Aerofit Case Study

About 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.

Business Problem

The market research team at AeroFit wants to identify the characteristics of the target audience for each type of treadmill offered by the company, to provide a better recommendation of the treadmills to the new customers. The team decides to investigate whether there are differences across the product with respect to customer characteristics.

Lets Analyse the data

In [177…
```
# Importing libraries

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

In [178…
```
# Reading data set

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

In [179…
```
# checking shape of the data set

df.shape
```

Out[179]:
```
(180, 9)
```

We can see that there are 180 rows in the data with 9 columns

In [181…
```
# checking info of data

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
```

We can see that all the 9 column have 180 rows each there is no null values and the columns are of different data types like object and int64.

In [ ]:

In [182…
```
# describing the data and adding median as an extra stats

def describe(df, stats):
    d = df.describe()
    return pd.concat([d,df.reindex(d.columns, axis = 1).agg(stats)])

describe(df, ['median'])
```

Out[182]:

	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
median	26.000000	16.000000	3.000000	3.000000	50596.500000	94.000000

Here this the the description of the whole data, we can see there are not much difference between mean and median so we can conclude that there are not many outliers in the data

In [ ]:

In [187…
```
## data mapping

df['Selling Price'] = np.where(df['Product']=='KP281' , 1500 ,
                               np.where(df['Product']=='KP481' , 1750 ,
                                        np.where(df['Product']=='KP781' , 2500 ,'')))
```

In [188…
```
## data mapping

df['Level'] = np.where(df['Product']=='KP281' , 'Entry' ,
                       np.where(df['Product']=='KP481' , 'Mid' ,
                                np.where(df['Product']=='KP781' , 'Advance','')))
```

Data mapping is done, each product type is matched to its selling price and its category

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.

In [ ]:

In [153…
```
## checing null values
df.isnull().sum()
```

Out[153]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

As we can see there are no null values in the data.

In [ ]:

Checking for value count of each variable

In [157…
```
## value count for product

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

Out[157]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

In [158…
```
## Value count for Age

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

Out[158]:
```
25    25
23    18
24    12
26    12
28     9
35     8
33     8
30     7
38     7
21     7
22     7
27     7
31     6
34     6
29     6
20     5
40     5
32     4
19     4
48     2
37     2
45     2
47     2
46     1
50     1
18     1
44     1
43     1
41     1
39     1
36     1
42     1
Name: Age, dtype: int64
```

In [190…
```
## Value count for Gender

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

Out[190]:
```
Male      104
Female     76
Name: Gender, dtype: int64
```

In [191…
```
## Value count for Education

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

Out[191]:
```
16    85
14    55
18    23
15     5
13     5
12     3
21     3
20     1
Name: Education, dtype: int64
```

In [161…
```
## Value count for MaritalStatus

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

Out[161]:
```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

In [162…
```
## Value count for Usage

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

Out[162]:
```
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

In [192…
```
## Value count for Fitness

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

Out[192]:
```
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

In [165…
```
## Value count for Income

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

Out[165]:
```
45480    14
52302     9
46617     8
54576     8
53439     8
         ..
65220     1
55713     1
68220     1
30699     1
95508     1
Name: Income, Length: 62, dtype: int64
```

In [166…
```
## Value count for Miles

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

Out[166]:
```
85     27
95     12
66     10
75     10
47      9
106     9
94      8
113     8
53      7
100     7
180     6
200     6
56      6
64      6
127     5
160     5
42      4
150     4
38      3
74      3
170     3
120     3
103     3
132     2
141     2
280     1
260     1
300     1
240     1
112     1
212     1
80      1
140     1
21      1
169     1
188     1
360     1
Name: Miles, dtype: int64
```

As it is clearly shown there is no issue in the data on the basis of value counts these are the discrete variables with no null values

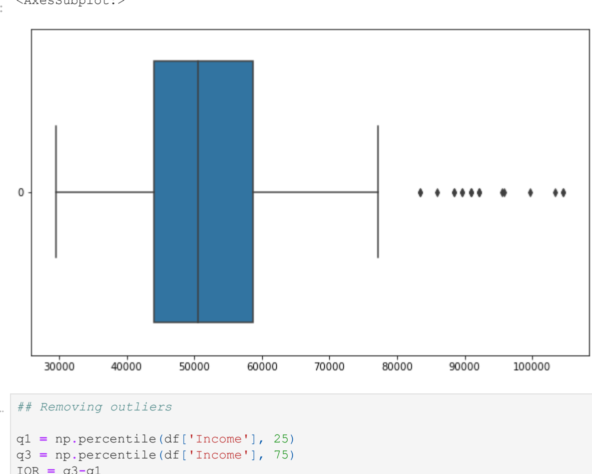
In [ ]:

Checking for Outliers

In [167…
```
## Box plot to check for outliers in Income

sns.boxplot(data = df['Income'], orient='h')
```

Out[167]: <AxesSubplot:>



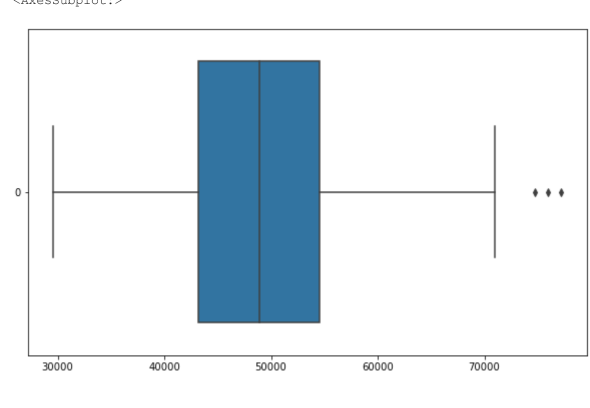
In [168…
```
## Removing outliers

q1 = np.percentile(df['Income'], 25)
q3 = np.percentile(df['Income'], 75)
IQR = q3-q1
min_val = q1 - 1.5*IQR
max_val = q3 + 1.5*IQR
df['Income'] = df['Income'].loc[(df['Income']>=min_val) & (df['Income']<=max_val)]
```

In [169…
```
## box plot after removing outliers from Income column

sns.boxplot(data = df['Income'], orient='h')
```

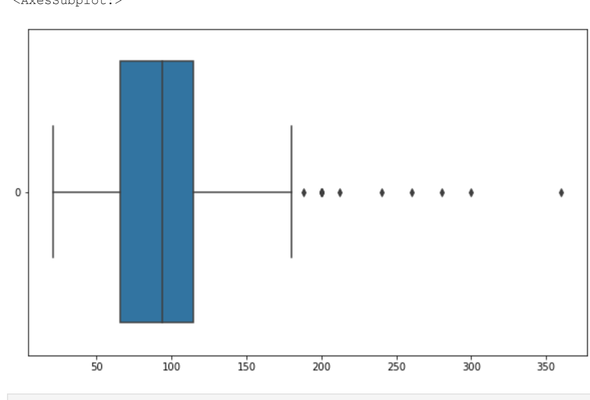
Out[169]: <AxesSubplot:>



In [170…
```
## Box plot to check for outliers in Miles

sns.boxplot(data = df['Miles'], orient='h')
```

Out[170]: <AxesSubplot:>



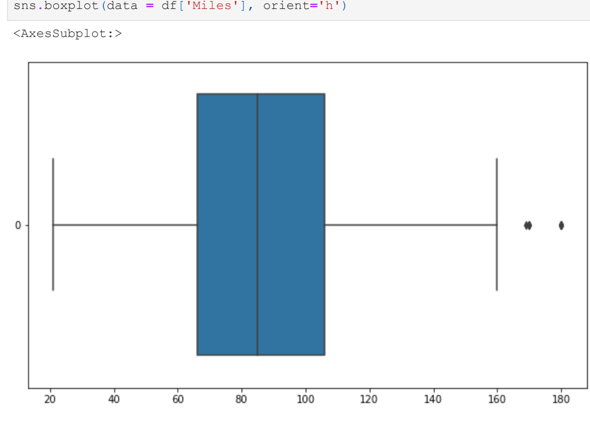
In [171…
```
## Removing Outliers

q1 = np.percentile(df['Miles'], 25)
q3 = np.percentile(df['Miles'], 75)
IQR = q3-q1
min_val = q1 - 1.5*IQR
max_val = q3 + 1.5*IQR
df['Miles'] = df['Miles'].loc[(df['Miles']>=min_val) & (df['Miles']<=max_val)]
```

In [172…
```
## box plot after removing outliers from Miles column

sns.boxplot(data = df['Miles'], orient='h')
```

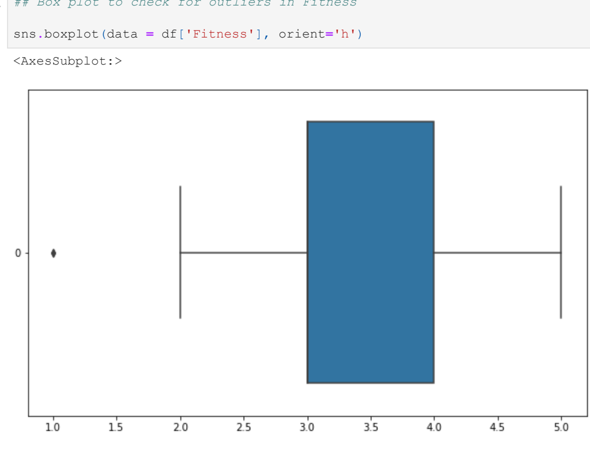
Out[172]: <AxesSubplot:>



In [173…
```
## Box plot to check for outliers in Fitness

sns.boxplot(data = df['Fitness'], orient='h')
```

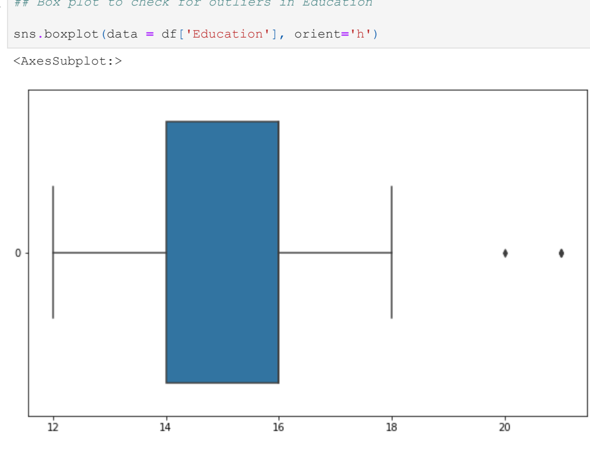
Out[173]: <AxesSubplot:>



In [174…
```
## Box plot to check for outliers in Education

sns.boxplot(data = df['Education'], orient='h')
```

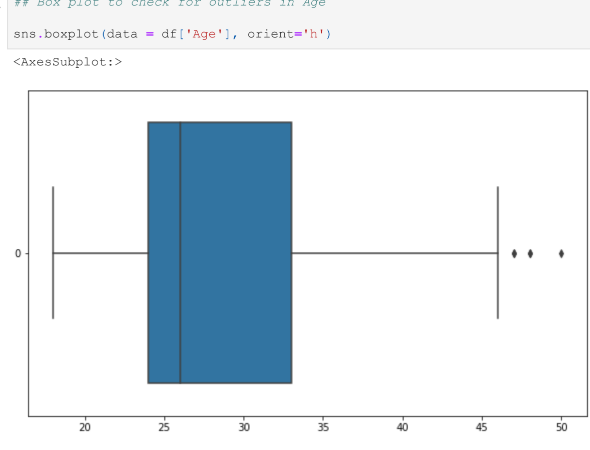
Out[174]: <AxesSubplot:>



In [175…
```
## Box plot to check for outliers in Age

sns.boxplot(data = df['Age'], orient='h')
```

Out[175]: <AxesSubplot:>



When the box plot is plotted for Education and miles some outliers are observed then we remove the outliers and agian box plot is plotted for the data now we see very less outliers and in rest of the columns there are negligible outliers

In [ ]:

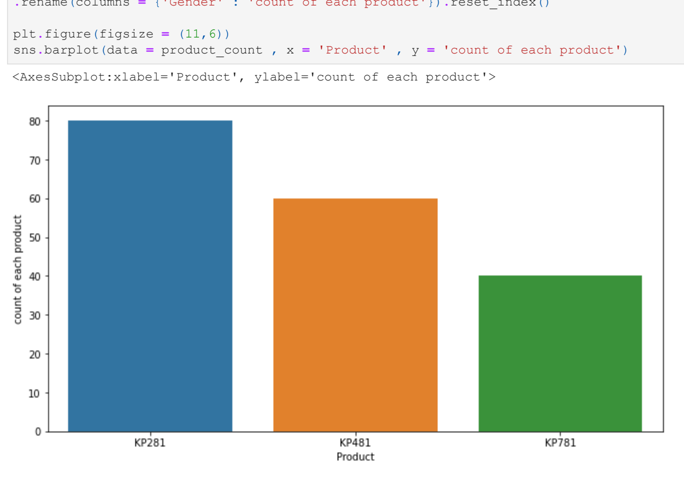
UNIVARIATE ANALYSIS

In [25]:
```
## count of each product

product_count = df.groupby(['Product'])\
.agg({'Gender' : 'count'})\
.sort_values('Gender', ascending = False)\
.rename(columns = {'Gender' : 'count of each product'}).reset_index()

plt.figure(figsize = (11,6))
sns.barplot(data = product_count, x = 'Product' , y = 'count of each product')
```

Out[25]: <AxesSubplot:xlabel='Product', ylabel='count of each product'>



As we can see the most purchased product is KP281 followed by KP481 and least purchased product is KP781

In [ ]:

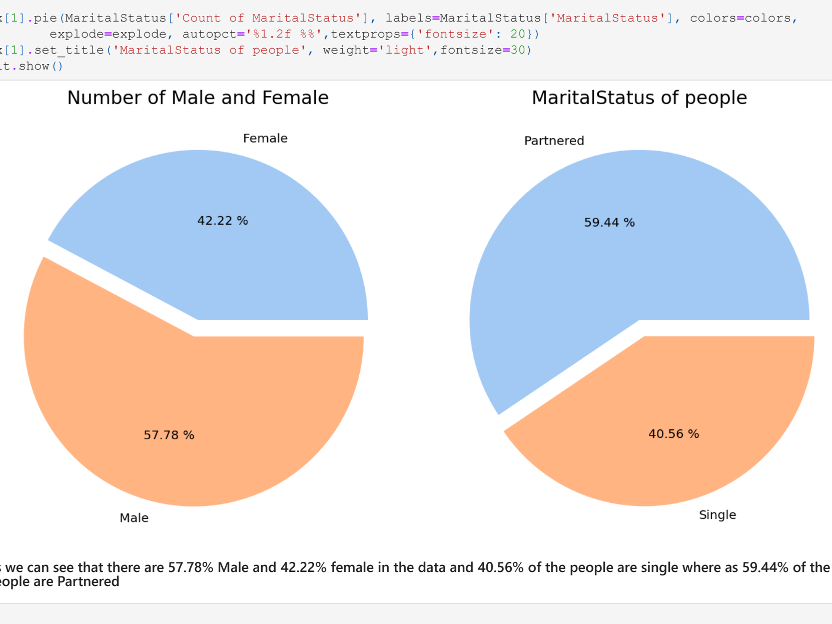
In [27]:
```
## pie plot for the gender and MaritalStatus of people

gender_count = df.groupby(['Gender'])\
.agg({'Product' : 'count'})\
.rename(columns = {'Product' : 'count of each gender'}).reset_index()
MaritalStatus = df.groupby(['MaritalStatus'])\
.agg({'Product' : 'count'})\
.rename(columns = {'Product' : 'Count of MaritalStatus'}).reset_index()

fig, ax = plt.subplots(1,2, figsize=(20, 10), tight_layout=True)
colors = sns.color_palette('pastel')
explode = [0, 0.1,]

ax[0].pie(gender_count['count of each gender'], labels=gender_count['Gender'], colors=colors,
        explode=explode, autopct='%1.2f %%',textprops={'fontsize': 20})
ax[0].set_title('Number of Male and Female', weight='light',fontsize=30)

ax[1].pie(MaritalStatus['Count of MaritalStatus'], labels=MaritalStatus['MaritalStatus'], colors=colors,
        explode=explode, autopct='%1.2f %%',textprops={'fontsize': 20})
ax[1].set_title('MaritalStatus of people', weight='light',fontsize=30)
plt.show()
```



As we can see that there are 57.78% Male and 42.22% female in the data and 40.56% of the people are single where as 59.44% of the people are Partnered

In [ ]:

Creating bins for the analysis of income, miles and age of people

In [195…
```
## Bins for Income

bins = [29000,50000,70000,90000,110000]
groups = ['29000-50000','50000-70000','70000-90000','90000-110000']
df['Income_range'] = pd.cut(df['Income'],bins,labels=groups)
```

In [196…
```
## Bins for Miles

bins = [20,70,120,170,220,270,320,370]
groups = ['20-70' , '70-120' , '120-170' ,'170-220' , '220-270' , '270-320' , '320-370']
df['Miles_range']  = pd.cut(df['Miles'] , bins , labels = groups)
```

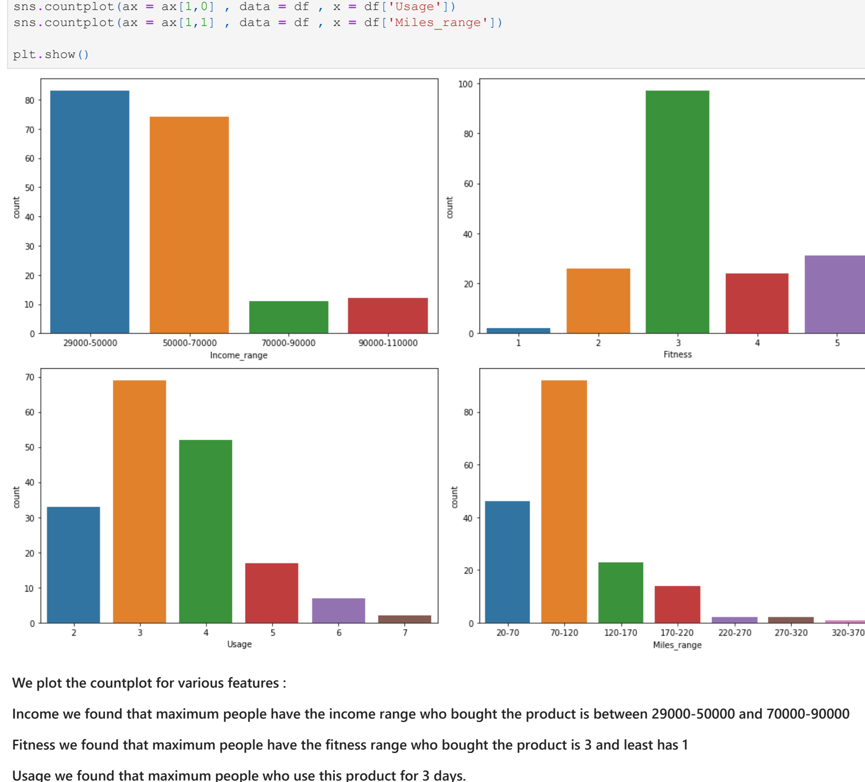
In [197…
```
## Bins for Age

bins = [15,20,25,30,35,40,45,50]
groups = ['15-20' , '20-25' , '25-30' ,'30-35' , '35-40' , '40-45' , '45-50']
df['Age_range']  = pd.cut(df['Age'] , bins , labels = groups)
```

In [198…
```
## plotting countplot for the analysis

fig, ax = plt.subplots(2,2, figsize=(15, 10), tight_layout=True)
sns.countplot(ax = ax[0,0] , data = df , x = df['Income_range'])
sns.countplot(ax = ax[0,1] , data = df , x = df['Fitness'])
sns.countplot(ax = ax[1,0] , data = df , x = df['Usage'])
sns.countplot(ax = ax[1,1] , data = df , x = df['Miles_range'])

plt.show()
```



We plot the countplot for various features :

Income we found that maximum people have the income range who bought the product is between 29000-50000 and 70000-90000

Fitness we found that maximum people have the fitness range who bought the product is 3 and least has 1

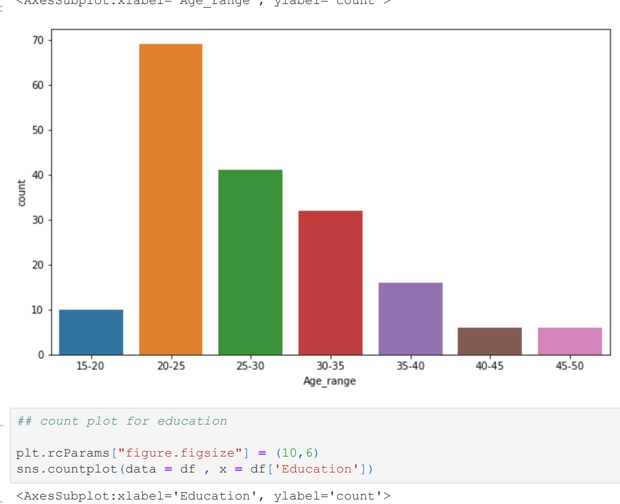
Usage we found that maximum people who use this product for 3 days.

Miles we found that among the people who bought this product maximum of people run 70-120 miles each week

In [199…
```
## count plot for age

plt.rcParams["figure.figsize"] = (10,6)
sns.countplot(data = df , x = df['Age_range'])
```

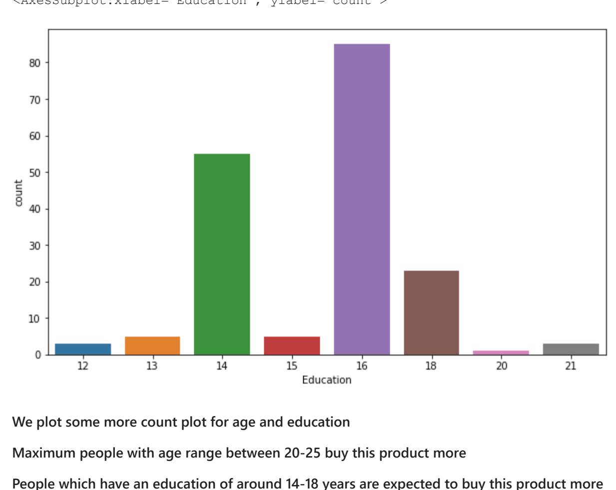
Out[199]: <AxesSubplot:xlabel='Age_range', ylabel='count'>



In [200…
```
## count plot for education

plt.rcParams["figure.figsize"] = (10,6)
sns.countplot(data = df , x = df['Education'])
```

Out[200]: <AxesSubplot:xlabel='Education', ylabel='count'>



We plot some more count plot for age and education

Maximum people with age range between 20-25 buy this product more

People which have an education of around 14-18 years are expected to buy this product more

In [201…
```
df.columns
```

Out[201]:
```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles', 'Level', 'Selling Price', 'Income_range',
       'Miles_range', 'Age_range'],
      dtype='object')
```

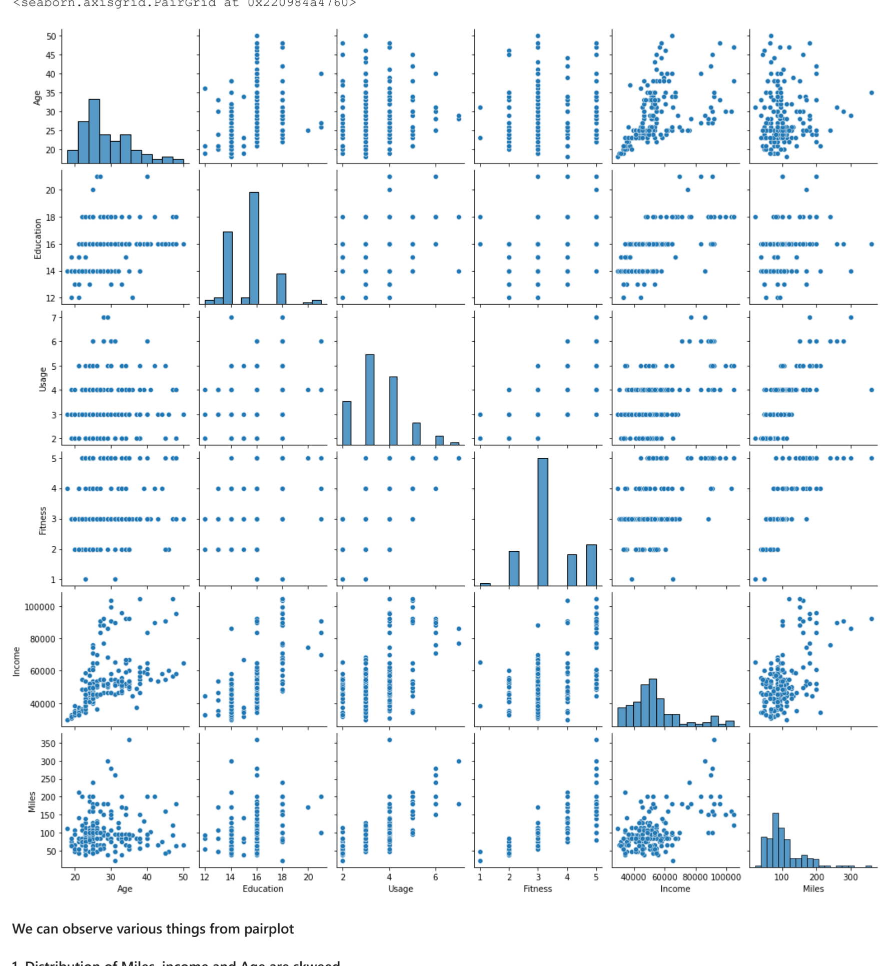
BIVARIATE ANALYSIS

In [202…
```
df1 = df.drop(['Income_range',
     'Miles_range', 'Age_range'],axis=1)
```

In [203…
```
## pair plot for bivariate analysis

sns.pairplot(df1)
```

Out[203]: <seaborn.axisgrid.PairGrid at 0x220984a4760>



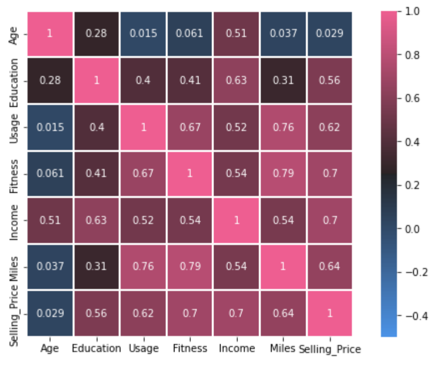
We can observe various things from pairplot

1. Distribution of Miles, income and Age are skweed

2. There is a positive corelation between Age and Income, Education and Income, Fitness and miles, usage and miles, fitness and usage

In [50]:
```
## heat map for to visualise correlation between two variables

corr = df1.corr(method = "pearson")
# mask = np.triu(np.ones_like(corr, dtype=bool))
cmap = sns.diverging_palette(250,354,80,60,center='dark',as_cmap=True)
sns.heatmap(corr, vmax=1, vmin=-.5, cmap=cmap, square=True, linewidths=.2,annot = True)
plt.show()
```



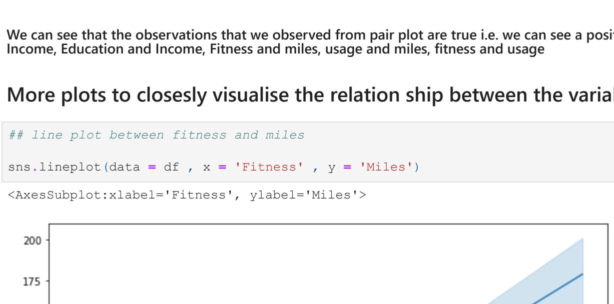
We can see that the observations that we observed from pair plot are true i.e. we can see a positive correlation between Age and Income, Education and Income, Fitness and miles, usage and miles, fitness and usage

More plots to closely visualise the relation ship between the variables

```python
## line plot between fitness and miles

sns.lineplot(data = df , x = 'Fitness' , y = 'Miles')
```

```
<AxesSubplot:xlabel='Fitness', ylabel='Miles'>
```

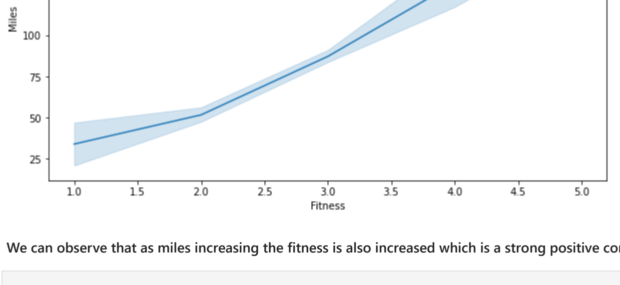


We can observe that as miles increasing the fitness is also increased which is a strong positive correlation

```python
## line plot between usage and miles

sns.lineplot(data = df , x = 'Usage' , y = 'Miles')
```

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

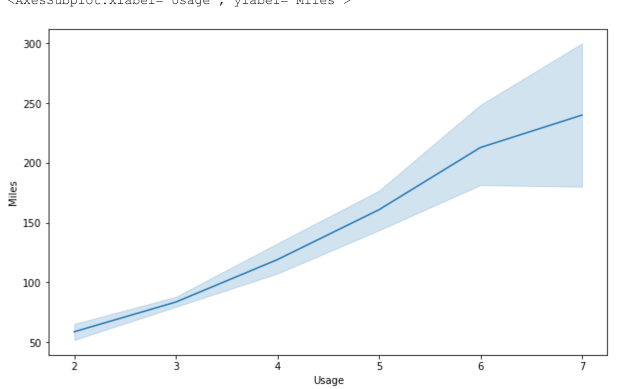


We can observe that as usage increasing the miles is also increased which is a strong positive correlation

```python
## line plot between Education and Income

sns.lineplot(data =df , x = 'Education' , y = 'Income')
```

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



We can observe that as Education increasing the Income is also increased which is a positive correlation

Marginal,Conditional Probability and Contigency Tables

Table for product vs age of people

```python
## Contigency table between product purchased and age of the people

product_and_age_columns = pd.crosstab(df['Product'] ,df['Age_range'],normalize = 'columns')
```

```python
product_and_age_columns.head()
```

Age_range Product	15-20	20-25	25-30	30-35	35-40	40-45	45-50
KP281	0.6	0.405797	0.512195	0.34375	0.500	0.500000	0.500000
KP481	0.4	0.347826	0.170732	0.53125	0.375	0.166667	0.166667
KP781	0.0	0.246377	0.317073	0.12500	0.125	0.333333	0.333333

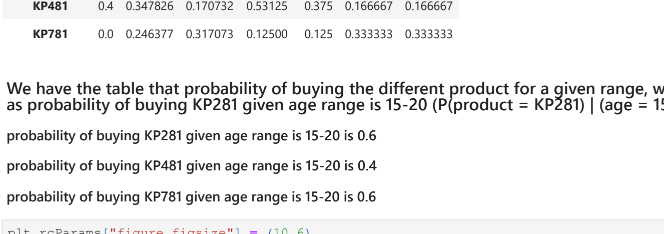
We have the table that probability of buying the different product for a given range, we can interpret this table as probability of buying KP281 given age range is 15-20 (P(product = KP281) | (age = 15-20)) for ex :

probability of buying KP281 given age range is 15-20 is 0.6

probability of buying KP481 given age range is 15-20 is 0.4

probability of buying KP781 given age range is 15-20 is 0.6

```python
plt.rcParams["figure.figsize"] = (10,6)
a = product_and_age_columns.plot(kind = 'bar')
a.legend(title='Age_Range', bbox_to_anchor=(1, 1.02),
         loc='upper left')
plt.show()
```



This plot depicts the probability of various product given age range

People with age 15-20 they tends to buy product : KP281

People with age 30-35 they tends to buy product : KP481

People with age 40-45 and 45-50 they tends to buy product : KP781

We can conclude this that aged people are more tends to buy KP281 because their income is more and they need a good quality of treadmill, where ever the young people whose age is between 15-20 or 20-30 more tends to buy KP481 because they have less income so they need a product which is in their budget, and KP781 is more preferable for the mid age ranged people.

Table for product vs Miles of people

```python
## Contigency table between product purchased and Miles_range of the people

product_and_miles_columns = pd.crosstab(df['Product'] ,df['Miles_range'], normalize = 'columns')
```

```python
product_and_miles_columns.head()
```

Miles_range Product	20-70	70-120	120-170	170-220	220-270	270-320	320-370
KP281	0.608696	0.500000	0.217391	0.071429	0.0	0.0	0.0
KP481	0.391304	0.369565	0.304348	0.071429	0.0	0.0	0.0
KP781	0.000000	0.130435	0.478261	0.857143	1.0	1.0	1.0

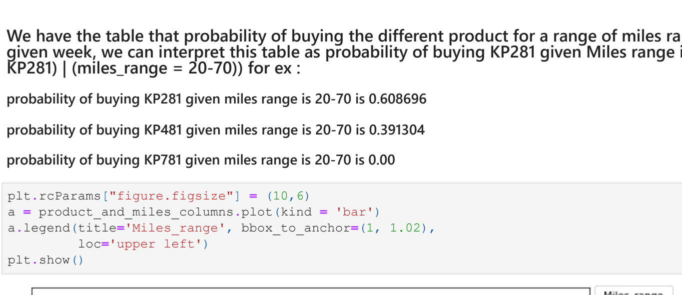
We have the table that probability of buying the different product for a range of miles ran by the people in a given week, we can interpret this table as probability of buying KP281 given Miles range is 20-70 (P(product = KP281) | (miles_range = 20-70)) for ex :

probability of buying KP281 given miles range is 20-70 is 0.608696

probability of buying KP481 given miles range is 20-70 is 0.391304

probability of buying KP781 given miles range is 20-70 is 0.00

```python
plt.rcParams["figure.figsize"] = (10,6)
a = product_and_miles_columns.plot(kind = 'bar')
a.legend(title='Miles_range', bbox_to_anchor=(1, 1.02),
         loc='upper left')
plt.show()
```



This plot depicts the probability of various product given miles

People with miles range 20-70 they tends to buy product : KP481 or KP281

People with miles range 70-120 they tends to buy product : KP481 or KP281

People with miles range 120-170, 220-270, 270-320 and 320-370 they tends to buy product : KP781

We can conclude this that the people who ran less miles per week are more tends to buy KP481 or KP281 because there usage is less due to which they dont tends to buy high price product, where ever the people who ran miles between 70-120 more tends to buy KP481 or KP281 because they have a normal usage of the product so they want a mid range product for them and KP781 is more preferable for the people who ran for more then 120 miles per week and it is a premium product so it will be more durable for them to buy this.

Table for product vs Income

```python
## Contigency table between product purchased and Income_range of the people

product_and_income_columns = pd.crosstab(df['Product'] ,df['Income_range'], normalize = 'columns')
```

```python
product_and_income_columns.head()
```

Income_range Product	29000-50000	50000-70000	70000-90000	90000-110000
KP281	0.578313	0.432432	0.0	0.0
KP481	0.361446	0.405405	0.0	0.0
KP781	0.060241	0.162162	1.0	1.0

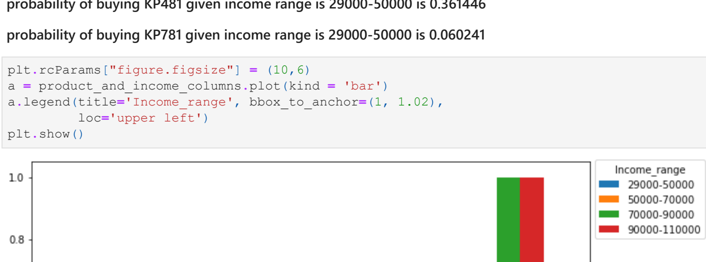
We have the table that probability of buying the different product for the income range of people, we can interpret this table as probability of buying KP281 given income range is 29000-50000 (P(product = KP281) | (miles_range = 29000-50000)) for ex :

probability of buying KP281 given income range is 29000-50000 is 0.578313

probability of buying KP481 given income range is 29000-50000 is 0.361446

probability of buying KP781 given income range is 29000-50000 is 0.060241

```python
plt.rcParams["figure.figsize"] = (10,6)
a = product_and_income_columns.plot(kind = 'bar')
a.legend(title='Income_range', bbox_to_anchor=(1, 1.02),
         loc='upper left')
plt.show()
```



This plot depicts the probability of various product given miles

People with income range is 29000-50000 they tends to buy product : KP281

People with income range is 50000-70000 they tends to buy product : KP481 or KP281

People with income range is 70000-90000, 90000-110000 they tends to buy product : KP781

We can conclude this that people with low income range will buy KP281, where as the people with intermediate income range will buy KP281 or KP481 where ever the people with income range more the 70000 will always buy KP781

Table for product vs Fitness

```python
## Contigency table between product purchased and Fitness of the people

product_and_fitness_columns = pd.crosstab(df['Product'] ,df['Fitness'], normalize = 'columns')
```

```python
product_and_fitness_columns.head()
```

Fitness Product	1	2	3	4	5
KP281	0.5	0.538462	0.556701	0.375000	0.064516
KP481	0.5	0.461538	0.402062	0.333333	0.000000
KP781	0.0	0.000000	0.041237	0.291667	0.935484

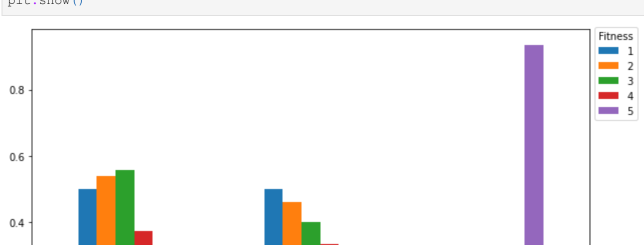
We have the table that probability of buying the different product for the given fitness of people, we can interpret this table as probability of buying KP281 given fitness of people is 1 (P(product = KP281) | (fitness=1)) for ex :

probability of buying KP281 given fitness is 1 is 0.5

probability of buying KP481 given fitness is 1 is 0.5

probability of buying KP781 given fitness is 1 is 0.0

```python
plt.rcParams["figure.figsize"] = (10,6)
a = product_and_fitness_columns.plot(kind = 'bar')
a.legend(title='Fitness', bbox_to_anchor=(1, 1.02),
         loc='upper left')
plt.show()
```



This plot depicts the probability of various product given miles

People with fitness is 1 they tends to buy product : KP281 or KP481

People with fitness is 2 they tends to buy product : KP281 or KP481

People with fitness is 3 they tends to buy product : KP281

People with fitness is 4 they tends to buy product : KP281 or KP481

People with fitness is 5 they tends to buy product : KP281 or KP781

We can conclude this that people with fitness between 1-4 tends to buy KP281 or KP481 and fitness as 5 will buy KP781

Table for product vs Usage

```python
## Contigency table between product purchased and Usage of the people

product_and_Usage_columns = pd.crosstab(df['Product'] ,df['Usage'], normalize = 'columns')
```

```python
product_and_Usage_columns.head()
```

Usage Product	2	3	4	5	6	7
KP281	0.575758	0.536232	0.423077	0.117647	0.0	0.0
KP481	0.424242	0.449275	0.230769	0.176471	0.0	0.0
KP781	0.000000	0.014493	0.346154	0.705882	1.0	1.0

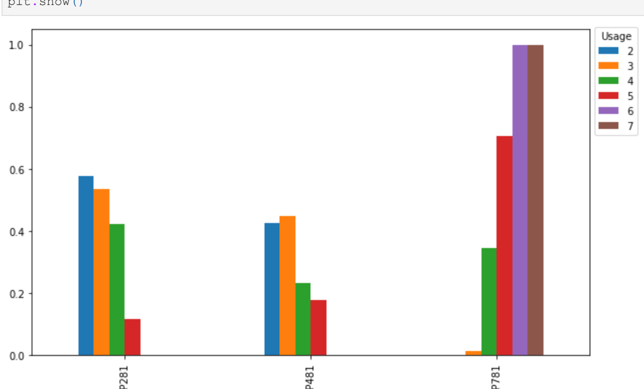
We have the table that probability of buying the different product for the given Usage of people, we can interpret this table as probability of buying KP281 given Usage of people is 2 (P(product = KP281) | (Usage=2)) for ex :

probability of buying KP281 given Usage is 2 is 0.575758

probability of buying KP481 given Usage is 2 is 0.424242

probability of buying KP781 given Usage is 2 is 0.000000

```python
plt.rcParams["figure.figsize"] = (10,6)
a = product_and_Usage_columns.plot(kind = 'bar')
a.legend(title='Usage', bbox_to_anchor=(1, 1.02),
         loc='upper left')
plt.show()
```



This plot depicts the probability of various product given miles

People with usage is 2 they tends to buy product : KP281

People with usage is 3 they tends to buy product : KP281 or KP481

People with usage is 4 they tends to buy product : KP281

People with usage is 5 they tends to buy product : KP781

People with usage is 6 they tends to buy product : KP781

People with usage is 7 they tends to buy product : KP781

We can conclude this that people with usage between 2-4 tends to buy KP281 or KP481 and usage above 4 will buy KP781.

Table for product vs MaritalStatus

```python
## Contigency table between product purchased and MaritalStatus of the people

product_and_MaritalStatus_columns = pd.crosstab(df['Product'] ,df['MaritalStatus'], normalize = 'columns')
```

```python
product_and_MaritalStatus_columns.head()
```

MaritalStatus Product	Partnered	Single
KP281	0.448598	0.438356
KP481	0.336449	0.328767
KP781	0.214953	0.232877

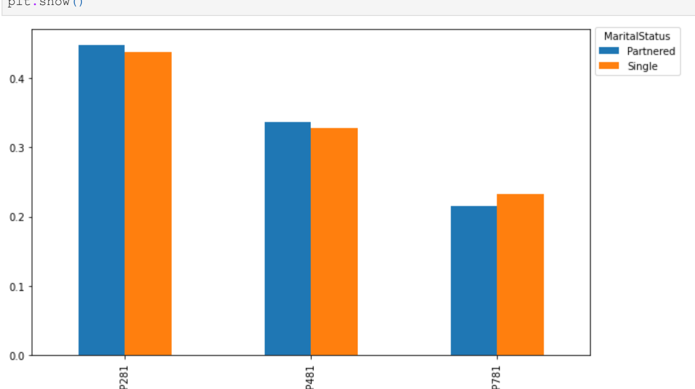
We have the table that probability of buying the different product for the given MaritalStatus of people, we can interpret this table as probability of buying KP281 given MaritalStatus of people is Partnered (P(product = KP281) | (MaritalStatus=Partnered)) for ex :

probability of buying KP281 given MaritalStatus is Partnered is 0.448598

probability of buying KP481 given MaritalStatus is Partnered is 0.336449

probability of buying KP781 given MaritalStatus is Partnered is 0.214953

```python
plt.rcParams["figure.figsize"] = (10,6)
a = product_and_MaritalStatus_columns.plot(kind = 'bar')
a.legend(title='MaritalStatus', bbox_to_anchor=(1, 1.02),
         loc='upper left')
plt.show()
```



This plot depicts the probability of various product given miles

People with MaritalStatus is Partnered they tends to buy product : KP281 or KP481 or KP781

People with MaritalStatus is Single they tends to buy product : KP281 or KP481 or KP781

We can conclude this that there is not much difference in the probability of buying the product on the MaritalStatus of the people.

Table for product vs Gender

```python
## Contigency table between product purchased and MaritalStatus of the people

product_and_Gender_columns = pd.crosstab(df['Product'] ,df['Gender'], normalize = 'columns')
```

```python
product_and_Gender_columns.head()
```

Gender Product	Female	Male
KP281	0.526316	0.384615
KP481	0.381579	0.298077
KP781	0.092105	0.317308

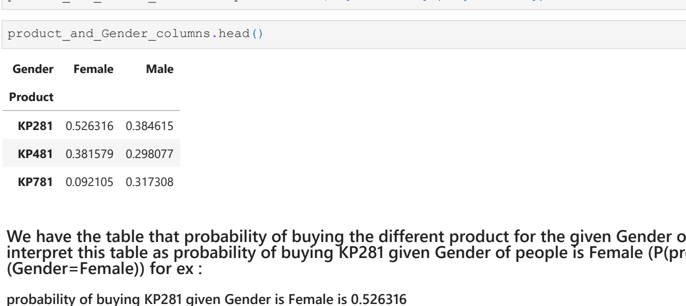
We have the table that probability of buying the different product for the given Gender of people, we can interpret this table as probability of buying KP281 given Gender of people is Female (P(product = KP281) | (Gender=Female)) for ex :

probability of buying KP281 given Gender is Female is 0.526316

probability of buying KP481 given Gender is Female is 0.381579

probability of buying KP781 given Gender is Female is 0.092105

```python
plt.rcParams["figure.figsize"] = (10,6)
a = product_and_Gender_columns.plot(kind = 'bar')
a.legend(title='Gender', bbox_to_anchor=(1, 1.02),
         loc='upper left')
plt.show()
```



This plot depicts the probability of various product given miles

People with Gender is Female they tends to buy product : KP281 or KP481

People with Gender is Male they tends to buy product : KP781

We can conclude this that Female is more probale to buy KP281 or KP481 where ever men is more probale to buy KP781 this is due to may be male will require more intense workout and due to that they will buy a more premium product and female will not require that much of intense workout so they prefer and low range or mid range product

Table for product vs Education

```python
product_and_Education_columns = pd.crosstab(df['Product'] ,df['Education'], normalize = 'columns')
```

```python
product_and_Education_columns.head()
```

Education Product	12	13	14	15	16	18	20	21
KP281	0.666667	0.6	0.545455	0.8	0.458824	0.086957	0.0	0.0
KP481	0.333333	0.4	0.418182	0.2	0.364706	0.086957	0.0	0.0
KP781	0.000000	0.0	0.036364	0.0	0.176471	0.826087	1.0	1.0

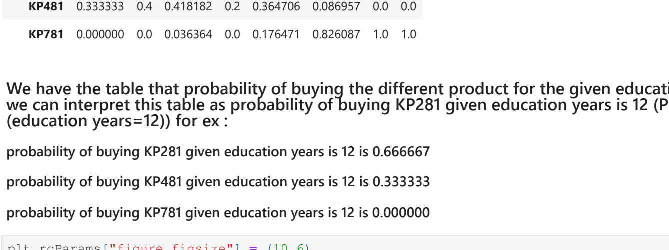
We have the table that probability of buying the different product for the given education years of the people, we can interpret this table as probability of buying KP281 given education years is 12 (P(product = KP281) | (education years=12)) for ex :

probability of buying KP281 given education years is 12 is 0.666667

probability of buying KP481 given education years is 12 is 0.333333

probability of buying KP781 given education years is 12 is 0.000000

```python
plt.rcParams["figure.figsize"] = (10,6)
a = product_and_Education_columns.plot(kind = 'bar')
a.legend(title='Education', bbox_to_anchor=(1, 1.02),
         loc='upper left')
plt.show()
```



This plot depicts the probability of various product given miles

People with education years is 12 they tends to buy product : KP281

People with education years is 12 they tends to buy product : KP281

People with education years is 13 they tends to buy product : KP281

People with education years is 14 they tends to buy product : KP281

People with education years is 15 they tends to buy product : KP281

People with education years is 16 they tends to buy product : KP281 or KP481

People with education years is 18 they tends to buy product : KP781

People with education years is 20 they tends to buy product : KP781

People with education years is 21 they tends to buy product : KP781

We can conclude this that people with education year between 12 to 16 years will more probable to buy KP281 or some will also buy KP481 might be the reason that they are not completed with their education so they they did not get much time for the workout and secondaly they have no or less source or income due to which they will buy the budget product, but the people who have above 16 years of education they will buy KP781 because most of them have completed their education and they will get time for workout and also have a stable soruce of income

```python
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Level	Selling Price	Income_range	Miles_range	Age_range
0	KP281	18	Male	14	Single	3	4	29562	112	Entry	1500	29000-50000	70-120	15-20
1	KP281	19	Male	15	Single	2	3	31836	75	Entry	1500	29000-50000	70-120	15-20
2	KP281	19	Female	14	Partnered	4	3	30699	66	Entry	1500	29000-50000	20-70	15-20
3	KP281	19	Male	12	Single	3	3	32973	85	Entry	1500	29000-50000	70-120	15-20
4	KP281	20	Male	13	Partnered	4	2	35247	47	Entry	1500	29000-50000	20-70	15-20

Product profiling table

```python
from tabulate import tabulate
table = [['Product', 'Age', 'Gender' , 'Education' , 'MaritialStatus', 'Usage' , 'Fitness', 'Income_Range', 'Mi
        , ['KP281', '15-40', 'male, female', '12-16','partnered, single' ,'2-4','1-4', '29000-70000','20-120']
        ['KP481', '15-40', 'male, female','12-16','partnered, single','2-4','1-4', '29000-70000','20-120'],
        ['KP781', '40-45', 'male','18,20,21','partnered or single','5-7','4-5', '70000-110000','120-370']]

print(tabulate(table, headers='firstrow', tablefmt='fancy_grid'))
```

```
| Product    | Age   | Gender       | Education | MaritialStatus      | Usage | Fitness | Income_Range |
Miles_range  |
| KP281      | 15-40 | male, female | 12-16     | partnered, single   | 2-4   | 1-4     | 29000-70000  |
20-120       |
| KP481      | 15-40 | male, female | 12-16     | partnered, single   | 2-4   | 1-4     | 29000-70000  |
20-120       |
| KP781      | 40-45 | male         | 18,20,21  | partnered or single | 5-7   | 4-5     | 70000-110000 |
120-370      |
```

From this table we can check that for a given information of person which product compnay should recommend to that customer.

Useful Insights from the data

1. Data contains 180 rows and 9 columns and there are no null values, there are some outliers in income and miles.

2. Most purchased product is KP281 with a probability of 0.44 then KP481 with a probability of 0.33 then KP781 with a probability of 0.22

3. There are 57.78% Male and 42.22% female in the data and 40.56% of the people are single whereas 59.44% of the people are Partnered.

4. Income we found that most of the people have the income range between 29000-50000 and 70000-90000

5. Fitness we found that most of the people have the fitness range is 3 and least has 1

6. Usage we found that maximum people who use this product for 3 days.

7. Miles we found that among the people who bought this product maximum of people run 70-120 miles each week

8. Maximum people with age range between 20-25 buy this product more

9. People which have an education of around 14-18 years are expected to buy this product more

10. Distribution of Miles, income and Age are skewed

11. There is a positive corelation between Age and Income, Education and Income, Fitness and miles, usage and miles, fitness, and usage

12. Aged people are more tends to buy KP281 because their income is more and they need a good quality of treadmill, wherever the young people whose age is between 15-20 or 20-30 more tends to buy KP481 because they have less income, so they need a product which is in their budget, and KP781 is preferable for the mid age ranged people.

13. People who ran less miles per week are more tends to buy KP481 or KP281 because there usage is less due to which they don't tends to buy high price product, where ever the people who ran miles between 70-120 more tends to buy KP481 or KP281 because they have a normal usage of the product so they want a mid-range product for them and KP781 is more preferable for the people who ran for more than 120 miles per week and it is a premium product so it will be more durable for them to buy this.

14. People with low-income range will buy KP281, whereas the people with intermediate income range will buy KP281 or KP481 wherever the people with income range more the 70000 will always buy KP781

15. Female is more probable to buy KP281 or KP481 wherever men is more probable to buy KP781 this is due to may be male will require more intense workout and due to that they will buy a more premium product and female will not require that much of intense workout, so they prefer and low range or mid-range product

16. People with education year between 12 to 16 years will more probable to buy KP281 or some will also buy KP481 might be the reason that they are not completed with their education so they did not get much time for the workout and secondly they have no or less source or income due to which they will buy the budget product, but the people who have above 16 years of education they will buy KP781 because most of them have completed their education and they will get time for workout and also have a stable source of income

17. People with fitness between 1-4 tends to buy KP281 or KP481 and fitness as 5 will buy KP781

18. People with usage between 2-4 tends to buy KP281 or KP481 and usage above 4 will buy KP781.

19. There is not much difference in the probability of buying the product on the Marital Status of the people.

Recommended Business Insights

1. The sale of KP781 is less as compared to other products so there should be some offers given on that product so that people will be fascinated with those offers and will tends to buy KP781 due to which sales will also increase.

2. Company so some marketing in between the youth about the KP781 because we have seen that youth buy KP281 or KP481 so company should make a strategy like providing these machines in the gyms because most of the youth will go to gym and they use this machine, and they will tend to buy this

3. Company should give some free trials usage of these machines so that people will try this first because of free trail and their buying probability will increase.

4. There should be some offers for female customers as we have seen that most of the customers are male so to increase the female customers there should be some offers given to the female customers and there should be a female trainer also available in the stores so that female customers will feel more comfortable and their product buying probability increases.

5. There should also be some offers given to the students so that their probability will also increase to buying the product, there should be some health awareness sessions among the students in their school or colleges so that they will be aware about the product.

6. Company should hire a health icon as brand Ambassador of the company for ex. a person which is followed by most of the people for health tips because it is more probable that people will follow their icons so if that person advertise the products, then the probability that people will buy that product will increase.

7. Company should give some other health or gym items with these products to increase the sales because people who get some extra gym items like a Dumble or skipping rope will be more fascinated towards this product.

8. Company should contact some popular gyms to use their products in the gym so that people will be aware about their products and will increase their probability to buy those products.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [281…

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: