In [5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:
```
df = pd.read_csv("aerofit_treadmill.csv")
df
```

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
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

180 rows × 9 columns

In [7]:
```
df.describe(include="all")
```

Out[7]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	In
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.00
unique	3	NaN	2	NaN	2	NaN	NaN	
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	
freq	80	NaN	104	NaN	107	NaN	NaN	
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.57
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.68
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.00
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.75
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.50
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.00
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.00

In [8]:
```
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
```

Observations:

```
  1. There are 180 Rows and 9 Columns.
  2. There are no missing values in data.
  3. Minimum and Maximum age of the person is 18 and 50, mean 28.79
and 75% of the persons have age less than or equal to 33.
  4. Out of 180 data of gender, 104 persons are Male and rest are
Female.
  5. Most of the people are having 16 years of education i.e 75% of
persons are having education <= 16 years.
  6. Product name KP281 is the most frequent product with values
80.
  7. Frequency of Marital Status "PArtnered" is 107 out of 180.
  8. There must be outliers in column Income and Miles as the
standard deviation of these data are very high.
```

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

Out[9]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

1. There are 3 unique products "KP281", "KP481", "KP781".

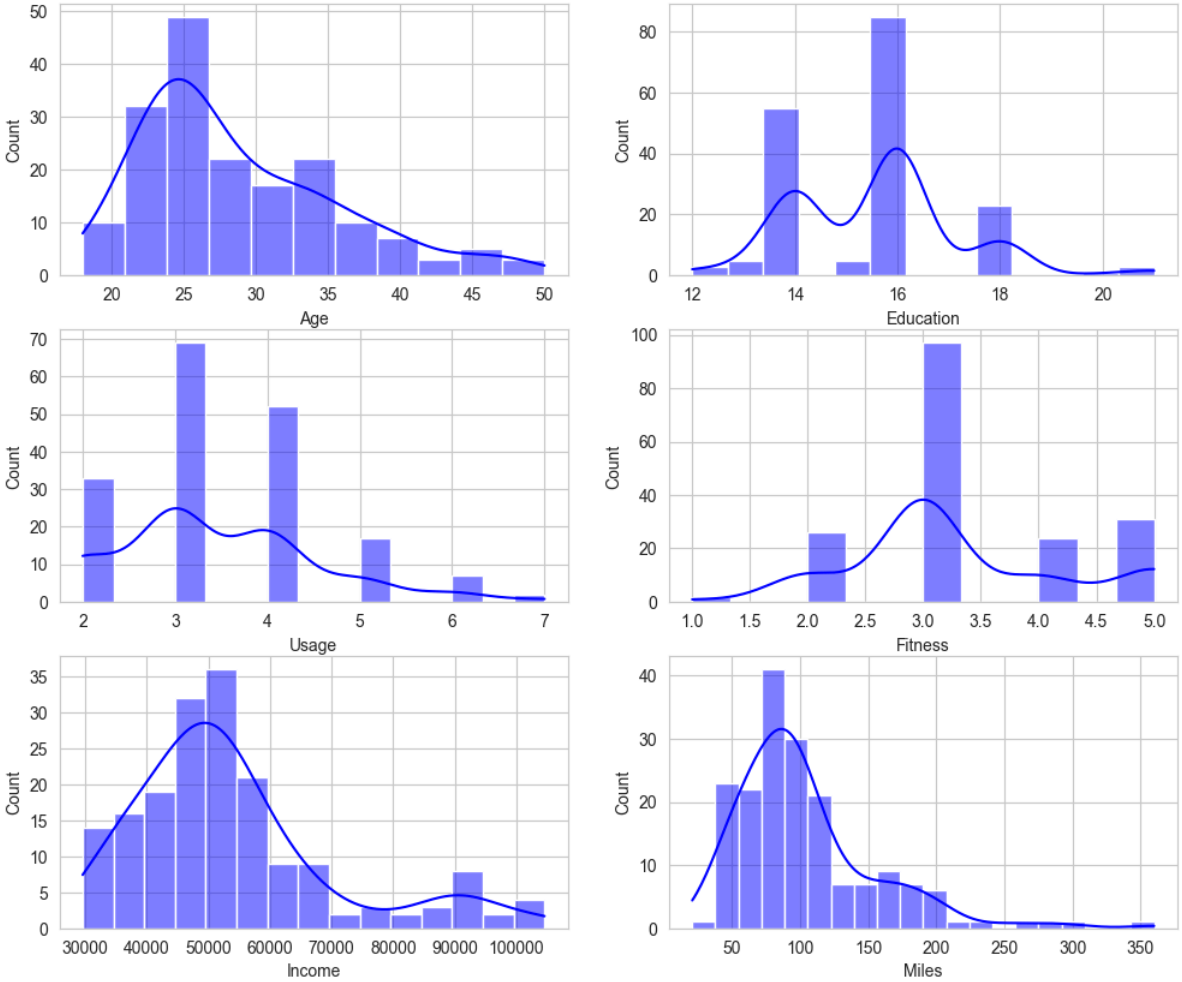
Univariate Analysis

Understanding the distribution of the data for the quantitative attributes.

1. Age.
2. Education.
3. Usage.
4. Fitness.
5. Income.
6. Miles.

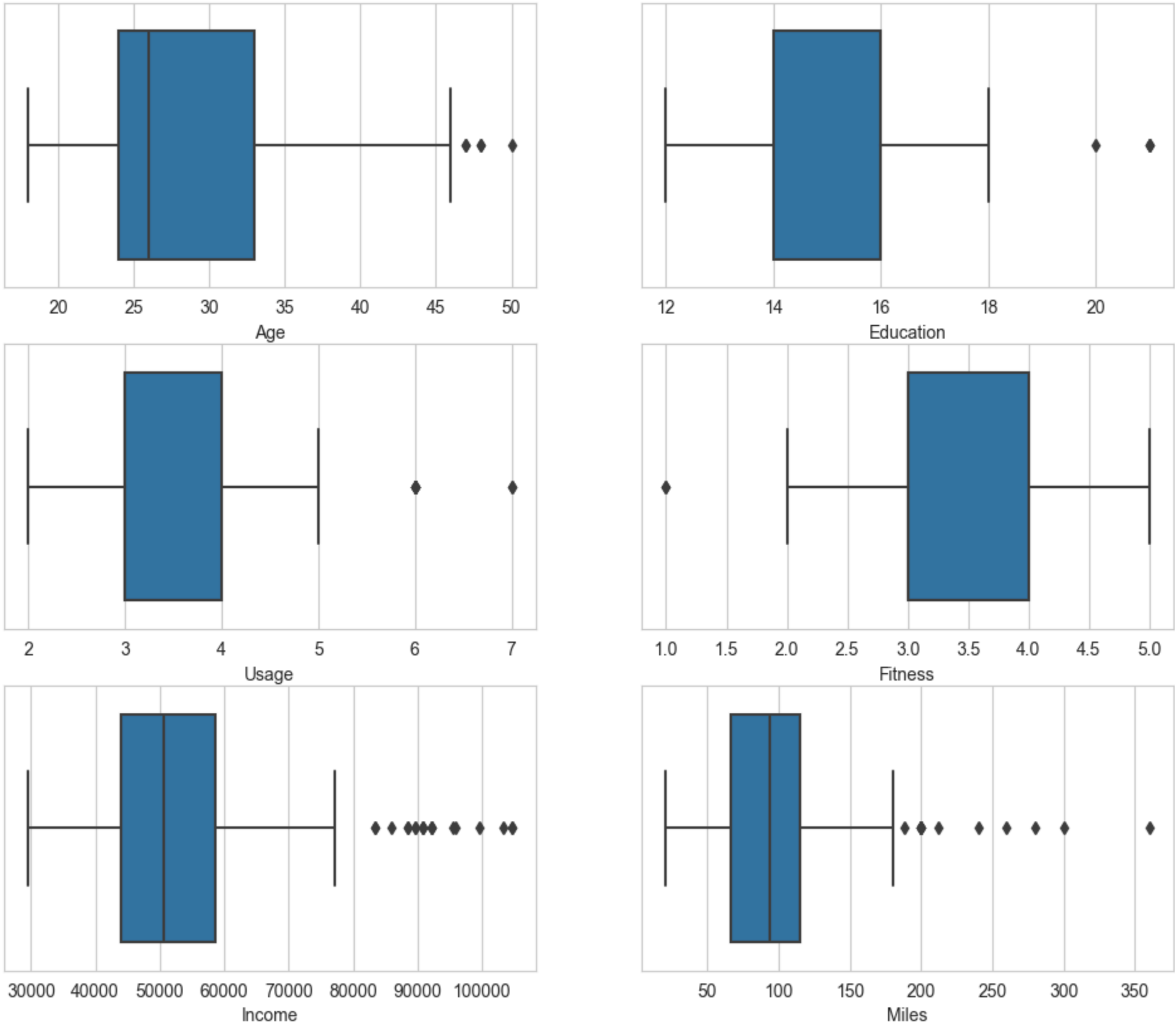
In [10]:
```
fig,axis = plt.subplots(3, 2, figsize=(12,10))
sns.histplot(data=df, x='Age', kde=True, ax=axis[0, 0], color='blue')
```

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



Outliers detection using Boxplot

In [11]:
```
fig, axis = plt.subplots(3, 2, figsize=(12,10))
sns.boxplot(data=df, x='Age', orient="h", ax=axis[0, 0])
sns.boxplot(data=df, x='Education', orient="h", ax=axis[0, 1])
sns.boxplot(data=df, x='Usage', orient="h", ax=axis[1, 0])
sns.boxplot(data=df, x='Fitness', orient="h", ax=axis[1, 1])
sns.boxplot(data=df, x='Income', orient="h", ax=axis[2, 0])
sns.boxplot(data=df, x='Miles', orient="h", ax=axis[2, 1])
plt.show()
```



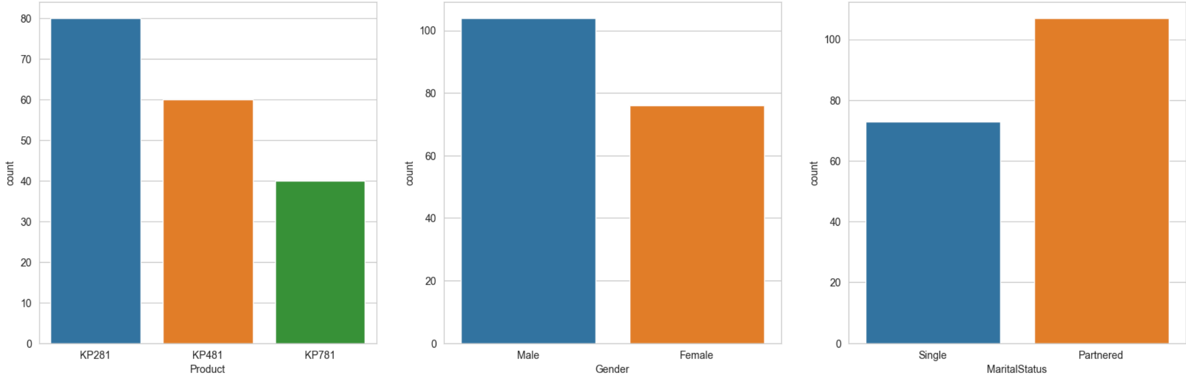
Observations.

```
From Boxplot we can clearly find out that:
1. "Income" and "Miles" have more outliers than other parameters.
```

Understanding the distribution of the qualitative attributes:

1. Product
2. Gender
3. Marital Status

In [12]:
```
fig, axs = plt.subplots(1,3, figsize=(20, 6))
sns.countplot(data=df, x='Product', ax=axs[0])
sns.countplot(data=df, x='Gender', ax=axs[1])
sns.countplot(data=df, x='MaritalStatus', ax=axs[2])
plt.show()
```



Observations.

1. "KP281" is the most frequent product.
2. There are more "Males" in data than "Females".
3. More "Partnered" persons are there in the data.

To be precise - normalized count for each person variable is shown below

In [13]:
```
df1 = df[['Product', 'Gender', 'MaritalStatus']].melt()
df1.groupby(["variable", "value"])[['value']].count()/len(df)
```

Out[13]:

		value
variable	**value**	
Gender	**Female**	0.422222
	Male	0.577778
MaritalStatus	**Partnered**	0.594444
	Single	0.405556
Product	**KP281**	0.444444
	KP481	0.333333
	KP781	0.222222

Observations.

Product

```
1. 44.44% of the customers have purchased KP281 product.
2. 33.33% of the customers have purchased KP481 product.
3. 22.22% of the customers have purchased KP781 product.
```

Gender

```
1. 57.78% of the customer are Male and rest are Females.
```

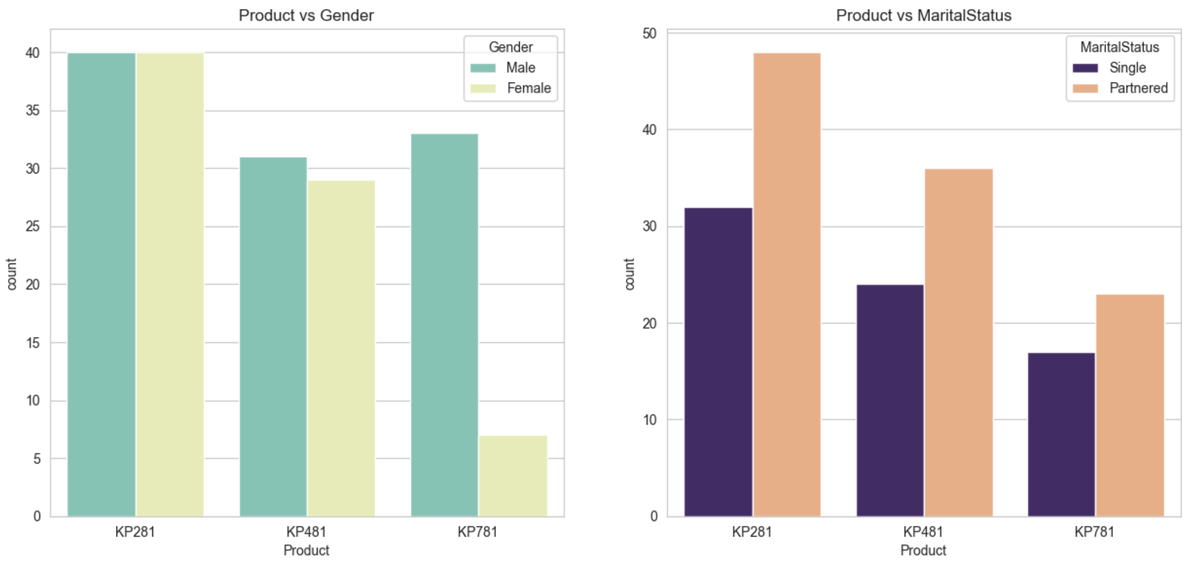
Marital Status

```
1. 59.44% of the customers are partnered.
```

Bivariate Analysis

checking if features - Gender and marital status have any effect on the product purchased

In [14]:
```
sns.set_style(style='whitegrid')
fig, axs = plt.subplots(1, 2, figsize=(15, 6.5))
sns.countplot(data=df, x='Product', hue='Gender', palette=['#7fcdbb', '#edf8b1'], a
axs[0].set_title("Product vs Gender")
sns.countplot(data=df, x='Product', hue="MaritalStatus", palette=['#432371', '#fAAE
axs[1].set_title("Product vs MaritalStatus")
plt.show()
```



Oberservations

Product vs Gender

```
1. Equal number of Males and Females have purchased KP281 product
and almost same for the product KP481.
2. Most of male customer have purchased the KP781.
```

Product vs MaritalStatus

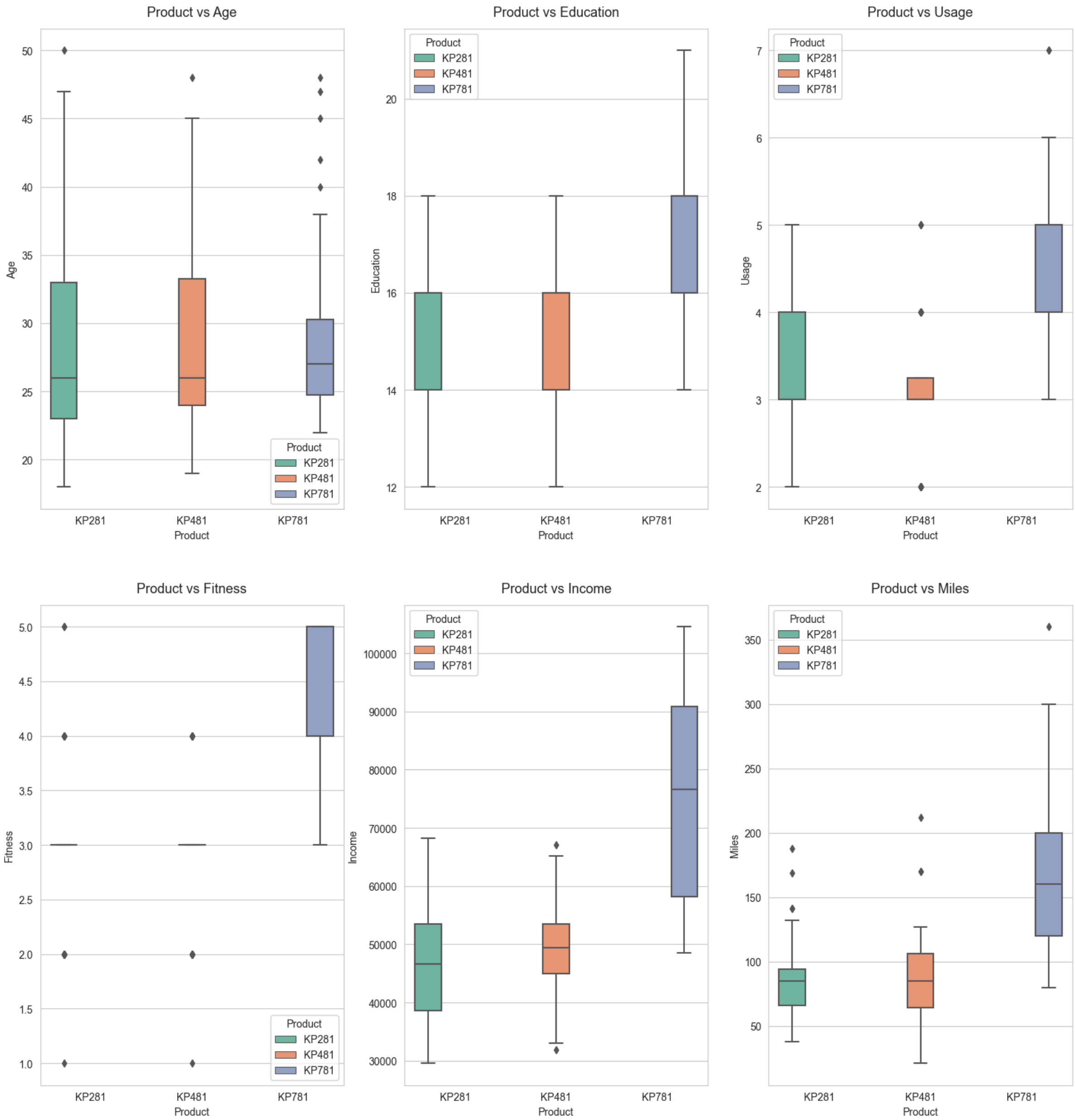
```
1. Customers who is Partnered, is most likely to purchase the
product and it is true for all the products.
```

Checking if following features have any effect on the product purchased

1. Age
2. Education
3. Usage

4. Fitness
5. Income
6. Miles

In [17]:
```
var = ["Age", "Education", "Usage", "Fitness", "Income", "Miles"]
sns.set_style("whitegrid")
fig,axs = plt.subplots(2, 3, figsize=(18, 12))
fig.subplots_adjust(top=1.3)
count = 0
for i in range(2):
    for j in range(3):
        sns.boxplot(data=df, x='Product', y=var[count], ax=axs[i, j], hue="Product"
        axs[i,j].set_title(f"Product vs {var[count]}",pad=12,fontsize=13)
        count += 1
```



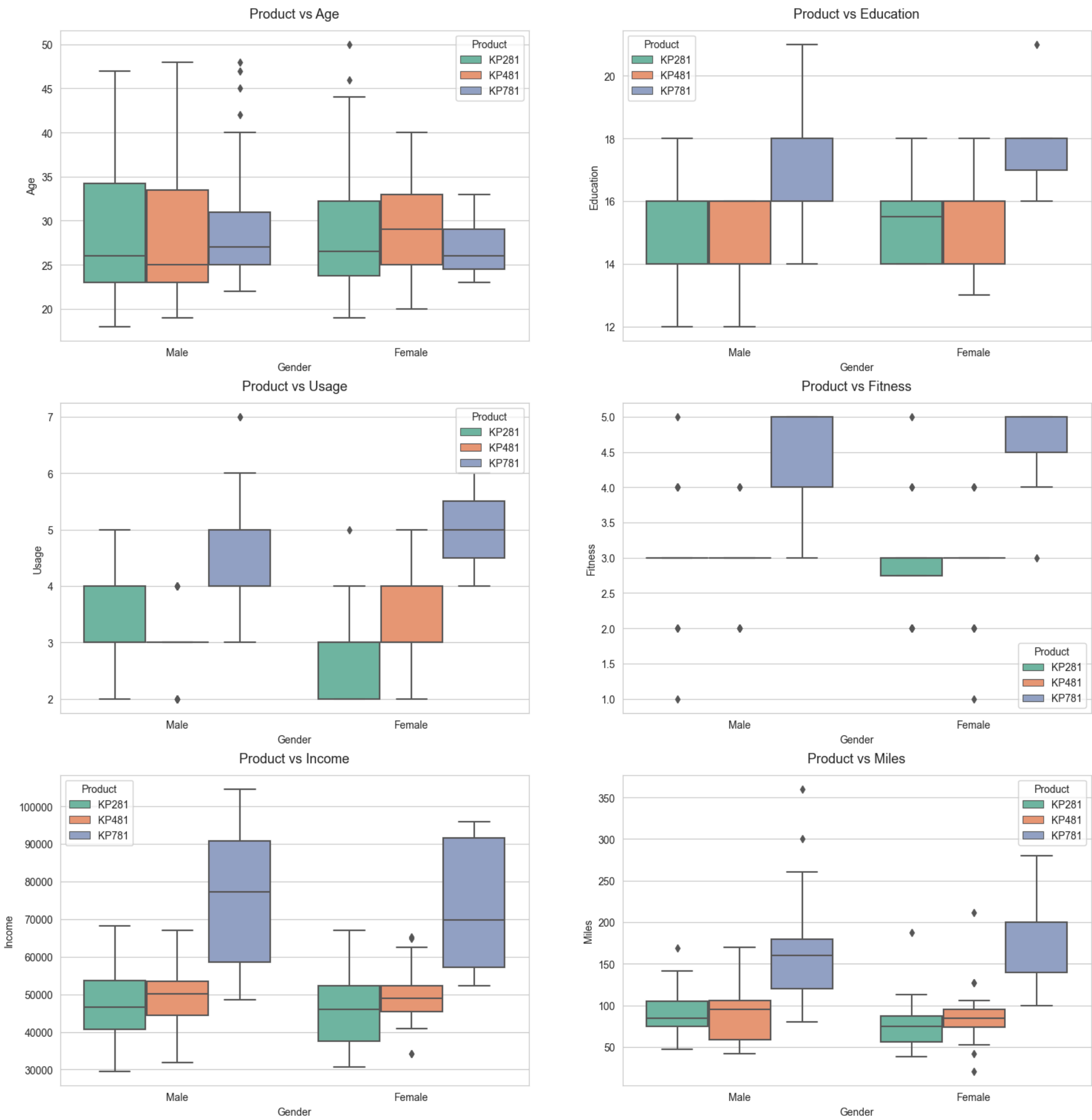
Multivariate Analysis

Checking if following features have any effect on the product purchased

1. Age

2. Education
3. Usage
4. Fitness
5. Income
6. Miles

In [18]:
```
var= ['Age','Education','Usage','Fitness','Income','Miles']
sns.set_style("whitegrid")
fig,axs=plt.subplots(3,2,figsize=(18,12))
fig.subplots_adjust(top=1.3)
count=0
for i in range(3):
    for j in range(2):
        sns.boxplot(data=df,x='Gender',y=var[count],hue='Product',ax=axs[i,j],palet
        axs[i,j].set_title(f"Product vs {var[count]}",pad=12,fontsize=13)
        count+=1
```



Observations

1. In both Gender, Customers whose education is greater than 16 prefer to buy KP781 product.
2. In both Gender, Customer who are planning to use treadmill more than 4 times prefer to buy KP781 Product.
3. Females who are planning to use treadmill 3-4 times a week, are more likely to buy KP481 product
4. In both Gender, Customer whose income is more than 55000, are more likely to buy KP781 product

Computing Marginal and Conditional Probability

Marginal Probability

In [20]:
```
pd.concat([df.Product.value_counts(), df.Product.value_counts(normalize=True)], key
```

Out[20]:

	counts	Marginal_Prob
KP281	80	0.444444
KP481	60	0.333333
KP781	40	0.222222

Conditinal Probability

Probability of each product given gender

In [21]:
```
def p_prod_given_gender(gender, print_marginal=False):
    if gender!= "Female" and gender!= "Male":
        return "Invalid gender value."

    df1= pd.crosstab(index=df['Gender'],columns=[df['Product']])
    p_781= df1['KP781'][gender] / df1.loc[gender].sum()
    p_481= df1['KP481'][gender] / df1.loc[gender].sum()
    p_281= df1['KP281'][gender] / df1.loc[gender].sum()

    if print_marginal:
        print(f"P(Male): {df1.loc['Male'].sum()/len(df):.2f}")
        print(f"P(Female): {df1.loc['Female'].sum()/len(df):.2f}")

    print(f"P(KP781/{gender}):{p_781:.2f}")
    print(f"P(KP481/{gender}):{p_481:.2f}")
    print(f"P(KP281/{gender}):{p_281:.2f}\n")

p_prod_given_gender('Male',True)
p_prod_given_gender('Female')
```

```
P(Male): 0.58
P(Female): 0.42
P(KP781/Male):0.32
P(KP481/Male):0.30
P(KP281/Male):0.38

P(KP781/Female):0.09
P(KP481/Female):0.38
P(KP281/Female):0.53
```

Probability of each product given marital status

In [22]:

```
def p_prod_given_MaritalStatus(status, print_marginal=False):
    if status!= "Single" and status!= "Partnered":
        return " invalid MaritalStatus value."

    df1= pd.crosstab(index=df['MaritalStatus'],columns=[df['Product']])
    p_781= df1['KP781'][status] / df1.loc[status].sum()
    p_481= df1['KP481'][status] / df1.loc[status].sum()
    p_281= df1['KP281'][status] / df1.loc[status].sum()

    if print_marginal:
        print(f"P(Single): {df1.loc['Single'].sum()/len(df):.2f}")
        print(f"P(Partnered): {df1.loc['Partnered'].sum()/len(df):.2f}\n")

    print(f"P(KP781/{status}):{p_781:.2f}")
    print(f"P(KP481/{status}):{p_481:.2f}")
    print(f"P(KP281/{status}):{p_281:.2f}\n")

p_prod_given_MaritalStatus('Single',True)
p_prod_given_MaritalStatus('Partnered')
```

```
P(Single): 0.41
P(Partnered): 0.59

P(KP781/Single):0.23
P(KP481/Single):0.33
P(KP281/Single):0.44

P(KP781/Partnered):0.21
P(KP481/Partnered):0.34
P(KP281/Partnered):0.45
```

Recommendations

1. KP781 should bw marketed as a Premium Model and marketing it to high income groups and educational over 20 years market segments could result in more sales.
2. Aerofit should conduct market research to determine if it can attract customers with income under 40000 to expand its customer base.
3. The KP781 is a premium model, so it is ideally suited for sporty people who have a high average weekly mileage..

In [ ]: