In [ ]:

In [3]:
```
####################    Data preparation  ###########################

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sys
from scipy.io import arff



from scipy.io import arff
!pip install ydata-profiling
!jupyter nbextension enable --py widgetsnbextension
!pip install matplotlib
!pip install graphviz
```

```
Requirement already satisfied: ydata-profiling in ./anaconda3/lib/python3.1
0/site-packages (4.7.0)
Requirement already satisfied: pydantic>=2 in ./anaconda3/lib/python3.10/sit
e-packages (from ydata-profiling) (2.6.4)
Requirement already satisfied: scipy<1.12,>=1.4.1 in ./anaconda3/lib/python
3.10/site-packages (from ydata-profiling) (1.11.4)
Requirement already satisfied: phik<0.13,>=0.11.1 in ./anaconda3/lib/python
3.10/site-packages (from ydata-profiling) (0.12.4)
Requirement already satisfied: wordcloud>=1.9.1 in ./anaconda3/lib/python3.1
0/site-packages (from ydata-profiling) (1.9.3)
Requirement already satisfied: numpy<2,>=1.16.0 in ./anaconda3/lib/python3.1
0/site-packages (from ydata-profiling) (1.26.4)
Requirement already satisfied: matplotlib<3.9,>=3.2 in ./anaconda3/lib/pytho
n3.10/site-packages (from ydata-profiling) (3.8.3)
Requirement already satisfied: typeguard<5,>=4.1.2 in ./anaconda3/lib/python
3.10/site-packages (from ydata-profiling) (4.1.5)
Requirement already satisfied: seaborn<0.13,>=0.10.1 in ./anaconda3/lib/pyth
on3.10/site-packages (from ydata-profiling) (0.12.2)
Requirement already satisfied: htmlmin==0.1.12 in ./anaconda3/lib/python3.1
0/site-packages (from ydata-profiling) (0.1.12)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in ./anaconda3/lib/python
3.10/site-packages (from ydata-profiling) (3.1.3)
Requirement already satisfied: pandas!=1.4.0,<3,>1.1 in ./anaconda3/lib/pyth
on3.10/site-packages (from ydata-profiling) (2.2.1)
Requirement already satisfied: imagehash==4.3.1 in ./anaconda3/lib/python3.1
0/site-packages (from ydata-profiling) (4.3.1)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in ./anaconda3/lib/python
3.10/site-packages (from ydata-profiling) (6.0.1)
Requirement already satisfied: dacite>=1.8 in ./anaconda3/lib/python3.10/sit
e-packages (from ydata-profiling) (1.8.1)
Requirement already satisfied: multimethod<2,>=1.4 in ./anaconda3/lib/python
3.10/site-packages (from ydata-profiling) (1.11.2)
Requirement already satisfied: statsmodels<1,>=0.13.2 in ./anaconda3/lib/pyt
hon3.10/site-packages (from ydata-profiling) (0.14.1)
Requirement already satisfied: tqdm<5,>=4.48.2 in ./anaconda3/lib/python3.1
0/site-packages (from ydata-profiling) (4.65.0)
Requirement already satisfied: requests<3,>=2.24.0 in ./anaconda3/lib/python
3.10/site-packages (from ydata-profiling) (2.31.0)
Requirement already satisfied: numba<1,>=0.56.0 in ./anaconda3/lib/python3.1
0/site-packages (from ydata-profiling) (0.59.1)
Requirement already satisfied: visions[type_image_path]<0.7.7,>=0.7.5 in ./a
naconda3/lib/python3.10/site-packages (from ydata-profiling) (0.7.6)
Requirement already satisfied: pillow in ./anaconda3/lib/python3.10/site-pac
kages (from imagehash==4.3.1->ydata-profiling) (10.2.0)
Requirement already satisfied: PyWavelets in ./anaconda3/lib/python3.10/site
-packages (from imagehash==4.3.1->ydata-profiling) (1.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in ./anaconda3/lib/python3.1
0/site-packages (from jinja2<3.2,>=2.11.1->ydata-profiling) (2.1.3)
Requirement already satisfied: contourpy>=1.0.1 in ./anaconda3/lib/python3.1
0/site-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (1.2.0)
Requirement already satisfied: packaging>=20.0 in ./anaconda3/lib/python3.1
0/site-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in ./anaconda3/lib/python3.1
0/site-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (3.1.2)
Requirement already satisfied: kiwisolver>=1.3.1 in ./anaconda3/lib/python3.
10/site-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (1.4.5)
```

```
Requirement already satisfied: fonttools>=4.22.0 in ./anaconda3/lib/python3.
10/site-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (4.50.0)
Requirement already satisfied: cycler>=0.10 in ./anaconda3/lib/python3.10/si
te-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (0.12.1)
Requirement already satisfied: python-dateutil>=2.7 in ./anaconda3/lib/pytho
n3.10/site-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (2.8.2)
Requirement already satisfied: llvmlite<0.43,>=0.42.0dev0 in ./anaconda3/li
b/python3.10/site-packages (from numba<1,>=0.56.0->ydata-profiling) (0.42.0)
Requirement already satisfied: tzdata>=2022.7 in ./anaconda3/lib/python3.10/
site-packages (from pandas!=1.4.0,<3,>1.1->ydata-profiling) (2024.1)
Requirement already satisfied: pytz>=2020.1 in ./anaconda3/lib/python3.10/si
te-packages (from pandas!=1.4.0,<3,>1.1->ydata-profiling) (2023.3.post1)
Requirement already satisfied: joblib>=0.14.1 in ./anaconda3/lib/python3.10/
site-packages (from phik<0.13,>=0.11.1->ydata-profiling) (1.3.2)
Requirement already satisfied: typing-extensions>=4.6.1 in ./anaconda3/lib/p
ython3.10/site-packages (from pydantic>=2->ydata-profiling) (4.10.0)
Requirement already satisfied: annotated-types>=0.4.0 in ./anaconda3/lib/pyt
hon3.10/site-packages (from pydantic>=2->ydata-profiling) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in ./anaconda3/lib/pyth
on3.10/site-packages (from pydantic>=2->ydata-profiling) (2.16.3)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./anaconda3/lib/python
3.10/site-packages (from requests<3,>=2.24.0->ydata-profiling) (2.1.0)
Requirement already satisfied: idna<4,>=2.5 in ./anaconda3/lib/python3.10/si
te-packages (from requests<3,>=2.24.0->ydata-profiling) (3.4)
Requirement already satisfied: charset-normalizer<4,>=2 in ./anaconda3/lib/p
ython3.10/site-packages (from requests<3,>=2.24.0->ydata-profiling) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in ./anaconda3/lib/python
3.10/site-packages (from requests<3,>=2.24.0->ydata-profiling) (2024.2.2)
Requirement already satisfied: patsy>=0.5.4 in ./anaconda3/lib/python3.10/si
te-packages (from statsmodels<1,>=0.13.2->ydata-profiling) (0.5.6)
Requirement already satisfied: networkx>=2.4 in ./anaconda3/lib/python3.10/s
ite-packages (from visions[type_image_path]<0.7.7,>=0.7.5->ydata-profiling)
(3.2.1)
Requirement already satisfied: attrs>=19.3.0 in ./anaconda3/lib/python3.10/s
ite-packages (from visions[type_image_path]<0.7.7,>=0.7.5->ydata-profiling)
(23.1.0)
Requirement already satisfied: six in ./anaconda3/lib/python3.10/site-packag
es (from patsy>=0.5.4->statsmodels<1,>=0.13.2->ydata-profiling) (1.16.0)
Enabling notebook extension jupyter-js-widgets/extension...
      - Validating: OK
Requirement already satisfied: matplotlib in ./anaconda3/lib/python3.10/site
-packages (3.8.3)
Requirement already satisfied: fonttools>=4.22.0 in ./anaconda3/lib/python3.
10/site-packages (from matplotlib) (4.50.0)
Requirement already satisfied: kiwisolver>=1.3.1 in ./anaconda3/lib/python3.
10/site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: pyparsing>=2.3.1 in ./anaconda3/lib/python3.1
0/site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: pillow>=8 in ./anaconda3/lib/python3.10/site-
packages (from matplotlib) (10.2.0)
Requirement already satisfied: python-dateutil>=2.7 in ./anaconda3/lib/pytho
n3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in ./anaconda3/lib/python3.1
0/site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in ./anaconda3/lib/python3.10/si
te-packages (from matplotlib) (0.12.1)
```

```
Requirement already satisfied: numpy<2,>=1.21 in ./anaconda3/lib/python3.10/
site-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in ./anaconda3/lib/python3.1
0/site-packages (from matplotlib) (23.2)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.10/site-p
ackages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: graphviz in ./anaconda3/lib/python3.10/site-p
ackages (0.20.2)
```

In [4]:

```python
import pandas as pd
from scipy.io import arff

data_file = "churn.arff"

# Load ARFF file
data, meta = arff.loadarff(data_file)

# Convert data to DataFrame
df = pd.DataFrame(data)

# Decode object columns if needed
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = df[col].str.decode('utf-8')

# Look at loaded data and data types
print(df.dtypes)
```

```
State                              object
Account Length                    float64
Area Code                          object
Phone Number                       object
Inter Plan                         object
VoiceMail Plan                     object
No of Vmail Mesgs                 float64
Total Day Min                     float64
Total Day calls                   float64
Total Day Charge                  float64
Total Evening Min                 float64
Total Evening Calls               float64
Total Evening Charge              float64
Total Night Minutes               float64
Total Night Calls                 float64
Total Night Charge                float64
Total Int Min                     float64
Total Int Calls                   float64
Total Int Charge                  float64
No of Calls Customer Service      float64
Churn                              object
dtype: object
```

In [ ]:

In [ ]:

In [5]:
```
# Display the first few rows of the DataFrame
df.head(10)
```

Out[5]:

	State	Account Length	Area Code	Phone Number	Inter Plan	VoiceMail Plan	No of Vmail Mesgs	Total Day Min	Total Day calls	Total Day Charge	...
0	OH	107.0	A415	371-7191	no	yes	26.0	161.6	123.0	27.47	...
1	NJ	137.0	A415	358-1921	no	no	0.0	243.4	114.0	41.38	...
2	OH	84.0	A408	375-9999	yes	no	0.0	299.4	71.0	50.90	...
3	OK	75.0	A415	330-6626	yes	no	0.0	166.7	113.0	28.34	...
4	AL	118.0	A510	391-8027	yes	no	0.0	223.4	98.0	37.98	...
5	MA	121.0	A510	355-9993	no	yes	24.0	218.2	88.0	37.09	...
6	MO	147.0	A415	329-9001	yes	no	0.0	157.0	79.0	26.69	...
7	LA	117.0	A408	335-4719	no	no	0.0	184.5	97.0	31.37	...
8	WV	141.0	A415	330-8173	yes	yes	37.0	258.6	84.0	43.96	...
9	IN	65.0	A415	329-6603	no	no	0.0	129.1	137.0	21.95	...

10 rows × 21 columns

In [6]:
```
# look at meta information about data, such as null values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 21 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   State                         3333 non-null   object 
 1   Account Length                3333 non-null   float64
 2   Area Code                     3333 non-null   object 
 3   Phone Number                  3333 non-null   object 
 4   Inter Plan                    3333 non-null   object 
 5   VoiceMail Plan                3333 non-null   object 
 6   No of Vmail Mesgs             3333 non-null   float64
 7   Total Day Min                 3333 non-null   float64
 8   Total Day calls               3333 non-null   float64
 9   Total Day Charge              3333 non-null   float64
 10  Total Evening Min             3333 non-null   float64
 11  Total Evening Calls           3333 non-null   float64
 12  Total Evening Charge          3333 non-null   float64
 13  Total Night Minutes           3333 non-null   float64
 14  Total Night Calls             3333 non-null   float64
 15  Total Night Charge            3333 non-null   float64
 16  Total Int Min                 3333 non-null   float64
 17  Total Int Calls               3333 non-null   float64
 18  Total Int Charge              3333 non-null   float64
 19  No of Calls Customer Service  3333 non-null   float64
 20  Churn                         3333 non-null   object 
dtypes: float64(15), object(6)
memory usage: 546.9+ KB
```

In [7]:
```
# Find max, min, mean and standard deviation of attributes.

df.describe()
```

Out[7]:

	Account Length	No of Vmail Mesgs	Total Day Min	Total Day calls	Total Day Charge	T Evening
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000
mean	101.064806	8.099010	179.775098	100.435644	30.562307	200.980
std	39.822106	13.688365	54.467389	20.069084	9.259435	50.713
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	74.000000	0.000000	143.700000	87.000000	24.430000	166.600
50%	101.000000	0.000000	179.400000	101.000000	30.500000	201.400
75%	127.000000	20.000000	216.400000	114.000000	36.790000	235.300
max	243.000000	51.000000	350.800000	165.000000	59.640000	363.700

In [8]:
```
df.shape
```

Out[8]: (3333, 21)

In [9]:
```
column_names = df.columns
column_names
```

Out[9]:
```
Index(['State', 'Account Length', 'Area Code', 'Phone Number', 'Inter Pla
n',
       'VoiceMail Plan', 'No of Vmail Mesgs', 'Total Day Min',
       'Total Day calls', 'Total Day Charge', 'Total Evening Min',
       'Total Evening Calls', 'Total Evening Charge', 'Total Night Minute
s',
       'Total Night Calls', 'Total Night Charge', 'Total Int Min',
       'Total Int Calls', 'Total Int Charge', 'No of Calls Customer Servic
e',
       'Churn'],
      dtype='object')
```

In [10]:
```
# Finding missing values

df.isnull().sum()
```

Out[10]:
```
State                          0
Account Length                 0
Area Code                      0
Phone Number                   0
Inter Plan                     0
VoiceMail Plan                 0
No of Vmail Mesgs              0
Total Day Min                  0
Total Day calls                0
Total Day Charge               0
Total Evening Min              0
Total Evening Calls            0
Total Evening Charge           0
Total Night Minutes            0
Total Night Calls              0
Total Night Charge             0
Total Int Min                  0
Total Int Calls                0
Total Int Charge               0
No of Calls Customer Service   0
Churn                          0
dtype: int64
```

In [11]:
```
# Handle duplicates

print(df.drop_duplicates(inplace=True))
```

```
None
```

In [12]:
```
# Identify numerical variables

numeric_variables = df.select_dtypes(include=['int64', 'float64']).columns.t

# Print the list of numerical variables

print("Numerical Variables:")
print(numeric_variables)
```

```
Numerical Variables:
['Account Length', 'No of Vmail Mesgs', 'Total Day Min', 'Total Day calls', 'Total Day Charge', 'Total Evening Min', 'Total Evening Calls', 'Total Evening Charge', 'Total Night Minutes', 'Total Night Calls', 'Total Night Charge', 'Total Int Min', 'Total Int Calls', 'Total Int Charge', 'No of Calls Customer Service']
```

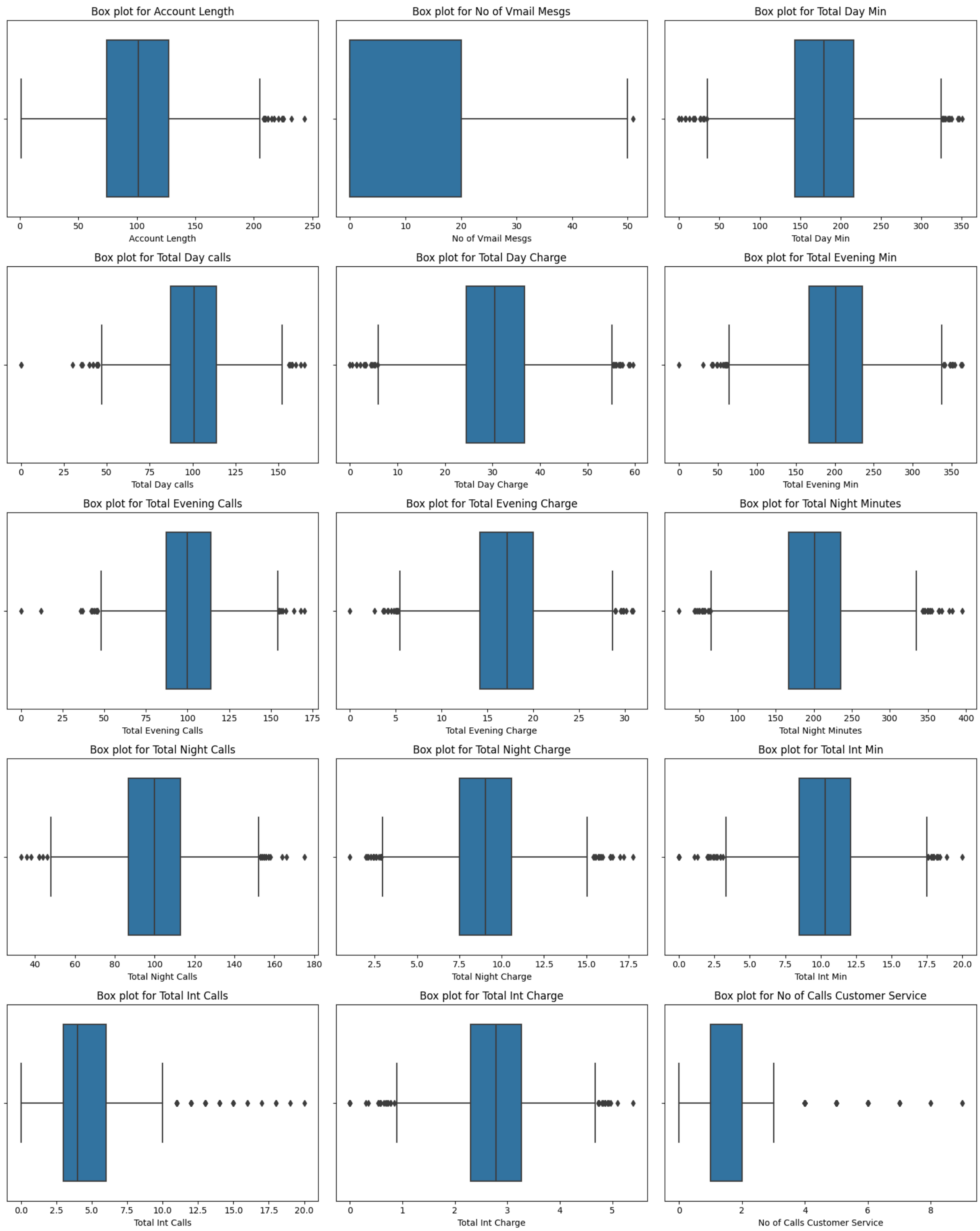
In [13]:

```python
# Identify categorical variables

categorical_variables = df.select_dtypes(include=['object']).columns.tolist(

# Print the list of categorical variables

print("Categorical Variables:")
print(categorical_variables)
```

```
Categorical Variables:
['State', 'Area Code', 'Phone Number', 'Inter Plan', 'VoiceMail Plan', 'Churn']
```

In [14]:

```python
#Determine any outlier values(records)for numeric attributes and create box



# Select numeric attributes
numeric_attributes = df.select_dtypes(include=['int64', 'float64']).columns

# Calculate the number of rows needed for the subplots

num_attributes = len(numeric_attributes)
num_rows = (num_attributes // 3) + (num_attributes % 3 > 0)

# Create box plots for numeric attributes

plt.figure(figsize=(16, 4 * num_rows))
for i, column in enumerate(numeric_attributes, 1):
    plt.subplot(num_rows, 3, i)
    sns.boxplot(x=df[column])
    plt.title(f'Box plot for {column}')

plt.tight_layout()
plt.show()
```



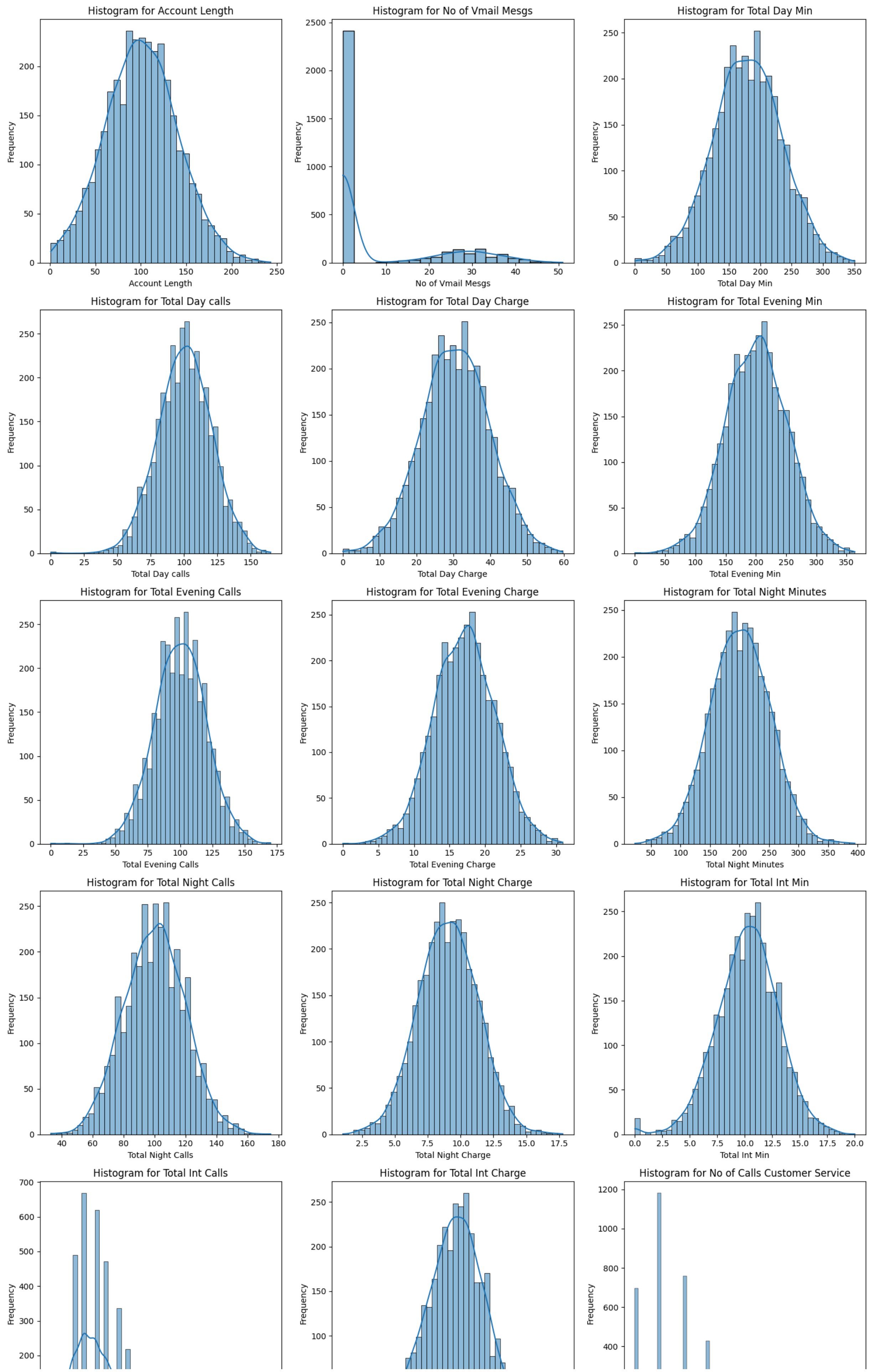
In [15]:

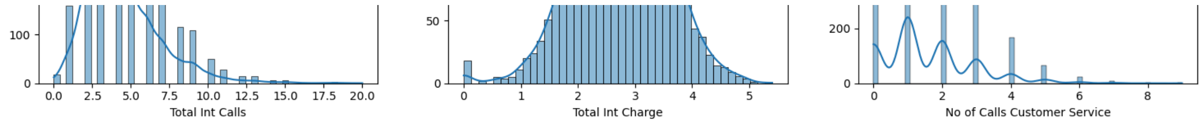
```
# To analyze the distribution of numeric attributes and create Histogram



num_attributes = len(numeric_attributes)


num_cols = 3  # Number of columns in the subplot grid
num_rows = (num_attributes // num_cols) + (num_attributes % num_cols > 0)
```

```
plt.figure(figsize=(15, 5 * num_rows))

# Plot histograms for numeric attributes

for i, column in enumerate(numeric_attributes, 1):
    plt.subplot(num_rows, num_cols, i)
    sns.histplot(df[column], kde=True)
    plt.title(f'Histogram for {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```

```
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

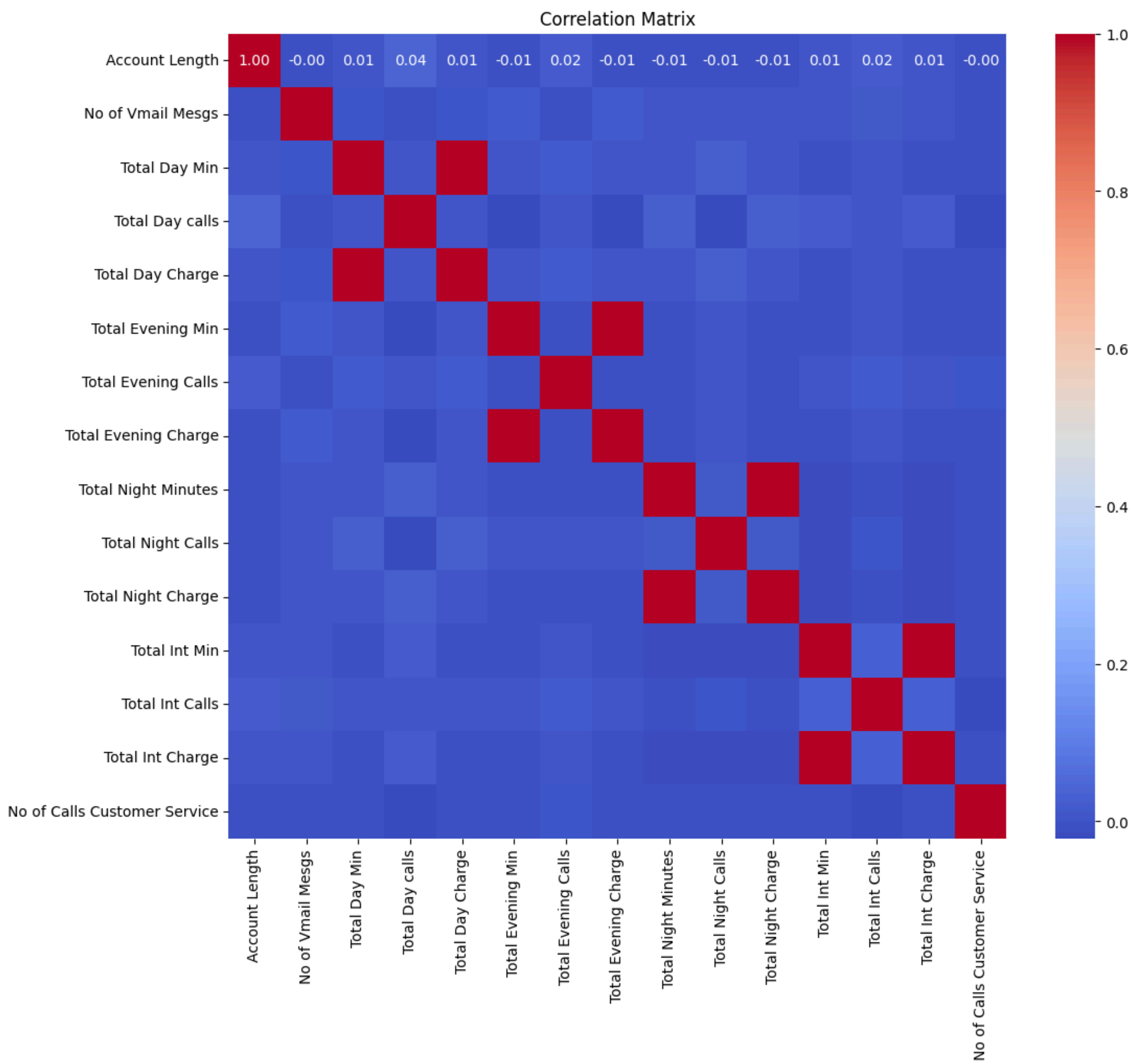
```
/Users/zhila/anaconda3/lib/python3.10/site-packages/seaborn/_oldcore.py:111
9: FutureWarning: use_inf_as_na option is deprecated and will be removed in
a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```


Histogram for Account Length
Account Length
Frequency
Histogram for No of Vmail Mesgs
No of Vmail Mesgs
Histogram for Total Day Min
Total Day Min
Histogram for Total Day calls
Total Day calls
Histogram for Total Day Charge
Total Day Charge
Histogram for Total Evening Min
Total Evening Min
Histogram for Total Evening Calls
Total Evening Calls
Histogram for Total Evening Charge
Total Evening Charge
Histogram for Total Night Minutes
Total Night Minutes
Histogram for Total Night Calls
Total Night Calls
Histogram for Total Night Charge
Total Night Charge
Histogram for Total Int Min
Total Int Min
Histogram for Total Int Calls
Histogram for Total Int Charge
Histogram for No of Calls Customer Service



In [11]:

```
# Select only numerical columns
numeric_df = df.select_dtypes(include=['float64', 'int64'])

# Calculate correlation matrix
correlation_matrix = numeric_df.corr()

# Plot correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```



In [16]:

```
####### Construct contingency table and perform chi−squared test to assess a



from scipy.stats import chi2_contingency
```

```
# Contingency table
contingency_table = pd.crosstab(df['State'], df['Churn'])

# Chi-squared test
chi2, p, _, _ = chi2_contingency(contingency_table)
print(f"Chi-squared p-value: {p}")
```

```
Chi-squared p-value: 0.002296221552011188
```

In [17]:

```
####  Determine whether the dataset has an imbalanced class distribution ###


# Check the class distribution of the target variable

class_distribution = df['Churn'].value_counts()

# Print the class distribution

print("Class Distribution:")

print(class_distribution)

# Check if the dataset has an imbalanced class distribution

is_imbalanced = class_distribution.nunique() > 1

# Print the result

if is_imbalanced:
    print("The dataset has an imbalanced class distribution.")
else:
    print("The dataset has a balanced class distribution.")
```

```
Class Distribution:
Churn
FALSE    2850
TRUE      483
Name: count, dtype: int64
The dataset has an imbalanced class distribution.
```

In [18]:

```
#Let's create a list for our categorical columns for  Churn data set



cat_cols = ["State", "Area Code", "Phone Number", "Inter Plan", "VoiceMail P


# Create a copy of the data frame in memory with a different name
df_onehot = df.copy()

# Convert only categorical variables/features to dummy/one-hot features
df_onehot = pd.get_dummies(df_onehot, columns=cat_cols, prefix=cat_cols)

# Print the dataset
print(df_onehot)
```

```
# Create a copy of the data frame in memory with a different name
df_onehot=df.copy()

#convert only categorical variables/features to dummy/one-hot features
df_onehot = pd.get_dummies(df, columns=cat_cols, prefix = cat_cols)

#print the dataset
df_onehot
```

```
      Account Length  No of Vmail Mesgs  Total Day Min  Total Day calls  \
0              107.0               26.0          161.6            123.0
1              137.0                0.0          243.4            114.0
2               84.0                0.0          299.4             71.0
3               75.0                0.0          166.7            113.0
4              118.0                0.0          223.4             98.0
...              ...                ...            ...              ...
3328            68.0                0.0          231.1             57.0
3329            28.0                0.0          180.8            109.0
3330           184.0                0.0          213.8            105.0
3331            74.0               25.0          234.4            113.0
3332           128.0               25.0          265.1            110.0

      Total Day Charge  Total Evening Min  Total Evening Calls  \
0                27.47              195.5                103.0
1                41.38              121.2                110.0
2                50.90               61.9                 88.0
3                28.34              148.3                122.0
4                37.98              220.6                101.0
...                ...                ...                  ...
3328             39.29              153.4                 55.0
3329             30.74              288.8                 58.0
3330             36.35              159.6                 84.0
3331             39.85              265.9                 82.0
3332             45.07              197.4                 99.0

      Total Evening Charge  Total Night Minutes  Total Night Calls  ...  \
0                    16.62                254.4              103.0  ...
1                    10.30                162.6              104.0  ...
2                     5.26                196.9               89.0  ...
3                    12.61                186.9              121.0  ...
4                    18.75                203.9              118.0  ...
...                    ...                  ...                ...  ...
3328                 13.04                191.3              123.0  ...
3329                 24.55                191.9               91.0  ...
3330                 13.57                139.2              137.0  ...
3331                 22.60                241.4               77.0  ...
3332                 16.78                244.7               91.0  ...

      Phone Number_422-6690  Phone Number_422-7728  Phone Number_422-8268  \
0                     False                  False                  False
1                     False                  False                  False
2                     False                  False                  False
3                     False                  False                  False
4                     False                  False                  False
...                     ...                    ...                    ...
3328                  False                  False                  False
3329                  False                  False                  False
3330                  False                  False                  False
3331                  False                  False                  False
3332                  False                  False                  False

      Phone Number_422-8333  Phone Number_422-8344 Phone Number_422-9964  \
0                     False                  False                 False
1                     False                  False                 False
2                     False                  False                 False
```

```
3                    False                  False                  False
4                    False                  False                  False
...                    ...                    ...                    ...
3328                 False                  False                  False
3329                 False                  False                  False
3330                 False                  False                  False
3331                 False                  False                  False
3332                 False                  False                  False

      Inter Plan_no  Inter Plan_yes  VoiceMail Plan_no  VoiceMail Plan_yes
0              True           False              False                True
1              True           False               True               False
2             False            True               True               False
3             False            True               True               False
4             False            True               True               False
...             ...             ...                ...                 ...
3328           True           False               True               False
3329           True           False               True               False
3330          False            True               True               False
3331           True           False              False                True
3332           True           False              False                True

[3333 rows x 3407 columns]
```

Out[18]:

	Account Length	No of Vmail Mesgs	Total Day Min	Total Day calls	Total Day Charge	Total Evening Min	Total Evening Calls	Total Evening Charge	Total Night Minutes	T N C
0	107.0	26.0	161.6	123.0	27.47	195.5	103.0	16.62	254.4	1(
1	137.0	0.0	243.4	114.0	41.38	121.2	110.0	10.30	162.6	1(
2	84.0	0.0	299.4	71.0	50.90	61.9	88.0	5.26	196.9	ε
3	75.0	0.0	166.7	113.0	28.34	148.3	122.0	12.61	186.9	1
4	118.0	0.0	223.4	98.0	37.98	220.6	101.0	18.75	203.9	1
...	...	...	...	...	...	...	...	...	...	
3328	68.0	0.0	231.1	57.0	39.29	153.4	55.0	13.04	191.3	1:
3329	28.0	0.0	180.8	109.0	30.74	288.8	58.0	24.55	191.9	
3330	184.0	0.0	213.8	105.0	36.35	159.6	84.0	13.57	139.2	1
3331	74.0	25.0	234.4	113.0	39.85	265.9	82.0	22.60	241.4	
3332	128.0	25.0	265.1	110.0	45.07	197.4	99.0	16.78	244.7	

3333 rows × 3407 columns

In [ ]:

In [19]:

```
#Repeat the train test set split

from sklearn.model_selection import train_test_split
```

```
class_col_name="Churn"
one_hot_feature_names=df_onehot.columns[df_onehot.columns != class_col_name]
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(df_onehot.loc[:, one_hot
```

In [20]:

```
# Repeat Naive Bayes modeling
from sklearn.naive_bayes import MultinomialNB

#Create a MultiNomial NB Classifier
nb = MultinomialNB()

#Train the model using the training sets
nb.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = nb.predict(X_test)
print ("Succesfully done..")
```

```
Succesfully done..
```

In [21]:

```
print("Number of features used ",nb.n_features_in_)
print("Classes ",nb.classes_)
print("Number of records for classes ",nb.class_count_)
print("Log prior probability for classes ", nb.class_log_prior_)
print("Log conditional probability for each feature given a class\n",nb.feat
```

```
Number of features used  3406
Classes  ['FALSE' 'TRUE']
Number of records for classes  [2000.  333.]
Log prior probability for classes  [-0.15400781 -1.94676778]
Log conditional probability for each feature given a class
 [[-2.35240444 -4.81236658 -1.8039874  ... -9.66722848 -7.32471068
  -8.18058615]
 [-2.38746885 -5.36209714 -1.70638185 ... -8.24401576 -7.2151181
  -8.7786926 ]]
```

In [23]:

```
from sklearn.metrics import confusion_matrix
cf=confusion_matrix(y_test, y_pred)
print ("Confusion Matrix")
print(cf)
tn, fp, fn, tp=cf.ravel()
print ("TP: ", tp,", FP: ", fp,", TN: ", tn,", FN:", fn)
```

```
Confusion Matrix
[[758  92]
 [ 84  66]]
TP:  66 , FP:  92 , TN:  758 , FN: 84
```

In [24]:

```
from sklearn.metrics import classification_report
from sklearn import metrics

print(classification_report(y_test, y_pred))
```

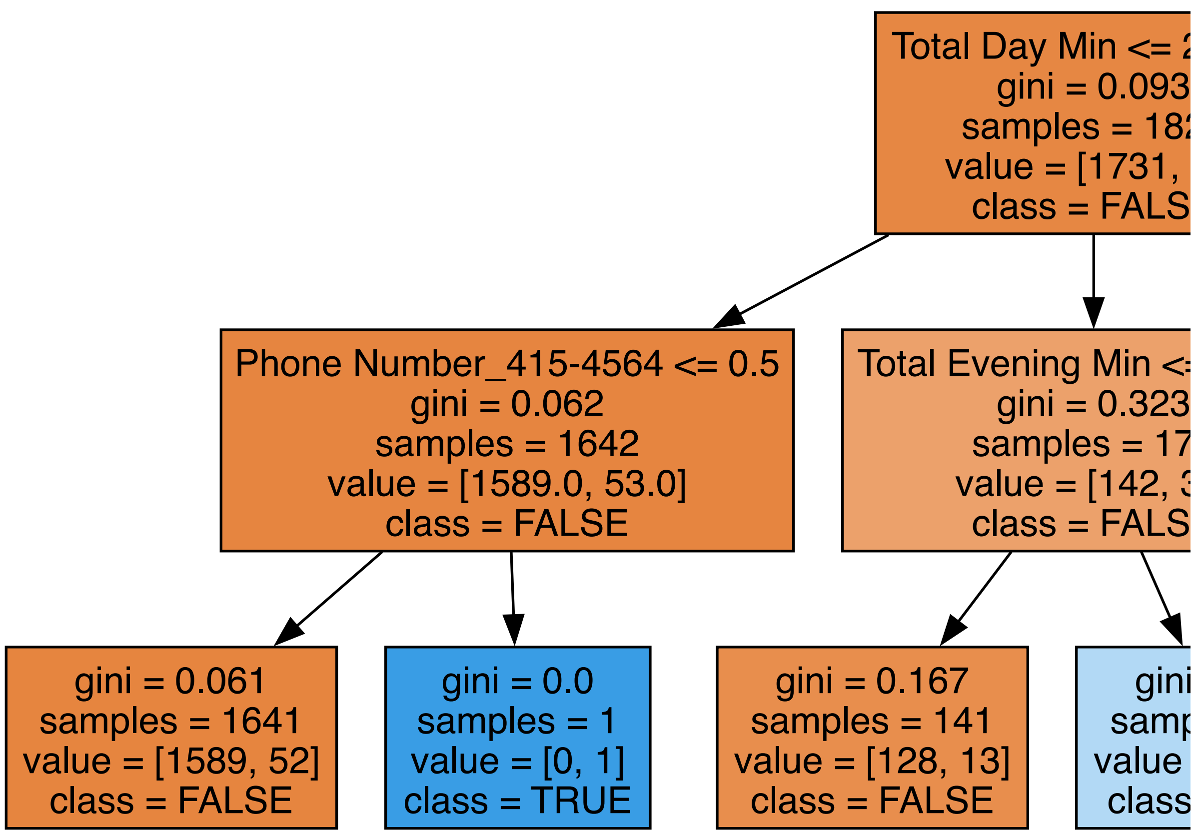
```
              precision    recall  f1-score   support

       FALSE       0.90      0.89      0.90       850
        TRUE       0.42      0.44      0.43       150

    accuracy                           0.82      1000
   macro avg       0.66      0.67      0.66      1000
weighted avg       0.83      0.82      0.83      1000
```

In [25]:

```
from sklearn import tree
clf = tree.DecisionTreeClassifier(max_depth=5)
clf = clf.fit(X_train, y_train)
import graphviz
#Get unique class values to display on the tree
class_values=df_onehot[class_col_name].unique()
print ("class Names",class_values)

dot_data = tree.export_graphviz(clf, out_file=None,
                                feature_names=one_hot_feature_names,
                                class_names=class_values,
                                filled=True)

# Draw graph
graph = graphviz.Source(dot_data, format="png")
graph
```

```
class Names ['FALSE' 'TRUE']
```

Out[25]:



In [21]:
```
# Perform prediction on the test set
y_pred = clf.predict(X_test)
```

In [26]:
```
# Get classification report
from sklearn.metrics import classification_report
from sklearn import metrics

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       FALSE       0.90      0.89      0.90       850
        TRUE       0.42      0.44      0.43       150

    accuracy                           0.82      1000
   macro avg       0.66      0.67      0.66      1000
weighted avg       0.83      0.82      0.83      1000
```

In [27]:
```
from ydata_profiling import ProfileReport

# Generate the data profiling report
report = ProfileReport(df)
report
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Overview

Dataset statistics

Number of variables	21
Number of observations	3333
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	546.9 KiB
Average record size in memory	168.0 B

Variable types

Text	2
Numeric	15
Categorical	1
Boolean	3

Alerts

`Inter Plan` is highly imbalanced (54.1%)	**Imbalance**
`Phone Number` has unique values	**Unique**

Out[27]:

In [ ]: