Aerofit Case Study

Exploratory data analysis (EDA) is an approach to:

-analyzing and understanding data that is focused on finding patterns and relationships in the data, rather than on testing hypotheses or making predictions

-visualizing the data and looking for trends, patterns, and anomalies, as well as summarizing the main characteristics of the data.

EDA is an important step in the data science process because it helps you get to know your data, identify any problems or issues with the data, and formulate hypotheses for further analysis.

Business Problem: 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.

Approach: We are doing EDA on AeroFit data, employing various Python libraries (NumPy, Pandas, Matplotlib, Seaborn), and computing the required Marginal and Condition Probabilities, with a primary objective to create a customer profile for each product, unearth valuable insights and offer actionable recommendations.

Collab link: https://colab.research.google.com/drive/1zOxpt9wLBOhcDYGdUTdEJ8Sc1msVybkP?usp=sharing

Exploratory data analysis (EDA)

Exploring the data

-df.head()

We can check the different attributes present in our dataset, and the type of the data for each column.

	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

-df.info()

Gives the idea on number of rows present in the dataset, data type of each column, and number non null values present.

```
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
```

-netflix.describe(include='object').T

Returns the descriptive statistics summary of the dataframe.

	count	unique	top	freq
Product	180	3	KP281	80
Gender	180	2	Male	104
MaritalStatus	180	2	Partnered	107

-df.describe(include='number').T

	count	mean	std	min	25%	50%	75%	max
Age	180.0	28.788889	6.943498	18.0	24.00	26.0	33.00	50.0
Education	180.0	15.572222	1.617055	12.0	14.00	16.0	16.00	21.0
Usage	180.0	3.455556	1.084797	2.0	3.00	3.0	4.00	7.0
Fitness	180.0	3.311111	0.958869	1.0	3.00	3.0	4.00	5.0
Income	180.0	53719.577778	16506.684226	29562.0	44058.75	50596.5	58668.00	104581.0
Miles	180.0	103.194444	51.863605	21.0	66.00	94.0	114.75	360.0

-df.shape

Returns shape of the dataframe.

(180,9)

-Checking for missing values in each column.

```
#we have no Null values
df.isnull().sum()
```

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

On exploring the dataset, we see the following observations:

1. Missing values: There are no missing values in any of the columns.

2. There are 3 unique products in the dataset.

3. Standard deviation for Income & Miles is very high. These variables might have the

outliers in it.

Non-Graphical Analysis:

We check the value counts for each of the categorical attributes:

-Product

```
df['Product'].value_counts()
```

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

```
# Percentage Distribution of Products
df['Product'].value_counts(normalize=True)*100.0
```

```
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: Product, dtype: float64
```

-Gender

```
df['Gender'].value_counts()
```

```
Male      104
Female     76
Name: Gender, dtype: int64
```

-Marital Status

```
df['MaritalStatus'].value_counts()
```

```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

Observations:

1. KP281 is the most frequent product.
2. Most of our customers are Male.
3. Most of our customers are Partnered.

We will create bucket on other continuous variable, to help us categorize the customer with respect to the product.

```
>df['AgeBucket']=pd.cut(df['Age'],bins=[15,25,35,45,55],labels=['Young Adults','Old Adults','Middle Age','Old'])

>df['EducationLevel']=pd.cut(df['Education'],bins=[10,14,17,22],labels=['Basic','Intermediate','Advance'])

>df['UsageLevel']=pd.cut(df['Usage'],bins=[1,3,5,7],labels=['Low','Moderate','High'])

>df['FitnessLevel']=pd.cut(df['Fitness'],bins=[0,2,4,5],labels=['Begineer','Advance','Elite'])

>df['IncomeLevel']=pd.cut(df['Income'],bins=[25000,45000,60000,80000,150000],labels=['Low','Moderate','High','Very High'])
```

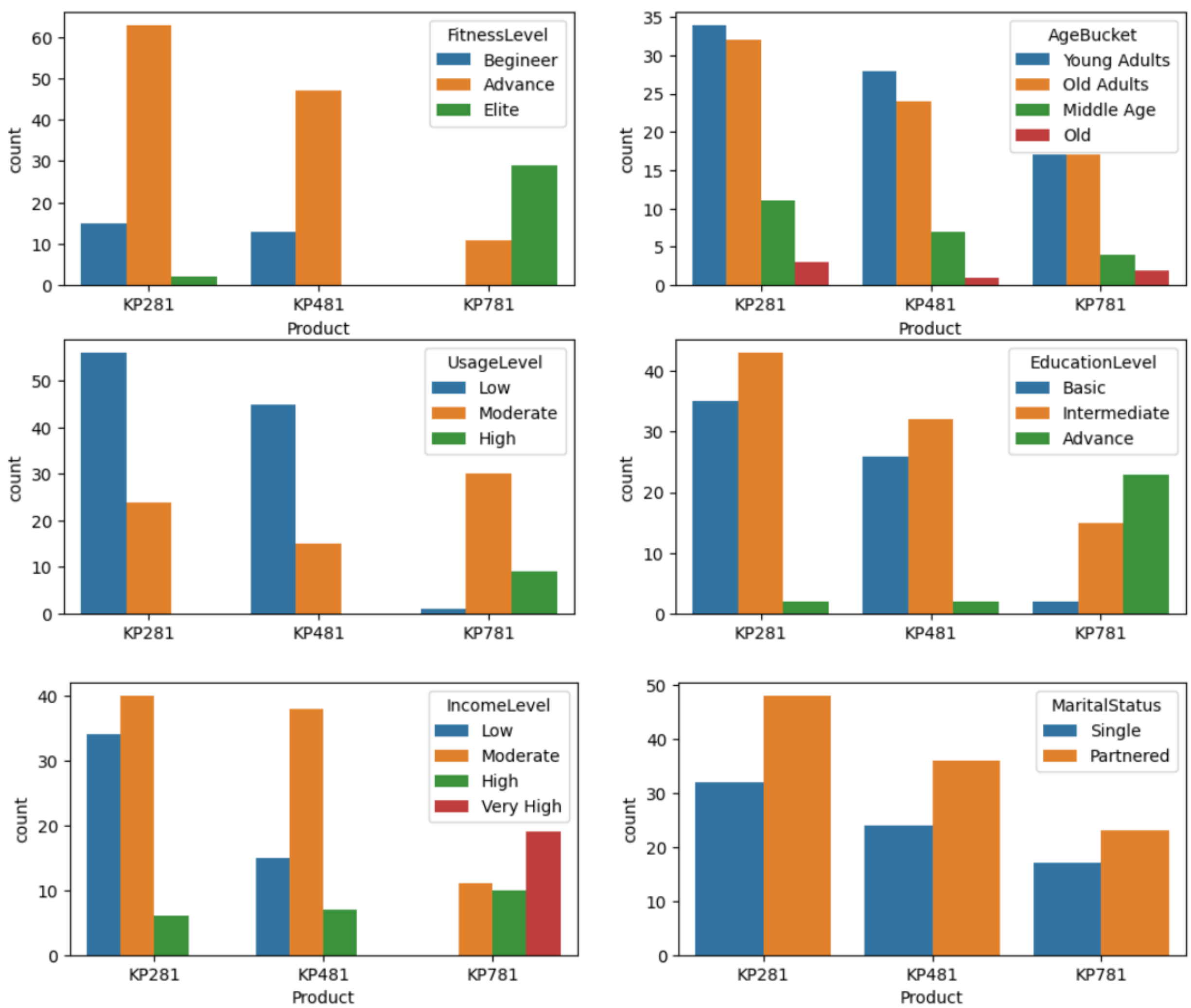
Our Dataset after creating different buckets looks like:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeBucket	EducationLevel	UsageLevel	FitnessLevel	IncomeLevel
0	KP281	18	Male	14	Single	3	4	29562	112	Young Adults	Basic	Low	Advance	Low
1	KP281	19	Male	15	Single	2	3	31836	75	Young Adults	Intermediate	Low	Advance	Low
2	KP281	19	Female	14	Partnered	4	3	30699	66	Young Adults	Basic	Moderate	Advance	Low
3	KP281	19	Male	12	Single	3	3	32973	85	Young Adults	Basic	Low	Advance	Low
4	KP281	20	Male	13	Partnered	4	2	35247	47	Young Adults	Basic	Moderate	Begineer	Low

Graphical Analysis:

We try to plot some graphs to gain some insights.

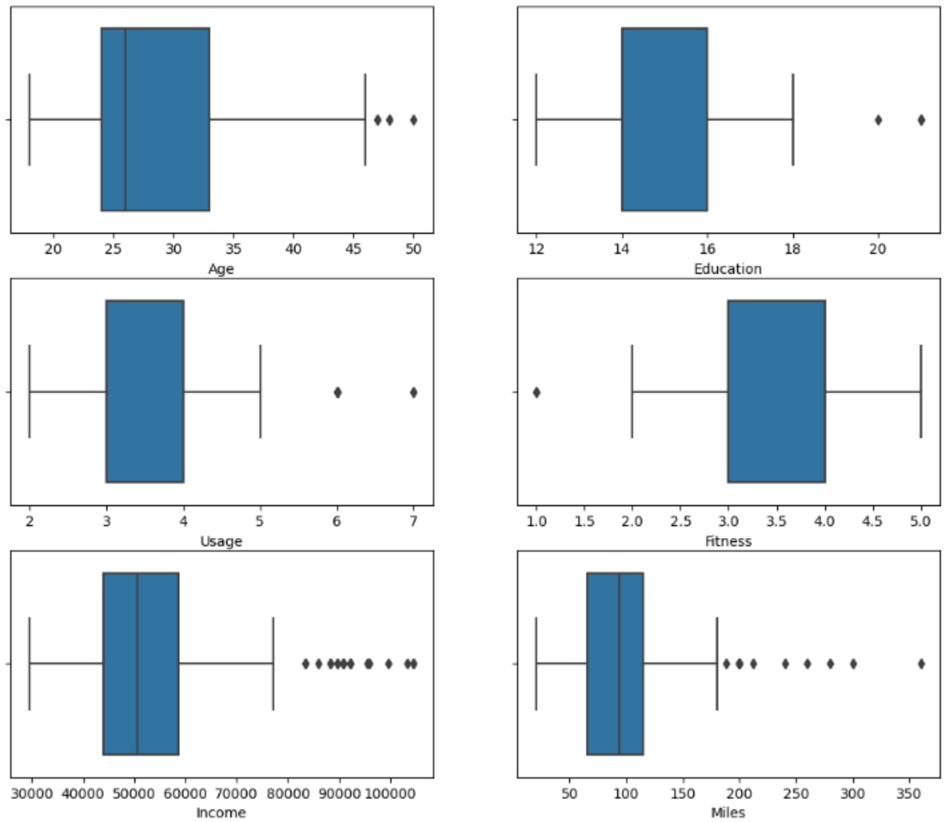
- Distribution of Product for different categories



Observations:

- Only the Elite level and handful of Advanced customers prefer the expensive Treadmill(KP781), else other customers prefer the lesser expensive Models(KP281, KP481)
- We see a similar distribution for all three products with respect to age group
- Only the customers with High-Moderate usage expectations prefer the expensive Treadmill(KP781), else other customers prefer the lesser expensive Models(KP281, KP481)
- Only Customers with Advance level of education prefer the expensive Treadmill(KP781), we see similar distribution for KP281 & KP481
- Customers with very high Income only purchase the KP781, we see similar distribution for KP281 & KP481
- We see a similar distribution for all three products with respect to Marital Status.

➢ We will check for outliers using Box Plot



Observation:

- We see a lot of outliers in Income and Miles.
- Not many outliers in other variables.

We will create Contingency Tables to compute Marginal and Conditional Probability

```
pd.crosstab(df['Gender'],df['Product'],margins='All')
```

Product	KP281	KP481	KP781	All
Gender				
Female	40	29	7	76
Male	40	31	33	104
All	80	60	40	180

```
pd.crosstab(df['Gender'],df['Product'],normalize=True,margins=True)
```

Product	KP281	KP481	KP781	All
Gender				
Female	0.222222	0.161111	0.038889	0.422222
Male	0.222222	0.172222	0.183333	0.577778
All	0.444444	0.333333	0.222222	1.000000

```python
pd.crosstab(df['Gender'],df['Product'],normalize='index',margins=True)
```

Product	KP281	KP481	KP781
Gender			
Female	0.526316	0.381579	0.092105
Male	0.384615	0.298077	0.317308
All	0.444444	0.333333	0.222222

```python
pd.crosstab([df['Gender'],df['MaritalStatus']],df['Product'],margins=True,normalize='index')
```

	Product	KP281	KP481	KP781
Gender	**MaritalStatus**			
Female	**Partnered**	0.586957	0.326087	0.086957
	Single	0.433333	0.466667	0.100000
Male	**Partnered**	0.344262	0.344262	0.311475
	Single	0.441860	0.232558	0.325581
All		0.444444	0.333333	0.222222

```python
pd.crosstab([df['AgeBucket'],df['FitnessLevel']],df['Product'],margins=True,normalize='index')
```

	Product	KP281	KP481	KP781
AgeBucket	**FitnessLevel**			
Young Adults	**Begineer**	0.500000	0.500000	0.000000
	Advance	0.500000	0.403846	0.096154
	Elite	0.076923	0.000000	0.923077
Old Adults	**Begineer**	0.583333	0.416667	0.000000
	Advance	0.500000	0.395833	0.104167
	Elite	0.076923	0.000000	0.923077
Middle Age	**Begineer**	0.000000	1.000000	0.000000
	Advance	0.611111	0.333333	0.055556
	Elite	0.000000	0.000000	1.000000
Old	**Begineer**	1.000000	0.000000	0.000000
	Advance	0.666667	0.333333	0.000000
	Elite	0.000000	0.000000	1.000000
All		0.444444	0.333333	0.222222

```python
pd.crosstab([df['IncomeLevel'],df['UsageLevel']],df['Product'],margins=True,normalize='index')
```

IncomeLevel	UsageLevel	KP281	KP481	KP781
Low	Low	0.611111	0.388889	0.000000
	Moderate	0.923077	0.076923	0.000000
Moderate	Low	0.500000	0.482143	0.017857
	Moderate	0.363636	0.333333	0.303030
High	Low	0.600000	0.400000	0.000000
	Moderate	0.000000	0.300000	0.700000
	High	0.000000	0.000000	1.000000
Very High	Moderate	0.000000	0.000000	1.000000
	High	0.000000	0.000000	1.000000
All		0.444444	0.333333	0.222222

```python
pd.crosstab([df['IncomeLevel'],df['FitnessLevel']],df['Product'],margins=True,normalize='index')
```

IncomeLevel	FitnessLevel	KP281	KP481	KP781
Low	Begineer	0.545455	0.454545	0.000000
	Advance	0.729730	0.270270	0.000000
	Elite	1.000000	0.000000	0.000000
Moderate	Begineer	0.533333	0.466667	0.000000
	Advance	0.484375	0.484375	0.031250
	Elite	0.100000	0.000000	0.900000
High	Begineer	0.500000	0.500000	0.000000
	Advance	0.312500	0.375000	0.312500
	Elite	0.000000	0.000000	1.000000
Very High	Advance	0.000000	0.000000	1.000000
	Elite	0.000000	0.000000	1.000000
All		0.444444	0.333333	0.222222

```
pd.crosstab([df['FitnessLevel'],df['UsageLevel']],df['Product'],margins=True,normalize='index')
```

FitnessLevel	UsageLevel	KP281	KP481	KP781
Begineer	**Low**	0.538462	0.461538	0.000000
	Moderate	0.500000	0.500000	0.000000
Advance	**Low**	0.560000	0.440000	0.000000
	Moderate	0.466667	0.311111	0.222222
	High	0.000000	0.000000	1.000000
Elite	**Low**	0.000000	0.000000	1.000000
	Moderate	0.090909	0.000000	0.909091
	High	0.000000	0.000000	1.000000
All		0.444444	0.333333	0.222222

Observations:

1.We can see from the crosstabs Probability of which treadmill the customer buys is highly dependent on the following factors- Income, Fitness Level and Usage Level

2.We cannot infer much insight from probability distribution of Gender, Marital Status and Age.

Recommendations:

1. To recommend the correct Treadmill to the customer, we should gather at least 2 of the following attributes.

- Income
- Fitness Level
- Usage Level

2. Knowing the above features of a customer, helps us in targeting the right customer cohort.