Business case study -Aerofit

In [1]:

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

In [2]:

```
df = pd.read_csv('aerofit_treadmill.csv')
```

In [3]:

```
df
```

Out[3]:

	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
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

180 rows × 9 columns

In [4]:

```
df.head()
```

Out[4]:

	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

In [5]:

```
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
```

In [6]:

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

Out[6]:

```
0
```

In [7]:

```
df.dtypes
```

Out[7]:

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

In [8]:

```
df.describe()
```

Out[8]:

	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

In [9]:

```
# Total number of unique Product ids
df['Product'].nunique()
```

Out[9]:

```
3
```

In [10]:

```
# unique list of product ids
df['Product'].unique().tolist()
```

Out[10]:

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

In [11]:

```
# Total number of unique ages
total_uniq_age = df['Age'].nunique()
total_uniq_age
```

Out[11]:

```
32
```

In [12]:

```
# list of unique ages
df['Age'].unique()
```

Out[12]:

```
array([18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 43, 44, 46, 47, 50, 45, 48, 42],
      dtype=int64)
```

In [13]:

```
# Number of Male and Female customers
df['Gender'].value_counts()
```

Out[13]:

```
Gender
Male      104
Female     76
Name: count, dtype: int64
```

In [14]:

```
# list of unique Educations
df['Education'].unique().tolist()
```

Out[14]:

```
[14, 15, 12, 13, 16, 18, 20, 21]
```

In [15]:

```
#Number of customer againts the rating scale 1 to 5
df['Fitness'].value_counts().sort_index()
```

Out[15]:

```
Fitness
1     2
2    26
3    97
4    24
5    31
Name: count, dtype: int64
```

In [16]:

```
# Number of customers with 3 different product types
df['Product'].value_counts().sort_index()
```

Out[16]:

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

In [17]:

```
# Number of customers counts on Usage
df['Usage'].value_counts().sort_index()
```

Out[17]:

```
Usage
2    33
3    69
4    52
5    17
6     7
7     2
Name: count, dtype: int64
```

In [18]:

```
# Number of Single and Partnered customers
df['MaritalStatus'].value_counts()
```

Out[18]:

```
MaritalStatus
Partnered    107
Single        73
Name: count, dtype: int64
```

Summary

- **‘KP281, KP481, KP781 are the 3 different products**
- **‘Most commonly purchased treadmill product type is KP281**
- **There are 32 unique ages**
- **‘104 Males and 76 Females are in the customers list**
- **‘8 unique set of Educations (14, 15, 12, 13, 16, 18, 20, 21)**
- **‘Highest rated Fitness rating is 3**
- **‘Most customers usage treadmill atleast 3 days per week**
- **‘Majority of the customers who have purchased are Married/Partnered**

In [19]:

```
# Converting Int data type of fitness rating to object data type
df_cat = df
df_cat['Fitness_category'] = df.Fitness
df_cat.head()
```

Out[19]:

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

In [20]:

```
df_cat["Fitness_category"].replace({1:"Poor Shape",
2:"Bad Shape",
3:"Average Shape",
4:"Good Shape",
5:"Excellent Shape"},inplace=True)
df_cat.head()
```

Out[20]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Fitness_category

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

In [21]:

```
df.describe()
```

Out[21]:

	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

- **Mean Age of the given customer dataset is 28.78**
- **Minimum Age of the customer starts from 18 and maximum age is 50**
- **25% of the customers age is 24**
- **75% of the customer age is 33**
- **Average usage per week for a customer is 3 days**
- **Average Fitness rating is 3 with most common fitness rating is 4**
- **Average Income of the purchased customer is around 54K per year**
- **Highest salary recorded for the customer is around 104K per year**
- **Maximum distance covered by the customer in treadmill is 360 miles**
- **Most of the customers cover a distance of 114 miles with an average of 103 - miles**
- **Around 25% of the customer cover an average of 66 miles**

Normalizing

In [22]:

```
# for unique list of products, listed in percentage
sr = df['Product'].value_counts(normalize=True)
stat = sr.map(lambda calc: round(100*calc,2))
stat
```

Out[22]:

```
Product
KP281    44.44
KP481    33.33
KP781    22.22
Name: proportion, dtype: float64
```

- **44.44% of customers bought KP281 product type**
- **33.33% of customers bought KP481 product type**
- **22.22% of customers bought KP781 product type**

In [23]:

```
# Customer Gender statistics listed in % percentage
```

```
gender = df['Gender'].value_counts(normalize=True)
gender_res = gender.map(lambda calc: round(100*calc,2))
gender_res
```

Out[23]:

```
Gender
Male      57.78
Female    42.22
Name: proportion, dtype: float64
```

- **57.78% of customers are Male and 42.22% customers are Female**

In [24]:

```
# Customers Marital Status listed in percentage
marital_status = df['MaritalStatus'].value_counts(normalize=True)
marital_status_res = marital_status.map(lambda calc:round(100*calc,2))
marital_status_res
```

Out[24]:

```
MaritalStatus
Partnered    59.44
Single       40.56
Name: proportion, dtype: float64
```

- **59.44% of customers are Married/Partnered**
- **40.56% of customers are Single**

In [28]:

```
# Customer rating of their fitness (listed in %)
rating = df['Fitness'].value_counts(normalize=True).map(lambda calc:
round(100*calc,2)).reset_index()
rating.rename(columns={'index':'Rating'},inplace=True)
rating
```

Out[28]:

	Fitness	proportion
0	3	53.89
1	5	17.22
2	2	14.44
3	4	13.33
4	1	1.11

- **More than 53% of customers have rated themselves as average in fitness (rated 3)**
- **14% of customers have rated their fitness less than average**
- **Over 17% of customers have peak fitness ratings**

In [29]:

```
# Usage: Number of days used per week (listed in %)
usage = df['Usage'].value_counts(normalize=True).map(lambda calc:
round(100*calc,2)).reset_index()
usage.rename(columns={'index':'DaysPerWeek'},inplace=True)
usage
```

Out[29]:

	Usage	proportion
0	3	38.33

	Usage	proportion
1	4	28.89
2	2	18.33
3	5	9.44
4	6	3.89
5	7	1.11

- **Around 39% of customers use 3 days per week**
- **Less than 2% of customers use 7 days per week**

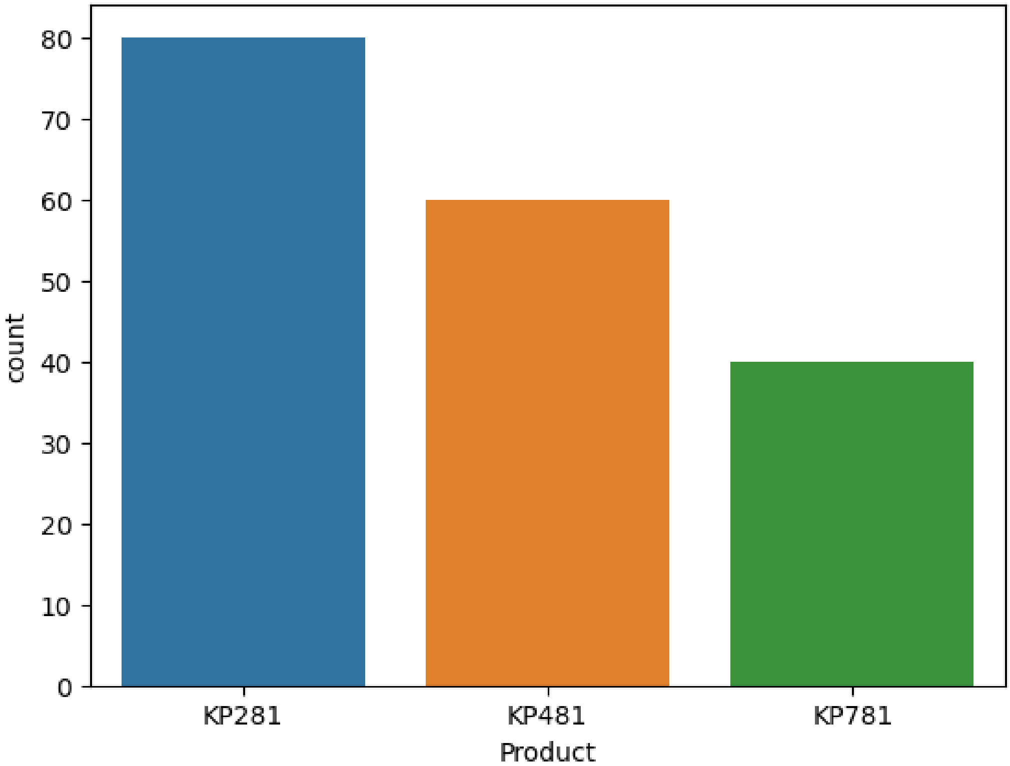
Visual Analysis - Univariate & Bivariate Univariate Analysis

In [30]:

```
# Product Analysis - count plot
sns.countplot(data=df,x='Product')
plt.show
```

Out[30]:

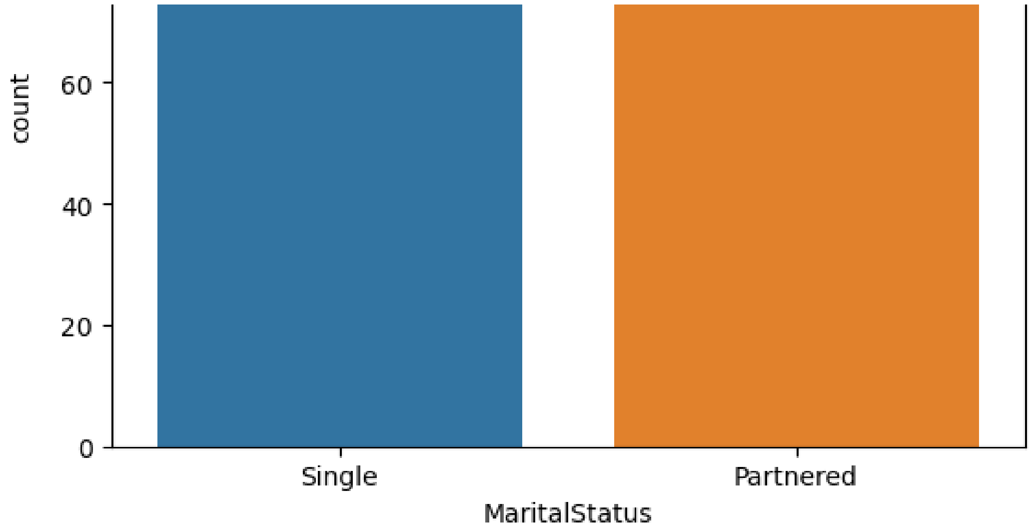
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



- **KP281 is the most commonly purchase product type**
- **KP481 is the second most top product type purchased**
- **KP781 is the least purchased product type**

In [31]:

```
# Marital Status Analysis - Count plot
sns.countplot(data=df,x='MaritalStatus')
plt.show()
```



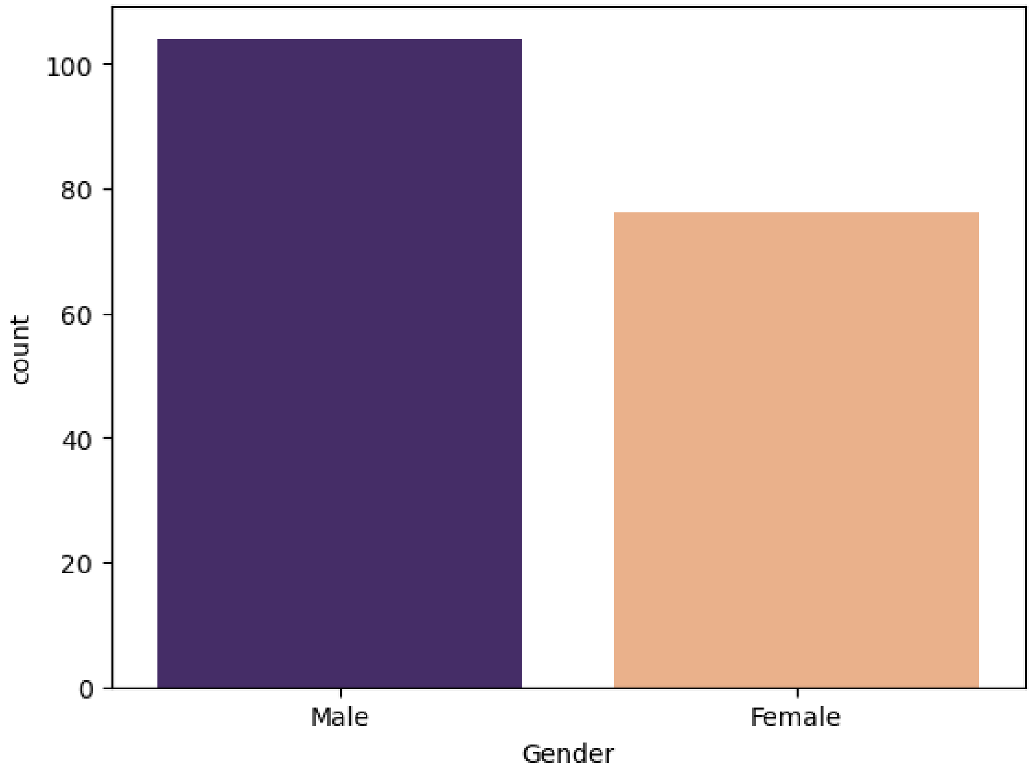
Most products purchased by couples/Married/Partnered customer category

In [32]:

```
# Gender Analysis - Count Plot
sns.countplot(data=df,x='Gender',palette=['#432371',"#FAAE7B"])
plt.show
```

Out[32]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



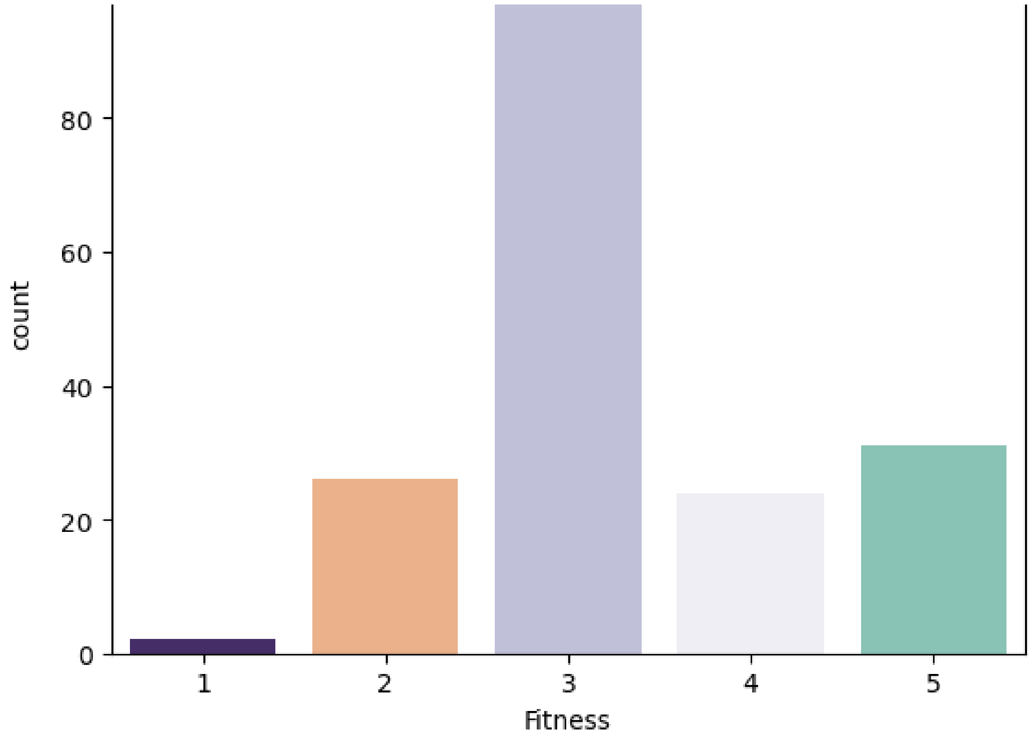
Most products purchased by Males, females are less interested in the product compared to Males

In [36]:

```
# Fitness rating analysis - count plot
sns.countplot(data=df,x='Fitness',palette=['#432371',"#FAAE7B","#bcbddc",
"#efedf5",'#7fcdbb'])
plt.show
```

Out[36]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



More than 90 customers have rated their physical fitness rating as Average. Excellent shape is the second highest rating provided by the customers

In [37]:

```
# Income Analysis - Distplot
sns.distplot(df.Income,rug=True)
plt.show()
```

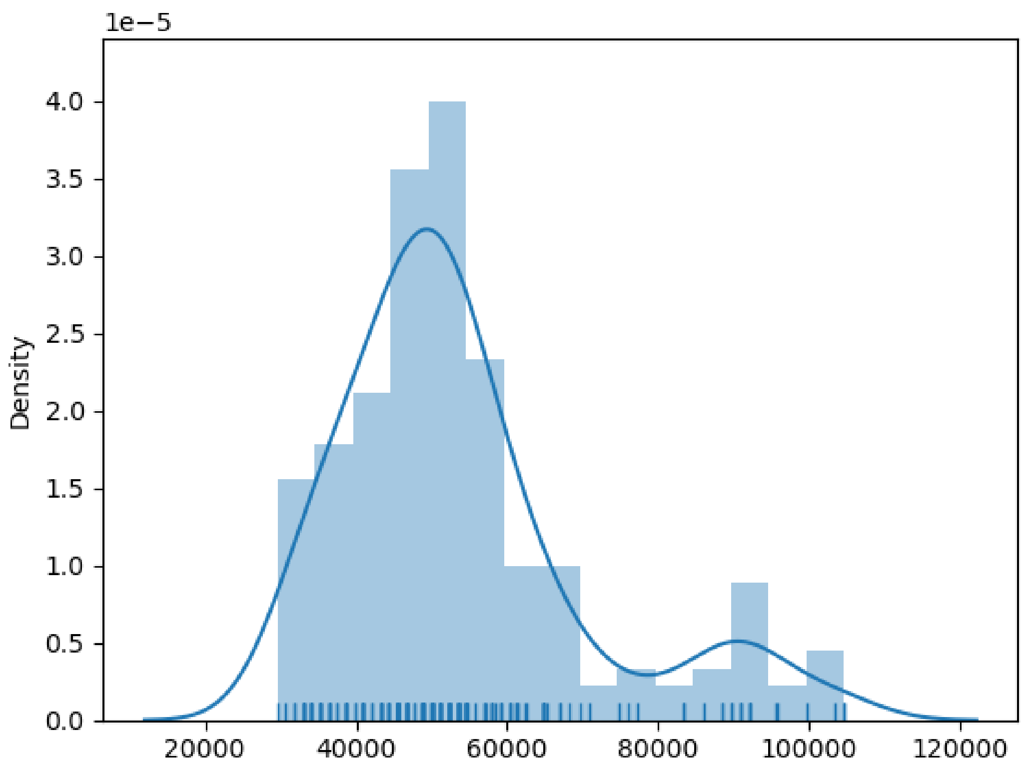
```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_8884\1001525557.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Income,rug=True)
```



Income

- **Most of customers who have purchased the product have a average income between 40K to 60K**
- **Average Income density is over 3.0**

In [38]:

```
# Fitness Rating Analysis - Distplot
sns.distplot(df.Fitness)
plt.show()
```

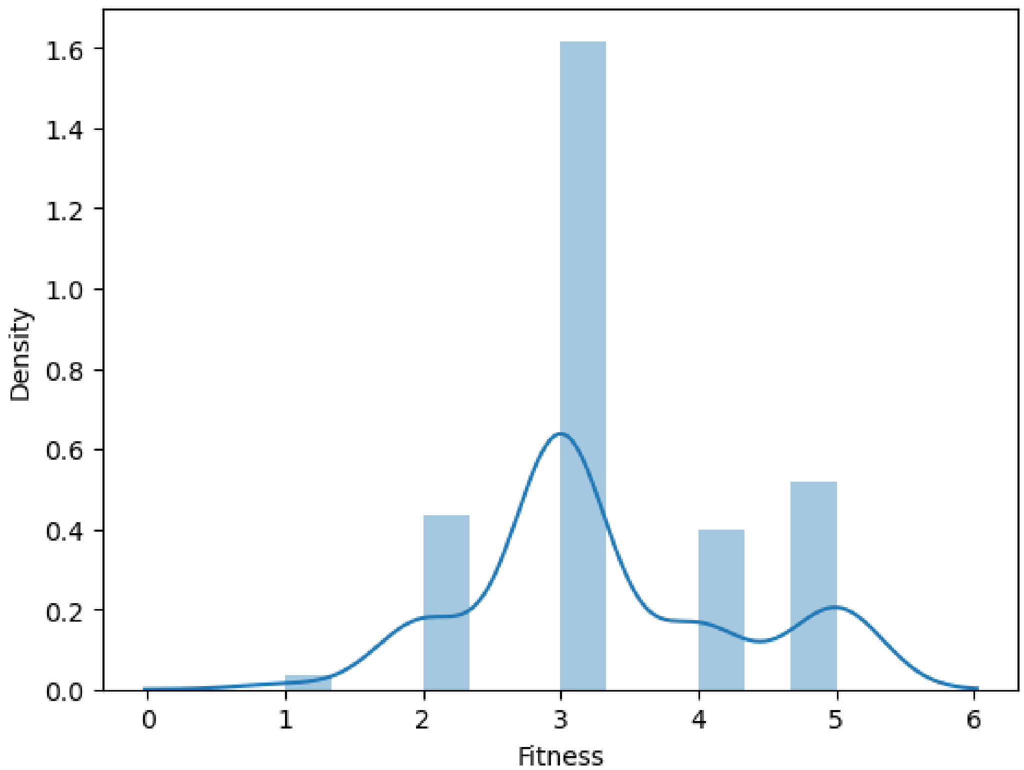
```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_8884\3412044600.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Fitness)
```



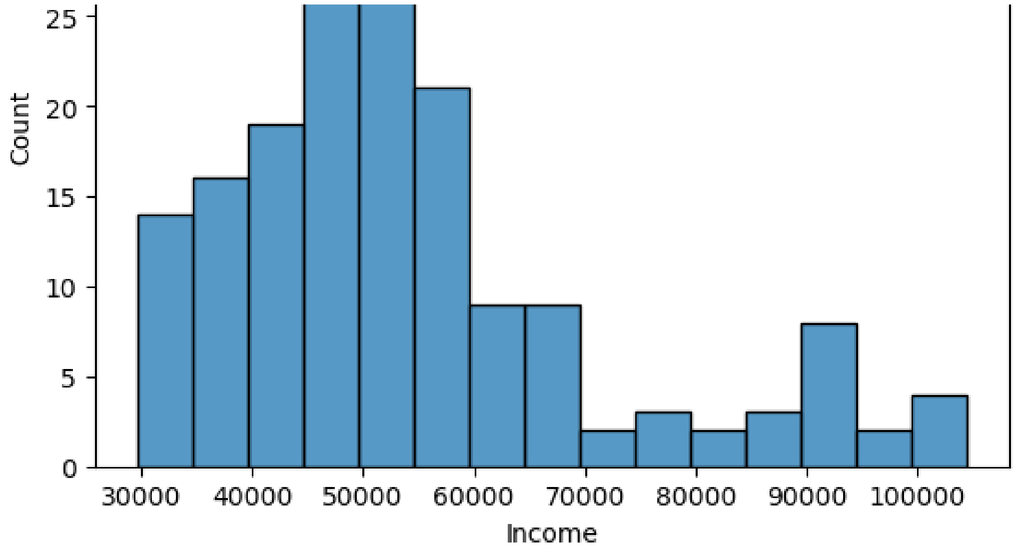
- **Over 1.5 density customer population have rated their physical fitness rating as Average**
- **Second highest customer population density have rated Excellent shape as their fitness rating**

In [39]:

```
# Income Analysis - Histogram
sns.histplot(data=df,x='Income')
```

Out[39]:

```
<Axes: xlabel='Income', ylabel='Count'>
```



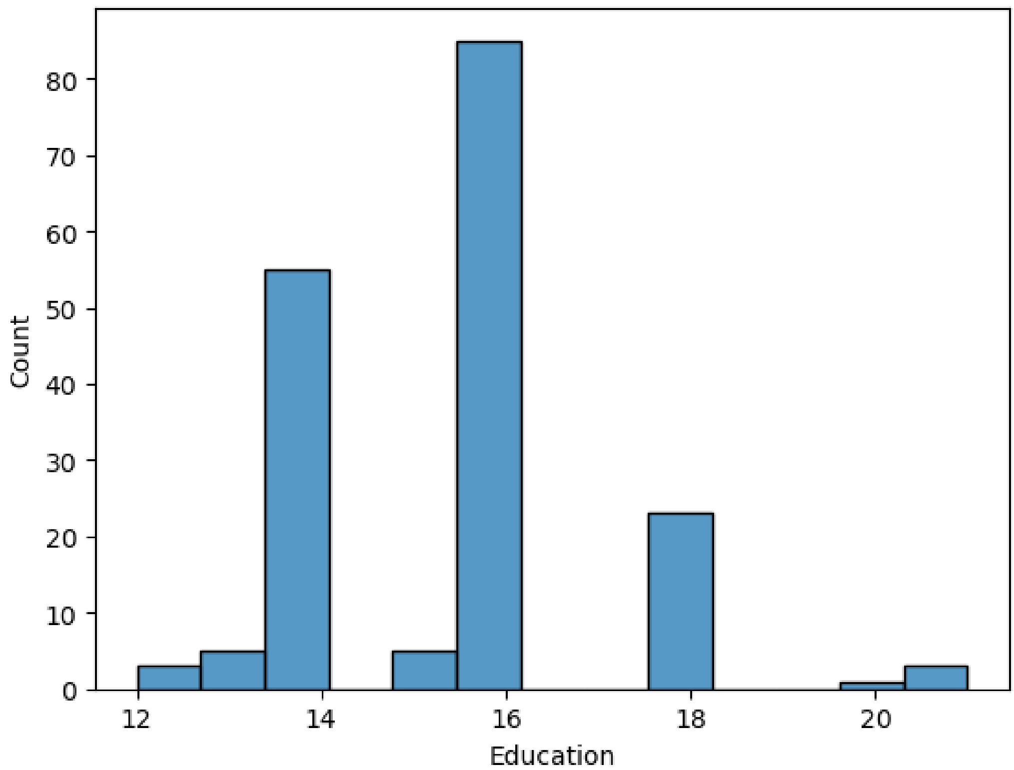
- **More than 35 customers earn 50-55K per year**
- **More than 30 customers earn 45-50K per year**
- **More than 20 customers earn 55-60K per year**

In [40]:

```
# Education Analysis - Histogram
sns.histplot(data=df,x='Education')
```

Out[40]:

```
<Axes: xlabel='Education', ylabel='Count'>
```



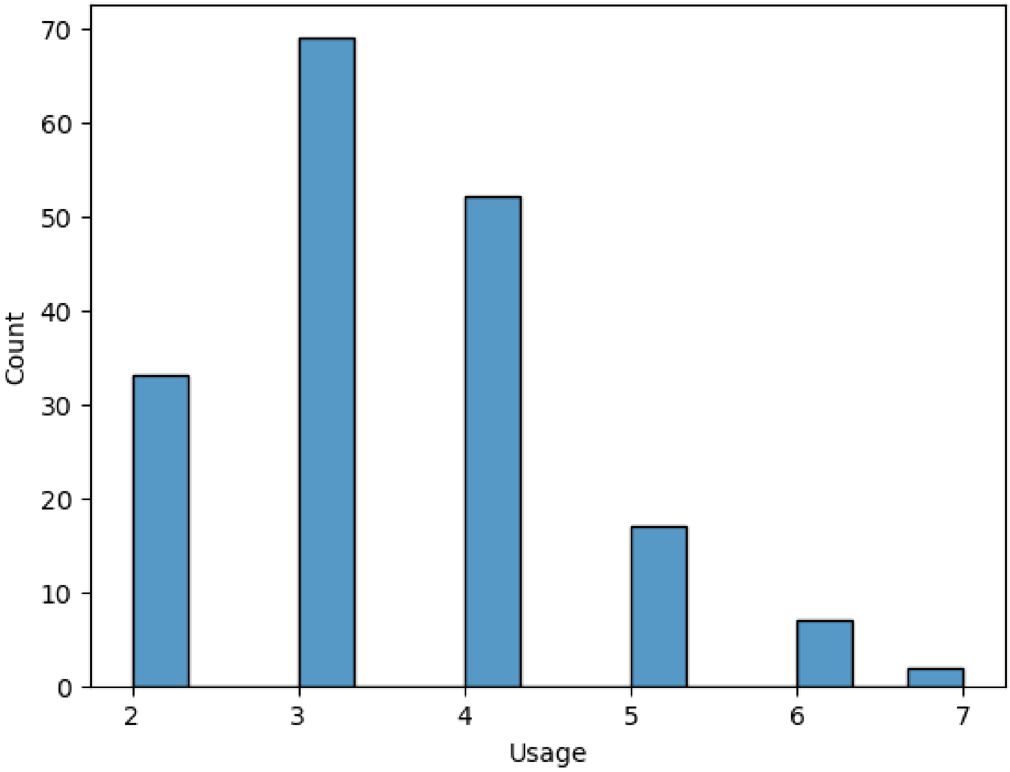
- **Highest number of customers have 16 as their Education**
- **14 is the second highest education among the customers**
- **20 is the least education among the customers**

In [41]:

```
# Usage Analysis - Histogram
sns.histplot(data=df,x='Usage')
```

Out[41]:

<Axes: xlabel='Usage', ylabel='Count'>



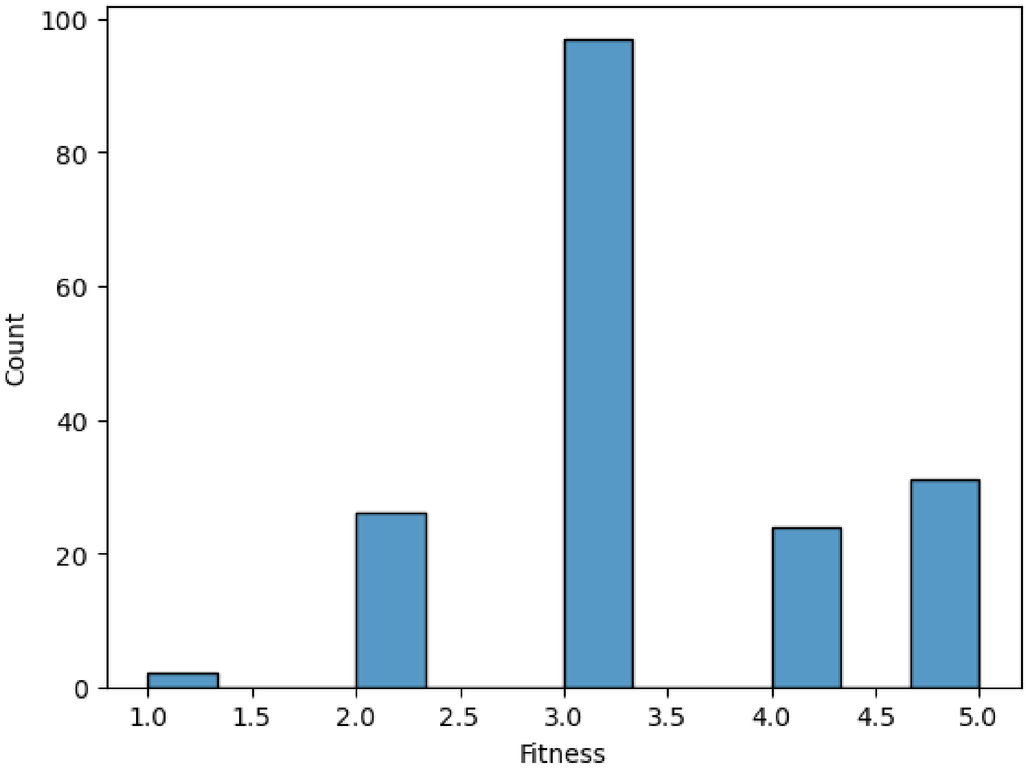
- **3 days per week is the most common usage among the customers**
- **4 days and 2 days per week is the second and third highest usage among the customers**
- **Very few customers use product 7 days per week**

In [42]:

```
# Fitness Analysis - Histogram
sns.histplot(data=df,x='Fitness')
```

Out[42]:

<Axes: xlabel='Fitness', ylabel='Count'>

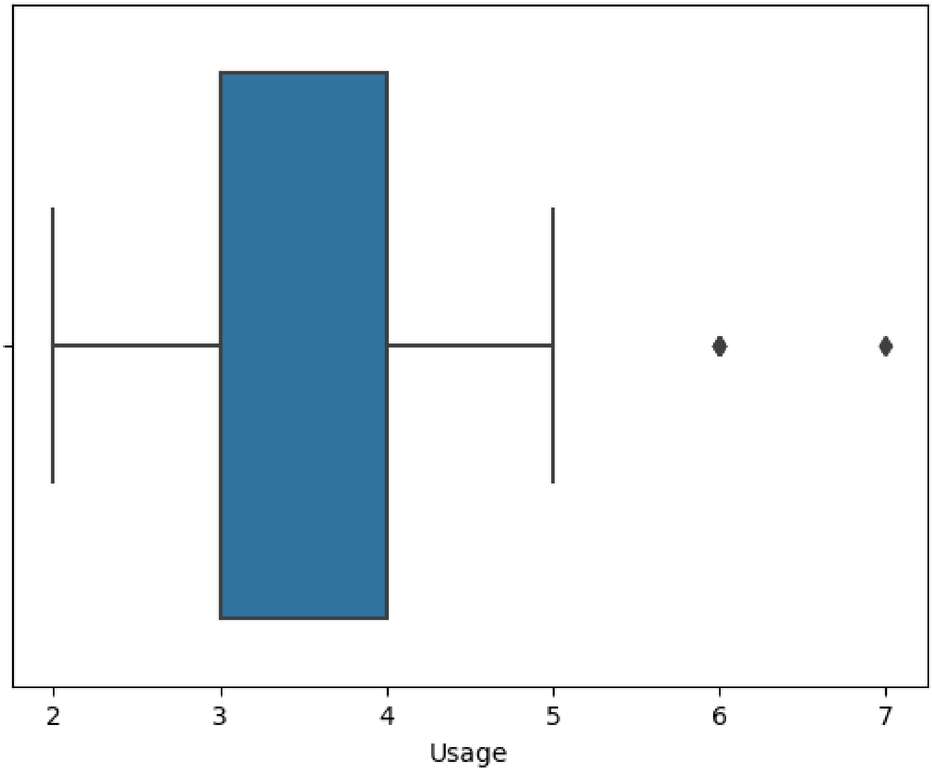


- **Average shape is the most rating customers have given for fitness rating**
- **Around 40 customers have stated Excelled Shape as fitness rating**

- **Around 40 customers have stated Excellent Shape as fitness rating**

In [43]:

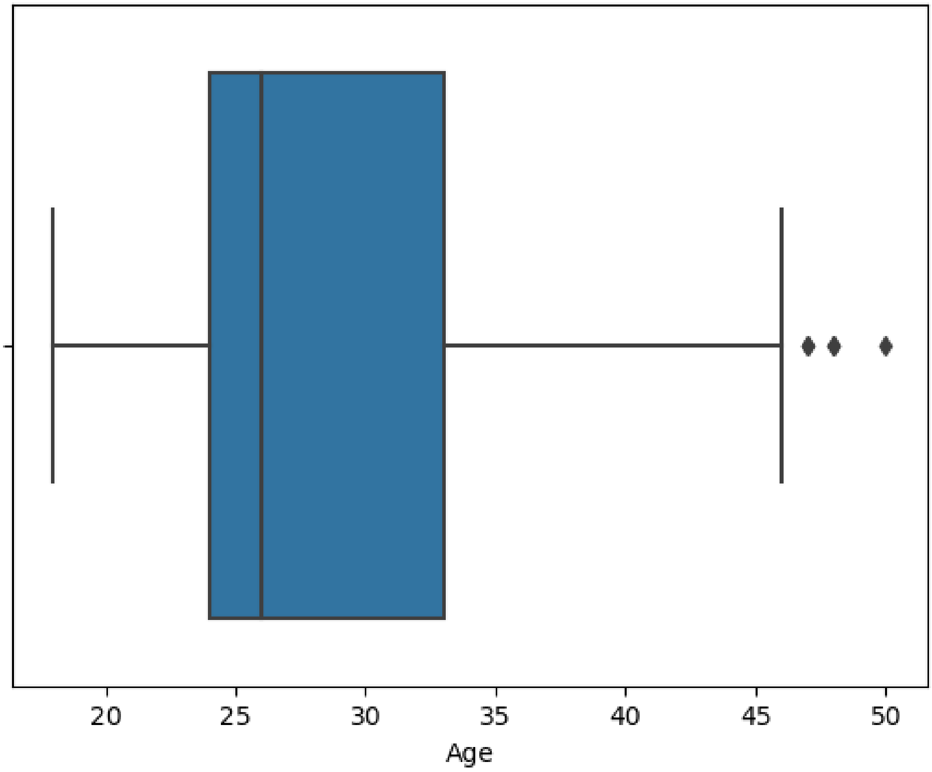
```
# Usage Analysis - Box plot
sns.boxplot(data=df,x='Usage')
plt.show()
```



- **3 to 4 days is the most preferred usage days for customers**
- **6 and 7 days per week is roughly the usage days for few customers (Outliers)**

In [44]:

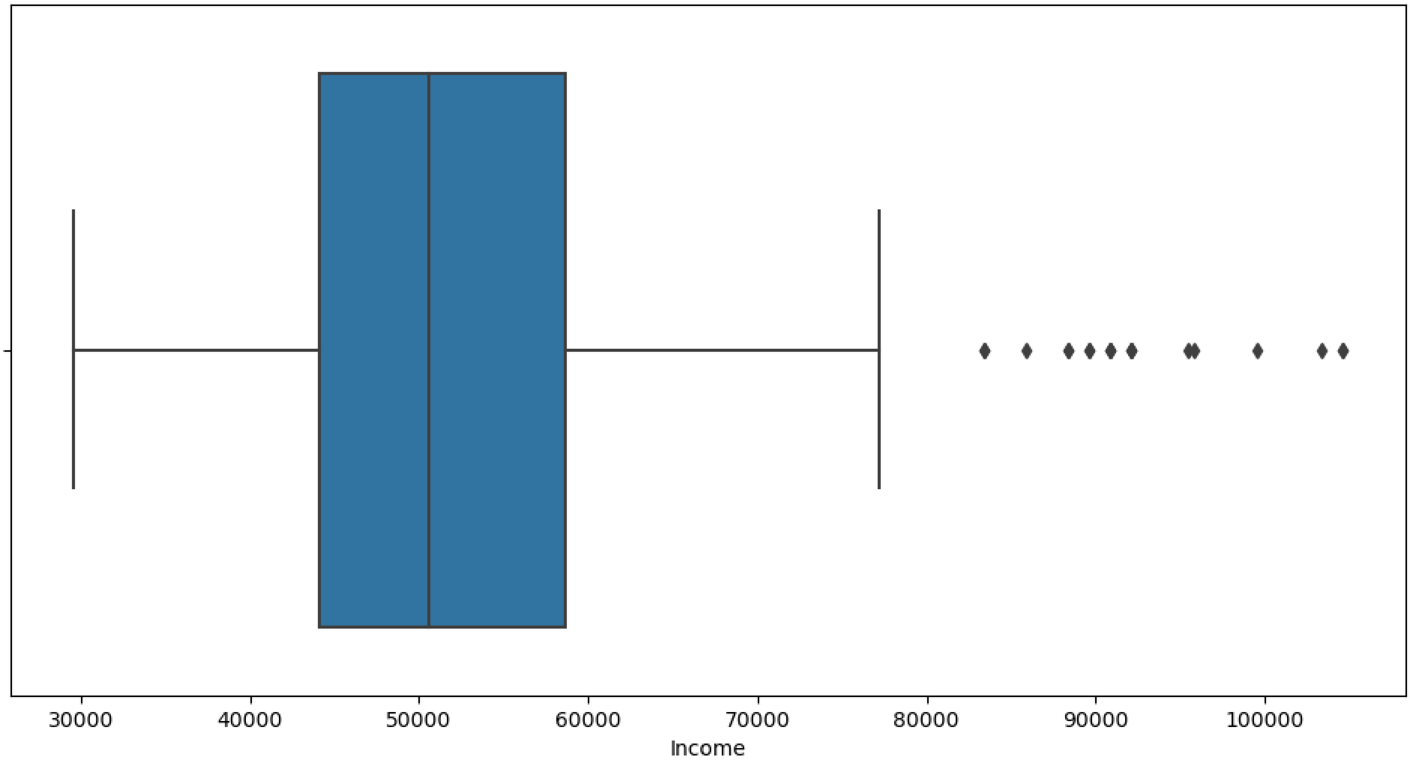
```
# Age Analysis - Box plot
sns.boxplot(data=df,x='Age')
plt.show()
```



- **23 to 34 is the most common customer age group that has purchased the product**
- **Above 45 years old customers are very few compared to the young age group given in the dataset**

In [45]:

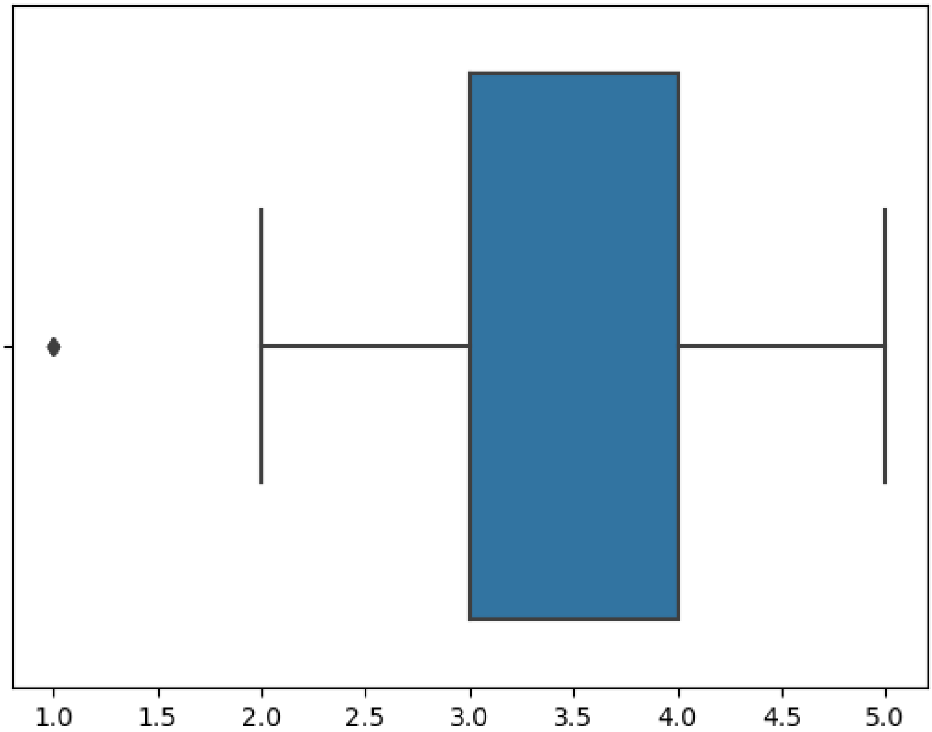
```
# Income Analysis - Box plot
plt.figure(figsize=(12,6))
sns.boxplot(data=df,x='Income')
plt.show()
```



- **Few customers have income above 80K per annum(Outliers)**
- **Most customers earn from 45K to around 60K per annum**

In [46]:

```
# Fitness Rating Analysis - Box plot
sns.boxplot(data=df,x='Fitness')
plt.show()
```



Fitness

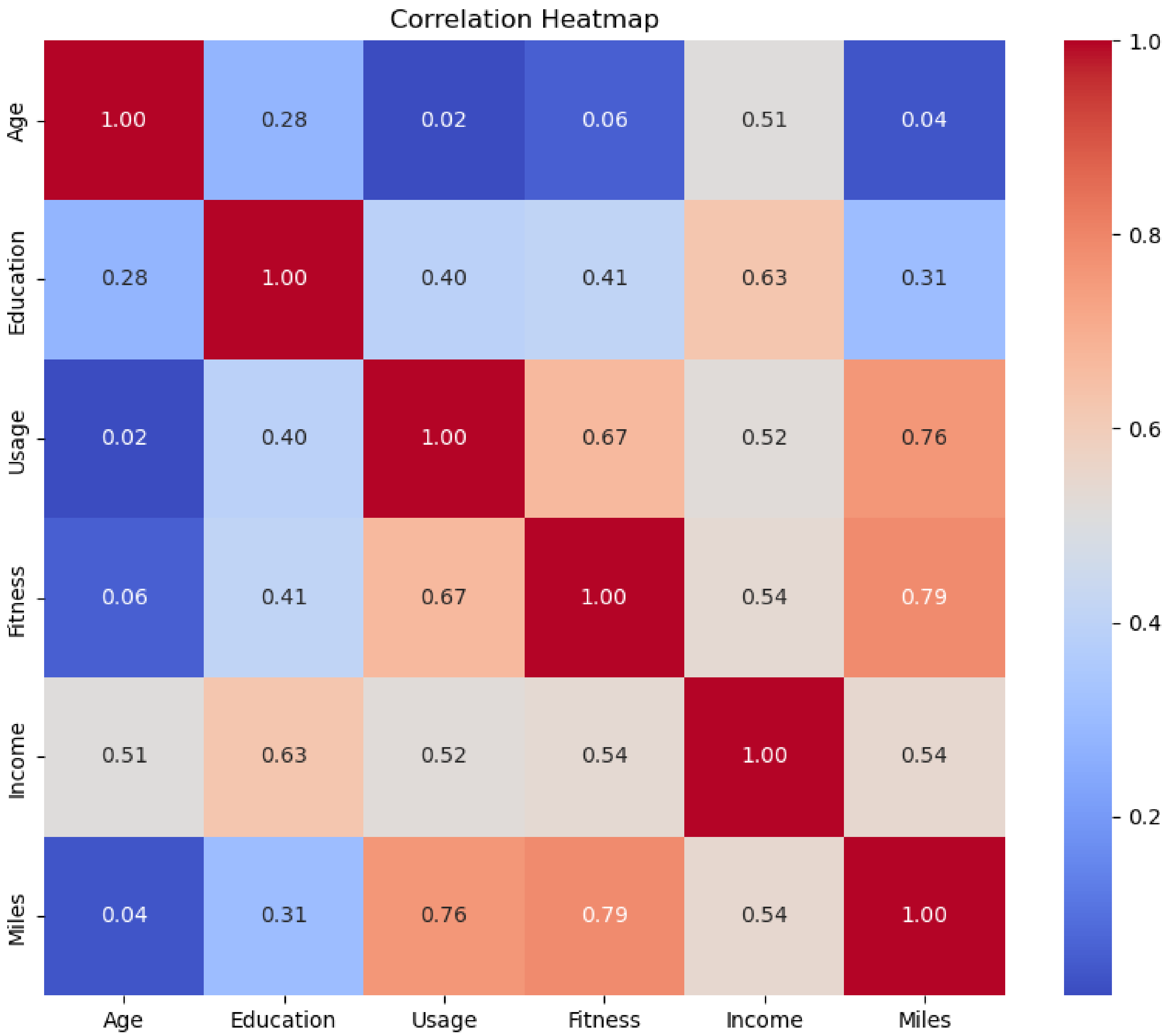
- **Couple of customers have rated their fitness rating as 1 - Poor Shape**
- **Most customers have rated fitness rating as 3.0 to 4.0**
- **For correlation: Heatmaps, Pairplots**

In [48]:

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

# Calculate the correlation matrix
correlation_matrix = attributes.corr()

# Create a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```

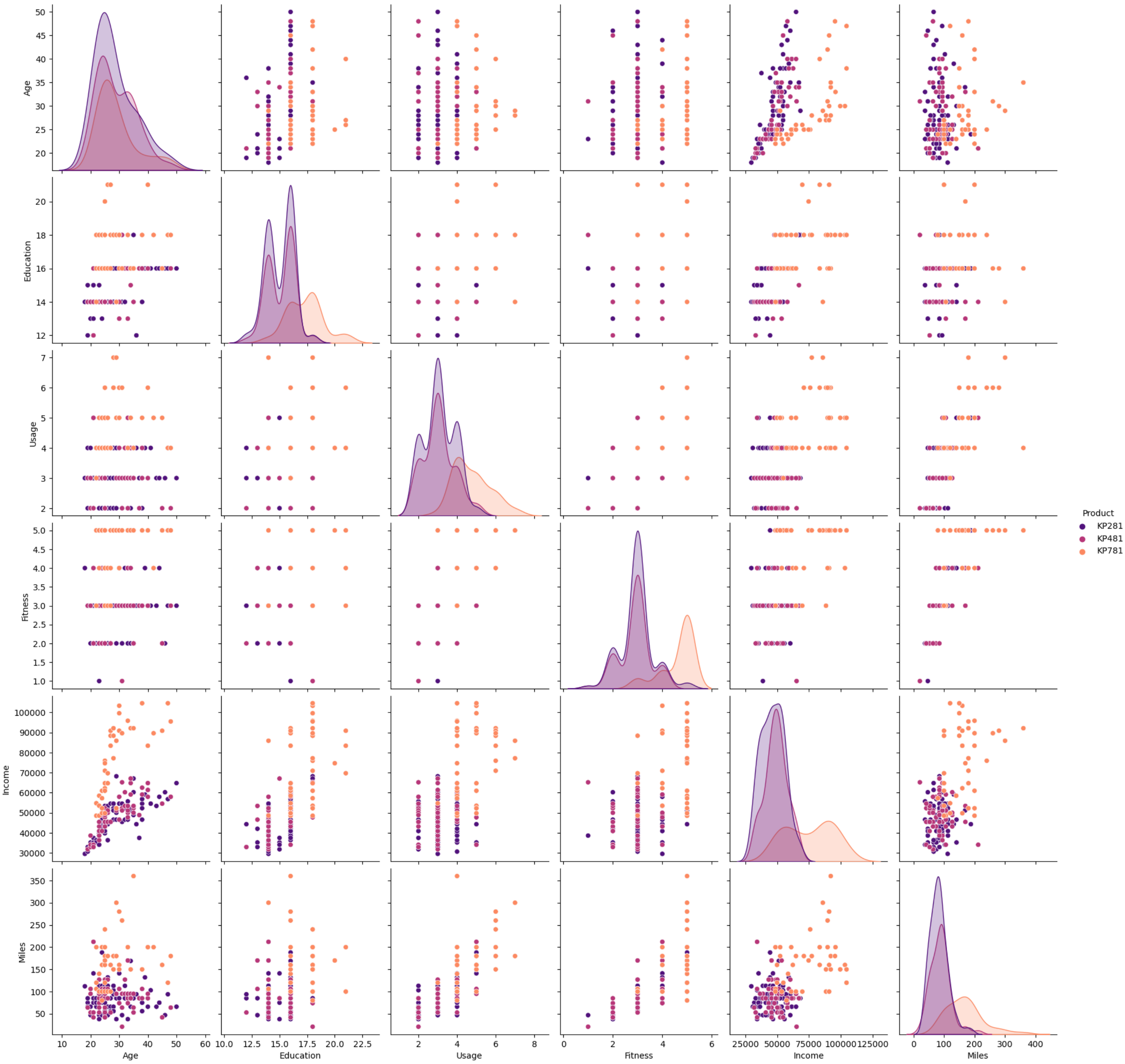


- **Miles and Fitness and Miles and Usage are highly correlated, which means if a customer's fitness level is high they use more treadmills.**
- **Income and education show a strong correlation. High-income and highly educated people prefer high-end models (KP781).**
- **There is no corelation between Usage & Age or Fitness & Age which mean Age should not be barrier to use treadmills or specific model of treadmills.**

In [49]:

```
sns.pairplot(data=df, hue='Product', palette= 'magma', height=3)
plt.show()
```

```
C:\Users\ASUS\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figur
e layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



Bivariate Analysis

In [50]:

```
# Average usage of each product type by the customer
df.groupby('Product')['Usage'].mean()
```

Out[50]:

```
Product
KP281    3.087500
KP481    3.066667
KP781    4.775000
Name: Usage, dtype: float64
```

- **Mean usage for product KP281 is 3.08**
- **Mean usage for product KP481 is 3.06**

- **Mean usage for product KP781 is 4.77**

In [51]:

```
# Average Age of customer using each product
df.groupby('Product')['Age'].mean()
```

Out[51]:

```
Product
KP281    28.55
KP481    28.90
KP781    29.10
Name: Age, dtype: float64
```

- **Mean Age of the customer who purchased product KP281 is 28.55**
- **Mean Age of the customer who purchased product KP481 is 28.90**
- **Mean Age of the customer who purchased product KP781 is 29.10**

In [52]:

```
# Average Education of customer using each product
df.groupby('Product')['Education'].mean()
```

Out[52]:

```
Product
KP281    15.037500
KP481    15.116667
KP781    17.325000
Name: Education, dtype: float64
```

Mean Education qualification of the customer who purchased product KP281 is 15.03 Mean Education qualification of the customer who purchased product KP481 is 15.11 Mean Education qualification of the customer who purchased product KP781 is 17.32

In [53]:

```
# Average customer fitness rating for each product type purchased
df.groupby('Product')['Fitness'].mean()
```

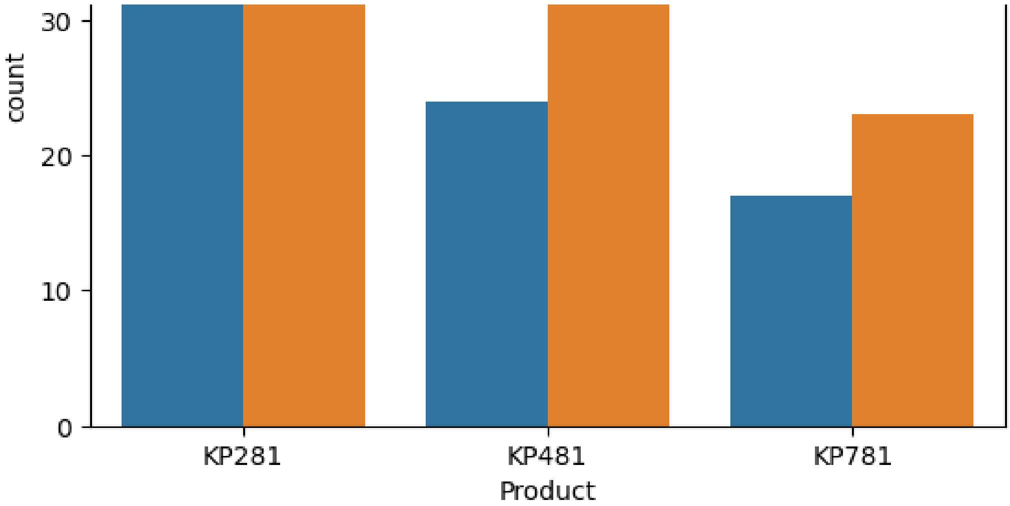
Out[53]:

```
Product
KP281    2.9625
KP481    2.9000
KP781    4.6250
Name: Fitness, dtype: float64
```

- **Customer fitness mean for product KP281 is 2.96**
- **Customer fitness mean for product KP481 is 2.90**
- **Customer fitness mean for product KP781 is 4.62**

In [54]:

```
# Product purchased among Married/Partnered and Single
sns.countplot(data=df,x='Product',hue='MaritalStatus')
plt.show()
```

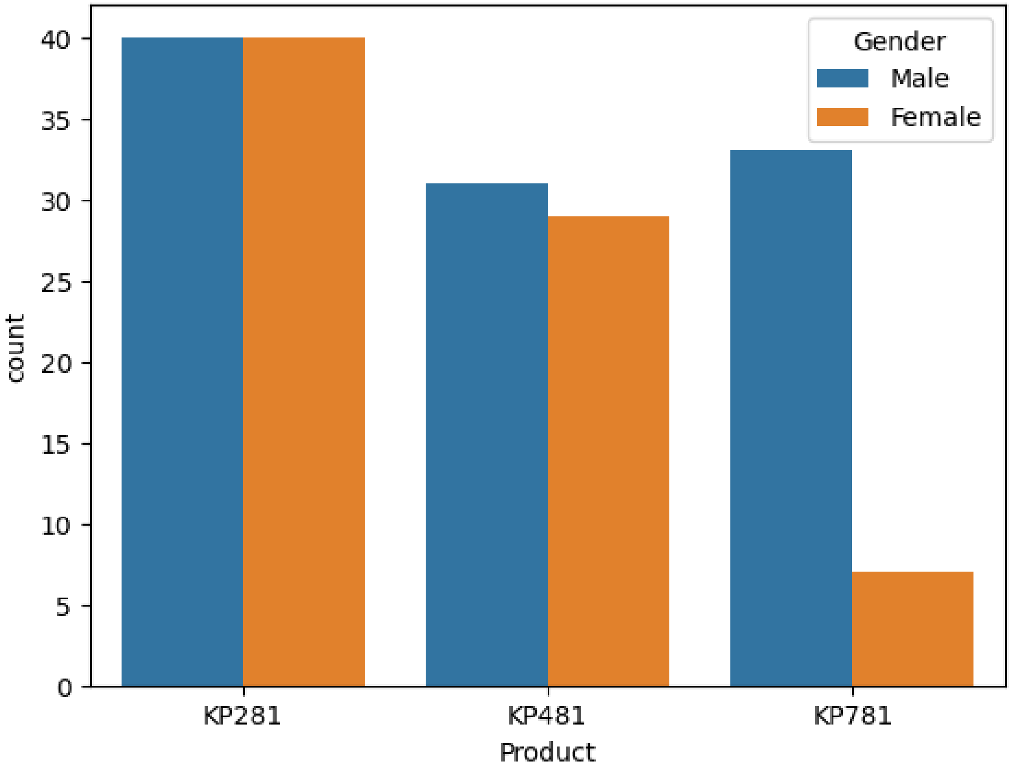


From the above countplot

- **KP281 is the most preferred product among customers**
- **KP481 is the second most preferred product among the customers**
- **Between Singles and Partnered, Partnered customers are the major product purchasers**

In [55]:

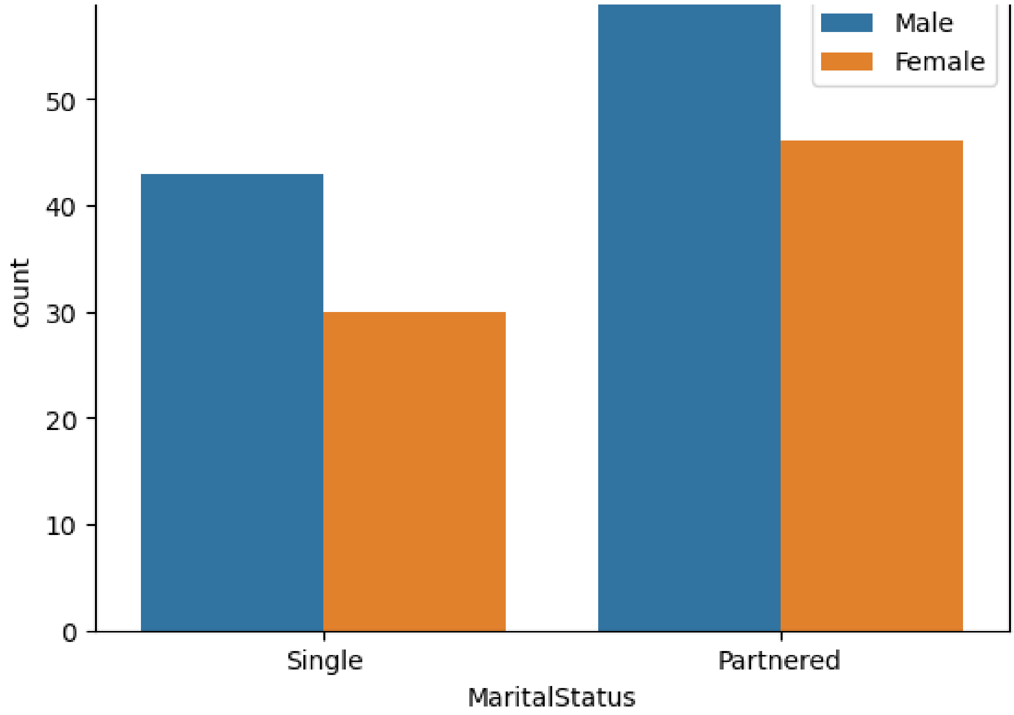
```
# Product purchased among Male and Female
sns.countplot(data=df,x='Product',hue='Gender')
plt.show()
```



- **KP281 Product is the equally preferred by both male and female genders**
- **KP781 Product is mostly preferred among the Male customers**
- **Overall Male customers are the highest product purchases**

In [56]:

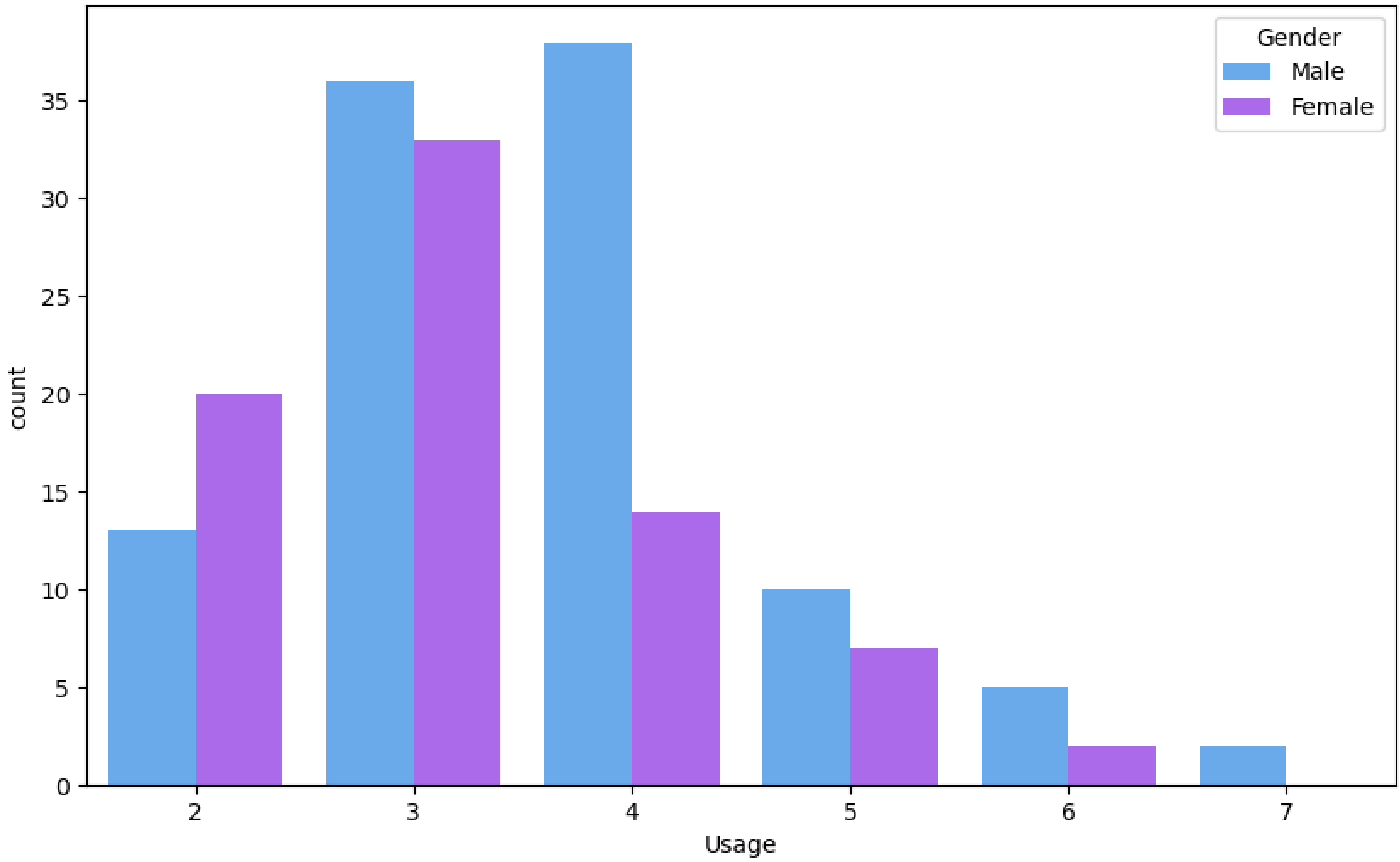
```
# Count among Gender and their Marital Status
sns.countplot(data=df,x='MaritalStatus',hue='Gender')
plt.show()
```



- **Partnered customers are the most buyers of aerofit product**
- **Out of both Single and Partnered customers, Male customers are significantly high**
- **Female customers are considerably low compared to Male customers**

In [57]:

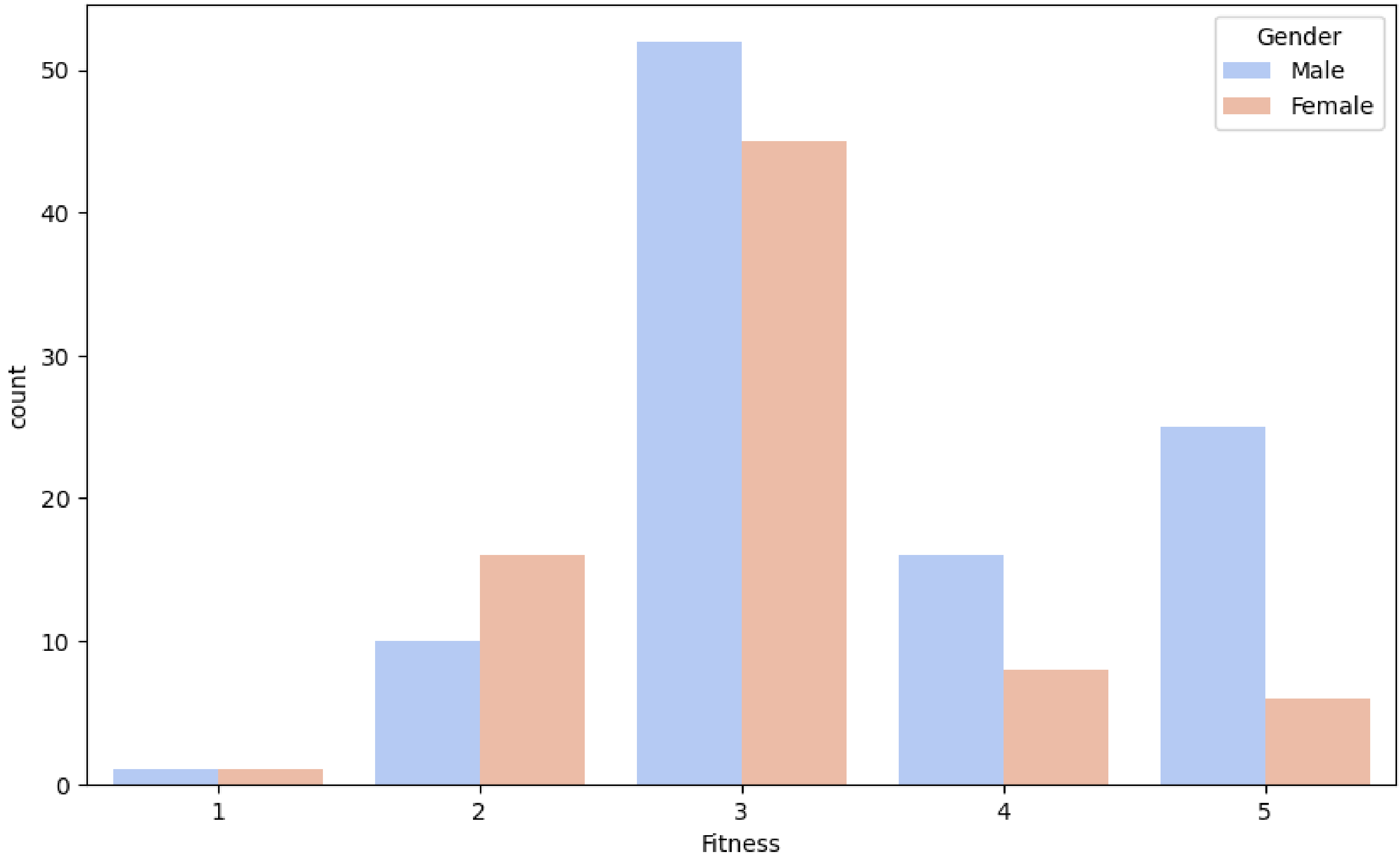
```
# Purchased product usage among Gender
plt.figure(figsize=(10,6))
sns.countplot(data=df,x='Usage',hue='Gender',palette='cool')
plt.show()
```



- **Among Male and Female genders, Male's usage is 4 days per week**
- **Female customers mostly use 3 days per week**
- **Only few Male customers use 7 days per week whereas female customer's maximum usage is only 6 days per week**

In [58]:

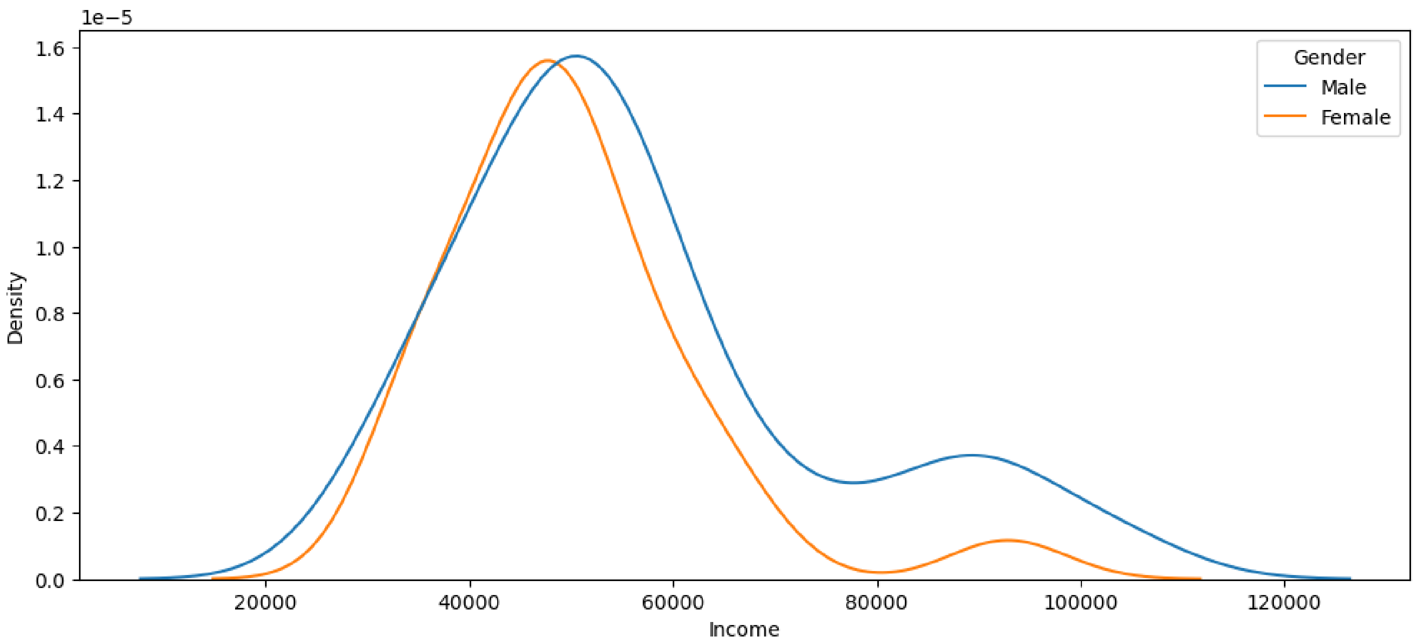
```
# Fitness rating among the customers categorised by Gender
plt.figure(figsize=(10,6))
sns.countplot(data=df,x='Fitness',hue='Gender',palette='coolwarm')
plt.show()
```



Among the fitness rating both Male and Female most have rated as average Significant number of Male customers are at Excellent shape compared to Female customers

In [59]:

```
# Product purchased Customers Income and their Gender
plt.figure(figsize=(12,5))
sns.kdeplot(data=df,x='Income',hue='Gender')
plt.show()
```



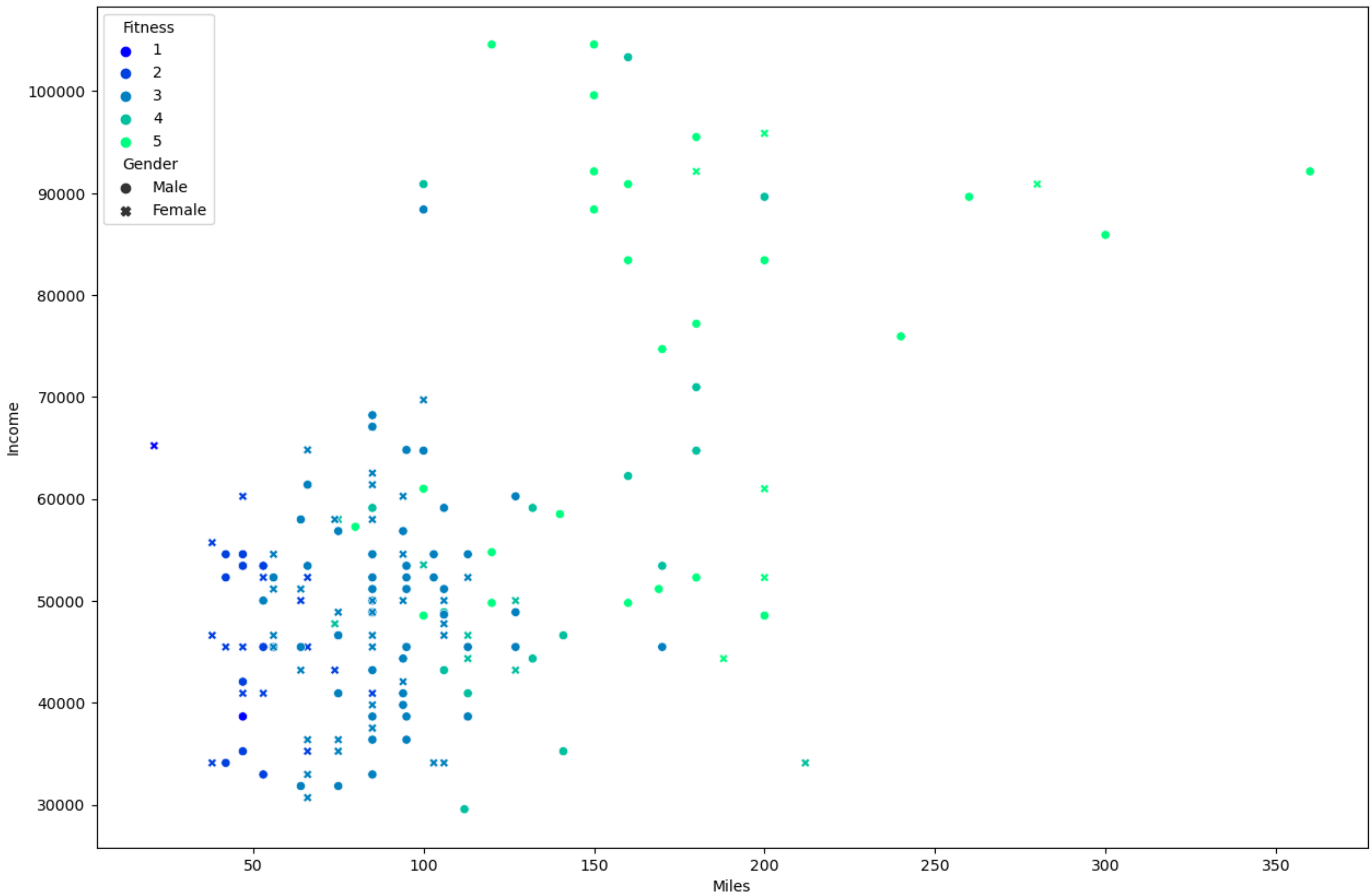
From the above diagram, we can conclude the spike from 40K to around 80K is the most common income per annum of the customers

In [60]:

```
# Scatter Plot
plt.figure(figsize=(15,10))
sns.scatterplot(x='Miles',y='Income',data=df,hue='Fitness',style='Gender',palette='winte
r')
```

Out[60]:

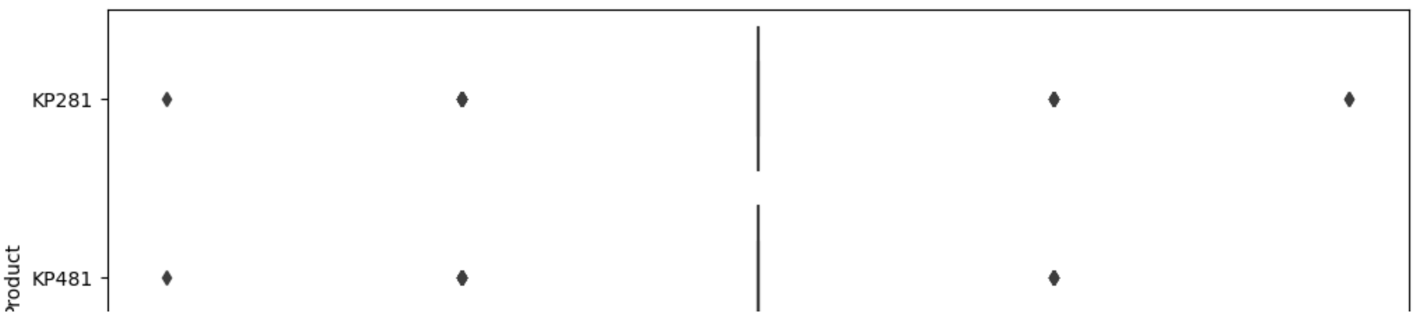
```
<Axes: xlabel='Miles', ylabel='Income'>
```

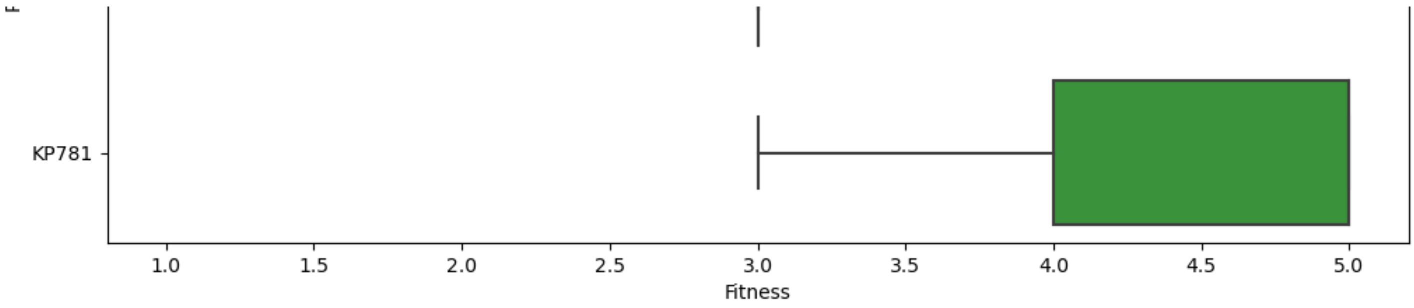


- **Above scattered Plot shows the overall picture over customer's income, how much they exercise (run/walk miles) given their gender and their fitness level.**
- **Most of the customer's fitness level is around 3 to 4 . and it says people who run more miles are having good fitness level.**
- **Though there is a trend with income and miles. But there are very few customers who earn a lot and run more miles.**

In [61]:

```
# Fitness of customer with each product
plt.figure(figsize=(12,5))
sns.boxplot(x='Fitness',y='Product',data=df)
plt.show()
```





Customers with excellent shape are significantly using KP781 product type KP481 and KP281 product type are scattered across the fitness rating

In [67]:

```
total_male_customers = len(df[df['Gender'] == 'Male'])
kp781_male_customers = len(df[(df['Gender'] == 'Male') & (df['Product'] == 'KP781')])

probability_male_kp781 = kp781_male_customers / total_male_customers

print(f"Probability of a male customer buying KP781: {probability_male_kp781:.2%}")
```

```
Probability of a male customer buying KP781: 31.73%
```

In [68]:

```
import pprint

for col in df.columns.tolist()[1:]:
    print("Feature: ",col)
    print()
    print("Absolute numbers: ")
    pprint.pprint(pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True))
    print()
    print("Normalized numbers: ")
    pprint.pprint(pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True, nor
malize=True))
    print()
    print("Marginal probs by gender(normalized): ")
    pprint.pprint(pd.crosstab(index=df['Gender'],columns=df
  ['Product'],margins=True, normalize='index'))
    print()
    print("Marginal probs by product(normalized): ")
    pprint.pprint(pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True, nor
malize='columns'))
    print("--"*50)
```

```
Feature:  Age

Absolute numbers: 
Product  KP281  KP481  KP781  All
Gender                           
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180

Normalized numbers: 
Product     KP281     KP481     KP781       All
Gender                                         
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized): 
Product     KP281     KP481     KP781
Gender                               
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222
```

```
Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
-----------------------------------------------------------------------------------------
-------------
Feature:  Gender

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized):
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
-----------------------------------------------------------------------------------------
-------------
Feature:  Education

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized):
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
-----------------------------------------------------------------------------------------
-------------
Feature:  MaritalStatus

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
```

```
Female       40      29       7   76
Male         40      31      33  104
All          80      60      40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized):
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
---------------------------------------------------------------------------------------
-------------
Feature:  Usage

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
Female       40      29       7   76
Male         40      31      33  104
All          80      60      40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized):
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
---------------------------------------------------------------------------------------
-------------
Feature:  Fitness

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
Female       40      29       7   76
Male         40      31      33  104
All          80      60      40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized):
```

```
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
--------------------------------------------------------------------------------------
-------------
Feature:  Income

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized):
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
--------------------------------------------------------------------------------------
-------------
Feature:  Miles

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized):
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
--------------------------------------------------------------------------------------
```

```
-------------
Feature:  Fitness_category

Absolute numbers:
Product  KP281  KP481  KP781  All
Gender
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180

Normalized numbers:
Product     KP281     KP481     KP781       All
Gender
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000

Marginal probs by gender(normalized):
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222

Marginal probs by product(normalized):
Product  KP281     KP481  KP781       All
Gender
Female     0.5  0.483333  0.175  0.422222
Male       0.5  0.516667  0.825  0.577778
-----------------------------------------------------------------------------------------
-------------
```

Customer Age Group Analysis

In [69]:

```
df_cat['age_group'] = df_cat.Age
df_cat.head()
```

Out[69]:

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

In [71]:

```
# 0-21 -> Teen
# 22-35 -> Adult
# 36-45 -> Middle Age
# 46-60 -> Elder Age
df_cat.age_group = pd.cut(df.
age_group,bins=[0,21,35,45,60],labels=['Teen','Adult','Middle Aged','Elder'])
```

In [72]:

```
df_cat.head()
```

Out[72]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Fitness_category	age_group
0	KP281	18	Male	14	Single	3	4	29562	112	Good Shape	Teen
1	KP281	19	Male	15	Single	2	3	31836	75	Average Shape	Teen

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Fitness_category	age_group
2	KP281	19	Female	14	Partnered	4	3	30699	66	Average Shape	Teen
3	KP281	19	Male	12	Single	3	3	32973	85	Average Shape	Teen
4	KP281	20	Male	13	Partnered	4	2	35247	47	Bad Shape	Teen

In [73]:

```
df_cat.age_group.value_counts()
```

Out[73]:

```
age_group
Adult           135
Middle Aged      22
Teen             17
Elder             6
Name: count, dtype: int64
```

In [74]:

```
df_cat.loc[df_cat.Product=='KP281']["age_group"].value_counts()
```

Out[74]:

```
age_group
Adult           56
Middle Aged     11
Teen            10
Elder            3
Name: count, dtype: int64
```

In [75]:

```
df_cat.loc[df_cat.Product=='KP481']["age_group"].value_counts()
```

Out[75]:

```
age_group
Adult           45
Teen             7
Middle Aged      7
Elder            1
Name: count, dtype: int64
```

In [76]:

```
df_cat.loc[df_cat.Product=='KP781']["age_group"].value_counts()
```

Out[76]:

```
age_group
Adult           34
Middle Aged      4
Elder            2
Teen             0
Name: count, dtype: int64
```

In [77]:

```
pd.crosstab(index=df_cat.Product,columns=df_cat.age_group,margins=True)
```

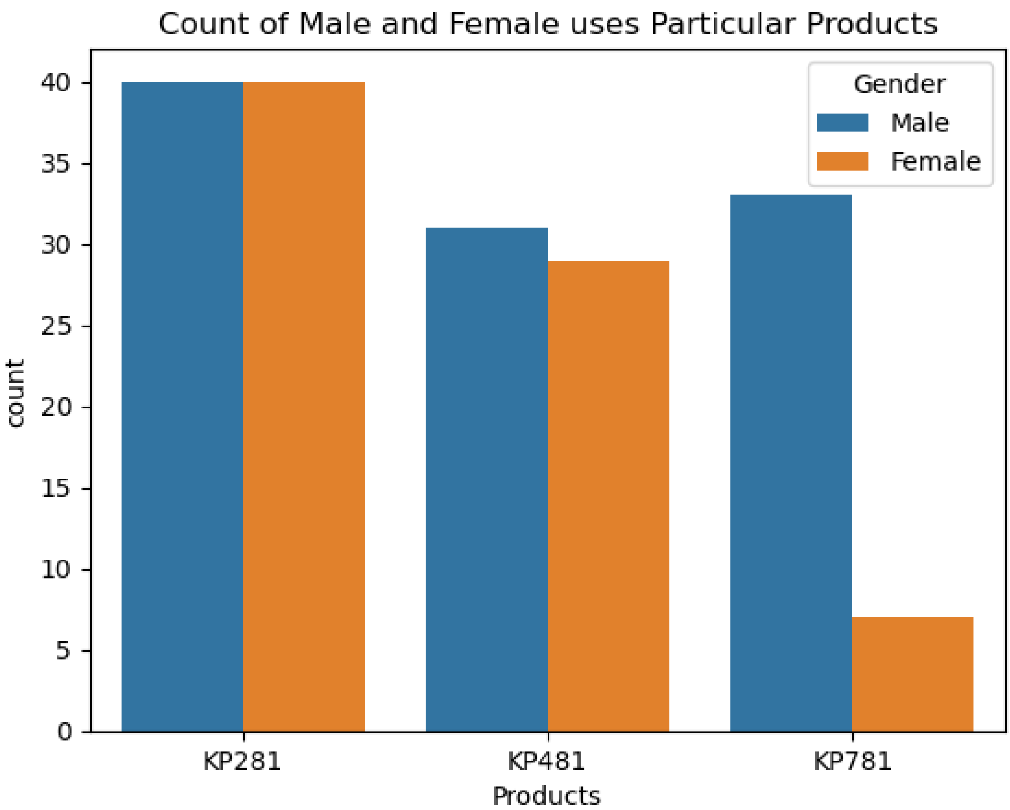
Out[77]:

age_group	Teen	Adult	Middle Aged	Elder	All
Product					
KP281	10	56	11	3	80
KP481	7	45	7	1	60
KP781	0	34	4	2	40

age_group	Teen	Adult	Middle Aged	Elder	All
All	17	135	22	6	180

In [78]:

```
sns.countplot(x = "Product", data= df, hue = "Gender")
plt.xlabel("Products")
plt.title("Count of Male and Female uses Particular Products")
plt.show()
```



In [79]:

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

Out[79]:

Gender / Product	Female	Male	All
KP281	40	40	80
KP481	29	31	60
KP781	7	33	40
All	76	104	180

In [80]:

```
np.round(((pd.crosstab(df.Product,df.Gender,margins=True))/180)*100,2)
```

Out[80]:

Gender / Product	Female	Male	All
KP281	22.22	22.22	44.44
KP481	16.11	17.22	33.33
KP781	3.89	18.33	22.22
All	42.22	57.78	100.00

Marginal Probability

Marginal Probability

- Probability of Male Customer Purchasing any product is : 57.77 %
- Probability of Female Customer Purchasing any product is : 42.22 %
- Marginal Probability of any customer buying
- product KP281 is : 44.44 % (cheapest / entry level product)
- product KP481 is : 33.33 % (intermediate user level product)
- product KP781 is : 22.22 % (Advanced product with ease of use that help in covering longer

In [81]:

```
np.round((pd.crosstab([df.Product],df.
Gender,margins=True,normalize="columns"))*100,2)
```

Out[81]:

Gender	Female	Male	All
Product			
KP281	52.63	38.46	44.44
KP481	38.16	29.81	33.33
KP781	9.21	31.73	22.22

Probability of Selling Product

- KP281 | Female = 52 %
- KP481 | Female = 38 %
- KP781 | Female = 10 %
- KP281 | male = 38 %
- KP481 | male = 30 %
- KP781 | male = 32 %
- Probability of Female customer buying KP281(52.63%) is more than male(38.46%).
- KP281 is more recommended for female customers.

- Probability of Male customer buying Product KP781(31.73%) is way more than female(9.21%).
- Probability of Female customer buying Product KP481(38.15%) is significantly higher than male (29.80%.)
- KP481 product is specifically recommended for Female customers who are intermediate user.

Objective: Customer Profiling for Each Product

Customer profiling based on the 3 product categories provided:-

KP281

- Easily affordable entry level product, which is also the maximum selling product.
- KP281 is the most popular product among the entry level customers.
- This product is easily afforded by both Male and Female customers.
- Average distance covered in this model is around 70 to 90 miles.
- Product is used 3 to 4 times a week.
- Most of the customer who have purchased the product have rated Average shape as the fitness rating.
- Younger to Elder beginner level customers prefer this product.
- Single female & Partnered male customers bought this product more than single male customers.
- Income range between 39K to 53K have preferred this product.

KP481

- This is an Intermediate level Product.
- KP481 is the second most popular product among the customers.
- Fitness Level of this product users varies from Bad to Average Shape depending on their usage.

- Customers Prefer this product mostly to cover more miles than fitness.
- Average distance covered in this product is from 70 to 130 miles per week.
- More Female customers prefer this product than males.
- Probability of Female customer buying KP481 is significantly higher than male.
- KP481 product is specifically recommended for Female customers who are intermediate user.
- Three different age groups prefer this product - Teen, Adult and middle aged.
- Average Income of the customer who buys KP481 is 49K.
- Average Usage of this product is 3 days per week.
- More Partnered customers prefer this product.
- There are slightly more male buyers of the KP481.
- The distance travelled on the KP481 treadmill is roughly between 75 - 100
- Miles. It is also the 2nd most distance travelled model.

KP781

- Due to the High Price & being the advanced type, customer prefers less of this product.
- Customers use this product mainly to cover more distance.
- Customers who use this product have rated excelled shape as fitness rating.
- Customer walk/run average 120 to 200 or more miles per week on his product.
- Customers use 4 to 5 times a week at least.
- Female Customers who are running average 180 miles (extensive exercise) , are using product
- KP781, which is higher than Male average using same product.
- Probability of Male customer buying Product KP781(31.73%) is way more than female(9.21%).
- Probability of a single person buying KP781 is higher than Married customers. So , KP781 is also
- recommended for people who are single and exercises more.
- Middle aged to higher age customers tend to use this model to cover more distance.
- Average Income of KP781 buyers are over 75K per annum
- Partnered Female bought KP781 treadmill compared to Partnered Male.
- Customers who have more experience with previous aerofit products tend to buy this product

Recommendation

- Female who prefer exercising equipments are very low here. Hence, we should run a marketing
- campaign on to encourage women to exercise more
- KP281 & KP481 treadmills are preferred by the customers whose annual income lies in the range of 39K - 53K Dollars. These models should promoted as budget treadmills.
- As KP781 provides more features and functionalities, the treadmill should be marketed for professionals and athletes.
- KP781 product should be promotted using influencers and other international atheletes.
- Research required for expanding market beyond 50 years of age considering health pros and cons.
- Provide customer support and recommend users to upgrade from lower versions to next level versions after consistent usages.
- KP781 can be recommended for Female customers who exercises extensively along with easy usage guidance since this type is advanced.
- Target the Age group above 40 years to recommend Product KP781.

```
In [ ]:
```