Aerofit Case Study

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from scipy.stats import norm
import matplotlib.pyplot as plt
from statsmodels.distributions.empirical_distribution import ECDF
```

Table of contents

1. Initial Exploration

1.1 Problem Statement

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.

1.2 Basic Analysis

In [2]:
```
df = pd.read_csv("aerofit_treadmill.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

In [3]:
```
print(f"Shape of data {df.shape}")
```

```
Shape of data (180, 9)
```

180 rows and 9 Columns

In [4]:
```
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
```

- No null values observed
- 6 Numerical Columns
- 3 Categorical Columns

In [5]:
```
print(f"Null Values: {df.isna().sum().sum()}")
```

```
Null Values: 0
```

1.3 Data Type Conversion

Discrete Data Type to Categorical

In [6]:
```
df["Education"] = df["Education"].astype("object")
df["Usage"] = df["Usage"].astype("object")
df["Fitness"] = df["Fitness"].astype("object")
```

In [7]:
```
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    object
 4   MaritalStatus  180 non-null    object
 5   Usage          180 non-null    object
 6   Fitness        180 non-null    object
 7   Income         180 non-null    int64 
 8   Miles          180 non-null    int64 
dtypes: int64(3), object(6)
```



```
12.8+ KB
```

Income in 1000's(K)

In [8]:
```
df["Income"] = round(df["Income"]/1000,2)
```

1.4 Statistical Analysis

In [9]:
```
numerical_analysis = df.describe(percentiles=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]).T
numerical_analysis
```

Out[9]:

	count	mean	std	min	10%	20%	30%	40%	50%	60%	70%	80%	90%	max
Age	180.0	28.788889	6.943498	18.00	22.00	23.00	24.70	25.00	26.000	29.000	31.300	34.2	38.10	50.00
Income	180.0	53.720000	16.507097	29.56	35.25	40.93	45.48	47.75	50.595	52.756	55.059	61.4	83.42	104.58
Miles	180.0	103.194444	51.863605	21.00	53.00	65.60	75.00	85.00	94.000	100.000	107.800	132.0	180.00	360.00

- Average Age is 29 yrs old
- Average I ncome is 53K
- Average miles expected to covered is 103

In [10]:
```
categorical_analysis = df.describe(include="object").T
categorical_analysis
```

Out[10]:

	count	unique	top	freq
Product	180	3	KP281	80
Gender	180	2	Male	104
Education	180	8	16	85
MaritalStatus	180	2	Partnered	107
Usage	180	6	3	69
Fitness	180	5	3	97

1.4.1 Interquartile Range

In [11]:
```
def iqr(x):
    q3 = x.quantile(0.75)
    q1 = x.quantile(0.25)
    iqr = round(q3-q1,2)
    print(f"Interquartile Range: {iqr}")
    lower = round(max(q1 - 1.5 * iqr, 0),2)
    upper = round(q3 + 1.5 * iqr,2)
    print(f"Range from ({lower},{upper})")
```

In [12]:
```
df["Income"].max() - df["Income"].min()
```

Out[12]:
```
75.02
```

In [13]:
```
for col in df.columns[df.dtypes != "object"]:
    print(col,":-")
    print("-"*30)
    iqr(df[col])
    print(f"Min Value {df[col].min()}, Max Value {df[col].max()}")
    print("="*30,end="\n\n")
```

```
Age :-
------------------------------
Interquartile Range: 9.0
Range from (10.5,46.5)
Min Value 18, Max Value 50
==============================

Income :-
------------------------------
Interquartile Range: 14.61
Range from (22.14,80.59)
Min Value 29.56, Max Value 104.58
==============================

Miles :-
------------------------------
Interquartile Range: 48.75
Range from (0,187.88)
Min Value 21, Max Value 360
==============================
```

- Major Outliers observed in Miles & Income
- Spread of Miles is quite high
- Spread of age is low

1.5 Unique Values & Value Counts

Product

In [14]:
```
print(f'Total Unique values: {df["Product"].nunique()}')
df["Product"].value_counts()
```

Out[14]:
```
Total Unique values: 3
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

- Most common bough product is KP281

Age

In [15]:
```
print(f'Total Unique values: {df["Age"].nunique()}')
```

```
Total Unique values: 32
```

Gender

In [16]:
```
print(f'Total Unique values: {df["Gender"].nunique()}')
df["Gender"].value_counts()
```

Out[16]:
```
Total Unique values: 2
Male      104
Female     76
Name: Gender, dtype: int64
```

- Male predominately buys treadmill

Education

In [17]:
```
print(f'Total Unique values: {df["Education"].nunique()}')
df["Education"].value_counts()
```

```
Total Unique values: 8
```

Out[17]:
```
16    85
14    55
18    23
15     5
13     5
12     3
21     3
20     1
Name: Education, dtype: int64
```

- 16 years most common education

Marital Status

In [18]:
```
print(f'Total Unique values: {df["MaritalStatus"].nunique()}')
df["MaritalStatus"].value_counts()
```

```
Total Unique values: 2
```

Out[18]:
```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

- Couples are more likely to buy treadmill

Usage

In [19]:
```
print(f'Total Unique values: {df["Usage"].nunique()}')
df["Usage"].value_counts()
```

```
Total Unique values: 6
```

Out[19]:
```
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

- The majority of customers expected usage to be 3

Fitness

In [20]:
```
print(f'Total Unique values: {df["Fitness"].nunique()}')
df["Fitness"].value_counts()
```

```
Total Unique values: 5
```

Out[20]:
```
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

- The majority of customers consider themselves to be of average fitness level.

Income

In [21]:
```
print(f'Total Unique values: {df["Income"].nunique()}')
```

```
Total Unique values: 61
```

Miles

In [22]:
```
print(f'Total Unique values: {df["Miles"].nunique()}')
```

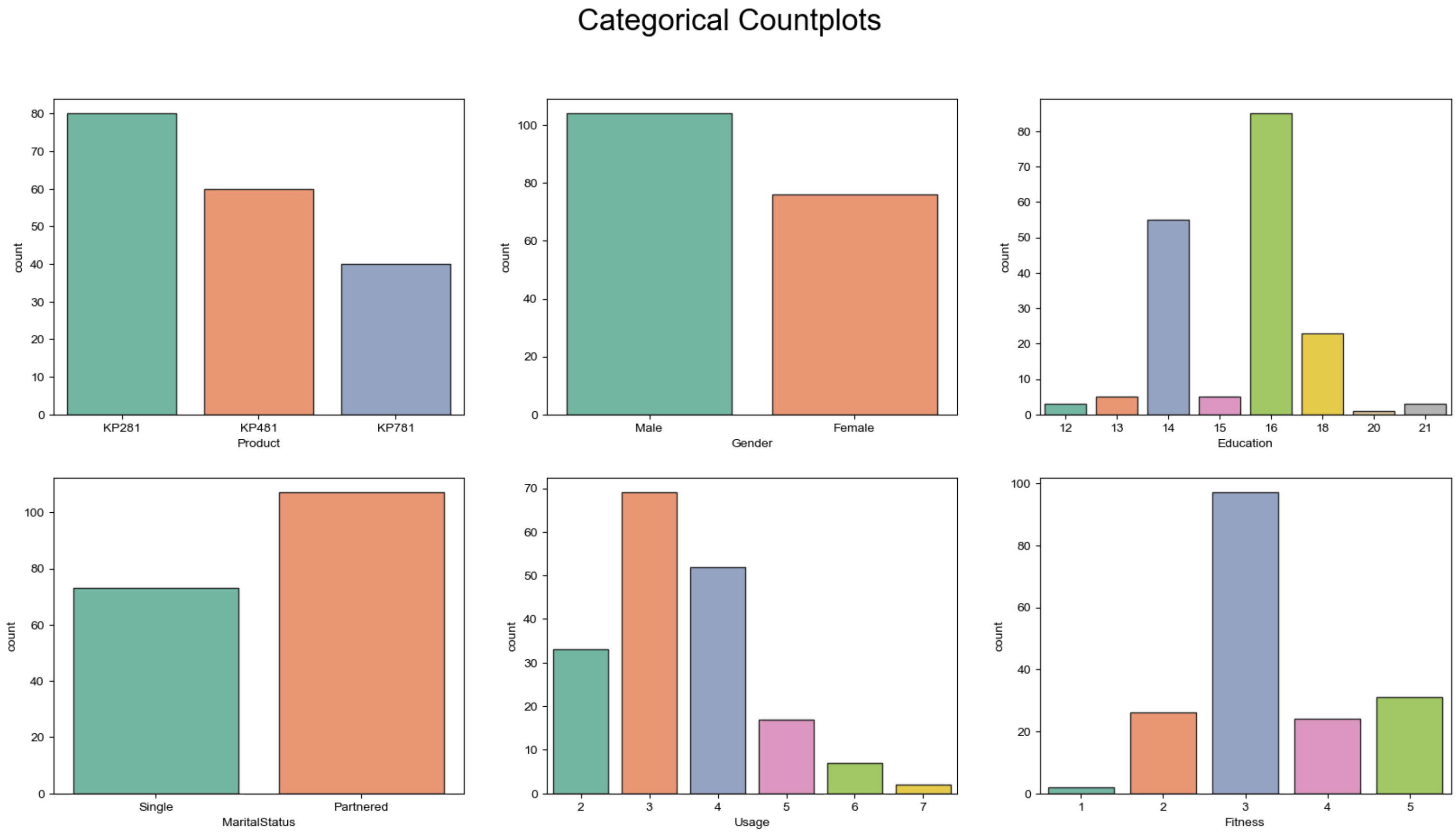
```
Total Unique values: 37
```

2. Visual Analysis

2.1 Univariate Analysis

2.1.1 Categorical Analysis

In [23]:
```
fig, ax = plt.subplots(2,3,figsize=(20,10))
plt.suptitle("Categorical Countplots",fontsize=24)

sns.set_palette(palette="Set2",n_colors=8)
sns.set_style(style="whitegrid")

sns.countplot(x = df["Product"],edgecolor = "0.15",ax=ax[0,0])
sns.countplot(x = df["Gender"],edgecolor = "0.15",ax=ax[0,1])
sns.countplot(x = df["Education"],edgecolor = "0.15",ax=ax[0,2])
sns.countplot(x = df["MaritalStatus"],edgecolor = "0.15",ax=ax[1,0])
sns.countplot(x = df["Usage"],edgecolor = "0.15",ax=ax[1,1])
sns.countplot(x = df["Fitness"],edgecolor = "0.15",ax=ax[1,2])

plt.show()
```



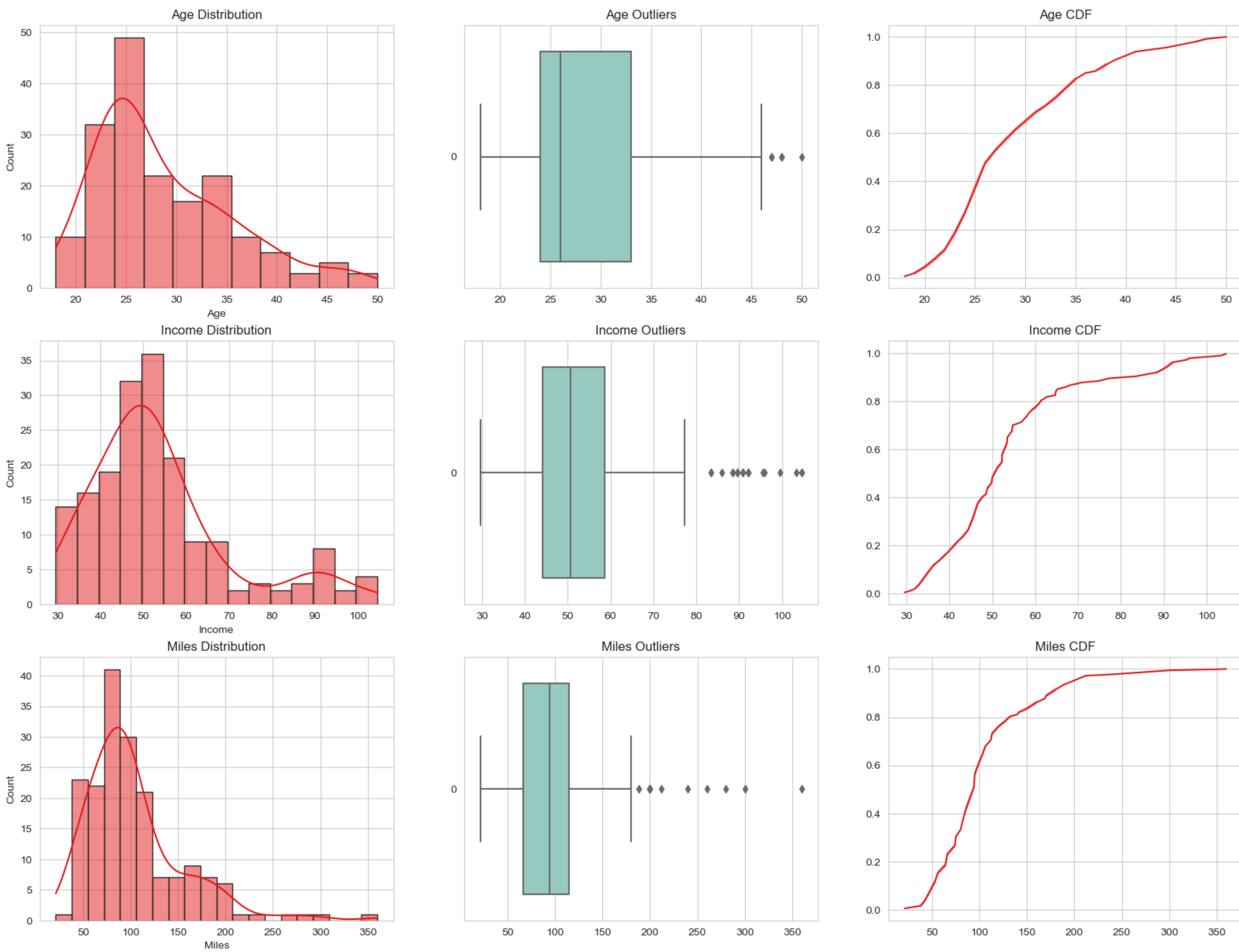
- KP281 is the most commonly ordered treadmill.
- Males account for the majority of the orders.
- The most common years of education among customers are 16, followed by 14.
- The majority of customers are married or have a partner.
- The usage of the treadmills is generally low.
- Most customers consider their fitness level to be average.

2.1.2 Continuous Columns

In [72]:

```python
fig,ax = plt.subplots(3,3,figsize=(20,15))
plt.suptitle("Continuous Columns",fontsize=24)

sns.set_palette(palette="Set3",n_colors=8)
sns.set_style(style="whitegrid")

ax[0,0].set_title("Age Distribution")
sns.histplot(df["Age"],kde="True",edgecolor = "0.15", ax=ax[0,0])

ax[0,1].set_title("Age Outliers")
sns.axes_style(style="white")
sns.boxplot(df["Age"],orient='h', ax=ax[0,1])

ax[0,2].set_title("Age CDF")
e = ECDF(df["Age"])
sns.axes_style(style="whitegrid")
sns.lineplot(x = e.x, y = e.y,ax=ax[0,2])

ax[1,0].set_title("Income Distribution")
sns.histplot(df["Income"],kde="True",edgecolor = "0.15", ax=ax[1,0])

ax[1,1].set_title("Income Outliers")
sns.boxplot(df["Income"],orient='h', ax=ax[1,1])

ax[1,2].set_title("Income CDF")
e = ECDF(df["Income"])
sns.lineplot(x = e.x, y = e.y,ax=ax[1,2])

ax[2,0].set_title("Miles Distribution")
sns.histplot(df["Miles"],kde="True",edgecolor = "0.15", ax=ax[2,0])

ax[2,1].set_title("Miles Outliers")
sns.boxplot(df["Miles"],orient='h', ax=ax[2,1])

ax[2,2].set_title("Miles CDF")
e = ECDF(df["Miles"])
sns.lineplot(x = e.x, y = e.y,ax=ax[2,2])

plt.show()
```

Continuous Columns



- The median age of customers is 26, with a few outliers in the upper whisker.
- In terms of age, 80 percent of customers have an age less than 35.
- The median income of customers is 50K, with outliers in the upper whisker.
- Among income levels, 80 percent of customers have an income less than 60K.
- The median expected miles per week is 95, with outliers in the upper whisker.
- Among customers, 80 percent expect to run at least 130 miles per week.

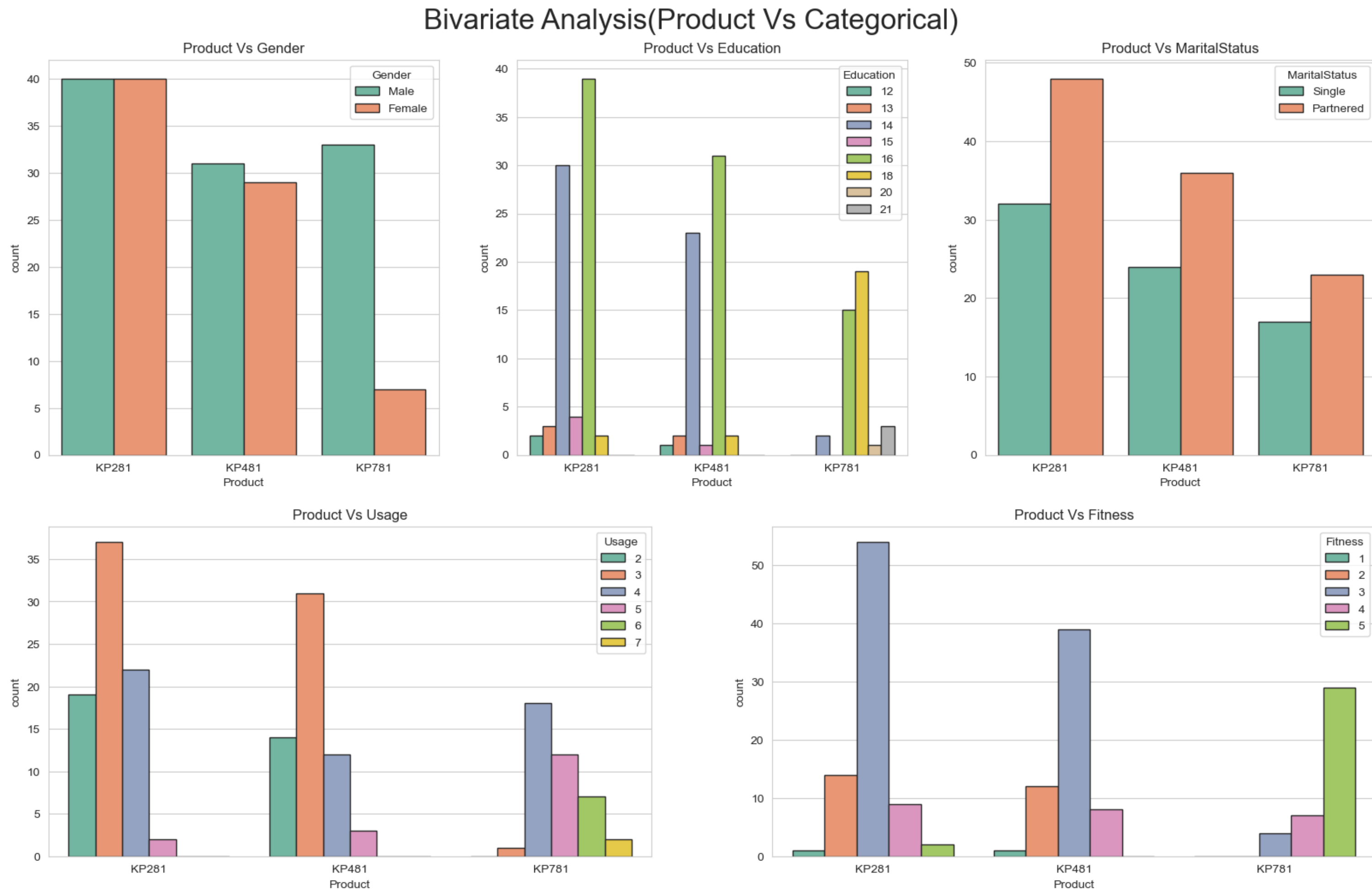
2.2 Bivariate Analysis

2.2.1 Categorical Columns

In [25]:

```
fig,ax = plt.subplots(1,3,figsize=(20,6))
plt.suptitle("Bivariate Analysis(Product Vs Categorical)",fontsize=24)

sns.set_palette(palette="Set2",n_colors=8)
sns.set_style(style="whitegrid")

cols = df.columns[(df.columns!="Product") & (df.dtypes == "object") ].to_list()
for i in range(1):
    for j in range(3):
        ax[j].set_title("Product Vs " + cols[0])
        if df[cols[0]].dtype == int:
            sns.boxplot(data=df, x = "Product", y = cols[0], ax=ax[j])
        else:
            sns.countplot(data=df, x = "Product", hue = cols[0], edgecolor = "0.15", ax=ax[j])
        cols.pop(0)

fig,ax = plt.subplots(1,2,figsize=(20,5))
sns.set_palette(palette="Set2",n_colors=8)
sns.set_style(style="whitegrid")

for i in range(1):
    for j in range(2):
        ax[j].set_title("Product Vs " + cols[0])
        if df[cols[0]].dtype == int:
            sns.boxplot(data=df, x = "Product", y = cols[0], ax=ax[j])
        else:
            sns.countplot(data=df, x = "Product", hue = cols[0], edgecolor = "0.15", ax=ax[j])
        cols.pop(0)
plt.show()
```

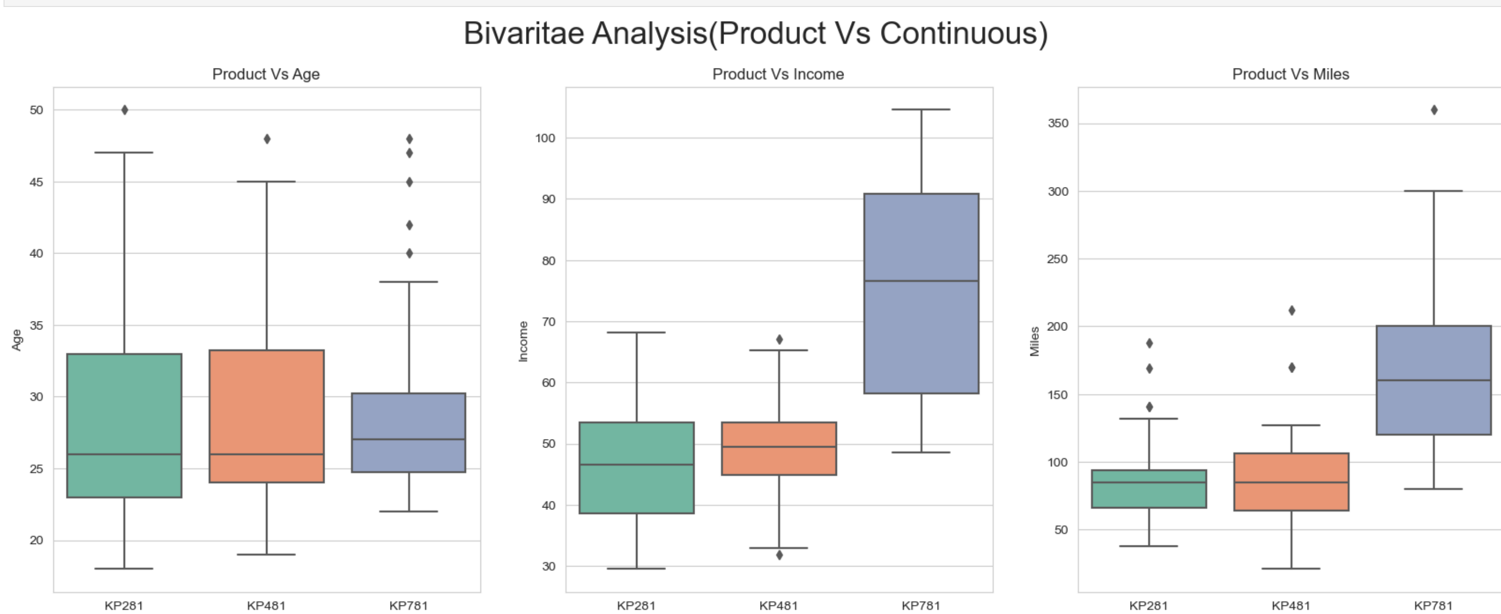
Loading [MathJax]/extensions/Safe.js



- Male customers show a higher preference for ordering KP481 and KP781 treadmills.
- Customers have higher expected usage for KP781 compared to other treadmill models.
- Customers who purchase KP781 tend to consider themselves more fit than others.

2.2.2 Continuous Columns

In [26]:
```
fig,ax = plt.subplots(1,3,figsize=(20,7))
plt.suptitle("Bivaritae Analysis(Product Vs Continuous)",fontsize=24)

sns.set_palette(palette="Set2",n_colors=8)
sns.set_style(style="whitegrid")

cols = df.columns[(df.columns!="Product") & (df.dtypes != "object") ].to_list()
for i in range(1):
    for j in range(3):
        ax[j].set_title("Product Vs " + cols[0])
        if df[cols[0]].dtype != "object":
            sns.boxplot(data=df, x = "Product", y = cols[0], ax=ax[j])
        else:
            sns.countplot(data=df, x = "Product", hue = cols[0], edgecolor = "0.15", ax=ax[j])
        cols.pop(0)
plt.show()
```



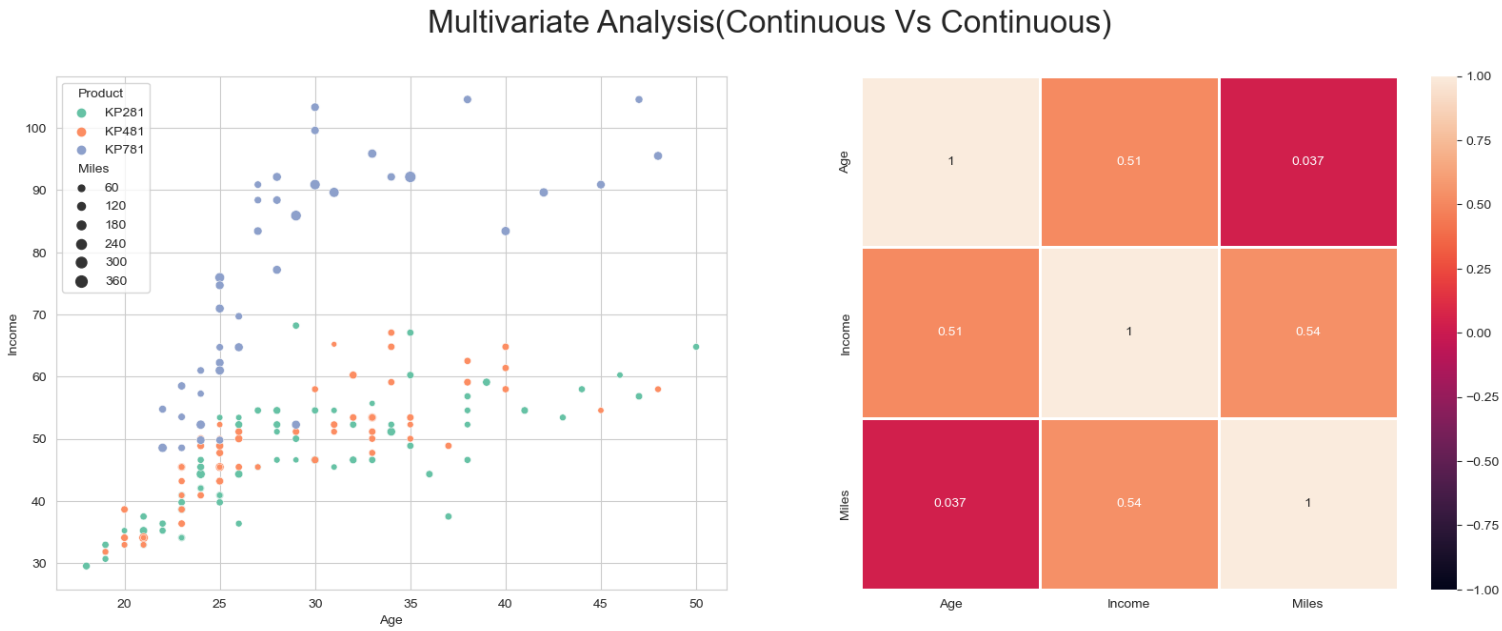
- KP781 treadmills are predominantly purchased by the adult audience.

- Customers with a higher income bracket are more likely to buy KP781.
- Individuals who purchase KP781 tend to have a higher inclination to run more compared to others.

2.3 Multivariate Analysis

Continuous Vs Continuous Vs Categorical

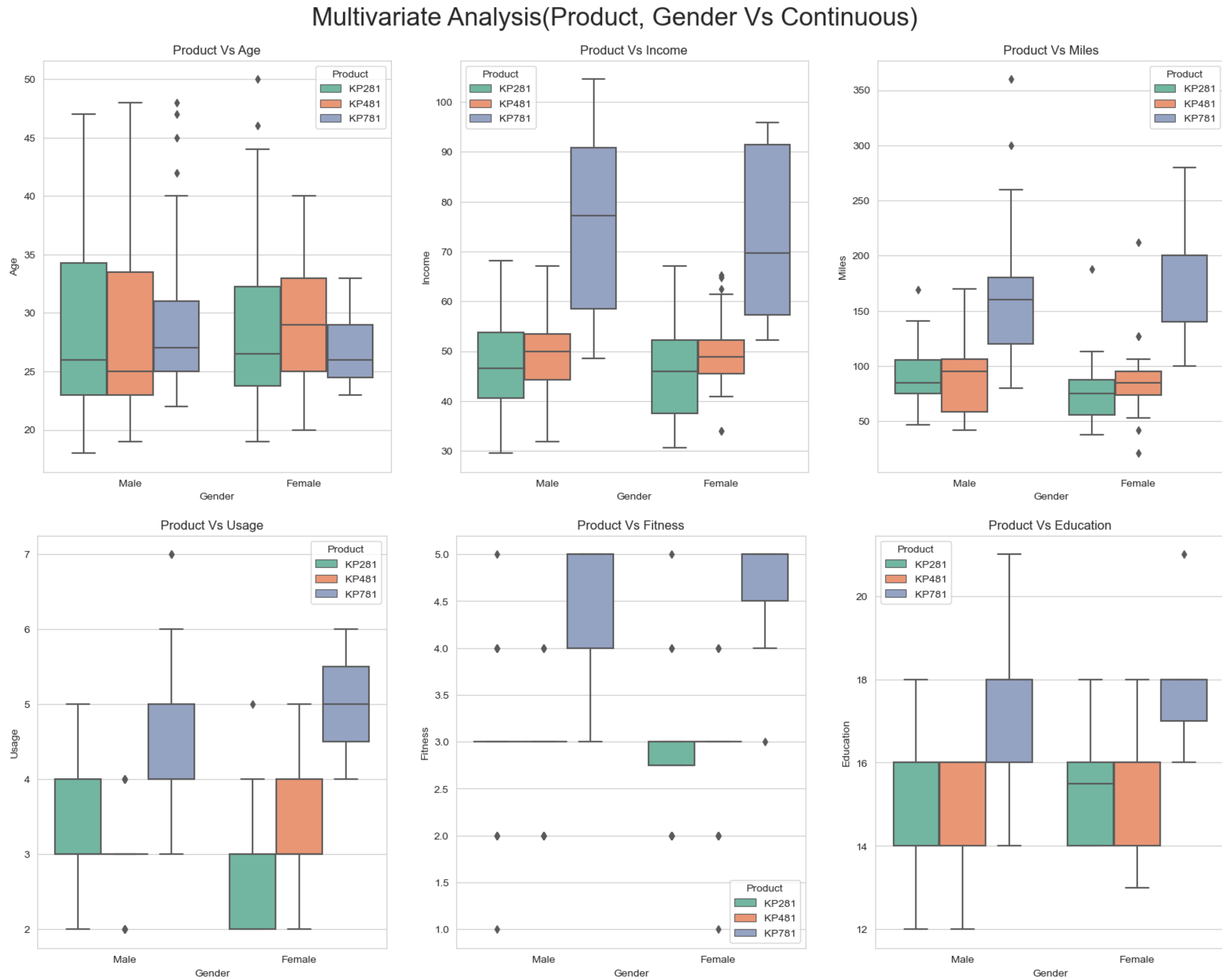
In [27]:
```
fig,ax = plt.subplots(1,2,figsize=(20,7))
plt.suptitle("Multivariate Analysis(Continuous Vs Continuous)",fontsize=24)

sns.set_palette(palette="Set2",n_colors=8)
sns.set_style(style="whitegrid")

df_corr = df.copy()

sns.scatterplot(data=df,x="Age",y="Income",size="Miles",hue="Product",ax=ax[0])
sns.heatmap(df_corr.corr(),annot=True,linewidths=1,ax=ax[1],vmin=-1)


plt.show()
```



- There is a mild positive correlation between income and age among the customers.
- KP781 users in the age range of 23-37 tend to cover the highest number of miles on average.

Product Gender

In [28]:
```
fig,ax = plt.subplots(1,3,figsize=(20,7))
plt.suptitle("Multivariate Analysis(Product, Gender Vs Continuous)",fontsize=24)

sns.set_palette(palette="Set2",n_colors=8)
sns.set_style(style="whitegrid")

cols = df.columns[(df.columns!="Product") & (df.dtypes != "object") ].to_list()
for i in range(1):
    for j in range(3):
        ax[j].set_title("Product Vs " + cols[0])
        sns.boxplot(data=df, x = "Gender", y = cols[0], ax=ax[j],hue="Product")
        cols.pop(0)

fig,ax = plt.subplots(1,3,figsize=(20,7))
cols = ["Usage","Fitness","Education"]
for i in range(1):
    for j in range(3):
        ax[j].set_title("Product Vs " + cols[0])
        sns.boxplot(data=df, x = "Gender", y = cols[0], ax=ax[j],hue="Product")
        cols.pop(0)

plt.show()
```

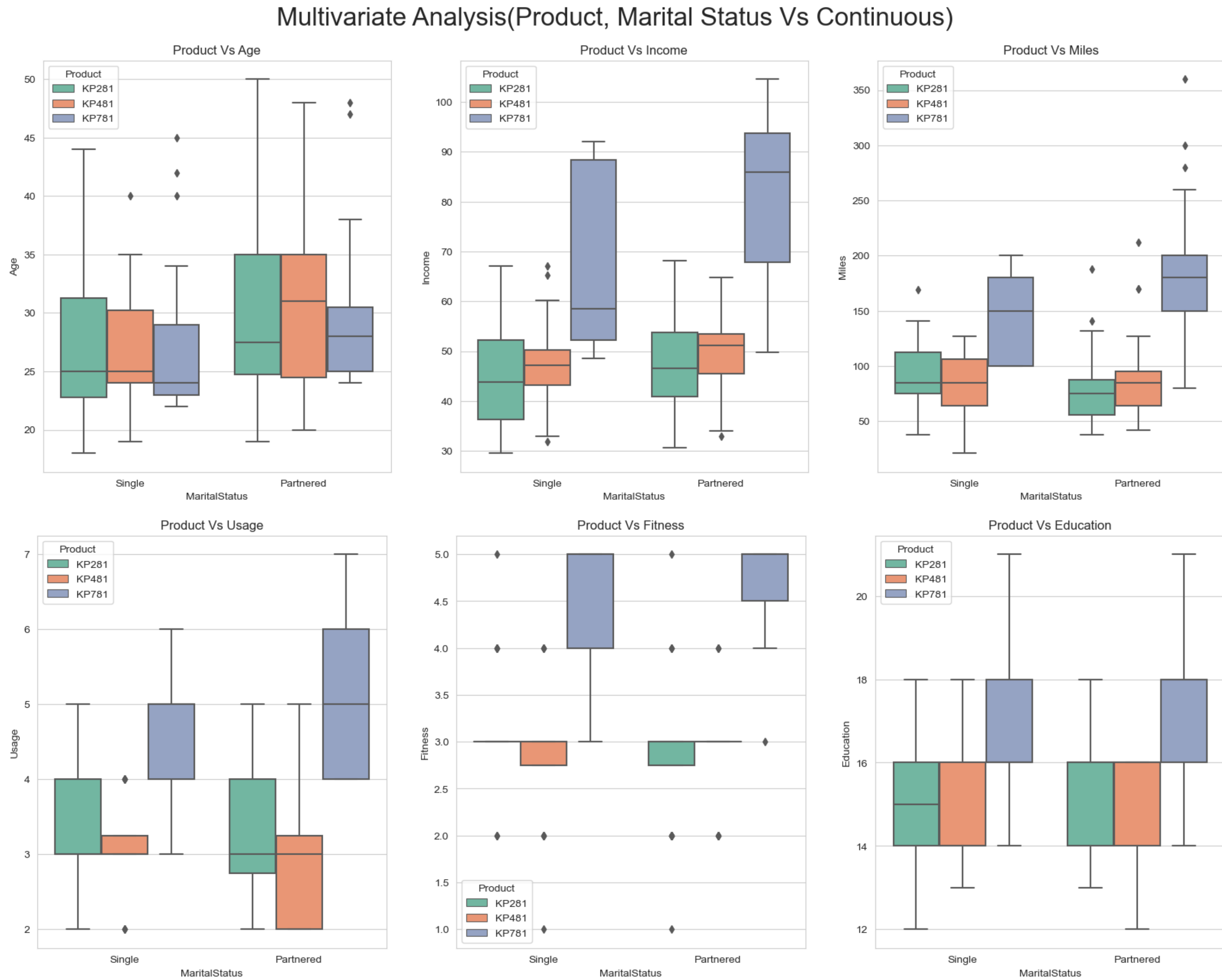


- Female customers show a preference for KP481, while male customers prefer KP781.
- Customers who have higher education tend to prefer KP781.
- KP281 customers do not have plans to use the treadmill frequently.

Product Marital Status

In [29]:

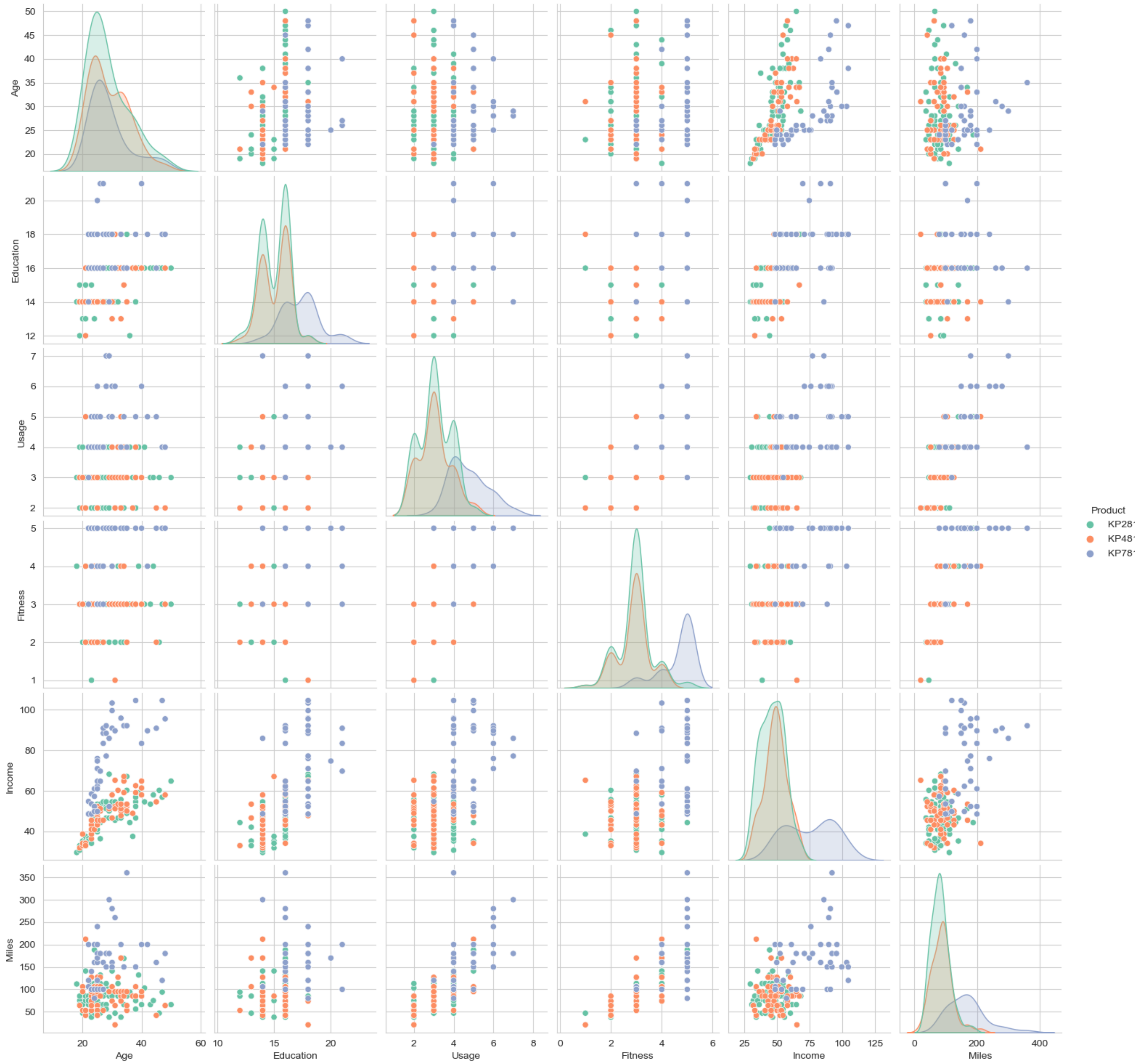
```
fig,ax = plt.subplots(1,3,figsize=(20,7))
plt.suptitle("Multivariate Analysis(Product, Marital Status Vs Continuous)",fontsize=24)

sns.set_palette(palette="Set2",n_colors=8)
sns.set_style(style="whitegrid")

cols = df.columns[(df.columns!="Product") & (df.dtypes != "object") ].to_list()
for i in range(1):
    for j in range(3):
        ax[j].set_title("Product Vs " + cols[0])
        sns.boxplot(data=df, x = "MaritalStatus", y = cols[0], ax=ax[j],hue="Product")
        cols.pop(0)

fig,ax = plt.subplots(1,3,figsize=(20,7))
cols = ["Usage","Fitness","Education"]
for i in range(1):
    for j in range(3):
        ax[j].set_title("Product Vs " + cols[0])
        sns.boxplot(data=df, x = "MaritalStatus", y = cols[0], ax=ax[j],hue="Product")
        cols.pop(0)

plt.show()
```



- Couples show a preference for KP481 treadmills.

Pairplot

In [30]:
```
sns.set_palette(palette="Set2",n_colors=8)
sns.set_style(style="whitegrid")
sns.pairplot(data=df,hue="Product")
plt.show()
```



- It is important to note that correlations involving discrete values may lead to misleading insights.
- There is a mild correlation between age and miles.
- Income and miles are correlated.
- Usage and miles are correlated.

3. Probability

Binning based on IQR Range

```
Age :-
-----------------------------
Interquartile Range: 9.0
Range from (10.5,46.5)
=============================

Income :-
-----------------------------
Interquartile Range: 14.61
Range from (22.14,80.59)
=============================

Miles :-
-----------------------------
Interquartile Range: 48.75
Range from (0,187.88)
=============================
```

In [31]:

```
bins = np.arange(0,70,10)
labels = ['<10','10-20','20-30','30-40','40-50','50+']
df['AgeBin'] = pd.cut(df['Age'], bins=bins, labels=labels)

bins = np.arange(0,140,20)
labels = ['<20','20-40','40-60','60-80','80-100','100+']
'] = pd.cut(df['Income'], bins=bins, labels=labels)
```

```
bins = np.arange(0,280,40)
labels = ['<40','40-80','80-120','120-160','160-200','200+']
df['MilesBin'] = pd.cut(df['Miles'], bins=bins, labels=labels)

bins = [0,12,14,16,18,20,22]
labels = ['0-12','12-14','14-16','16-18','18-20','20+']
df['EducationBin'] = pd.cut(df['Education'], bins=bins, labels=labels)

df.head()
```

Out[31]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeBin	IncomeBin	MilesBin	EducationBin
0	KP281	18	Male	14	Single	3	4	29.56	112	10-20	20-40	80-120	12-14
1	KP281	19	Male	15	Single	2	3	31.84	75	10-20	20-40	40-80	14-16
2	KP281	19	Female	14	Partnered	4	3	30.70	66	10-20	20-40	40-80	12-14
3	KP281	19	Male	12	Single	3	3	32.97	85	10-20	20-40	80-120	0-12
4	KP281	20	Male	13	Partnered	4	2	35.25	47	10-20	20-40	40-80	12-14

Gender

In [32]:
```
gender_prob = round(pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True, normalize=True),2)
gender_prob
```

Out[32]:

Product Gender	KP281	KP481	KP781	All
Female	0.22	0.16	0.04	0.42
Male	0.22	0.17	0.18	0.58
All	0.44	0.33	0.22	1.00

In [33]:
```
product_prob_given_gender = pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
product_prob_given_gender
```

Out[33]:

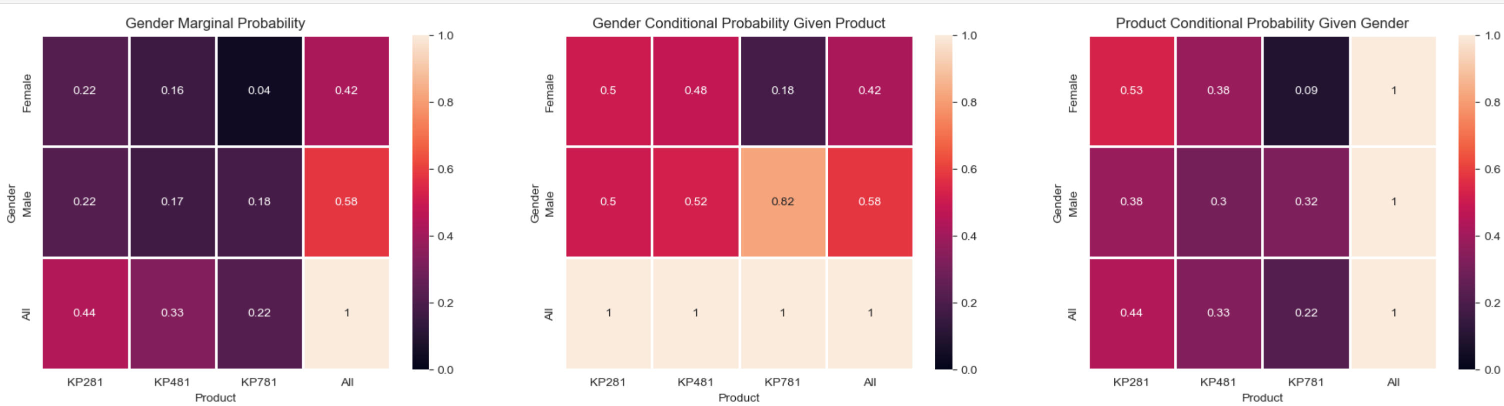
Product Gender	KP281	KP481	KP781	All
Female	0.53	0.38	0.09	1.0
Male	0.38	0.30	0.32	1.0
All	0.44	0.33	0.22	1.0

In [34]:
```
gender_prob_given_product = pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)
gender_prob_given_product
```

Out[34]:

Product Gender	KP281	KP481	KP781	All
Female	0.5	0.48	0.18	0.42
Male	0.5	0.52	0.82	0.58
All	1.0	1.00	1.00	1.00

In [35]:
```
fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Gender Marginal Probability")
sns.heatmap(gender_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Gender Conditional Probability Given Product")
sns.heatmap(gender_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Gender")
sns.heatmap(product_prob_given_gender, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```



- There is a higher likelihood of males purchasing KP781 treadmills.
- KP281 treadmills are predominantly favored by women.
- Women show less interest in purchasing KP781 treadmills compared to other products.

Marital Status

In [36]:
```
marital_prob = round(pd.crosstab(index=df['MaritalStatus'],columns=df['Product'],margins=True, normalize=True),2)
marital_prob
```

Out[36]:

Product MaritalStatus	KP281	KP481	KP781	All
Partnered	0.27	0.20	0.13	0.59
Single	0.18	0.13	0.09	0.41
All	0.44	0.33	0.22	1.00

In [37]:
```
product_prob_given_marital = pd.crosstab(index=df['MaritalStatus'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
product_prob_given_marital
```

Loading [MathJax]/extensions/Safe.js

Out[37]:

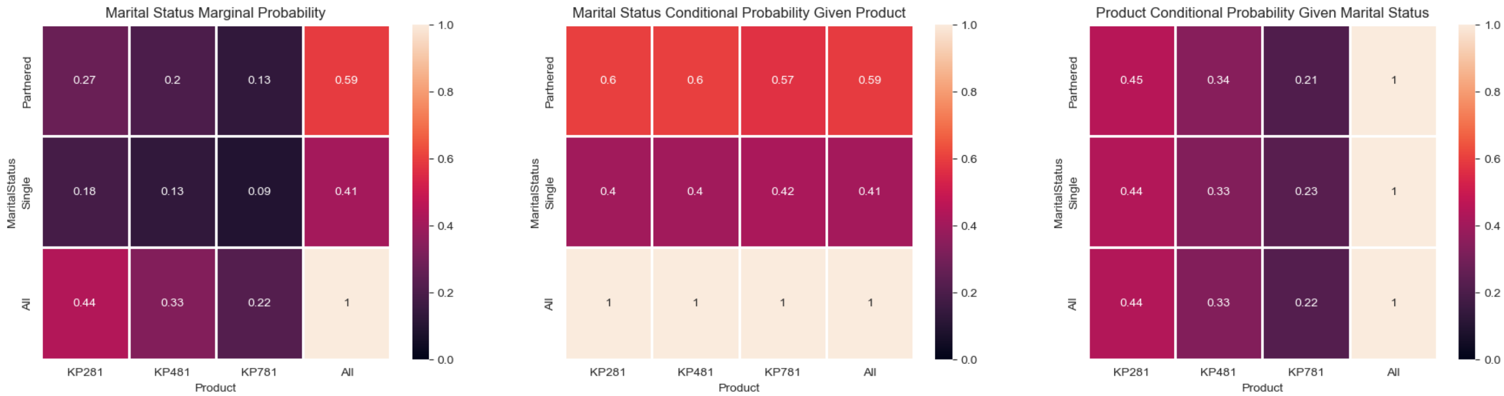
Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	0.45	0.34	0.21	1.0
Single	0.44	0.33	0.23	1.0
All	0.44	0.33	0.22	1.0

In [38]:
```
marital_prob_given_product = pd.crosstab(index=df['MaritalStatus'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)
marital_prob_given_product
```

Out[38]:

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	0.6	0.6	0.57	0.59
Single	0.4	0.4	0.42	0.41
All	1.0	1.0	1.00	1.00

In [39]:
```
fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Marital Status Marginal Probability")
sns.heatmap(marital_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Marital Status Conditional Probability Given Product")
sns.heatmap(marital_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Marital Status")
sns.heatmap(product_prob_given_marital, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```



- The target audience for the treadmills is expected to primarily consist of married individuals.

Usage

In [40]:
```
usage_prob = round(pd.crosstab(index=df['Usage'],columns=df['Product'],margins=True, normalize=True),2)
usage_prob
```

Out[40]:

Product	KP281	KP481	KP781	All
Usage				
2	0.11	0.08	0.00	0.18
3	0.21	0.17	0.01	0.38
4	0.12	0.07	0.10	0.29
5	0.01	0.02	0.07	0.09
6	0.00	0.00	0.04	0.04
7	0.00	0.00	0.01	0.01
All	0.44	0.33	0.22	1.00

In [41]:
```
product_prob_given_usage = pd.crosstab(index=df['Usage'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
product_prob_given_usage
```

Out[41]:

Product	KP281	KP481	KP781	All
Usage				
2	0.58	0.42	0.00	1.0
3	0.54	0.45	0.01	1.0
4	0.42	0.23	0.35	1.0
5	0.12	0.18	0.71	1.0
6	0.00	0.00	1.00	1.0
7	0.00	0.00	1.00	1.0
All	0.44	0.33	0.22	1.0

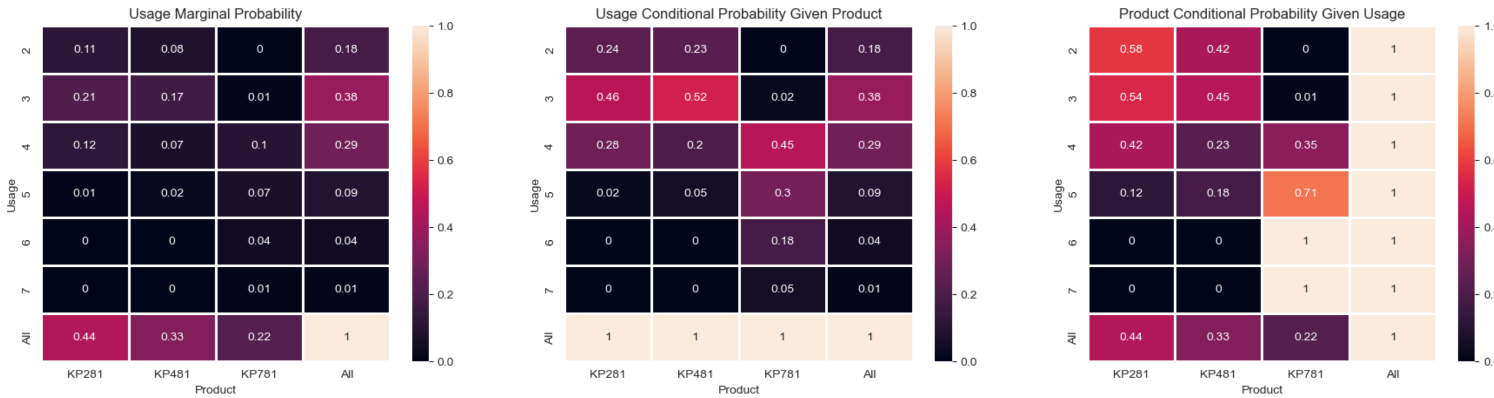
In [42]:
```
usage_prob_given_product = pd.crosstab(index=df['Usage'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)
usage_prob_given_product
```

Out[42]:

Product	KP281	KP481	KP781	All
Usage				
2	0.24	0.23	0.00	0.18
3	0.46	0.52	0.02	0.38
4	0.28	0.20	0.45	0.29
5	0.02	0.05	0.30	0.09
6	0.00	0.00	0.18	0.04
7	0.00	0.00	0.05	0.01
All	1.00	1.00	1.00	1.00

In [43]:
```
fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Usage Marginal Probability")
sns.heatmap(usage_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)
```

```python
ax[1].set_title("Usage Conditional Probability Given Product")
sns.heatmap(usage_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Usage")
sns.heatmap(product_prob_given_usage, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```



- Customers who rate their treadmill usage as 3 are more inclined to purchase KP281 treadmills.
- Buyers who rate their usage as 5 or above are more likely to opt for KP781 treadmills.

Fitness

In [44]:
```python
fitness_prob = round(pd.crosstab(index=df['Fitness'],columns=df['Product'],margins=True, normalize=True),2)
fitness_prob
```

Out[44]:

Product Fitness	KP281	KP481	KP781	All
1	0.01	0.01	0.00	0.01
2	0.08	0.07	0.00	0.14
3	0.30	0.22	0.02	0.54
4	0.05	0.04	0.04	0.13
5	0.01	0.00	0.16	0.17
All	0.44	0.33	0.22	1.00

In [45]:
```python
product_prob_given_fitness = pd.crosstab(index=df['Fitness'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
product_prob_given_fitness
```

Out[45]:

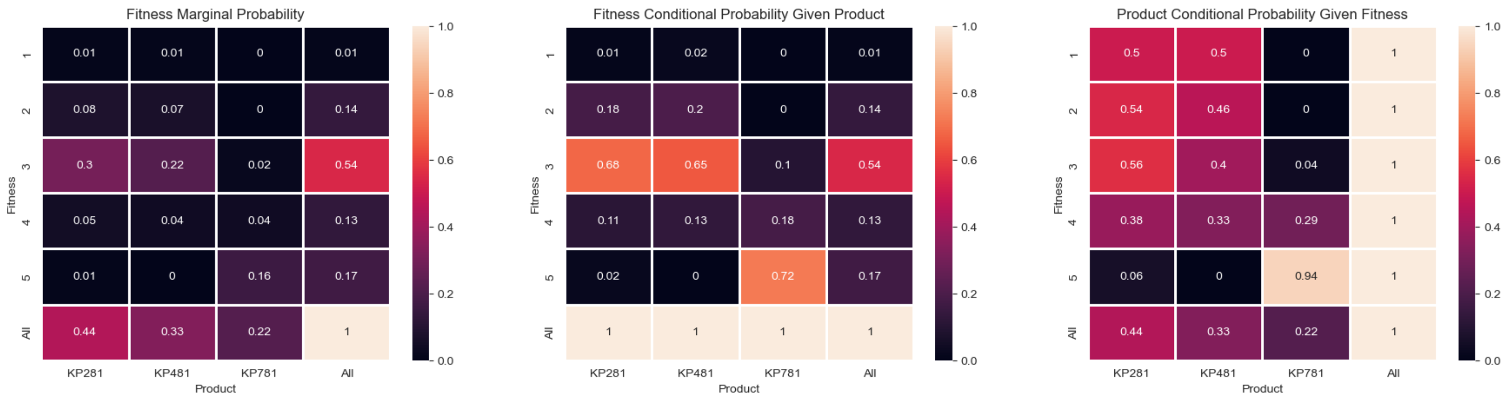
Product Fitness	KP281	KP481	KP781	All
1	0.50	0.50	0.00	1.0
2	0.54	0.46	0.00	1.0
3	0.56	0.40	0.04	1.0
4	0.38	0.33	0.29	1.0
5	0.06	0.00	0.94	1.0
All	0.44	0.33	0.22	1.0

In [46]:
```python
fitness_prob_given_product = pd.crosstab(index=df['Fitness'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)
fitness_prob_given_product
```

Out[46]:

Product Fitness	KP281	KP481	KP781	All
1	0.01	0.02	0.00	0.01
2	0.18	0.20	0.00	0.14
3	0.68	0.65	0.10	0.54
4	0.11	0.13	0.18	0.13
5	0.02	0.00	0.72	0.17
All	1.00	1.00	1.00	1.00

In [47]:
```python
fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Fitness Marginal Probability")
sns.heatmap(fitness_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Fitness Conditional Probability Given Product")
sns.heatmap(fitness_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Fitness")
sns.heatmap(product_prob_given_fitness, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```



- Customers who rate their treadmill usage as 3 are more likely to purchase KP281 treadmills.
- Buyers who indicate a usage rating of 5 are more inclined to choose KP781 treadmills.

Education

In [48]:
```
education_prob = round(pd.crosstab(index=df['EducationBin'],columns=df['Product'],margins=True, normalize=True),2)
education_prob
```

Out[48]:

Product EducationBin	KP281	KP481	KP781	All
0-12	0.01	0.01	0.00	0.02
12-14	0.18	0.14	0.01	0.33
14-16	0.24	0.18	0.08	0.50
16-18	0.01	0.01	0.11	0.13
18-20	0.00	0.00	0.01	0.01
20+	0.00	0.00	0.02	0.02
All	0.44	0.33	0.22	1.00

In [49]:
```
product_prob_given_education = pd.crosstab(index=df['EducationBin'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
product_prob_given_education
```

Out[49]:

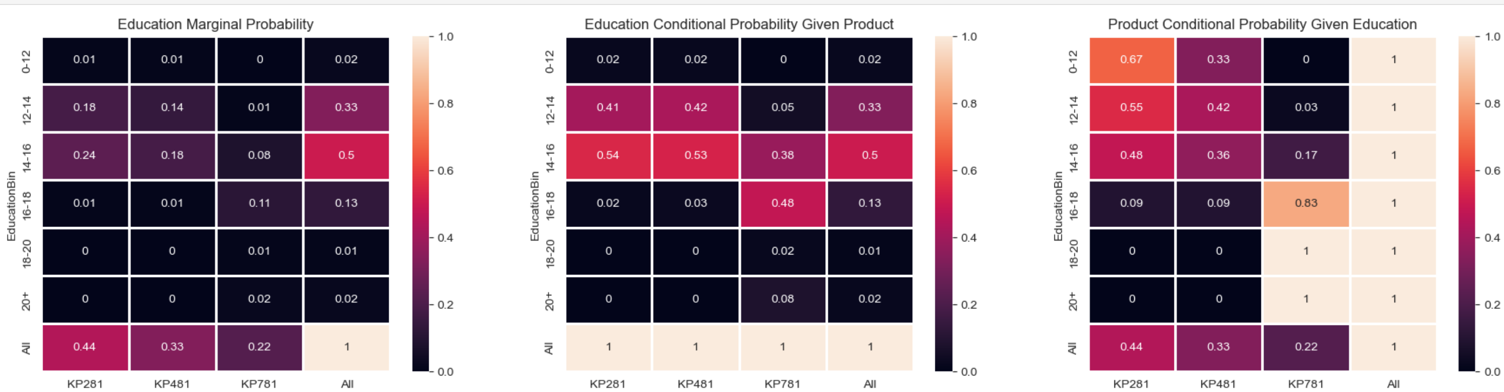
Product EducationBin	KP281	KP481	KP781	All
0-12	0.67	0.33	0.00	1.0
12-14	0.55	0.42	0.03	1.0
14-16	0.48	0.36	0.17	1.0
16-18	0.09	0.09	0.83	1.0
18-20	0.00	0.00	1.00	1.0
20+	0.00	0.00	1.00	1.0
All	0.44	0.33	0.22	1.0

In [50]:
```
education_prob_given_product = pd.crosstab(index=df['EducationBin'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)
education_prob_given_product
```

Out[50]:

Product EducationBin	KP281	KP481	KP781	All
0-12	0.02	0.02	0.00	0.02
12-14	0.41	0.42	0.05	0.33
14-16	0.54	0.53	0.38	0.50
16-18	0.02	0.03	0.48	0.13
18-20	0.00	0.00	0.02	0.01
20+	0.00	0.00	0.08	0.02
All	1.00	1.00	1.00	1.00

In [51]:
```
fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Education Marginal Probability")
sns.heatmap(education_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Education Conditional Probability Given Product")
sns.heatmap(education_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Education")
sns.heatmap(product_prob_given_education, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```



- Individuals with 0-12 years of education are more likely to purchase KP281 treadmills.
- KP781 treadmills are most commonly purchased by customers with 16+ years of education.

Age

In [52]:
```
age_prob = round(pd.crosstab(index=df['AgeBin'],columns=df['Product'],margins=True, normalize=True),2)
age_prob
```

Out[52]:

Product AgeBin	KP281	KP481	KP781	All
10-20	0.03	0.02	0.00	0.06
20-30	0.27	0.17	0.17	0.61
30-40	0.11	0.13	0.03	0.27
40-50	0.03	0.01	0.02	0.07
All	0.44	0.33	0.22	1.00

In [53]:
```
product_prob_given_age = pd.crosstab(index=df['AgeBin'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
product_prob_given_age
```

Out[53]:

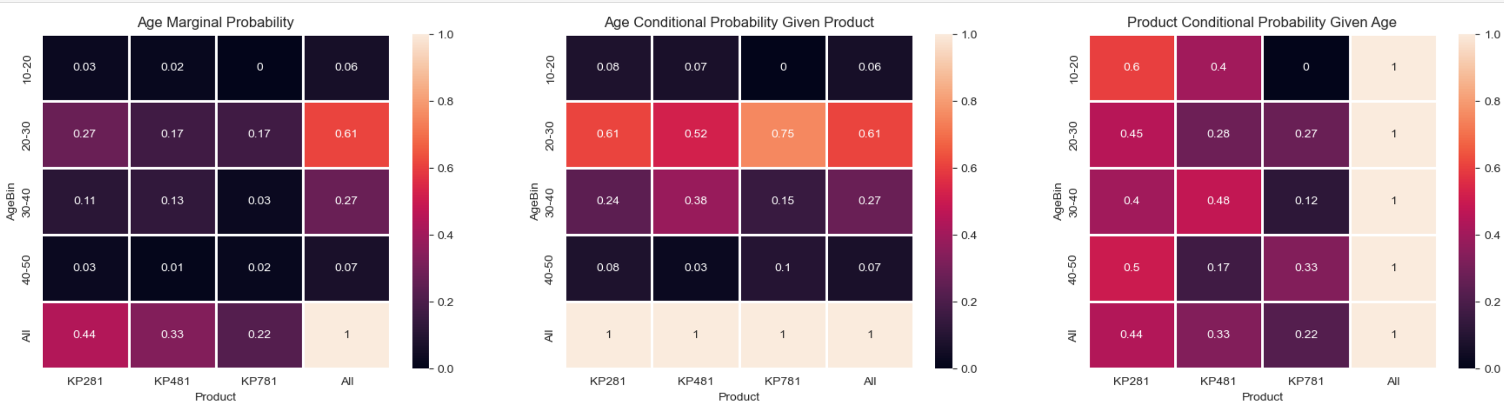
Product AgeBin	KP281	KP481	KP781	All
10-20	0.60	0.40	0.00	1.0
20-30	0.45	0.28	0.27	1.0
30-40	0.40	0.48	0.12	1.0
40-50	0.50	0.17	0.33	1.0
All	0.44	0.33	0.22	1.0

In [54]:
```
age_prob_given_product = pd.crosstab(index=df['AgeBin'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)
age_prob_given_product
```

Out[54]:

Product AgeBin	KP281	KP481	KP781	All
10-20	0.08	0.07	0.00	0.06
20-30	0.61	0.52	0.75	0.61
30-40	0.24	0.38	0.15	0.27
40-50	0.08	0.03	0.10	0.07
All	1.00	1.00	1.00	1.00

In [55]:
```
fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Age Marginal Probability")
sns.heatmap(age_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Age Conditional Probability Given Product")
sns.heatmap(age_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Age")
sns.heatmap(product_prob_given_age, linewidth = 1, annot = True,ax= ax[2],vmin=0)

plt.show()
```



- The age groups of 18-20 and 40-50 are more likely to purchase KP281 treadmills.
- Individuals in the age range of 30-40 years are most likely to be the buyers of KP481 treadmills.

Income

In [56]:
```
income_prob = round(pd.crosstab(index=df['IncomeBin'],columns=df['Product'],margins=True, normalize=True),2)
income_prob
```

Out[56]:

Product IncomeBin	KP281	KP481	KP781	All
20-40	0.13	0.05	0.00	0.18
40-60	0.28	0.24	0.06	0.59
60-80	0.03	0.04	0.06	0.13
80-100	0.00	0.00	0.09	0.09
100+	0.00	0.00	0.02	0.02
All	0.44	0.33	0.22	1.00

In [57]:
```
product_prob_given_income = pd.crosstab(index=df['IncomeBin'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
product_prob_given_income
```

Out[57]:

Product	KP281	KP481	KP781	All
IncomeBin				
20-40	0.72	0.28	0.00	1.0
40-60	0.48	0.42	0.10	1.0
60-80	0.26	0.30	0.43	1.0
80-100	0.00	0.00	1.00	1.0
100+	0.00	0.00	1.00	1.0
All	0.44	0.33	0.22	1.0

In [58]:

```
income_prob_given_product = pd.crosstab(index=df['IncomeBin'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)
income_prob_given_product
```

Out[58]:

Product	KP281	KP481	KP781	All
IncomeBin				
20-40	0.29	0.15	0.00	0.18
40-60	0.64	0.73	0.28	0.59
60-80	0.08	0.12	0.25	0.13
80-100	0.00	0.00	0.40	0.09
100+	0.00	0.00	0.08	0.02
All	1.00	1.00	1.00	1.00

In [59]:

```
fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Income Marginal Probability")
sns.heatmap(income_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Income Conditional Probability Given Product")
sns.heatmap(income_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Income")
sns.heatmap(product_prob_given_income, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```

- Customers with a low income bracket (20-40) are more likely to show interest in KP281 treadmills.
- Individuals in the middle income bracket (40-60) are most likely to purchase KP481 treadmills.
- KP781, the premium product, predominantly attracts buyers in the high income bracket (80+).

Miles

In [60]:

```
miles_prob = round(pd.crosstab(index=df['MilesBin'],columns=df['Product'],margins=True, normalize=True),2)
miles_prob
```

Out[60]:

Product	KP281	KP481	KP781	All
MilesBin				
<40	0.02	0.01	0.00	0.02
40-80	0.20	0.11	0.01	0.32
80-120	0.20	0.18	0.06	0.44
120-160	0.02	0.03	0.06	0.11
160-200	0.01	0.01	0.07	0.10
200+	0.00	0.01	0.01	0.01
All	0.45	0.34	0.20	1.00

In [61]:

```
product_prob_given_miles = pd.crosstab(index=df['MilesBin'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
product_prob_given_miles
```

Out[61]:

Product	KP281	KP481	KP781	All
MilesBin				
<40	0.75	0.25	0.00	1.0
40-80	0.62	0.36	0.02	1.0
80-120	0.46	0.40	0.14	1.0
120-160	0.21	0.26	0.53	1.0
160-200	0.12	0.12	0.76	1.0
200+	0.00	0.50	0.50	1.0
All	0.45	0.34	0.20	1.0

In [62]:

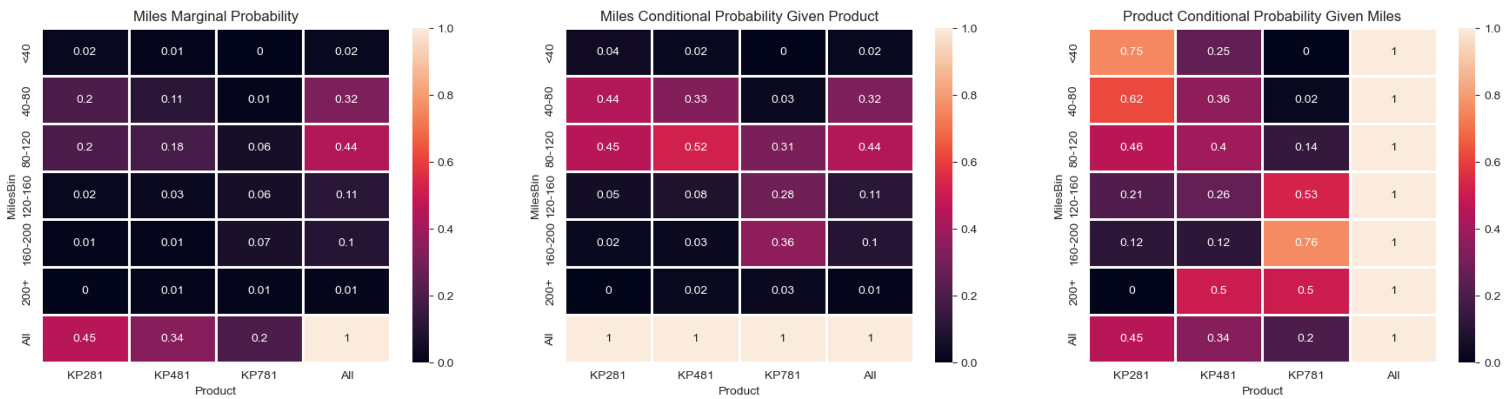
```
miles_prob_given_product = pd.crosstab(index=df['MilesBin'],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)
miles_prob_given_product
```

Out[62]:

Product	KP281	KP481	KP781	All
MilesBin				
<40	0.04	0.02	0.00	0.02
40-80	0.44	0.33	0.03	0.32
80-120	0.45	0.52	0.31	0.44
120-160	0.05	0.08	0.28	0.11
160-200	0.02	0.03	0.36	0.10
200+	0.00	0.02	0.03	0.01
All	1.00	1.00	1.00	1.00

In [63]:

```
fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Miles Marginal Probability")
sns.heatmap(miles_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Miles Conditional Probability Given Product")
sns.heatmap(miles_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Miles")
sns.heatmap(product_prob_given_miles, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```

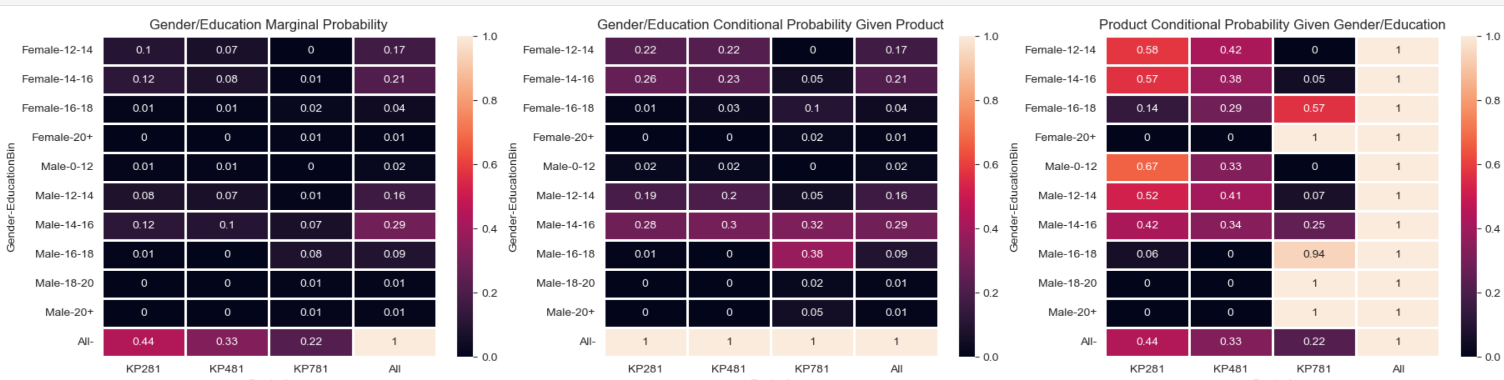


- Customers who run below 40 miles or between 40-80 miles are more likely to buy KP281 treadmills.
- Individuals who run between 120-200 miles are most likely to purchase KP781 treadmills.

Gender Education

In [64]:

```
gender_ed_prob = round(pd.crosstab(index=[df["Gender"],df["EducationBin"]],columns=df["Product"],margins=True,normalize=True),2)
product_prob_given_gender_ed = pd.crosstab(index=[df["Gender"],df["EducationBin"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
gender_ed_prob_given_product = pd.crosstab(index=[df["Gender"],df["EducationBin"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)

fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Gender/Education Marginal Probability")
sns.heatmap(gender_ed_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Gender/Education Conditional Probability Given Product")
sns.heatmap(gender_ed_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Gender/Education")
sns.heatmap(product_prob_given_gender_ed, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```

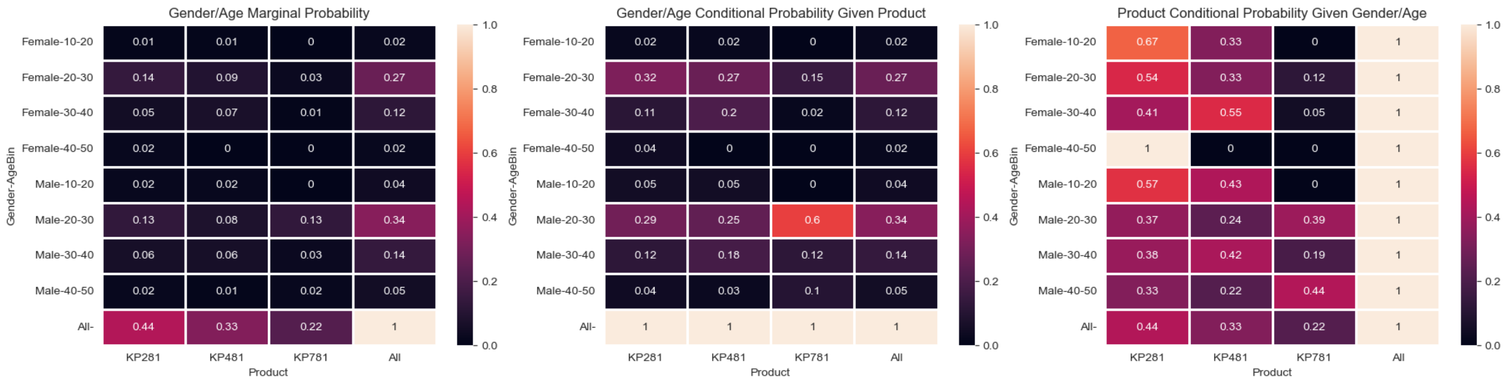


- Female buyers with 12-16 years of education are more likely to purchase KP281 treadmills, while females with 20+ years of education show interest in buying KP781.
- Male customers with 0-12 years of education are most likely to purchase KP281, while males above 16 years of education are more inclined to invest in the premium product, KP781.

Gender Age

In [65]:

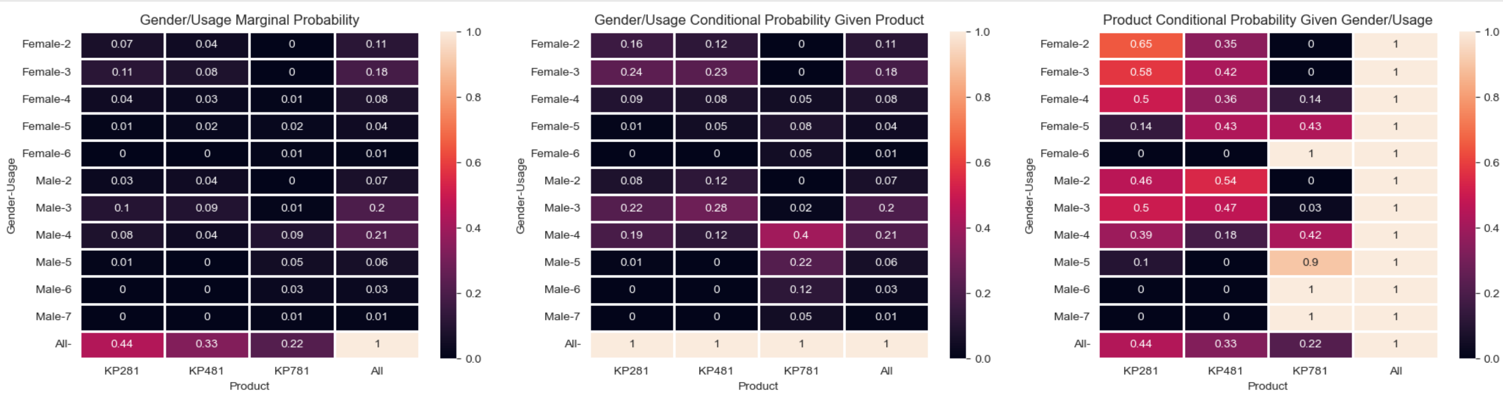
```
gender_age_prob = round(pd.crosstab(index=[df["Gender"],df["AgeBin"]],columns=df["Product"],margins=True,normalize=True),2)
product_prob_given_gender_age = pd.crosstab(index=[df["Gender"],df["AgeBin"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
gender_age_prob_given_product = pd.crosstab(index=[df["Gender"],df["AgeBin"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)

fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Gender/Age Marginal Probability")
sns.heatmap(gender_age_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Gender/Age Conditional Probability Given Product")
sns.heatmap(gender_age_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Gender/Age")
sns.heatmap(product_prob_given_gender_age, linewidth = 1, annot = True,ax= ax[2],vmin=0)
```



- The majority of KP281 purchases are made by females aged 40-50 years.
- Males aged 10-20 are least likely to purchase KP781.

Gender Usage

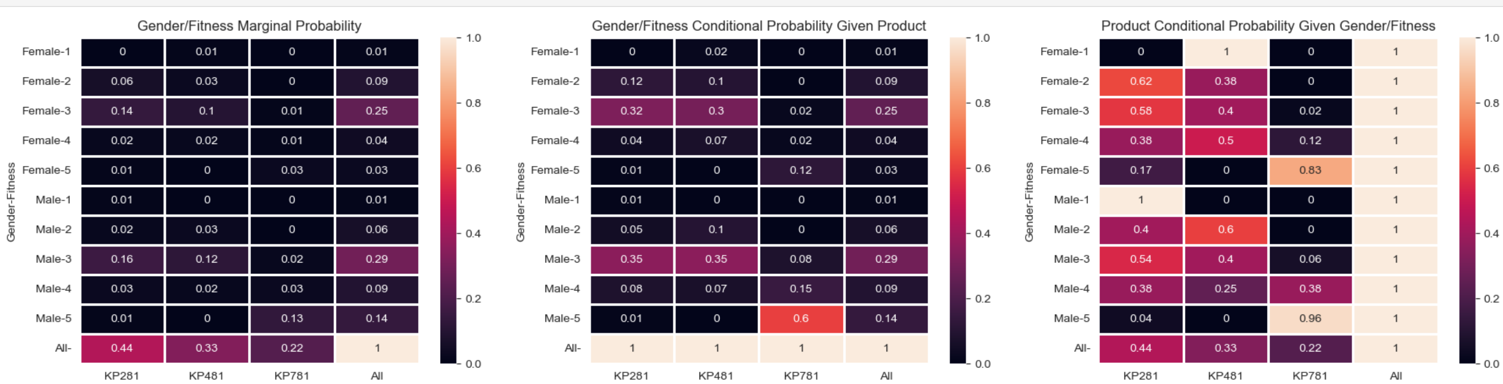
In [66]:
```
gender_usage_prob = round(pd.crosstab(index=[df["Gender"],df["Usage"]],columns=df["Product"],margins=True,normalize=True),2)
product_prob_given_gender_usage = pd.crosstab(index=[df["Gender"],df["Usage"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
gender_usage_prob_given_product = pd.crosstab(index=[df["Gender"],df["Usage"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)

fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Gender/Usage Marginal Probability")
sns.heatmap(gender_usage_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Gender/Usage Conditional Probability Given Product")
sns.heatmap(gender_usage_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Gender/Usage")
sns.heatmap(product_prob_given_gender_usage, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```



- Women with lower usage rates are more likely to choose KP281 as their preferred treadmill.
- Females who use the product more frequently are more inclined to purchase KP781.
- Males who rate their usage as 2 are predominantly the buyers of KP481.
- Males with higher usage rates are more likely to purchase KP781.

Gender Fitness

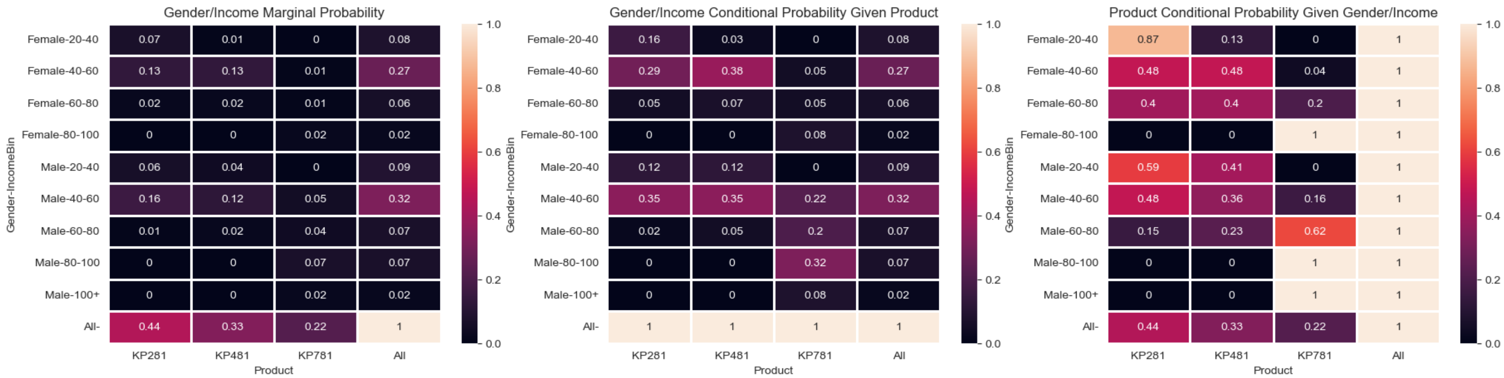
In [67]:
```
gender_fitness_prob = round(pd.crosstab(index=[df["Gender"],df["Fitness"]],columns=df["Product"],margins=True,normalize=True),2)
product_prob_given_gender_fitness = pd.crosstab(index=[df["Gender"],df["Fitness"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
gender_fitness_prob_given_product = pd.crosstab(index=[df["Gender"],df["Fitness"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)

fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Gender/Fitness Marginal Probability")
sns.heatmap(gender_fitness_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Gender/Fitness Conditional Probability Given Product")
sns.heatmap(gender_fitness_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Gender/Fitness")
sns.heatmap(product_prob_given_gender_fitness, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```



- KP281 treadmills are primarily favored by women with low fitness levels.
- Male buyers who rate themselves as 1 in fitness are most likely to purchase KP281.
- Customers who purchased KP781 treadmills tend to be the fittest males.

Gender Income

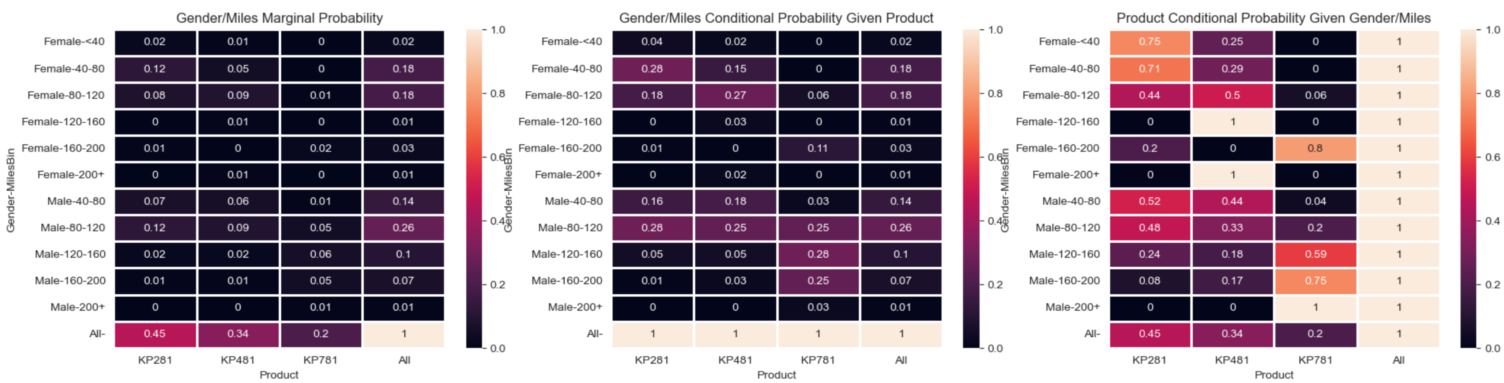
In [68]:
```
gender_income_prob = round(pd.crosstab(index=[df["Gender"],df["IncomeBin"]],columns=df["Product"],margins=True,normalize=True),2)
product_prob_given_gender_income = pd.crosstab(index=[df["Gender"],df["IncomeBin"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
gender_income_prob_given_product = pd.crosstab(index=[df["Gender"],df["IncomeBin"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)

fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Gender/Income Marginal Probability")
sns.heatmap(gender_income_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Gender/Income Conditional Probability Given Product")
sns.heatmap(gender_income_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Gender/Income")
sns.heatmap(product_prob_given_gender_income, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```



- Customers, both women and men, in the income band of 20-40 are most likely to choose KP281 treadmills.
- High-income earners, regardless of gender, are more inclined to purchase KP781.

Gender Miles

In [69]:
```
gender_miles_prob = round(pd.crosstab(index=[df["Gender"],df["MilesBin"]],columns=df["Product"],margins=True,normalize=True),2)
product_prob_given_gender_miles = pd.crosstab(index=[df["Gender"],df["MilesBin"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
gender_income_prob_given_miles = pd.crosstab(index=[df["Gender"],df["MilesBin"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)

fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Gender/Miles Marginal Probability")
sns.heatmap(gender_miles_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Gender/Miles Conditional Probability Given Product")
sns.heatmap(gender_income_prob_given_miles, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Gender/Miles")
sns.heatmap(product_prob_given_gender_miles, linewidth = 1, annot = True,ax= ax[2],vmin=0)


plt.show()
```

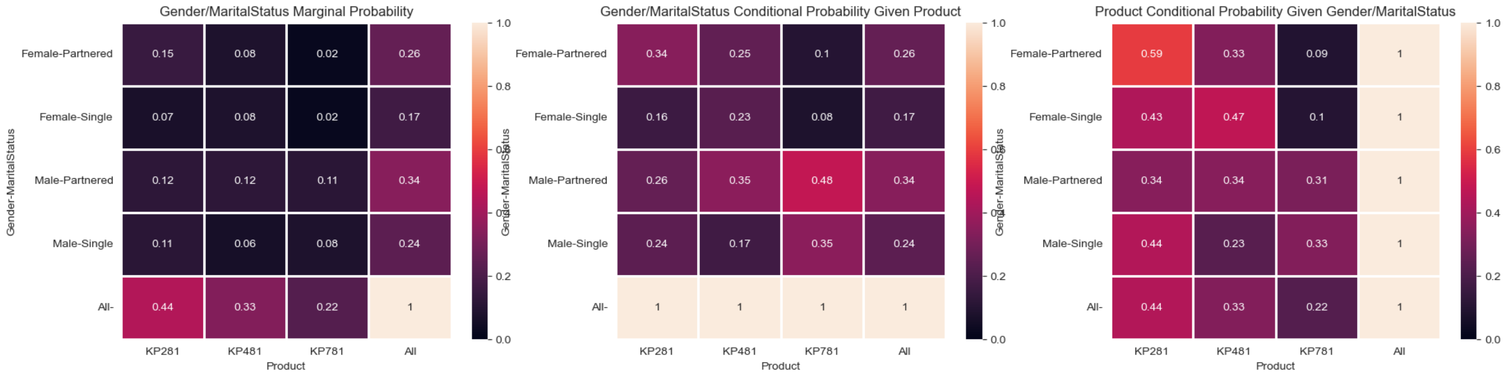


- Female buyers have the following purchase preferences:
  - Below 40 and 40-80 mile runners predominantly spend on KP281.
  - 120-160 and 200+ mile runners mostly spend on KP481.
  - 160-200 mile runners tend to spend on KP781.
- Male buyers who run between 40-120 miles are most likely to purchase KP281, whereas individuals running 160+ miles are more inclined to buy KP781.

Gender Marital

In [70]:
```
gender_marital_prob = round(pd.crosstab(index=[df["Gender"],df["MaritalStatus"]],columns=df["Product"],margins=True,normalize=True),2)
product_prob_given_gender_marital = pd.crosstab(index=[df["Gender"],df["MaritalStatus"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=1).round(2)
gender_marital_prob_given_product = pd.crosstab(index=[df["Gender"],df["MaritalStatus"]],columns=df['Product'],margins=True).apply(lambda x: x/x.iloc[-1],axis=0).round(2)

fig,ax = plt.subplots(1,3,figsize=(22,5))

ax[0].set_title("Gender/MaritalStatus Marginal Probability")
sns.heatmap(gender_marital_prob, linewidth = 1, annot = True,ax= ax[0],vmin=0)

ax[1].set_title("Gender/MaritalStatus Conditional Probability Given Product")
sns.heatmap(gender_marital_prob_given_product, linewidth = 1, annot = True,ax= ax[1],vmin=0)

ax[2].set_title("Product Conditional Probability Given Gender/MaritalStatus")
sns.heatmap(product_prob_given_gender_marital, linewidth = 1, annot = True,ax= ax[2],vmin=0)
```

```
plt.show()
```



- Married females are the primary buyers of KP281 treadmills.
- Single females are more likely to purchase KP481 treadmills.
- Married males show no clear preference between KP281 and KP481.
- Single males are mostly likely to spend on KP281 treadmills.

4. Insights

4.1 Buisness Insights

Insights commented under each plot

4.2 Recommendations

- Focus on attracting more men to buy KP781 treadmills.
- Target women customers for KP281 treadmills, as they prefer them more.
- Promote KP781 treadmills to women to increase their interest.
- Recommend KP281 to customers who plan to use the treadmill 3 times a week and KP781 to those who plan to use it 5 or more times.
- Highlight the benefits of KP281 for customers with less education and the premium features of KP781 for those with more education.
- Customize marketing for customers aged 18-20 and 40-50 to promote KP281, and target customers aged 30-40 for KP481.
- Advertise KP281 to customers with low income, KP481 to those with middle income, and KP781 to customers with high income.
- Emphasize the fitness features of KP281 to women with low fitness levels, and promote the advanced fitness capabilities of KP781.
- Market KP481 to couples, highlighting the benefits of shared fitness goals.
- Encourage KP281 customers to use the treadmill more frequently through incentives and personalized training plans.

4.3 Customer Profiling

Customer Profiling for Each Product:

KP281 Treadmills:

- Predominantly favored by women, especially those aged 40-50.
- Women with lower fitness levels and 12-16 years of education show a higher preference.
- Customers who rate their treadmill usage as 3, run below 40 miles per week, and have a lower income bracket (20-40) are more likely to purchase KP281.
- Target audience: Women in the age range of 40-50, with lower fitness levels, moderate education, and lower usage expectations.

KP481 Treadmills:

- Attracts a mix of male and female customers, but males aged 10-20 are least likely to purchase.
- Individuals in the age range of 30-40, with middle income (40-60), and planning to run 120-160 or 200+ miles per week tend to choose KP481.
- Target audience: Both men and women in the age range of 30-40, with middle income, and higher running expectations.

KP781 Treadmills:

- More likely to be purchased by males, especially those with higher education (16+ years) and income (80+).
- Fittest males who rate themselves as 1 in fitness and plan to run 160+ miles per week show a preference for KP781.
- Target audience: Educated, high-income males with a strong fitness focus and high running expectations.

Overall, KP281 targets women with lower fitness levels and usage expectations, KP481 appeals to individuals with moderate income and running goals, while KP781 targets high-income, educated, and fitness-oriented males with higher running expectations.