Aerofit Case Study - Mayank Khanduja

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

Objectives

1. Perform descriptive analytics to create a customer profile for each AeroFit treadmill product by developing appropriate tables and charts.
2. For each AeroFit treadmill product, construct two-way contingency tables and compute all conditional and marginal probabilities along with their insights/impact on the business.

In [281]:
```
import numpy as np
import pandas as pd
from datetime import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
import plotly.express as px
```

In [2]:
```
df = pd.read_csv("C:\\Users\\Test\\Desktop\\aerofit.csv")
df.head()
```

Out[2]:

	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

```
## DATASET CHARACTERSTICS
Product Purchased: KP281, KP481 and KP781, are the 3 different types of treadmills that
are purchased by customers
Age : In years, age of the customer who purchased
Gender: Gender of the purchased customer
Education: represented in years
Marital Status: Single or partnered
Usage: The average number of times the customer has planned to use the treadmill each week
Fitness: Self rated fitness of the user rated from 1 (as poor shape) to 5 (as excellent
shape)
Miles: The average number of miles the customer expects to walk or run each week
Income: Annual income of the user in Dollars
```

Basic Dataset Characterstics

In [3]:
```
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 [4]:
```
df.describe()
```

Out[4]:

	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 [5]:
```
df.describe(include= "object")
```

Out[5]:

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

OBSERVATIONS #1

```
# The dataset has 180 rows and 9 columns
# There are no null values in the dataset
1. Age: The mean Age observed is 28.7 years while median age is 26 years
2. Education: Mean Education observed is 15.5 years while median is 16 years
3. Usage: The Average usage of product is 3.45 days in a week.
4. Self_Rated_Fitness: The average fitness ranking is 3.3
5. Income: The average income is observed to be 53719.57
6. Miles: Average Miles customer walked is 103
7. Product: Top most used product is KP281
8. Gender : Majority of the users are Male
9. MartialStatus : Majority of users are Married.
```

Non Graphical Analysis

In [6]:
```
df.head()
```

Out[6]:

	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 [7]:
```
df["Product"].value_counts()
```

Out[7]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

In [8]:
```
df["Gender"].value_counts()
```

Out[8]:
```
Male      104
Female     76
Name: Gender, dtype: int64
```

In [9]:
```
df["MaritalStatus"].value_counts()
```

Out[9]:
```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

In [10]:
```
df["Usage"].value_counts()
```

Out[10]:
```
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

In [11]:
```
df["Fitness"].value_counts()
```

Out[11]:
```
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

In [12]:
```
df.groupby("Gender")["MaritalStatus"].value_counts()
```

Out[12]:
```
Gender  MaritalStatus
Female  Partnered        46
        Single           30
Male    Partnered        61
        Single           43
Name: MaritalStatus, dtype: int64
```

In [13]:
```
df.groupby("Product")["Gender"].value_counts()
```

Out[13]:
```
Product  Gender
KP281    Female    40
         Male      40
KP481    Male      31
         Female    29
KP781    Male      33
         Female     7
Name: Gender, dtype: int64
```

In [14]:
```
df.groupby("Product")["MaritalStatus"].value_counts()
```

Out[14]:
```
Product  MaritalStatus
KP281    Partnered        48
         Single           32
KP481    Partnered        36
         Single           24
KP781    Partnered        23
         Single           17
Name: MaritalStatus, dtype: int64
```

In [15]:
```
df.groupby("Product")["Fitness"].value_counts()
```

Out[15]:
```
Product  Fitness
KP281    3          54
         2          14
         4           9
         5           2
         1           1
KP481    3          39
         2          12
         4           8
         1           1
KP781    5          29
         4           7
         3           4
Name: Fitness, dtype: int64
```

In [16]:
```
df.groupby("Product")["Usage"].value_counts()
```

Out[16]:
```
Product  Usage
KP281    3        37
         4        22
         2        19
         5         2
KP481    3        31
         2        14
         4        12
         5         3
KP781    4        18
         5        12
         6         7
         7         2
         3         1
Name: Usage, dtype: int64
```

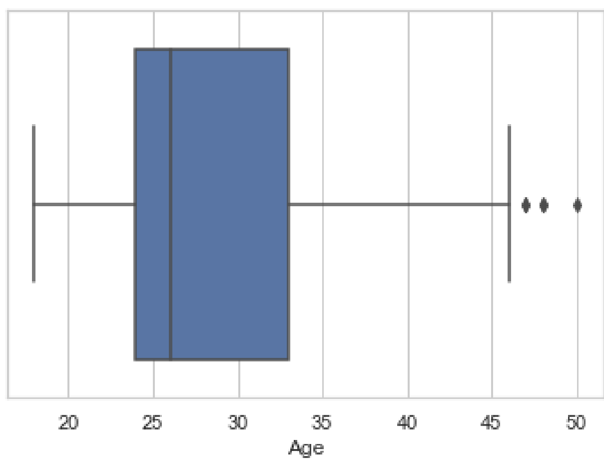
OBSERVATIONS #2

1. The data shows high usage of KP281 treadmill
2. Top Most buyers are Partnered Males
3. The product has high acceptance in Partnered maritial status
4. Majority of users use the product on alternative basis or 3 days a week.
5. Majority of users feel they are average looking i.e. they have a fitness score of 3
6. Those who consider themselves fit i.e. have fitness score of 5 has high acceptance for KP781 product

Graphical Analysis - Univariate

Age Group Analysis

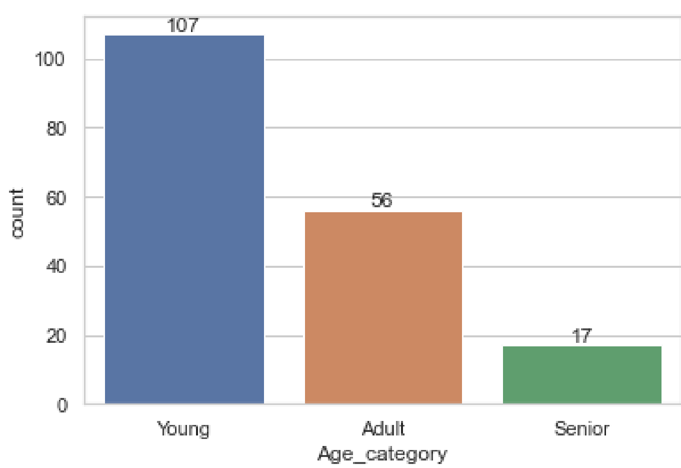
In [23]:
```
sns.boxplot(data=df,x='Age')
plt.show()
```



In [17]:
```
# Converting Age column into categorical values
age_bins = [16,28,39,51]
age_labels = ["Young","Adult","Senior"]
df["Age_category"] = pd.cut(df["Age"],bins= age_bins,labels= age_labels)
```

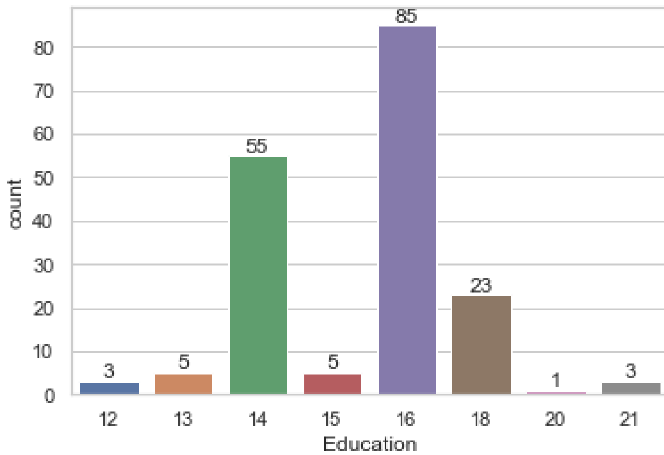
In [37]:
```
sns.set(style='whitegrid')
sns.countplot(data= df,x="Age_category")
value_counts = df['Age_category'].value_counts()
for index, value in enumerate(value_counts):
    plt.annotate(str(value), xy=(index, value), ha='center', va='bottom')

plt.show()
```



Education Analysis

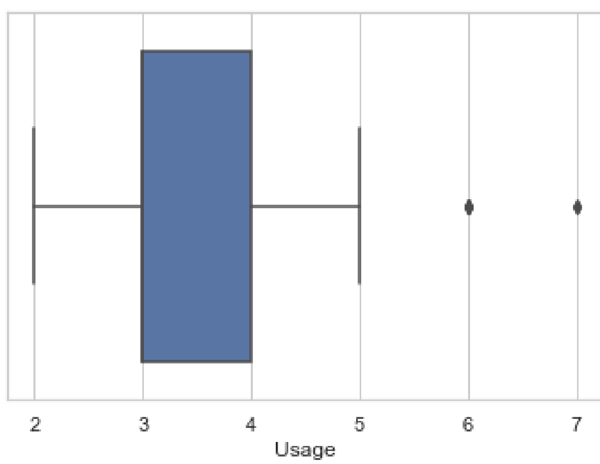
In [39]:
```
#education
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="Education")
value_counts = df["Education"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```



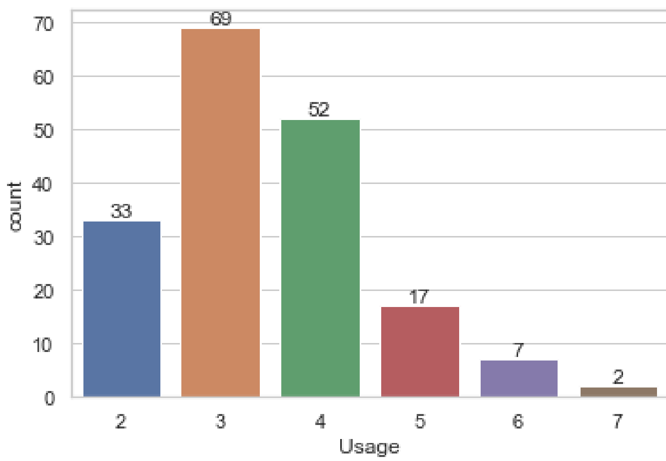
Usage Analysis

In [38]:
```

sns.boxplot(data=df,x='Usage')
plt.show()
```

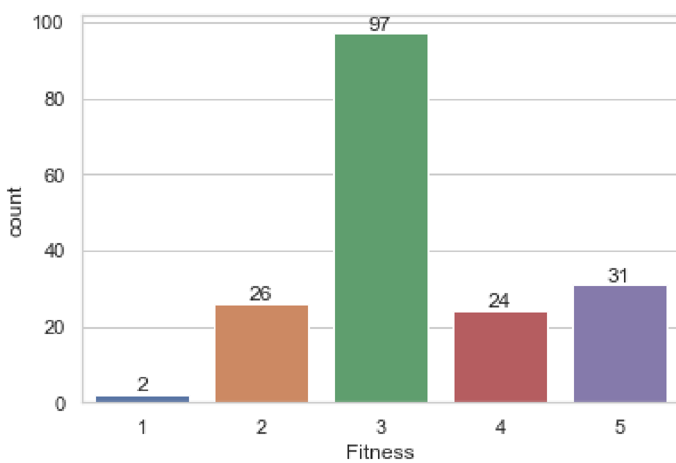


In [40]:
```
#usage
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="Usage")
value_counts = df["Usage"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```



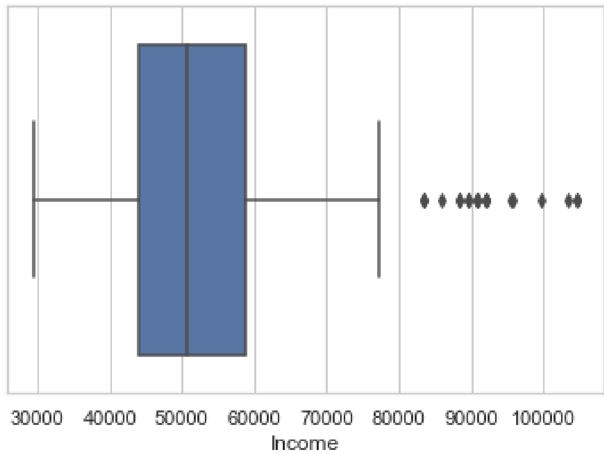
Fitness Analysis

In [41]:
```
# fitness
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="Fitness")
value_counts = df["Fitness"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```

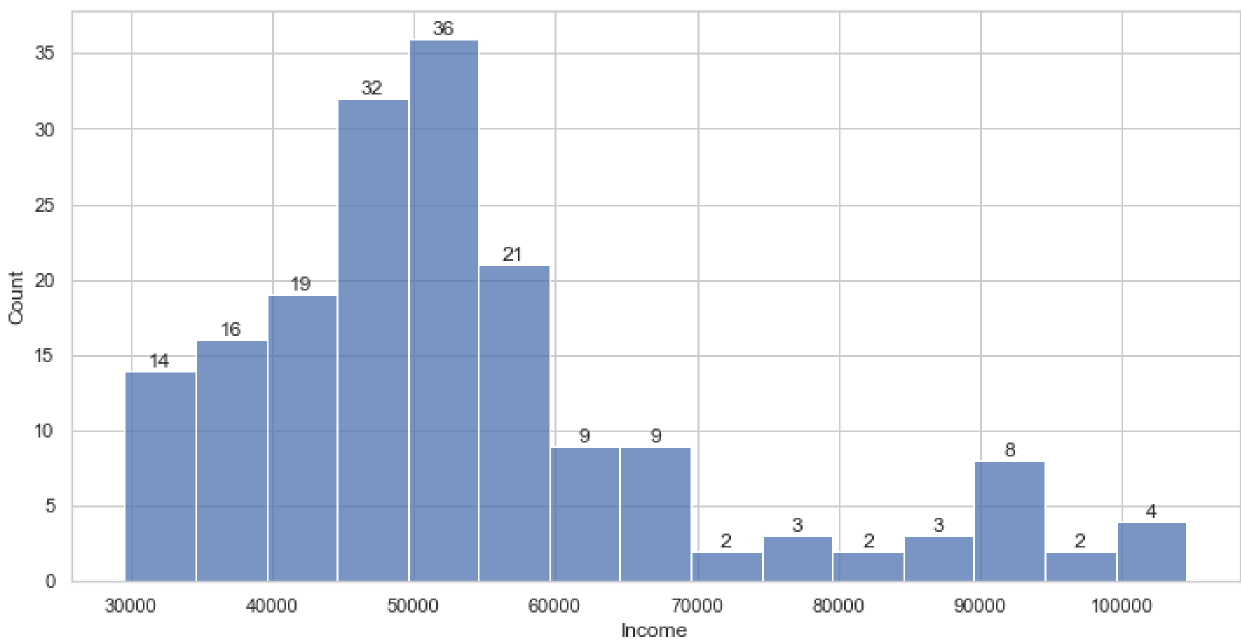


Income Analysis

In [49]:
```
sns.boxplot(data=df,x='Income')
plt.show()
```

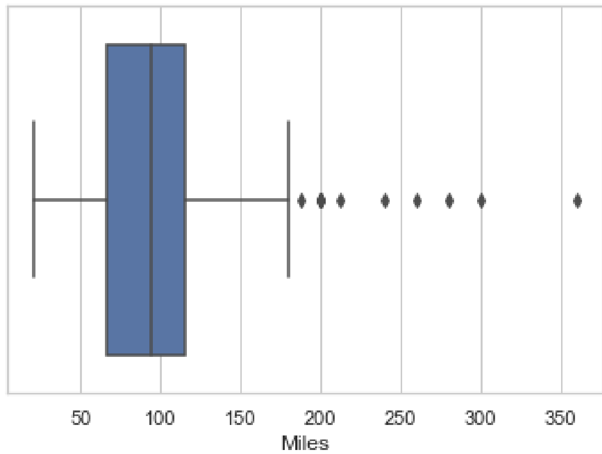


In [47]:
```
plt.figure(figsize=(12,6))
ax=sns.histplot(data=df,x='Income')
value_counts = df["Income"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
```

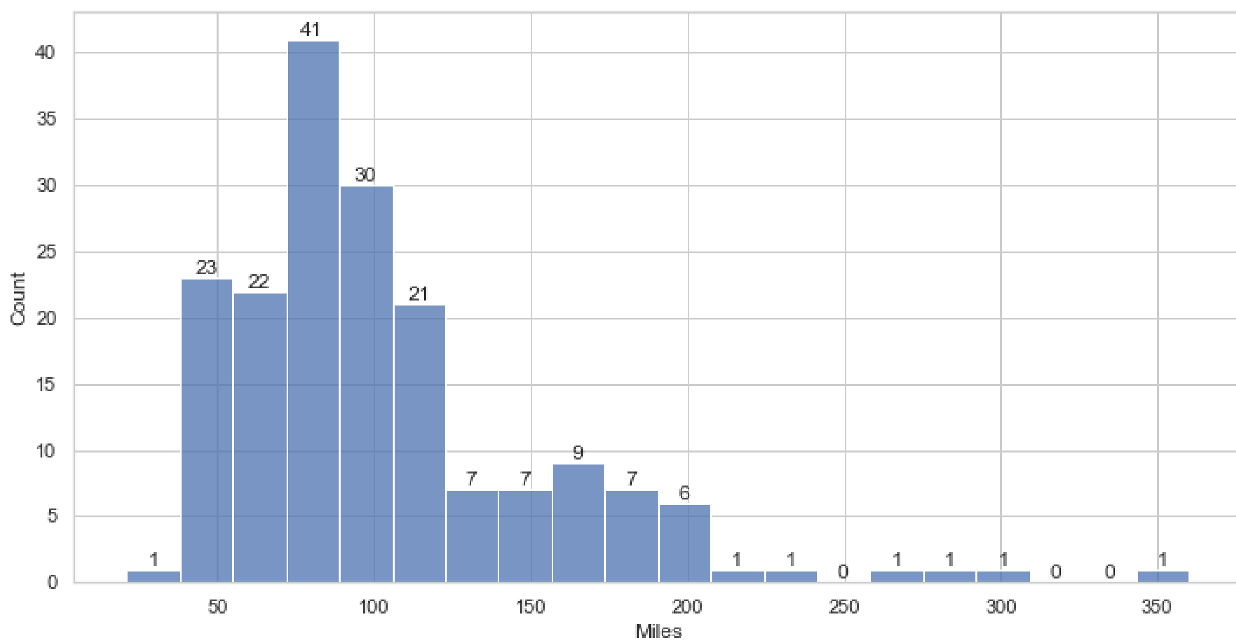


Miles Ran Analysis

In [44]:
```
sns.boxplot(data=df,x='Miles')
plt.show()
```



In [48]:
```
plt.figure(figsize=(12,6))
ax= sns.histplot(data=df,x='Miles')
value_counts = df["Usage"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
```



OBSERVATIONS #3

1. Age: The target audience for our product mainly falls within the age range of 25 to 32 years, with the median age being 26 years.
2. Education: Our product seems to appeal to individuals with a relatively higher level of education, as most users have 16 years of education

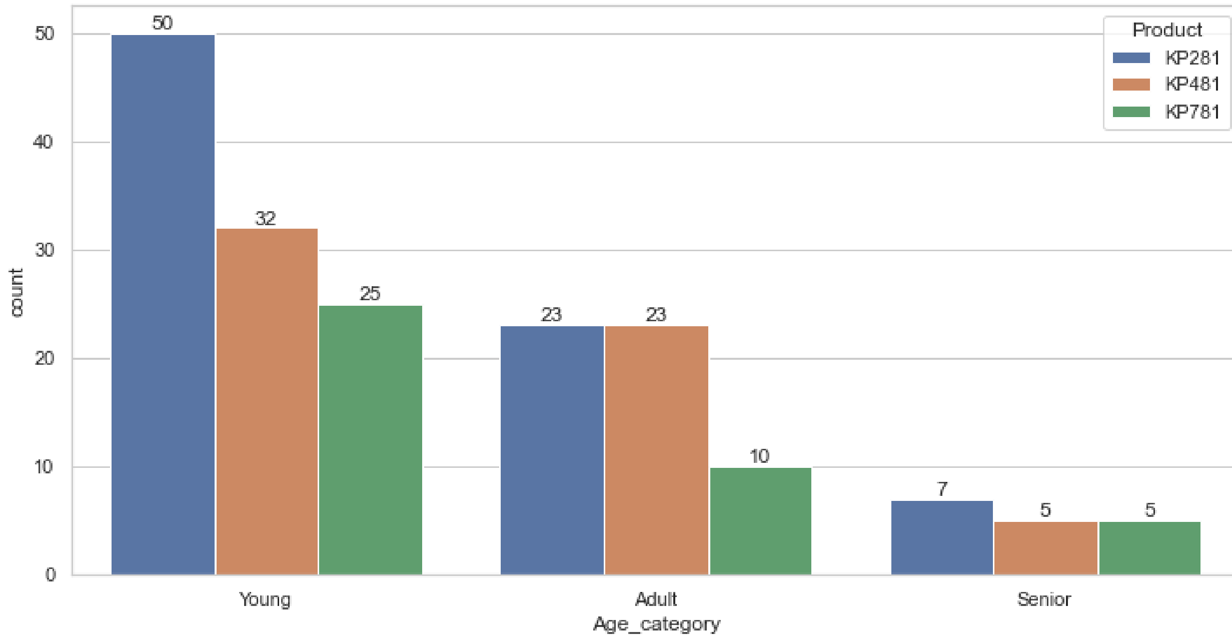
3. Usage: The majority of our users use the product 3 to 4 days a week.
4. Fitness - Majority of users feels that they are average looking
5. Income: Our product is resonating well with the middle-income segment, with the median income observed at $50,000
6. Miles: The data suggests that the majority of users have run between 50 to 100 miles.

Graphical Analysis - Bivariate

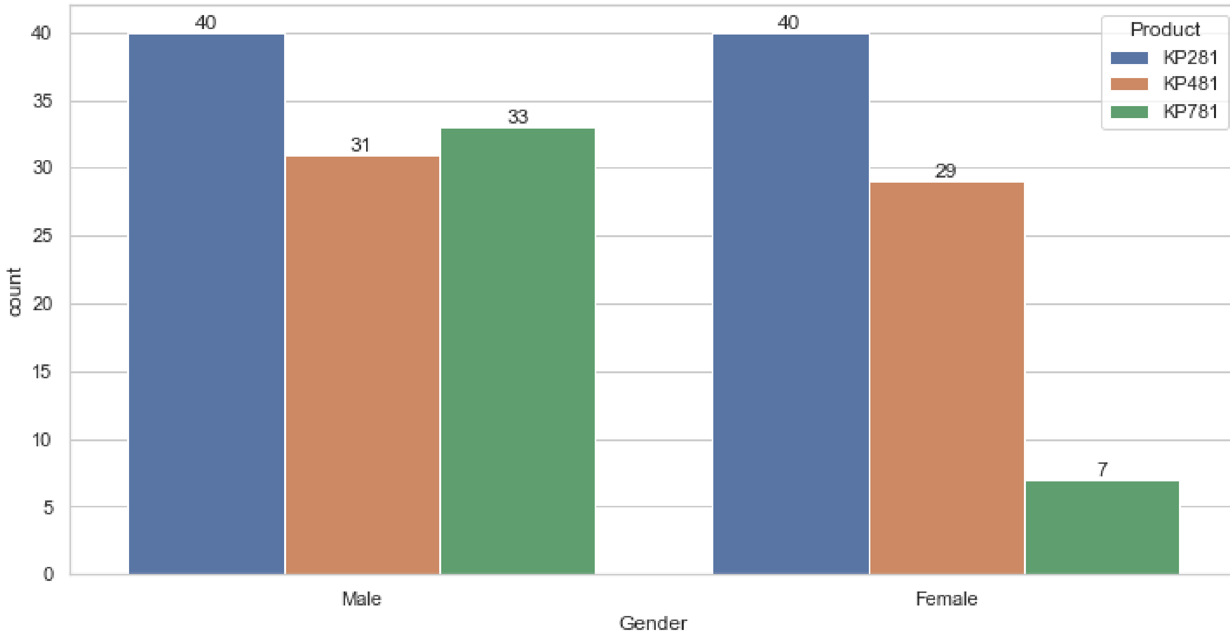
Co-relation between different Attributes

In [113]:
```
plt.figure(figsize=(20,6))
ax = sns.heatmap(df.corr(),annot=True,fmt='.4f',linewidths=.5,cmap='magma')
plt.yticks(rotation=0)
plt.show()
```

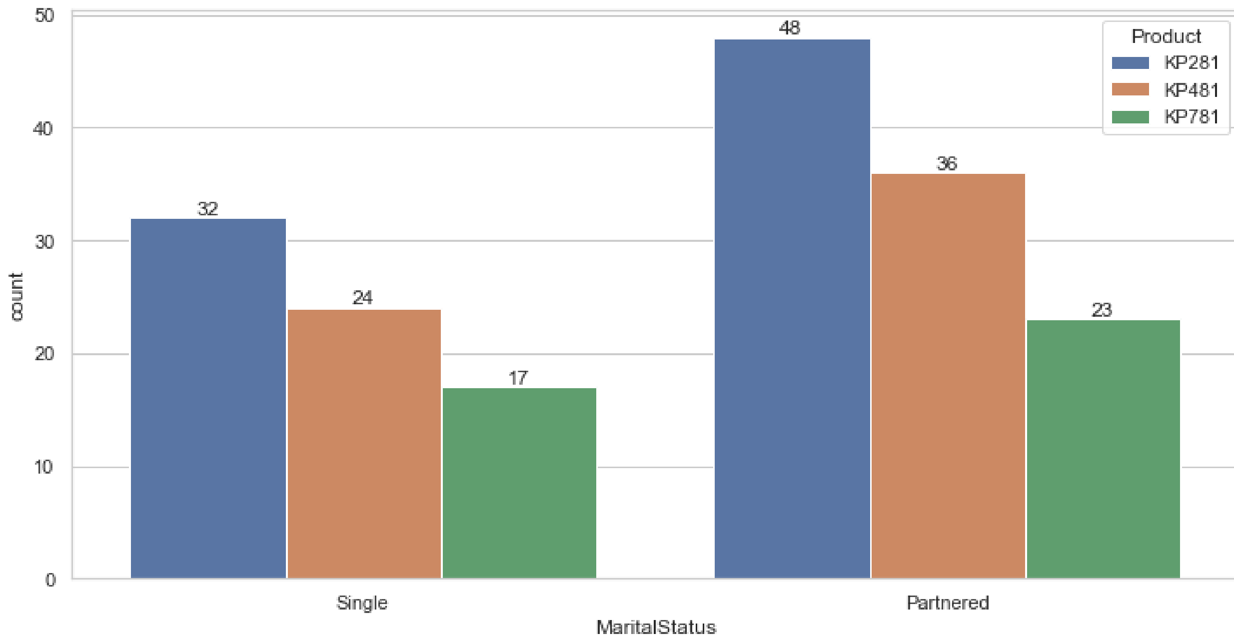
In [88]:
```
# PRODUCT AND AGE
plt.figure(figsize=(12,6))
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="Age_category",hue="Product")
value_counts = df["Age_category"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```



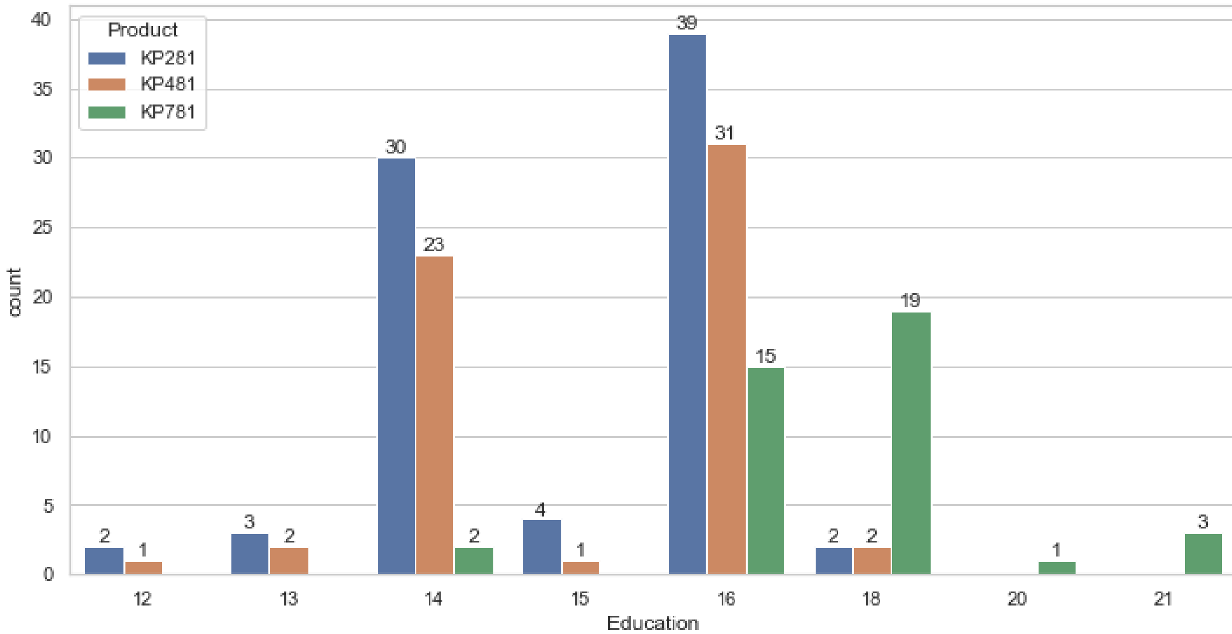
In [86]:
```
# PRODUCT AND GENDER
plt.figure(figsize=(12,6))
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="Gender",hue="Product")
value_counts = df["Gender"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```



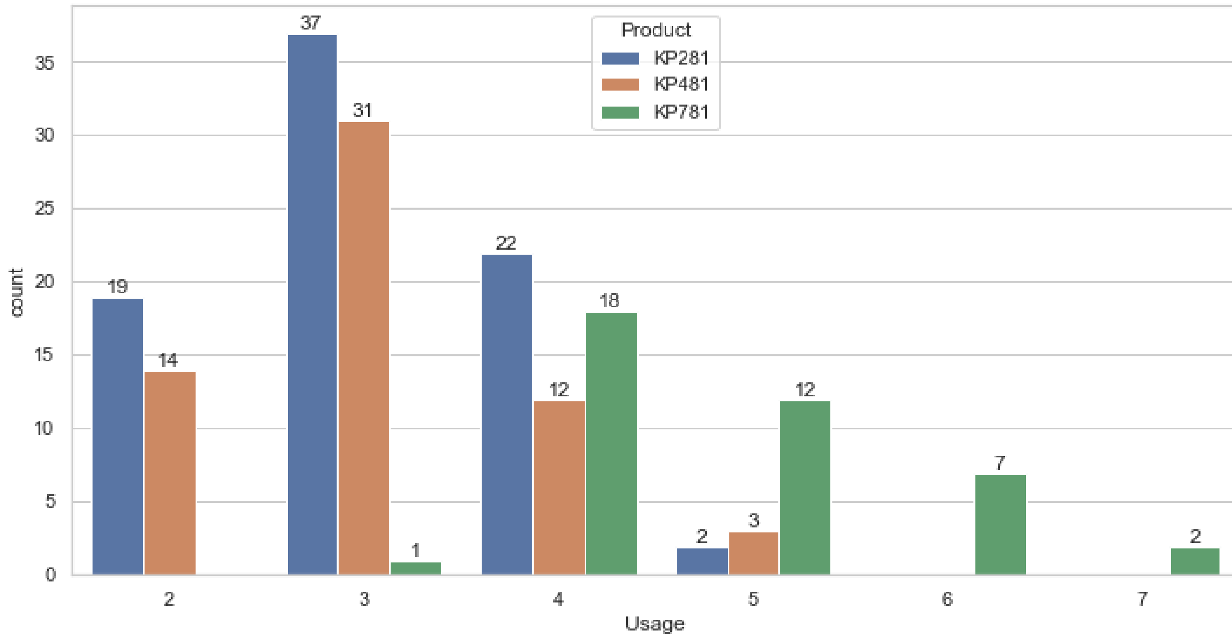
In [90]:
```
# PRODUCT AND MARITAL STATUS
plt.figure(figsize=(12,6))
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="MaritalStatus",hue="Product")
value_counts = df["MaritalStatus"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```



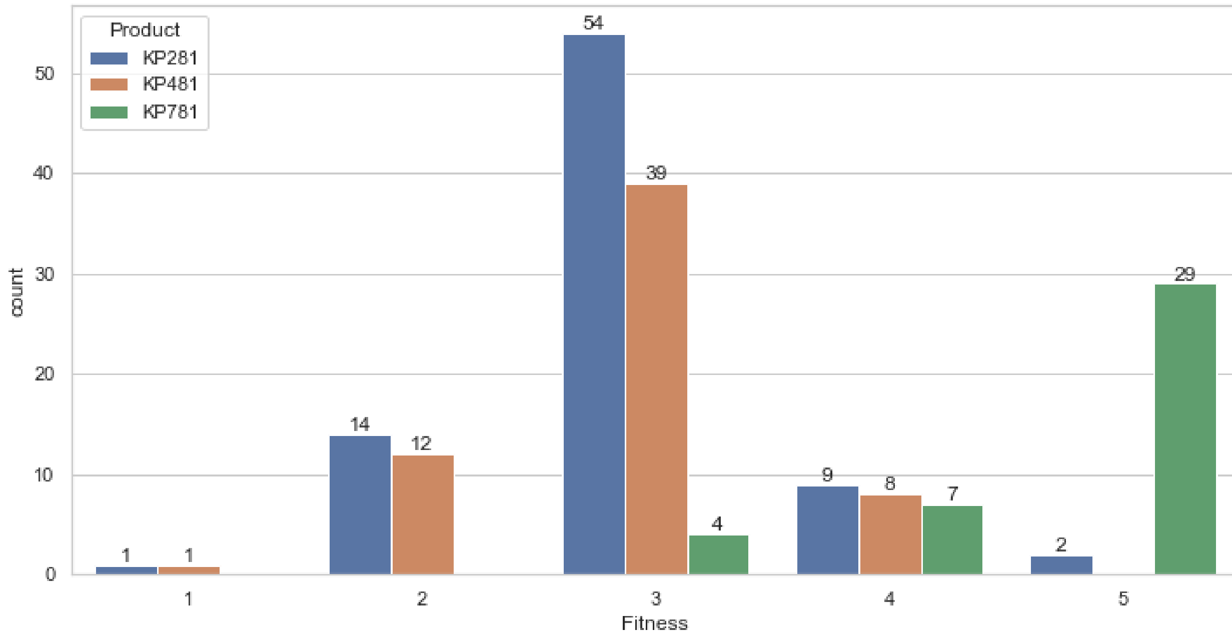
In [92]:
```
# PRODUCT AND EDUCATION
plt.figure(figsize=(12,6))
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="Education",hue="Product")
value_counts = df["Education"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```



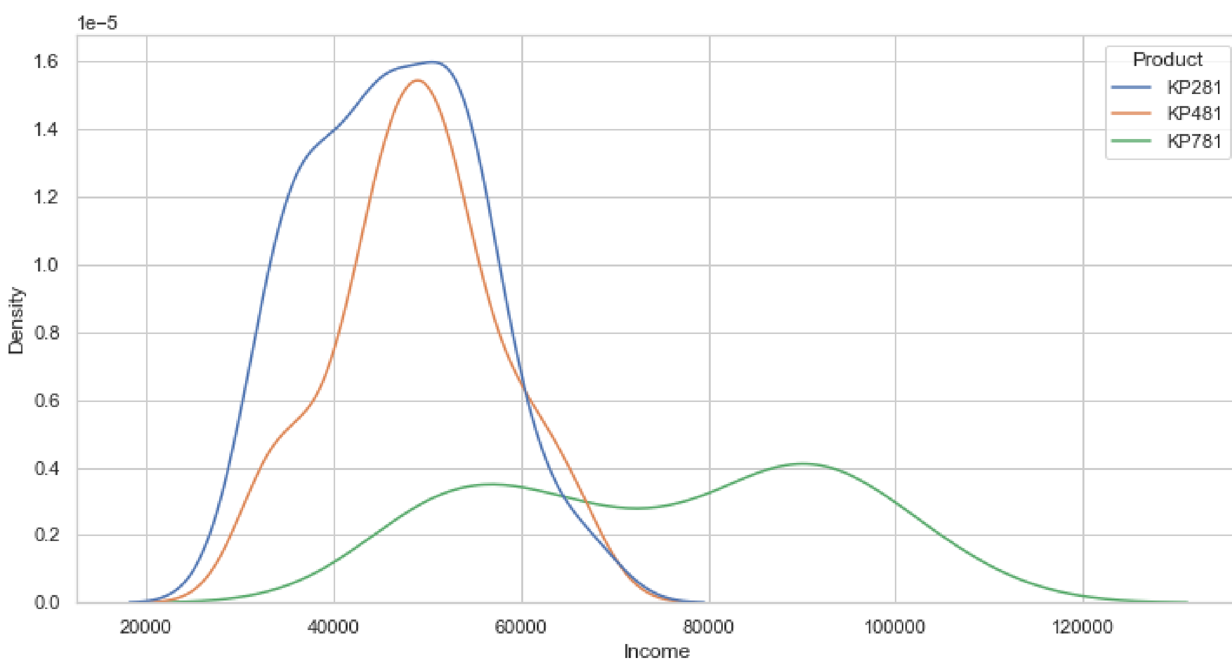
In [94]:
```
# PRODUCT AND USAGE
plt.figure(figsize=(12,6))
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="Usage",hue="Product")
value_counts = df["Usage"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```



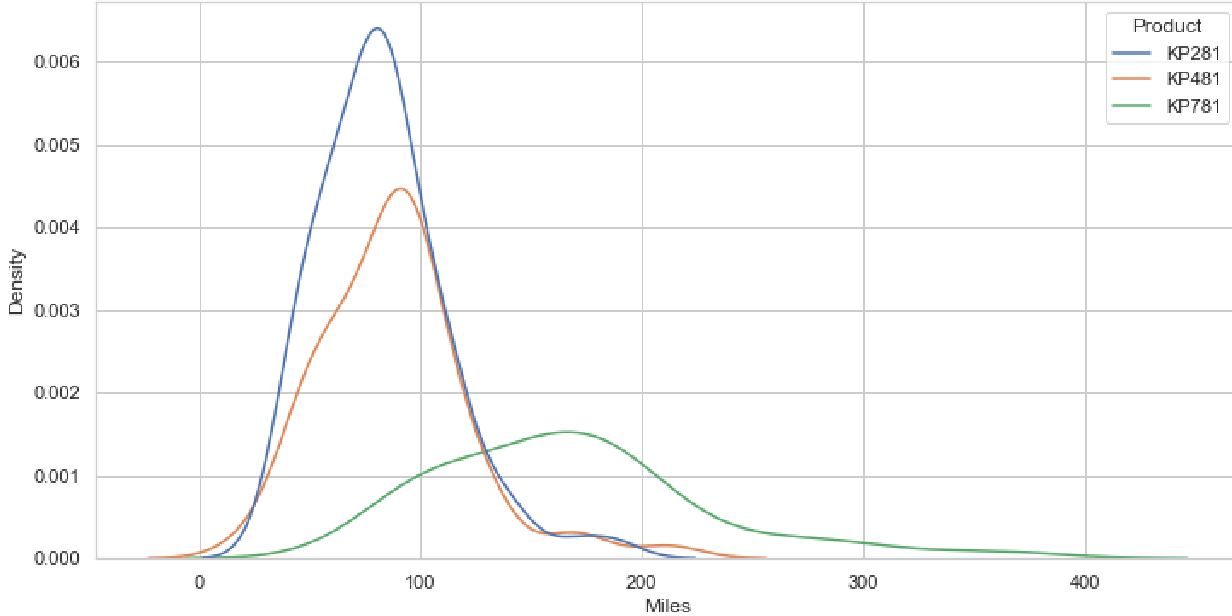
In [96]:
```
# PRODUCT AND FITNESS
plt.figure(figsize=(12,6))
sns.set(style='whitegrid')
ax= sns.countplot(data= df,x="Fitness",hue="Product")
value_counts = df["Fitness"].value_counts()
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'), (p.get_x() + p.get_width() / 2., p.get_height
                ha='center', va='bottom')
plt.show()
```



In [102]:
```
# PRODUCT AND INCOME
plt.figure(figsize=(12,6))
sns.set(style='whitegrid')
sns.kdeplot(data= df,x="Income",hue="Product")
plt.show()
```



In [106]:
```
# PRODUCT AND MILES
plt.figure(figsize=(12,6))
sns.set(style='whitegrid')
sns.kdeplot(data= df,x="Miles",hue="Product")
plt.show()
```



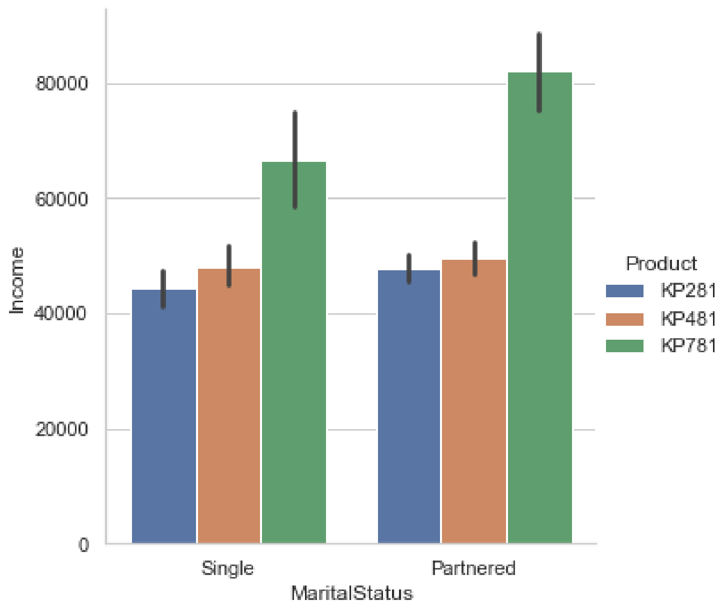
OBSERVATIONS #4

1. PRODUCT AND AGE: KP281 treadmill is highly preferred by the Young Generation (aged 18-28).For Adults, both KP281 and KP481 treadmills are equally popular, suggesting a balanced preference between the two models. Seniors, on the other hand, show a relatively similar level of interest in both KP481 and KP781 treadmills.
2. PRODUCT AND GENDER: KP281 emerges as the top choice among both male and female users, indicating its universal appeal across genders. However, it's worth noting that there is a lower uptake of KP781 among females compared to other models.
3. PRODUCT AND MARTIAL STATUS The pattern shows a higher demand among individuals with a partner (patterned) compared to single individuals.
4. PRODUCT AND EDUCATION Users with more years of education tend to prefer the advanced features of KP781, making it an attractive choice for individuals with higher levels of expertise and education.
5. PRODUCT AND USAGE For those aiming to use the treadmill for four or more days a week, KP781 stands out as the ideal choice. Meanwhile, users planning to exercise 3 to 4 days a week show a strong preference for KP281, making it a popular option for regular workouts.
6. PRODUCT AND FITNESS Customers who rate their fitness levels as above average find KP781 to be a compelling choice while on the other hand, users with below-average fitness tend to gravitate towards KP281 or KP481
7. PRODUCT AND INCOME KP781 treadmill is a hit among customers with an income above 80k dollars making it an aspirational choice for higher-income individuals. For customers with incomes ranging between 20k to 60k dollars, both KP281 and KP481 treadmills have a strong presence, capturing a substantial share of this market segment.
8. PRODUCT AND MILES Users aiming to run less than 100 miles prefer the comfortable and efficient options of KP281 and KP481. In contrast, those planning to exceed 100 miles find the high-performance features of KP781 more appealing

Graphical Analysis - Multivariate

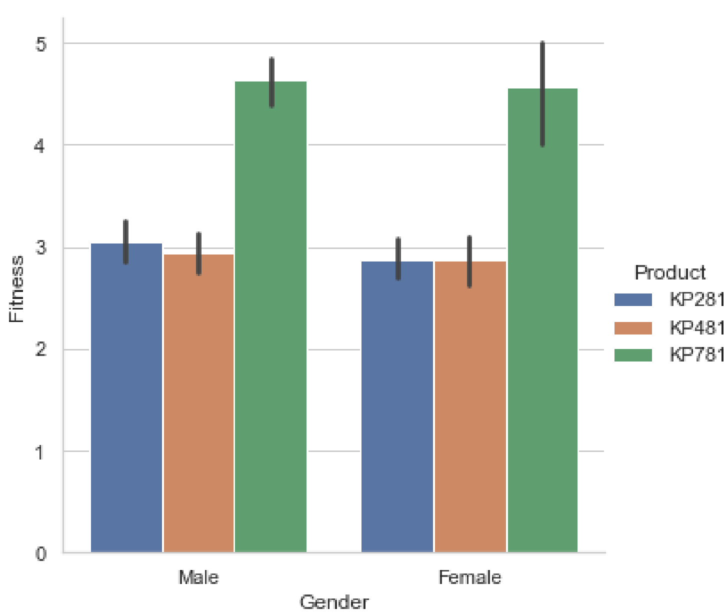
In [127]:
```
# MARITAL STATUS , INCOME AND PRODUCT
plt.figure(figsize=(12, 6))
sns.catplot(x='MaritalStatus',y='Income',hue='Product',data=df,kind='bar')
plt.show()
```

```
<Figure size 864x432 with 0 Axes>
```



In [125]:
```
# PRODUCT, FITNESS AND GENDER
plt.figure(figsize=(12, 6))
sns.catplot(x='Gender',y='Fitness',hue='Product',data=df,kind='bar')
plt.show()
```

```
<Figure size 864x432 with 0 Axes>
```



Joint Probabilities

P [AGE ∩ PRODUCT]

In [162]:
```
# PROBABILITY OF AGE ∩ PRODUCT
ap = pd.crosstab(index= df["Age_category"],columns= df["Product"],normalize= True)
ap
```

Out[162]:

Product Age_category	KP281	KP481	KP781
Young	0.277778	0.177778	0.138889
Adult	0.127778	0.127778	0.055556
Senior	0.038889	0.027778	0.027778

In [175]:
```
print("Younger Generation i.e. 18-28 years of age")
print("Probability of Purchasing KP281 for Young Generation :", round(ap["KP281"]["Young"],2
print("Probability of Purchasing KP481 for Young Generation :", round(ap["KP481"]["Young"],2
print("Probability of Purchasing KP781 for Young Generation :", round(ap["KP781"]["Young"],2
print()
print("Adult generation i.e.29-39 years of age")
print("Probability of Purchasing KP281 for Adult Generation :", round(ap["KP281"]["Adult"],2
print("Probability of Purchasing KP481 for Adult Generation :", round(ap["KP481"]["Adult"],2
print("Probability of Purchasing KP781 for Adult Generation :", round(ap["KP781"]["Adult"],2
print("Senior generation i.e. above 40 years of age")
print()
print("Probability of Purchasing KP281 for Senior Generation :", round(ap["KP281"]["Senior"]
print("Probability of Purchasing KP481 for Senior Generation :", round(ap["KP481"]["Senior"]
print("Probability of Purchasing KP781 for Senior Generation :", round(ap["KP781"]["Senior"]
```

```
Younger Generation i.e. 18-28 years of age
Probability of Purchasing KP281 for Young Generation : 0.28
Probability of Purchasing KP481 for Young Generation : 0.18
Probability of Purchasing KP781 for Young Generation : 0.14

Adult generation i.e.29-39 years of age
Probability of Purchasing KP281 for Adult Generation : 0.13
Probability of Purchasing KP481 for Adult Generation : 0.13
Probability of Purchasing KP781 for Adult Generation : 0.06
Senior generation i.e. above 40 years of age

Probability of Purchasing KP281 for Senior Generation : 0.04
Probability of Purchasing KP481 for Senior Generation : 0.03
Probability of Purchasing KP781 for Senior Generation : 0.03
```

P[GENDER ∩ PRODUCT]

In [178]:
```
# PROBABILITY OF GENDER ∩ PRODUCT
gp= pd.crosstab(index= df["Gender"],columns= df["Product"],normalize= True)
gp
```

Out[178]:

Product Gender	KP281	KP481	KP781
Female	0.222222	0.161111	0.038889
Male	0.222222	0.172222	0.183333

In [192]:
```
print("Gender- Male")
print("Probability of Male purchasing KP281:",round(gp["KP281"]["Male"],2))
print("Probability of Male purchasing KP481:",round(gp["KP481"]["Male"],2))
print("Probability of Male purchasing KP781:",round(gp["KP781"]["Male"],2))
print()
print("Gender-Female")
print("Probability of Female purchasing KP281:",round(gp["KP281"]["Female"],2))
print("Probability of Female purchasing KP481:",round(gp["KP481"]["Female"],2))
print("Probability of Female purchasing KP781:",round(gp["KP781"]["Female"],2))
print()
```

```
Gender- Male
Probability of Male purchasing KP281: 0.22
Probability of Male purchasing KP481: 0.17
Probability of Male purchasing KP781: 0.18

Gender-Female
Probability of Female purchasing KP281: 0.22
Probability of Female purchasing KP481: 0.16
Probability of Female purchasing KP781: 0.04
```

P [PRODUCT ∩ EDUCATION]

In [241]:
```
# PROBABILITY OF PRODUCT ∩ EDUCATION
pe = pd.crosstab(index= df["Education"],columns= df["Product"],normalize= True)
pe = pd.DataFrame(pe).reset_index()
pe.set_index("Education",inplace=True)
pe
```

Out[241]:

Product Education	KP281	KP481	KP781
12	0.011111	0.005556	0.000000
13	0.016667	0.011111	0.000000
14	0.166667	0.127778	0.011111
15	0.022222	0.005556	0.000000
16	0.216667	0.172222	0.083333
18	0.011111	0.011111	0.105556
20	0.000000	0.000000	0.005556
21	0.000000	0.000000	0.016667

P[PRODUCT ∩ MARITAL STATUS]

In [242]:
```
# PROBABILITY OF PRODUCT ∩ MARITAL STATUS
pd.crosstab(index= df["MaritalStatus"],columns= df["Product"],normalize= True)
```

Out[242]:

Product	KP281	KP481	KP781
MaritalStatus			
Partnered	0.266667	0.200000	0.127778
Single	0.177778	0.133333	0.094444

P[PRODUCT ∩ FITNESS]

In [244]:
```
# PROBABILITY OF PRODUCT ∩ FITNESS
pd.crosstab(index= df["Fitness"],columns= df["Product"],normalize= True)
```

Out[244]:

Product	KP281	KP481	KP781
Fitness			
1	0.005556	0.005556	0.000000
2	0.077778	0.066667	0.000000
3	0.300000	0.216667	0.022222
4	0.050000	0.044444	0.038889
5	0.011111	0.000000	0.161111

P[PRODUCT ∩ INCOME]

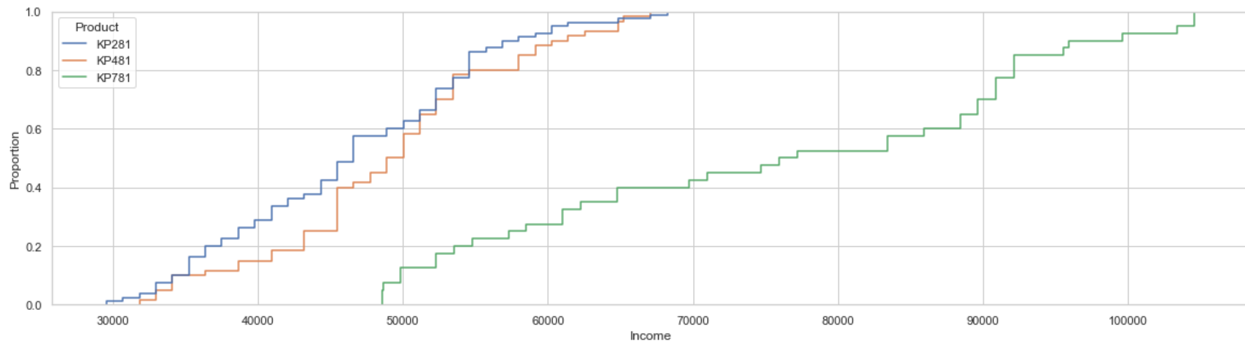
In [249]:
```
# PROBABILITY OF PRODUCT ∩ INCOME
income_bins= [28000,55000,80000,150000]
income_labels= ["Moderate","High","Elite"]
df["income_category"]= pd.cut(df["Income"],bins= income_bins,labels= income_labels)
pd.crosstab(index= df["income_category"],columns= df["Product"],normalize= True)
```

Out[249]:

Product	KP281	KP481	KP781
income_category			
Moderate	0.383333	0.266667	0.050000
High	0.061111	0.066667	0.066667
Elite	0.000000	0.000000	0.105556

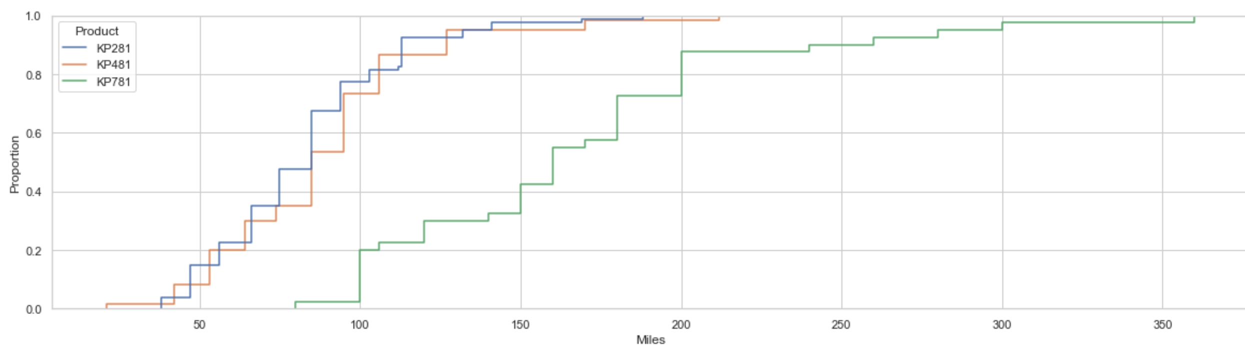
CDF PRODUCT AND INCOME

In [251]:
```
# CDF OF INCOME
plt.figure(figsize=(20,5))
sns.ecdfplot(data=df,x='Income',hue='Product',complementary=False)
plt.show()
```



CDE PRODUCT AND MILES

In [252]:
```
# PROBABILITY OF PRODUCT ∩ MILES
# CDF OF MILES
plt.figure(figsize=(20,5))
sns.ecdfplot(data=df,x='Miles',hue='Product',complementary=False)
plt.show()
```



P[PRODUCT ∩ (GENDER ∩ MARITAL STATUS)]

In [255]:
```
# P[PRODUCT ∩ (GENDER & MARITAL STATUS)]
pd.crosstab(index= [df["Gender"],df["MaritalStatus"]],columns= df["Product"],normalize= True
```

Out[255]:

	Product	KP281	KP481	KP781
Gender	MaritalStatus			
Female	Partnered	0.150000	0.083333	0.022222
	Single	0.072222	0.077778	0.016667
Male	Partnered	0.116667	0.116667	0.105556
	Single	0.105556	0.055556	0.077778

P[PRODUCT ∩ (GENDER ∩ USAGE)]

In [256]:
```
#P[PRODUCT ∩ (GENDER & USAGE)
pd.crosstab(index= [df["Gender"],df["Usage"]],columns= df["Product"],normalize= True)
```

Out[256]:

Gender	Usage	KP281	KP481	KP781
Female	2	0.072222	0.038889	0.000000
	3	0.105556	0.077778	0.000000
	4	0.038889	0.027778	0.011111
	5	0.005556	0.016667	0.016667
	6	0.000000	0.000000	0.011111
Male	2	0.033333	0.038889	0.000000
	3	0.100000	0.094444	0.005556
	4	0.083333	0.038889	0.088889
	5	0.005556	0.000000	0.050000
	6	0.000000	0.000000	0.027778
	7	0.000000	0.000000	0.011111

P[PRODUCT ∩ (GENDER ∩ FITNESS)]

In [257]:
```
# P[PRODUCT ∩ (GENDER & FITNESS)]
pd.crosstab(index= [df["Gender"],df["Fitness"]],columns= df["Product"],normalize= True)
```

Out[257]:

Gender	Fitness	KP281	KP481	KP781
Female	1	0.000000	0.005556	0.000000
	2	0.055556	0.033333	0.000000
	3	0.144444	0.100000	0.005556
	4	0.016667	0.022222	0.005556
	5	0.005556	0.000000	0.027778
Male	1	0.005556	0.000000	0.000000
	2	0.022222	0.033333	0.000000
	3	0.155556	0.116667	0.016667
	4	0.033333	0.022222	0.033333
	5	0.005556	0.000000	0.133333

P[PRODUCT ∩ (GENDER ∩ INCOME)]

In [259]:
```
# P[PRODUCT ∩ (GENDER & INCOME)]
pd.crosstab(index= [df["Gender"],df["income_category"]],columns= df["Product"],normalize= Tr
```

Out[259]:

Gender	income_category	KP281	KP481	KP781
Female	Moderate	0.188889	0.127778	0.011111
	High	0.033333	0.033333	0.011111
	Elite	0.000000	0.000000	0.016667
Male	Moderate	0.194444	0.138889	0.038889
	High	0.027778	0.033333	0.055556
	Elite	0.000000	0.000000	0.088889

P[PRODUCT ∩ (GENDER & AGE)]

In [260]:
```
# P[PRODUCT ∩ (GENDER & AGE)]
pd.crosstab(index= [df["Gender"],df["Age_category"]],columns= df["Product"],normalize= True)
```

Out[260]:

Gender	Age_category	KP281	KP481	KP781
Female	Young	0.144444	0.077778	0.027778
	Adult	0.061111	0.072222	0.011111
	Senior	0.016667	0.011111	0.000000
Male	Young	0.133333	0.100000	0.111111
	Adult	0.066667	0.055556	0.044444
	Senior	0.022222	0.016667	0.027778

OBSERVATIONS #5

1. P[AGE ∩ PRODUCT] - The young generation shows a strong preference for the KP281 treadmill
2. P[GENDER ∩ PRODUCT]- Both males and females have a high probability of choosing the KP281 treadmill,however kp781 is least prefferesd by females
3. P[PRODUCT ∩ EDUCATION]- Additional data and investigation will help understand how education influences treadmill choices
4. P[PRODUCT ∩ MARITAL STATUS]- Both partnered and single individuals show a strong preference for the KP281 treadmill
5. P[PRODUCT ∩ FITNESS]- Users who perceive themselves as below average in fitness have a lower preference for KP781
6. P[PRODUCT ∩ INCOME]- Customers in the income range of $30, 000 to$50,000 show a preference for both KP281 and KP481 treadmills However, the acceptance of KP781 is lower in the income range of $50, 000 to$ 70,000, with the highest preference observed for KP781 among customers with an income exceeding $70,000
7. P[PRODUCT ∩ MILES]- Customers who ran above 200 miles show a higher preference for the KP781 treadmill, indicating its popularity among more active users.
8. P[PRODUCT ∩ (GENDER & MARITAL STATUS)]- Partnered females prefer KP281, while single females are open to both KP281 and KP481. Partnered males show a preference for both KP281 and KP481, while single males have a higher probability of choosing KP281.

9. P[PRODUCT ∩ (GENDER & USAGE)]- Females tend to avoid buying KP781, while there is no significant impact of usage on product preferences
10. P[PRODUCT ∩ (GENDER & FITNESS)]- Both male and female customers who feel more fit tend to prefer KP281
11. P[PRODUCT ∩ (GENDER & INCOME)]- Families with moderate income levels show a preference for KP281 irrespective of gender.

Conditional Probabilities

P [PRODUCT | AGE]

In [268]:
```
# P[PRODUCT | AGE]
pd.crosstab(index= df["Age_category"],columns= df["Product"],margins=True,normalize="index")
```

Out[268]:

Product	KP281	KP481	KP781
Age_category			
Young	0.467290	0.299065	0.233645
Adult	0.410714	0.410714	0.178571
Senior	0.411765	0.294118	0.294118
All	0.444444	0.333333	0.222222

P [PRODUCT | MARITAL STATUS]

In [265]:
```
# P[PRODUCT | MARITAL STATUS]
pd.crosstab(index= df["MaritalStatus"],columns= df["Product"],margins=True,normalize="index"
```

Out[265]:

Product	KP281	KP481	KP781
MaritalStatus			
Partnered	0.448598	0.336449	0.214953
Single	0.438356	0.328767	0.232877
All	0.444444	0.333333	0.222222

P[PRODUCT | GENDER]

In [269]:
```
# P[PRODUCT | GENDER]
pd.crosstab(index= df["Gender"],columns= df["Product"],margins=True,normalize="index")
```

Out[269]:

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

P[PRODUCT | EDUCATION]

In [270]:
```
# P[PRODUCT | EDUCATION]
pd.crosstab(index= df["Education"],columns= df["Product"],margins=True,normalize="index")
```

Out[270]:

Product Education	KP281	KP481	KP781
12	0.666667	0.333333	0.000000
13	0.600000	0.400000	0.000000
14	0.545455	0.418182	0.036364
15	0.800000	0.200000	0.000000
16	0.458824	0.364706	0.176471
18	0.086957	0.086957	0.826087
20	0.000000	0.000000	1.000000
21	0.000000	0.000000	1.000000
All	0.444444	0.333333	0.222222

P[PRODUCT | FITNESS]

In [271]:
```
# P[PRODUCT | FITNESS]
pd.crosstab(index= df["Fitness"],columns= df["Product"],margins=True,normalize="index")
```

Out[271]:

Product Fitness	KP281	KP481	KP781
1	0.500000	0.500000	0.000000
2	0.538462	0.461538	0.000000
3	0.556701	0.402062	0.041237
4	0.375000	0.333333	0.291667
5	0.064516	0.000000	0.935484
All	0.444444	0.333333	0.222222

P[PRODUCT | INCOME]

In [273]:
```
# P[PRODUCT | INCOME]
pd.crosstab(index= df["income_category"],columns= df["Product"],margins=True,normalize="inde
```

Out[273]:

Product income_category	KP281	KP481	KP781
Moderate	0.547619	0.380952	0.071429
High	0.314286	0.342857	0.342857
Elite	0.000000	0.000000	1.000000
All	0.444444	0.333333	0.222222

P[PRODUCT | (GENDER ∩ MARITAL STATUS)]

In [274]:
```
# P[PRODUCT | (GENDER ∩ MARITAL STATUS)]
pd.crosstab(index= [df["Gender"],df["MaritalStatus"]],columns= df["Product"],normalize= "ind
```

Out[274]:

Gender	MaritalStatus	KP281	KP481	KP781
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

P[PRODUCT | (GENDER ∩ USAGE)]

In [275]:
```
# P[PRODUCT | (GENDER ∩ USAGE)]
pd.crosstab(index= [df["Gender"],df["Usage"]],columns= df["Product"],normalize= "index")
```

Out[275]:

Gender	Usage	KP281	KP481	KP781
Female	2	0.650000	0.350000	0.000000
	3	0.575758	0.424242	0.000000
	4	0.500000	0.357143	0.142857
	5	0.142857	0.428571	0.428571
	6	0.000000	0.000000	1.000000
Male	2	0.461538	0.538462	0.000000
	3	0.500000	0.472222	0.027778
	4	0.394737	0.184211	0.421053
	5	0.100000	0.000000	0.900000
	6	0.000000	0.000000	1.000000
	7	0.000000	0.000000	1.000000

P[PRODUCT | (GENDER ∩ FITNESS)]

In [276]:
```
# P[PRODUCT | (GENDER ∩ FITNESS)]
pd.crosstab(index= [df["Gender"],df["Fitness"]],columns= df["Product"],normalize= "index")
```

Out[276]:

Gender	Fitness	KP281	KP481	KP781
Female	1	0.000000	1.000000	0.000000
	2	0.625000	0.375000	0.000000
	3	0.577778	0.400000	0.022222
	4	0.375000	0.500000	0.125000
	5	0.166667	0.000000	0.833333
Male	1	1.000000	0.000000	0.000000
	2	0.400000	0.600000	0.000000
	3	0.538462	0.403846	0.057692
	4	0.375000	0.250000	0.375000
	5	0.040000	0.000000	0.960000

P [PRODUCT | (GENDER ∩ INCOME)]

In [278]:
```
# P[PRODUCT | (GENDER ∩ INCOME)]
pd.crosstab(index= [df["Gender"],df["income_category"]],columns= df["Product"],normalize= "i
```

Out[278]:

Gender	income_category	KP281	KP481	KP781
Female	Moderate	0.576271	0.389831	0.033898
	High	0.428571	0.428571	0.142857
	Elite	0.000000	0.000000	1.000000
Male	Moderate	0.522388	0.373134	0.104478
	High	0.238095	0.285714	0.476190
	Elite	0.000000	0.000000	1.000000

P [PRODUCT | (GENDER ∩ AGE)]

In [279]:
```
# P[PRODUCT | (GENDER ∩ AGE)]
pd.crosstab(index= [df["Gender"],df["Age_category"]],columns= df["Product"],normalize= "inde
```

Out[279]:

	Product	KP281	KP481	KP781
Gender	Age_category			
Female	Young	0.577778	0.311111	0.111111
	Adult	0.423077	0.500000	0.076923
	Senior	0.600000	0.400000	0.000000
Male	Young	0.387097	0.290323	0.322581
	Adult	0.400000	0.333333	0.266667
	Senior	0.333333	0.250000	0.416667

Observations #6

1. P[PRODUCT | AGE] - The KP281 treadmill is a favorite among the youth generation (aged 18-28). For adults, there is a balanced preference between KP281 and KP481 while Seniors are more interested in KP481 and KP781 models.
2. P[PRODUCT | GENDER] - Females show a lower preference for KP781 compared to other products
3. P[PRODUCT | EDUCATION]- Users with 12-15 years of education demonstrate a strong affinity for KP281, while those with more than 15 years of education prefer KP781.
4. P[PRODUCT | MARITAL STATUS]- KP281 has broad appeal among both married and single individuals
5. P[PRODUCT | FITNESS]- Customers who perceive themselves as average or below average in fitness tend to prefer KP281
6. P[PRODUCT | INCOME]- KP281 suits individuals with moderate income, while KP781 appeals to the elite class.
7. P[PRODUCT | (GENDER & MARITAL STATUS)]- Customers who are female and partnered or male and single have a high probability of buying KP281
8. P[PRODUCT | (GENDER & USAGE)]- Female users who plan to use the treadmill 2 to 4 days a week prefer KP281, while male users with more than 3 days a week usage lean towards KP781.
9. P[PRODUCT | (GENDER & FITNESS)]- Users who feel more fit tend to opt for KP481 or KP781.
10. P[PRODUCT | (GENDER & INCOME)]- High-income females prefer KP281 or KP481, while high-income males prefer KP481 or KP781
11. P[PRODUCT | (GENDER & AGE)]- Young males and females prefer KP281, adult females favor KP481, and adult males prefer KP281. Seniors show interest in KP781

Customer Profiling

KP281 Treadmill

Target Customer: The KP281 treadmill caters to the young generation (aged 18-28) who seek a versatile fitness solution. They prefer regular workouts and value a treadmill with a good mix of features and affordability.

Key Characterstics : These users are typically between the ages of 18 and 28, well-educated with 12-15 years of education, and have a moderate income. They use the treadmill 3 to 4 days a week and consider themselves average or below average in fitness. KP281 aligns well with their preferences.

Additional Details: Single female & Partnered male customers bought this product more than single male

KP481 Treadmill

Target Customer: The KP481 treadmill appeals to adults (aged 26-35) looking for a well-rounded fitness experience. They seek a balance between basic and advanced features, making the KP481 a suitable choice.This is the second most popular choice of users.

Key Characterstics : These customers are generally between the ages of 26 and 35, with varied education levels. They use the treadmill 3 to 4 days a week, and their fitness levels range from average to above average. KP481 caters to their preferences for a versatile workout routine.

Additional Details: KP481 product is specifically recommended for Female customers who are intermediate user. Probability of Female customer buying KP481 is significantly higher than male.

KP781 Treadmill

Target Customer: The KP781 treadmill is designed for customers seeking high-performance fitness equipment. It appeals to those with higher income and fitness enthusiasts aiming for intense workouts.

Key Characterstics : These customers have a diverse age range, with seniors (aged 34-42) showing particular interest. They have more than 15 years of education and higher income levels, often exceeding $70,000. They use the treadmill frequently, running more than 200 miles. KP781's advanced features and performance capabilities are well-suited to their fitness goals.

Additional Details: Probability of Male customer buying Product KP781(31.73%) is way more than female(9.21%). This the least preferred product for females. This product is preferred by the customer where the correlation between Education and Income is High.

Recommendations

KP281 Treadmill

Target Audience : Young Generation (aged 18-28) with moderate income

1. Social Media Campaigns - Utilize platforms like Instagram, Facebook, and YouTube to showcase engaging workout videos, success stories, and challenges using the KP281 treadmill.
2. University Partnerships- Partner with universities and fitness clubs to offer special discounts and promotions to students. Conduct on-campus events or fitness challenges to raise awareness and generate interest in the KP281 treadmill.

KP481 Treadmill

Target Audience: Adult Generation (aged 26-35) with varied income levels

1. Free Trial Period: Offer a limited-time free trial period for the KP481 treadmill, allowing potential customers to experience its features before committing to a purchase.
2. Online Ad Campaigns: Online platforms like Facebook and Google or amazon , myntra can be targetted to reach the adult audience.

KP781 Treadmill

Target Audience: Seniors (aged 34-42) with higher income levels and extensive exercise routines

1. Premium Branding: Position the KP781 treadmill as a premium fitness equipment brand with an emphasis on advanced technology and durability
2. Personalized Training Services: Personalized training services can boost up sales since elder population may not know exact usage of product
3. Promotions by Atheletes - The product can be branded among top most brands used by Atheletes or elite peoples.

In [ ]:

```

```