TELECOM CHURN ANALYSIS PROJECT

Nguyen Trong Duc

OVERVIEW

01 - Data Overview

02 - EDA

03 - Data Preprocessing

04 - Building Model

1- DATA OVERVIEW

1- DATA DISCOVERY

This is a dataset containing customer information of a telecom business. This data has 3333 rows and 20 columns, includes:

State: State where the customer lives
Account lenght: The length of time a customer uses a business's services
Area code: Area code
International plan: Indicates whether the customer is subscribed to an international calling plan (Yes or No)
Voice mail plan: Indicates whether the customer is subscribed to a voice mail service plan (Yes or No)
Number vmail messages: The number of voicemail messages the customer has received
Total day minutes: Total number of calling minutes during daytime hours
Total day calls: Total number of calls made during the daytime
Total day charge: Total charges for calls during daytime hours
Total eve minutes: Total number of calling minutes during the evening hours
Total eve calls: Total number of calls made during the evening
Total eve charge: Total charges for calls during evening hours
Total night minutes: Total number of calling minutes during night time
Total night calls: Total number of calls made during night time
Total night charge: Total charges for calls during night time
Total intl minutes: Total international calling minutes
Total intl calls: Total number of international calls
Total intl charge: Total charges for international calls
Customer service calls: Total customer service calls
Churn: Indicates whether the customer churned (True is Yes, False is No)

DATA SAMPLE

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls	Total intl charge	Customer service calls	Churn
0	LA	117	408	No	No	0	184.5	97	31.37	351.6	80	29.89	215.8	90	9.71	8.7	4	2.35	1	False
1	IN	65	415	No	No	0	129.1	137	21.95	228.5	83	19.42	208.8	111	9.40	12.7	6	3.43	4	True
2	NY	161	415	No	No	0	332.9	67	56.59	317.8	97	27.01	160.6	128	7.23	5.4	9	1.46	4	True
3	SC	111	415	No	No	0	110.4	103	18.77	137.3	102	11.67	189.6	105	8.53	7.7	6	2.08	2	False
4	HI	49	510	No	No	0	119.3	117	20.28	215.1	109	18.28	178.7	90	8.04	11.1	1	3.00	1	False
5	AK	36	408	No	Yes	30	146.3	128	24.87	162.5	80	13.81	129.3	109	5.82	14.5	6	3.92	0	False
6	MI	65	415	No	No	0	211.3	120	35.92	162.6	122	13.82	134.7	118	6.06	13.2	5	3.56	3	False
7	ID	119	415	No	No	0	159.1	114	27.05	231.3	117	19.66	143.2	91	6.44	8.8	3	2.38	5	True
8	VA	10	408	No	No	0	186.1	112	31.64	190.2	66	16.17	282.8	57	12.73	11.4	6	3.08	2	False
9	WI	68	415	No	No	0	148.8	70	25.30	246.5	164	20.95	129.8	103	5.84	12.1	3	3.27	3	False

Check null

```python
#Check null
df.isna().sum()
```

```
State                     0
Account length            0
Area code                 0
International plan        0
Voice mail plan           0
Number vmail messages     0
Total day minutes         0
Total day calls           0
Total day charge          0
Total eve minutes         0
Total eve calls           0
Total eve charge          0
Total night minutes       0
Total night calls         0
Total night charge        0
Total intl minutes        0
Total intl calls          0
Total intl charge         0
Customer service calls    0
Churn                     0
dtype: int64
```

```python
#Check duplicate
duplicate_row = df[df.duplicated]
len(duplicate_row)
```

```
0
```

Check duplicate

CHECK VALUES

- Check if the dataset has negative values or not

```
[ ] #Filter out columns with numeric data types (int and float)
    numeric_df = df.select_dtypes(include=['int64', 'float64'])
    #Check
    contains_negative = (numeric_df < 0).any().any()
    if contains_negative:
        print("DataFrame contains negative values.")
    else:
        print("DataFrame does not contain any negative values.")

    DataFrame does not contain any negative values.
```

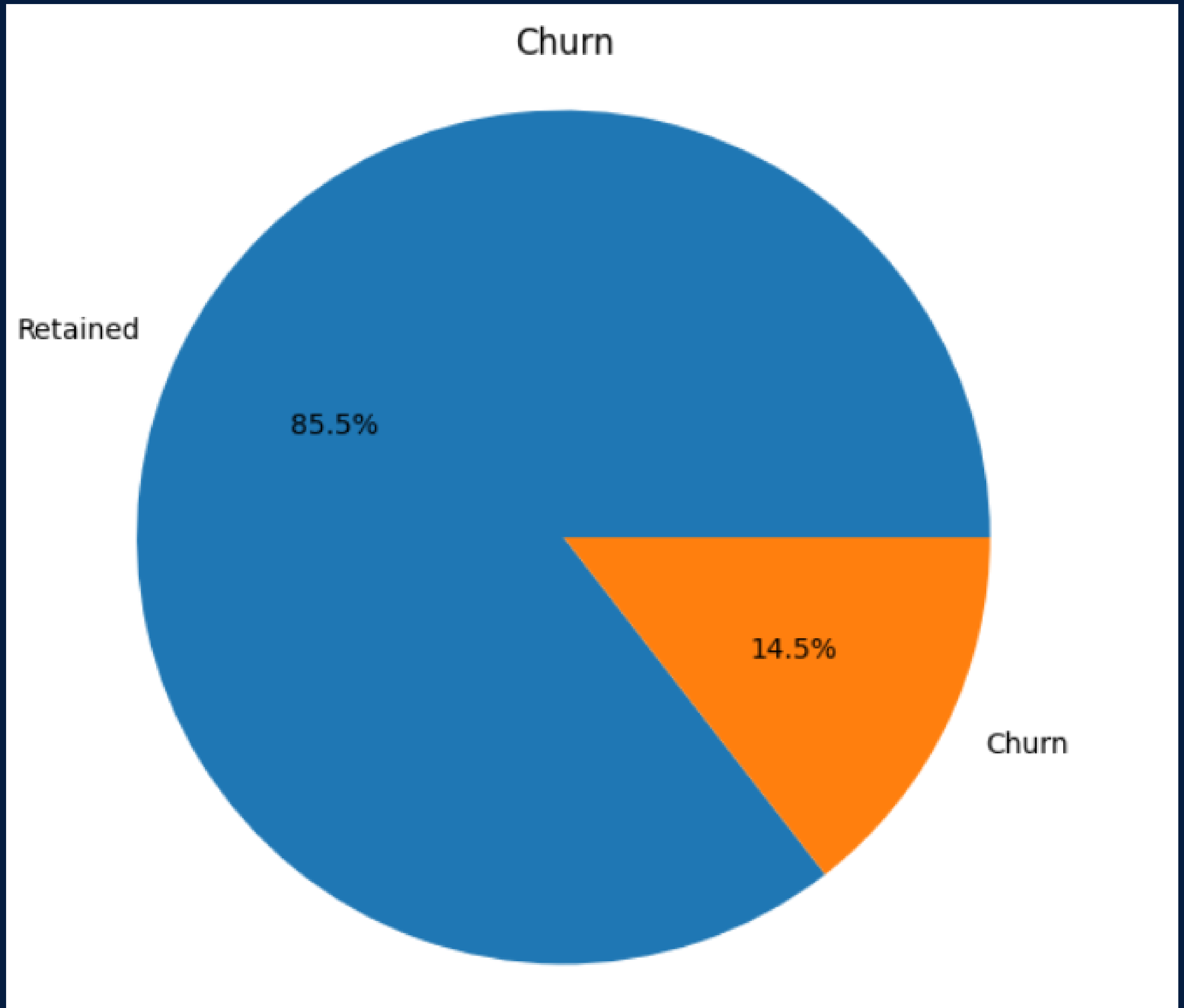
- Check values in categorical columns

```
[ ] #International plan
    df['International plan'].unique()

    array(['No', 'Yes'], dtype=object)

[ ] #Voice mail plan
    df['Voice mail plan'].unique()

    array(['No', 'Yes'], dtype=object)
```

➔ **Conclusion:** This dataset seems quite good. It has no null or duplicate values and these data types is correct. Besides, the values in the columns are valid.

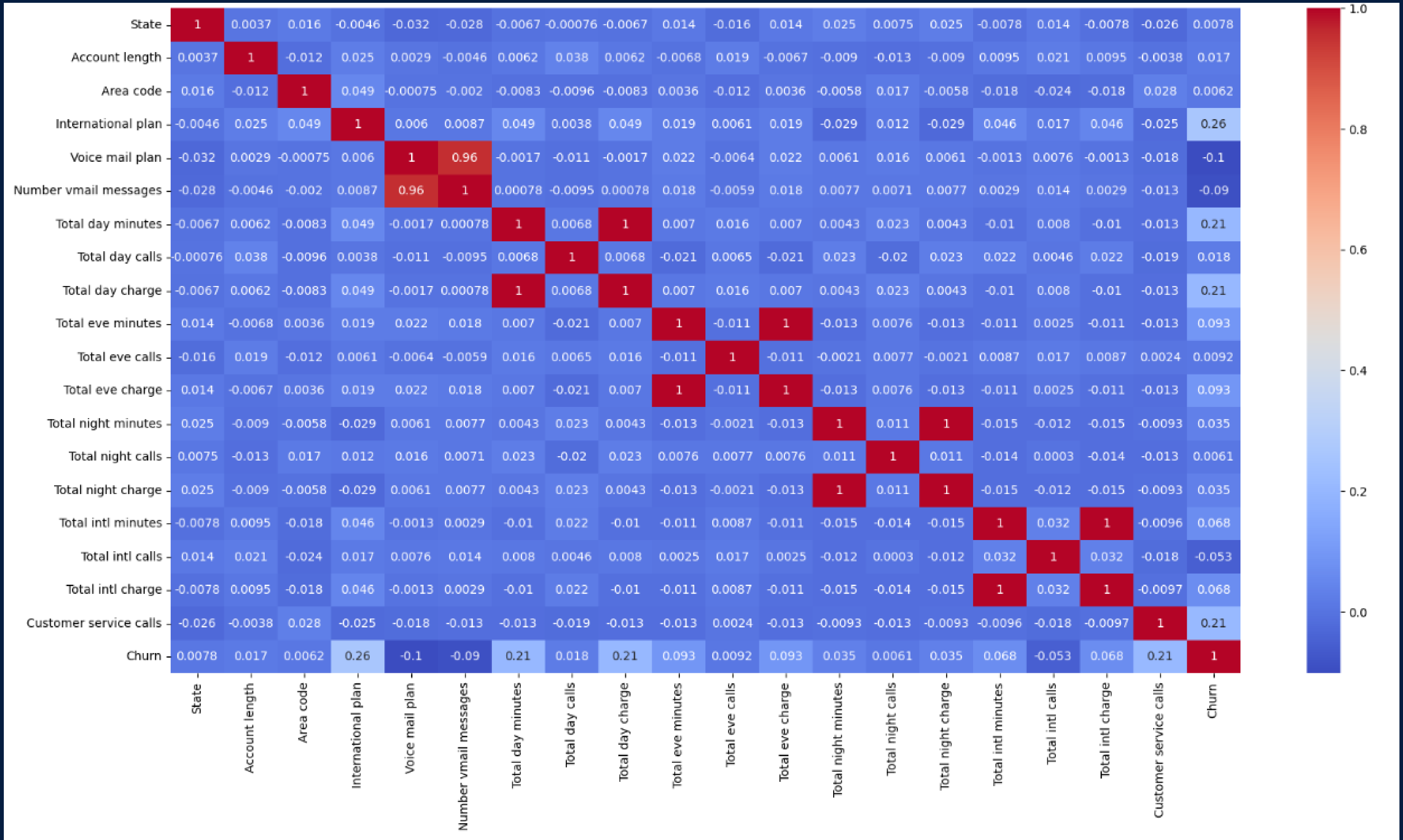
2- EDA

TARGET VARIANCE



There are 2850 users (85,5%) retained and 483 users (14,5%) churned. **This data is highly imbalanced**, it need to be processed before building the models.

HEATMAP


State 1 0.0037 0.016 -0.0046 -0.032 -0.028 -0.0067 -0.00076 -0.0067 0.014 -0.016 0.014 0.025 0.0075 0.025 -0.0078 0.014 -0.0078 -0.026 0.0078
Account length 0.0037 1 -0.012 0.025 0.0029 -0.0046 0.0062 0.038 0.0062 -0.0068 0.019 -0.0067 -0.009 -0.013 -0.009 0.0095 0.021 0.0095 -0.0038 0.017
Area code 0.016 -0.012 1 0.049 -0.00075 -0.002 -0.0083 -0.0096 -0.0083 0.0036 -0.012 0.0036 -0.0058 0.017 -0.0058 -0.018 -0.024 -0.018 0.028 0.0062
International plan -0.0046 0.025 0.049 1 0.006 0.0087 0.049 0.0038 0.049 0.019 0.0061 0.019 -0.029 0.012 -0.029 0.046 0.017 0.046 -0.025 0.26
Voice mail plan -0.032 0.0029 -0.00075 0.006 1 0.96 -0.0017 -0.011 -0.0017 0.022 -0.0064 0.022 0.0061 0.016 0.0061 -0.0013 0.0076 -0.0013 -0.018 -0.1
Number vmail messages -0.028 -0.0046 -0.002 0.0087 0.96 1 0.00078 -0.0095 0.00078 0.018 -0.0059 0.018 0.0077 0.0071 0.0077 0.0029 0.014 0.0029 -0.013 -0.09
Total day minutes -0.0067 0.0062 -0.0083 0.049 -0.0017 0.00078 1 0.0068 1 0.007 0.016 0.007 0.0043 0.023 0.0043 -0.01 0.008 -0.01 -0.013 0.21
Total day calls -0.00076 0.038 -0.0096 0.0038 -0.011 -0.0095 0.0068 1 0.0068 -0.021 0.0065 -0.021 0.023 -0.02 0.023 0.022 0.0046 0.022 -0.019 0.018
Total day charge -0.0067 0.0062 -0.0083 0.049 -0.0017 0.00078 1 0.0068 1 0.007 0.016 0.007 0.0043 0.023 0.0043 -0.01 0.008 -0.01 -0.013 0.21
Total eve minutes 0.014 -0.0068 0.0036 0.019 0.022 0.018 0.007 -0.021 0.007 1 -0.011 1 -0.013 0.0076 -0.013 -0.011 0.0025 -0.011 -0.013 0.093
Total eve calls -0.016 0.019 -0.012 0.0061 -0.0064 -0.0059 0.016 0.0065 0.016 -0.011 1 -0.011 -0.0021 0.0077 -0.0021 0.0087 0.017 0.0087 0.0024 0.0092
Total eve charge 0.014 -0.0067 0.0036 0.019 0.022 0.018 0.007 -0.021 0.007 1 -0.011 1 -0.013 0.0076 -0.013 -0.011 0.0025 -0.011 -0.013 0.093
Total night minutes 0.025 -0.009 -0.0058 -0.029 0.0061 0.0077 0.0043 0.023 0.0043 -0.013 -0.0021 -0.013 1 0.011 1 -0.015 -0.012 -0.015 -0.0093 0.035
Total night calls 0.0075 -0.013 0.017 0.012 0.016 0.0071 0.023 -0.02 0.023 0.0076 0.0077 0.0076 0.011 1 0.011 -0.014 0.0003 -0.014 -0.013 0.0061
Total night charge 0.025 -0.009 -0.0058 -0.029 0.0061 0.0077 0.0043 0.023 0.0043 -0.013 -0.0021 -0.013 1 0.011 1 -0.015 -0.012 -0.015 -0.0093 0.035
Total intl minutes -0.0078 0.0095 -0.018 0.046 -0.0013 0.0029 -0.01 0.022 -0.01 -0.011 0.0087 -0.011 -0.015 -0.014 -0.015 1 0.032 1 -0.0096 0.068
Total intl calls 0.014 0.021 -0.024 0.017 0.0076 0.014 0.008 0.0046 0.008 0.0025 0.017 0.0025 -0.012 0.0003 -0.012 0.032 1 0.032 -0.018 -0.053
Total intl charge -0.0078 0.0095 -0.018 0.046 -0.0013 0.0029 -0.01 0.022 -0.01 -0.011 0.0087 -0.011 -0.015 -0.014 -0.015 1 0.032 1 -0.0097 0.068
Customer service calls -0.026 -0.0038 0.028 -0.025 -0.018 -0.013 -0.013 -0.019 -0.013 -0.013 0.0024 -0.013 -0.0093 -0.013 -0.0093 -0.0096 -0.018 -0.0097 1 0.21
Churn 0.0078 0.017 0.0062 0.26 -0.1 -0.09 0.21 0.018 0.21 0.093 0.0092 0.093 0.035 0.0061 0.035 0.068 -0.053 0.068 0.21 1
State
Account length
Area code
International plan
Voice mail plan
Number vmail messages
Total day minutes
Total day calls
Total day charge
Total eve minutes
Total eve calls
Total eve charge
Total night minutes
Total night calls
Total night charge
Total intl minutes
Total intl calls
Total intl charge
Customer service calls
Churn
1.0
0.8
0.6
0.4
0.2
0.0

INSIGHT

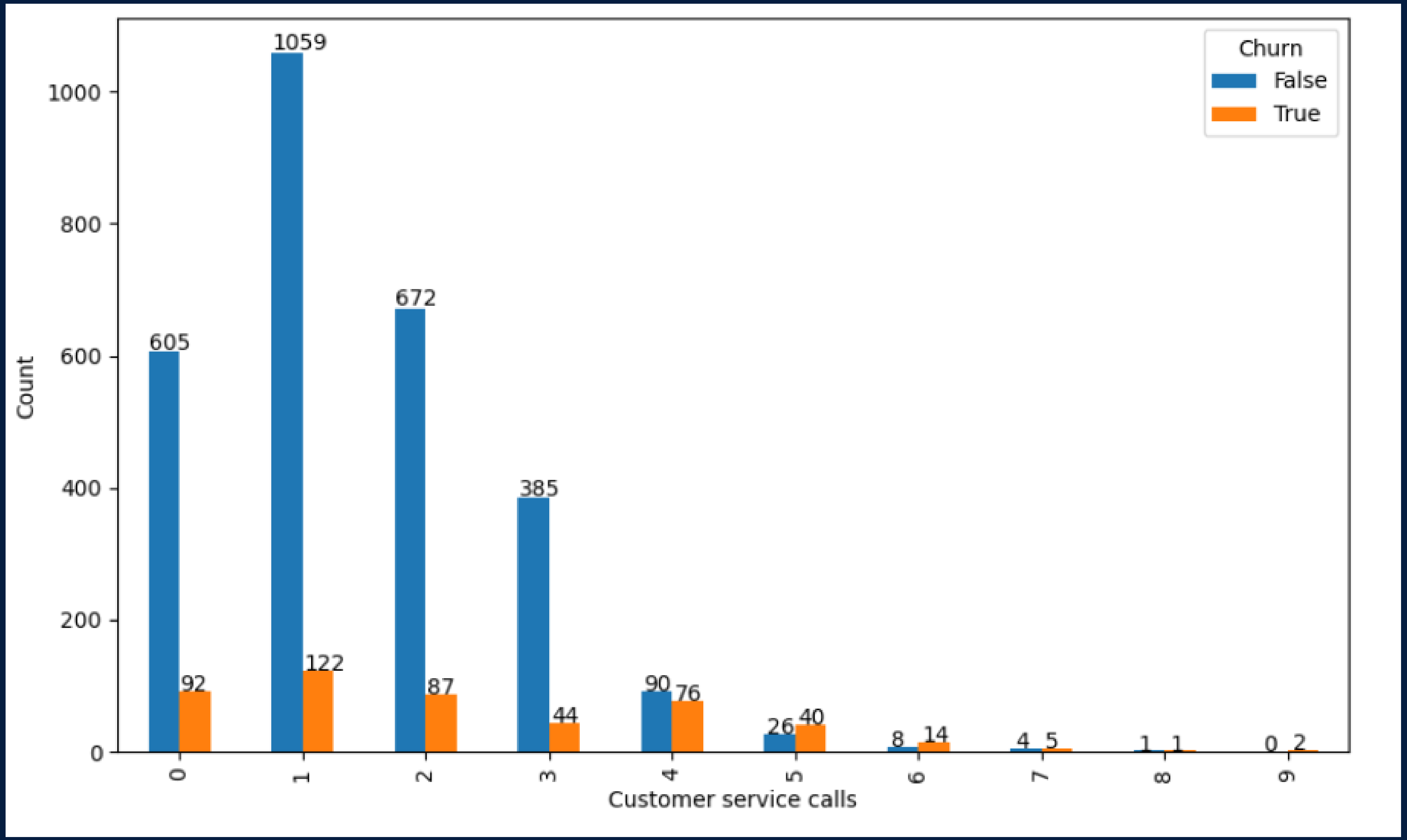
After drawing and observing **Plot distribution of individual predictors by churn**, combined with using **Heatmap,** I got some insights about this dataset.

1- Churn rates increase when the number of service calls increases

2- More customers churn when Total day minutes is greater than 300 and Total day charge is greater than 50

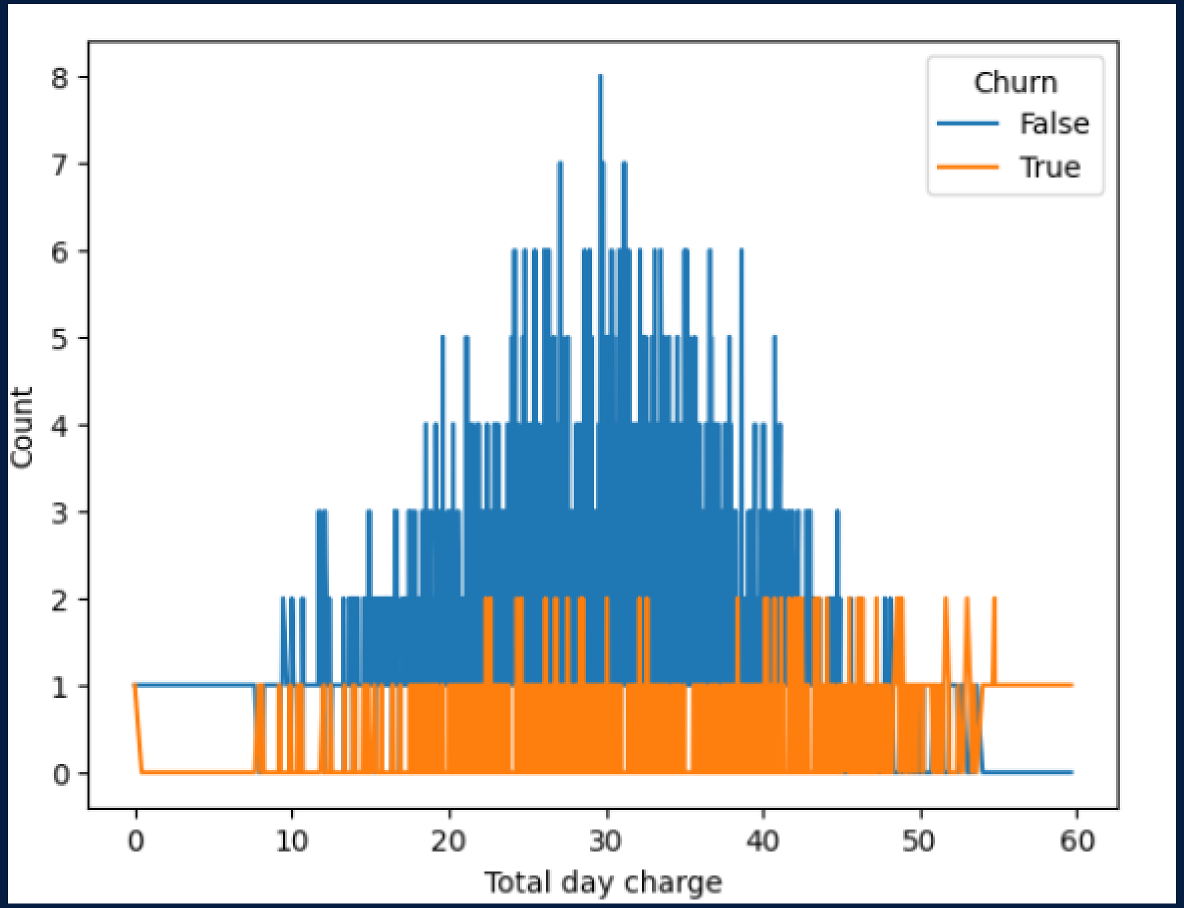
3- Churn rates increase when the customers subcribed the International plan

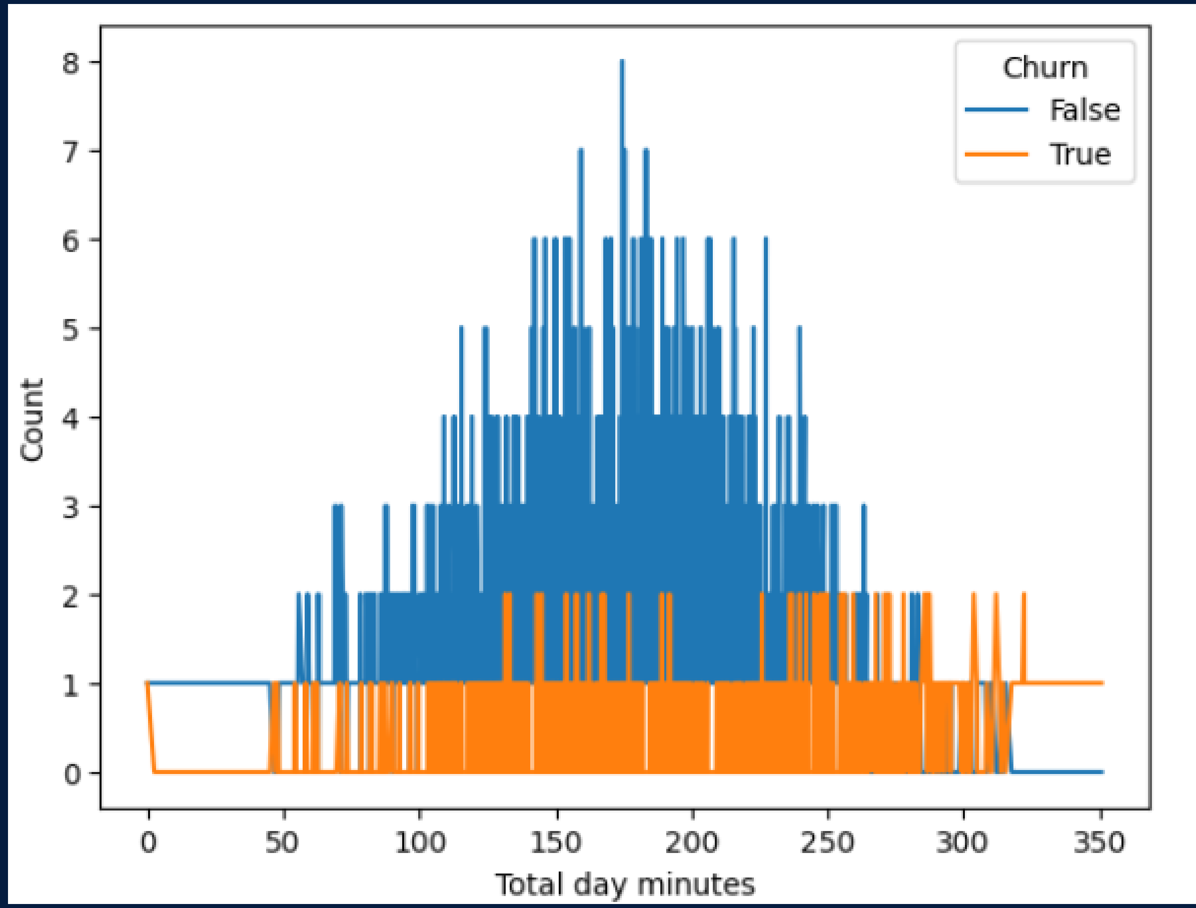
Churn rates increase when the number of service calls increases



Through the chart, we can see that the number of customers churned is always lower. However, with a total number of calls of 4, the number of customers churned and retained is almost equal. And from the 5th call, the number of customer churned is always higher.

More customers churn when Total day minutes is greater than 300 and Total day charge is greater than 50

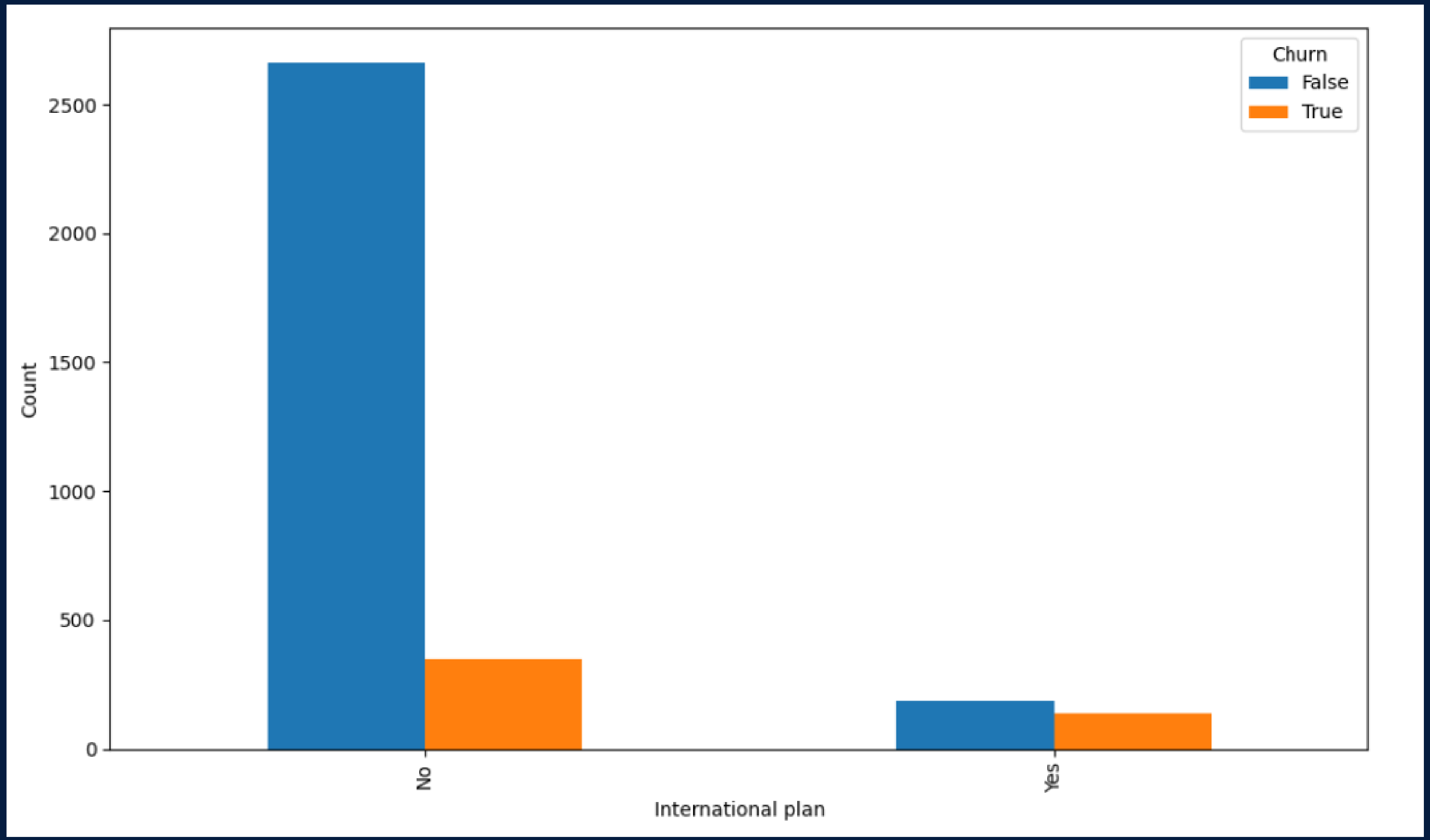




Through the chart, we see that the number of customers retained is always higher, but when the Total day minutes is greater than 300, the number of customers churned is higher. The same thing happens when Total day charge is greater than 50. This is understandable because these two factors are closely related; the more you call, the higher the cost.

Churn rates increase when the customers subcribed the International plan

For the group of customers who didn't subcribe for the International plan, the number of retained customers was significantly higher. However, for subcribed groups, the number of retained customers and churned customers was almost equal.



3- Data Preprocessing

ENCODE

```
#LabelEncoder
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
#Make copies
df_80ml = df_80.copy()
df_20ml = df_20.copy()
#Tranform
for col in df_80ml.select_dtypes(include=['object', 'bool']).columns:
    #Tranform df_80
    df_80ml[col] = label_encoder.fit_transform(df_80ml[col])
    #Tranform for df_20 with trained LabelEncoder
    df_20ml[col] = label_encoder.transform(df_20ml[col])
```

SPLIT DATA

```
X_train = df_80ml.drop('Churn', axis = 1)
y_train = df_80ml['Churn']
X_test = df_20ml.drop('Churn', axis = 1)
y_test = df_20ml['Churn']
```

4- BUILDING MODELS

EVALUATION METHODS

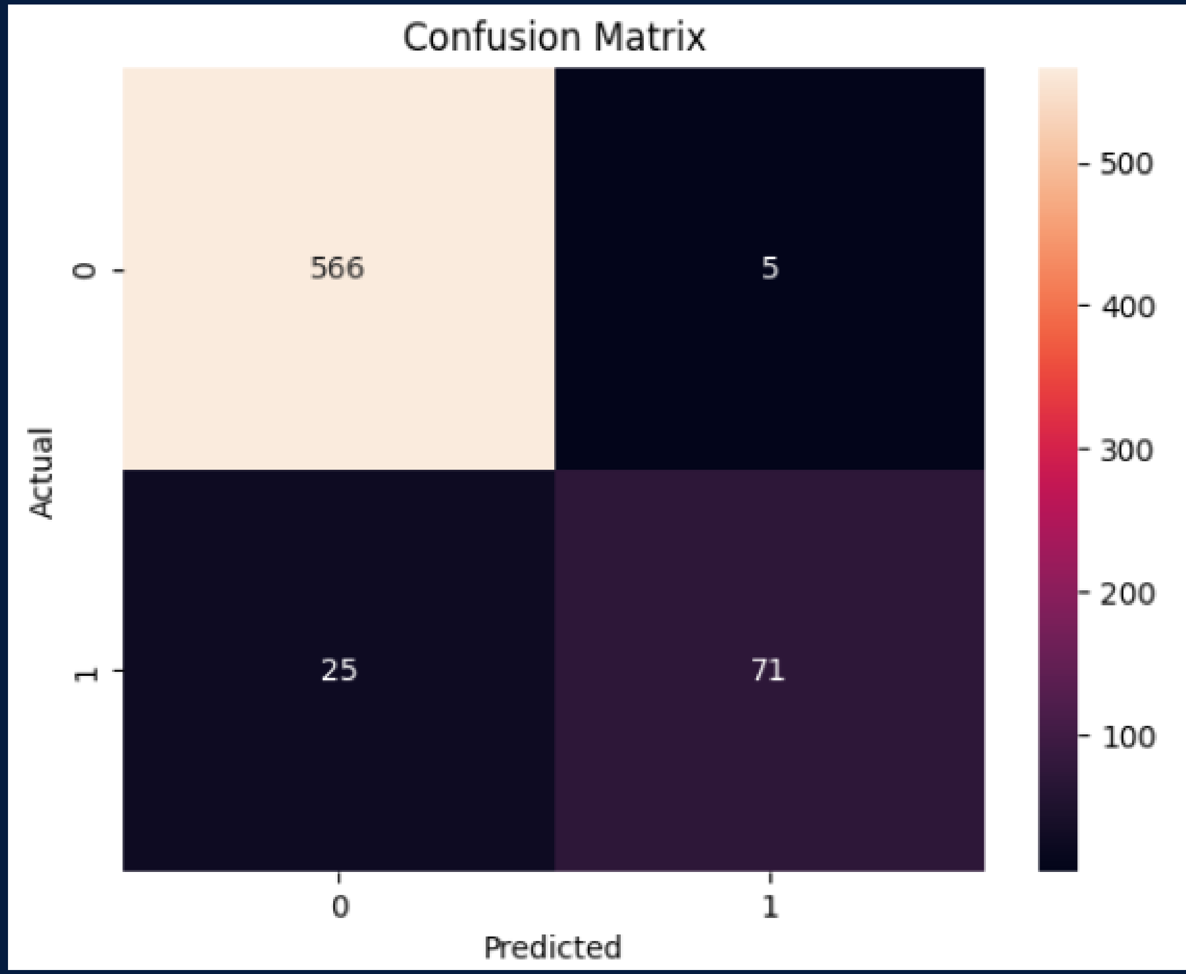
Since this is an **imbalanced dataset**, in addition to the **Classification Accuracy**, I will use the **Confusion Matrix** to evaluate models. I also use **SMOTEENN** to increase model performance. I will give **more importance** to good prediction models for **class 1**, because it represents the class of customers likely to leave, which businesses are always concerned about first.

I builded and evaluated 4 models:

- **Decision Tree**
- **Random Forest**
- **K-Nearest Neighbor**
- **Support Vector Machine**

Model evaluation

➢ Random Forest



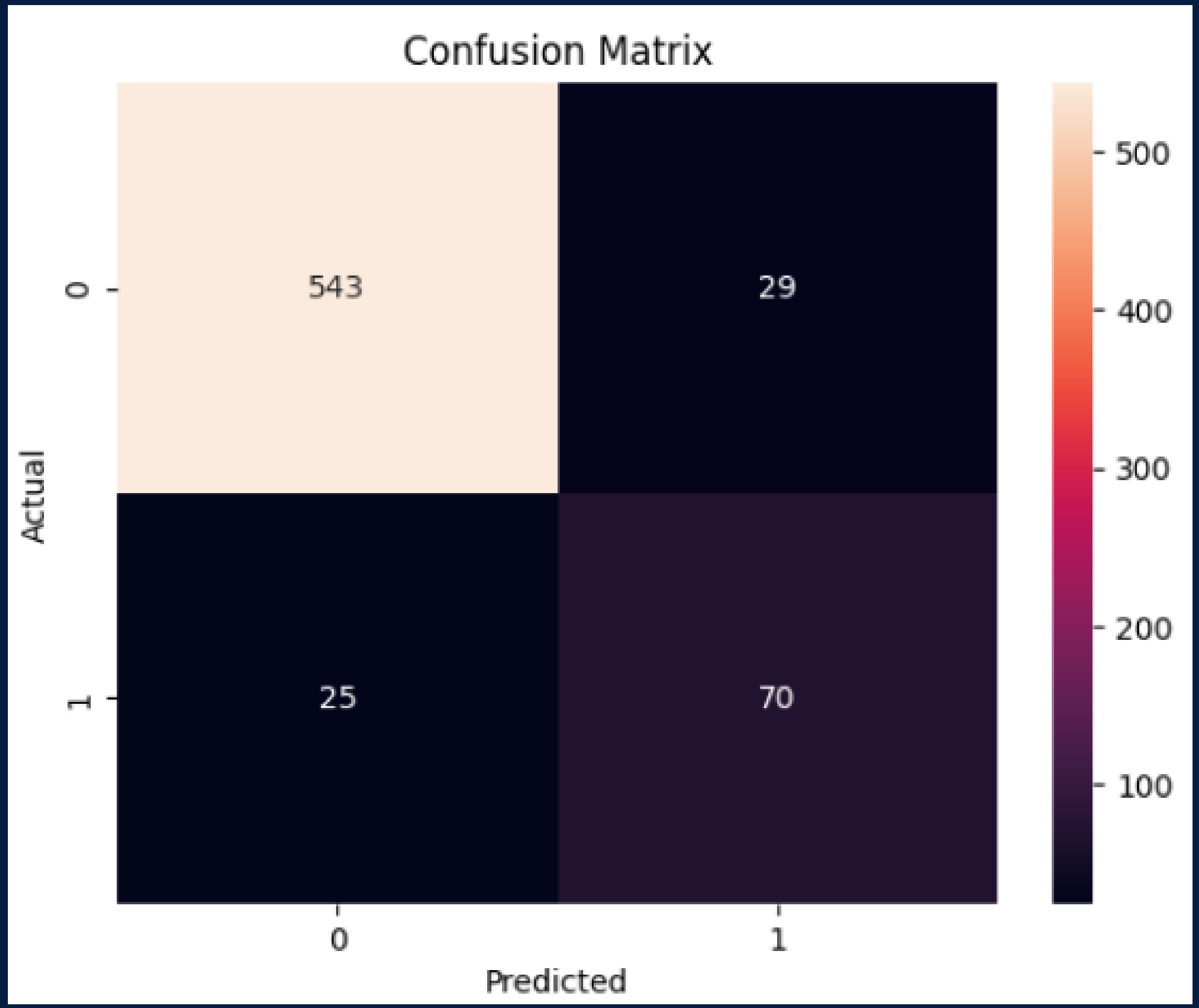
Training Time: 0.26821279525756836 seconds
Classification Accuracy: 0.9550224887556222
Classification Error: 0.044977511244377766
Classification Report:

	precision	recall	f1-score	support
0	0.96	0.99	0.97	571
1	0.93	0.74	0.83	96
accuracy			0.96	667
macro avg	0.95	0.87	0.90	667
weighted avg	0.95	0.96	0.95	667

The best model is **Random Forest** with the highet Classification Accuracy and best prediction for class 1

Model evaluation

➢ Decision Tree




```
Training Time: 0.2348172664642334 seconds
Classification Accuracy: 0.9190404797601199
Classification Error: 0.08095952023988007
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.95      0.95       572
           1       0.71      0.74      0.72        95

    accuracy                           0.92       667
   macro avg       0.83      0.84      0.84       667
weighted avg       0.92      0.92      0.92       667
```

Decision Tree is also a good model with the shortest training time

Thank's For Watching