```
In [4]: %pip install numpy
        %pip install pandas
        %pip install scikit-learn
        %pip install matplotlib
        %pip install seaborn
```

```
Collecting numpy
  Downloading numpy-2.3.1-cp312-cp312-macosx_14_0_arm64.whl.metadata (62 kB)
Downloading numpy-2.3.1-cp312-cp312-macosx_14_0_arm64.whl (5.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.1/5.1 MB 5.5 MB/s eta 0:00:00a
0:00:01
Installing collected packages: numpy
Successfully installed numpy-2.3.1

[notice] A new release of pip is available: 24.2 -> 25.1.1
[notice] To update, run: pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Collecting pandas
  Downloading pandas-2.3.1-cp312-cp312-macosx_11_0_arm64.whl.metadata (91 k
B)
Requirement already satisfied: numpy>=1.26.0 in /Users/arun/myenv/lib/python
3.12/site-packages (from pandas) (2.3.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /Users/arun/myenv/l
ib/python3.12/site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /Users/arun/myenv/lib/python3.12/
site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Downloading pandas-2.3.1-cp312-cp312-macosx_11_0_arm64.whl (10.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.7/10.7 MB 5.3 MB/s eta 0:00:0
0 0:00:01
Downloading pytz-2025.2-py2.py3-none-any.whl (509 kB)
Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.3.1 pytz-2025.2 tzdata-2025.2

[notice] A new release of pip is available: 24.2 -> 25.1.1
[notice] To update, run: pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Collecting scikit-learn
  Using cached scikit_learn-1.7.1-cp312-cp312-macosx_12_0_arm64.whl.metadata
(11 kB)
Requirement already satisfied: numpy>=1.22.0 in /Users/arun/myenv/lib/python
3.12/site-packages (from scikit-learn) (2.3.1)
Collecting scipy>=1.8.0 (from scikit-learn)
  Using cached scipy-1.16.0-cp312-cp312-macosx_14_0_arm64.whl.metadata (61 k
B)
Collecting joblib>=1.2.0 (from scikit-learn)
  Using cached joblib-1.5.1-py3-none-any.whl.metadata (5.6 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Using cached threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Using cached scikit_learn-1.7.1-cp312-cp312-macosx_12_0_arm64.whl (8.6 MB)
Using cached joblib-1.5.1-py3-none-any.whl (307 kB)
Using cached scipy-1.16.0-cp312-cp312-macosx_14_0_arm64.whl (20.8 MB)
Using cached threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
Successfully installed joblib-1.5.1 scikit-learn-1.7.1 scipy-1.16.0 threadpo
olctl-3.6.0

[notice] A new release of pip is available: 24.2 -> 25.1.1
```

```
[notice] To update, run: pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Collecting matplotlib
  Downloading matplotlib-3.10.3-cp312-cp312-macosx_11_0_arm64.whl.metadata
(11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.2-cp312-cp312-macosx_11_0_arm64.whl.metadata (5.
5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.59.0-cp312-cp312-macosx_10_13_universal2.whl.metad
ata (107 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Using cached kiwisolver-1.4.8-cp312-cp312-macosx_11_0_arm64.whl.metadata
(6.2 kB)
Requirement already satisfied: numpy>=1.23 in /Users/arun/myenv/lib/python3.
12/site-packages (from matplotlib) (2.3.1)
Requirement already satisfied: packaging>=20.0 in /Users/arun/myenv/lib/pyth
on3.12/site-packages (from matplotlib) (25.0)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-11.3.0-cp312-cp312-macosx_11_0_arm64.whl.metadata (9.0
kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.2.3-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: python-dateutil>=2.7 in /Users/arun/myenv/li
b/python3.12/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /Users/arun/myenv/lib/python3.12/
site-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
Downloading matplotlib-3.10.3-cp312-cp312-macosx_11_0_arm64.whl (8.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.1/8.1 MB 7.6 MB/s eta 0:00:00a
0:00:01
Downloading contourpy-1.3.2-cp312-cp312-macosx_11_0_arm64.whl (255 kB)
Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.59.0-cp312-cp312-macosx_10_13_universal2.whl (2.8 M
B)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.8/2.8 MB 8.4 MB/s eta 0:00:00a
0:00:01
Using cached kiwisolver-1.4.8-cp312-cp312-macosx_11_0_arm64.whl (65 kB)
Downloading pillow-11.3.0-cp312-cp312-macosx_11_0_arm64.whl (4.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.7/4.7 MB 7.7 MB/s eta 0:00:00a
0:00:01
Downloading pyparsing-3.2.3-py3-none-any.whl (111 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cyc
ler, contourpy, matplotlib
Successfully installed contourpy-1.3.2 cycler-0.12.1 fonttools-4.59.0 kiwiso
lver-1.4.8 matplotlib-3.10.3 pillow-11.3.0 pyparsing-3.2.3

[notice] A new release of pip is available: 24.2 -> 25.1.1
[notice] To update, run: pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Collecting seaborn
  Using cached seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /Users/arun/myenv/li
b/python3.12/site-packages (from seaborn) (2.3.1)
Requirement already satisfied: pandas>=1.2 in /Users/arun/myenv/lib/python3.
```

```
12/site-packages (from seaborn) (2.3.1)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /Users/arun/myenv/
lib/python3.12/site-packages (from seaborn) (3.10.3)
Requirement already satisfied: contourpy>=1.0.1 in /Users/arun/myenv/lib/pyt
hon3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /Users/arun/myenv/lib/python
3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /Users/arun/myenv/lib/py
thon3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.59.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /Users/arun/myenv/lib/py
thon3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /Users/arun/myenv/lib/pyth
on3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (25.0)
Requirement already satisfied: pillow>=8 in /Users/arun/myenv/lib/python3.1
2/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /Users/arun/myenv/lib/pyt
hon3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /Users/arun/myenv/li
b/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.po
st0)
Requirement already satisfied: pytz>=2020.1 in /Users/arun/myenv/lib/python
3.12/site-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /Users/arun/myenv/lib/pytho
n3.12/site-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in /Users/arun/myenv/lib/python3.12/
site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn)
(1.17.0)
Using cached seaborn-0.13.2-py3-none-any.whl (294 kB)
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2

[notice] A new release of pip is available: 24.2 -> 25.1.1
[notice] To update, run: pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

In [5]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style='whitegrid')
```

In [7]:

```python
# Loading Data
df = pd.read_csv("Databel - Data.csv")
df.head(3)
```

Out[7]:

	Customer ID	Churn Label	Account Length (in months)	Local Calls	Local Mins	Intl Calls	Intl Mins	Intl Active	Intl Plan	Extra International Charges
0	4444-BZPU	No	1	3	8.0	0.0	0.0	No	no	0.0
1	5676-PTZX	No	33	179	431.3	0.0	0.0	No	no	0.0
2	8532-ZEKQ	No	44	82	217.6	0.0	0.0	No	yes	0.0

3 rows × 29 columns

Data Check

In [8]:

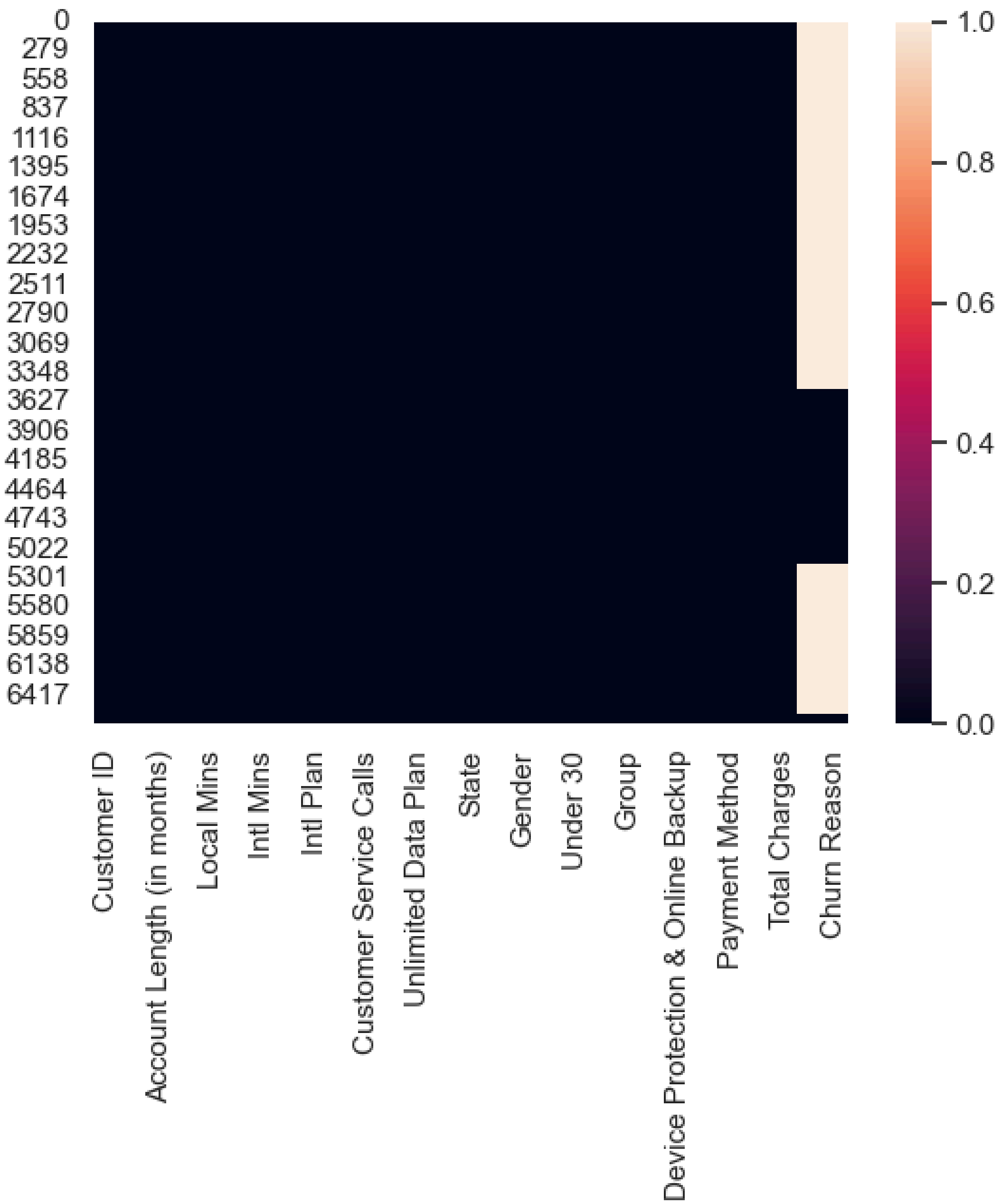
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6687 entries, 0 to 6686
Data columns (total 29 columns):
 #   Column                              Non-Null Count  Dtype  
---  ------                              --------------  -----  
 0   Customer ID                         6687 non-null   object 
 1   Churn Label                         6687 non-null   object 
 2   Account Length (in months)          6687 non-null   int64  
 3   Local Calls                         6687 non-null   int64  
 4   Local Mins                          6687 non-null   float64
 5   Intl Calls                          6687 non-null   float64
 6   Intl Mins                           6687 non-null   float64
 7   Intl Active                         6687 non-null   object 
 8   Intl Plan                           6687 non-null   object 
 9   Extra International Charges         6687 non-null   float64
 10  Customer Service Calls              6687 non-null   int64  
 11  Avg Monthly GB Download             6687 non-null   int64  
 12  Unlimited Data Plan                 6687 non-null   object 
 13  Extra Data Charges                  6687 non-null   int64  
 14  State                               6687 non-null   object 
 15  Phone Number                        6687 non-null   object 
 16  Gender                              6687 non-null   object 
 17  Age                                 6687 non-null   int64  
 18  Under 30                            6687 non-null   object 
 19  Senior                              6687 non-null   object 
 20  Group                               6687 non-null   object 
 21  Number of Customers in Group        6687 non-null   int64  
 22  Device Protection & Online Backup   6687 non-null   object 
 23  Contract Type                       6687 non-null   object 
 24  Payment Method                      6687 non-null   object 
 25  Monthly Charge                      6687 non-null   int64  
 26  Total Charges                       6687 non-null   int64  
 27  Churn Category                      1769 non-null   object 
 28  Churn Reason                        1769 non-null   object 
dtypes: float64(4), int64(9), object(16)
memory usage: 1.5+ MB
```

In [9]:

```
# Heatmap for check Null Values
sns.heatmap(df.isnull())
print("No Null Values in DataSet")
```

```
No Null Values in DataSet
```



> *There is only null values in Churn Reason*

In [10]:
```
import warnings
warnings.filterwarnings('ignore')
print(f"Columns : {df.shape[0]} \nRows : {df.shape[1]}")
```

```
Columns : 6687
Rows : 29
```

In [11]:
```
df.describe()
```

Out[11]:

	Account Length (in months)	Local Calls	Local Mins	Intl Calls	Intl Mins	Ex Internatio Charg
count	6687.000000	6687.000000	6687.000000	6687.000000	6687.000000	6687.0000
mean	32.337820	130.974129	322.752864	51.097524	130.070624	33.6417
std	24.595689	121.893966	288.619931	103.592369	243.527828	76.3468
min	1.000000	1.000000	4.000000	0.000000	0.000000	0.0000
25%	9.000000	31.000000	76.900000	0.000000	0.000000	0.0000
50%	29.000000	98.000000	250.500000	0.000000	0.000000	0.0000
75%	55.000000	199.000000	498.050000	52.000000	140.400000	16.4000
max	77.000000	918.000000	1234.200000	1120.000000	1372.500000	585.8000

In [12]:

```
df.describe(include="object").transpose()
```

Out[12]:

	count	unique	top	freq
Customer ID	6687	6687	4444-BZPU	1
Churn Label	6687	2	No	4891
Intl Active	6687	2	No	4116
Intl Plan	6687	2	no	6036
Unlimited Data Plan	6687	2	Yes	4494
State	6687	51	WV	213
Phone Number	6687	6677	359-9794	2
Gender	6687	3	Male	3379
Under 30	6687	2	No	5400
Senior	6687	2	No	5460
Group	6687	2	No	5166
Device Protection & Online Backup	6687	2	No	4393
Contract Type	6687	3	Month-to-Month	3411
Payment Method	6687	3	Direct Debit	3702
Churn Category	1769	5	Competitor	805
Churn Reason	1769	20	Competitor made better offer	303

In [13]:

```
df.head(3)
```

Out[13]:

	Customer ID	Churn Label	Account Length (in months)	Local Calls	Local Mins	Intl Calls	Intl Mins	Intl Active	Intl Plan	Extra International Charges
0	4444-BZPU	No	1	3	8.0	0.0	0.0	No	no	0.0
1	5676-PTZX	No	33	179	431.3	0.0	0.0	No	no	0.0
2	8532-ZEKQ	No	44	82	217.6	0.0	0.0	No	yes	0.0

3 rows × 29 columns

EDA

In [14]:

```
print(f"Total Number of Customers: {df['Customer ID'].nunique()}")
print(f"Total NUmber of Churned Customers: {df['Churn Label'].value_counts()
print(f"Rate of Churned Customers : {(df['Churn Label'].value_counts()[1]/df
```

```
Total Number of Customers: 6687
Total NUmber of Churned Customers: 1796
Rate of Churned Customers : 26.86 %
```

In [16]:

```
reason=df[df['Churn Label'] == 'Yes'][['Churn Label','Churn Reason','Churn C
```

In [17]:

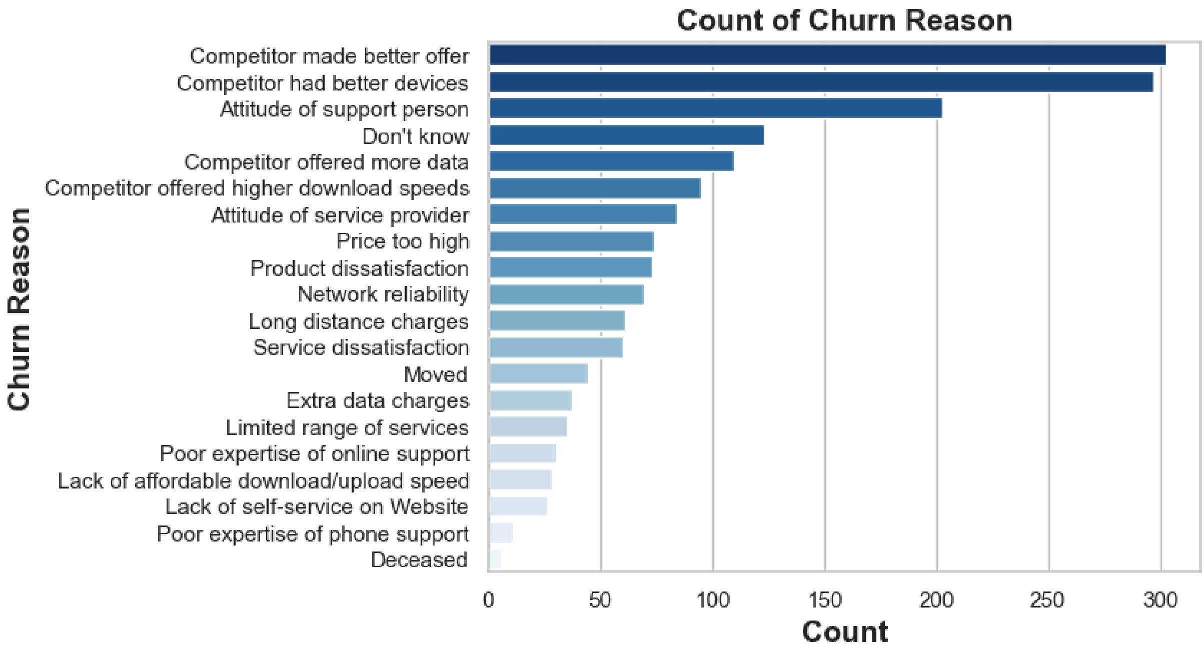
```
#Find Churn Reason
churn_reason=reason['Churn Reason'].value_counts().reset_index()
churn_reason.head(3)
```

Out[17]:

	Churn Reason	count
0	Competitor made better offer	303
1	Competitor had better devices	297
2	Attitude of support person	203

In [25]:

```
## Create a barplot which Will Show the Churn Reason
sns.barplot(x='count',y='Churn Reason',data=churn_reason,palette='Blues_r')
plt.title('Count of Churn Reason',fontweight='bold',fontsize=15)
plt.ylabel('Churn Reason',fontweight='bold',fontsize=15)
plt.xlabel('Count',fontweight='bold',fontsize=15)
plt.show()
```



In [19]:

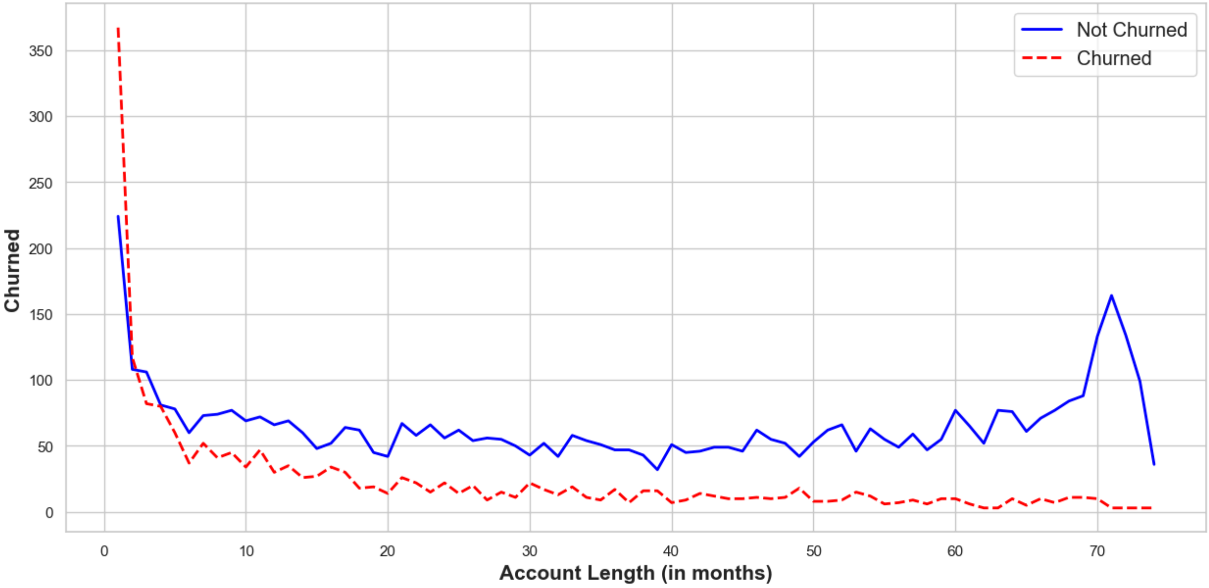
```
churned=df[df['Churn Label'] == 'Yes'].groupby('Account Length (in months)')
retained=df[df['Churn Label'] == 'No'].groupby('Account Length (in months)')
merged_df = pd.merge(churned, retained, on='Account Length (in months)', how
merged_df.head()
```

Out[19]:

	Account Length (in months)	Churned	Not Churned
0	1	367	224
1	2	117	108
2	3	82	106
3	4	80	81
4	5	60	78

In [20]:

```
plt.figure(figsize=(15,7))
sns.lineplot(x='Account Length (in months)',y='Not Churned',data=merged_df,l
sns.lineplot(x='Account Length (in months)',y='Churned',data=merged_df,label
plt.ylabel('Churned',fontweight='bold',fontsize=15)
plt.xlabel('Account Length (in months)',fontweight='bold',fontsize=15)
plt.legend(fontsize='large')
plt.show()
```

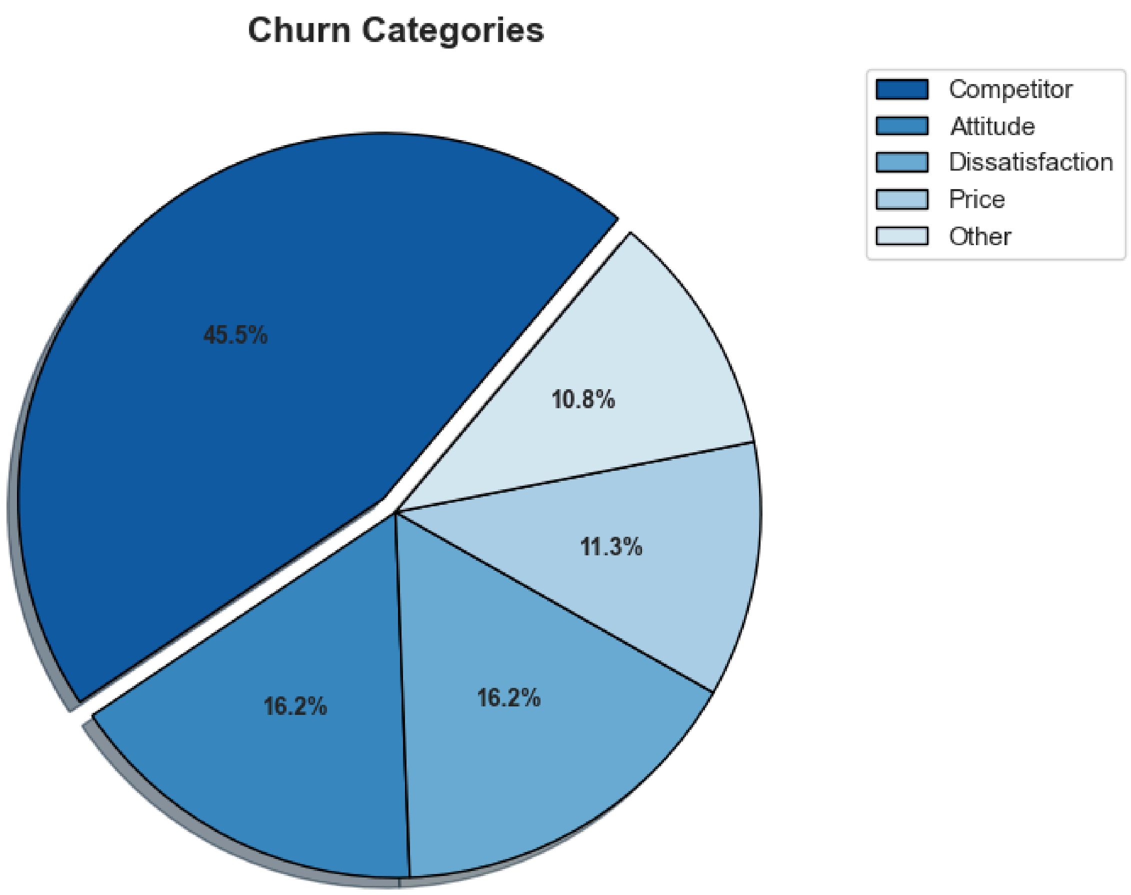


In [ ]:

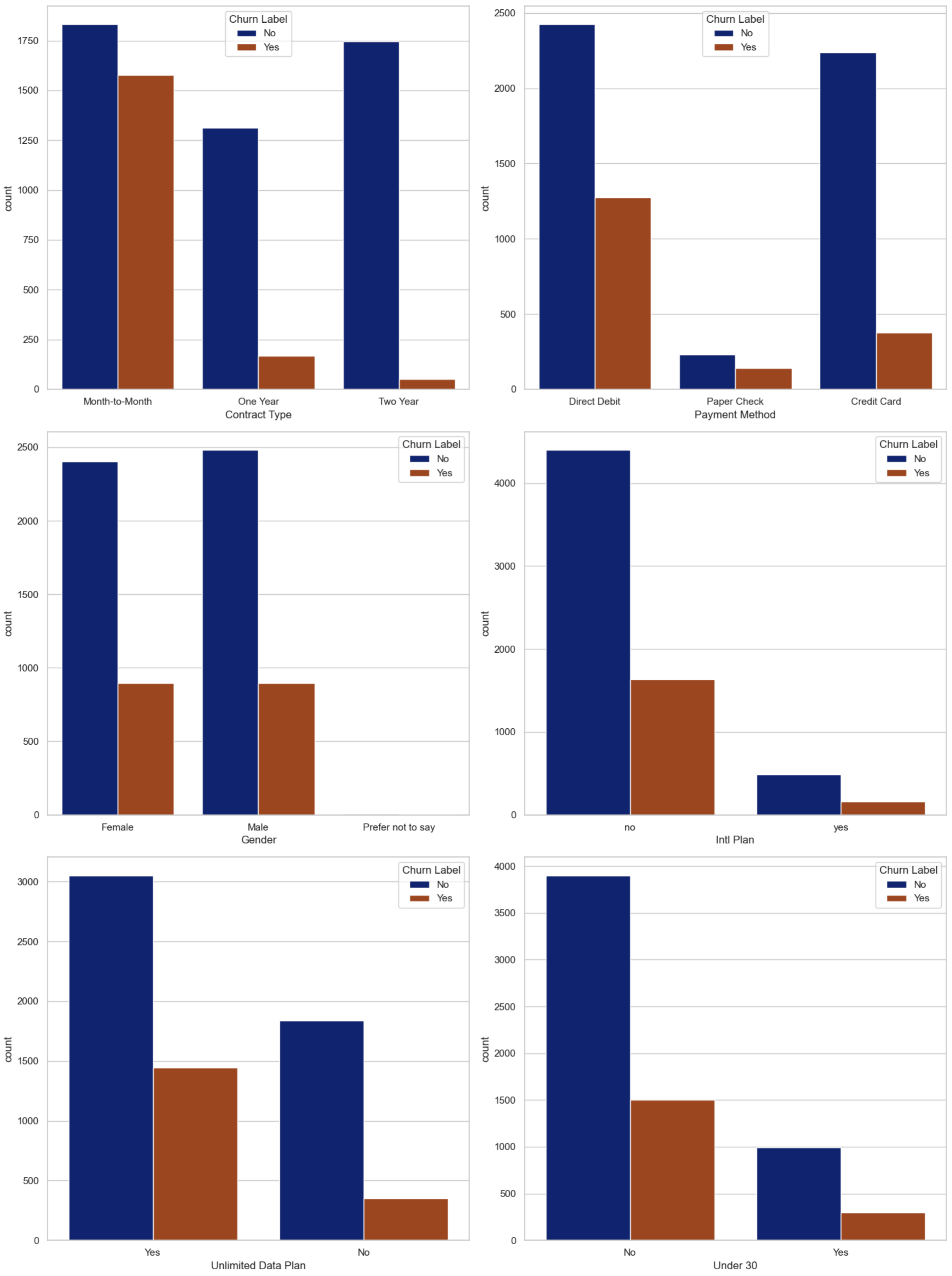
```
reason['Churn Category'].value_counts()
labels=reason['Churn Category'].value_counts().index
```

In [27]:

```
import matplotlib.pyplot as plt

# Example data
exp = [0.05, 0, 0, 0, 0]
col = ['#FF9999','#66B3FF','#99FF99','#FFCC99','#FFD700']


# Prepare data for Values and label
values = reason['Churn Category'].value_counts()
labels = values.index
exp = [0.05] + [0] * (len(values) - 1)  # explode only the first slice

# Generate professional blue palette
col = sns.color_palette("Blues_r", len(values))

# Plot pie chart
plt.figure(figsize=(7, 7))
plt.pie(x=values.values,
        explode=exp,
        autopct="%1.1f%%",
        shadow=True,
        radius=1,
        colors=col,
        textprops={"fontsize": 10, "fontweight": 'bold'},
        wedgeprops={"linewidth": 1, "edgecolor": 'black'},
        rotatelabels=False,
        startangle=50)

plt.legend(labels, bbox_to_anchor=(1, 1))
plt.title("Churn Categories", fontsize=15, fontweight='bold')
plt.show()
```



```
In [48]: # Set a dark professional palette
         with sns.color_palette("dark"):
             # Create a 3x2 grid of subplots
             fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(15, 20))

             # Plot 6 categorical features against Churn Label
             sns.countplot(x="Contract Type", data=df, hue="Churn Label", ax=axes[0,
             sns.countplot(x="Payment Method", data=df, hue="Churn Label", ax=axes[0,
             sns.countplot(x="Gender", data=df, hue="Churn Label", ax=axes[1, 0])
             sns.countplot(x="Intl Plan", data=df, hue="Churn Label", ax=axes[1, 1])
             sns.countplot(x="Unlimited Data Plan", data=df, hue="Churn Label", ax=ax
             sns.countplot(x="Under 30", data=df, hue="Churn Label", ax=axes[2, 1])

             # Set overall spacing
             plt.tight_layout()
             plt.show()
```



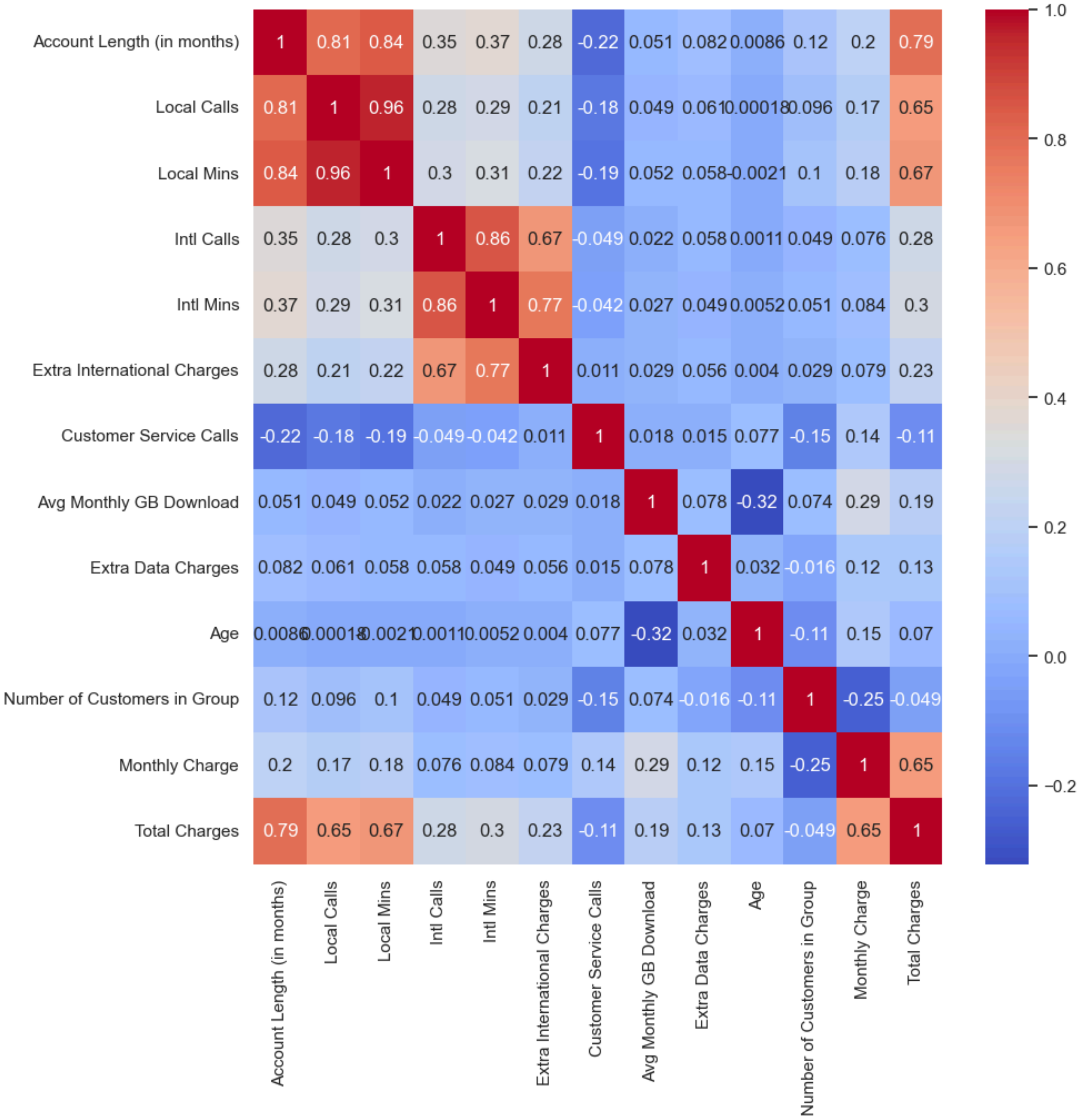
In [29]:

```
cor=df[['Account Length (in months)','Local Calls', 'Local Mins', 'Intl Call
    'Customer Service Calls','Avg Monthly GB Download', 'Extra Data Charges'
    'Number of Customers in Group', 'Monthly Charge', 'Total Charges']].corr
```

In [30]:

```
plt.figure(figsize=(10,10))
sns.heatmap(cor,annot=True,cmap='coolwarm')
```

```
Out[30]: <Axes: >
```



```
In [ ]:
```

```
In [31]: values=[df['Customer ID'].nunique(),df['Churn Label'].value_counts()[1],(df[
```

```
In [32]: metric=["Total Number of Customers","Total NUmber of Churned Customers","Rat
```

```
In [33]: summary=pd.DataFrame({'metric':metric,'Values':values},index=range(1,4))
         summary
```

Out[33]:

	metric	Values
1	Total Number of Customers	6687.00
2	Total NUmber of Churned Customers	1796.00
3	Rate of Churned Customers	26.86

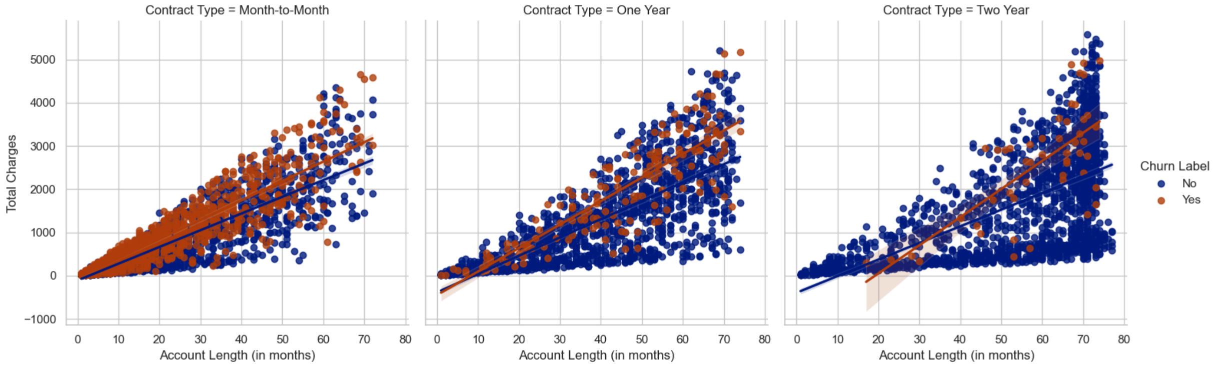
```
In [34]: df.groupby('Contract Type')['Avg Monthly GB Download'].mean().reset_index(na
```

Out[34]:

	Contract Type	Avg Monthly GB Download
0	Month-to-Month	7.154500
1	One Year	6.934415
2	Two Year	5.630495

```
In [35]: with sns.color_palette("dark"):
             plt.figure(figsize=(11,7))
             sns.lmplot(y='Total Charges',x='Account Length (in months)',data=df,hue=
                     col='Contract Type')
```

```
<Figure size 1100x700 with 0 Axes>
```



```
In [36]: print("Majorit of Our Customers")
         df['State'].value_counts().head().reset_index(name='No. of Customers')
```

```
Majorit of Our Customers
```

Out[36]:

	State	No. of Customers
0	WV	213
1	MN	168
2	NY	167
3	AL	161
4	OH	158

```
In [37]: df[df['Churn Label']=='Yes']['State'].value_counts().reset_index(name='Custc
```

Out[37]:

	State	Customer Churned
0	WV	57
1	OH	55
2	OR	48
3	MD	46
4	AL	46

- *Majority of our customers from West Virginia US state(WV)*
- *Most of the customers is also churned from WV*

Logistic Regression

In [38]:

```
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix, accuracy
```

In [39]:

```
column_selection=df[['Churn Label', 'Account Length (in months)',
       'Local Calls', 'Local Mins', 'Intl Calls', 'Intl Mins', 'Intl Active'
       'Intl Plan', 'Extra International Charges', 'Customer Service Calls',
       'Avg Monthly GB Download', 'Unlimited Data Plan', 'Extra Data Charges
        'Gender', 'Age', 'Under 30', 'Senior', 'Group',
       'Number of Customers in Group', 'Device Protection & Online Backup',
       'Contract Type', 'Payment Method', 'Monthly Charge', 'Total Charges',
       'Churn Category','Churn Reason']]
column_selection.head(3)
```

Out[39]:

	Churn Label	Account Length (in months)	Local Calls	Local Mins	Intl Calls	Intl Mins	Intl Active	Intl Plan	Extra International Charges	Customer Service Calls
0	No	1	3	8.0	0.0	0.0	No	no	0.0	0
1	No	33	179	431.3	0.0	0.0	No	no	0.0	0
2	No	44	82	217.6	0.0	0.0	No	yes	0.0	0

3 rows × 26 columns

In [49]:

```
features=column_selection.drop('Churn Label',axis=1)
X=pd.get_dummies(features,drop_first=True)
y=df['Churn Label']
```

```
In [50]: X_train, X_test, y_train, y_test = train_test_split(
         ...     X, y, test_size=0.5, random_state=42)
```

```
In [42]: logmodel=linear_model.LogisticRegression()
```

```
In [43]: logmodel.fit(X_train,y_train)
```

```
Out[43]: ▾ LogisticRegression
         ▶ Parameters
```

```
In [51]: predictions=logmodel.predict(X_test)
```

```
In [45]: print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

          No       0.89      0.93      0.91      2447
         Yes       0.79      0.68      0.73       897

    accuracy                           0.87      3344
   macro avg       0.84      0.81      0.82      3344
weighted avg       0.86      0.87      0.86      3344
```

```
In [52]: print(confusion_matrix(y_test,predictions))
```

```
[[2282  165]
 [ 283  614]]
```

```
In [47]: print(accuracy_score(y_test,predictions))
```

```
0.8660287081339713
```

ACCURACY 86.60 %

```
In [ ]:
```