

Executive Summary

This case study analyzes customer purchasing behavior for AeroFit treadmills. The analysis shows that **income**, **fitness level**, and **usage intensity** are the primary drivers of product selection, while age has minimal influence.

KP781 serves a distinct premium segment consisting of high-income, high-fitness, and high-usage customers, whereas KP281 and KP481 cater to mass-market and mid-tier users with similar behavioral patterns.

```
In [3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: # Problem Statement (AeroFit):
# AeroFit wants to understand who buys which treadmill model so they can:

# Target the right customers
# Improve marketing & sales strategy
# Recommend products better

# Goal of Analysis:
# Create customer profiles for each treadmill product.
```

```
In [4]: df = pd.read_csv(r"C:\Users\shubh\Downloads\AeroFit_treadmill.csv")
df.head()
```

```
Out[4]:
```

	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 [5]: df.shape
```

```
Out[5]: (180, 9)
```

Insight:
The dataset contains 180 records with no missing values, making it suitable for analysis without additional cleaning.

```
In [6]: df.info()
# Checked dataset shape and structure
# Verified data types and null values
# Reviewed summary statistics to understand overall distributions
```

```
<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 [7]: df.describe()
```

	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 [8]: # which product sell the most
# KP281: Highest sales volume
# KP481: Moderate sales
# KP781: Lowest volume
# KP281 is the mass-market entry-level product, while KP781 is a niche premium product.
df["Product"].value_counts()
```

```
Out[8]:
```

```
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
```

```
In [9]: # who is buying each product
# gender distribution by product
# KP281 & KP481: Gender-balanced
# KP781: Strong male dominance
# Premium treadmills attract more performance-oriented male users.
pd.crosstab(df["Product"],df["Gender"], normalize = "index")*100
```

```
Out[9]:
```

	Gender	Female	Male
Product			
KP281		50.000000	50.000000
KP481		48.333333	51.666667
KP781		17.500000	82.500000

Insight:

- KP781 users show significantly higher usage intensity.
- KP281 and KP481 users display similar, moderate usage patterns.

```
In [12]: df.groupby(['Product', 'Gender']).size()
```

```
Out[12]:
```

Product	Gender	
KP281	Female	40
	Male	40
KP481	Female	29
	Male	31
KP781	Female	7
	Male	33

```
In [17]: # age distribution by product
df.groupby('Product')['Age'].describe()
# Mean age across all products ~ late 20s
# No major age-based differentiation
# Insight:
# Age is not a strong driver of product choice.
```

```
Out[17]:
```

	count	mean	std	min	25%	50%	75%	max
Product								
KP281	80.0	28.55	7.221452	18.0	23.00	26.0	33.00	50.0
KP481	60.0	28.90	6.645248	19.0	24.00	26.0	33.25	48.0
KP781	40.0	29.10	6.971738	22.0	24.75	27.0	30.25	48.0

```
In [21]: # income vs product
df.groupby("Product")["Income"].describe()
# KP281 & KP481: Overlapping income ranges
# KP781: Significantly higher income levels
# Insight:
# Income is a key differentiator, especially for KP781.
```

```
Out[21]:
```

	count	mean	std	min	25%	50%	75%	max
Product								
KP281	80.0	46418.025	9075.783190	29562.0	38658.00	46617.0	53439.0	68220.0
KP481	60.0	48973.650	8653.989388	31836.0	44911.50	49459.5	53439.0	67083.0
KP781	40.0	75441.575	18505.836720	48556.0	58204.75	76568.5	90886.0	104581.0

```
In [23]: # Fitness level vs product
df.groupby("Product")["Fitness"].describe()
# KP281 & KP481: Average fitness levels (mostly 3)
# KP781: High fitness levels (4-5)
# Insight:
# KP781 is bought by customers who are already fit, not beginners.
```

```
Out[23]:
```

	count	mean	std	min	25%	50%	75%	max
Product								
KP281	80.0	2.9625	0.664540	1.0	3.0	3.0	3.0	5.0
KP481	60.0	2.9000	0.629770	1.0	3.0	3.0	3.0	4.0
KP781	40.0	4.6250	0.667467	3.0	4.0	5.0	5.0	5.0

Note: Usage and Miles show identical distributions, indicating redundancy. In a production setting, one of these variables could be removed.

```
In [24]: df.groupby('Product')['Usage'].describe()
# KP781 users show much higher usage intensity
# KP281 & KP481 users have similar, moderate usage
# Insight:
# KP781 customers are serious and consistent users, validating the premium positioning.
```

```
Out[24]:
```

	count	mean	std	min	25%	50%	75%	max
Product								
KP281	80.0	82.787500	28.874102	38.0	66.0	85.0	94.0	188.0
KP481	60.0	87.933333	33.263135	21.0	64.0	85.0	106.0	212.0
KP781	40.0	166.900000	60.066544	80.0	120.0	160.0	200.0	360.0

```
In [25]: df.groupby('Product')['Miles'].describe()
# KP781 users show much higher usage intensity
# KP281 & KP481 users have similar, moderate usage
# Insight:
# KP781 customers are serious and consistent users, validating the premium positioning.
```

```
Out[25]:
```

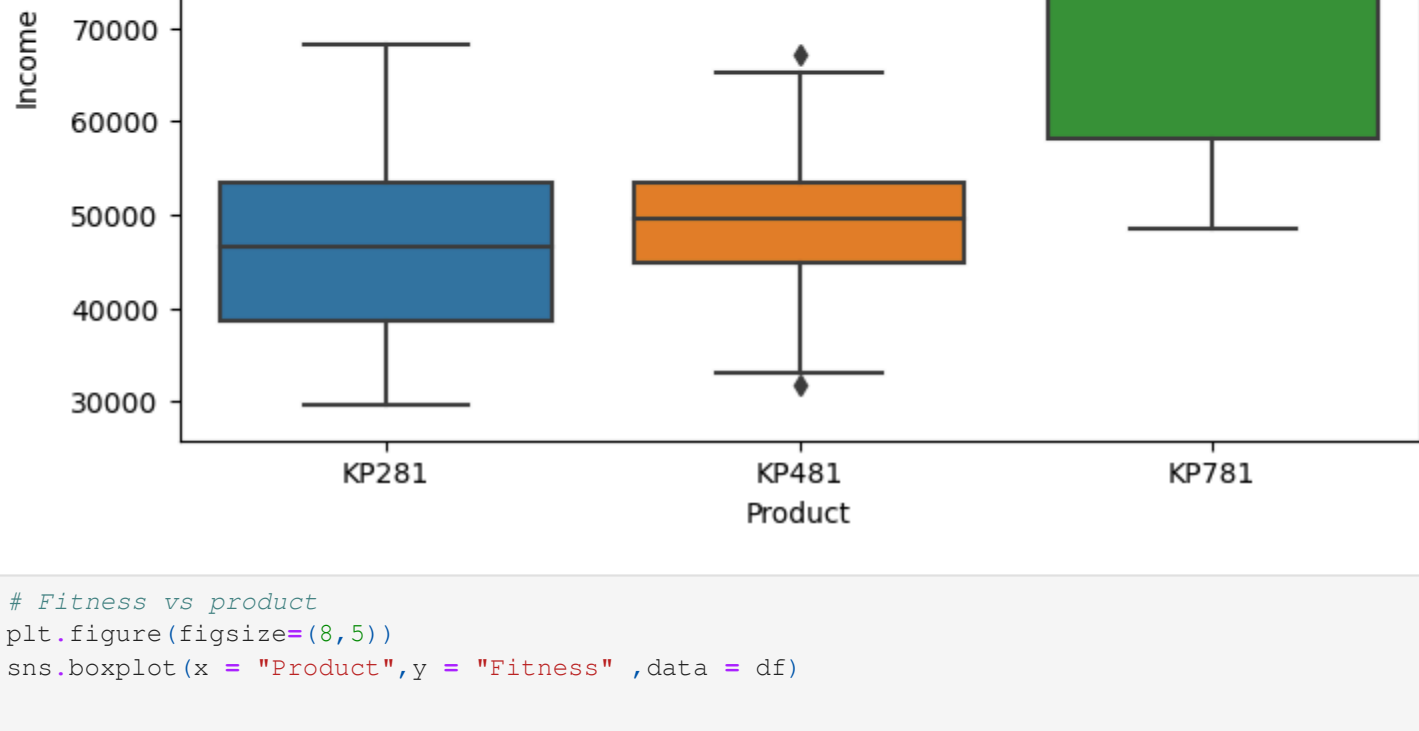
	count	mean	std	min	25%	50%	75%	max
Product								
KP281	80.0	82.787500	28.874102	38.0	66.0	85.0	94.0	188.0
KP481	60.0	87.933333	33.263135	21.0	64.0	85.0	106.0	212.0
KP781	40.0	166.900000	60.066544	80.0	120.0	160.0	200.0	360.0

```
In [27]: # Data Visualizations
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8,5))
sns.boxplot(x = "Product",y = "Income",data = df)

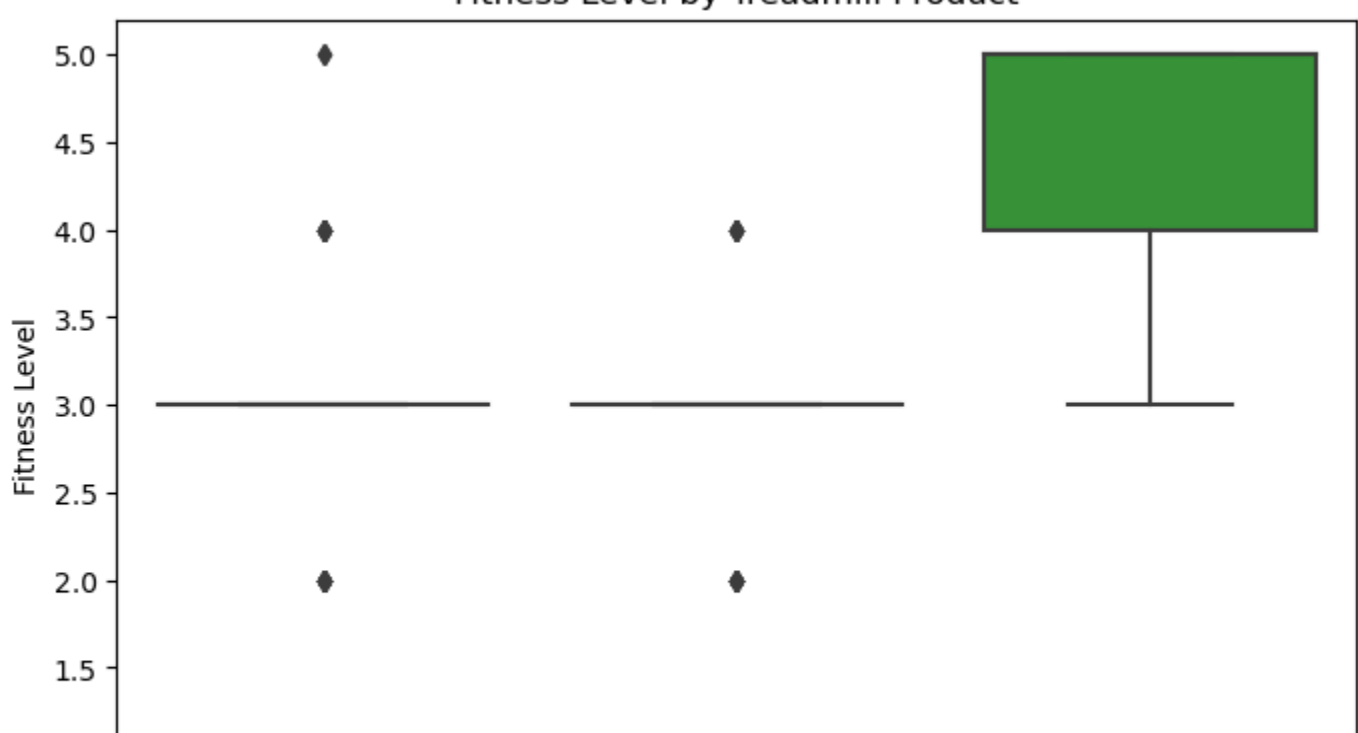
plt.title("Income Distribution by Treadmill Product")
plt.xlabel("Product")
plt.ylabel("Income")
plt.show()

# Boxplots were used to validate:
# Income differences across products
# Fitness level concentration
# Usage intensity patterns
# Visual analysis strongly supported all numerical findings.
```



```
In [28]: # Fitness vs product
plt.figure(figsize=(8,5))
sns.boxplot(x = "Product",y = "Fitness",data = df)

plt.title("Fitness Level by Treadmill Product")
plt.xlabel("Product")
plt.ylabel("Fitness Level")
plt.show()
```



```
In [19]: plt.figure(figsize=(8,5))
sns.boxplot(x = "Product",y = "Usage",data = df)
plt.xlabel("Usage by Treadmill Product")
plt.ylabel("Weekly Usage")
plt.show()
```



```
In [ ]: # Final Customer Personas
# KP281 - Entry-Level / Beginner Segment
# Balanced gender
# Mid-income customers
# Average fitness
# Moderate usage
# Goal: Start or maintain basic fitness
# KP481 - Mid-Level / Regular Users
# Slightly higher income than KP281
# Similar fitness and usage patterns
# Acts as a transition product
# KP781 - Premium / Performance Segment
# Predominantly male
# High-income customers
# High fitness levels
# Very high usage intensity
# Goal: Performance and endurance training
```

Final Business Recommendation

Product choice at AeroFit is primarily driven by income, fitness level, and usage intensity rather than age.

- KP281 should be marketed as an entry-level treadmill for beginners.
- KP481 should be positioned as a comfort-oriented upgrade.
- KP781 should maintain premium pricing and target performance-focused users.