BUSINESS CASE STUDY AEROFIT

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.

Perform descriptive analytics to create a customer profile for each AeroFit treadmill product by developing appropriate tables and charts. 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.

Dataset

The company collected the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months. The dataset has the following features:

- **Product Purchased**: KP281, KP481, or KP781
- **Age**: In years
- **Gender**: Male/Female
- **Education**: In years
- **MaritalStatus**: Single or partnered
- **Usage**: The average number of times the customer plans to use the treadmill each week.
- **Income**: Annual income (in $)
- **Fitness**: Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 is the excellent shape.
- **Miles**: The average number of miles the customer expects to walk/run each week

In [103…
```
#importing packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [66]:
```
data=pd.read_csv(r"C:\Users\varun\Desktop\projects\aerofit_treadmill.csv")
```

In [67]:
```
data.info()
```

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

In [68]:
```
data["Product"].value_counts()
```

Out[68]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

In [69]:
```
data.describe(include='all')
```

Out[69]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

Insight:

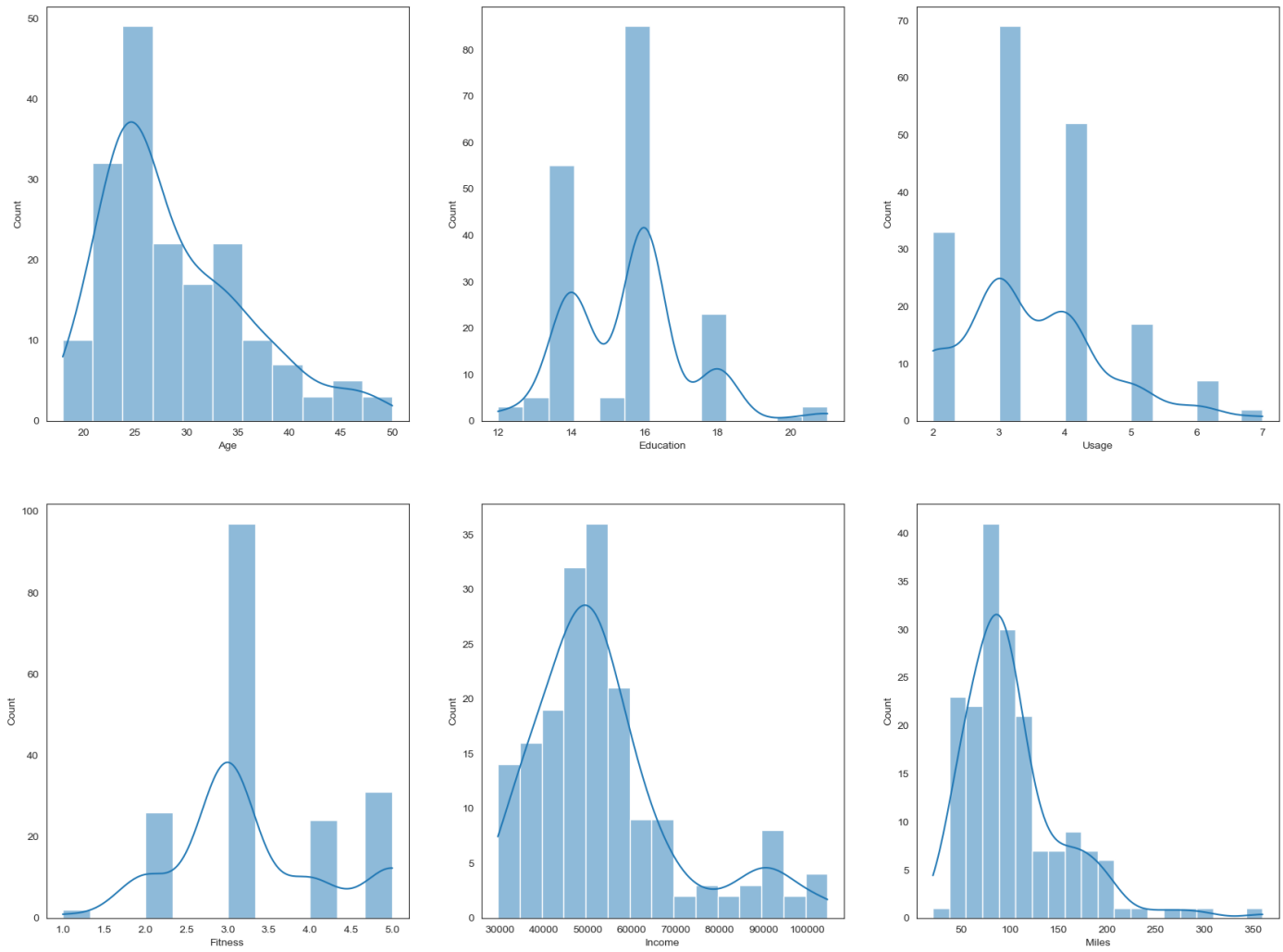
Here's a brief summary of your data:

- The dataset contains 180 entries (rows) and 9 attributes (columns).
- There are no missing values.
- The age of individuals ranges from 18 to 50, with an average of 28.79. 75% of individuals are 33 or younger.
- The dataset includes more males (104) than females.
- Most individuals have 16 years of education, with 75% having 16 years or less.
- The most frequent product is KP281, appearing 80 times.
- The majority of individuals are partnered (107 out of 180).
- The columns 'Income' and 'Miles' may contain outliers due to high standard deviation.
- There are 3 unique products "KP281" , "KP481" , "KP781" .

Univariate Analysis and Bivariate Analysis

In [85]:
```
fig, axis = plt.subplots(2,3 , figsize=(15,10))
fig.subplots_adjust(top=1.3,right=1.2)

sns.histplot(data=data,x="Age",kde=True,ax=axis[0,0])
sns.histplot(data=data,x="Education",kde=True,ax=axis[0,1])
sns.histplot(data=data,x="Usage",kde=True, ax=axis[0,2])
sns.histplot(data=data,x="Fitness",kde=True,ax=axis[1,0])
sns.histplot(data=data,x="Income",kde=True,ax=axis[1,1])
sns.histplot(data=data,x="Miles",kde=True, ax=axis[1,2])
plt.show()
```

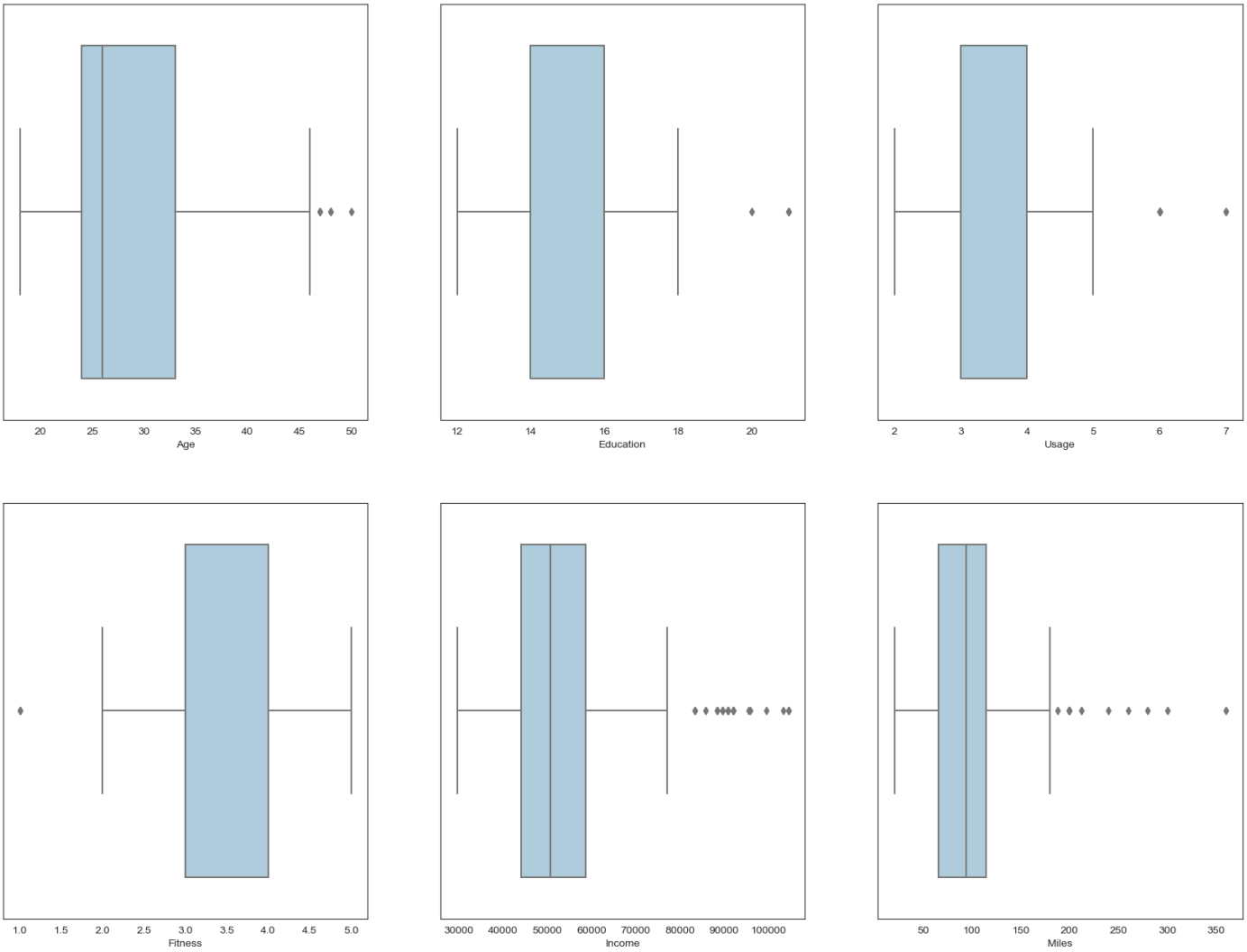


In [102…

```
fig, axis = plt.subplots(2,3 , figsize=(15,10))

fig.subplots_adjust(top=1.3,right=1.2)
sns.boxplot(data=data,x="Age",palette='Paired',ax=axis[0,0])
sns.boxplot(data=data,x="Education",palette='Paired',ax=axis[0,1])
sns.boxplot(data=data,x="Usage",palette='Paired', ax=axis[0,2])
sns.boxplot(data=data,x="Fitness",palette='Paired',ax=axis[1,0])
sns.boxplot(data=data,x="Income",palette='Paired',ax=axis[1,1])
sns.boxplot(data=data,x="Miles",palette='Paired', ax=axis[1,2])

plt.show()
```

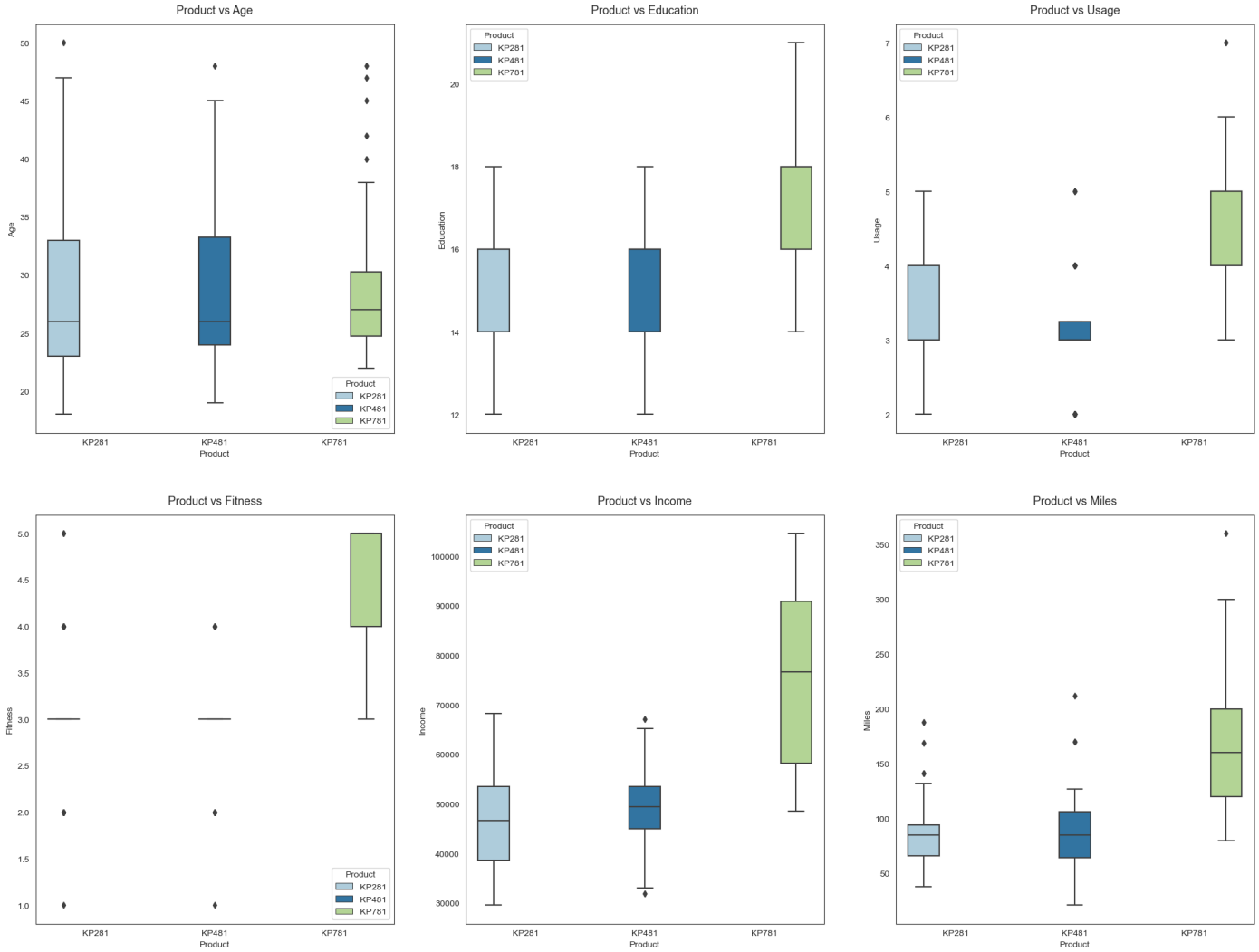


Insight:

- "Income" and "Miles" have more ouliers than other parameters.

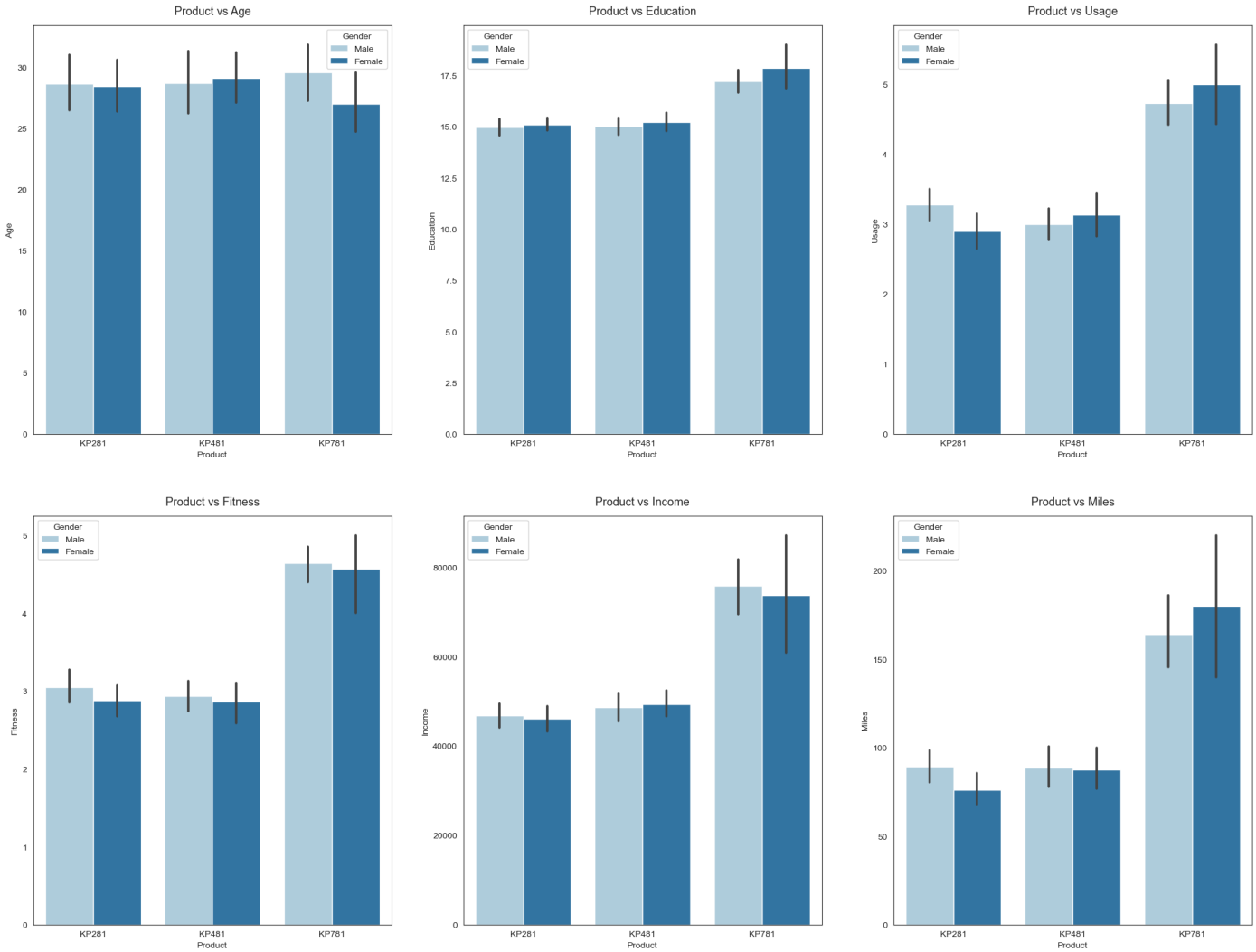
In [73]:
```
var= ['Age','Education','Usage','Fitness','Income','Miles']
sns.set_style("white")

fig,axs=plt.subplots(2,3,figsize=(18,12))
fig.subplots_adjust(top=1.3,right=1.2)
count=0
for i in range(2):
 for j in range(3):
  sns.boxplot(data=data,x='Product',y=var[count],ax=axs[i,j],hue='Product',palette="Paired")
  axs[i,j].set_title(f"Product vs {var[count]}",pad=12,fontsize=13)
  count +=1
```



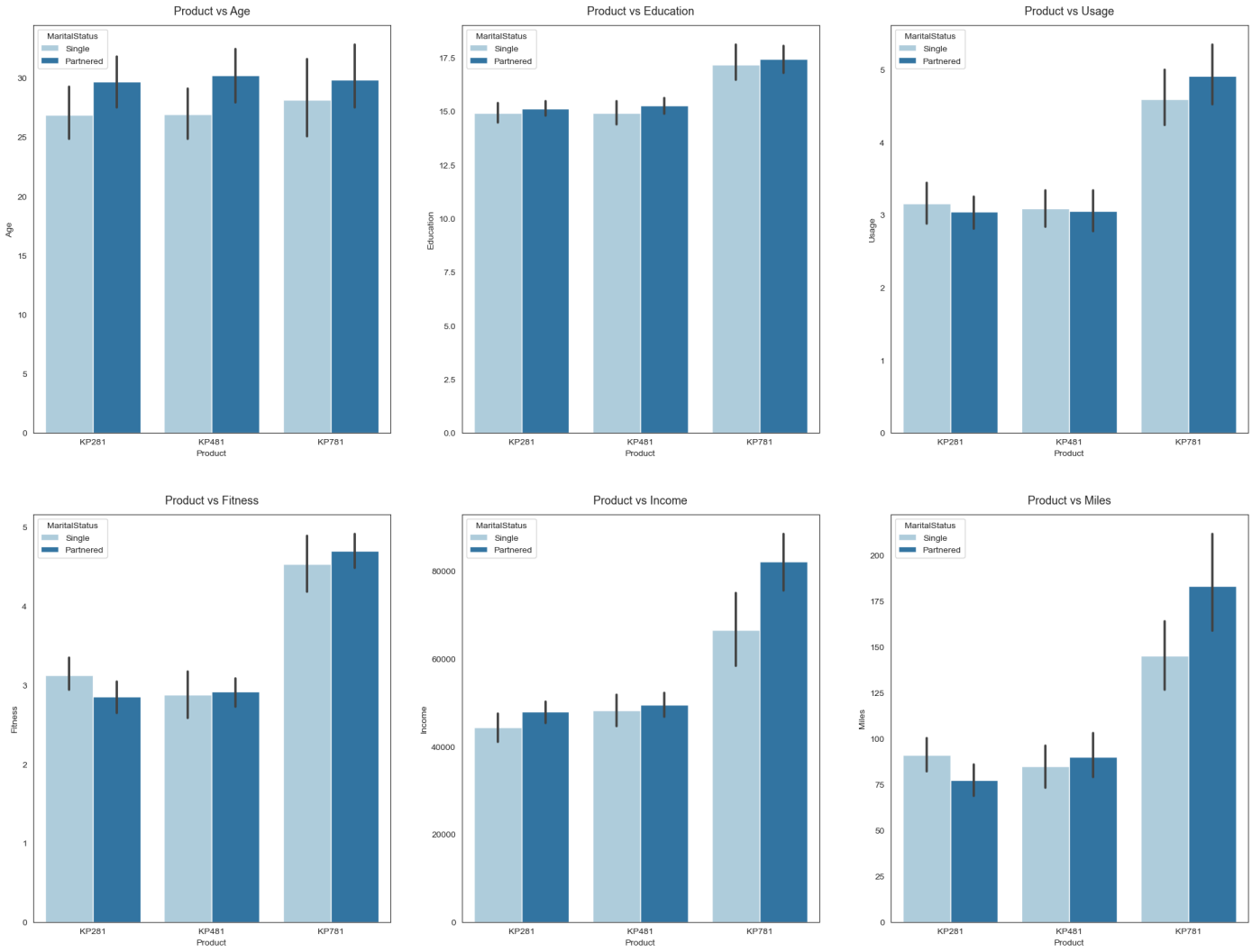
In [87]:
```
ar= ['Age','Education','Usage','Fitness','Income','Miles']
sns.set_style("white")

fig,axs=plt.subplots(2,3,figsize=(18,12))
fig.subplots_adjust(top=1.3,right=1.2)
count=0
for i in range(2):
 for j in range(3):
  sns.barplot(data=data,x='Product',y=var[count],ax=axs[i,j],hue='Gender',palette="Paired")
  axs[i,j].set_title(f"Product vs {var[count]}",pad=12,fontsize=13)
  count +=1
```



In [75]:
```
ar= ['Age','Education','Usage','Fitness','Income','Miles']
sns.set_style("white")

fig,axs=plt.subplots(2,3,figsize=(18,12))
fig.subplots_adjust(top=1.3,right=1.2)
count=0
for i in range(2):
 for j in range(3):
  sns.barplot(data=data,x='Product',y=var[count],ax=axs[i,j],hue='MaritalStatus',palette="Paired")
  axs[i,j].set_title(f"Product vs {var[count]}",pad=12,fontsize=13)
  count +=1
```



Insight

Product vs Gender

- Equall number of Males and Females have purchased KP281 product and almost same for the product KP481.
- Most of the male customers have purchased the KP781

Product vs MaritalStatus

- Customers who is Partnered , is more likely to purchase the product and it is true for all the products

Product vs Age

- Customers purchasing products KP281 & KP481 are having same age median value.
- Customers whose age lies between 25-30, are more likely to buy KP781 product(Partnered).

Product vs Education

- Customers whose education is greater than 16, have more chances to purchase the kp781 product.
- While the customers with education less than 16 have equal chances of purchasing kp281 or kp481.

Product vs Usage

- Customers who are planning to use the treadmill greater than 4 times a week, are more likely to purchase the kp781 product
- While the other customers are likely to purchasing kp281 or kp481.

Product vs Fitness

- The more the customer is fit (fitness >= 3), higher the chances of the customer to purchase the kp781 product

Product vs Income

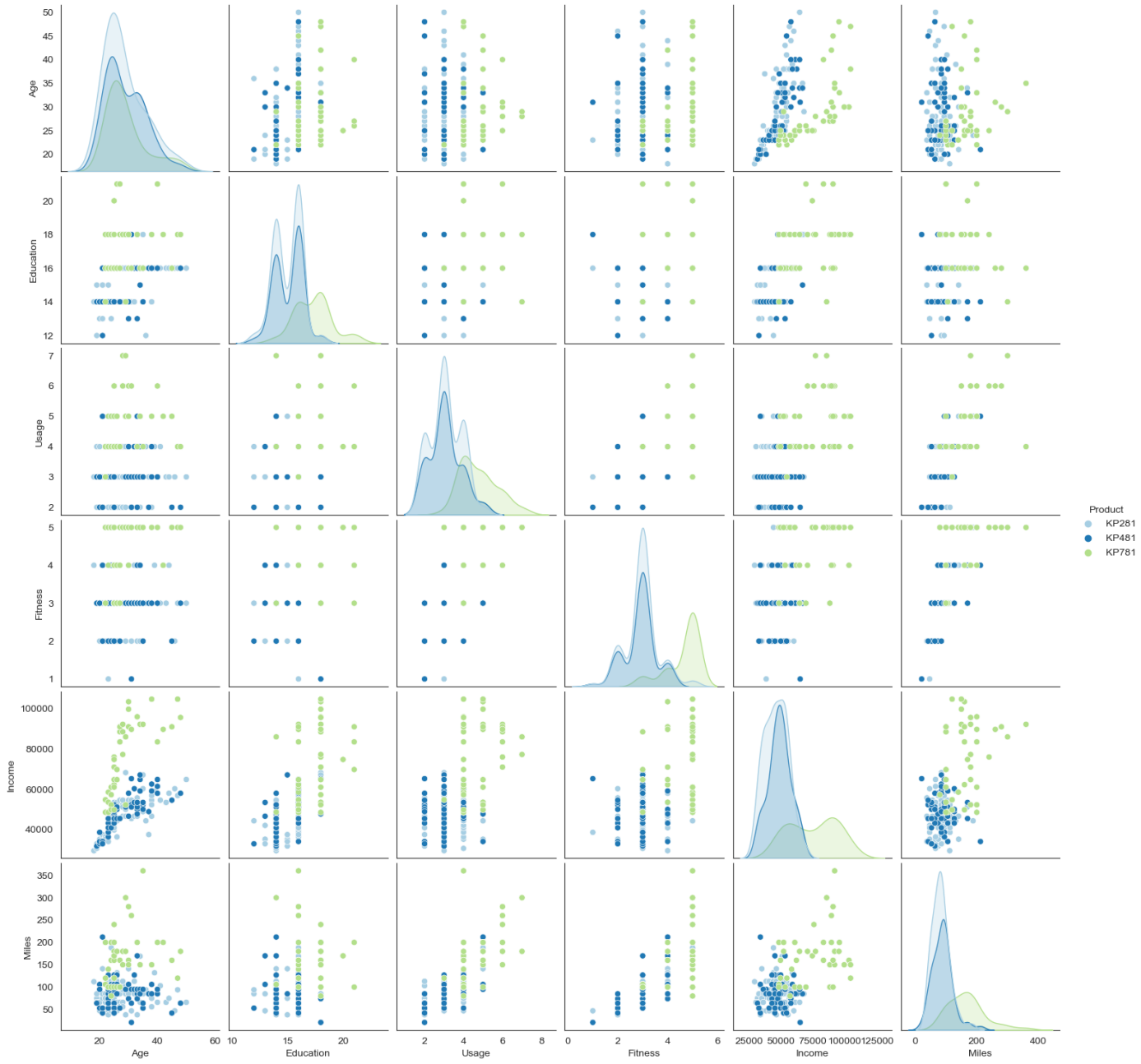
- Higher the income of the customer (income >= 60000), higher the chances of the customer to purchase the kp781 product.

Product vs Miles

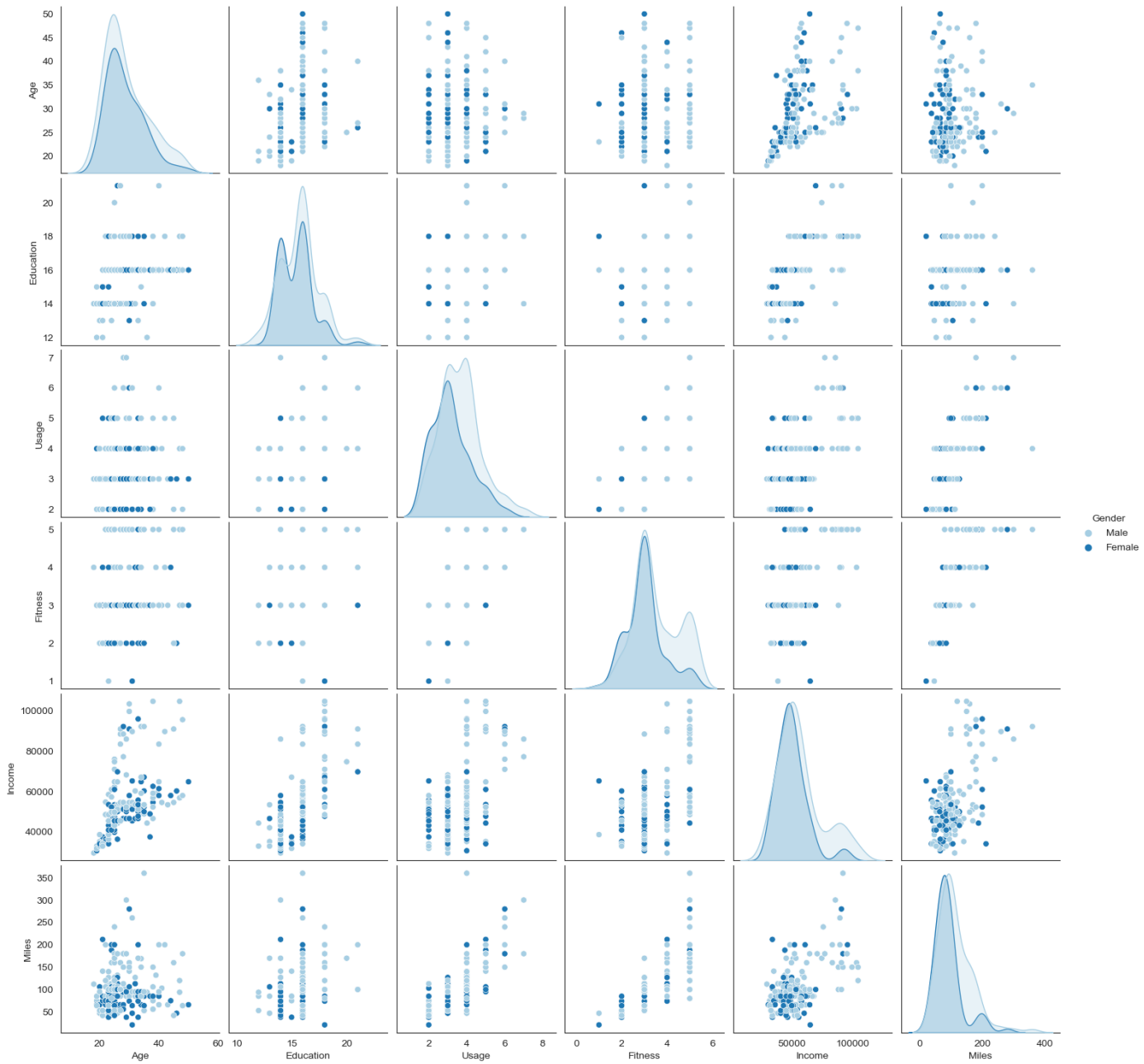
- if the customer expects to walk/run greater than 120 miles per week, it is more likely that the customer will buy kp781 product(Partnered prople are likely to buy most)

In [76]:
```
sns.set_style("white")
sns.pairplot(data=data,hue='Product',palette='Paired')
```

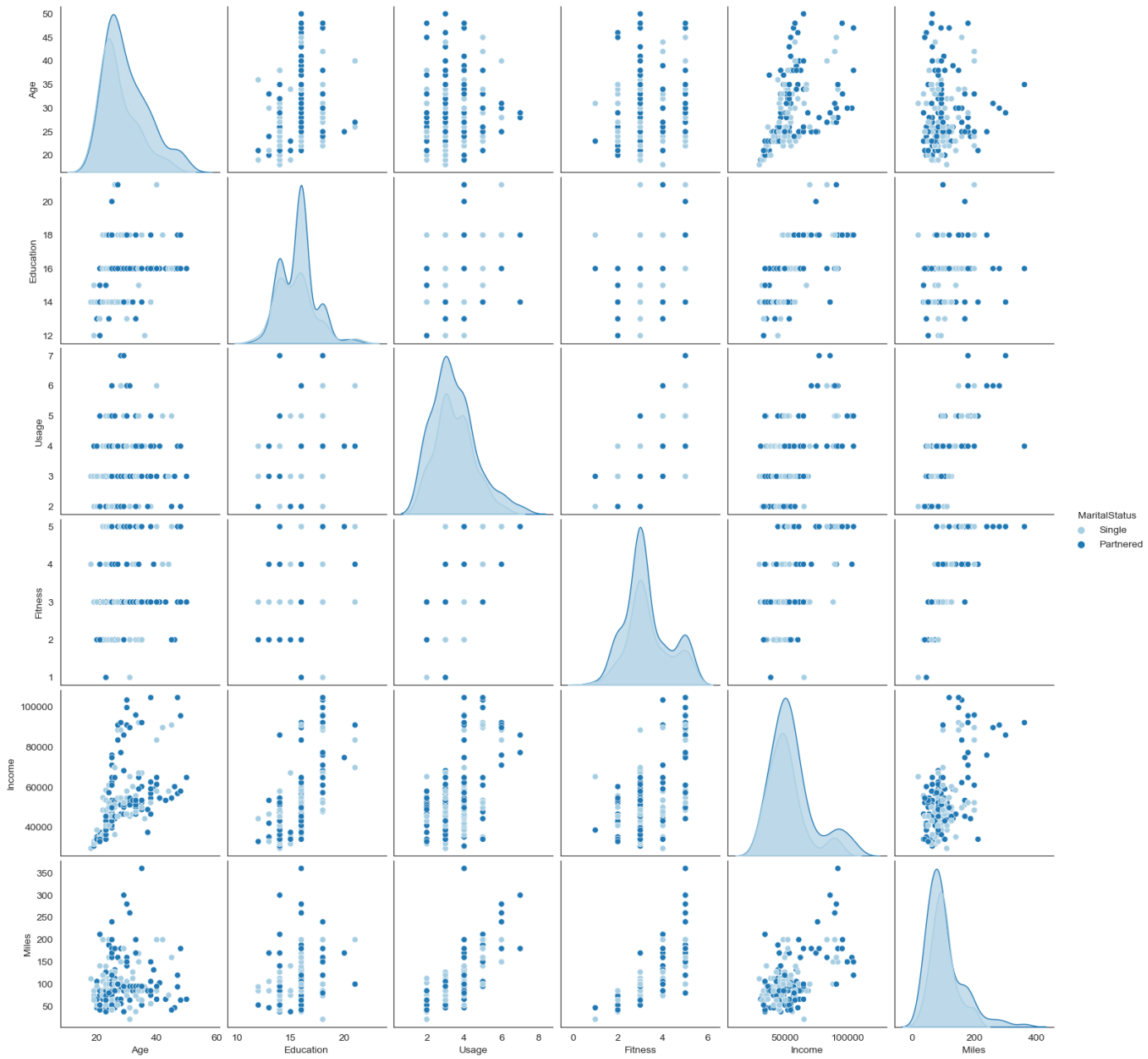
Out[76]:
```
<seaborn.axisgrid.PairGrid at 0x29f04ebf520>
```



In [77]:
```
sns.set_style("white")
sns.pairplot(data,hue='Gender',palette="Paired")
plt.show()
```

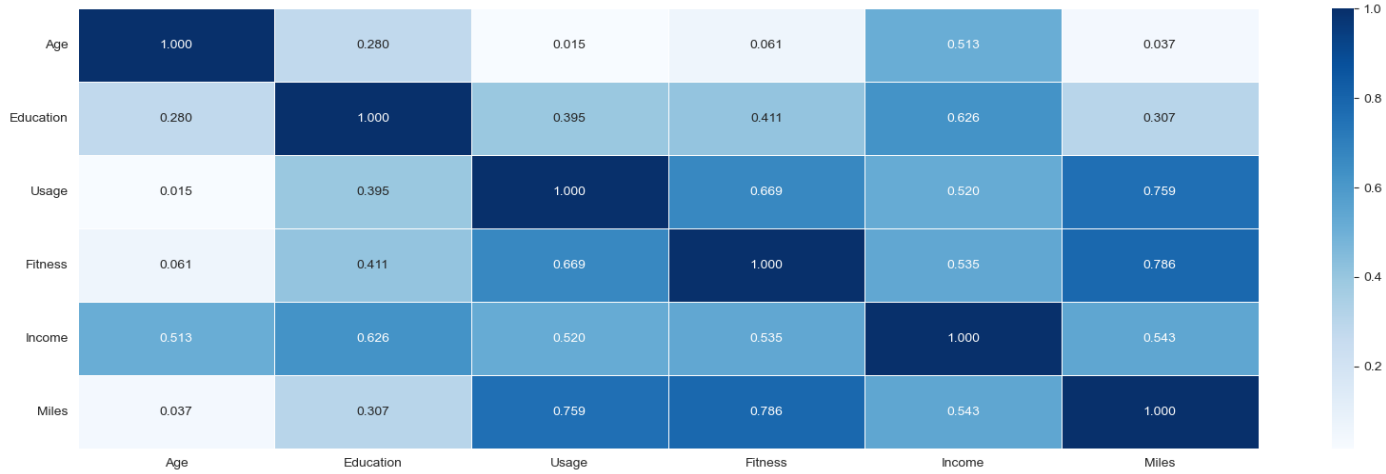


In [78]:
```
sns.set_style('white')
sns.pairplot(data,hue='MaritalStatus',palette="Paired")
plt.show()
```



In [92]:
```
#Correlation HeatMap
plt.figure(figsize=(20,6))
ax = sns.heatmap(data.corr(),annot=True,fmt='.3f',linewidths=.5,cmap='Blues')
plt.yticks(rotation=0)
plt.show()
```

```
C:\Users\varun\AppData\Local\Temp\ipykernel_20632\950116467.py:3: FutureWarning: The default value of numeric_only in D
ataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the va
lue of numeric_only to silence this warning.
  ax = sns.heatmap(data.corr(),annot=True,fmt='.3f',linewidths=.5,cmap='Blues')
```



Insight

- Correlation between Age and Miles is 0.036
- Correlation between Education and Income is 0.62

- Correlation between Usage and Fitness is 0.66
- Correlation between Fitness and Age is 0.06
- Correlation between Income and Usage is 0.51
- Correlation between Miles and Fitnessis 0.786

In [115…
```
data_category = data.copy()
data_category['age_category'] = pd.cut(data_category['Age'], bins=[0,16,30,45,60], labels=['Teen','Adult','Middle Aged'
print(data_category.head())
```

```
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
0   KP281   18    Male         14        Single      3        4   29562
1   KP281   19    Male         15        Single      2        3   31836
2   KP281   19  Female         14     Partnered      4        3   30699
3   KP281   19    Male         12        Single      3        3   32973
4   KP281   20    Male         13     Partnered      4        2   35247

   Miles age_category
0    112        Adult
1     75        Adult
2     66        Adult
3     85        Adult
4     47        Adult
```

In [116…
```
data_category.age_category.value_counts()
```

Out[116]:
```
Adult          120
Middle Aged     54
Elder            6
Teen             0
Name: age_category, dtype: int64
```

In [117…
```
data_category.loc[data_category.Product=='KP281']["age_category"].value_counts()
```

Out[117]:
```
Adult          55
Middle Aged    22
Elder           3
Teen            0
Name: age_category, dtype: int64
```

In [118…
```
data_category.loc[data_category.Product=='KP481']["age_category"].value_counts()
```

Out[118]:
```
Adult          35
Middle Aged    24
Elder           1
Teen            0
Name: age_category, dtype: int64
```

In [119…
```
data_category.loc[data_category.Product=='KP781']["age_category"].value_counts()
```

Out[119]:
```
Adult          30
Middle Aged     8
Elder           2
Teen            0
Name: age_category, dtype: int64
```

Missing Value & Outlier Detection

In [110…
```
data.isna().sum()
```

Out[110]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

No null values

In [81]:
```
data.duplicated().sum()
```

Out[81]:
```
0
```

No duplicates found

In [ ]:
```
q_75, q_25 = np.percentile(data['Miles'], [75 ,25])
miles_iqr = q_75 - q_25
print("Inter Quartile Range for Miles is", miles_iqr)
```

In [ ]:
```
q_75, q_25 = np.percentile(data['Usage'], [75 ,25])
usage_iqr = q_75 - q_25
print("Inter Quartile Range for Usage is", usage_iqr)
```

In [ ]:
```
q_75, q_25 = np.percentile(data['Income'], [75 ,25])
income_iqr = q_75 - q_25
print("Inter Quartile Range for Incomeis", income_iqr)
```

In [ ]:
```
q_75, q_25 = np.percentile(data['Education'], [75 ,25])
edu_iqr = q_75 - q_25
print("Inter Quartile Range for Education is", edu_iqr )
```

In [ ]:
```
q_75, q_25 = np.percentile(data['Fitness'], [75 ,25])
fitness_iqr = q_75 - q_25
print("Inter Quartile Range for Fitness is", fitness_iqr )
```

Business Insights based on Non-Graphical and Visual Analysis

In [121…
```
round(pd.crosstab(index=data_category.Product,columns=data_category.age_category,normalize=True,margins=True)*100,2)
```

Out[121]:

age_category	Adult	Middle Aged	Elder	All
Product				
KP281	30.56	12.22	1.67	44.44
KP481	19.44	13.33	0.56	33.33
KP781	16.67	4.44	1.11	22.22
All	66.67	30.00	3.33	100.00

In [ ]:
```
round(data['Product'].value_counts(normalize=True)*100,2)
```

Probability of buying **KP281, KP481 & KP781** are **44%, 33% & 22%** respectively

In [ ]:
```
round(data['MaritalStatus'].value_counts(normalize=True)*100,2)
```

Probability of **Partnered** and **Single** Customers are **59% and 41%** respectively

In [ ]:
```
round(data['Gender'].value_counts(normalize=True)*100,2)
```

Probability of **Male** and **Female** Customers are **58 and 42%** respectively

In [95]:
```
round(pd.crosstab(columns=data["Fitness"],index=data["Product"],normalize=True)*100,2)
```

Out[95]:

Fitness	1	2	3	4	5
Product					
KP281	0.56	7.78	30.00	5.00	1.11
KP481	0.56	6.67	21.67	4.44	0.00
KP781	0.00	0.00	2.22	3.89	16.11

Probability of people with fitness 3 will mostl by **KP281** and **KP481**.People with higher fitness level chooses **KP781**

In [91]:
```
np.round((pd.crosstab([data.Product],data.Gender,margins=True,normalize="columns"))*100,2)
```

Out[91]:

Gender	Female	Male	All
Product			
KP281	52.63	38.46	44.44
KP481	38.16	29.81	33.33
KP781	9.21	31.73	22.22

Insight

Product KP281

The probability of a female customer buying this product is **52.63%**, which is higher than the probability of a male customer **(38.46%)**. Therefore, this product is more recommended for female customers.

Product KP481

The probability of a female customer buying this product is **38.15%**, which is significantly higher than the probability of a male customer **(29.80%)**. This product is specifically recommended for female customers who are intermediate users.

Product KP781

The probability of a male customer buying this product is **31.73%**, which is significantly higher than the probability of a female customer **(9.21%)**. Therefore, this product is more recommended for male customers.

In [ ]: