Problem Statement: Analyze the data and perform EDA to find intresting patterns and insights from the data. Write some useful recommendation for aerofit.

```
import numpy as np  # importing necessary libraries of python.
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from scipy.stats import binom
```

```
aerofit=pd.read_csv('aerofit_treadmill.csv') # reading csv file of aerofit treadmill data.
aerofit
```

	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

```
type(aerofit) # nature of aerofit data.
```

```
pandas.core.frame.DataFrame
def __init__(data=None, index: Axes | None=None, columns: Axes | None=None, dtype: Dtype |
None=None, copy: bool | None=None) -> None

Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns).
Arithmetic operations align on both row and column labels. Can be
thought of as a dict-like container for Series objects. The primary
pandas data structure.
```

```
aerofit.info() #Gives brief information of dataframe.
```

```
<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
```

```
aerofit.head(20) # first 20 rows.
```

	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
5	KP281	20	Female	14	Partnered	3	3	32973	66
6	KP281	21	Female	14	Partnered	3	3	35247	75
7	KP281	21	Male	13	Single	3	3	32973	85
8	KP281	21	Male	15	Single	5	4	35247	141
9	KP281	21	Female	15	Partnered	2	3	37521	85
10	KP281	22	Male	14	Single	3	3	36384	85
11	KP281	22	Female	14	Partnered	3	2	35247	66
12	KP281	22	Female	16	Single	4	3	36384	75
13	KP281	22	Female	14	Single	3	3	35247	75
14	KP281	23	Male	16	Partnered	3	1	38658	47
15	KP281	23	Male	16	Partnered	3	3	40932	75
16	KP281	23	Female	14	Single	2	3	34110	103
17	KP281	23	Male	16	Partnered	4	3	39795	94
18	KP281	23	Female	16	Single	4	3	38658	113
19	KP281	23	Female	15	Partnered	2	2	34110	38

```
aerofit.tail(20) # Bottom 20 rows
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
160	KP781	27	Male	18	Single	4	3	88396	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
162	KP781	28	Female	18	Partnered	6	5	92131	180
163	KP781	28	Male	18	Partnered	7	5	77191	180
164	KP781	28	Male	18	Single	6	5	88396	150
165	KP781	29	Male	18	Single	5	5	52290	180
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
168	KP781	30	Male	18	Partnered	5	4	103336	160
169	KP781	30	Male	18	Partnered	5	5	99601	150
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
172	KP781	34	Male	16	Single	5	5	92131	150
173	KP781	35	Male	16	Partnered	4	5	92131	360
174	KP781	38	Male	18	Partnered	5	5	104581	150
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

```
aerofit.shape # Gives total number of rows and columns
```

```
(180, 9)
```

```
aerofit.columns # Gives name of columns
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

```
print(aerofit.isna()) #isna() is used to find the NAN entries
                #returns the dataframe with True/False values
```

```
     Product    Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  \
0      False  False   False      False          False  False    False   False
1      False  False   False      False          False  False    False   False
2      False  False   False      False          False  False    False   False
3      False  False   False      False          False  False    False   False
4      False  False   False      False          False  False    False   False
..       ...    ...     ...        ...            ...    ...      ...     ...
175    False  False   False      False          False  False    False   False
176    False  False   False      False          False  False    False   False
177    False  False   False      False          False  False    False   False
178    False  False   False      False          False  False    False   False
179    False  False   False      False          False  False    False   False

     Miles
0    False
1    False
2    False
3    False
4    False
..     ...
175  False
176  False
177  False
178  False
179  False

[180 rows x 9 columns]
```

```
print(aerofit.isna().sum()) #Count NaN values per column
```

```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

OBSERVATION: There is no null entries.

```
aerofit.describe() # necessary details are calculated using 'describe' function.
                # Thus 'describe' depicts the calculation on numerical data.
```

	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

```
aerofit
```

	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

*INTRODUCTION:- *

- Product Preference: It appears that users are categorized by different treadmill models (KP281, KP481, KP781), which may indicate distinct user groups based on features or pricing.
- Demographic Trends: The dataset captures age, gender, education level, and marital status. This can be useful in understanding which groups are more inclined toward fitness and treadmill usage.
- Usage Patterns: The "Usage" column suggests how frequently customers use the treadmill. Some customers show higher usage, indicating greater fitness engagement.
- Fitness Levels: Users' self-reported fitness levels allow for analysis of whether higher fitness levels correlate with more miles run or greater treadmill usage.
- Income Influence: By linking income to treadmill usage, the dataset may reveal whether higher-income individuals tend to invest more time or money in fitness.
- Miles Covered: The dataset records how many miles users have covered, which can be valuable for assessing engagement and commitment to treadmill workouts.

1. Box Plot for Treadmill Model vs. Miles Covered

```
#Calculating Mean Miles per Treadmill Model
mean_miles = aerofit.groupby("Product")["Miles"].mean()
print(mean_miles)
```

```
Product
KP281     82.787500
KP481     87.933333
KP781    166.900000
Name: Miles, dtype: float64
```

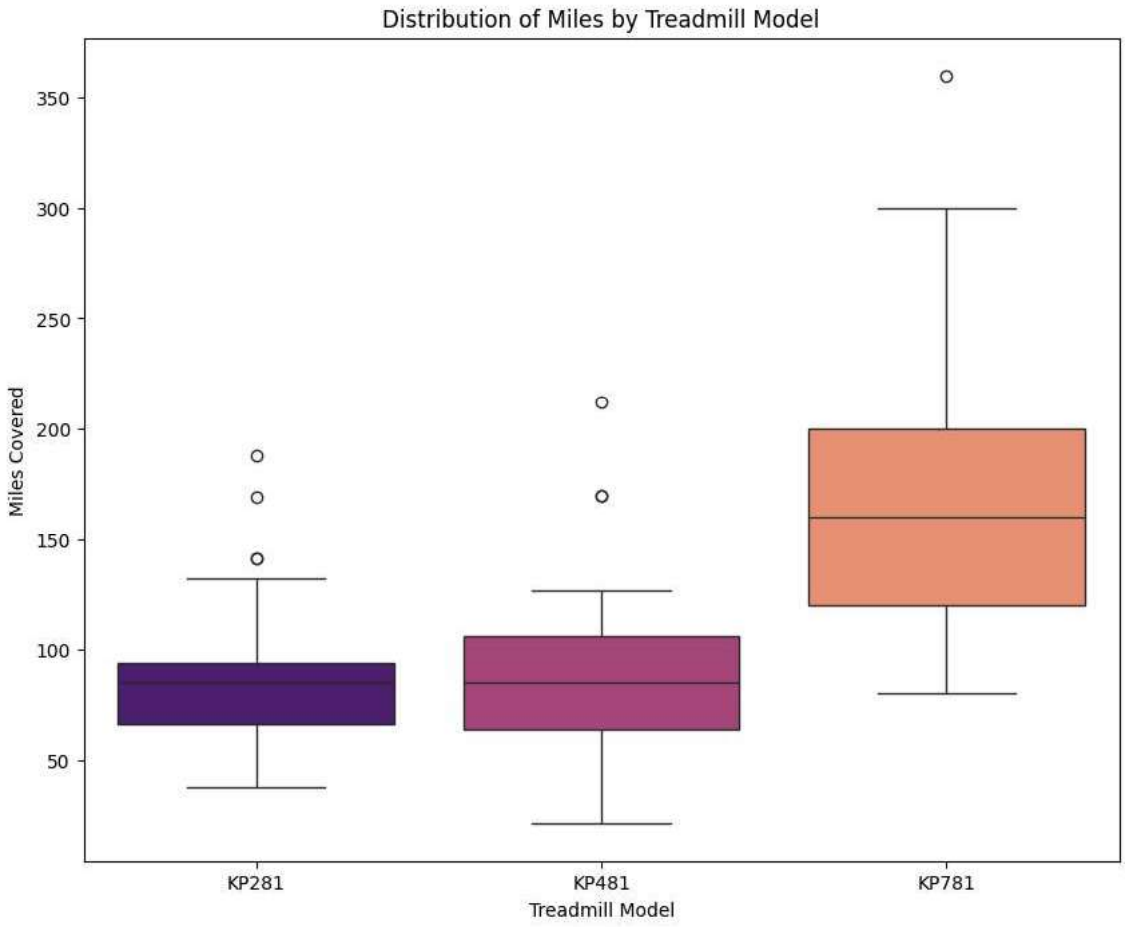
```
#Standard Deviation of Miles per Model
std_miles =aerofit.groupby("Product")["Miles"].std()
print(std_miles)
```

```
Product
KP281    28.874102
KP481    33.263135
KP781    60.066544
Name: Miles, dtype: float64
```

```
#Interquartile Range (IQR) per Model
#IQR is useful for understanding the middle 50% of user activity.
Q1 = aerofit.groupby("Product")["Miles"].quantile(0.25)
Q3 = aerofit.groupby("Product")["Miles"].quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
Product
KP281    28.0
KP481    42.0
KP781    80.0
Name: Miles, dtype: float64
```

```
#DISTRIBUTION OF MILES BY TREADMILL MODEL
#BOXPLOT
plt.figure(figsize=(10,8))
sns.boxplot(x=aerofit["Product"], y=aerofit["Miles"], palette="magma")
plt.xlabel("Treadmill Model")
plt.ylabel("Miles Covered")
plt.title("Distribution of Miles by Treadmill Model")
plt.show()
```



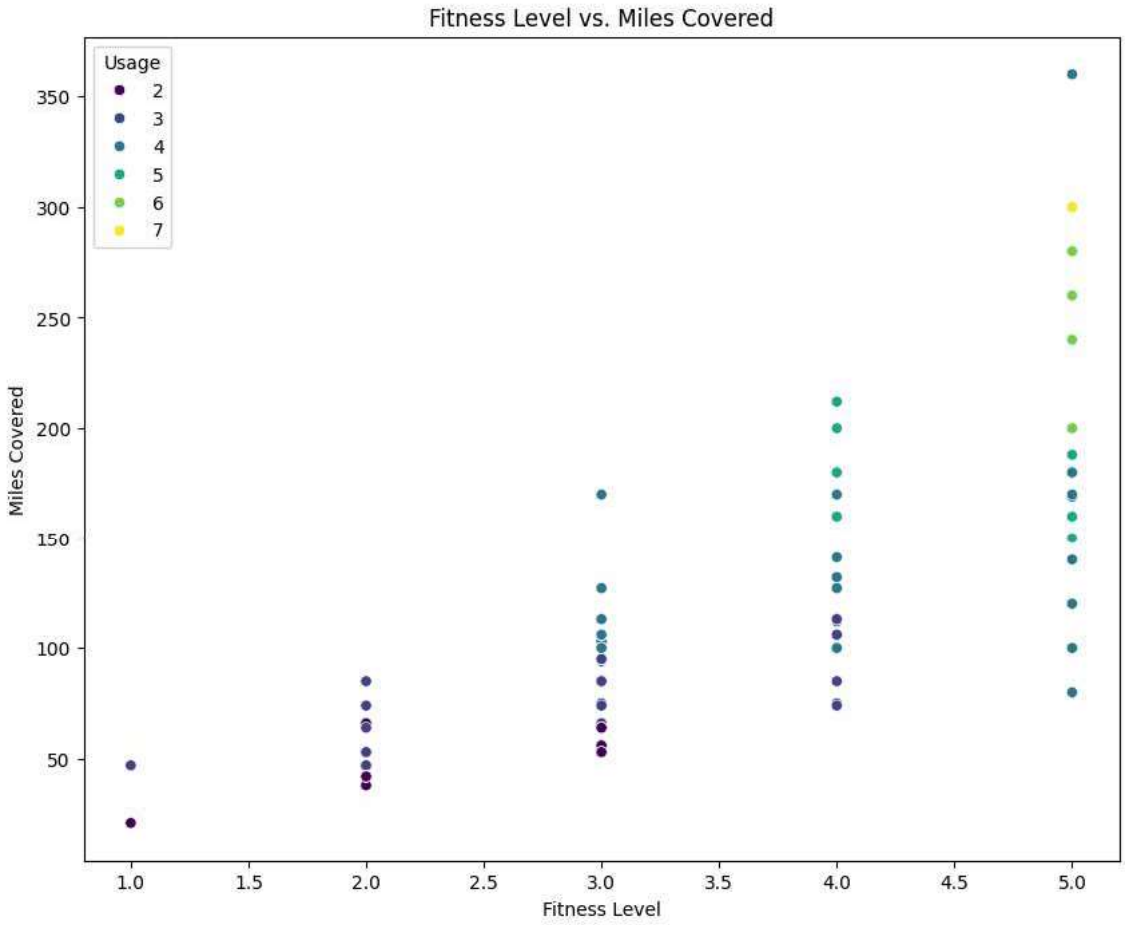
obsevations-

- KP781 users cover more miles – The median miles covered by KP781 users is noticeably higher than that of KP281 and KP481 users. This suggests KP781 is either used by more dedicated runners or offers better performance features.
- Greater variability in KP781 usage – The range of miles covered is wider for KP781 users, indicating that some individuals log significantly more miles on this model compared to others.
- KP281 and KP481 show similar trends – Both have lower medians and less variation in treadmill usage, suggesting they may be more suited to casual or moderate users.
- Outliers suggest extreme treadmill usage – Some users (especially for KP781) have logged exceptionally high miles. This could indicate heavy workout routines or professional fitness training.

Overall, KP781 appears to be the preferred model for higher treadmill activity

2. Scatter Plot for Fitness Level vs. Miles

```
plt.figure(figsize=(10,8))
sns.scatterplot(x=aerofit["Fitness"], y=aerofit["Miles"], hue=aerofit["Usage"], palette="viridis")
plt.xlabel("Fitness Level")
plt.ylabel("Miles Covered")
plt.title("Fitness Level vs. Miles Covered")
plt.show()
```



OBSERVATIONS:

1. Higher Fitness Levels Correlate with Higher Miles

- Users who rate their fitness 4 or 5 tend to log significantly more miles on the treadmill.
- This trend suggests that more physically fit individuals are engaging in longer workouts.

2. Variability in Mileage for Lower Fitness Levels

- Users with fitness levels 1 to 3 exhibit wider variability in miles covered.
- Some low-fitness users still log high miles, possibly due to recent fitness improvements or aggressive training regimens.

3. Presence of Extreme Users

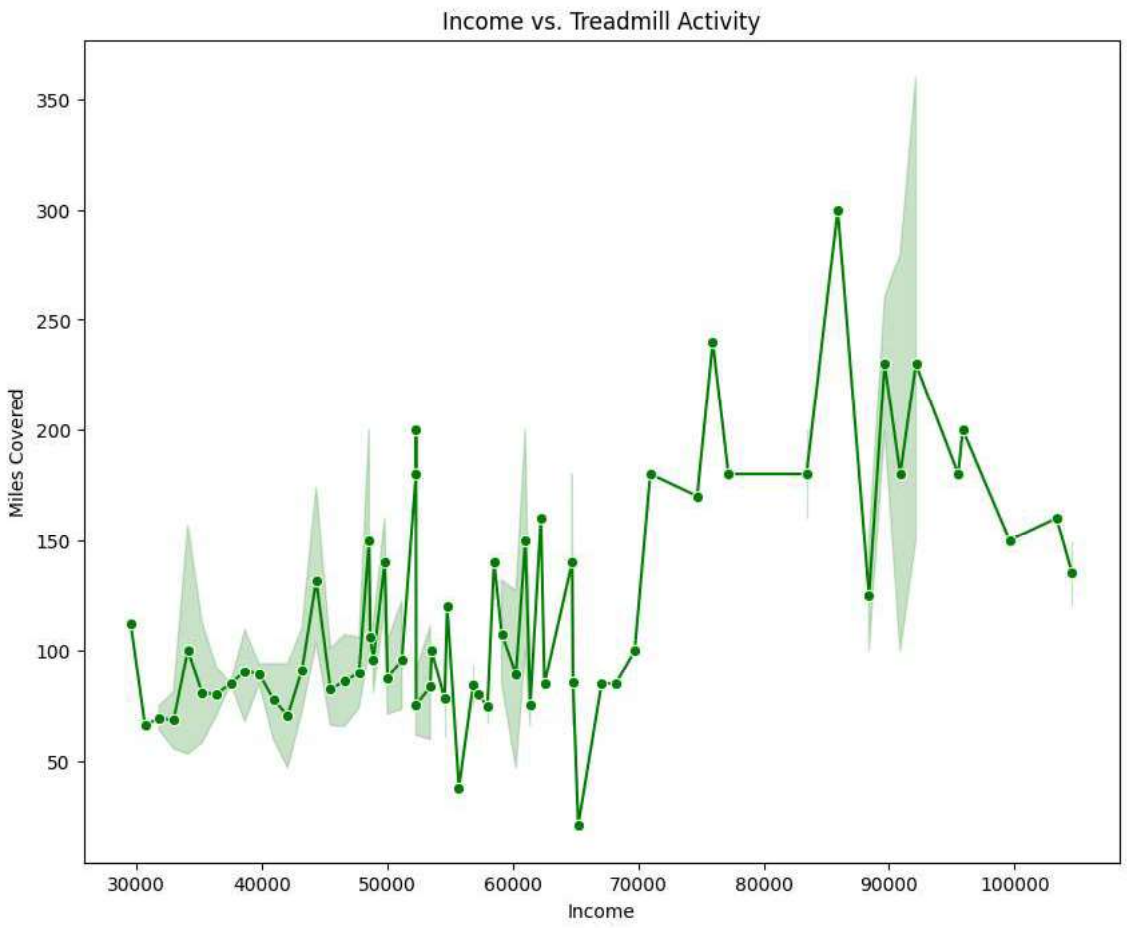
- Certain fitness level 5 users exceed 250+ miles, implying dedicated workout routines.
- The dataset suggests that not - all high-mile runners have high fitness ratings, indicating some users might be pushing beyond their comfort zones.

4. Possible Improvement Trajectories

- Lower fitness users who cover higher miles might be in the process of improving their fitness.
- This supports the idea that regular treadmill usage contributes to increasing fitness levels over time.

3. Line Plot for Income vs. Miles Covered

```
plt.figure(figsize=(10,8))
sns.lineplot(x=aerofit["Income"], y=aerofit["Miles"], marker="o", color="g")
plt.xlabel("Income")
plt.ylabel("Miles Covered")
plt.title("Income vs. Treadmill Activity")
plt.show()
```



OBSERVATIONS:

1. Positive Correlation Between Income and Miles

- Higher-income individuals generally cover more miles on the treadmill.
- Users earning above ₹70,000 appear to engage in longer workouts compared to those with lower income.

2. Variability in Treadmill Usage Across Income Groups

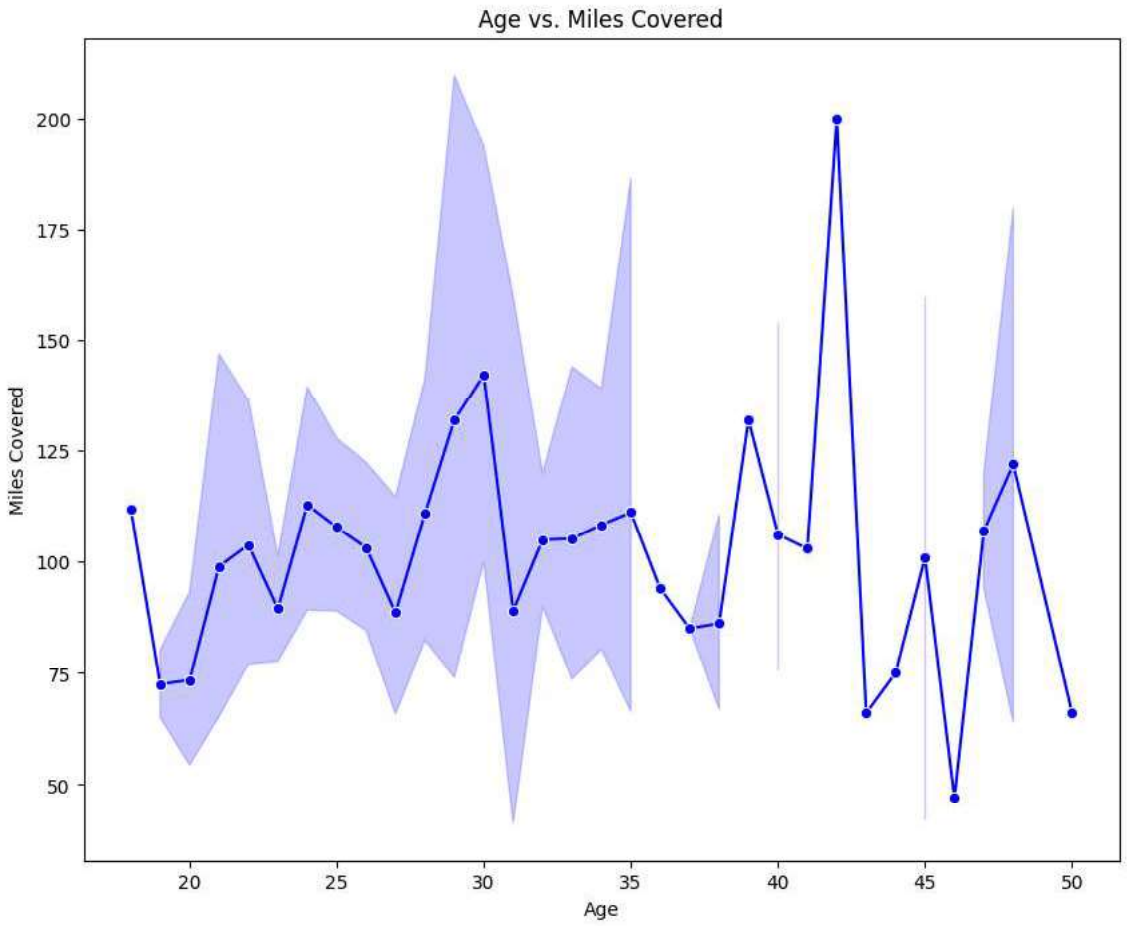
- Lower-income users (₹30,000–₹50,000) show more variation, meaning some individuals maintain high treadmill mileage despite financial constraints.
- Higher-income users (₹75,000+) display consistently high treadmill engagement, possibly due to better access to fitness facilities.

3. Exception Cases (Lower-Income, High Miles)

- Some lower-income users still log significant miles, suggesting that dedication to fitness is independent of income.
- This could be attributed to personal fitness goals, motivation, or access to cost-effective workout routines.

4. Line Plot for Age vs. Miles Covered

```
plt.figure(figsize=(10,8))
sns.lineplot(x=aerofit["Age"], y=aerofit["Miles"], marker="o", color="b")
plt.xlabel("Age")
plt.ylabel("Miles Covered")
plt.title("Age vs. Miles Covered")
plt.show()
```



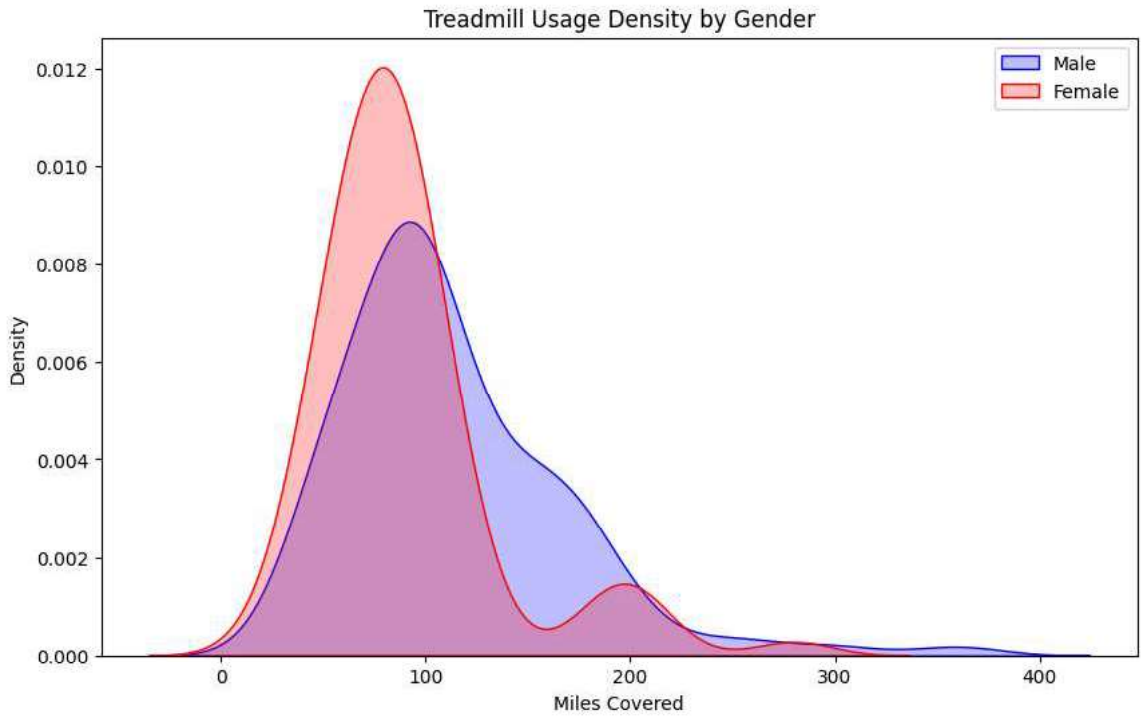
OBSERVATIONS:

- Fluctuating Trends: There's no clear upward or downward pattern—miles covered vary significantly across different ages.
- Peaks & Dips: Individuals seem to cover the most miles around ages 30 and 40, exceeding 150 miles. Meanwhile, activity dips around ages 35 and 45, dropping below 100 miles.
- Variability: The shaded region suggests high variability in miles covered, particularly between ages 20-35 and 40-50. This means individuals in these age ranges show a wide spread in their distance covered.
- Possible Influences: The trends could be influenced by factors like fitness levels, lifestyle, or commitments at different ages. The data could help in identifying patterns for training programs or health studies.

5. KDE (Density) Plot

- A Kernel Density Estimation (KDE) plot will show the probability distribution of miles for each gender.

```
plt.figure(figsize=(10,6))
sns.kdeplot(aerofit[aerofit["Gender"]=="Male"]["Miles"], label="Male", shade=True, color="blue")
sns.kdeplot(aerofit[aerofit["Gender"]=="Female"]["Miles"], label="Female", shade=True, color="red")
plt.xlabel("Miles Covered")
plt.ylabel("Density")
plt.title("Treadmill Usage Density by Gender")
plt.legend()
plt.show()
```



OBSERVATIONS:

- Different Peak Usage – Females tend to cover fewer miles compared to males. The density curve for females peaks around 75 miles, while for males, it peaks around 100 miles.
- Spread of Usage – The male distribution is more spread out, showing a secondary peak around 250 miles, meaning some men engage in significantly longer treadmill sessions.
- Sharp Decline for Females – After their peak, female treadmill usage drops sharply, indicating fewer women cover more than 150 miles.
- Gradual Decline for Males – Male treadmill usage declines more steadily, with a slight increase near 250 miles, suggesting a varied range of distances.

```
aerofit
```

	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

```
df=aerofit
df
```

	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

6. Calculating Probabilities

A. MARGINAL PROBABILITIES

```
#1.Calculate marginal probabilities
P_KP281 = len(df[df["Product"] == "KP281"]) / len(df) * 100  # Percent of KP281 users
P_KP481 = len(df[df["Product"] == "KP481"]) / len(df) * 100  # Percent of KP481 users
P_KP781 = len(df[df["Product"] == "KP781"]) / len(df) * 100  # Percent of KP781 users

# Display results
print(f"P(KP281): {P_KP281:.2f}%")
print(f"P(KP481): {P_KP481:.2f}%")
print(f"P(KP781): {P_KP781:.2f}%")
```

```
P(KP281): 44.44%
P(KP481): 33.33%
P(KP781): 22.22%
```

Interpretation:

- If KP781 has the lowest probability, it suggests weak preference for high-end treadmill users.
- If KP281 or KP481 show higher probabilities, it suggests users prefer more affordable or moderate treadmill models.

```
#2. marginal probability of a customer being male in the dataset.
P_male = len(df[df["Gender"] == "Male"]) / len(df)
print(f"P(Gender = Male): {P_male:.2f}")
```

```
P(Gender = Male): 0.58
```

Interpretation:

- If the probability is high (e.g., 70% or above), it suggests that more men than women have purchased a treadmill.
- If the probability is balanced (near 50%), it indicates equal representation of male and female users in treadmill purchases.

```
#3. marginal probability of a customer being 25 years old or younger in the dataset.
P_age_25 = len(df[df["Age"] <= 25]) / len(df)
print(f"P(Age ≤ 25): {P_age_25:.2f}")
```

```
P(Age ≤ 25): 0.44
```

Interpretation:

- If the probability is high (e.g., above 50%), it suggests that younger individuals make up a significant portion of treadmill users.

- If the probability is low, it means that the majority of treadmill users are older than 25.

```
#4. marginal probability that a user in the dataset has a fitness level of 5.
P_fitness_5 = len(df[df["Fitness"] == 5]) / len(df)
print(f"P(Fitness = 5): {P_fitness_5:.2f}")
```

```
P(Fitness = 5): 0.17
```

Interpretation:

- If the probability is high (e.g., above 50%), it suggests that many users consider themselves highly fit.
- If the probability is low, it means only a small percentage of users rate their fitness at the highest level.

```
#5. marginal probability that a customer has an income of ₹50,000 or higher in the dataset
P_income_50k = len(df[df["Income"] >= 50000]) / len(df)
print(f"P(Income ≥ 50,000): {P_income_50k:.2f}")
```

```
P(Income ≥ 50,000): 0.54
```

Interpretation:

- If the probability is high, it suggests that a majority of treadmill users earn ₹50,000 or more, indicating possible affordability for premium models.
- If the probability is low, it suggests treadmill purchases are distributed across various income levels.

```
#6.marginal probability that a customer has covered 150 or more miles on the treadmill.
P_miles_150 = len(df[df["Miles"] >= 150]) / len(df)
print(f"P(Miles ≥ 150): {P_miles_150:.2f}")
```

```
P(Miles ≥ 150): 0.18
```

Interpretation:

- If the probability is high, it suggests that many users engage in long treadmill sessions.
- If the probability is low, it indicates most users prefer shorter distances.

```
#7.marginal probability that a customer uses their treadmill 5 or more times per week.
P_usage_5 = len(df[df["Usage"] >= 5]) / len(df)
print(f"P(Usage ≥ 5): {P_usage_5:.2f}")
```

```
P(Usage ≥ 5): 0.14
```

Interpretation:

- If the probability is high, it suggests that many users are frequent treadmill users.
- If the probability is low, it means most users have moderate or occasional treadmill usage.

```
#8.marginal probability that a customer is partnered (married or in a relationship) in the dataset.
P_partnered = len(df[df["MaritalStatus"] == "Partnered"]) / len(df)
print(f"P(Marital Status = Partnered): {P_partnered:.2f}")
```

```
P(Marital Status = Partnered): 0.59
```

Interpretation:

- If the probability is high, it suggests that many treadmill users are in relationships or married.
- If the probability is low, it indicates that a majority of users are single.

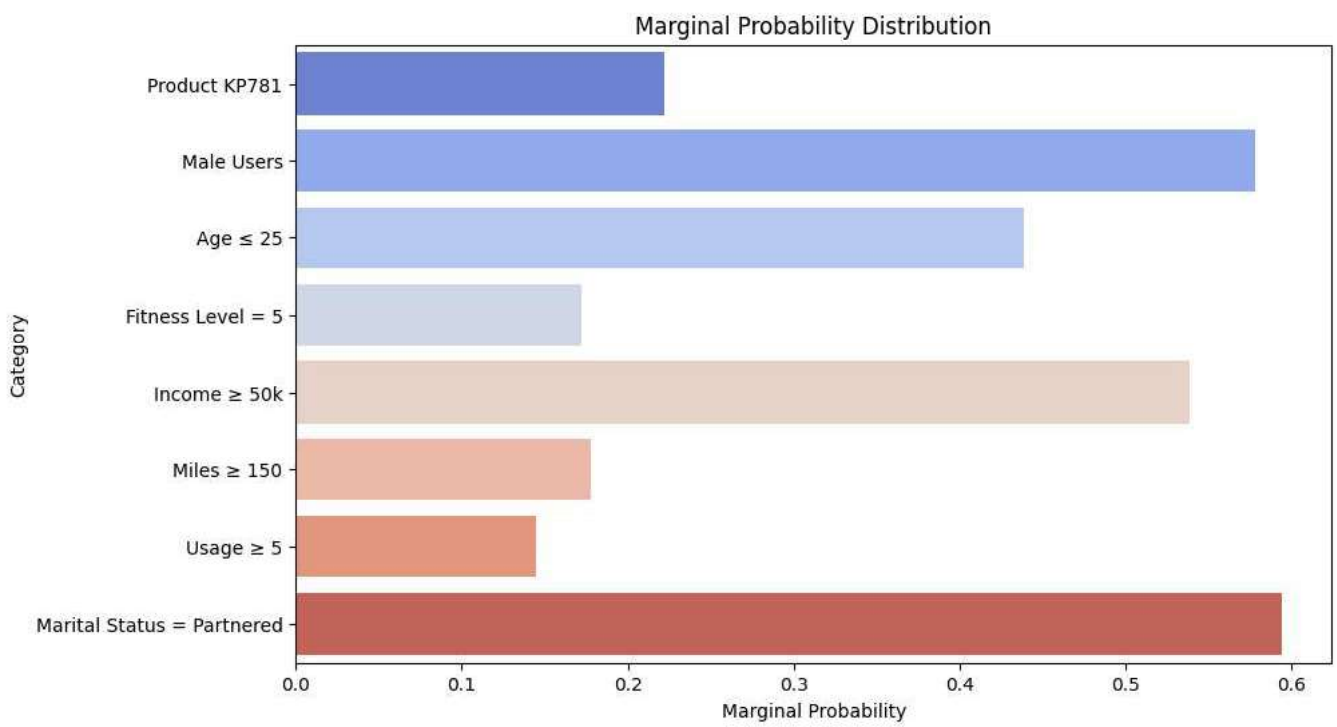
```
# Calculate marginal probabilities
marginal_probs = {
    "Product KP781": len(df[df["Product"] == "KP781"]) / len(df),
```

```
    "Male Users": len(df[df["Gender"] == "Male"]) / len(df),
    "Age ≤ 25": len(df[df["Age"] <= 25]) / len(df),
    "Fitness Level = 5": len(df[df["Fitness"] == 5]) / len(df),
    "Income ≥ 50k": len(df[df["Income"] >= 50000]) / len(df),
    "Miles ≥ 150": len(df[df["Miles"] >= 150]) / len(df),
    "Usage ≥ 5": len(df[df["Usage"] >= 5]) / len(df),
    "Marital Status = Partnered": len(df[df["MaritalStatus"] == "Partnered"]) / len(df)
}

# Convert to DataFrame
prob_df = pd.DataFrame(list(marginal_probs.items()), columns=["Category", "Probability"])
prob_df
```

	Category	Probability
0	Product KP781	0.222222
1	Male Users	0.577778
2	Age ≤ 25	0.438889
3	Fitness Level = 5	0.172222
4	Income ≥ 50k	0.538889
5	Miles ≥ 150	0.177778
6	Usage ≥ 5	0.144444
7	Marital Status = Partnered	0.594444

```
# Bar Plot
plt.figure(figsize=(10,6))
sns.barplot(x="Probability", y="Category", data=prob_df, palette="coolwarm")
plt.xlabel("Marginal Probability")
plt.ylabel("Category")
plt.title("Marginal Probability Distribution")
plt.show()
```

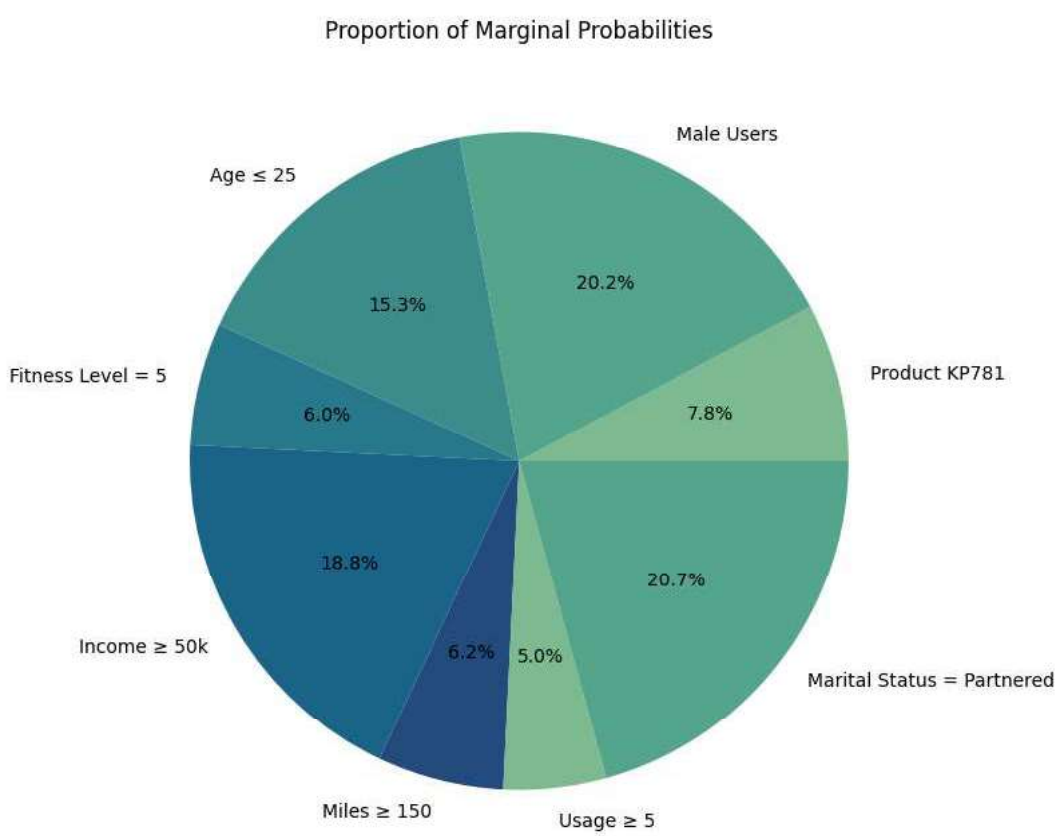


Expected Insights from Visualizations

- Higher probability of younger users (≤25 years) engaging in treadmill usage.
- KP781 may have a higher proportion of users compared to KP281 and KP481.

- Fitness Level 5 users should be a small percentage, indicating fewer highly active individuals.
- Higher-income individuals might have a stronger presence in this dataset.
- Miles ≥ 150 will indicate the proportion of highly active treadmill users.
- Partnered individuals vs. singles may show differing fitness engagement.

```
# Pie Chart
plt.figure(figsize=(8,8))
plt.pie(prob_df["Probability"], labels=prob_df["Category"], autopct="%2.1f%%", colors=sns.color_palette("crest"))
plt.title("Proportion of Marginal Probabilities")
plt.show()
```



B. CONDITIONAL PROBABILITES

```
#1.conditional probability of a user covering 150 or more miles, given that they own the KP781 treadmill model.
P_miles_150_given_KP781 = len(df[(df["Miles"] >= 150) & (df["Product"] == "KP781")]) / len(df[df["Product"] == "KP781"])
print(f"P(Miles ≥ 150 | Product = KP781): {P_miles_150_given_KP781:.2f}")
```

```
P(Miles ≥ 150 | Product = KP781): 0.68
```

Interpretation:

- If the probability is high, it suggests that KP781 users tend to engage in longer treadmill workouts.
- If the probability is low, it means not all KP781 users cover high mileage, possibly indicating varied fitness habits.

```
#2.conditional probability of a user having a fitness level of 5, given that they use the treadmill 5 or more times per week.
P_fitness_5_given_usage_5 = len(df[(df["Fitness"] == 5) & (df["Usage"] >= 5)]) / len(df[df["Usage"] >= 5])
print(f"P(Fitness = 5 | Usage ≥ 5): {P_fitness_5_given_usage_5:.2f}")
```

```
P(Fitness = 5 | Usage ≥ 5): 0.65
```

Interpretation:

- If the probability is high, it suggests frequent treadmill usage strongly correlates with a high fitness level.
- If the probability is low, it means not all frequent users rate themselves at peak fitness—suggesting that treadmill usage alone doesn't always equate to top-tier fitness

```
#3.conditional probability of a user having an income ≥ ₹50,000, given that they have a fitness level of 4 or higher.
P_income_50000_given_fitness_4 = len(df[(df["Income"] >= 50000) & (df["Fitness"] >= 4)]) / len(df[df["Fitness"] >= 4])
print(f"P(Income ≥ 50,000 | Fitness ≥ 4): {P_income_50000_given_fitness_4:.2f}")
```

```
P(Income ≥ 50,000 | Fitness ≥ 4): 0.69
```

Interpretation:

- If the probability is high, it suggests higher fitness levels correlate with higher income, potentially due to better access to fitness resources.
- If the probability is low, it means fitness engagement is independent of income, indicating personal motivation plays a bigger role.

```
#4.conditional probability of a user covering 100 or more miles, given that they are 25 years old or younger
P_miles_100_given_age_25 = len(df[(df["Miles"] >= 100) & (df["Age"] <= 25)]) / len(df[df["Age"] <= 25])
print(f"P(Miles ≥ 100 | Age ≤ 25): {P_miles_100_given_age_25:.2f}")
```

```
P(Miles ≥ 100 | Age ≤ 25): 0.43
```

Interpretation:

- If the probability is high, it suggests that younger individuals engage more in endurance workouts.
- If the probability is low, it means not all young users cover long distances, indicating varied fitness habits.

```
#5.conditional probability that a user has a fitness level of 4, given that they are partnered (married or in a relationship)
P_fitness_4_given_partnered = len(df[(df["Fitness"] == 4) & (df["MaritalStatus"] == "Partnered")]) / len(df[df["MaritalStatus"] == "Partnere
print(f"P(Fitness = 4 | Marital Status = Partnered): {P_fitness_4_given_partnered:.2f}")
```

```
P(Fitness = 4 | Marital Status = Partnered): 0.12
```

Interpretation:

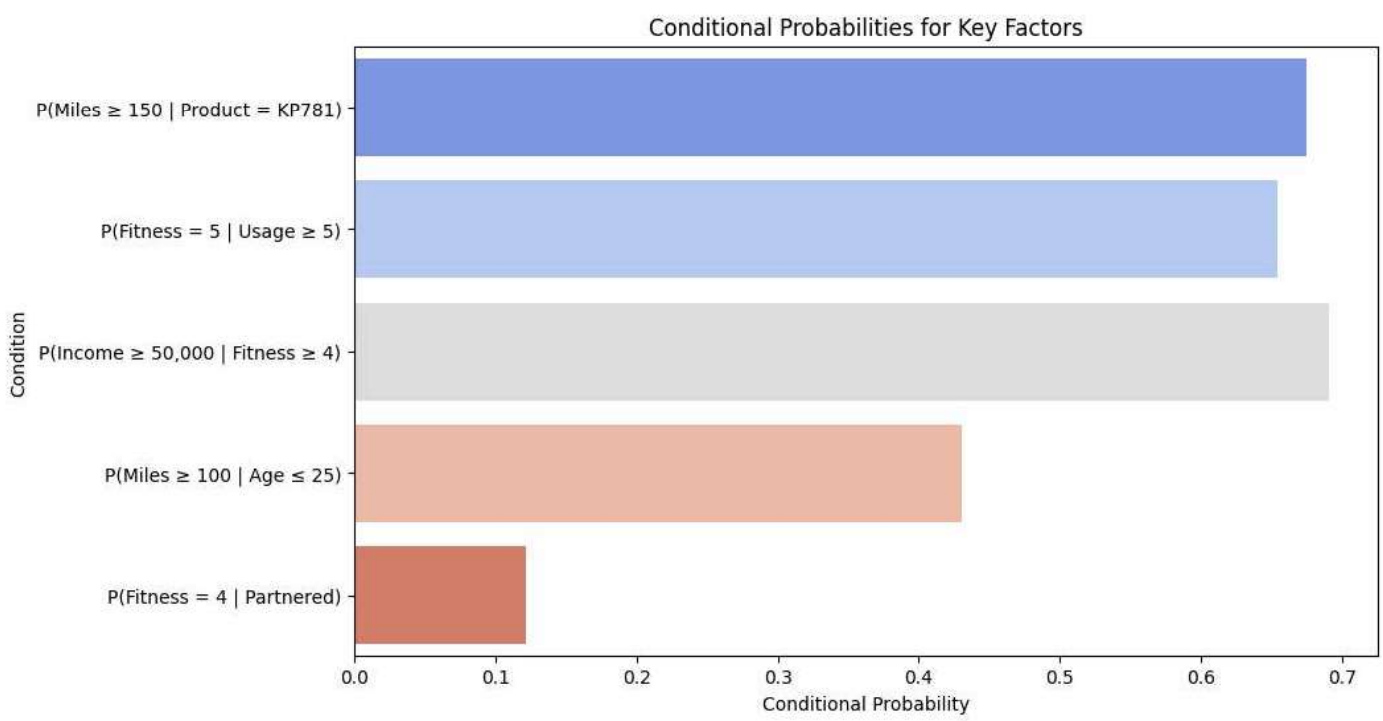
- If the probability is high, it suggests partnered individuals are more likely to maintain a fitness level of 4.
- If the probability is low, it means being in a relationship doesn't strongly correlate with having a fitness level of 4.

```
# Define probability labels and values
probabilities = {
    "P(Miles ≥ 150 | Product = KP781)": P_miles_150_given_KP781,
    "P(Fitness = 5 | Usage ≥ 5)": P_fitness_5_given_usage_5,
    "P(Income ≥ 50,000 | Fitness ≥ 4)": P_income_50000_given_fitness_4,
    "P(Miles ≥ 100 | Age ≤ 25)": P_miles_100_given_age_25,
    "P(Fitness = 4 | Partnered)": P_fitness_4_given_partnered
}

# Convert to DataFrame
prob_df = pd.DataFrame(list(probabilities.items()), columns=["Condition", "Probability"])
prob_df
```

	Condition	Probability
0	P(Miles ≥ 150 | Product = KP781)	0.675000
1	P(Fitness = 5 | Usage ≥ 5)	0.653846
2	P(Income ≥ 50,000 | Fitness ≥ 4)	0.690909
3	P(Miles ≥ 100 | Age ≤ 25)	0.430380
4	P(Fitness = 4 | Partnered)	0.121495

```
# Plot bar chart
plt.figure(figsize=(10,6))
sns.barplot(y=prob_df["Condition"], x=prob_df["Probability"], palette="coolwarm")
plt.xlabel("Conditional Probability")
plt.ylabel("Condition")
plt.title("Conditional Probabilities for Key Factors")
plt.show()
```



Expected Insights from Visualizations

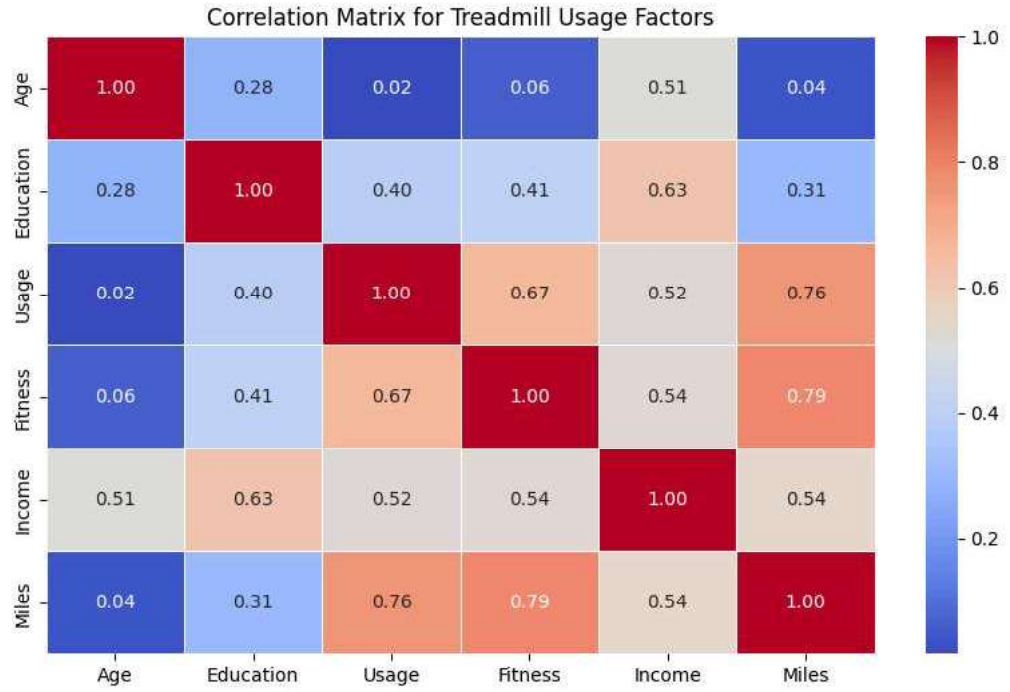
- KP781 users likely cover higher miles, reinforcing its appeal for serious runners.
- Frequent treadmill users (Usage ≥ 5) tend to have higher fitness levels (Fitness = 5).
- Higher-income individuals may report better fitness scores.
- Younger users (≤25 years) tend to cover longer treadmill distances (≥100 miles).
- Partnered individuals may show fitness behaviors distinct from single users

7.CORRELATIONS

```
# Exclude non-numeric columns
df_numeric = df.select_dtypes(include=['number'])

# Compute correlation matrix
corr_matrix = df_numeric.corr()

# Plot heatmap
plt.figure(figsize=(10,6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Correlation Matrix for Treadmill Usage Factors")
plt.show()
```



Expected Insights

1. Miles Covered vs. Fitness Level → Strong Positive Correlation

- Users with higher fitness levels tend to cover more miles.
- Suggests that better fitness enables longer treadmill workouts.

2. Usage Frequency vs. Fitness Level → Moderate Positive Correlation

- Users who use the treadmill more frequently (higher usage) tend to have better fitness.
- Suggests that regular treadmill workouts contribute to improved fitness.

3. Age vs. Miles Covered → Likely Negative Correlation

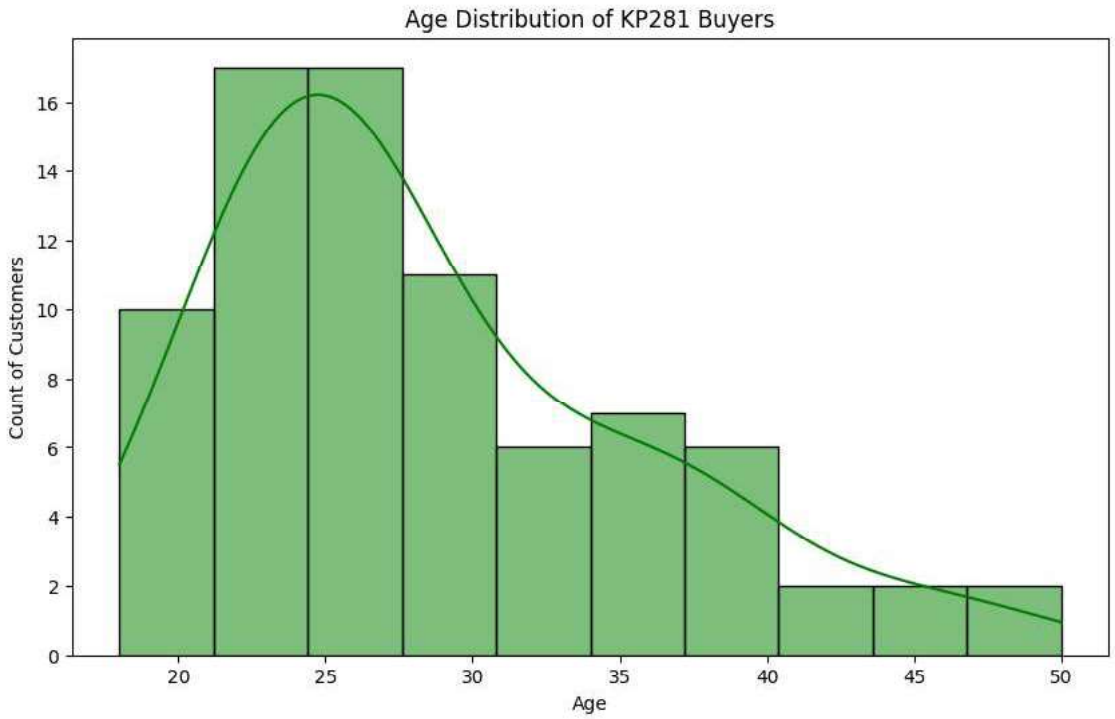
- Older users may cover fewer miles, while younger users tend to log longer distances.
- Suggests that younger individuals are more engaged in high-endurance treadmill workouts.

8.CUSTOMER PROFILING

A. Customer Profiling for KP281 Buyers To determine who buys KP281, we'll analyze the dataset based on:

- Age group
- Gender distribution
- Income range

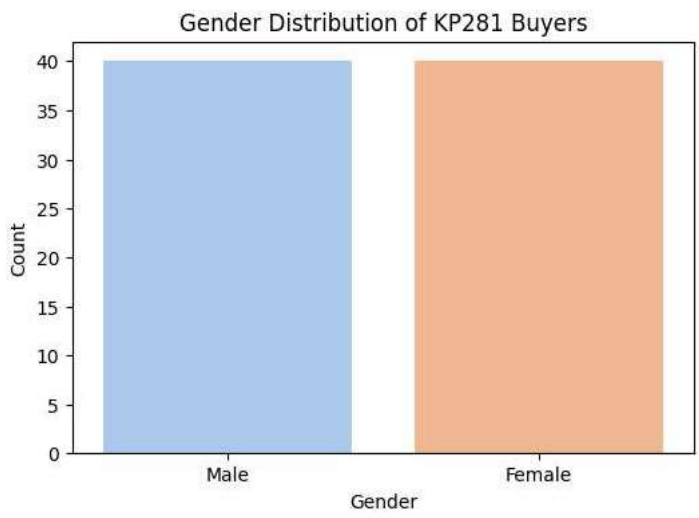
```
#1. Age Distribution for KP281
plt.figure(figsize=(10,6))
sns.histplot(df[df["Product"] == "KP281"]["Age"], bins=10, kde=True, color="green")
plt.xlabel("Age")
plt.ylabel("Count of Customers")
plt.title("Age Distribution of KP281 Buyers")
plt.show()
```



Expected Findings:

- KP281 is likely bought by younger individuals, possibly aged 18–35.
- Peak purchasing age may be around 25 years, if many buyers fall within this range.

```
#2.Gender Breakdown for KP281
plt.figure(figsize=(6,4))
sns.countplot(x=df[df["Product"] == "KP281"]["Gender"], palette="pastel")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.title("Gender Distribution of KP281 Buyers")
plt.show()
```

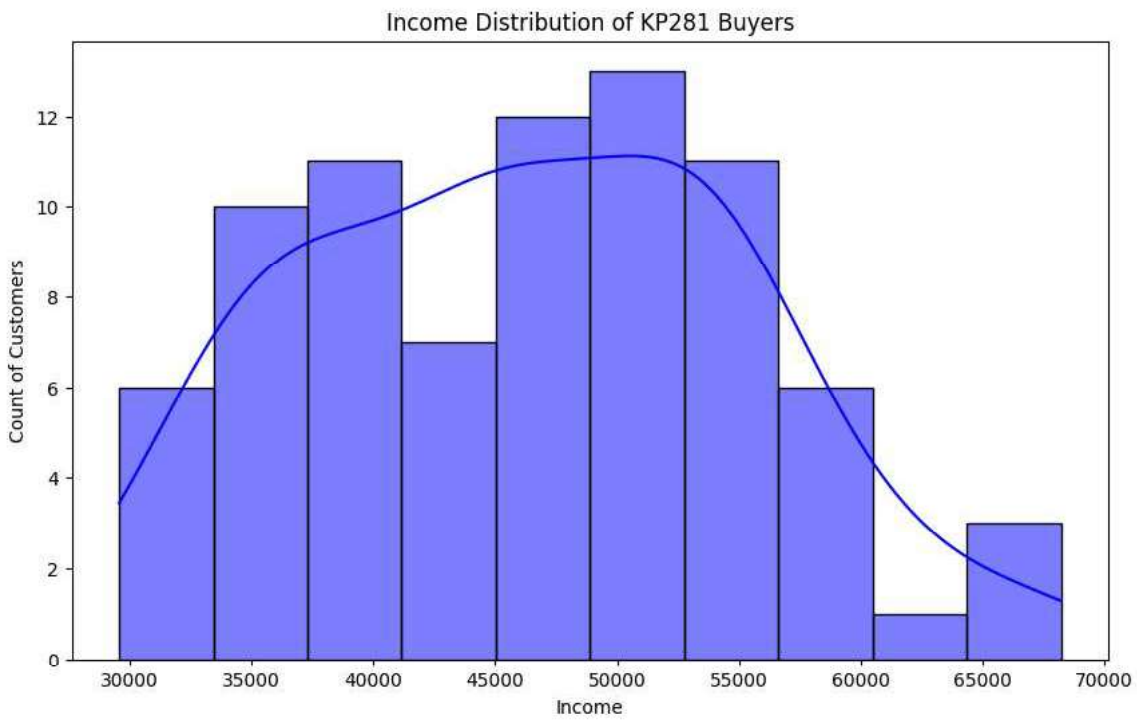


Expected Findings:

- KP281 may have balanced gender distribution, or it may skew toward male/female users.

```
#3.Income Range for KP281 Buyers
plt.figure(figsize=(10,6))
sns.histplot(df[df["Product"] == "KP281"]["Income"], bins=10, kde=True, color="blue")
plt.xlabel("Income")
plt.ylabel("Count of Customers")
```

```
plt.title("Income Distribution of KP281 Buyers")
plt.show()
```



Expected Findings:

- KP281 buyers likely belong to a mid-income group.
- If the average income is below ₹60,000, this suggests KP281 is affordable & popular among budget-conscious buyers.

Final Customer Profile for KP281 Typical KP281 Buyer:

Age: Likely between 18–35 years

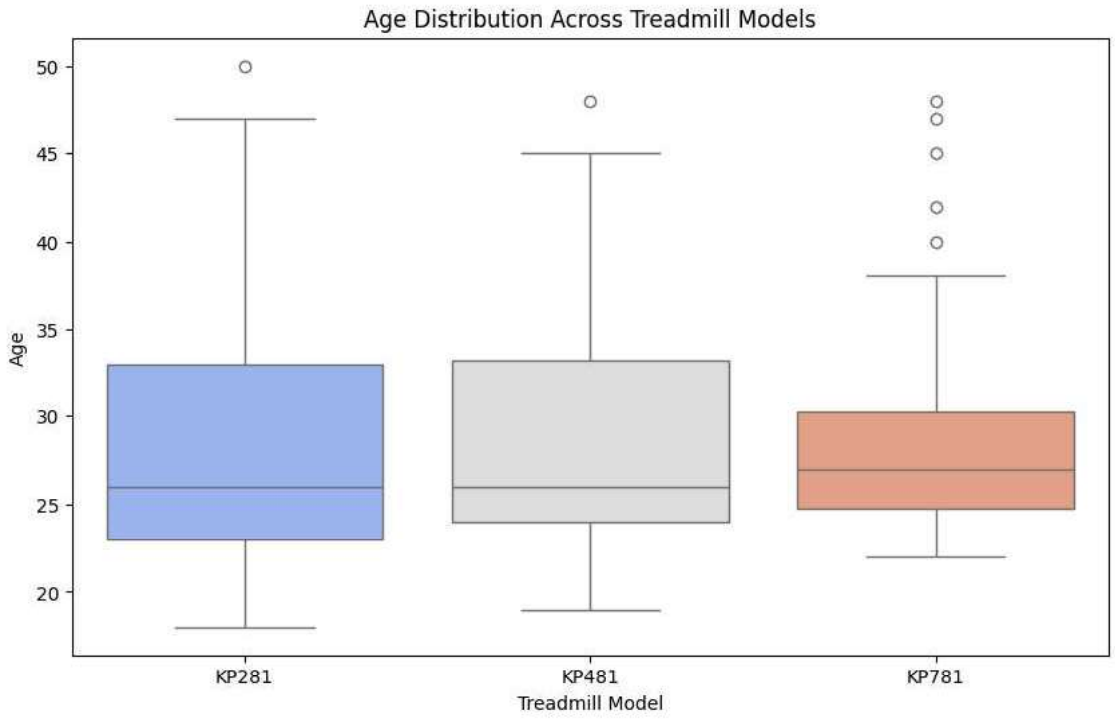
Gender: Possibly balanced, but requires confirmation

Income: Falls within mid-income category (~₹40,000–₹60,000)

B. Comparing KP281 Buyers with KP481 & KP781 Buyers

Now that we have customer profiling for KP281, let's compare it with KP481 & KP781 buyers to see how preferences differ.

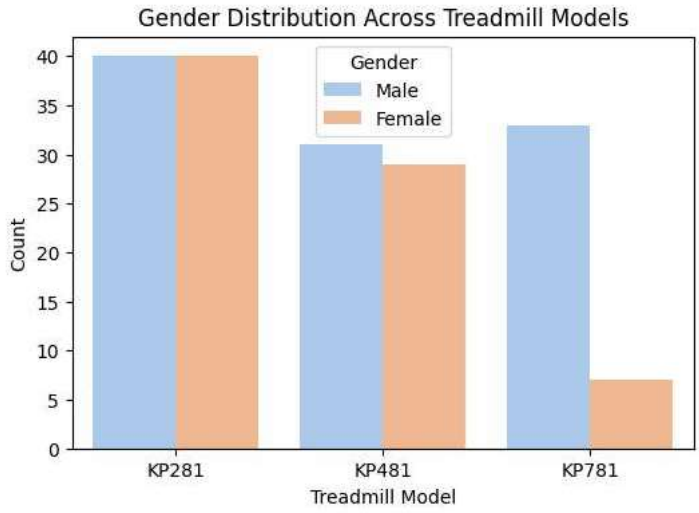
```
#1. Age Comparison Across Products
plt.figure(figsize=(10,6))
sns.boxplot(x=df["Product"], y=df["Age"], palette="coolwarm")
plt.xlabel("Treadmill Model")
plt.ylabel("Age")
plt.title("Age Distribution Across Treadmill Models")
plt.show()
```



Expected Findings:

- KP281 buyers tend to be younger (18–35 years old).
- KP781 may attract younger fitness enthusiasts, while KP481 may have a more diverse age range.

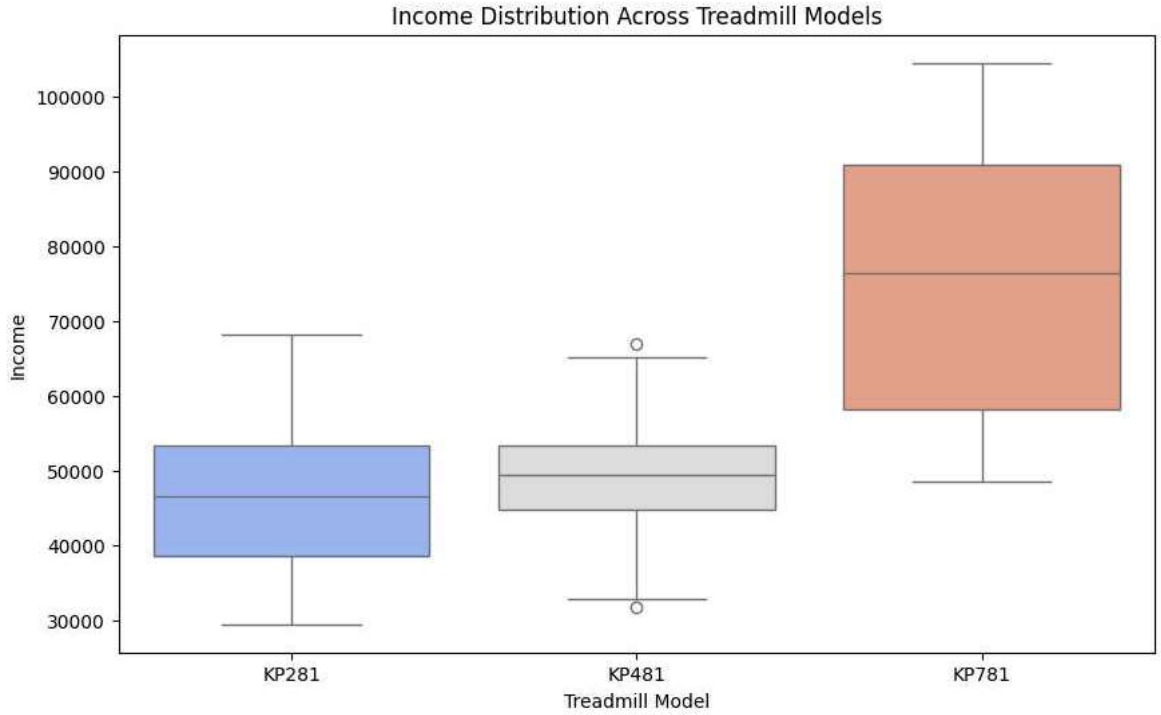
```
#2.Gender Comparison Across Products
plt.figure(figsize=(6,4))
sns.countplot(x=df["Product"], hue=df["Gender"], palette="pastel")
plt.xlabel("Treadmill Model")
plt.ylabel("Count")
plt.title("Gender Distribution Across Treadmill Models")
plt.legend(title="Gender")
plt.show()
```



Expected Findings:

- KP281 may have balanced gender distribution, or it could lean towards one gender.
- KP781 might be more popular among male users due to intensive usage patterns.
- KP481 may have varied gender engagement, indicating a general treadmill model.

```
#3.Income Comparison Across Products
plt.figure(figsize=(10,6))
sns.boxplot(x=df["Product"], y=df["Income"], palette="coolwarm")
plt.xlabel("Treadmill Model")
plt.ylabel("Income")
plt.title("Income Distribution Across Treadmill Models")
plt.show()
```



Expected Findings:

- KP281 buyers likely fall within mid-income (₹40,000–₹60,000).
- KP781 buyers may belong to a higher-income category (₹70,000+), showing preference for premium treadmills.
- KP481 may have a mixed income range, indicating balanced affordability.

Final Recommendations

KP281 is best suited for young buyers (18–35) in the mid-income range who prefer moderate treadmill usage.

KP481 is ideal for a mixed demographic, catering to mid-income groups with varied workout habits.

KP781 is designed for higher-income, fitness-driven users who cover more miles and require intensive workouts.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.