```
# Import the libraries that will be used
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
!gdown 16a8HTBiD48ecCj9vXfmPI94-6cA_SVMz
```

```
Downloading...
From: https://drive.google.com/uc?id=16a8HTBiD48ecCj9vXfmPI94-6cA_SVMz
To: /content/aerofit_treadmill.txt
100% 7.28k/7.28k [00:00<00:00, 18.1MB/s]
```

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

```
#explore the data
```

```
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

```
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
```

```
df.shape
```

```
(180, 9)
```

```
df.describe(include='object').T
```

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

```
df.describe(include='number').T
```

	count	mean	std	min	25%	50%	75%	max
Age	180.0	28.788889	6.943498	18.0	24.00	26.0	33.00	50.0
Education	180.0	15.572222	1.617055	12.0	14.00	16.0	16.00	21.0

```
#Product Portfolio
#The KP281 is an entry-level treadmill that sells for $1,500.
#The KP481 is for mid-level runners that sell for $1,750.
#The KP781 treadmill is having advanced features that sell for $2,500.
```

```
#we have no Null values
df.isnull().sum()
```

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

```
# Non Graphical Analysis
```

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

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

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

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

```
# Percentage Distribution of Products
df['Product'].value_counts(normalize=True)*100.0
```

```
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: Product, dtype: float64
```

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

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

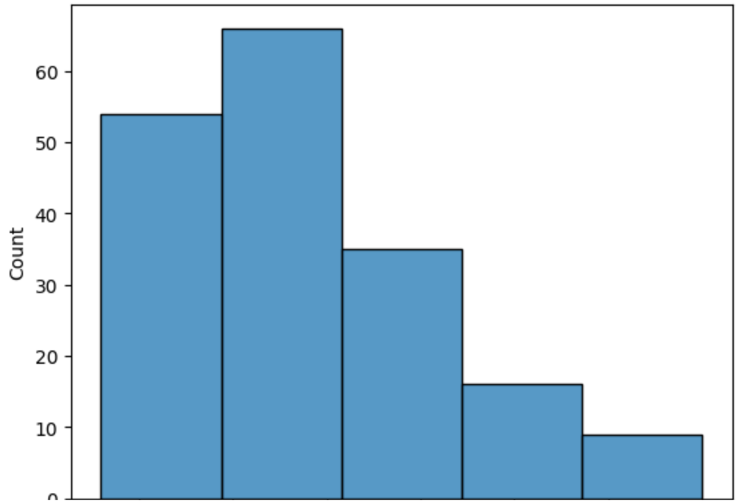
```
#Graphical Analysis
```

```
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

```
sns.histplot(df['Age'],bins=5)
```

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




```
# We have only few Customer below age 20
df.loc[df['Age']<20]
```

	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
80	KP481	19	Male	14	Single	3	3	31836	64

```
# We will create buckets based on Age
```

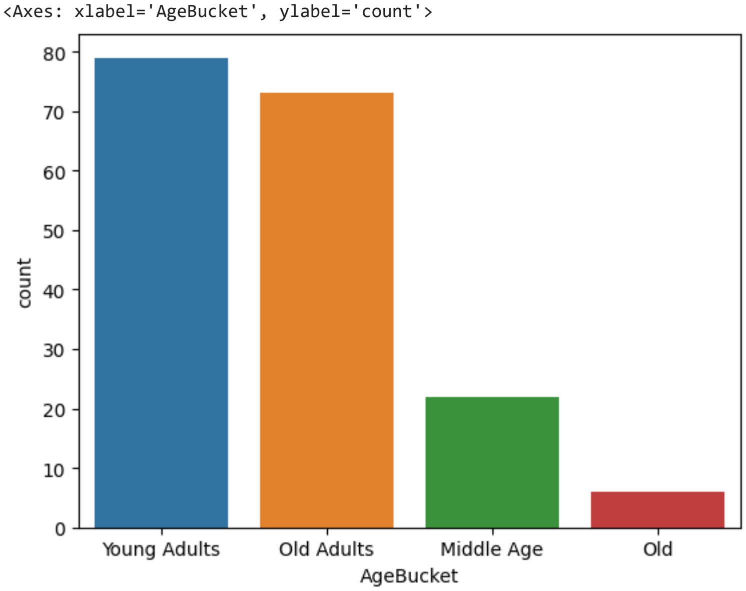
```
df['AgeBucket']=pd.cut(df['Age'],bins=[15,25,35,45,55],labels=['Young Adults','Old Adults','Middle Age','Old'])
```

```
df['AgeBucket'].value_counts()
```

```
Young Adults    79
Old Adults      73
Middle Age      22
Old              6
Name: AgeBucket, dtype: int64
```

```
sns.countplot(data=df,x='AgeBucket')
```

```
<Axes: xlabel='AgeBucket', ylabel='count'>
```



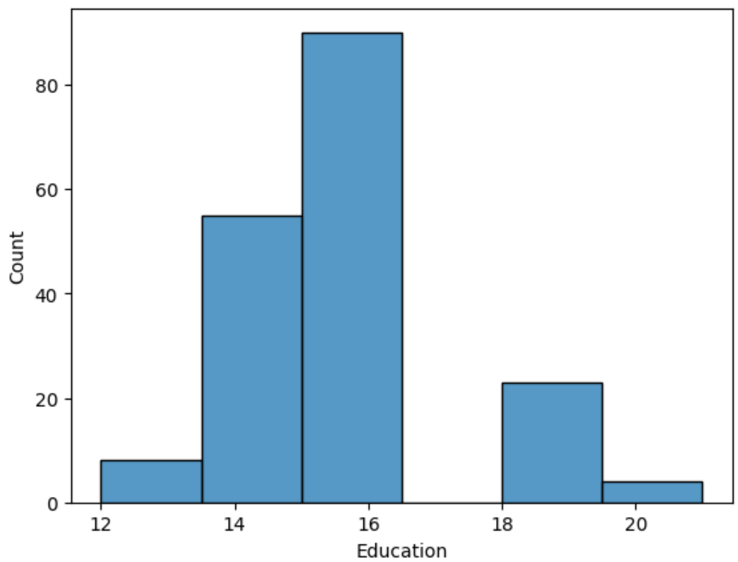
```
#Observation:
#We have most customers in the age range of 18-35
```

```
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeBucl
0	KP281	18	Male	14	Single	3	4	29562	112	You Ad
1	KP281	19	Male	15	Single	2	3	31836	75	You Ad
2	KP281	19	Female	14	Partnered	4	3	30699	66	You Ad

```
sns.histplot(df['Education'],bins=6)
```

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




```
df['Education'].value_counts()
```

```
16    85
14    55
18    23
15     5
13     5
12     3
21     3
20     1
Name: Education, dtype: int64
```

```
df['EducationLevel']=pd.cut(df['Education'],bins=[10,14,17,22],labels=['Basic','Intermediate','Advance'])
```

```
sns.countplot(data=df,x='EducationLevel',label=True)
```

```
<Axes: xlabel='EducationLevel', ylabel='count'>
```


```
#Observation:
#We have most customers in the Basic to Intermediate Education level
```

Basic Intermediate Advance

```
df['Usage'].value_counts()
```

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

```
df['UsageLevel']=pd.cut(df['Usage'],bins=[1,3,5,7],labels=['Low','Moderate','High'])
```

```
df.loc[df['UsageLevel']=='High']
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeB
154	KP781	25	Male	18	Partnered	6	4	70966	180	
155	KP781	25	Male	18	Partnered	6	5	75946	240	
162	KP781	28	Female	18	Partnered	6	5	92131	180	Old
163	KP781	28	Male	18	Partnered	7	5	77191	180	Old
164	KP781	28	Male	18	Single	6	5	88396	150	Old
166	KP781	29	Male	14	Partnered	7	5	85906	300	Old
167	KP781	30	Female	16	Partnered	6	5	90886	280	Old

```
df['UsageLevel'].value_counts()
```

```
Low         102
Moderate     69
High          9
Name: UsageLevel, dtype: int64
```

```
sns.countplot(data=df,x='UsageLevel')
```

```
<Axes: xlabel='UsageLevel', ylabel='count'>
```

```
#Observation: Most Customers purchase treadmile for a Low-Moderate Usage
```

```
df.head(5)
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeBucl
0	KP281	18	Male	14	Single	3	4	29562	112	You Ad
1	KP281	19	Male	15	Single	2	3	31836	75	You Ad
2	KP281	19	Female	14	Partnered	4	3	30699	66	You Ad

```
df['Fitness'].value_counts()
```

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

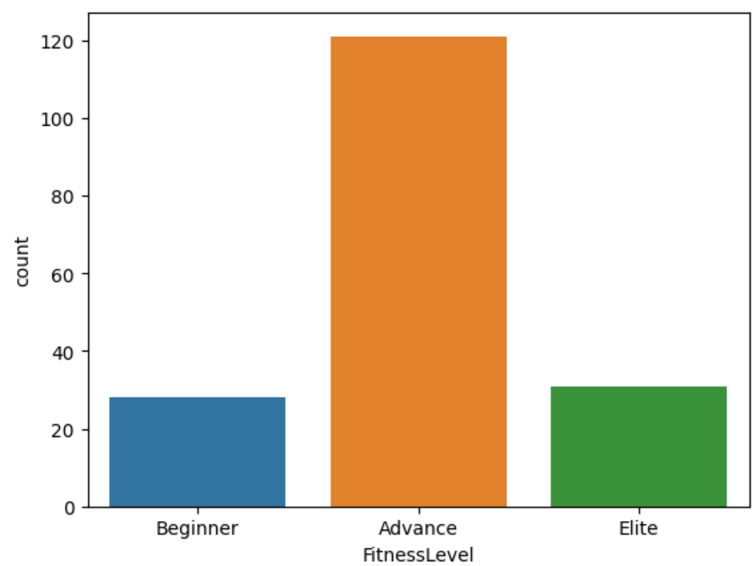
```
df['FitnessLevel']=pd.cut(df['Fitness'],bins=[0,2,4,5],labels=['Begineer','Advance','Elite'])
```

```
df['FitnessLevel'].value_counts()
```

```
Advance     121
Elite        31
Begineer     28
Name: FitnessLevel, dtype: int64
```

```
sns.countplot(data=df,x='FitnessLevel')
```

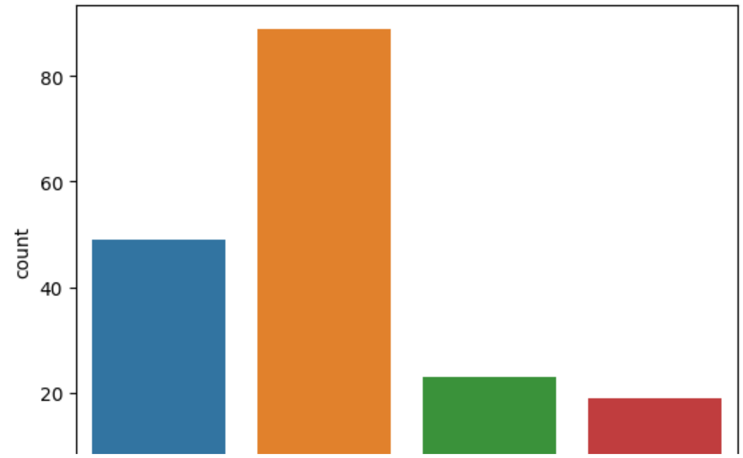
```
<Axes: xlabel='FitnessLevel', ylabel='count'>
```




```
df['IncomeLevel']=pd.cut(df['Income'],bins=[25000,45000,60000,80000,150000],labels=['Low','Moderate','High','Very High'])
```

```
#Income-Level Distribution of Customers
sns.countplot(data=df,x='IncomeLevel')
```

```
<Axes: xlabel='IncomeLevel', ylabel='count'>
```

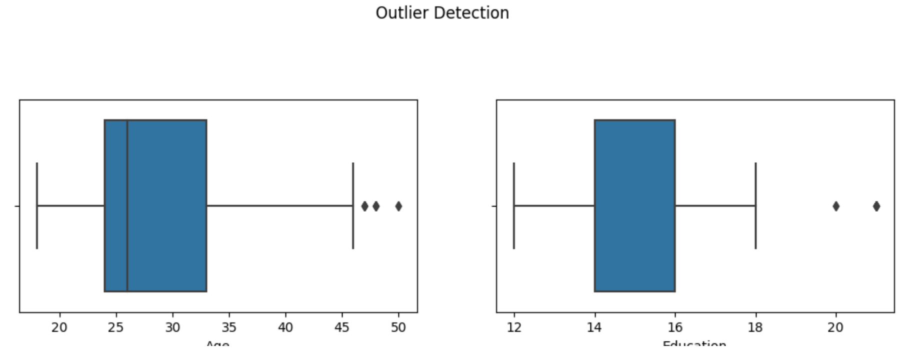



```
#Observation: Most Customers are Advance level of fitness, we have equal distribution of Begineer and Elite customers aswell
```

```
# We will check for outliers using Box Plot
```

```
plt.figure(figsize=(12,10))
plt.subplot(3,2,1)
plt.suptitle('Outlier Detection')
sns.boxplot(data=df,x='Age')
plt.subplot(3,2,2)
sns.boxplot(data=df,x='Education')
plt.subplot(3,2,3)
sns.boxplot(data=df,x='Usage')
plt.subplot(3,2,4)
sns.boxplot(data=df,x='Fitness')
plt.subplot(3,2,5)
sns.boxplot(data=df,x='Income')
plt.subplot(3,2,6)
sns.boxplot(data=df,x='Miles')
```

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




```
#Observation: We see a lot of outliers in Income and Miles
```

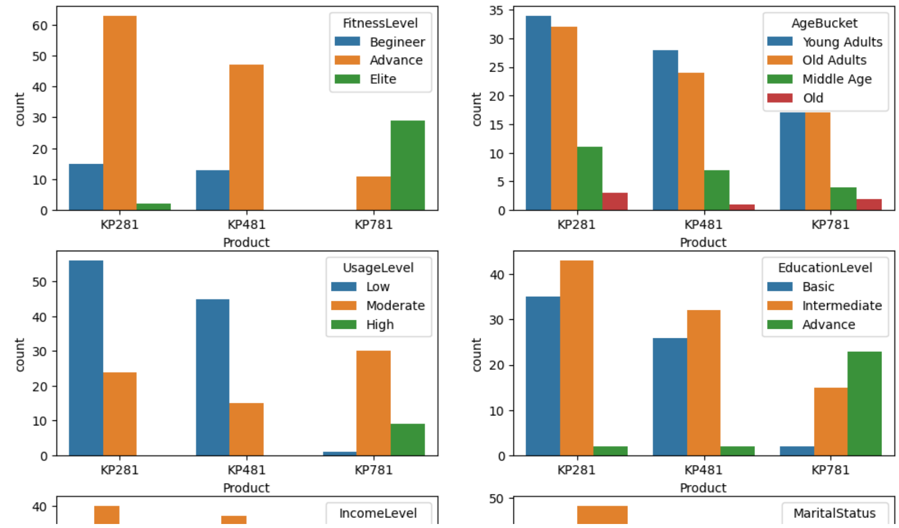
```
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeBucl
0	KP281	18	Male	14	Single	3	4	29562	112	You Ad
1	KP281	19	Male	15	Single	2	3	31836	75	You Ad
2	KP281	19	Female	14	Partnered	4	3	30699	66	You Ad

```
#Product vs Fitness Level
#Product vs Age
#Product vs Usage
#Product vs Education Level
#Product vs Income
#Product vs Marital Status


plt.figure(figsize=(12,10))
plt.subplot(3,2,1)
sns.countplot(data=df,x='Product',hue='FitnessLevel')
plt.subplot(3,2,2)
sns.countplot(data=df,x='Product',hue='AgeBucket')
plt.subplot(3,2,3)
sns.countplot(data=df,x='Product',hue='UsageLevel')
plt.subplot(3,2,4)
sns.countplot(data=df,x='Product',hue='EducationLevel')
plt.subplot(3,2,5)
sns.countplot(data=df,x='Product',hue='IncomeLevel')
plt.subplot(3,2,6)
sns.countplot(data=df,x='Product',hue='MaritalStatus')
```

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




```
#Observations:
#1.Only the Elite level and handfull of Advanced customers prefer the expensive Treadmile(KP781), else other customers prefer the lesser expe
#2.We see a similar distribution for all three products with respect to age group
#3.Only the customers with High-Moderate usage expectations prefer the expensive Treadmile(KP781), else other customers prefer the lesser exp
#4.Only Customers with Advance level of education prefer the expensive Treadmile(KP781), we see similar distribution for KP281 & KP481
#5.Customers with very high Income only purchase the KP781, we see similar distribution for KP281 & KP481
#6.We see a similar distribution for all three products with respect to Marital Status
```

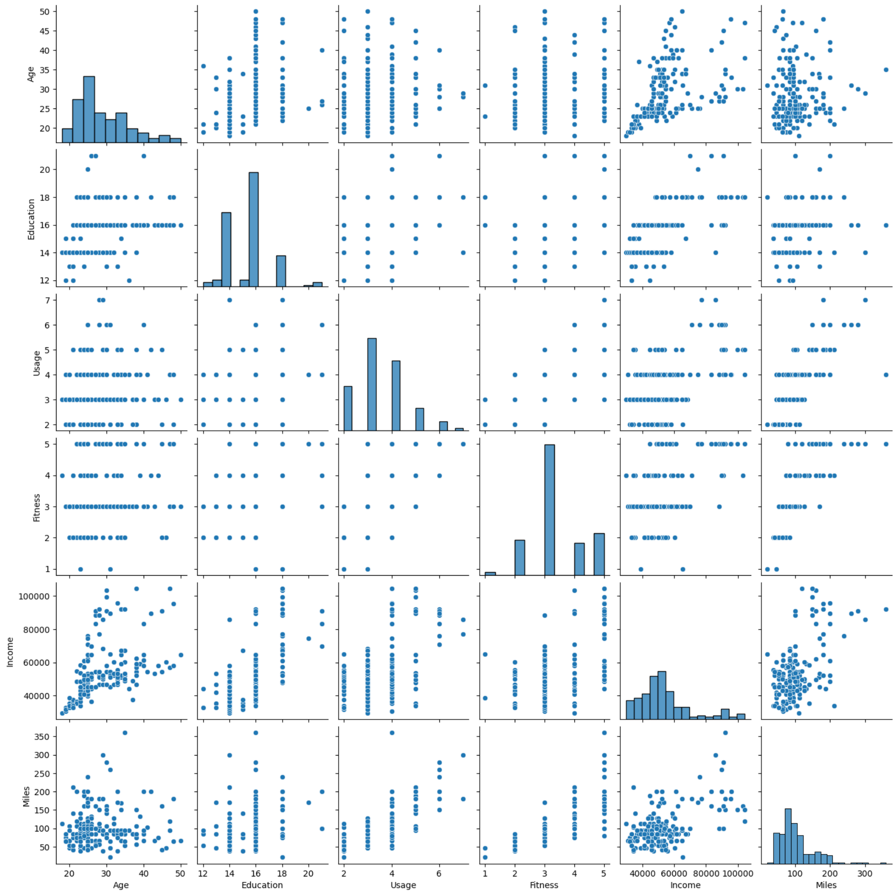
Product Product

```
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeBucl
0	KP281	18	Male	14	Single	3	4	29562	112	You Ad
1	KP281	19	Male	15	Single	2	3	31836	75	You Ad
2	KP281	19	Female	14	Partnered	4	3	30699	66	You Ad

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7ec255127e80>
```




```
df.head(5)
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeBucl
0	KP281	18	Male	14	Single	3	4	29562	112	You Ad
1	KP281	19	Male	15	Single	2	3	31836	75	You Ad
2	KP281	19	Female	14	Partnered	4	3	30699	66	You Ad

```
#We will create Contingency Tables to compute Marginal and Conditional Probability
```

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

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

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

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

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

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

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

	Product	KP281	KP481	KP781
Gender	**MaritalStatus**			
Female	**Partnered**	0.586957	0.326087	0.086957
	Single	0.433333	0.466667	0.100000
Male	**Partnered**	0.344262	0.344262	0.311475
	Single	0.441860	0.232558	0.325581
All		0.444444	0.333333	0.222222

```
pd.crosstab([df['AgeBucket'],df['FitnessLevel']],df['Product'],margins=True,normalize='index')
```

	Product	KP281	KP481	KP781
AgeBucket	**FitnessLevel**			
Young Adults	**Begineer**	0.500000	0.500000	0.000000
	Advance	0.500000	0.403846	0.096154
	Elite	0.076923	0.000000	0.923077
Old Adults	**Begineer**	0.583333	0.416667	0.000000
	Advance	0.500000	0.395833	0.104167
	Elite	0.076923	0.000000	0.923077
Middle Age	**Begineer**	0.000000	1.000000	0.000000
	Advance	0.611111	0.333333	0.055556
	Elite	0.000000	0.000000	1.000000
Old	**Begineer**	1.000000	0.000000	0.000000
	Advance	0.666667	0.333333	0.000000
	Elite	0.000000	0.000000	1.000000
All		0.444444	0.333333	0.222222

```
pd.crosstab([df['IncomeLevel'],df['FitnessLevel']],df['Product'],margins=True,normalize='index')
```

	Product	KP281	KP481	KP781
IncomeLevel	FitnessLevel			
Low	Begineer	0.545455	0.454545	0.000000
	Advance	0.729730	0.270270	0.000000
	Elite	1.000000	0.000000	0.000000
Moderate	Begineer	0.533333	0.466667	0.000000
	Advance	0.484375	0.484375	0.031250
	Elite	0.100000	0.000000	0.900000
High	Begineer	0.500000	0.500000	0.000000
	Advance	0.312500	0.375000	0.312500

```
pd.crosstab([df['IncomeLevel'],df['UsageLevel']],df['Product'],margins=True,normalize='index')
```

	Product	KP281	KP481	KP781
IncomeLevel	UsageLevel			
Low	Low	0.611111	0.388889	0.000000
	Moderate	0.923077	0.076923	0.000000
Moderate	Low	0.500000	0.482143	0.017857
	Moderate	0.363636	0.333333	0.303030
High	Low	0.600000	0.400000	0.000000
	Moderate	0.000000	0.300000	0.700000
	High	0.000000	0.000000	1.000000
Very High	Moderate	0.000000	0.000000	1.000000
	High	0.000000	0.000000	1.000000
All		0.444444	0.333333	0.222222

```
pd.crosstab([df['FitnessLevel'],df['UsageLevel']],df['Product'],margins=True,normalize='index')
```

	Product	KP281	KP481	KP781
FitnessLevel	UsageLevel			
Begineer	Low	0.538462	0.461538	0.000000
	Moderate	0.500000	0.500000	0.000000
Advance	Low	0.560000	0.440000	0.000000
	Moderate	0.466667	0.311111	0.222222
	High	0.000000	0.000000	1.000000
Elite	Low	0.000000	0.000000	1.000000
	Moderate	0.090909	0.000000	0.909091
	High	0.000000	0.000000	1.000000
All		0.444444	0.333333	0.222222

```
"""Observations:
1.We can see from the crosstabs Probability of which treadmile the customer buys is highly dependent on the following factors- Income, Fitnes
2.We cannot infer much insight from probability distribution of Gender, Marital Status and Age"""
```