Waze Project

Milestone 2 / 2a - Compile information about the data. Begin exploring the data.

Inspect and analyze data

The purpose of this project is to investigate and understand the data provided.

The goal is to use a dataframe contructed within Python to perform a cursory inspection of the provided dataset.

This notebook has two parts:

Part 1: Summary Information

Part 2: Initial Churned vs. Retained exploration

Identify data types and compile summary information

Imports and data loading

In [1]:
```
# Import packages for data manipulation
import pandas as pd
import numpy as np
```

In [2]:
```
# Load dataset into dataframe
df = pd.read_csv('waze_dataset.csv')
```

Summary information

In [3]:
```
df.head(10)
```

Out[3]:

	ID	label	sessions	drives	total_sessions	n_days_after_onboarding	total_navigations_fav1	total_navigati
0	0	retained	283	226	296.748273	2276	208	
1	1	retained	133	107	326.896596	1225	19	
2	2	retained	114	95	135.522926	2651	0	
3	3	retained	49	40	67.589221	15	322	
4	4	retained	84	68	168.247020	1562	166	
5	5	retained	113	103	279.544437	2637	0	
6	6	retained	3	2	236.725314	360	185	
7	7	retained	39	35	176.072845	2999	0	
8	8	retained	57	46	183.532018	424	0	
9	9	churned	84	68	244.802115	2997	72	

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   ID                       14999 non-null  int64  
 1   label                    14299 non-null  object 
 2   sessions                 14999 non-null  int64  
 3   drives                   14999 non-null  int64  
 4   total_sessions           14999 non-null  float64
 5   n_days_after_onboarding  14999 non-null  int64  
 6   total_navigations_fav1   14999 non-null  int64  
 7   total_navigations_fav2   14999 non-null  int64  
 8   driven_km_drives         14999 non-null  float64
 9   duration_minutes_drives  14999 non-null  float64
 10  activity_days            14999 non-null  int64  
 11  driving_days             14999 non-null  int64  
 12  device                   14999 non-null  object 
dtypes: float64(3), int64(8), object(2)
memory usage: 1.5+ MB
```

Null values and summary statistics

In [5]:
```
# Isolate rows with null values
null_df = df[df['label'].isnull()]
# Display summary stats of rows with null values
null_df.describe()
```

Out[5]:

	ID	sessions	drives	total_sessions	n_days_after_onboarding	total_navigations_fav1	t
count	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	
mean	7405.584286	80.837143	67.798571	198.483348	1709.295714	118.717143	
std	4306.900234	79.987440	65.271926	140.561715	1005.306562	156.308140	
min	77.000000	0.000000	0.000000	5.582648	16.000000	0.000000	
25%	3744.500000	23.000000	20.000000	94.056340	869.000000	4.000000	
50%	7443.000000	56.000000	47.500000	177.255925	1650.500000	62.500000	
75%	11007.000000	112.250000	94.000000	266.058022	2508.750000	169.250000	
max	14993.000000	556.000000	445.000000	1076.879741	3498.000000	1096.000000	

In [6]:
```
# Isolate rows without null values
not_null_df = df[~df['label'].isnull()]
# Display summary stats of rows without null values
not_null_df.describe()
```

Out[6]:

	ID	sessions	drives	total_sessions	n_days_after_onboarding	total_navigations_fav
count	14299.000000	14299.000000	14299.000000	14299.000000	14299.000000	14299.00000
mean	7503.573117	80.623820	67.255822	189.547409	1751.822505	121.7473
std	4331.207621	80.736502	65.947295	136.189764	1008.663834	147.7134
min	0.000000	0.000000	0.000000	0.220211	4.000000	0.0000
25%	3749.500000	23.000000	20.000000	90.457733	878.500000	10.0000
50%	7504.000000	56.000000	48.000000	158.718571	1749.000000	71.0000
75%	11257.500000	111.000000	93.000000	253.540450	2627.500000	178.0000

max	14998.000000	743.000000	596.000000	1216.154633	3500.000000	1236.00000

Null values - device counts

In [7]:
```
# Get count of null values by device
null_df['device'].value_counts()
```

Out[7]:
```
iPhone     447
Android    253
Name: device, dtype: int64
```

Of the 700 rows with null values, 447 were iPhone users and 253 were Android users.

In [8]:
```
# Calculate % of iPhone nulls and Android nulls
null_df['device'].value_counts(normalize=True)
```

Out[8]:
```
iPhone     0.638571
Android    0.361429
Name: device, dtype: float64
```

In [9]:
```
# Calculate % of iPhone users and Android users in full dataset
df['device'].value_counts(normalize=True)
```

Out[9]:
```
iPhone     0.644843
Android    0.355157
Name: device, dtype: float64
```

The distribution of missing values across different devices aligns with their overall presence in the data, suggesting no indication of a systematic reason behind the missing data.

Churned vs. Retained

In [10]:
```
# Calculate counts of churned vs. retained
print(df['label'].value_counts())
print()
print(df['label'].value_counts(normalize=True))
```

```
retained    11763
churned      2536
Name: label, dtype: int64

retained    0.822645
churned     0.177355
Name: label, dtype: float64
```

This dataset contains approximately 82% retained users and 18% churned users.

In [11]:
```
# Calculate median values of all columns for churned and retained users
df.groupby('label').median(numeric_only=True)
```

Out[11]:

	ID	sessions	drives	total_sessions	n_days_after_onboarding	total_navigations_fav1	total_naviga
label							
churned	7477.5	59.0	50.0	164.339042	1321.0	84.5	
retained	7509.0	56.0	47.0	157.586756	1843.0	68.0	

A few interesting observations jump out from this quick comparions.

Churned users averaged significantly fewer activity days and driving days than the retained users, yet they also averaged slightly more drives, kms driven, and minutes driven.

Churned vs. Retained - drive comparisons

In [12]:
```
# Group data by `label` and calculate the medians
medians_by_label = df.groupby('label').median(numeric_only=True)
print('Median kilometers per drive:')
# Divide the median distance by median number of drives
medians_by_label['driven_km_drives'] / medians_by_label['drives']
```

Out[12]:
```
Median kilometers per drive:
label
churned     73.053113
retained    73.716694
dtype: float64
```

There is not a significant difference between churned and retained median kilometers per dri. They both averaged ~73 km/drive. How many kilometers per driving day was this?

In [13]:
```
# Divide the median distance by median number of driving days
print('Median kilometers per driving day:')
medians_by_label['driven_km_drives'] / medians_by_label['driving_days']
```

Out[13]:
```
Median kilometers per driving day:
label
churned     608.775944
retained    247.477472
dtype: float64
```

Calculate the median number of drives per driving day for each group.

In [14]:
```
# Divide the median number of drives by median number of driving days
print('Median drives per driving day:')
medians_by_label['drives'] / medians_by_label['driving_days']
```

Out[14]:
```
Median drives per driving day:
label
churned     8.333333
retained    3.357143
dtype: float64
```

The median churned user traveled an average of 608 kilometers per driving day last month, which is nearly 2.5 times the distance covered by retained users on each drive day. Additionally, the median churned user had a disproportionately higher number of drives per drive day compared to retained users.

Churned vs. Retained - device type comparison

In [15]:
```
# For each label, calculate the number of Android users and iPhone users
df.groupby(['label', 'device']).size()
```

Out[15]:
```
label     device
churned   Android     891
          iPhone     1645
retained  Android    4183
          iPhone     7580
dtype: int64
```

In [16]:
```
# For each label, calculate the percentage of Android users and iPhone users
df.groupby('label')['device'].value_counts(normalize=True)
```

Out[16]:
```
label     device
churned   iPhone     0.648659
          Android    0.351341
retained  iPhone     0.644393
          Android    0.355607
Name: device, dtype: float64
```

The proportion of iPhone users to Android users remains consistent within both the churned and retained groups, and these ratios align with the overall dataset.

Conclusion

- **The dataset contains 700 missing values, and there is no discernible pattern to these missing values.**
- **Within the dataset, around 36% of the users were Android users, whereas approximately 64% were iPhone users.**
- **The median churned user traveled an average of 608 kilometers per driving day last month, which is nearly 2.5 times the distance covered by retained users on each drive day.**
- **The median churned user had a disproportionately higher number of drives per drive day compared to retained users.**
- **In general, churned users covered similar distances but had longer durations of driving within a shorter span of days compared to retained users.**
- **Churned users utilized the app approximately half as frequently as retained users during the same time frame**
- **Churn rate for both iPhone and Android users differed by less than one percentage point. There is no indication of any correlation between device type and churn, suggesting that device choice does not play a significant role in the churn rate.**