. Visual Analysis - Univariate & Bivariate

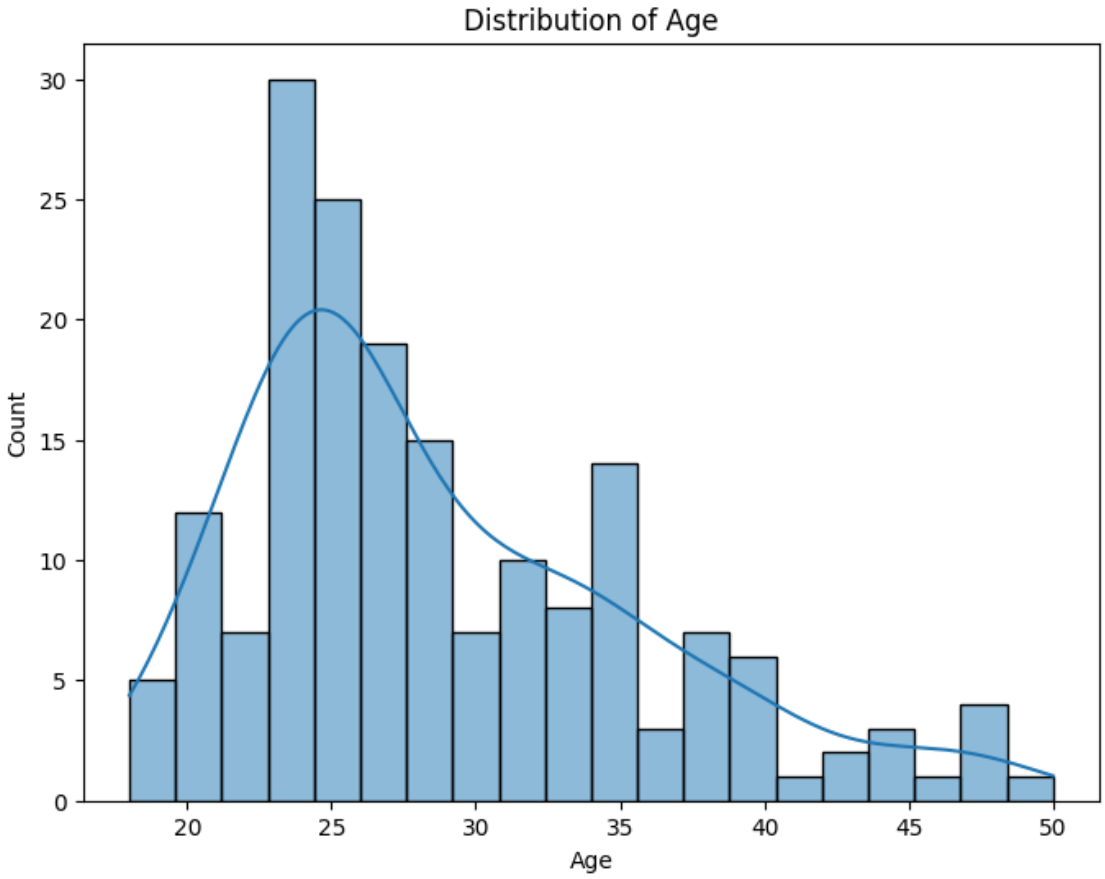
```
[ ]: KP781=df.loc[df['Product']=='KP781']
     KP281=df.loc[df['Product']=='KP281']
     KP481=df.loc[df['Product']=='KP481']
     labels=['KP281','KP481','KP781']
     sizes=[len(KP281),len(KP481),len(KP781)]
     colors = ['#557C55','#ADBC9F','#436850',]
     plt.pie(sizes, labels=labels, autopct="%1.1f%%", colors=colors, startangle=90)
```

```
[ ]: ([<matplotlib.patches.Wedge at 0x78642862e620>,
      <matplotlib.patches.Wedge at 0x78642862e560>,
      <matplotlib.patches.Wedge at 0x78642862eda0>],
     [Text(-1.0832885303005317, 0.19101298416420232, 'KP281'),
      Text(0.7070664144854606, -0.8426488506529128, 'KP481'),
      Text(0.707066296143735, 0.8426489499534076, 'KP781')],
     [Text(-0.5908846528911991, 0.10418890045320126, '44.4%'),
      Text(0.3856725897193421, -0.4596266458106797, '33.3%'),
      Text(0.38567252516930994, 0.45962669997458583, '22.2%')])
```

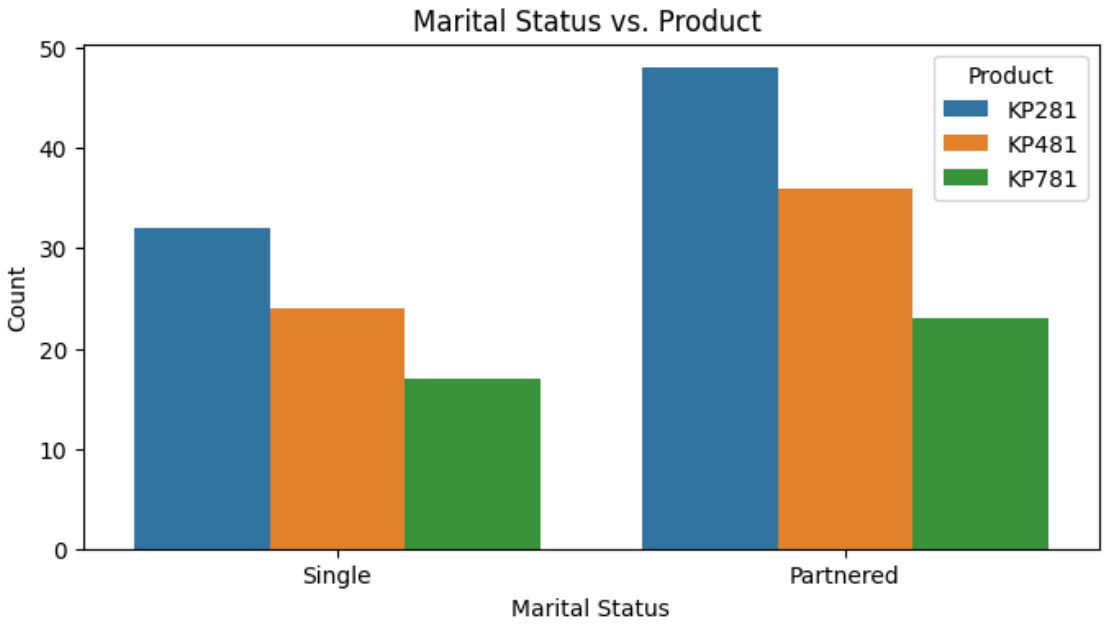


- 44.44% of the customers have purchased KP281 product.
- 33.33% of the customers have purchased KP481 product.
- 22.22% of the customers have purchased KP781 product.
- KP281 is the most frequent product.

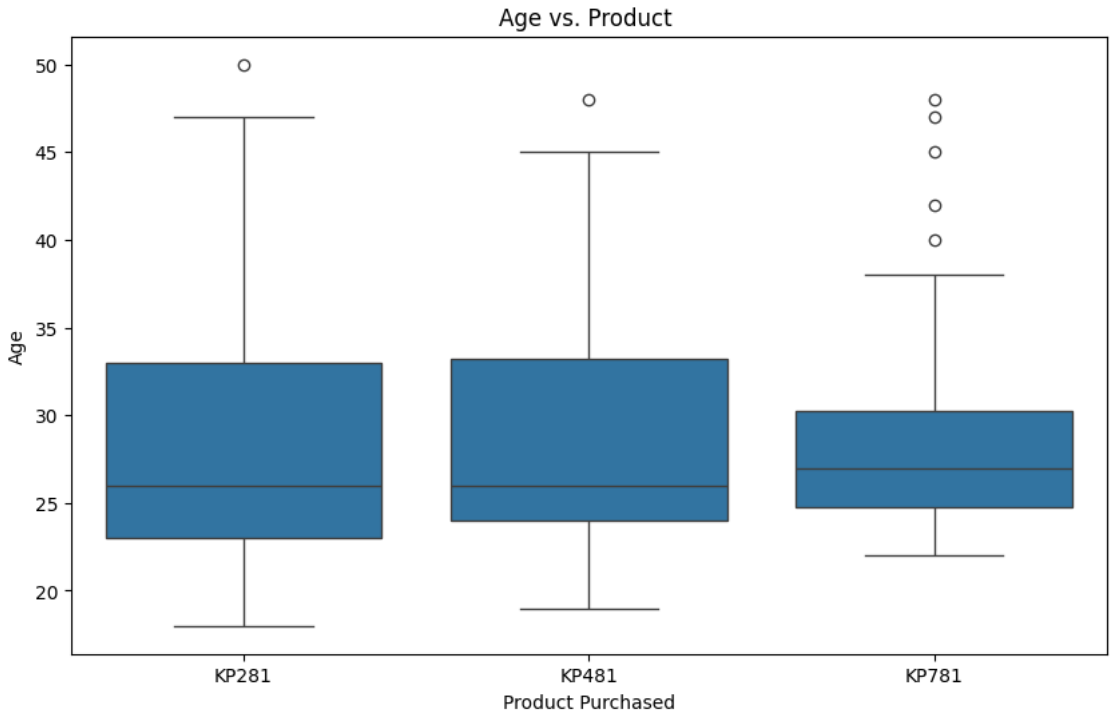
[ ]:
```
# Univariate analysis
plt.figure(figsize=(8, 6))
sns.histplot(df['Age'], bins=20, kde=True)
plt.title('Distribution of Age')
plt.show()
```



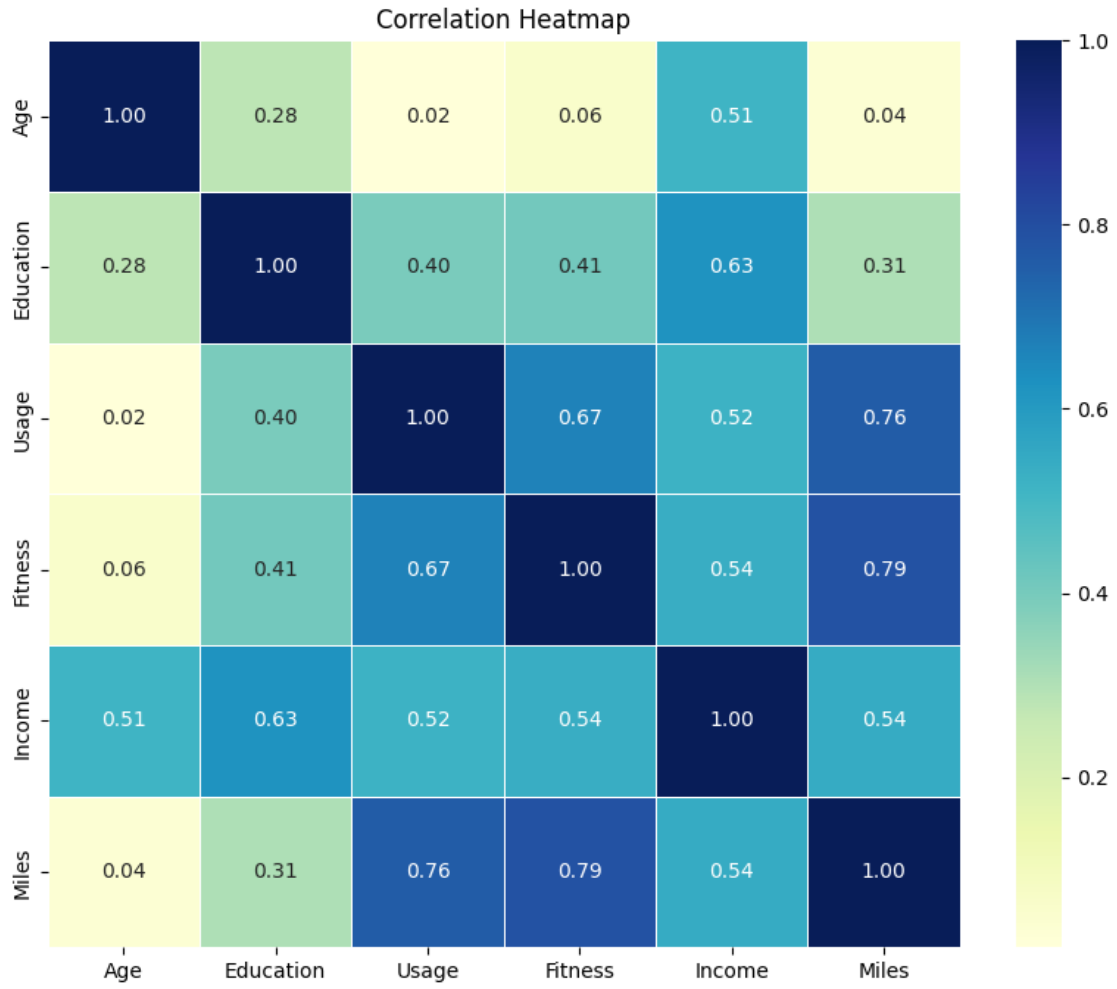
[ ]:
```
plt.figure(figsize=(8, 4))
sns.countplot(x='MaritalStatus', hue='Product', data=df)
plt.title('Marital Status vs. Product ')
plt.xlabel('Marital Status')
plt.ylabel('Count')
plt.show()
```



[ ]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Product', y='Age', data=df)
plt.title('Age vs. Product ')
plt.xlabel('Product Purchased')
plt.ylabel('Age')
plt.show()
```



[ ]:
```
corr = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='YlGnBu', fmt=".2f", linewidths=.5)
plt.title('Correlation Heatmap')
plt.show()
```



[ ]:
```
df['Age_Group'] = pd.cut(df['Age'], bins=[18, 35,55,100], labels=[ 'Middle-aged␣
↪Adults','Young Adults', 'Seniors'])
df['Usage_Category'] = pd.cut(df['Usage'], bins=[0, 2, 4, 10], labels=['Low␣
↪Usage', 'Moderate Usage', 'High Usage'])
df['Fitness_Level'] = pd.cut(df['Fitness'], bins=[1, 2, 4, 5], labels=['Low␣
↪Fitness', 'Moderate Fitness', 'High Fitness'])
df['Miles_Walked'] = pd.cut(df['Miles'], bins=[0, 50, 100, 1000], labels=['Low␣
↪Mileage', 'Moderate Mileage', 'High Mileage'])
```

[ ]:
```
df
```

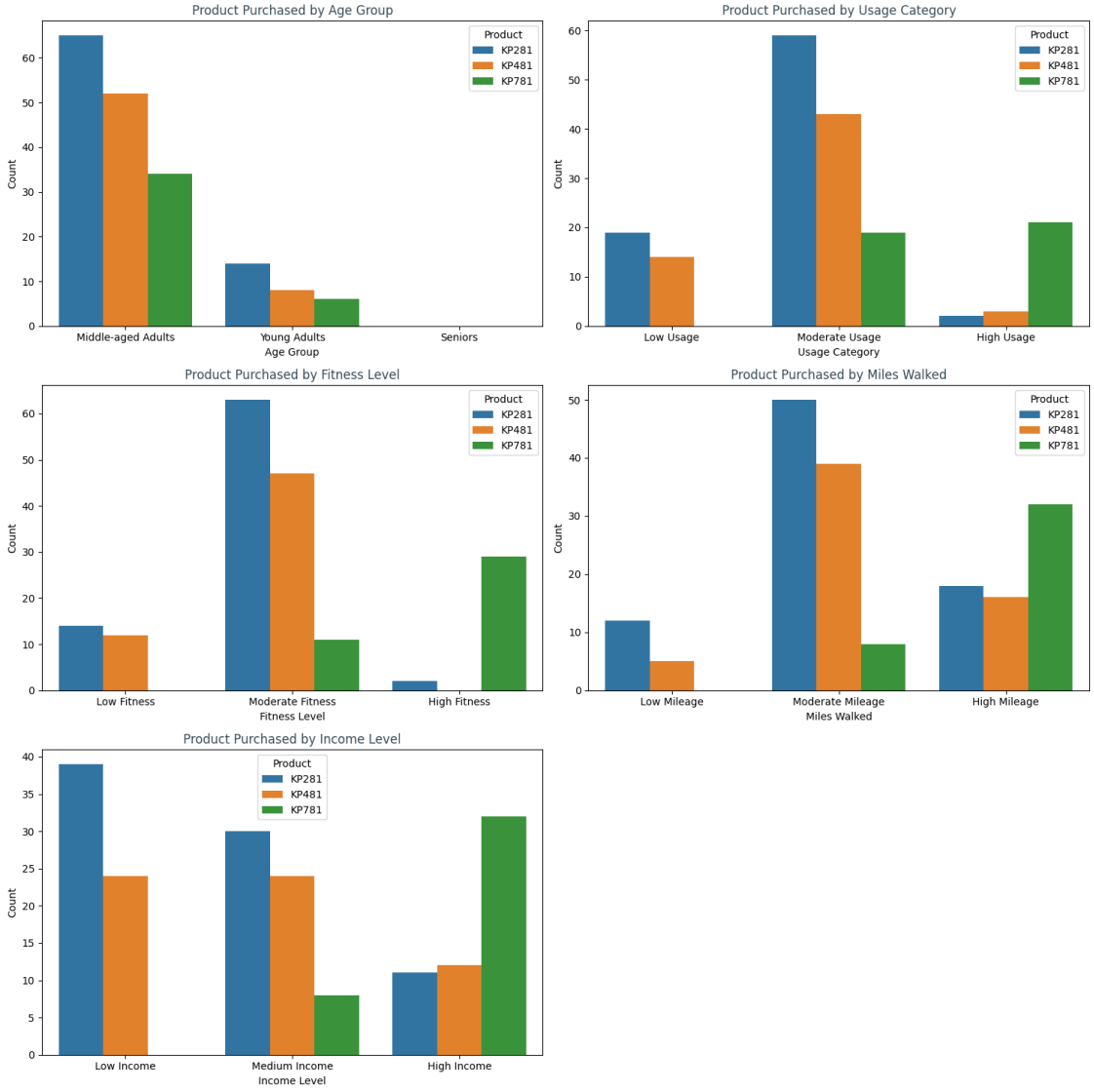
[ ]:

	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	

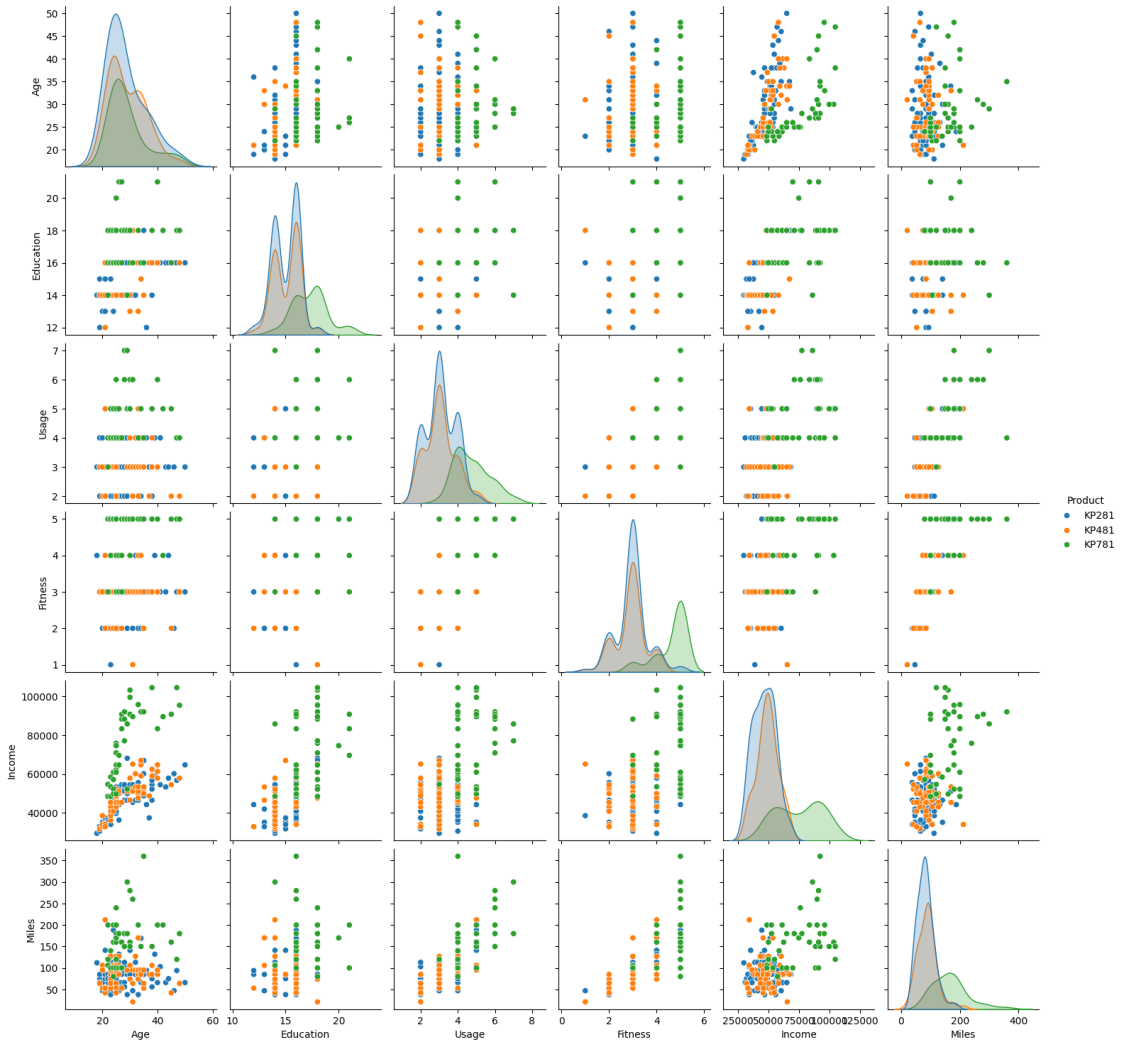
```
4     KP281   20    Male          13     Partnered      4         2   35247
..      …   …     …         …           …    …       …      …
175   KP781   40    Male          21         Single      6         5   83416
176   KP781   42    Male          18         Single      5         4   89641
177   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  Usage_Category     Fitness_Level      Miles_Walked Income_Level  \
0      112  Moderate Usage  Moderate Fitness      High Mileage   Low Income
1       75       Low Usage  Moderate Fitness  Moderate Mileage   Low Income
2       66  Moderate Usage  Moderate Fitness  Moderate Mileage   Low Income
3       85  Moderate Usage  Moderate Fitness  Moderate Mileage   Low Income
4       47  Moderate Usage       Low Fitness       Low Mileage   Low Income
..     …              …                 …                 …           …
175    200      High Usage      High Fitness      High Mileage  High Income
176    200      High Usage  Moderate Fitness      High Mileage  High Income
177    160      High Usage      High Fitness      High Mileage  High Income
178    120  Moderate Usage      High Fitness      High Mileage  High Income
179    180  Moderate Usage      High Fitness      High Mileage  High Income

              Age_Group
0                   NaN
1          Young Adults
2          Young Adults
3          Young Adults
4          Young Adults
..                  …
175  Middle-aged Adults
176  Middle-aged Adults
177  Middle-aged Adults
178  Middle-aged Adults
179  Middle-aged Adults

[180 rows x 14 columns]
```

[ ]:
```
fig, axes = plt.subplots(3, 2, figsize=(15, 15))

# Countplot for Age Group by Product Purchased
sns.countplot(ax=axes[0, 0], x='Age_Group', hue='Product', data=df)
axes[0, 0].set_title('Product Purchased by Age Group',color='#344955')
axes[0, 0].set_xlabel('Age Group')
axes[0, 0].set_ylabel('Count')

# Countplot for Usage Category by Product Purchased
sns.countplot(ax=axes[0, 1], x='Usage_Category', hue='Product', data=df)
axes[0, 1].set_title('Product Purchased by Usage Category',color='#344955')
```

```
axes[0, 1].set_xlabel('Usage Category')
axes[0, 1].set_ylabel('Count')

# Countplot for Fitness Level by Product Purchased
sns.countplot(ax=axes[1, 0], x='Fitness_Level', hue='Product', data=df)
axes[1, 0].set_title('Product Purchased by Fitness Level',color='#344955')
axes[1, 0].set_xlabel('Fitness Level')
axes[1, 0].set_ylabel('Count')

# Countplot for Miles Walked by Product Purchased
sns.countplot(ax=axes[1, 1], x='Miles_Walked', hue='Product', data=df)
axes[1, 1].set_title('Product Purchased by Miles Walked',color='#344955')
axes[1, 1].set_xlabel('Miles Walked')
axes[1, 1].set_ylabel('Count')

# Countplot for Income Level by Product Purchased
sns.countplot(ax=axes[2, 0], x='Income_Level', hue='Product', data=df)
axes[2, 0].set_title('Product Purchased by Income Level',color='#344955')
axes[2, 0].set_xlabel('Income Level')
axes[2, 0].set_ylabel('Count')

# Remove the empty subplot
fig.delaxes(axes[2, 1])

plt.tight_layout()
plt.show()
```

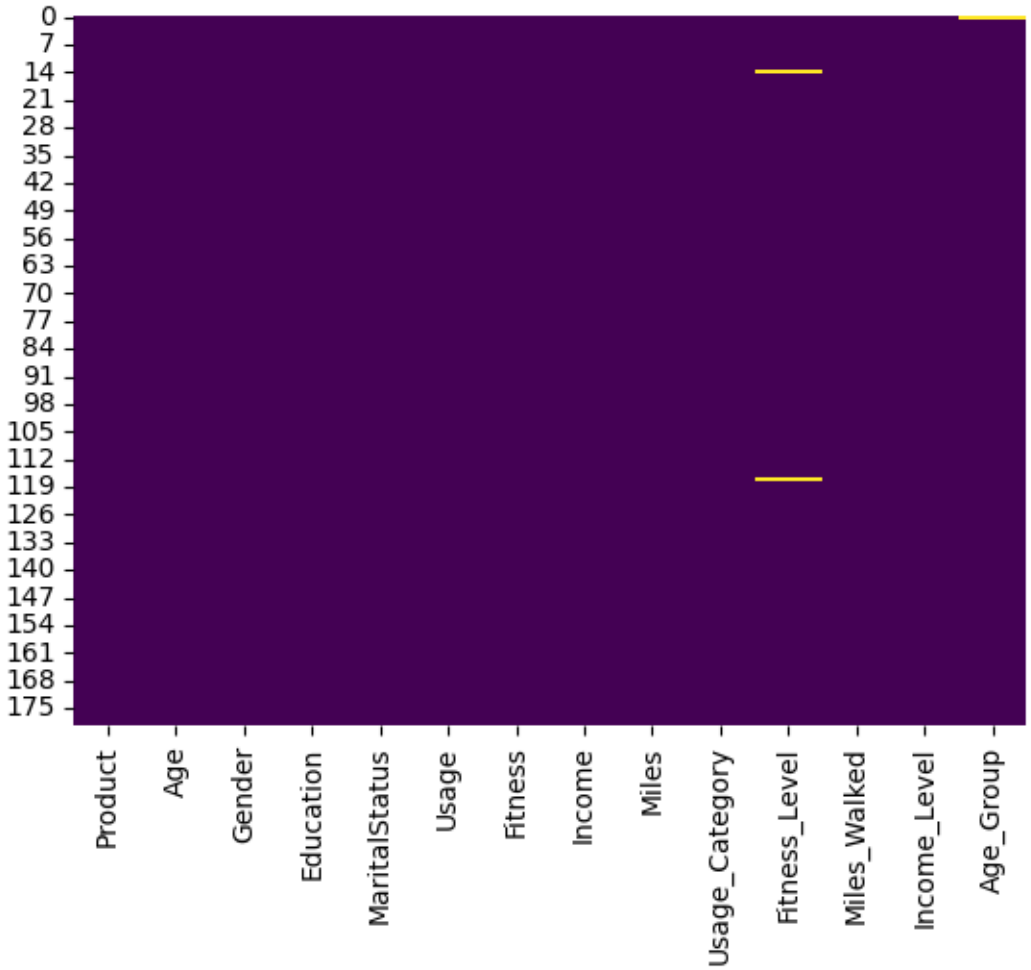


[ ]:
```
sns.pairplot(df, hue='Product')
plt.show()
```



4. Missing Value & Outlier Detection

[ ]:
```
sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.show()
```



6.Insights based on Non-Graphical and Visual Analysis

- Product_Purchased: Categorical with three possible values: KP281, KP481, or KP781.
- Age: Ranges from 18 to 65 years, indicating a diverse age range among customers.
- Gender: Two categories: Male and Female.
- Education: Ranges from 12 to 21 years, representing the education level of customers.
- MaritalStatus: Two categories: Single and Partnered.
- Usage: Ranges from 2 to 7 times per week, indicating varying levels of treadmill usage among customers.
- Income: Ranges from $50,000 to $100,000 annually, showing a range of income levels among customers.
- Fitness: Ranges from 2 to 5, representing different self-rated fitness levels among customers.
- Miles: Ranges from 50 to 110 miles per week, indicating varying expectations regarding treadmill usage among customers.

8. Recommendations

- Targeted Marketing: Understanding the age distribution can help in targeting specific age

groups for marketing campaigns.

- Product Bundling: Analyzing correlations between different products purchased can inform bundling strategies.
- Customer Segmentation: Based on demographics like age and marital status, segment customers for tailored marketing strategies.
- for example, if we find that younger customers with lower income prefer the entry-level treadmill, we can suggest targeting this demographic with promotions for the entry-level treadmill.
- Similarly, if we find that customers who rate their fitness level highly prefer the advanced treadmill, we can suggest targeting fitness enthusiasts with promotions for the advanced treadmill.