Q-1. Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset

```
from google.colab import files
uploaded = files.upload()
```

Choose Files aerofit_treadmill.csv

- **aerofit_treadmill.csv**(text/csv) - 7279 bytes, last modified: 10/9/2024 - 100% done

```
Saving aerofit_treadmill.csv to aerofit_treadmill.csv
```

```
import pandas as pd
# If you used Google Colab to upload
df = pd.read_csv('aerofit_treadmill.csv')
```

```
# Display data types of each column
data_types = df.dtypes
print("Data types of each column:\n", data_types)
```

```
Data types of each column:
 Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

```
# Get the shape of the dataset
shape = df.shape
print("Number of rows and columns:\n", shape)
```

```
Number of rows and columns:
 (180, 9)
```

```
# Check for missing values
missing_values = df.isnull().sum()
print("Number of missing values in each column:\n", missing_values)
```

```
Number of missing values in each column:
 Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

Q-2. #Detect Outliers ○ Find the outliers for every continuous variable in the dataset Hint: We want you to use boxplots to find the outliers in the given dataset ○ Remove/clip the data between the 5 percentile and 95 percentile Hint: We want You to use np.clip() for clipping the data

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

# Load the dataset
df = pd.read_csv('aerofit_treadmill.csv')

# Define continuous variables
continuous_vars = ['Age', 'Education', 'Usage', 'Income', 'Fitness', 'Miles']

# Create boxplots for each continuous variable
plt.figure(figsize=(15, 10))
for i, var in enumerate(continuous_vars, 1):
    plt.subplot(3, 2, i)
    plt.boxplot(df[var].dropna())  # Drop NA values for plotting
    plt.title(f'Boxplot of {var}')
plt.tight_layout()
plt.show()

# Clip the data for continuous variables
for var in continuous_vars:
    lower_bound = df[var].quantile(0.05)
    upper_bound = df[var].quantile(0.95)
    df[var] = np.clip(df[var], lower_bound, upper_bound)

# Display the updated DataFrame with clipped values
print(df[continuous_vars].describe())
```

Show hidden output

Q.3 Check if features like marital status, Gender, and age have any effect on the product purchased ○ Find if there is any relationship between the categorical variables and the output variable in the data. ○ Find if there is any relationship between the continuous variables and the output

variable in the data.

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

# Load the dataset
df = pd.read_csv('aerofit_treadmill.csv')

# Set the aesthetic style of the plots
sns.set(style="whitegrid")

# Count plots for categorical variables
plt.figure(figsize=(15, 5))

# Marital Status
plt.subplot(1, 3, 1)
sns.countplot(data=df, x='MaritalStatus', hue='Product')
plt.title('Product Purchased by Marital Status')

# Gender
plt.subplot(1, 3, 2)
sns.countplot(data=df, x='Gender', hue='Product')
plt.title('Product Purchased by Gender')

# Age (converted to categorical for the plot)
df['AgeGroup'] = pd.cut(df['Age'], bins=[0, 20, 30, 40, 50, 60, 70, 80], right=False)
plt.subplot(1, 3, 3)
sns.countplot(data=df, x='AgeGroup', hue='Product')
plt.title('Product Purchased by Age Group')

plt.tight_layout()
plt.show()

# Scatter plots for continuous variables
plt.figure(figsize=(15, 10))

# Age vs. Income
plt.subplot(3, 2, 1)
sns.scatterplot(data=df, x='Age', y='Income', hue='Product', alpha=0.6)
plt.title('Age vs. Income by Product Purchased')

# Education vs. Usage
plt.subplot(3, 2, 2)
sns.scatterplot(data=df, x='Education', y='Usage', hue='Product', alpha=0.6)
plt.title('Education vs. Usage by Product Purchased')

# Income vs. Miles
plt.subplot(3, 2, 3)
sns.scatterplot(data=df, x='Income', y='Miles', hue='Product', alpha=0.6)
plt.title('Income vs. Miles by Product Purchased')

# Fitness vs. Miles
plt.subplot(3, 2, 4)
sns.scatterplot(data=df, x='Fitness', y='Miles', hue='Product', alpha=0.6)
plt.title('Fitness vs. Miles by Product Purchased')

plt.tight_layout()
plt.show()
```

Show hidden output

Q.4 - Representing the Probability ○ Find the marginal probability (what percent of customers have purchased KP281, KP481, or KP781) ○ Find the probability that the customer buys a product based on each column. (Example: given that a customer is female, what is the probability she'll purchase a KP481)

```
import pandas as pd

# Load the dataset
df = pd.read_csv('aerofit_treadmill.csv')

# Step 1: Marginal Probability
marginal_counts = df['Product'].value_counts(normalize=True) * 100
print("Marginal Probability (Percentage of Customers by Product):\n", marginal_counts)

# Step 2: Probability Based on Each Column
# Probability of purchasing based on Gender
gender_product_prob = pd.crosstab(df['Gender'], df['Product'], normalize='index') * 100
print("\nProbability of Product Purchase by Gender (Percentage):\n", gender_product_prob)

# Probability of purchasing based on Marital Status
marital_status_product_prob = pd.crosstab(df['MaritalStatus'], df['Product'], normalize='index') * 100
print("\nProbability of Product Purchase by Marital Status (Percentage):\n", marital_status_product_prob)

# Step 3: Conditional Probability
# Example: Probability of purchasing KP481 given the customer is Female
conditional_probability = (df[(df['Gender'] == 'Female') & (df['Product'] == 'KP481')].shape[0] /
                           df[df['Gender'] == 'Female'].shape[0]) * 100

print(f"\nConditional Probability of purchasing KP481 given the customer is Female: {conditional_probability:.2f}%")
```

```
Marginal Probability (Percentage of Customers by Product):
 Product
KP281    44.444444
```

```
KP481    33.333333
KP781    22.222222
Name: proportion, dtype: float64

Probability of Product Purchase by Gender (Percentage):
 Product      KP281      KP481      KP781
Gender
Female   52.631579  38.157895   9.210526
Male     38.461538  29.807692  31.730769

Probability of Product Purchase by Marital Status (Percentage):
 Product            KP281      KP481      KP781
MaritalStatus
Partnered      44.859813  33.644860  21.495327
Single         43.835616  32.876712  23.287671

Conditional Probability of purchasing KP481 given the customer is Female: 38.16%
```

Q-5.Check the correlation among different factors ○ Find the correlation between the given features in the table.

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

# Load the dataset
df = pd.read_csv('aerofit_treadmill.csv')

# Step 1: Calculate the correlation matrix
# Select only continuous numerical columns for correlation
correlation_matrix = df[['Age', 'Education', 'Usage', 'Income', 'Fitness', 'Miles']].corr()

# Step 2: Create a heatmap to visualize the correlation matrix
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", square=True, cbar_kws={"shrink": .8})
plt.title('Correlation Heatmap')
plt.show()
```

Show hidden output

Q-6. 6. Customer profiling and recommendation ○ Make customer profilings for each and every product.

```
import pandas as pd

# Load the dataset
df = pd.read_csv('aerofit_treadmill.csv')

# Function to create customer profile
def create_customer_profile(product_name):
    profile = df[df['Product'] == product_name].describe(include='all')
    return profile

# Create profiles for each product
profile_kp281 = create_customer_profile('KP281')
profile_kp481 = create_customer_profile('KP481')
profile_kp781 = create_customer_profile('KP781')

print("Customer Profile for KP281:\n", profile_kp281)
print("\nCustomer Profile for KP481:\n", profile_kp481)
print("\nCustomer Profile for KP781:\n", profile_kp781)
```

Show hidden output

Start coding or generate with AI.