Waze Project

Course 2 - Get Started with Python

Welcome to the Waze Project!

Your Waze data analytics team is still in the early stages of their user churn project. Previously, you were asked to complete a project proposal by your supervisor, May Santner. You have received notice that your project proposal has been approved and that your team has been given access to Waze's user data. To get clear insights, the user data must be inspected and prepared for the upcoming process of exploratory data analysis (EDA).

A Python notebook has been prepared to guide you through this project. Answer the questions and create an executive summary for the Waze data team.

Course 2 End-of-course project: Inspect and analyze data

In this activity, you will examine data provided and prepare it for analysis. This activity will help ensure the information is,

1. Ready to answer questions and yield insights
2. Ready for visualizations
3. Ready for future hypothesis testing and statistical methods

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

The goal is to use a dataframe contructed within Python, perform a cursory inspection of the provided dataset, and inform team members of your findings.

This activity has three parts:

Part 1: Understand the situation

- How can you best prepare to understand and organize the provided information?

Part 2: Understand the data

- Create a pandas dataframe for data learning, future exploratory data analysis (EDA), and statistical activities
- Compile summary information about the data to inform next steps

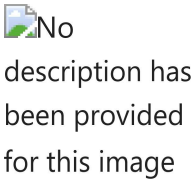
Part 3: Understand the variables

- Use insights from your examination of the summary data to guide deeper investigation into variables

Follow the instructions and answer the following questions to complete the activity. Then, you will complete an Executive Summary using the questions listed on the PACE Strategy Document.

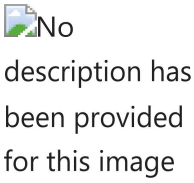
Be sure to complete this activity before moving on. The next course item will provide you with a completed exemplar to compare to your own work.

Identify data types and compile summary information



PACE stages

Throughout these project notebooks, you'll see references to the problem-solving framework, PACE. The following notebook components are labeled with the respective PACE stages: Plan, Analyze, Construct, and Execute.



PACE: Plan

Consider the questions in your PACE Strategy Document and those below to craft your response:

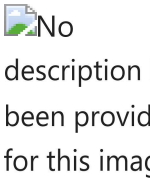
Task 1. Understand the situation

- How can you best prepare to understand and organize the provided driver data?

Begin by exploring your dataset and consider reviewing the Data Dictionary.

==> ENTER YOUR RESPONSE HERE

PACE: Analyze



Consider the questions in your PACE Strategy Document to reflect on the Analyze stage.

Task 2a. Imports and data loading

Start by importing the packages that you will need to load and explore the dataset. Make sure to use the following import statements:

- `import pandas as pd`
- `import numpy as np`

In [1]:
```
# Import packages for data manipulation

import pandas as pd
import numpy as np
```

Then, load the dataset into a dataframe. Creating a dataframe will help you conduct data manipulation, exploratory data analysis (EDA), and statistical activities.

Note: As shown in this cell, the dataset has been automatically loaded in for you. You do not need to download the .csv file, or provide more code, in order to access the dataset and proceed with this lab. Please continue with this activity by completing the following instructions.

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

Task 2b. Summary information

View and inspect summary information about the dataframe by **coding the following:**

1. df.head(10)
2. df.info()

Consider the following questions:

1. When reviewing the `df.head()` output, are there any variables that have missing values?
2. When reviewing the `df.info()` output, what are the data types? How many rows and columns do you have?
3. Does the dataset have any missing values?

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

Out[3]:

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

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
```

When reviewing the df.head() output, are there any variables that have missing values?

When reviewing the df.info() output, what are the data types? How many rows and columns do you have?

Does the dataset have any missing values?

Task 2c. Null values and summary statistics

Compare the summary statistics of the 700 rows that are missing labels with summary statistics of the rows that are not missing any values.

Question: Is there a discernible difference between the two populations?

In [5]:
```
# Isolate rows with null values
mask = df.isnull().any(axis=1)

# Display summary stats of rows with null values
df[mask].describe()
```

Out[5]:

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

In [6]:
```
# Isolate rows without null values
mask = df.notnull().all(axis=1)

# Display summary stats of rows without null values
df[mask].describe()
```

Out[6]:

	ID	sessions	drives	total_sessions	n_days_after_onboarding	t
count	14299.000000	14299.000000	14299.000000	14299.000000	14299.000000	
mean	7503.573117	80.623820	67.255822	189.547409	1751.822505	
std	4331.207621	80.736502	65.947295	136.189764	1008.663834	
min	0.000000	0.000000	0.000000	0.220211	4.000000	
25%	3749.500000	23.000000	20.000000	90.457733	878.500000	
50%	7504.000000	56.000000	48.000000	158.718571	1749.000000	
75%	11257.500000	111.000000	93.000000	253.540450	2627.500000	
max	14998.000000	743.000000	596.000000	1216.154633	3500.000000	

the comparison of the summary statistics for both groups doesn't reveal anything outstanding. Values like the mean and standard deviation on both groups is consistent

Task 2d. Null values - device counts

Next, check the two populations with respect to the `device` variable.

Question: How many iPhone users had null values and how many Android users had null values?

In [7]:
```
# Get count of null values by device
mask = df.isnull().any(axis=1)
grouped = df[mask].groupby('device')
grouped.count()
```

Out[7]:

device	ID	label	sessions	drives	total_sessions	n_days_after_onboarding	total_navigat
Android	253	0	253	253	253	253	
iPhone	447	0	447	447	447	447	

On rows that have null values, there are 447 iphone users versus 253 Android users

Now, of the rows with null values, calculate the percentage with each device—Android and iPhone. You can do this directly with the `value_counts()` function.

In [8]:
```
# Calculate % of iPhone nulls and Android nulls
device_null_counts = df[mask]['device'].value_counts(normalize=True) * 100
print(device_null_counts)
```

```
iPhone     63.857143
Android    36.142857
Name: device, dtype: float64
```

How does this compare to the device ratio in the full dataset?

In [9]:
```
# Calculate % of iPhone users and Android users in full dataset
device_all_dataset_counts = df['device'].value_counts(normalize=True) * 100
print(device_all_dataset_counts)
```

```
iPhone     64.484299
Android    35.515701
Name: device, dtype: float64
```

The percentage of missing values by each device is consistent with their representation in the data overall. The rows with some missing values are similar to the full dataset

There is nothing to suggest a non-random or systemic cause of the missing data.

Examine the counts and percentages of users who churned vs. those who were retained. How many of each group are represented in the data?

In [10]:
```
# Calculate counts of churned vs. retained
df.groupby('label').count()
```

Out[10]:

label	ID	sessions	drives	total_sessions	n_days_after_onboarding	total_navigations
churned	2536	2536	2536	2536	2536	
retained	11763	11763	11763	11763	11763	

In [11]:
```
label_all_dataset_counts = df['label'].value_counts(normalize=True) * 100
print(label_all_dataset_counts)
```

```
retained    82.264494
churned     17.735506
Name: label, dtype: float64
```

*This dataset contains 82% of retained users and 18% of churned users.

Next, compare the medians of each variable for churned and retained users. The reason for calculating the median and not the mean is that you don't want outliers to unduly affect the portrayal of a typical user. Notice, for example, that the maximum value in the `driven_km_drives` column is 21,183 km. That's more than half the circumference of the earth!

In [12]:
```
# Calculate median values of all columns for churned and retained users
median_values = df.groupby('label').median()
print(median_values)
```

```
              ID  sessions  drives  total_sessions  n_days_after_onboarding  \
label
churned   7477.5      59.0    50.0      164.339042                   1321.0
retained  7509.0      56.0    47.0      157.586756                   1843.0

          total_navigations_fav1  total_navigations_fav2  driven_km_drives  \
label
churned                     84.5                    11.0       3652.655666
retained                    68.0                     9.0       3464.684614

          duration_minutes_drives  activity_days  driving_days
label
churned                1607.183785           8.0           6.0
retained               1458.046141          17.0          14.0
```

This grouping shows an interesting snapshot of the two groups, churned vs. retained:

Users who churned averaged ~3 more drives in the last month than retained users, but retained users used the app on over twice as many days as churned users in the same time period.

The median churned user drove ~200 more kilometers and 2.5 more hours during the last month than the median retained user.

It seems that churned users had more drives in fewer days, and their trips were farther and longer in duration. Perhaps this is suggestive of a particular user profile. Additional exploration is necessary

Calculate the median kilometers per drive in the last month for both retained and churned users.

Begin by dividing the `driven_km_drives` column by the `drives` column. Then, group the results by churned/retained and calculate the median km/drive of each group.

In [14]:
```
# Add a column to df called `km_per_drive`
df['km_per_drive'] = df['driven_km_drives']/df['drives']

# Group by `label`, calculate the median, and isolate for km per drive
median_values_km_per_drive = df.groupby('label')['kilometers_per_drive'].median()
print(median_values_km_per_drive)
```

```
label
churned     74.109416
retained    75.014702
Name: kilometers_per_drive, dtype: float64
```

The median retained user drove about one more kilometer per drive than the median churned user. How many kilometers per driving day was this?

This doesn't seem significant from a distance perspective, now let's find out how many kilometers per driving day this is

To calculate this statistic, repeat the steps above using `driving_days` instead of `drives`.

In [16]:
```
# Add a column to df called `km_per_driving_day`
df['km_per_driving_day'] = df['driven_km_drives']/df['driving_days']

# Group by `label`, calculate the median, and isolate for km per driving day
median_values_km_per_driving_day = df.groupby('label')['km_per_driving_day'].median
print(median_values_km_per_driving_day)
```

```
label
churned     697.541999
retained    289.549333
Name: km_per_driving_day, dtype: float64
```

This reveals something important, the median values of both groups are relatively high for a casual driver, it will be important to get the distribution of users on km_per_driving_day

the difference between the true groups is substantial churned users drove about 2.4 times more (240% more) kilometers than retained users, 697.5 vs 289.5. Im starting to think that for people that are constantly on the road, there is not much added value in the road conditions reports from waze

Now, calculate the median number of drives per driving day for each group.

In [17]:
```
# Add a column to df called `drives_per_driving_day`
df['drives_per_driving_day']  = df['drives']/df['driving_days']

# Group by `label`, calculate the median, and isolate for drives per driving day
median_drives_per_driving_day = df.groupby('label')['drives_per_driving_day'].media
print(median_drives_per_driving_day)
```

```
label
churned     10.0000
retained     4.0625
Name: drives_per_driving_day, dtype: float64
```

The median user who churned drove 698 kilometers each day that they drove last month, which is almost ~240% the per-drive-day distance of retained users. The median churned user had a similarly disproporionate number of drives per drive day compared to retained users.

It is clear from these figures that, regardless of whether a user churned or not, the users represented in this data are serious drivers! It would probably be safe to assume that this data does not represent typical drivers at large. Perhaps the data and in particular the sample of churned users contains a high proportion of long-haul truckers.

In consideration of how much these users drive, it would be worthwhile to recommend to Waze that they gather more data on these super-drivers. It's possible that the reason for their driving so much is also the reason why the Waze app does not add value to meet their specific set of needs, which may differ from the needs of a more typical driver, such as a commuter.

Finally, examine whether there is an imbalance in how many users churned by device type.

Begin by getting the overall counts of each device type for each group, churned and retained.

In [29]:
```
# For each label, calculate the number of Android users and iPhone users
df.groupby(['label','device'])['device'].count()
```

Out[29]:
```
label     device
churned   Android     891
          iPhone     1645
retained  Android    4183
          iPhone     7580
Name: device, dtype: int64
```

Now, within each group, churned and retained, calculate what percent was Android and what percent was iPhone.

In [37]:
```
# For each label, calculate the percentage of Android users and iPhone users
grouped_counts = df.groupby(['label', 'device'])['device'].count()
percentage_df = grouped_counts.groupby(level=0).apply(lambda x: 100 * x / float(x.s
print(percentage_df)
```

```
     label   device  percentage
0   churned  Android   35.134069
1   churned   iPhone   64.865931
2  retained  Android   35.560656
3  retained   iPhone   64.439344
```

The ratio of iPhone users and Android users is consistent between the churned group and the retained group, and those ratios are both consistent with the ratio found in the overall dataset.

PACE: Construct

Note: The Construct stage does not apply to this workflow. The PACE framework can be adapted to fit the specific requirements of any project.

PACE: Execute

Consider the questions in your PACE Strategy Document and those below to craft your response:

Task 3. Conclusion Recall that your supervisor, May Santer, asked you to share your findings with the data team in an executive summary. Consider the following questions as you prepare to write your summary. Think about key points you may want to share with the team, and what information is most relevant to the user churn project. **Questions:** 1. Did the data contain any missing values? How many, and which variables were affected? Was there a pattern to the missing data? The dataset has 700 missieng values in the label column. There are no obvious pattern to the missing value 2. What is a benefit of using the median value of a sample instead of the mean? The Mean is subject to the influence of outliers, while the median represents the middle value of the distribution regardless of any outlying values 3. Did your investigation give rise to further questions that you would like to explore or ask the Waze team about? Yes, for example, the median user who churned drove 698 kilometers eachday they drove last month, which is about 240% the per-drive-day distance of retained users. It would be helpful to know how this data was collected and if it represents a non=random sample of users 4. What percentage of the users in the dataset were Android users and what percentage were iPhone users? Android is Approx. 36% of the sample, Iphone is 64% 5. What were some distinguishing characteristics of users who churned vs. users who were retained? In general, users who churned drove farther and longer in fewer days than retained users. They also used the app about half as many times as retained users over the same period. 6. Was there an appreciable difference in churn rate between iPhone users vs. Android users? No, the churn rate for both iPhone and Android users was within one percentage point of each other. There is nothing suggestive of churn being correlated with device

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.