1) Problem Statement:

- Aerofit is India's leading fitness equipment brand that manufactures residential and commercial fitness machines including treadmills.Now we want to understand the 3 models given which are doing great in some cases.Now we want to analyze the given data so that we can give our insights and recommendations to the Aerofit which makes their revenue higher

In [ ]:

```

```

In [263]:

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

In [125]:

```
data=pd.read_csv("aerofit_treadmill.csv")
```

In [126]:

```
data
```

Out[126]:

	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 [127]:

```
data.head()
```

Out[127]:

	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 [128]:

```
data.tail()
```

Out[128]:

	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

In [129]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Product        180 non-null    object
 1   Age            180 non-null    int64 
 2   Gender         180 non-null    object
 3   Education      180 non-null    int64 
 4   MaritalStatus  180 non-null    object
 5   Usage          180 non-null    int64 
 6   Fitness        180 non-null    int64 
 7   Income         180 non-null    int64 
 8   Miles          180 non-null    int64 
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

In [130]:

```
data.shape
```

Out[130]:

```
(180, 9)
```

In [131]:

```
d=data.select_dtypes(include=['int64'])
a=list(d.columns)
a
```

Out[131]:

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

Checking for null values

In [132]:

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

Out[132]:

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

- There are no null values in the diven data

In [133]:

```
data.duplicated().sum()
```

Out[133]:

```
0
```

Describing the data

In [134]:

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

Out[134]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	
count	180	180.000000	180	180.000000	180	180.000000	180.000000	18
unique	3	NaN	2	NaN	2	NaN	NaN	
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	
freq	80	NaN	104	NaN	107	NaN	NaN	
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	5371
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	1650
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	2956
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	4405
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	5059
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	5866
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	10458

Observations

- The min age that bought a treadmill was 18 year and max was 50 year
- KP281 was the most bought treadmill
- 50% of people who bought treadmill were under 26 years
- Most of treadmills were bought by males

In [135]:

```
data.nunique()
```

Out[135]:

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

In [136]:

```
data.columns
```

Out[136]:

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

Finding unique and abnormal data

In [137]:

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

Out[137]:

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

In [138]:

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

Out[138]:

```
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 [139]:

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

Out[139]:

```
array(['Male', 'Female'], dtype=object)
```

In [140]:

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

Out[140]:

```
array([14, 15, 12, 13, 16, 18, 20, 21], dtype=int64)
```

In [141]:

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

Out[141]:

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

In [142]:

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

Out[142]:

```
array([3, 2, 4, 5, 6, 7], dtype=int64)
```

In [143]:

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

Out[143]:

```
array([4, 3, 2, 1, 5], dtype=int64)
```

In [144]:

```
print(data['Income'].unique())
print(data['Income'].max())
```

```
[ 29562  31836  30699  32973  35247  37521  36384  38658  40932  34110
  39795  42069  44343  45480  46617  48891  53439  43206  52302  51165
  50028  54576  68220  55713  60261  67083  56850  59124  61398  57987
  64809  47754  65220  62535  48658  54781  48556  58516  53536  61006
  57271  52291  49801  62251  64741  70966  75946  74701  69721  83416
  88396  90886  92131  77191  52290  85906 103336  99601  89641  95866
 104581  95508]
104581
```

In [145]:

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

Out[145]:

```
array([112,  75,  66,  85,  47, 141, 103,  94, 113,  38, 188,  56, 132,
       169,  64,  53, 106,  95, 212,  42, 127,  74, 170,  21, 120, 200,
       140, 100,  80, 160, 180, 240, 150, 300, 280, 260, 360], dtype=int64)
```

- As we can see there is no abnormal data such as having different type of data in the same column

Identifying type of Variables

- Numerical Variables:

```
* Income
* Miles
```

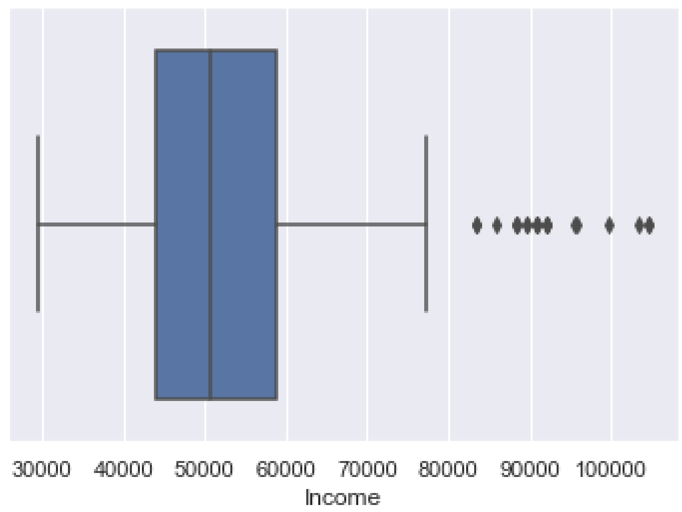
Detection of Outliers

In [146]:

```
sns.boxplot(x=data['Income'])
```

Out[146]:

<AxesSubplot:xlabel='Income'>

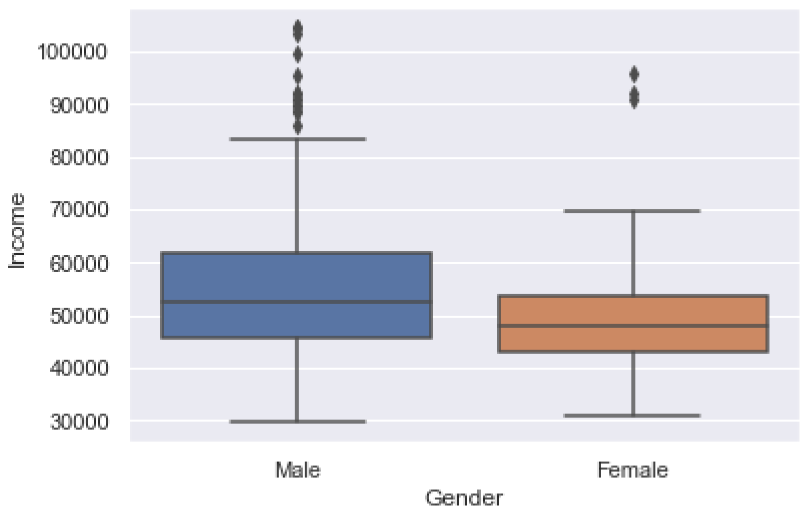


In [147]:

```
sns.boxplot(x=data['Gender'],y=data['Income'],data=data)
```

Out[147]:

<AxesSubplot:xlabel='Gender', ylabel='Income'>

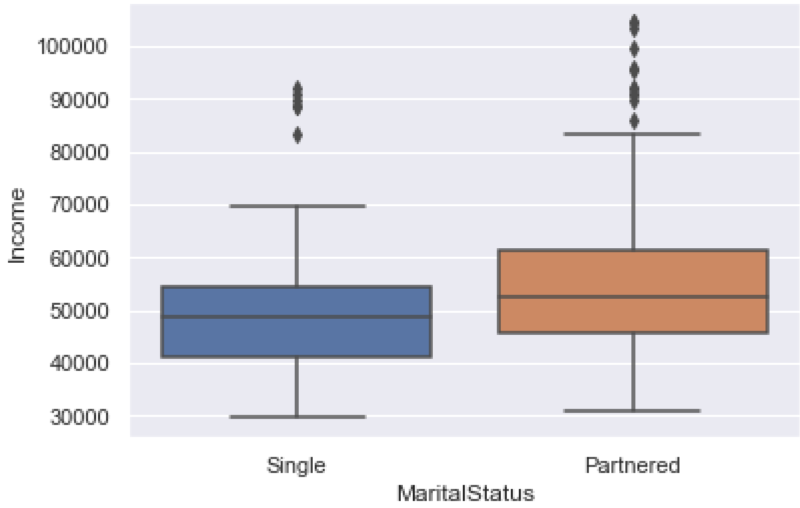


In [148]:

```
sns.boxplot(x=data['MaritalStatus'],y=data['Income'],data=data)
```

Out[148]:

```
<AxesSubplot:xlabel='MaritalStatus', ylabel='Income'>
```



- From above graphs we can say that there are outliers in the Income
- Males having higher income are tending to use more treadmills than females
- Partnered people havving higher income are using more compared to singles

Handling Outliers

In [149]:

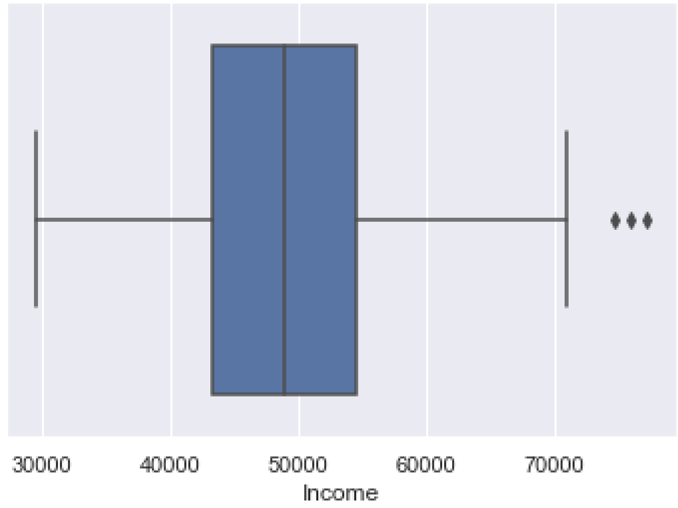
```
data1=data.copy()
```

In [150]:

```
Q3=data1['Income'].quantile(0.75)
Q1=data1['Income'].quantile(0.25)
IQR=Q3-Q1
data1=data1[(data1['Income']>Q1-1.5*IQR)&(data1['Income']<Q3+1.5*IQR)]
sns.boxplot(x=data1['Income'])
```

Out[150]:

```
<AxesSubplot:xlabel='Income'>
```

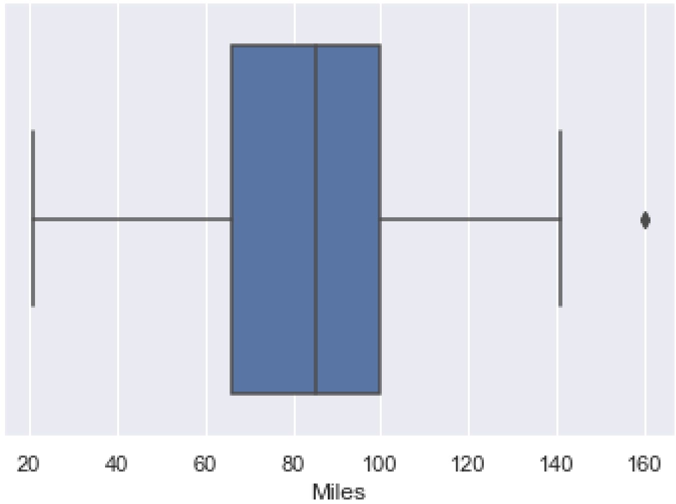


In [151]:

```
Q3=data1['Miles'].quantile(0.75)
Q1=data1['Miles'].quantile(0.25)
IQR=Q3-Q1
data1=data1[(data1['Miles']>Q1-1.5*IQR)&(data1['Miles']<Q3+1.5*IQR)]
sns.boxplot(x=data1['Miles'])
```

Out[151]:

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



- Above we have removed removed the outliers in Income and Miles columns

In [152]:

```
data1['Gender'] = data1['Gender'].astype("category")
data1['Product'] = data1['Product'].astype("category")
data1['MaritalStatus'] = data1['MaritalStatus'].astype("category")
data1['Fitness'] = data1['Fitness'].astype("category")
```

Univariate Analysis of Numerical Variables

Age

In [153]:

```
data1['Age'].mean()
```

Out[153]:

```
28.346938775510203
```

In [154]:

```
data1['Age'].median()
```

Out[154]:

26.0

In [ ]:

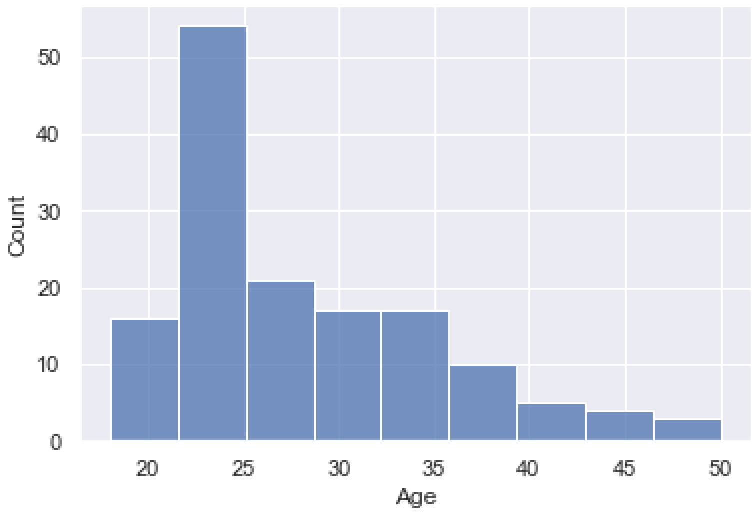
```
```

In [155]:

```
sns.histplot(x='Age',data=data1)
```

Out[155]:

<AxesSubplot:xlabel='Age', ylabel='Count'>



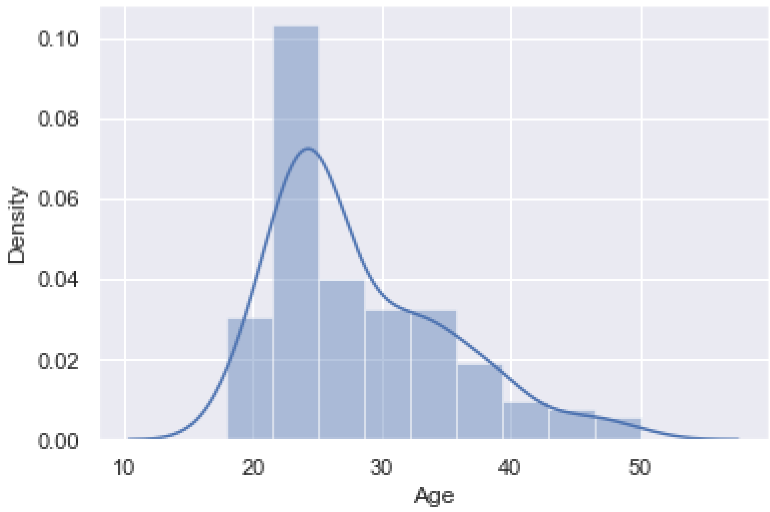
In [156]:

```
sns.distplot(data1['Age'])
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[156]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```

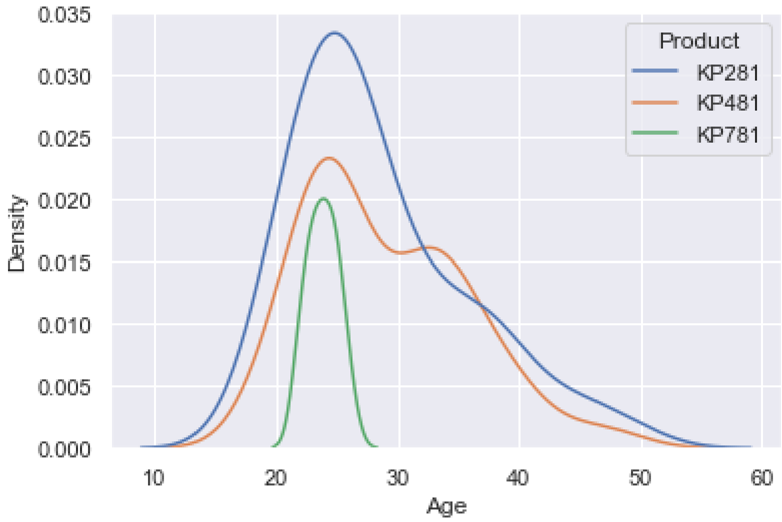


In [157]:

```
sns.kdeplot(x='Age',data=data1,hue='Product')
```

Out[157]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```



- From graphs we can see that mostly all types of products are bought by people between 20 -30 age gropup
- As the age is becoming higher that is older people buying these is very small

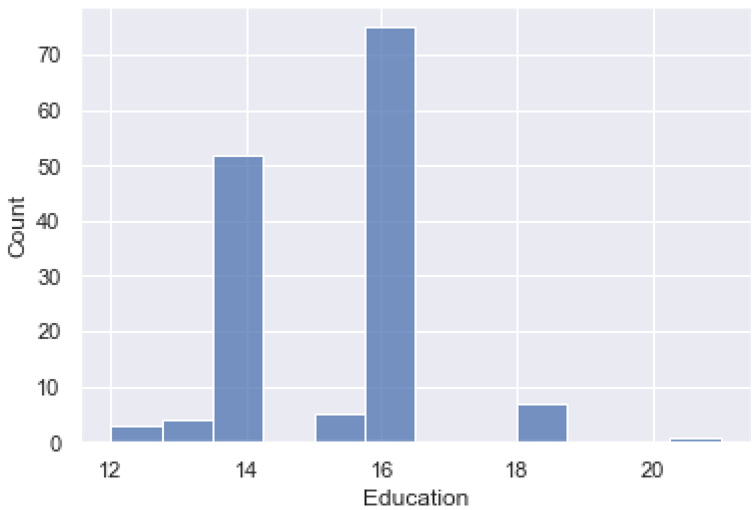
Education

In [158]:

```
sns.histplot(x='Education',data=data1)
```

Out[158]:

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



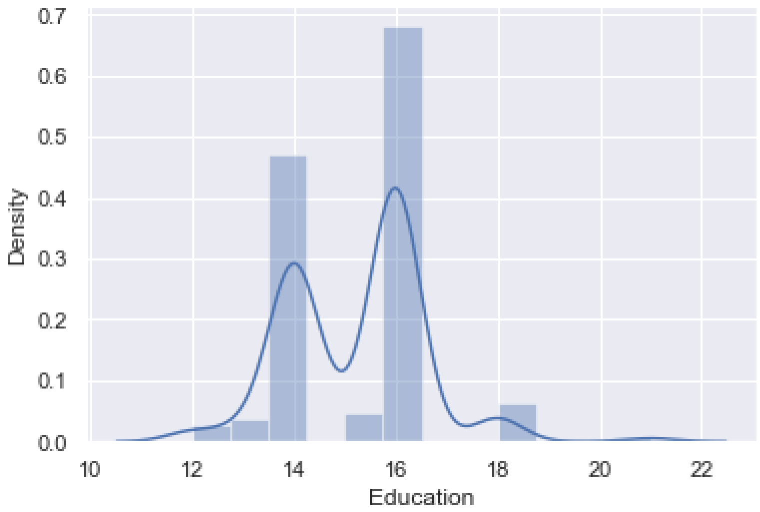
In [159]:

```
sns.distplot(data1['Education'])
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[159]:

```
<AxesSubplot:xlabel='Education', ylabel='Density'>
```



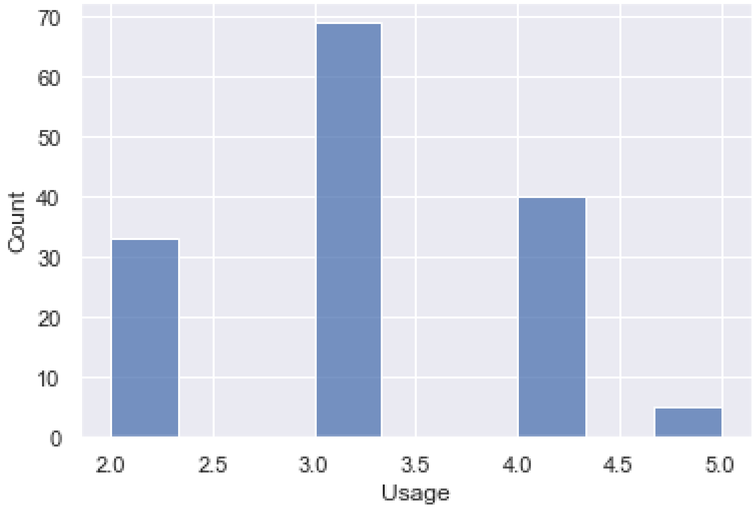
Usage

In [160]:

```
sns.histplot(x='Usage',data=data1)
```

Out[160]:

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



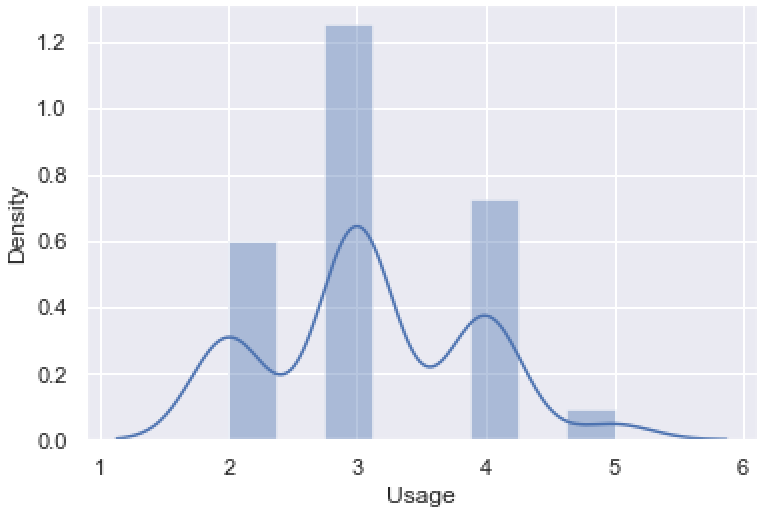
In [161]:

```
sns.distplot(data1['Usage'])
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[161]:

<AxesSubplot:xlabel='Usage', ylabel='Density'>

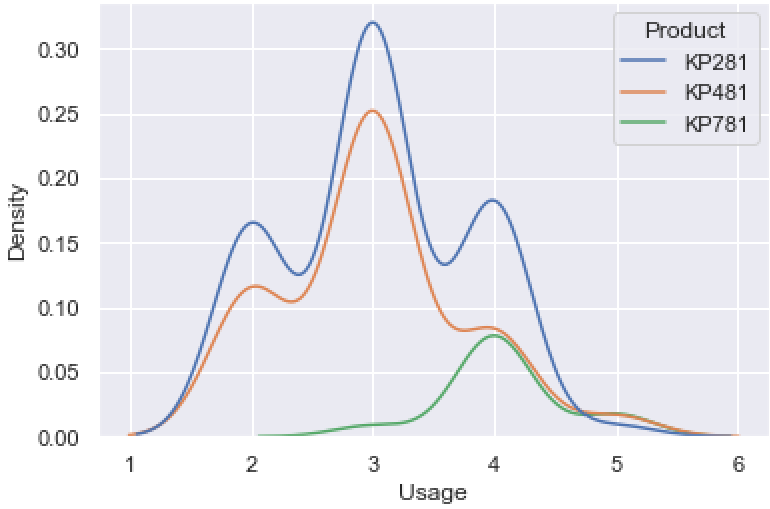


In [162]:

```
sns.kdeplot(data=data1,x='Usage',hue='Product')
```

Out[162]:

```
<AxesSubplot:xlabel='Usage', ylabel='Density'>
```



- From above we KP281 and KP481 models are used mostly 3 days where as model KP781 is used modtly 4 day

Univariate analysis of Categorical variables

Before diving there first we need to convert some numerical variables into categorical

In [171]:

```
bins=[15,20,30,40,55]
labels=['Below 20','20-30','30-40','Above 40']
data1['Category_Age']=pd.cut(data1['Age'],bins,labels=labels)
```

In [172]:

```
data1['Category_Age'].dtype
```

Out[172]:

```
CategoricalDtype(categories=['Below 20', '20-30', '30-40', 'Above 40'], orde
red=True)
```

In [178]:

```
bins=[29000,36000,60000,110000]
labels=['Low Income','Average Income','High Income']
data1['Income_Category']=pd.cut(data1['Income'],bins,labels=labels)
data1['Income_Category'].unique()
```

Out[178]:

```
['Low Income', 'Average Income', 'High Income']
Categories (3, object): ['Low Income' < 'Average Income' < 'High Income']
```

Univariate analysis of Categorical variables

In [179]:

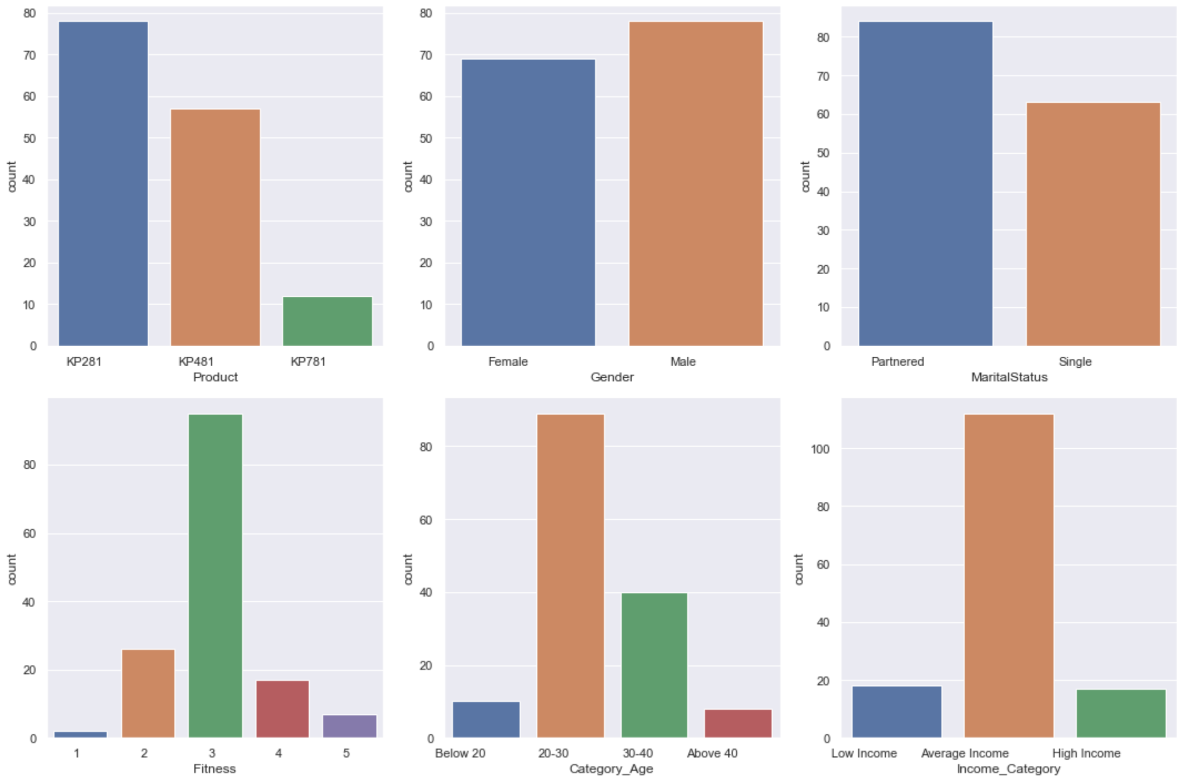
```
cols = 3
rows = 2
fig = plt.figure(figsize= (15,10))
all_cats = data1.select_dtypes(include='category')
for i, col in enumerate(all_cats):

    ax=fig.add_subplot(rows, cols, i+1)

    sns.countplot(x=data1[col], ax=ax)

    plt.xticks(ha='right')

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



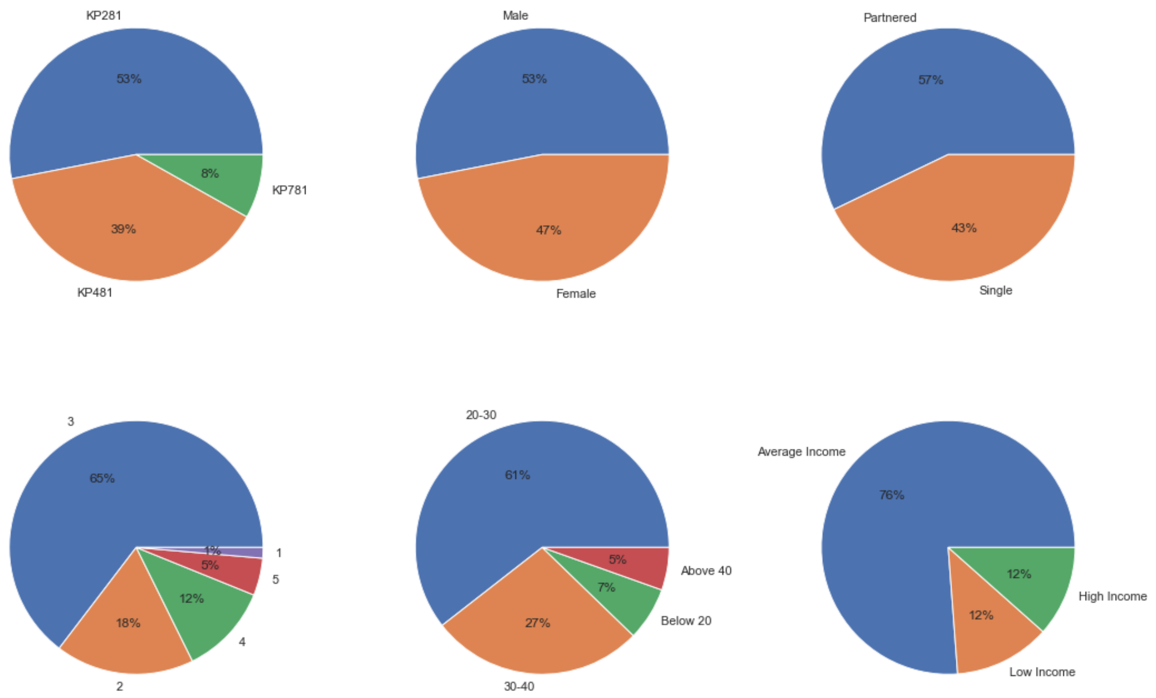
In [180]:

```
cols = 3
rows = 2
fig = plt.figure(figsize= (15,10))
all_cats = data1.select_dtypes(include='category')
for i, col in enumerate(all_cats):

    ax=fig.add_subplot(rows, cols, i+1)

    d=data1[col].value_counts()
    plt.pie(d,labels=d.index,autopct="%.0f%%")

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



Observations

- KP281 is the most sold product with a share of 53%
- Compared to singles,partnered people are most likely to buy treadmills
- Customers with fitness level 3 bought more treadmills
- Most treadmills were bought by people in 20-30 age interval
- Most treadmills were bought by people having income in range of 36000 to 60000

Check if features like marital status, age have any effect on the product purchased (using countplot, histplots, boxplots etc)

Bivariate Analysis

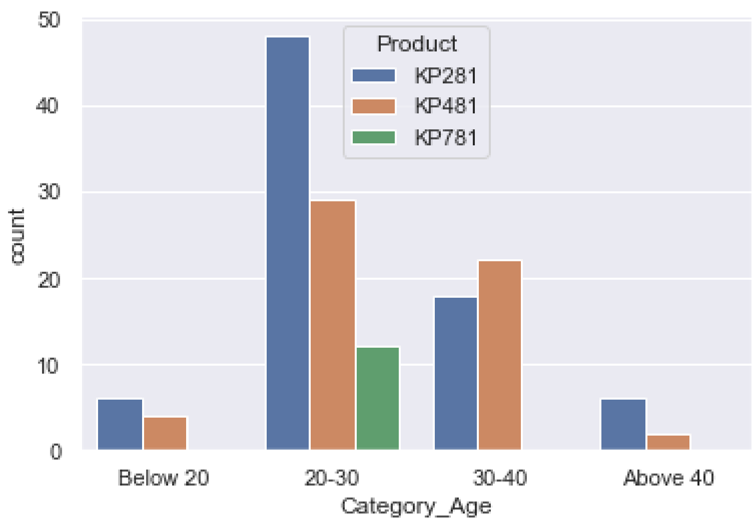
Category_Age vs Product

In [186]:

```
sns.countplot(data=data1,x='Category_Age',hue='Product')
```

Out[186]:

```
<AxesSubplot:xlabel='Category_Age', ylabel='count'>
```

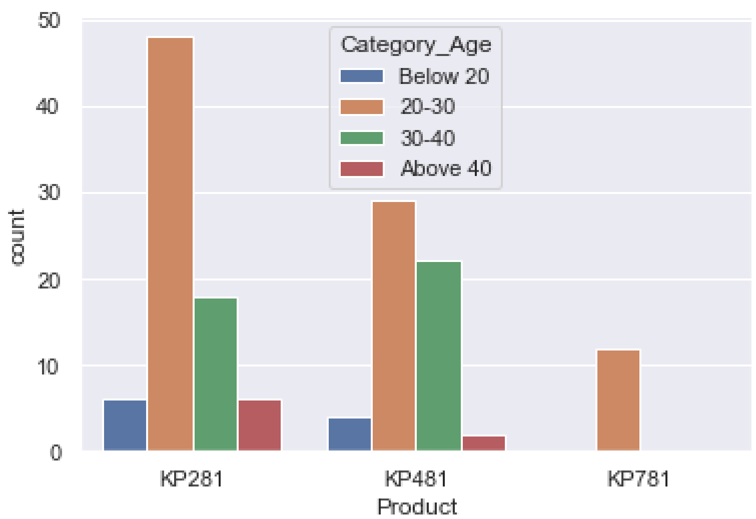


In [187]:

```
sns.countplot(data=data1,x='Product',hue='Category_Age')
```

Out[187]:

```
<AxesSubplot:xlabel='Product', ylabel='count'>
```



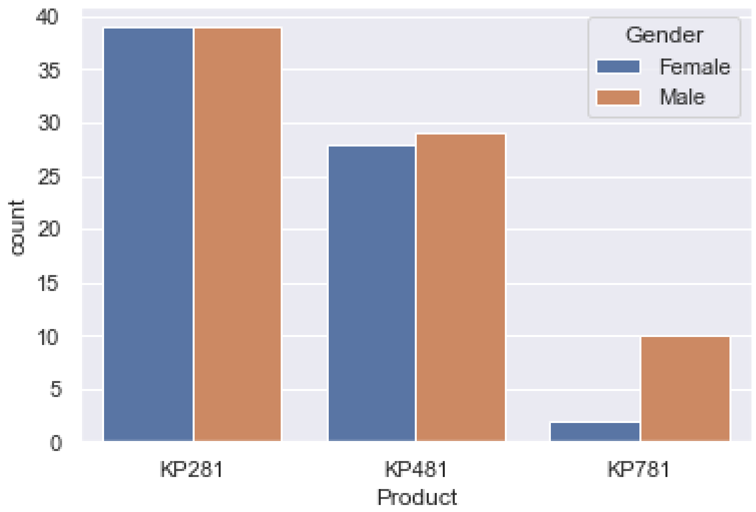
Gender vs Product

In [194]:

```
sns.countplot(data=data1,x='Product',hue='Gender')
```

Out[194]:

```
<AxesSubplot:xlabel='Product', ylabel='count'>
```

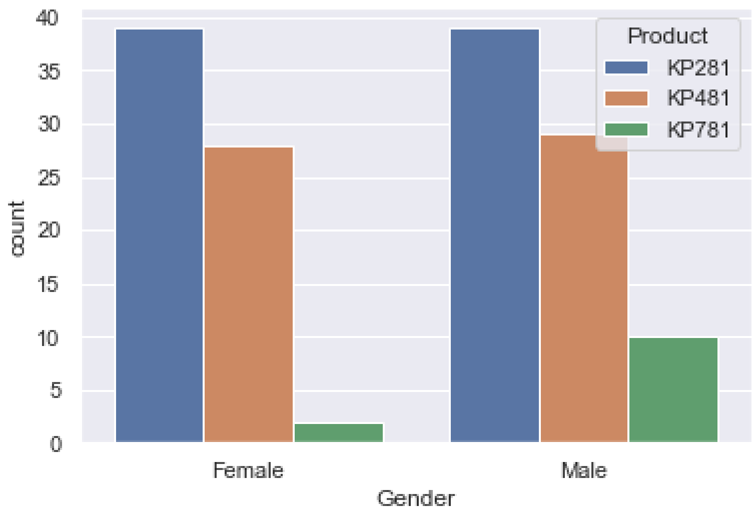


In [195]:

```
sns.countplot(data=data1,x='Gender',hue='Product')
```

Out[195]:

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```



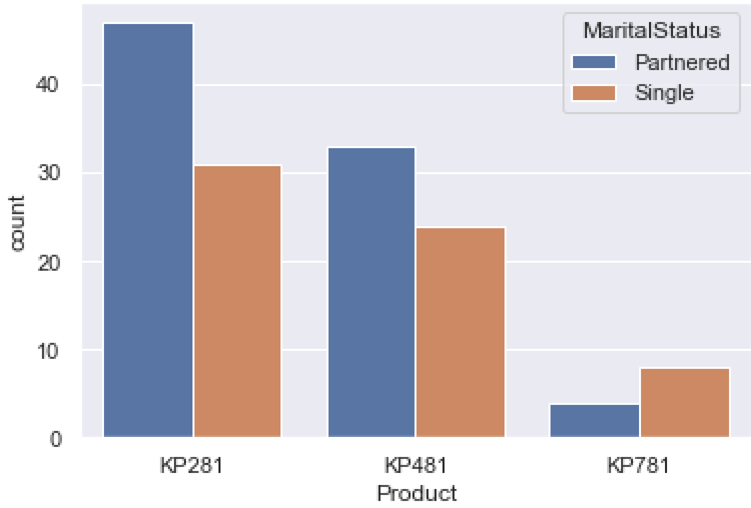
MaritalStatus vs Product

In [199]:

```
sns.countplot(data=data1,x='Product',hue='MaritalStatus')
```

Out[199]:

```
<AxesSubplot:xlabel='Product', ylabel='count'>
```

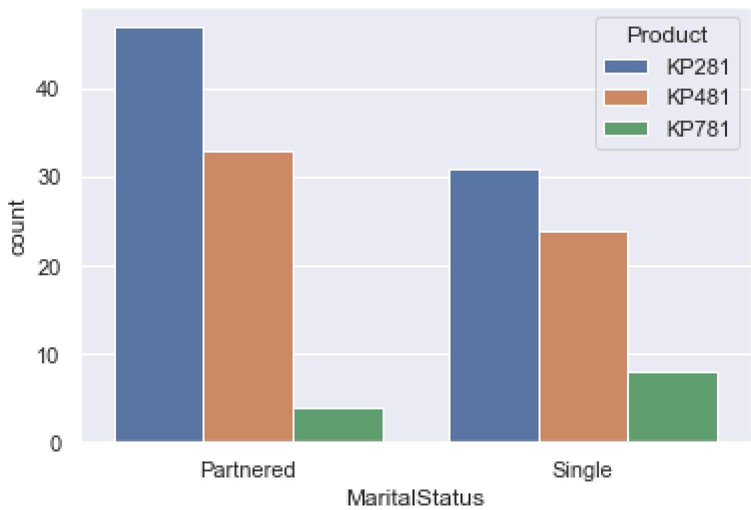


In [200]:

```
sns.countplot(data=data1,x='MaritalStatus',hue='Product')
```

Out[200]:

```
<AxesSubplot:xlabel='MaritalStatus', ylabel='count'>
```



In [ ]:

```

```

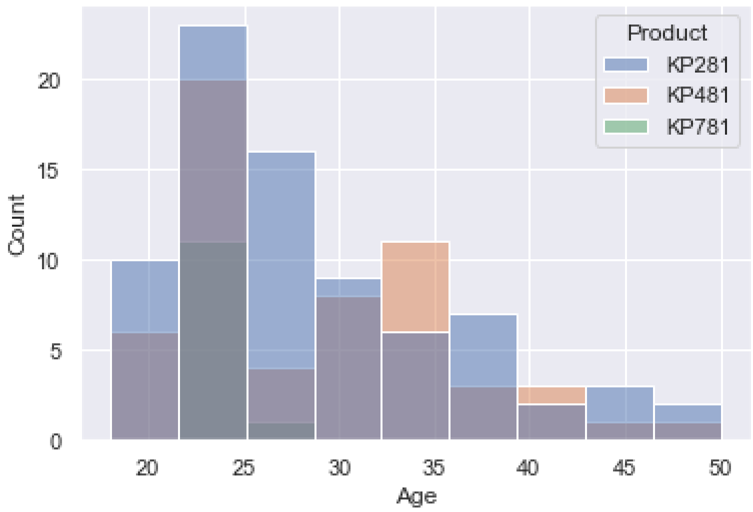
Numerical vs Product

In [204]:

```
sns.histplot(hue='Product',x='Age',data=data1)
```

Out[204]:

```
<AxesSubplot:xlabel='Age', ylabel='Count'>
```

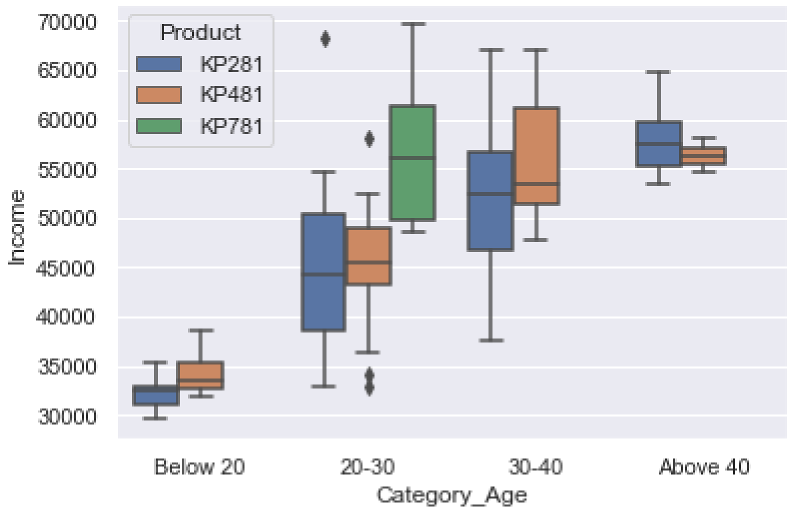


In [209]:

```
sns.boxplot(x='Category_Age',y='Income',hue='Product',data=data1)
```

Out[209]:

```
<AxesSubplot:xlabel='Category_Age', ylabel='Income'>
```



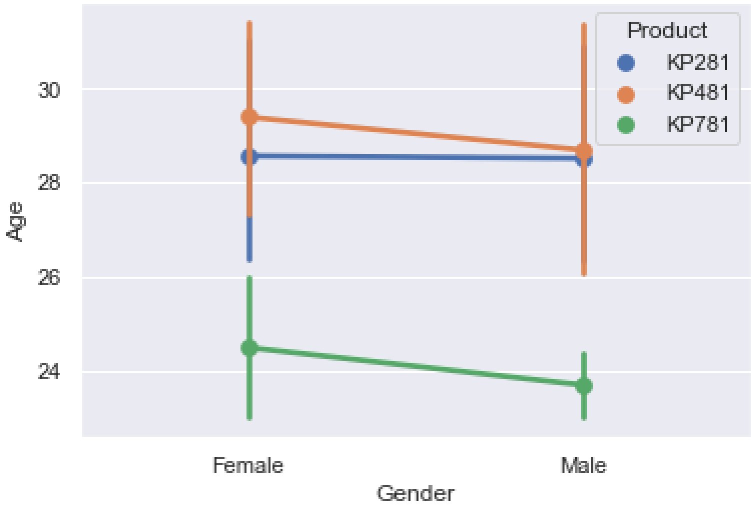
- KP281 model is only bought by people having age in between 20 to 30
- Age Below 20 are preferring to buy Kp481 than KP281

In [215]:

```
sns.pointplot(x=data1['Gender'],y=data1['Age'],hue=data1['Product'])
```

Out[215]:

```
<AxesSubplot:xlabel='Gender', ylabel='Age'>
```



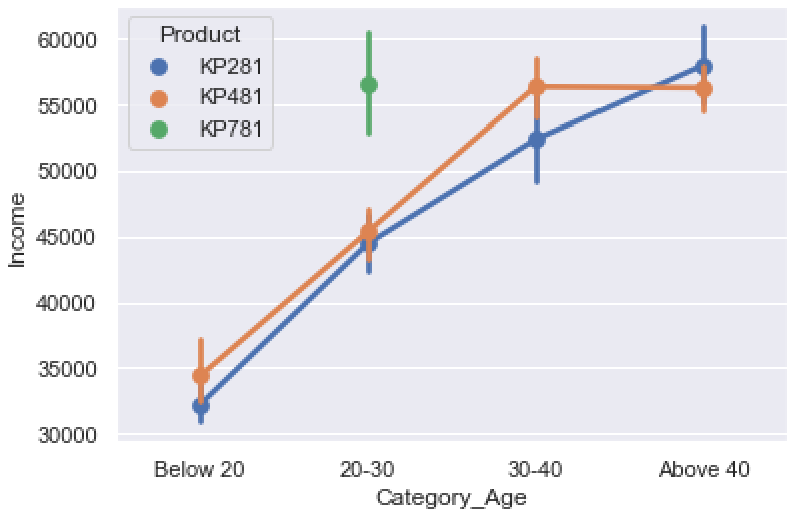
- KP 781 is only bought by people in between 20-30 and in those Men buy sooner than women
- In all the models men seem to buy treadmills earlier than women

In [208]:

```
sns.pointplot(x=data1['Category_Age'],y=data1['Income'],hue=data1['Product'])
```

Out[208]:

```
<AxesSubplot:xlabel='Category_Age', ylabel='Income'>
```

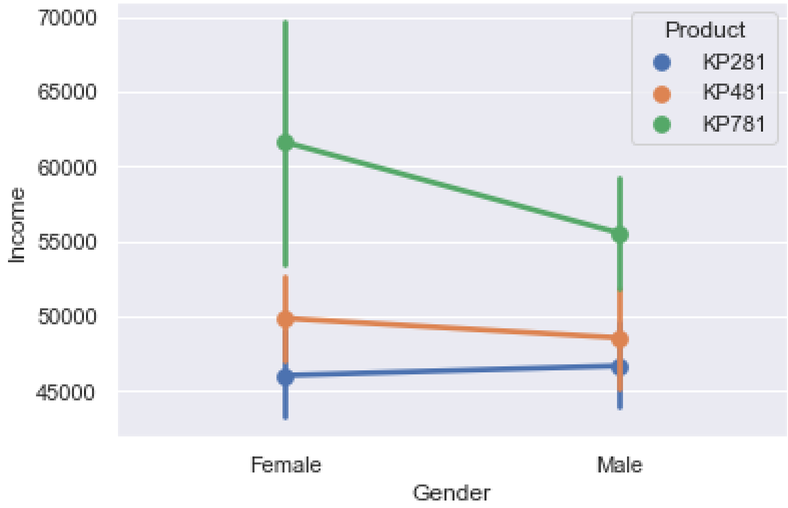


In [211]:

```
sns.pointplot(x=data1['Gender'],y=data1['Income'],hue=data1['Product'])
```

Out[211]:

```
<AxesSubplot:xlabel='Gender', ylabel='Income'>
```

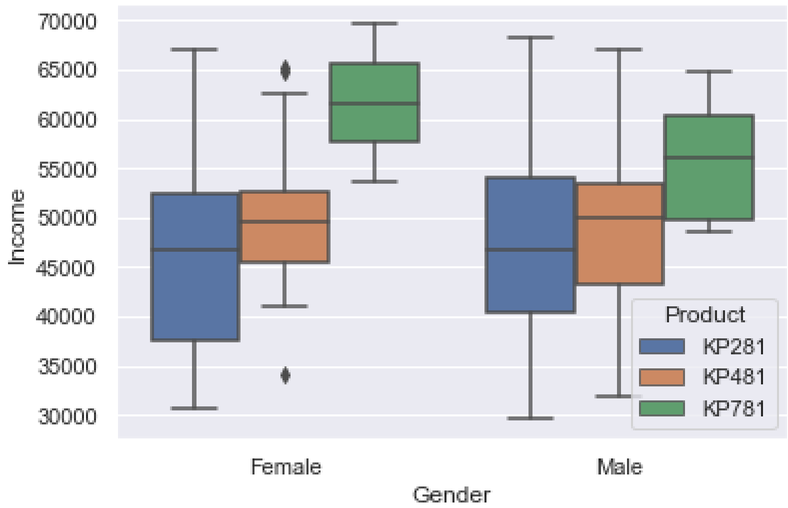


In [212]:

```
sns.boxplot(x='Gender',y='Income',hue='Product',data=data1)
```

Out[212]:

```
<AxesSubplot:xlabel='Gender', ylabel='Income'>
```



- Females with high income tend to buy KP781
- Low income Females bought more treadmills than males

Correlation Analysis

In [259]:

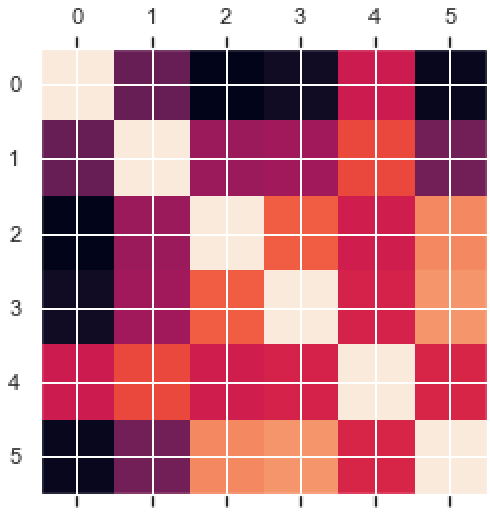
```
data2=data.copy()
```

In [260]:

```
data2.corr()
plt.matshow(data2.corr())
```

Out[260]:

```
<matplotlib.image.AxesImage at 0x18ab3acc7f0>
```

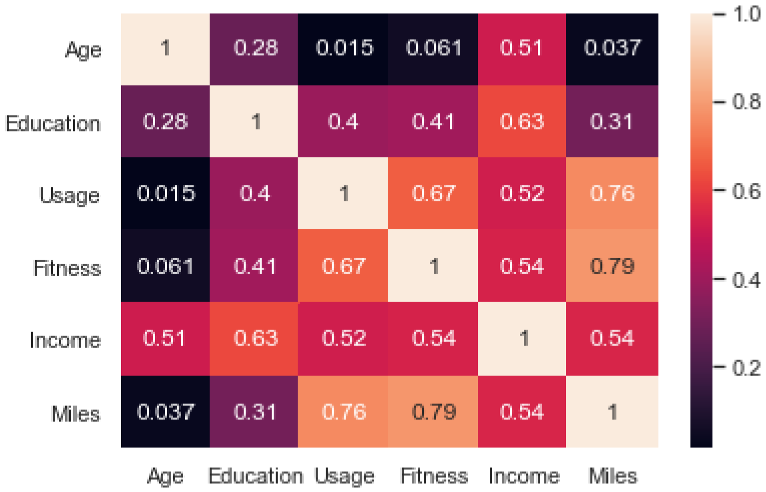


In [261]:

```
sns.heatmap(data2.corr(),annot=True)
```

Out[261]:

```
<AxesSubplot:>
```

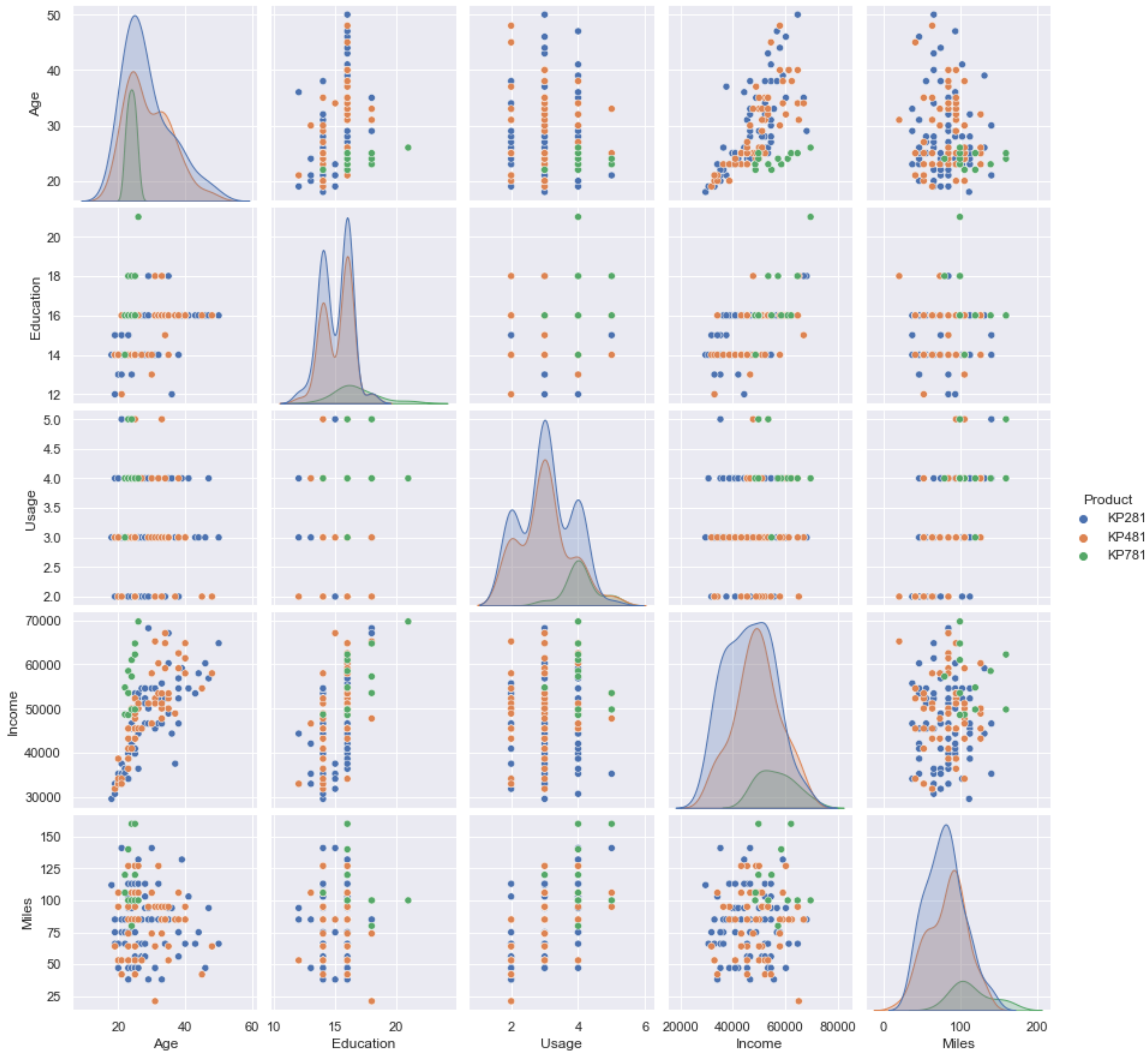


In [244]:

```
sns.pairplot(data1,hue='Product')
```

Out[244]:

<seaborn.axisgrid.PairGrid at 0x18ab052a7f0>



calculating Probabilities using crosstab

1)Between Product and Age_Category

In [245]:

```
pd.crosstab(index=data1['Product'],columns=[data1['Income_Category']],margins=True)
```

Out[245]:

Income_Category	Low Income	Average Income	High Income	All
Product				
KP281	13	59	6	78
KP481	5	45	7	57
KP781	0	8	4	12
All	18	112	17	147

- Marginal Probability
- Percentage of high income people bought treadmill

In [248]:

```
(17/147)*100   # in this way we can calculate marginal probabilities of all models
```

Out[248]:

11.564625850340136

In [250]:

```
# if we want to calculate conditional probability then choose the particular element in tha
# P(KP281/Average Income)
59/112  # p(A/B)=p(A,b)/p(B)==>P(A,B)=59,P(B)=112
```

Out[250]:

0.5267857142857143

In [255]:

```
pd.crosstab(index=data1['Product'],columns=[data1['Income_Category']],margins=True,normaliz
```

Out[255]:

Income_Category	Low Income	Average Income	High Income	All
Product				
KP281	0.722222	0.526786	0.352941	0.530612
KP481	0.277778	0.401786	0.411765	0.387755
KP781	0.000000	0.071429	0.235294	0.081633

In [256]:

```
## This normalize gives the probabilities directly
```

In [258]:

```
# In this way we can calculate marginal and conditional probabilities
```

In [ ]:

```

```

Conclusion

- KP-281 is the most sold product with 53% share in the market
- Most treadmills were bought by people having income in range of 36000 to 60000
- Most treadmills were bought by people in 20-30 age interval
- In all the models men seem to buy treadmills in the earlier age than women
- Customers with fitness level 3 bought more treadmills
- Compared to singles,partnered people are most likely to buy treadmills

Recommendations

- Aerofit should focus on the average income persons i.e person earning from 36000 to 60000 because they constitute 76% share
- Aerofit should bring models like KP-281 and KP-481 more with upgrades such that they should encourage persons from age 20-30 to buy.
- KP-781 should be introduced as premium model.
- The buying rate of females is same as male but in premium model the females rte is less,so aerofit should do reasearch in that area such that females use it more
- Aerofit should focus on sports persons because they have fitness rating above 3.
- They should concentrate on married people as well as singles but more focus should be on Married people.

*KP-281

- This model should be produced more because it is the trending and most sold
- Average Income people prefer this more

*KP-481

- This is should be upgrded more such that it should also be used by high income people
- These are mainly used by people aging 20-30

*KP-781

- This should be made as a premium model
- This model should be encouraged to be used by many local stars

In [ ]: