Lab Assignment 04

The objective of this lab assignment is to explore a dataset that contains information from customers of a telephone company (`data_lab_04.csv`). We will analyze the features in the dataset and try to determine which of these features are good indicators of customer churn (that is, loss of customers).

Instructions: ¶

Complete each task and question by filling in the blanks (`...`) with one or more lines of code or text. Each task and question is worth **0.5 points** (out of **10 points**).

Submission:

This assignment is due **Wednesday, September 25, at 11:59PM (Central Time)**.

This assignment must be submitted on Gradescope as a **PDF file** containing the completed code for each task and the corresponding output. Late submissions will be accepted within **0-12** hours after the deadline with a **0.5-point (5%) penalty** and within **12-24** hours after the deadline with a **2-point (20%) penalty**. No late submissions will be accepted more than 24 hours after the deadline.

This assignment is individual. Offering or receiving any kind of unauthorized or unacknowledged assistance is a violation of the University's academic integrity policies, will result in a grade of zero for the assignment, and will be subject to disciplinary action.

Part 1: Exploring the Dataset

In [28]:
```
# Load libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [29]:
```
# Load dataset
data = pd.read_csv('data_lab_04.csv')
```

In [30]:
```
# Display the first three rows of the dataset
data.head(3)
```

Out[30]:

	State	Account length	Area code	International plan	Voice mail plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	110

Task 01 (of 15): Display the first three rows and the first three columns of the dataset using the `iloc` and `loc` methods. *Hint:* Remember that the `iloc` method is used for indexing by integer position and the `loc` method is used for indexing by label.

In [31]:
```
data.iloc[[0,1,2],[0,1,2]]
```

Out[31]:

	State	Account length	Area code
0	KS	128	415
1	OH	107	415
2	NJ	137	415

In [32]:
```
data.loc[[0,1,2], ["State", "Account length", "Area code"]]
```

Out[32]:

	State	Account length	Area code
0	KS	128	415
1	OH	107	415
2	NJ	137	415

Task 02 (of 15): Determine the dimensionality of the dataset. Then, display information (data types, number of values) about the features in the dataset. *Hint:* Use methods `shape` and `info` .

In [33]:
```
data.shape
```

Out[33]: (3333, 20)

In [34]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 20 columns):
State                          3333 non-null object
Account length                 3333 non-null int64
Area code                      3333 non-null int64
International plan             3333 non-null object
Voice mail plan                3333 non-null object
Number voice mail messages     3333 non-null int64
Total day minutes              3333 non-null float64
Total day calls                3333 non-null int64
Total day charge               3333 non-null float64
Total eve minutes              3333 non-null float64
Total eve calls                3333 non-null int64
Total eve charge               3333 non-null float64
Total night minutes            3333 non-null float64
Total night calls              3333 non-null int64
Total night charge             3333 non-null float64
Total intl minutes             3333 non-null float64
Total intl calls               3333 non-null int64
Total intl charge              3333 non-null float64
Customer service calls         3333 non-null int64
Churn                          3333 non-null bool
dtypes: bool(1), float64(8), int64(8), object(3)
memory usage: 498.1+ KB
```

Question 01 (of 05): How many observations and how many features are in the dataset? What are the data types of the features? Are there any missing values?

Answer: There are 3333 observartions and 20 features (columns) in the dataset. The features such as State, International plan and Voice mail plan are of "object" data type. The data type of Account length, Area code, Number of voice mail messages, Total day calls, Total eve calls, Total night calls, Total intl calls and Customer service calls is of type "int64". The features: Total day minutes, Total day charge, Total eve minutes, Total eve charge, Total night minutes, Total night charge, Total intl minutes and Total intl charge; are of type "float64". And the feature Churn is of type "bool". There are no missing values.

Part 2: Transforming the Features

Task 03 (of 15): Change the data type of feature 'Churn' from bool to int64 and change the values of feature 'International plan' from Yes/No to True/False. *Hint:* Use methods `astype` and `map` .

In [35]:
```
data['Churn'] = data["Churn"].astype("int64")
# data["International plan"] = data["International plan"].astype("bool") astyp
e mapping no to True
change_values = pd.Series({'No' : False, 'Yes' : True})
data["International plan"] = data["International plan"].map(change_values)
data.head(3)
```

Out[35]:

	State	Account length	Area code	International plan	Voice mail plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls
0	KS	128	415	False	Yes	25	265.1	110	45.07	197.4	99
1	OH	107	415	False	Yes	26	161.6	123	27.47	195.5	103
2	NJ	137	415	False	No	0	243.4	114	41.38	121.2	110

Task 04 (of 15): Create a new numerical feature named 'Total charge' that contains the sum of the day, evening, and night charges. Then, sort the dataset in descending order by total charge. *Hint:* Use method `sort_values` .

In [36]:
```
data['Total charge'] = data["Total day charge"] + data["Total night charge"] +
data["Total eve charge"]
data.sort_values(by = ['Total charge'], ascending=False, inplace = True)
data.head(3)
```

Out[36]:

	State	Account length	Area code	International plan	Voice mail plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	...
985	NY	64	415	True	No	0	346.8	55	58.96	249.5	...
15	NY	161	415	False	No	0	332.9	67	56.59	317.8	...
365	CO	154	415	False	No	0	350.8	75	59.64	216.5	...

3 rows × 21 columns

Part 3: Summarizing the Features

Task 05 (of 15): Compute summary statistics for all numerical features and all non-numerical features. *Hint:* Use method `describe` with the appropriate parameters.

In [37]:
```
data.describe()
#data.describe(include = [np.number])
```

Out[37]:

	Account length	Area code	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total e minut
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.0000
mean	101.064806	437.182418	8.099010	179.775098	100.435644	30.562307	200.9803
std	39.822106	42.371290	13.688365	54.467389	20.069084	9.259435	50.7138
min	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000	0.0000
25%	74.000000	408.000000	0.000000	143.700000	87.000000	24.430000	166.6000
50%	101.000000	415.000000	0.000000	179.400000	101.000000	30.500000	201.4000
75%	127.000000	510.000000	20.000000	216.400000	114.000000	36.790000	235.3000
max	243.000000	510.000000	51.000000	350.800000	165.000000	59.640000	363.7000

In [38]:
```
data.describe(exclude=[np.number])
```

Out[38]:

	State	International plan	Voice mail plan
count	3333	3333	3333
unique	51	2	2
top	WV	False	No
freq	106	3010	2411

Task 06 (of 15): Group the data by feature 'Churn' and compute summary statistics for all numerical variables again. *Hint:* Use method `groupby` .

In [39]:
```
data.groupby("Churn").describe()
```

Out[39]:

	Account length								Area code		...	Cu se ca
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75
Churn												
0	2850.0	100.793684	39.88235	1.0	73.0	100.0	127.0	243.0	2850.0	437.074737	...	2
1	483.0	102.664596	39.46782	1.0	76.0	103.0	127.0	225.0	483.0	437.817805	...	4

2 rows × 136 columns

Task 07 (of 15): Compute the percentage of churned and non-churned customers. *Hint:* Use method `value_counts` with the appropriate parameters.

In [52]:
```
data["Churn"].value_counts(normalize = True)
```

Out[52]:
```
0    0.855086
1    0.144914
Name: Churn, dtype: float64
```

Task 08 (of 15): Compute the mean values of all numerical features for churned and non-churned customers. Notice the differences and similarities between both groups.

In [41]:
```
data.groupby("Churn").mean()
```

Out[41]:

Churn	Account length	Area code	International plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Tota min
0	100.793684	437.074737	0.065263	8.604561	175.175754	100.283158	29.780421	199
1	102.664596	437.817805	0.283644	5.115942	206.914079	101.335404	35.175921	212

In [42]:
```
...
```

Out[42]:
```
Ellipsis
```

Question 02 (of 05): What is the percentage of churned customers? What is the mean total charge for churned customers? What is the percentage of non-churned customers? What is the mean total charge for non-churned customers

Answer:

The percentage of Churned customers is 14.49 approximately. The mean total charge of Churned customers is around 62.46.

The percentage of non-Churned customers is 85.50 approximately. The mean total charge of non-Churned customers is around 55.70.

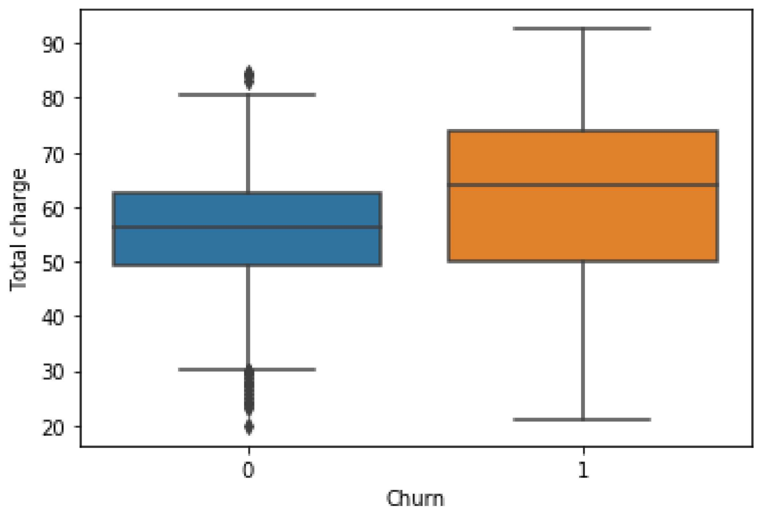
Part 4: Visualizing the Features

Task 09 (of 15): Visualize the summary statistics of churned and non-churned customers for feature 'Total charge'. *Hint:* Use function `seaborn.boxplot()` with the apropriate parameters. Make sure you group customers by feature 'Churn'!

In [43]:
```
sns.boxplot(x ="Churn", y = "Total charge", data = data)

#data1 = data.groupby("Churn").describe()["Total charge"].transpose()
#print(data1)
#num_columns = len(data1.columns)
#fig, axes = plt.subplots(1, num_columns, figsize = (10, 5))
#for i in range(num_columns):
#    sns.boxplot(y = data1.columns[i], data = data1,  orient = 'v',ax = axes
[i])
```

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x2239e03a2b0>



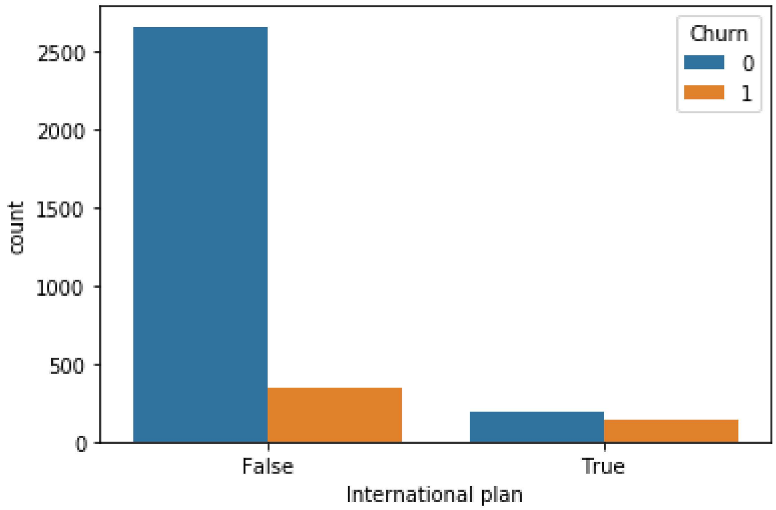
Question 03 (of 05): What do you observe in the plot?

Answer: There are no outliers in the boxplot of total charge of churned customers. The median, max, Q3, IQR, and the range of the total charge of churned customers is higher than those of the non-churned customers.

Task 10 (of 15): Visualize the number of churned and non-churned customers in each category of feature 'International plan'. *Hint:* Use function `seaborn.countplot()` with the apropriate parameters. Make sure you group customers by feature 'Churn'!

In [45]:
```
sns.countplot(x = "International plan", hue = "Churn",data = data)
```

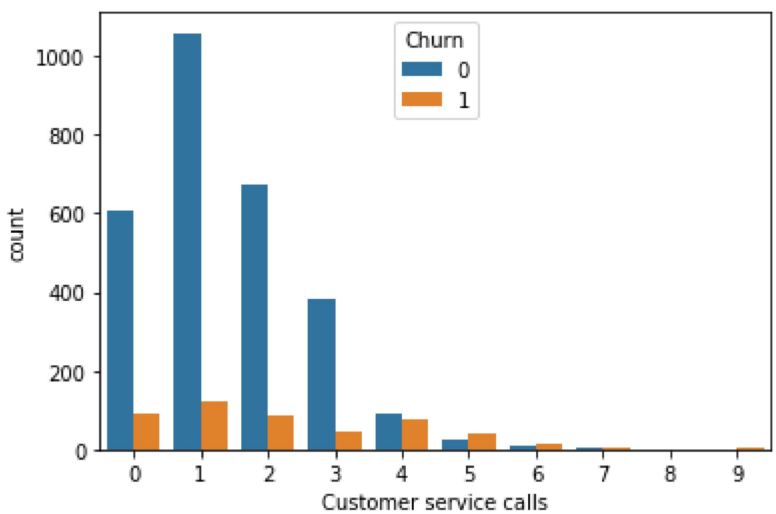
Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x2239f3fb908>



Task 11 (of 15): Visualize the number of churned and non-churned customers in each category of feature 'Customer service calls'. *Hint:* Use function `seaborn.countplot()` with the apropriate parameters. Make sure you group customers by feature 'Churn'!

In [46]:
```
sns.countplot(x = "Customer service calls",hue = "Churn", data = data)
```

Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x2239f57db38>



Task 12 (of 15): Create a new Boolean feature named 'Many customer service calls' that indicates whether a user has made more than 3 customer service calls.

In [47]:
```
data['Many customer service calls'] = (data["Customer service calls"]>3)
data.head(3)
```

Out[47]:

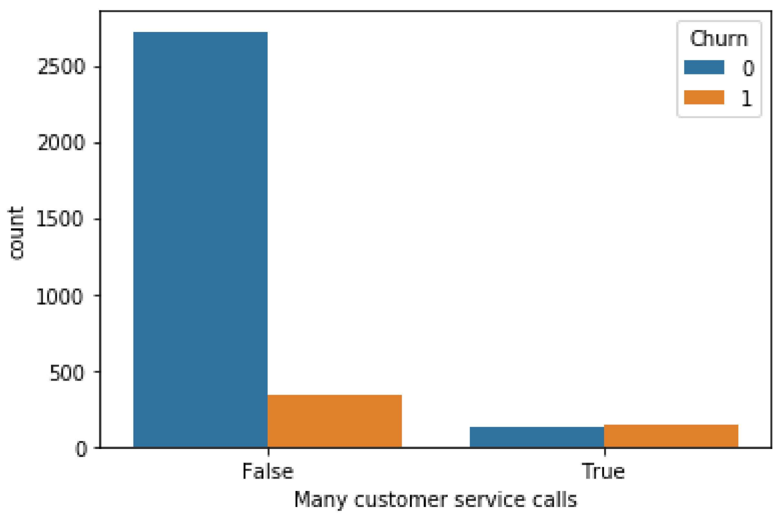
	State	Account length	Area code	International plan	Voice mail plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	...
985	NY	64	415	True	No	0	346.8	55	58.96	249.5	...
15	NY	161	415	False	No	0	332.9	67	56.59	317.8	...
365	CO	154	415	False	No	0	350.8	75	59.64	216.5	...

3 rows × 22 columns

Task 13 (of 15): Visualize the number of churned and non-churned customers in each category of feature 'Many customer service calls'. *Hint:* Use function `seaborn.countplot()` with the apropriate parameters. Make sure you group customers by feature 'Churn'!

In [48]:
```
sns.countplot(x = "Many customer service calls", hue = "Churn",data = data)
```

Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x2239f3fb278>



Question 04 (of 05): What do you observe in the plots?

Answer:

In case of the bar plot of international plan and churn, there are more non-churned customers with or without international plan. Particularly there are very large no of non-churned customers numbers with no international plan.

Same is the case for the bar plots of Customer service calls (<=4) and churn, however there are more no of churned customers with customer service calls >= 5 than non-churned customers.

And in case of the bar plots of Many customer service calls and churn, the no of churned customers without Many customer service calls is very larger compared to that of non-churned customers without Many customer service calls. But, the no of churned and non-churned customers with Many customer service calls are almost same (the no of non-churned customers is tad bit higher). It makes sense by looking at the bar plots of customer service calls (>=4) and the churn.

Part 5: Making Conclusions

Task 14 (of 15): Compute the churn rate (percentage of churned customers) for customers without international plan and for customers with international plan. *Hint:* Use method `value_counts` .

In [57]:
```
# Compute churn rate for customers without international plan
num_churned = data["Churn"][data["International plan"] == False].value_counts
().iloc[1]
num_nonchurned = data["Churn"][data["International plan"] == False].value_coun
ts().iloc[0]
churn_rate = (num_churned)/(num_churned+num_nonchurned)
#print(num_churned)
#print(num_nonchurned)
print(churn_rate)
```

```
0.11495016611295682
```

In [41]:
```
# Compute churn rate for customers with international plan
num_churned = data["Churn"][data["International plan"] == True].value_counts()
.iloc[1]
num_nonchurned = data["Churn"][data["International plan"] == True].value_count
s().iloc[0]
churn_rate = (num_churned)/(num_churned+num_nonchurned)
#print(num_churned)
#print(num_nonchurned)
print(churn_rate)
```

```
0.4241486068111455
```

Task 15 (of 15): Compute the churn rate (percentage of churned customers) for customers with 3 customer service calls or less and for customers with more than 3 service calls. *Hint:* Use method `value_counts`.

In [59]:
```
# Compute churn rate for customers with 3 customer service calls or less
num_churned = data["Churn"][data["Customer service calls"] <= 3].value_counts
().iloc[1]
num_nonchurned = data["Churn"][data["Customer service calls"] <= 3].value_coun
ts().iloc[0]
churn_rate = (num_churned)/(num_churned+num_nonchurned)
#print(num_churned)
#print(num_nonchurned)
print(churn_rate)
```

```
0.11252446183953033
```

In [60]:
```
# Compute churn rate for customers with more than 3 customer service calls
num_churned = data["Churn"][data["Customer service calls"] > 3].value_counts()
.iloc[1]
num_nonchurned = data["Churn"][data["Customer service calls"] > 3].value_count
s().iloc[0]
churn_rate = (num_churned)/(num_churned+num_nonchurned)
#print(num_churned)
#print(num_nonchurned)
print(churn_rate)
```

```
0.48314606741573035
```

Question 05 (of 05): What are your final conclusions from the exploration of features 'International plan' and 'Many customer service calls'? What other tasks would you perform to explore this dataset?

Answer:

As the churn rate with International plan is significantly higher than the churn rates without international plan, there is a postive correlation between International plan and no of churned customers.

Also as the churn rate with more than 3 customer service calls (Many customer service calls) is significantly higher than the churn rates with 3 customer service calls or less, there is a postive correlation between Many customer service calls and no of churned customers.

I would group the data by states and Voice mail plan, and find out if there is any correlation between states (or Voice mail plan) and the no of churned customers.

In [ ]: