Data prechecks

In [429]:

```
import numpy as np
import pandas as pd
data1=pd.read_csv("E:\\1 python\\5 Real timeProjects\\Churn_MV.csv",na_values=["",," ","?","
```

In [422]:

```
print(data1.shape)
print(data1.size)
print(data1.ndim)
print(data1.columns)
```

```
(6666, 22)
146652
2
Index(['Account Length', 'VMail Message', 'Day Mins', 'Eve Mins', 'Night Min
s',
       'Intl Mins', 'CustServ Calls', 'Churn', 'Intl Plan', 'VMail Plan',
       'Day Calls', 'Day Charge', 'Daily Charges MV', 'Eve Calls',
       'Eve Charge', 'Night Calls', 'Night Charge', 'Intl Calls',
       'Intl Charge', 'State', 'Area Code', 'Phone'],
      dtype='object')
```

In [423]:

```
print(data1.count())
print(data1.nunique())
print(data1.describe())
print(data1.info())
```

```
Account Length      3333
VMail Message       3333
Day Mins            3333
Eve Mins            3333
Night Mins          3333
Intl Mins           3333
CustServ Calls      3333
Churn               3333
Intl Plan           3333
VMail Plan          3333
Day Calls           3333
Day Charge          3333
Daily Charges MV    3283
Eve Calls           3333
Eve Charge          3333
Night Calls         3333
Night Charge        3333
Intl Calls          3333
Intl Charge         3333
```

In [424]:

```
print(data1.head(10))
print(data1.tail(10))
```

```
   Account Length  VMail Message  Day Mins  Eve Mins  Night Mins  Intl Min
s  \
0             NaN            NaN       NaN       NaN         NaN        Na
N
1           128.0           25.0     265.1     197.4       244.7       10.
0
2             NaN            NaN       NaN       NaN         NaN        Na
N
3           107.0           26.0     161.6     195.5       254.4       13.
7
4             NaN            NaN       NaN       NaN         NaN        Na
N
5           137.0            0.0     243.4     121.2       162.6       12.
2
6             NaN            NaN       NaN       NaN         NaN        Na
N
7            84.0            0.0     299.4      61.9       196.9        6.
6
8             NaN            NaN       NaN       NaN         NaN        Na
N
9            75.0            0.0     166.7     148.3       186.9       10.
1

   CustServ Calls  Churn  Intl Plan  VMail Plan  ...  Daily Charges MV  \
0             NaN    NaN        NaN         NaN  ...               NaN
1             1.0    0.0        0.0         1.0  ...             45.07
2             NaN    NaN        NaN         NaN  ...               NaN
3             1.0    0.0        0.0         1.0  ...             27.47
4             NaN    NaN        NaN         NaN  ...               NaN
5             0.0    0.0        0.0         0.0  ...             41.38
6             NaN    NaN        NaN         NaN  ...               NaN
7             2.0    0.0        1.0         0.0  ...             50.90
8             NaN    NaN        NaN         NaN  ...               NaN
9             3.0    0.0        1.0         0.0  ...             28.34

   Eve Calls  Eve Charge  Night Calls  Night Charge  Intl Calls  Intl Char
ge  \
0        NaN         NaN          NaN           NaN         NaN          N
aN
1       99.0       16.78         91.0         11.01         3.0         2.
70
2        NaN         NaN          NaN           NaN         NaN          N
aN
3      103.0       16.62        103.0         11.45         3.0         3.
70
4        NaN         NaN          NaN           NaN         NaN          N
aN
5      110.0       10.30        104.0          7.32         5.0         3.
29
6        NaN         NaN          NaN           NaN         NaN          N
aN
7       88.0        5.26         89.0          8.86         7.0         1.
78
8        NaN         NaN          NaN           NaN         NaN          N
aN
9      122.0       12.61        121.0          8.41         3.0         2.
```

```
73

   State  Area Code     Phone
0    NaN        NaN       NaN
1     KS      415.0  382-4657
2    NaN        NaN       NaN
3     OH      415.0  371-7191
4    NaN        NaN       NaN
5     NJ      415.0  358-1921
6    NaN        NaN       NaN
7     OH      408.0  375-9999
8    NaN        NaN       NaN
9     OK      415.0  330-6626

[10 rows x 22 columns]
      Account Length  VMail Message  Day Mins  Eve Mins  Night Mins  \
6656             NaN            NaN       NaN       NaN         NaN
6657           192.0           36.0     156.2     215.5       279.1
6658             NaN            NaN       NaN       NaN         NaN
6659            68.0            0.0     231.1     153.4       191.3
6660             NaN            NaN       NaN       NaN         NaN
6661            28.0            0.0     180.8     288.8       191.9
6662             NaN            NaN       NaN       NaN         NaN
6663           184.0            0.0     213.8     159.6       139.2
6664             NaN            NaN       NaN       NaN         NaN
6665            74.0           25.0     234.4     265.9       241.4

      Intl Mins  CustServ Calls  Churn  Intl Plan  VMail Plan  ...  \
6656        NaN             NaN    NaN        NaN         NaN  ...
6657        9.9             2.0    0.0        0.0         1.0  ...
6658        NaN             NaN    NaN        NaN         NaN  ...
6659        9.6             3.0    0.0        0.0         0.0  ...
6660        NaN             NaN    NaN        NaN         NaN  ...
6661       14.1             2.0    0.0        0.0         0.0  ...
6662        NaN             NaN    NaN        NaN         NaN  ...
6663        5.0             2.0    0.0        1.0         0.0  ...
6664        NaN             NaN    NaN        NaN         NaN  ...
6665       13.7             0.0    0.0        0.0         1.0  ...

      Daily Charges MV  Eve Calls  Eve Charge  Night Calls  Night Charge
\
6656               NaN        NaN         NaN          NaN           NaN
6657             26.55      126.0       18.32         83.0         12.56
6658               NaN        NaN         NaN          NaN           NaN
6659             39.29       55.0       13.04        123.0          8.61
6660               NaN        NaN         NaN          NaN           NaN
6661             30.74       58.0       24.55         91.0          8.64
6662               NaN        NaN         NaN          NaN           NaN
6663             36.35       84.0       13.57        137.0          6.26
6664               NaN        NaN         NaN          NaN           NaN
6665             39.85       82.0       22.60         77.0         10.86

      Intl Calls  Intl Charge  State  Area Code     Phone
6656         NaN          NaN    NaN        NaN       NaN
6657         6.0         2.67     AZ      415.0  414-4276
6658         NaN          NaN    NaN        NaN       NaN
6659         4.0         2.59     WV      415.0  370-3271
6660         NaN          NaN    NaN        NaN       NaN
6661         6.0         3.81     RI      510.0  328-8230
6662         NaN          NaN    NaN        NaN       NaN
6663        10.0         1.35     CT      510.0  364-6381
```

```
6664         NaN          NaN    NaN        NaN        NaN
6665         4.0         3.70     TN      415.0   400-4344

[10 rows x 22 columns]
```

Data type Conversion

In [425]:

```
print(len(data1))
a=data1[['Churn','Intl Plan','VMail Plan','Area Code']]
print(a)
print(a.columns)
len(a)
```

```
6666
      Churn  Intl Plan  VMail Plan  Area Code
0       NaN        NaN         NaN        NaN
1       0.0        0.0         1.0      415.0
2       NaN        NaN         NaN        NaN
3       0.0        0.0         1.0      415.0
4       NaN        NaN         NaN        NaN
...     ...        ...         ...        ...
6661    0.0        0.0         0.0      510.0
6662    NaN        NaN         NaN        NaN
6663    0.0        1.0         0.0      510.0
6664    NaN        NaN         NaN        NaN
6665    0.0        0.0         1.0      415.0

[6666 rows x 4 columns]
Index(['Churn', 'Intl Plan', 'VMail Plan', 'Area Code'], dtype='object')
```

Out[425]:

```
6666
```

In [426]:

```
for i in range(len(data1)):
    for j in range(4):
        a=a[0:].astype(object)
        print(a)
```

```
     Churn Intl Plan VMail Plan Area Code
0      NaN       NaN        NaN       NaN
1        0         0          1       415
2      NaN       NaN        NaN       NaN
3        0         0          1       415
4      NaN       NaN        NaN       NaN
...    ...       ...        ...       ...
6661     0         0          0       510
6662   NaN       NaN        NaN       NaN
6663     0         1          0       510
6664   NaN       NaN        NaN       NaN
6665     0         0          1       415

[6666 rows x 4 columns]
     Churn Intl Plan VMail Plan Area Code
0      NaN       NaN        NaN       NaN
1        0         0          1       415
2      NaN       NaN        NaN       NaN
3        0         0          1       415
```

In [427]:

```
len(data1)
```

Out[427]:

```
6666
```

In [428]:

```
a.iloc[0].dtype
```

Out[428]:

```
dtype('O')
```

In [10]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6666 entries, 0 to 6665
Data columns (total 4 columns):
Churn         3333 non-null object
Intl Plan     3333 non-null object
VMail Plan    3333 non-null object
Area Code     3333 non-null object
dtypes: object(4)
memory usage: 208.4+ KB
```

Null values Check and Imputation

In [169]:

```
data1.isnull().sum()
```

Out[169]:

```
Account Length      3333
VMail Message       3333
Day Mins            3333
Eve Mins            3333
Night Mins          3333
Intl Mins           3333
CustServ Calls      3333
Churn               3333
Intl Plan           3333
VMail Plan          3333
Day Calls           3333
Day Charge          3333
Daily Charges MV    3383
Eve Calls           3333
Eve Charge          3333
Night Calls         3333
Night Charge        3333
Intl Calls          3333
Intl Charge         3333
State               3333
Area Code           3333
Phone               3333
dtype: int64
```

In [170]:

```
data1.isnull().any()
```

Out[170]:

```
Account Length      True
VMail Message       True
Day Mins            True
Eve Mins            True
Night Mins          True
Intl Mins           True
CustServ Calls      True
Churn               True
Intl Plan           True
VMail Plan          True
Day Calls           True
Day Charge          True
Daily Charges MV    True
Eve Calls           True
Eve Charge          True
Night Calls         True
Night Charge        True
Intl Calls          True
Intl Charge         True
State               True
Area Code           True
Phone               True
dtype: bool
```

In [171]:

```
data1.duplicated().sum()
```

Out[171]:

```
3332
```

In [172]:

```python
data1=data1.dropna(how='all')
data1
```

Out[172]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Churn	Intl Plan	VMail Plan	...	Dail Charge M
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	0.0	0.0	1.0	...	45.0
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	0.0	0.0	1.0	...	27.4
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0.0	0.0	0.0	...	41.3
7	84.0	0.0	299.4	61.9	196.9	6.6	2.0	0.0	1.0	0.0	...	50.9
9	75.0	0.0	166.7	148.3	186.9	10.1	3.0	0.0	1.0	0.0	...	28.3
...	...	...	...	...	...	...	...	...	...	...	...	.
6657	192.0	36.0	156.2	215.5	279.1	9.9	2.0	0.0	0.0	1.0	...	26.5
6659	68.0	0.0	231.1	153.4	191.3	9.6	3.0	0.0	0.0	0.0	...	39.2
6661	28.0	0.0	180.8	288.8	191.9	14.1	2.0	0.0	0.0	0.0	...	30.7
6663	184.0	0.0	213.8	159.6	139.2	5.0	2.0	0.0	1.0	0.0	...	36.3
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0.0	0.0	1.0	...	39.8

3333 rows × 22 columns

In [173]:

```
data1.isna().sum()
```

Out[173]:

```
Account Length       0
VMail Message        0
Day Mins             0
Eve Mins             0
Night Mins           0
Intl Mins            0
CustServ Calls       0
Churn                0
Intl Plan            0
VMail Plan           0
Day Calls            0
Day Charge           0
Daily Charges MV    50
Eve Calls            0
Eve Charge           0
Night Calls          0
Night Charge         0
Intl Calls           0
Intl Charge          0
State                0
Area Code            0
Phone                0
dtype: int64
```

In [174]:

```
data1['Daily Charges MV'].fillna(data1['Daily Charges MV'].median(),inplace=True)
```

In [175]:

```
data1['Daily Charges MV'].isnull().sum()
```

Out[175]:

```
0
```

In [176]:

```
data1.isna().sum()
```

Out[176]:

```
Account Length      0
VMail Message       0
Day Mins            0
Eve Mins            0
Night Mins          0
Intl Mins           0
CustServ Calls      0
Churn               0
Intl Plan           0
VMail Plan          0
Day Calls           0
Day Charge          0
Daily Charges MV    0
Eve Calls           0
Eve Charge          0
Night Calls         0
Night Charge        0
Intl Calls          0
Intl Charge         0
State               0
Area Code           0
Phone               0
dtype: int64
```

Duplicate check

In [21]:

```
data1.duplicated().sum()
```

Out[21]:

```
0
```

To Find the Outliers in a Data Frame

In [177]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 22 columns):
Account Length      3333 non-null float64
VMail Message       3333 non-null float64
Day Mins            3333 non-null float64
Eve Mins            3333 non-null float64
Night Mins          3333 non-null float64
Intl Mins           3333 non-null float64
CustServ Calls      3333 non-null float64
Churn               3333 non-null float64
Intl Plan           3333 non-null float64
VMail Plan          3333 non-null float64
Day Calls           3333 non-null float64
Day Charge          3333 non-null float64
Daily Charges MV    3333 non-null float64
Eve Calls           3333 non-null float64
Eve Charge          3333 non-null float64
Night Calls         3333 non-null float64
Night Charge        3333 non-null float64
Intl Calls          3333 non-null float64
Intl Charge         3333 non-null float64
State               3333 non-null object
Area Code           3333 non-null float64
Phone               3333 non-null object
dtypes: float64(20), object(2)
memory usage: 598.9+ KB
```

In [203]:

```
print(len(data1))
a=data1[['Churn','Intl Plan','VMail Plan','Area Code']]
print(a)
print(a.columns)
len(a)
```

```
3333
      Churn  Intl Plan  VMail Plan  Area Code
1       0.0        0.0         1.0      415.0
3       0.0        0.0         1.0      415.0
5       0.0        0.0         0.0      415.0
7       0.0        1.0         0.0      408.0
9       0.0        1.0         0.0      415.0
...     ...        ...         ...        ...
6657    0.0        0.0         1.0      415.0
6659    0.0        0.0         0.0      415.0
6661    0.0        0.0         0.0      510.0
6663    0.0        1.0         0.0      510.0
6665    0.0        0.0         1.0      415.0

[3333 rows x 4 columns]
Index(['Churn', 'Intl Plan', 'VMail Plan', 'Area Code'], dtype='object')
```

Out[203]:

```
3333
```

In [204]:

```
for i in range(len(data1)):
    for j in range(4):
        a=a[0:].astype(object)
        print(a)
```

```
     Churn Intl Plan VMail Plan Area Code
1        0         0          1       415
3        0         0          1       415
5        0         0          0       415
7        0         1          0       408
9        0         1          0       415
...    ...       ...        ...       ...
6657     0         0          1       415
6659     0         0          0       415
6661     0         0          0       510
6663     0         1          0       510
6665     0         0          1       415

[3333 rows x 4 columns]
     Churn Intl Plan VMail Plan Area Code
1        0         0          1       415
3        0         0          1       415
5        0         0          0       415
7        0         1          0       408
```

In [205]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 4 columns):
Churn         3333 non-null object
Intl Plan     3333 non-null object
VMail Plan    3333 non-null object
Area Code     3333 non-null object
dtypes: object(4)
memory usage: 130.2+ KB
```

In [206]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 22 columns):
Account Length      3333 non-null float64
VMail Message       3333 non-null float64
Day Mins            3333 non-null float64
Eve Mins            3333 non-null float64
Night Mins          3333 non-null float64
Intl Mins           3333 non-null float64
CustServ Calls      3333 non-null float64
Churn               3333 non-null float64
Intl Plan           3333 non-null float64
VMail Plan          3333 non-null float64
Day Calls           3333 non-null float64
Day Charge          3333 non-null float64
Daily Charges MV    3333 non-null float64
Eve Calls           3333 non-null float64
Eve Charge          3333 non-null float64
Night Calls         3333 non-null float64
Night Charge        3333 non-null float64
Intl Calls          3333 non-null float64
Intl Charge         3333 non-null float64
State               3333 non-null object
Area Code           3333 non-null float64
Phone               3333 non-null object
dtypes: float64(20), object(2)
memory usage: 598.9+ KB
```

In [207]:

```
discretecolumns=a[['Churn','Intl Plan','VMail Plan','Area Code']]
discretecolumns1=data1[['State','Phone']]
continuouscolumns=data1[['Account Length','VMail Message','Day Mins','Eve Mins', 'Night Mir
```

In [208]:

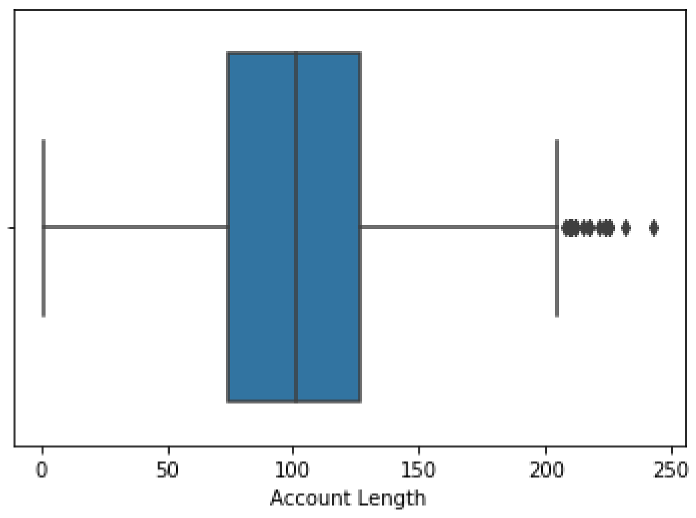
```

data1.nunique()
import seaborn as sns
import matplotlib.pyplot as plt
```

Univariate Analysis

In [26]:

```
for i in continuouscolumns:
    print("continuouscolumns:",i)
    sns.boxplot(continuouscolumns[i])
    plt.show()
```

```
continuouscolumns: Account Length
```

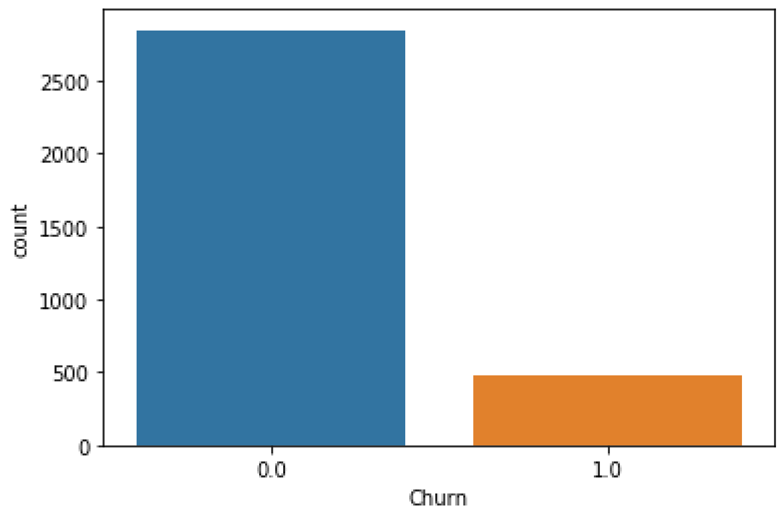


```
continuouscolumns: VMail Message
```

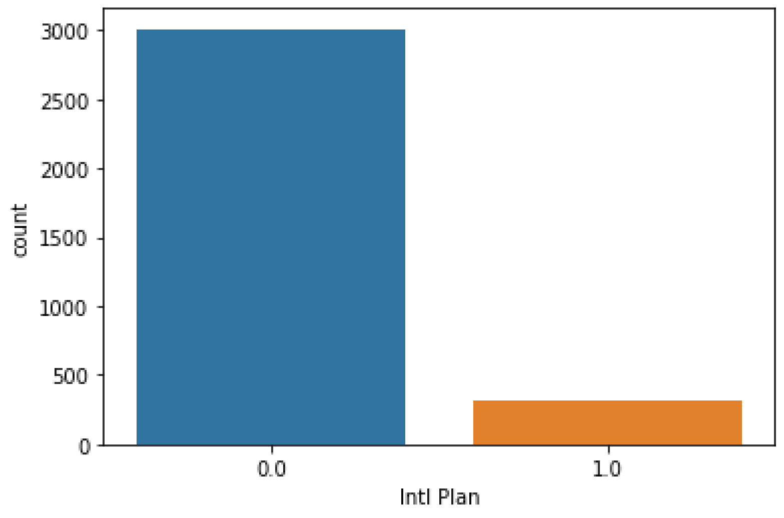
In [27]:

```
#  for discrete columns
for i in discretecolumns :
    print("discretecolumns:",i)
    sns.countplot(discretecolumns[i])
    plt.show()
```

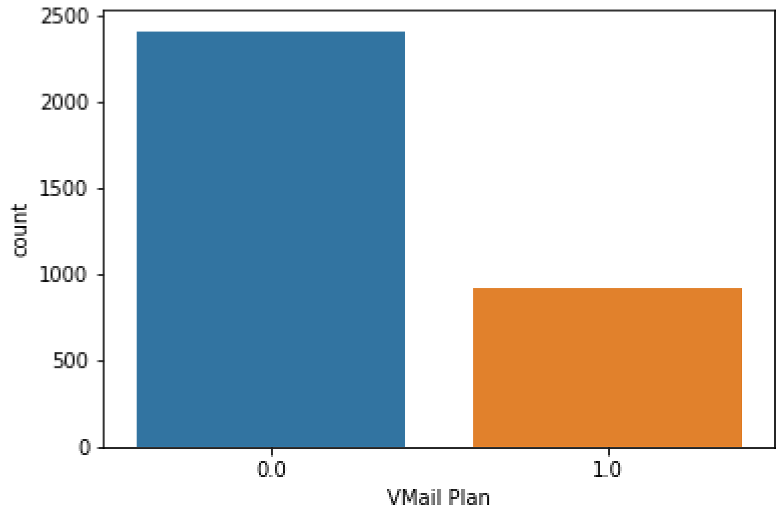
```
discretecolumns: Churn
```



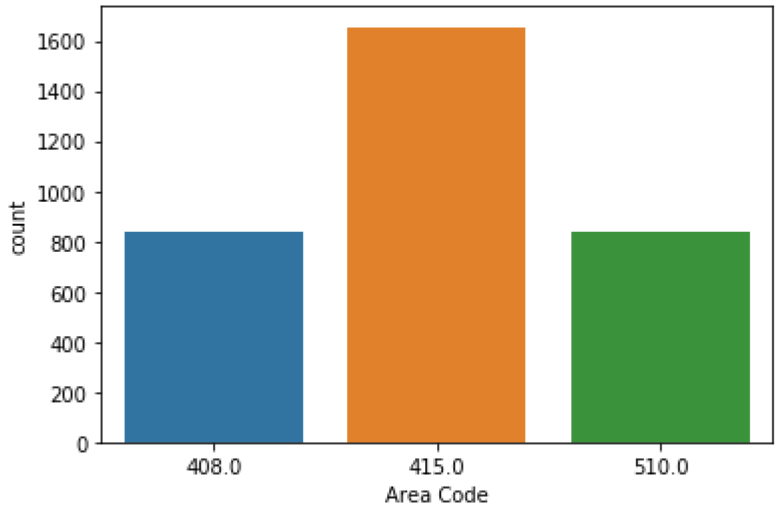
```
discretecolumns: Intl Plan
```



```
discretecolumns: VMail Plan
```



```
discretecolumns: Area Code
```



In [28]:

```
a=len(continuouscolumns)
for i in range(a):
    for j in range(17):
        sns.boxplot(continuouscolumns)
```

```
---------------------------------------------------------------------------
-
KeyboardInterrupt                         Traceback (most recent call las
t)
<ipython-input-28-e7f1f57d43ee> in <module>
      2 for i in range(a):
      3     for j in range(17):
----> 4         sns.boxplot(continuouscolumns)

~\Anaconda3\lib\site-packages\seaborn\categorical.py in boxplot(x, y, hue,
data, order, hue_order, orient, color, palette, saturation, width, dodge,
 fliersize, linewidth, whis, notch, ax, **kwargs)
   2235     kwargs.update(dict(whis=whis, notch=notch))
   2236
-> 2237     plotter.plot(ax, kwargs)
   2238     return ax
   2239

~\Anaconda3\lib\site-packages\seaborn\categorical.py in plot(self, ax, box
```

In [ ]:

```
print(data1.head(10))
```

#'Account Length',

#'VMail Message',

#'Day Mins',

#'Eve Mins',

#'Night Mins',

#'Intl Mins',

#'CustServ Calls',

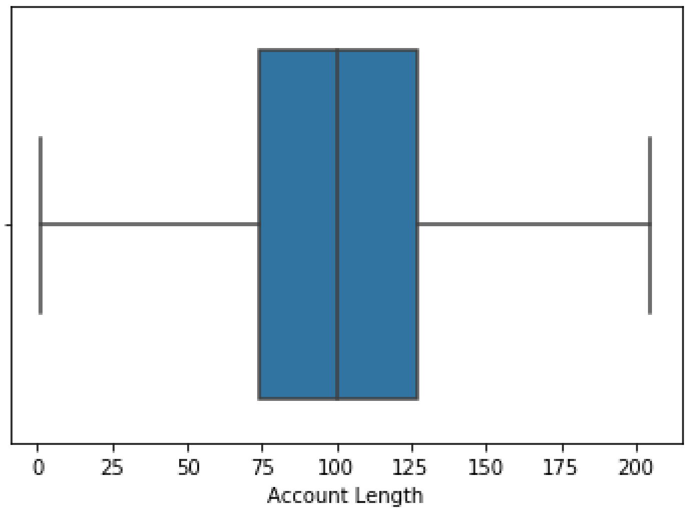
#'Day Calls', #'Day Charge', 'Daily Charges MV', 'Eve Calls', 'Eve Charge', 'Night Calls', 'Night Charge', 'Intl Calls', 'Intl Charge'

In [226]:

```
def fun(x,q=0.75):
    Q1=x.quantile(1-q)
    Q3=x.quantile(q)
    IQR=Q3-Q1
    return x
```

In [227]:

```
for i in continuouscolumns:
    Q1=continuouscolumns[i].quantile(0.25)
    Q3=continuouscolumns[i].quantile(0.75)
    IQR=Q3-Q1
    x=continuouscolumns[(continuouscolumns[i]>=(Q1-1.5*IQR))&(continuouscolumns[i]<=(Q3+1.5
#k=data.loc[filter]
#k
    sns.boxplot(x[i])
    plt.show()
```



Bi variate Analysis

In [28]:

```
data1.corr()
```

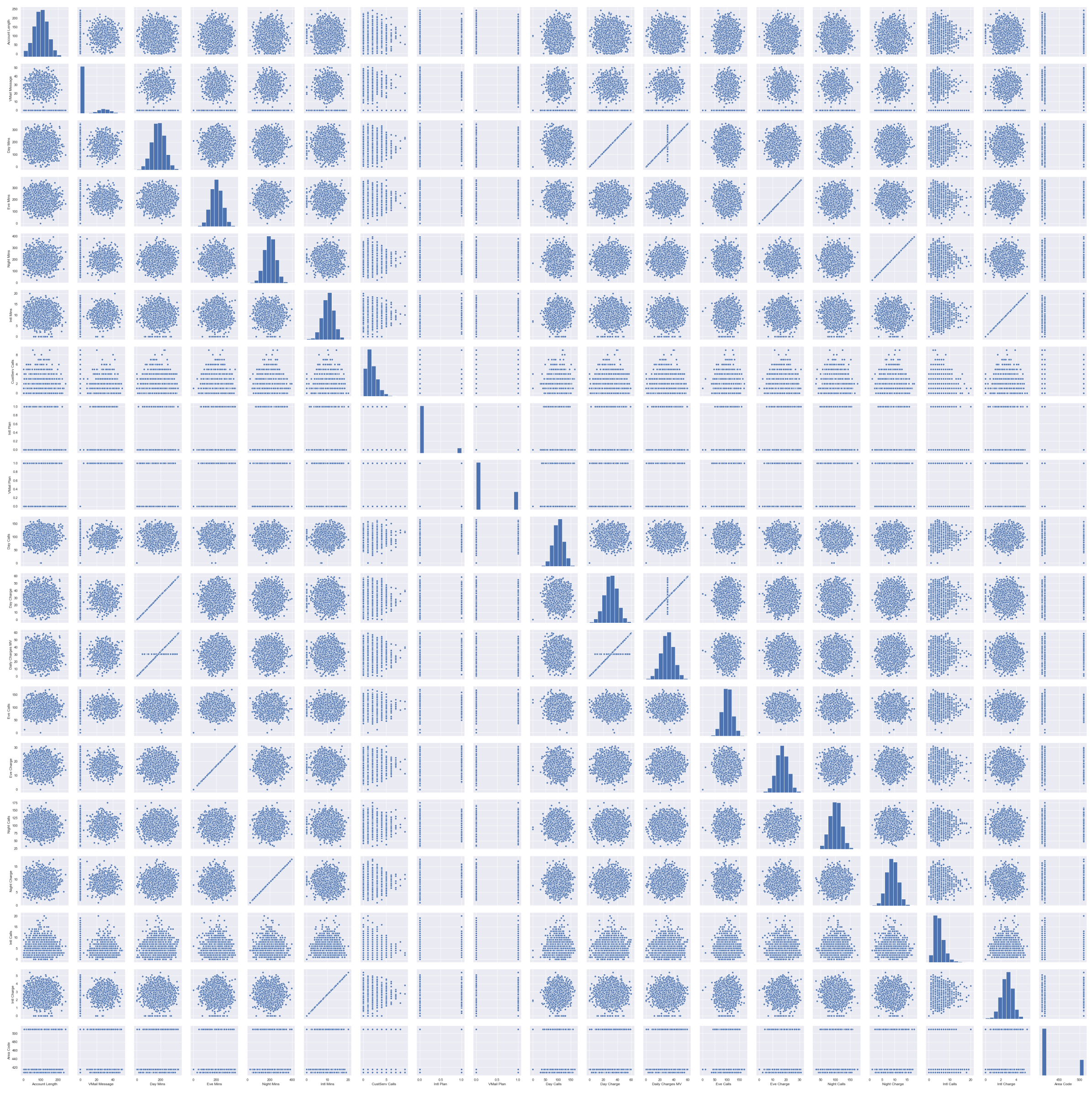
Out[28]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Churn
Account Length	1.000000	-0.004628	0.006216	-0.006757	-0.008955	0.009514	-0.003796	0.016541
VMail Message	-0.004628	1.000000	0.000778	0.017562	0.007681	0.002856	-0.013263	-0.089728
Day Mins	0.006216	0.000778	1.000000	0.007043	0.004323	-0.010155	-0.013423	0.205151
Eve Mins	-0.006757	0.017562	0.007043	1.000000	-0.012584	-0.011035	-0.012985	0.092796
Night Mins	-0.008955	0.007681	0.004323	-0.012584	1.000000	-0.015207	-0.009288	0.035493
Intl Mins	0.009514	0.002856	-0.010155	-0.011035	-0.015207	1.000000	-0.009640	0.068239
CustServ Calls	-0.003796	-0.013263	-0.013423	-0.012985	-0.009288	-0.009640	1.000000	0.208750
Churn	0.016541	-0.089728	0.205151	0.092796	0.035493	0.068239	0.208750	1.000000
Intl Plan	0.024735	0.008745	0.049396	0.019100	-0.028905	0.045871	-0.024522	0.259852
VMail Plan	0.002918	0.956927	-0.001684	0.021545	0.006079	-0.001318	-0.017824	-0.102148
Day Calls	0.038470	-0.009548	0.006750	-0.021451	0.022938	0.021565	-0.018942	0.018459
Day Charge	0.006214	0.000776	1.000000	0.007050	0.004324	-0.010157	-0.013427	0.205151
Daily Charges MV	0.008347	0.001987	0.986681	0.006556	0.004402	-0.013933	-0.011189	0.201291
Eve Calls	0.019260	-0.005864	0.015769	-0.011430	-0.002093	0.008703	0.002423	0.009233
Eve Charge	-0.006745	0.017578	0.007029	1.000000	-0.012592	-0.011043	-0.012987	0.092786
Night Calls	-0.013176	0.007123	0.022972	0.007586	0.011204	-0.013605	-0.012802	0.006141
Night Charge	-0.008960	0.007663	0.004300	-0.012593	0.999999	-0.015214	-0.009277	0.035496
Intl Calls	0.020661	0.013957	0.008033	0.002541	-0.012353	0.032304	-0.017561	-0.052844
Intl Charge	0.009546	0.002884	-0.010092	-0.011067	-0.015180	0.999993	-0.009675	0.068259
Area Code	-0.012463	-0.001994	-0.008264	0.003580	-0.005825	-0.018288	0.027572	0.006174

In [29]:

```
sns.set()
cols=['Account Length', 'VMail Message', 'Day Mins', 'Eve Mins', 'Night Mins','Intl Mins',
        'Day Calls', 'Day Charge', 'Daily Charges MV', 'Eve Calls','Eve Charge', 'Night Call
sns.pairplot(data1[cols],size = 2.6)
plt.show()
```

```
C:\Users\admin\Anaconda3\lib\site-packages\seaborn\axisgrid.py:2065: UserWar
ning: The `size` parameter has been renamed to `height`; pleaes update your
code.
  warnings.warn(msg, UserWarning)
```



In [420]:

```
sns.set()
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(x="Area Code",hue="State",data=discretecolumns)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-420-c01b2acd2aba> in <module>
      2 import seaborn as sns
      3 import matplotlib.pyplot as plt
----> 4 sns.countplot(x="Area Code",hue="State",data=discretecolumns)

NameError: name 'discretecolumns' is not defined
```

Linear Regression (Model 1) Target Variable is cust sev calls

In [55]:

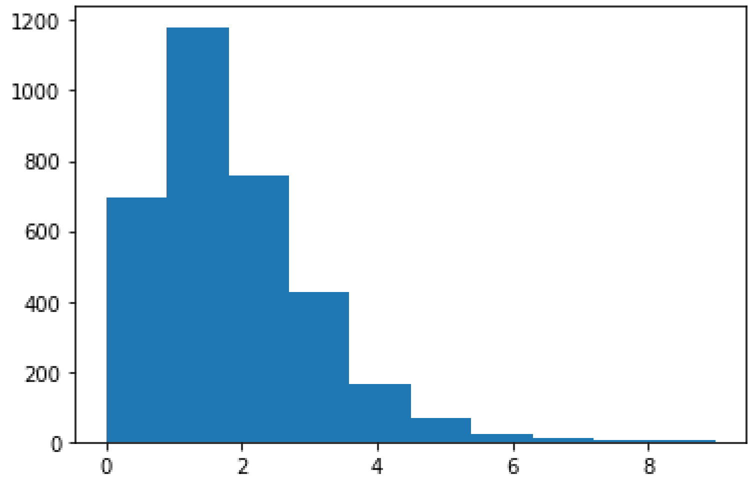
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import scipy as stats
```

In [59]:

```
#1.) Target variable should follow normal distribution
# can seen by using matlabplot and seaborn
plt.hist(y)
```

Out[59]:

```
(array([ 697., 1181.,  759.,  429.,  166.,   66.,   22.,    9.,    2.,
           2.]),
 array([0. , 0.9, 1.8, 2.7, 3.6, 4.5, 5.4, 6.3, 7.2, 8.1, 9. ]),
 <a list of 10 Patch objects>)
```

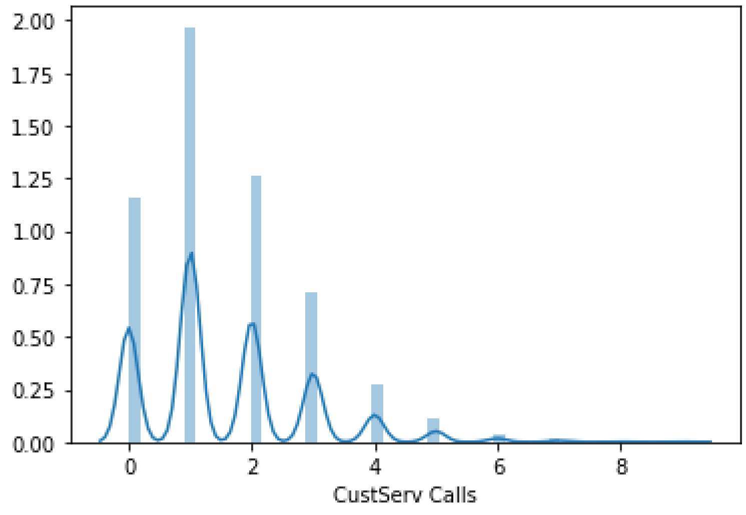


In [60]:

```
sns.distplot(y)
```

Out[60]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x299a41f4948>
```



In [120]:

```
# 2. input variables should be independent to each other ,that can be check by using correl
d=x.corr()
d
```

Out[120]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	Churn	Intl Plan
Account Length	1.000000	-0.004628	0.006216	-0.006757	-0.008955	0.009514	0.016541	0.024735
VMail Message	-0.004628	1.000000	0.000778	0.017562	0.007681	0.002856	-0.089728	0.008745
Day Mins	0.006216	0.000778	1.000000	0.007043	0.004323	-0.010155	0.205151	0.049396
Eve Mins	-0.006757	0.017562	0.007043	1.000000	-0.012584	-0.011035	0.092796	0.019100
Night Mins	-0.008955	0.007681	0.004323	-0.012584	1.000000	-0.015207	0.035493	-0.028905
Intl Mins	0.009514	0.002856	-0.010155	-0.011035	-0.015207	1.000000	0.068239	0.045871
Churn	0.016541	-0.089728	0.205151	0.092796	0.035493	0.068239	1.000000	0.259852
Intl Plan	0.024735	0.008745	0.049396	0.019100	-0.028905	0.045871	0.259852	1.000000
VMail Plan	0.002918	0.956927	-0.001684	0.021545	0.006079	-0.001318	-0.102148	0.006006
Day Calls	0.038470	-0.009548	0.006750	-0.021451	0.022938	0.021565	0.018459	0.003755
Day Charge	0.006214	0.000776	1.000000	0.007050	0.004324	-0.010157	0.205151	0.049398
Daily Charges MV	0.008347	0.001987	0.986681	0.006556	0.004402	-0.013933	0.201291	0.047733
Eve Calls	0.019260	-0.005864	0.015769	-0.011430	-0.002093	0.008703	0.009233	0.006114
Eve Charge	-0.006745	0.017578	0.007029	1.000000	-0.012592	-0.011043	0.092786	0.019106
Night Calls	-0.013176	0.007123	0.022972	0.007586	0.011204	-0.013605	0.006141	0.012451
Night Charge	-0.008960	0.007663	0.004300	-0.012593	0.999999	-0.015214	0.035496	-0.028913
Intl Calls	0.020661	0.013957	0.008033	0.002541	-0.012353	0.032304	-0.052844	0.017366
Intl Charge	0.009546	0.002884	-0.010092	-0.011067	-0.015180	0.999993	0.068259	0.045780
Area Code	-0.012463	-0.001994	-0.008264	0.003580	-0.005825	-0.018288	0.006174	0.048551

In [121]:

```
corr = x.corr()
rows,cols=x.shape
for i in list(corr.columns):
    for j in list(corr.columns):
        if corr.ix[i,j]>0.7 and corr.ix[i,j] != 1:
            print( i, ' ',j ,' ', corr.ix[i,j])
```

```
VMail Message   VMail Plan   0.9569266420696362
Day Mins   Day Charge   0.999999952190397
Day Mins   Daily Charges MV   0.9866811759470281
Eve Mins   Eve Charge   0.9999997760198517
Night Mins   Night Charge   0.99999921487588
Intl Mins   Intl Charge   0.9999927417510258
VMail Plan   VMail Message   0.9569266420696362
Day Charge   Day Mins   0.999999952190397
Day Charge   Daily Charges MV   0.9866804448512339
Daily Charges MV   Day Mins   0.9866811759470281
Daily Charges MV   Day Charge   0.9866804448512339
Eve Charge   Eve Mins   0.9999997760198517
Night Charge   Night Mins   0.99999921487588
Intl Charge   Intl Mins   0.9999927417510258

C:\Users\admin\Anaconda3\lib\site-packages\ipykernel_launcher.py:5: FutureWa
rning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ix-inde
xer-is-deprecated (http://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#ix-indexer-is-deprecated)
  """
C:\Users\admin\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: FutureWa
rning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ix-inde
xer-is-deprecated (http://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#ix-indexer-is-deprecated)
```

In [122]:

```
rows,cols=x.shape
cols
```

Out[122]:

```
21
```

In [25]:
```
x=x.drop(["Daily Charges MV","State", "Area Code", "Phone"],axis=1)
```

In [26]:
```
x
```

Out[26]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	Churn	Intl Plan	VMail Plan	Day Calls	Day Charge	Eve Calls
1	128.0	25.0	265.1	197.4	244.7	10.0	0.0	0.0	1.0	110.0	45.07	99.0
3	107.0	26.0	161.6	195.5	254.4	13.7	0.0	0.0	1.0	123.0	27.47	103.0
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0.0	0.0	114.0	41.38	110.0
7	84.0	0.0	299.4	61.9	196.9	6.6	0.0	1.0	0.0	71.0	50.90	88.0
9	75.0	0.0	166.7	148.3	186.9	10.1	0.0	1.0	0.0	113.0	28.34	122.0
...	...	...	...	...	...	...	...	...	...	...	...	...
6657	192.0	36.0	156.2	215.5	279.1	9.9	0.0	0.0	1.0	77.0	26.55	126.0
6659	68.0	0.0	231.1	153.4	191.3	9.6	0.0	0.0	0.0	57.0	39.29	55.0
6661	28.0	0.0	180.8	288.8	191.9	14.1	0.0	0.0	0.0	109.0	30.74	58.0
6663	184.0	0.0	213.8	159.6	139.2	5.0	0.0	1.0	0.0	105.0	36.35	84.0
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0.0	1.0	113.0	39.85	82.0

3333 rows × 17 columns

For categorical variables ,perform One Hot Encoding

In [321]:

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 21 columns):
Account Length      3333 non-null float64
VMail Message       3333 non-null float64
Day Mins            3333 non-null float64
Eve Mins            3333 non-null float64
Night Mins          3333 non-null float64
Intl Mins           3333 non-null float64
CustServ Calls      3333 non-null float64
Intl Plan           3333 non-null float64
VMail Plan          3333 non-null float64
Day Calls           3333 non-null float64
Day Charge          3333 non-null float64
Daily Charges MV    3333 non-null float64
Eve Calls           3333 non-null float64
Eve Charge          3333 non-null float64
Night Calls         3333 non-null float64
Night Charge        3333 non-null float64
Intl Calls          3333 non-null float64
Intl Charge         3333 non-null float64
State               3333 non-null object
Area Code           3333 non-null float64
Phone               3333 non-null object
dtypes: float64(19), object(2)
memory usage: 572.9+ KB
```

In [27]:

```
z=x[['Churn','Intl Plan','VMail Plan']]
z
for i in range(len(z)):
    for j in range(3):
        z=z[0:].astype(object)
        print(z)
```

```
     Churn Intl Plan VMail Plan
1        0         0          1
3        0         0          1
5        0         0          0
7        0         1          0
9        0         1          0
...    ...       ...        ...
6657     0         0          1
6659     0         0          0
6661     0         0          0
6663     0         1          0
6665     0         0          1

[3333 rows x 3 columns]
     Churn Intl Plan VMail Plan
1        0         0          1
3        0         0          1
5        0         0          0
7        0         1          0
```

In [28]:

```
z['Churn'].dtype
```

Out[28]:

```
dtype('O')
```

In [29]:

```
x.drop(['Churn','Intl Plan','VMail Plan'],axis=1)
x
```

Out[29]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	Churn	Intl Plan	VMail Plan	Day Calls	Day Charge	Eve Calls
1	128.0	25.0	265.1	197.4	244.7	10.0	0.0	0.0	1.0	110.0	45.07	99.0
3	107.0	26.0	161.6	195.5	254.4	13.7	0.0	0.0	1.0	123.0	27.47	103.0
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0.0	0.0	114.0	41.38	110.0
7	84.0	0.0	299.4	61.9	196.9	6.6	0.0	1.0	0.0	71.0	50.90	88.0
9	75.0	0.0	166.7	148.3	186.9	10.1	0.0	1.0	0.0	113.0	28.34	122.0
...	...	...	...	...	...	...	...	...	...	...	...	...
6657	192.0	36.0	156.2	215.5	279.1	9.9	0.0	0.0	1.0	77.0	26.55	126.0
6659	68.0	0.0	231.1	153.4	191.3	9.6	0.0	0.0	0.0	57.0	39.29	55.0
6661	28.0	0.0	180.8	288.8	191.9	14.1	0.0	0.0	0.0	109.0	30.74	58.0
6663	184.0	0.0	213.8	159.6	139.2	5.0	0.0	1.0	0.0	105.0	36.35	84.0
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0.0	1.0	113.0	39.85	82.0

3333 rows × 17 columns

In [30]:

```
data2 = pd.concat([x, z], axis=1, join='inner')
data2
```

Out[30]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	Churn	Intl Plan	VMail Plan	Day Calls	Day Charge	Eve Calls
1	128.0	25.0	265.1	197.4	244.7	10.0	0.0	0.0	1.0	110.0	45.07	99.0
3	107.0	26.0	161.6	195.5	254.4	13.7	0.0	0.0	1.0	123.0	27.47	103.0
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0.0	0.0	114.0	41.38	110.0
7	84.0	0.0	299.4	61.9	196.9	6.6	0.0	1.0	0.0	71.0	50.90	88.0
9	75.0	0.0	166.7	148.3	186.9	10.1	0.0	1.0	0.0	113.0	28.34	122.0
...	...	...	...	...	...	...	...	...	...	...	...	...
6657	192.0	36.0	156.2	215.5	279.1	9.9	0.0	0.0	1.0	77.0	26.55	126.0
6659	68.0	0.0	231.1	153.4	191.3	9.6	0.0	0.0	0.0	57.0	39.29	55.0
6661	28.0	0.0	180.8	288.8	191.9	14.1	0.0	0.0	0.0	109.0	30.74	58.0
6663	184.0	0.0	213.8	159.6	139.2	5.0	0.0	1.0	0.0	105.0	36.35	84.0
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0.0	1.0	113.0	39.85	82.0

3333 rows × 20 columns

In [31]:

```
data2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 20 columns):
Account Length    3333 non-null float64
VMail Message     3333 non-null float64
Day Mins          3333 non-null float64
Eve Mins          3333 non-null float64
Night Mins        3333 non-null float64
Intl Mins         3333 non-null float64
Churn             3333 non-null float64
Intl Plan         3333 non-null float64
VMail Plan        3333 non-null float64
Day Calls         3333 non-null float64
Day Charge        3333 non-null float64
Eve Calls         3333 non-null float64
Eve Charge        3333 non-null float64
Night Calls       3333 non-null float64
Night Charge      3333 non-null float64
Intl Calls        3333 non-null float64
Intl Charge       3333 non-null float64
Churn             3333 non-null object
Intl Plan         3333 non-null object
VMail Plan        3333 non-null object
dtypes: float64(17), object(3)
memory usage: 546.8+ KB
```

In [32]:

```
x=pd.get_dummies(data2)
x
```

Out[32]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	Churn	Intl Plan	VMail Plan	Day Calls	...	Night Calls	Ni Cha
1	128.0	25.0	265.1	197.4	244.7	10.0	0.0	0.0	1.0	110.0	...	91.0	1
3	107.0	26.0	161.6	195.5	254.4	13.7	0.0	0.0	1.0	123.0	...	103.0	1
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0.0	0.0	114.0	...	104.0	7
7	84.0	0.0	299.4	61.9	196.9	6.6	0.0	1.0	0.0	71.0	...	89.0	8
9	75.0	0.0	166.7	148.3	186.9	10.1	0.0	1.0	0.0	113.0	...	121.0	8
...	...	...	...	...	...	...	...	...	...	...	...	...	
6657	192.0	36.0	156.2	215.5	279.1	9.9	0.0	0.0	1.0	77.0	...	83.0	12
6659	68.0	0.0	231.1	153.4	191.3	9.6	0.0	0.0	0.0	57.0	...	123.0	8
6661	28.0	0.0	180.8	288.8	191.9	14.1	0.0	0.0	0.0	109.0	...	91.0	8
6663	184.0	0.0	213.8	159.6	139.2	5.0	0.0	1.0	0.0	105.0	...	137.0	6
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0.0	1.0	113.0	...	77.0	10

3333 rows × 23 columns

In [33]:

```
x.shape
```

Out[33]:

```
(3333, 23)
```

Build the model

In [159]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
```

In [160]:

```
x_train
```

Out[160]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	Churn	Intl Plan	VMail Plan	Day Calls	...	Night Calls	Ni Cha
3285	99.0	0.0	54.8	173.0	195.1	7.5	0.0	0.0	0.0	92.0	...	125.0	8
3621	124.0	0.0	194.0	241.0	227.5	11.9	0.0	0.0	0.0	103.0	...	153.0	10
6123	90.0	0.0	222.0	187.0	282.3	12.4	0.0	0.0	0.0	93.0	...	124.0	12
4857	40.0	0.0	81.7	210.2	212.0	11.3	1.0	0.0	0.0	123.0	...	64.0	9
2531	95.0	39.0	260.8	213.4	195.6	10.1	0.0	0.0	1.0	130.0	...	97.0	8
...	...	...	...	...	...	...	...	...	...	...	...	...	
5527	116.0	19.0	155.7	185.4	192.7	8.2	0.0	0.0	1.0	104.0	...	116.0	8
1811	161.0	0.0	191.9	70.9	204.8	13.4	1.0	0.0	0.0	113.0	...	107.0	9
2193	93.0	0.0	98.4	249.6	248.2	14.2	0.0	0.0	0.0	78.0	...	114.0	11
471	139.0	0.0	134.4	211.3	193.6	10.2	1.0	0.0	0.0	106.0	...	125.0	8
2123	132.0	31.0	174.5	245.6	172.8	10.3	0.0	0.0	1.0	101.0	...	76.0	7

2333 rows × 23 columns

In [161]:

```
x_test
```

Out[161]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	Churn	Intl Plan	VMail Plan	Day Calls	...	Night Calls	Ni Cha
4721	68.0	0.0	222.1	199.4	162.4	9.4	0.0	0.0	0.0	107.0	...	107.0	7
1201	102.0	0.0	102.6	246.0	170.5	9.1	0.0	0.0	0.0	89.0	...	140.0	7
3003	72.0	0.0	272.4	107.9	185.5	12.7	0.0	0.0	0.0	88.0	...	81.0	8
2229	108.0	15.0	165.1	267.0	250.7	10.9	0.0	0.0	1.0	85.0	...	114.0	11
1035	52.0	0.0	214.7	158.6	123.4	9.4	0.0	0.0	0.0	68.0	...	114.0	5
...	...	...	...	...	...	...	...	...	...	...	...	...	
6449	115.0	0.0	226.4	276.8	213.4	12.3	1.0	0.0	0.0	101.0	...	82.0	9
5237	116.0	27.0	175.5	210.6	294.8	6.9	0.0	1.0	1.0	137.0	...	121.0	13
4055	87.0	36.0	171.2	185.8	227.6	10.8	0.0	1.0	1.0	138.0	...	97.0	10
4813	81.0	0.0	145.6	287.9	181.7	9.2	0.0	0.0	0.0	59.0	...	121.0	8
5083	73.0	0.0	94.9	253.2	175.1	14.2	0.0	0.0	0.0	121.0	...	86.0	7

1000 rows × 23 columns

In [162]:

```
y_train
```

Out[162]:

```
3285    1.0
3621    0.0
6123    2.0
4857    6.0
2531    1.0
       ...
5527    3.0
1811    4.0
2193    1.0
471     5.0
2123    1.0
Name: CustServ Calls, Length: 2333, dtype: float64
```

In [163]:

```
y_test
```

Out[163]:

```
4721    2.0
1201    2.0
3003    0.0
2229    1.0
1035    2.0
       ...
6449    3.0
5237    1.0
4055    1.0
4813    2.0
5083    2.0
Name: CustServ Calls, Length: 1000, dtype: float64
```

In [164]:

```
lm=LinearRegression()
lm.fit(x_train,y_train)
```

Out[164]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fal
se)
```

In [165]:

```
lm.score(x_train,y_train)
```

Out[165]:

```
0.07609350866404363
```

In [166]:

```
lm.coef_
```

Out[166]:

```
array([-6.78129472e-05,  3.46762203e-03,  1.26260217e+00, -1.66224817e-01,
        5.64396120e-02,  2.40090540e+00,  3.59539896e-01, -1.55866671e-01,
       -3.17390035e-02, -9.53841522e-04, -7.43900699e+00,  3.97914964e-05,
        1.94498407e+00, -1.04222281e-03, -1.26248657e+00,  2.77132938e-03,
       -8.95663877e+00, -3.59539896e-01,  3.59539896e-01,  1.55866671e-01,
       -1.55866671e-01,  3.17390035e-02, -3.17390035e-02])
```

In [168]:

```
y_pred=lm.predict(x_test)
y_pred
```

Out[168]:

```
array([1.34637369, 1.56326901, 1.31478931, 1.36932427, 1.46516883,
       0.94099901, 1.40714536, 1.43904278, 2.3741758 , 1.48659312,
       1.35883334, 1.41135798, 1.36160543, 1.35527218, 1.35515213,
       1.34145823, 0.92646698, 1.57524264, 1.47027494, 1.29126907,
       1.43669017, 1.499066  , 1.49575403, 1.33471095, 1.46384978,
       1.83667034, 1.31658282, 2.558749  , 1.41222092, 1.23869408,
       1.42977401, 1.63044044, 1.56933736, 1.39225079, 1.43074253,
       1.39125438, 1.49275999, 1.65194021, 2.29810218, 1.34059191,
       1.46500622, 1.37844353, 1.53696945, 2.28816484, 1.7869904 ,
       2.15388728, 1.49096355, 1.22831156, 1.47903111, 2.08471829,
       2.66368503, 1.79327275, 1.35822463, 1.35455694, 1.15216705,
       1.59104613, 1.55798114, 1.4684836 , 1.37577355, 2.53088496,
       1.32640852, 1.43413368, 1.29440525, 2.29360294, 1.5372661 ,
       1.16845093, 1.64476987, 1.29908629, 2.57118977, 1.52996406,
       1.55357812, 1.26468673, 1.32540287, 1.50333805, 0.82459958,
       1.43937967, 1.44948103, 0.99143931, 1.20229671, 1.30007893,
       1.55280876, 1.61280734, 1.47969397, 1.6793945 , 1.83414716,
       1.69842216, 1.53440679, 1.36129759, 1.53059217, 2.076871  ,
```

In [ ]:

```

```

In [169]:

```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('Mean Absolute Percentage Error(MAPE):', np.mean(((abs(y_test - y_pred))/y_test)*100)
```

```
Mean Absolute Error: 1.031947009546339
Mean Squared Error: 1.6844028850849035
Root Mean Squared Error: 1.2978454781232254
Mean Absolute Percentage Error(MPAE): inf
```

Logistic Regression

In [430]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as stats
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

In [431]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6666 entries, 0 to 6665
Data columns (total 22 columns):
Account Length      3333 non-null float64
VMail Message       3333 non-null float64
Day Mins            3333 non-null float64
Eve Mins            3333 non-null float64
Night Mins          3333 non-null float64
Intl Mins           3333 non-null float64
CustServ Calls      3333 non-null float64
Churn               3333 non-null float64
Intl Plan           3333 non-null float64
VMail Plan          3333 non-null float64
Day Calls           3333 non-null float64
Day Charge          3333 non-null float64
Daily Charges MV    3283 non-null float64
Eve Calls           3333 non-null float64
Eve Charge          3333 non-null float64
Night Calls         3333 non-null float64
```

In [432]:

```
data1.head()
```

Out[432]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Churn	Intl Plan	VMail Plan	...	Daily Charges MV
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	0.0	0.0	1.0	...	45.07
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	0.0	0.0	1.0	...	27.47
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN

5 rows × 22 columns

In [433]:

```
data1.nunique()
```

Out[433]:

```
Account Length       212
VMail Message         46
Day Mins            1667
Eve Mins            1611
Night Mins          1591
Intl Mins            162
CustServ Calls        10
Churn                  2
Intl Plan              2
VMail Plan             2
Day Calls            119
Day Charge          1667
Daily Charges MV    1649
Eve Calls            123
Eve Charge          1440
Night Calls          120
Night Charge         933
Intl Calls            21
Intl Charge          162
State                 51
Area Code              3
Phone               3333
dtype: int64
```

In [434]:

```
data1["Churn"]=data1["Churn"].astype("object")

#"Intl Plan","VMail Plan","Area Code"]].astype("object")
```

In [435]:

```
data1["Intl Plan"]=data1["Intl Plan"].astype("object")
```

In [436]:

```
data1["VMail Plan"]=data1["VMail Plan"].astype("object")
```

In [437]:

```
data1["Area Code"]=data1["Area Code"].astype("object")
print(data1.isnull().sum())
```

```
Account Length      3333
VMail Message       3333
Day Mins            3333
Eve Mins            3333
Night Mins          3333
Intl Mins           3333
CustServ Calls      3333
Churn               3333
Intl Plan           3333
VMail Plan          3333
Day Calls           3333
Day Charge          3333
Daily Charges MV    3383
Eve Calls           3333
Eve Charge          3333
Night Calls         3333
Night Charge        3333
Intl Calls          3333
Intl Charge         3333
State               3333
Area Code           3333
Phone               3333
dtype: int64
```

In [438]:

```
data1=data1.dropna(how='all')
data1
```

Out[438]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Churn	Intl Plan	VMail Plan	...	Dail Charge M
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	0	0	1	...	45.0
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	0	0	1	...	27.4
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0	0	0	...	41.3
7	84.0	0.0	299.4	61.9	196.9	6.6	2.0	0	1	0	...	50.9
9	75.0	0.0	166.7	148.3	186.9	10.1	3.0	0	1	0	...	28.3
...	...	...	...	...	...	...	...	...	...	...	...	.
6657	192.0	36.0	156.2	215.5	279.1	9.9	2.0	0	0	1	...	26.5
6659	68.0	0.0	231.1	153.4	191.3	9.6	3.0	0	0	0	...	39.2
6661	28.0	0.0	180.8	288.8	191.9	14.1	2.0	0	0	0	...	30.7
6663	184.0	0.0	213.8	159.6	139.2	5.0	2.0	0	1	0	...	36.3
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0	0	1	...	39.8

3333 rows × 22 columns

In [439]:

```
data1.head()
```

Out[439]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Churn	Intl Plan	VMail Plan	...	Daily Charges MV
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	0	0	1	...	45.07
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	0	0	1	...	27.47
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0	0	0	...	41.38
7	84.0	0.0	299.4	61.9	196.9	6.6	2.0	0	1	0	...	50.90
9	75.0	0.0	166.7	148.3	186.9	10.1	3.0	0	1	0	...	28.34

5 rows × 22 columns

In [440]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 22 columns):
Account Length      3333 non-null float64
VMail Message       3333 non-null float64
Day Mins            3333 non-null float64
Eve Mins            3333 non-null float64
Night Mins          3333 non-null float64
Intl Mins           3333 non-null float64
CustServ Calls      3333 non-null float64
Churn               3333 non-null object
Intl Plan           3333 non-null object
VMail Plan          3333 non-null object
Day Calls           3333 non-null float64
Day Charge          3333 non-null float64
Daily Charges MV    3283 non-null float64
Eve Calls           3333 non-null float64
Eve Charge          3333 non-null float64
Night Calls         3333 non-null float64
Night Charge        3333 non-null float64
Intl Calls          3333 non-null float64
Intl Charge         3333 non-null float64
State               3333 non-null object
Area Code           3333 non-null object
Phone               3333 non-null object
dtypes: float64(16), object(6)
memory usage: 598.9+ KB
```

In [441]:

```
data1.isnull().sum()
```

Out[441]:

```
Account Length       0
VMail Message        0
Day Mins             0
Eve Mins             0
Night Mins           0
Intl Mins            0
CustServ Calls       0
Churn                0
Intl Plan            0
VMail Plan           0
Day Calls            0
Day Charge           0
Daily Charges MV    50
Eve Calls            0
Eve Charge           0
Night Calls          0
Night Charge         0
Intl Calls           0
Intl Charge          0
State                0
Area Code            0
Phone                0
dtype: int64
```

In [442]:

```
data1["Daily Charges MV"].fillna(data1["Daily Charges MV"].median(),inplace=True)
```

In [443]:

```
data1.isnull().sum()
```

Out[443]:

```
Account Length      0
VMail Message       0
Day Mins            0
Eve Mins            0
Night Mins          0
Intl Mins           0
CustServ Calls      0
Churn               0
Intl Plan           0
VMail Plan          0
Day Calls           0
Day Charge          0
Daily Charges MV    0
Eve Calls           0
Eve Charge          0
Night Calls         0
Night Charge        0
Intl Calls          0
Intl Charge         0
State               0
Area Code           0
Phone               0
dtype: int64
```

In [444]:

```
data1=data1.drop(["State","Area Code","Phone","Day Charge","Eve Charge"],axis=1)
data1
```

Out[444]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Churn	Intl Plan	VMail Plan	Day Calls	I Cha
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	0	0	1	110.0	4
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	0	0	1	123.0	2
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0	0	0	114.0	4
7	84.0	0.0	299.4	61.9	196.9	6.6	2.0	0	1	0	71.0	5
9	75.0	0.0	166.7	148.3	186.9	10.1	3.0	0	1	0	113.0	2
...	...	...	...	...	...	...	...	...	...	...	...	
6657	192.0	36.0	156.2	215.5	279.1	9.9	2.0	0	0	1	77.0	2
6659	68.0	0.0	231.1	153.4	191.3	9.6	3.0	0	0	0	57.0	3
6661	28.0	0.0	180.8	288.8	191.9	14.1	2.0	0	0	0	109.0	3
6663	184.0	0.0	213.8	159.6	139.2	5.0	2.0	0	1	0	105.0	3
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0	0	1	113.0	3

3333 rows × 17 columns

In [445]:

```
data1.head()
```

Out[445]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Churn	Intl Plan	VMail Plan	Day Calls	Dail Charge M
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	0	0	1	110.0	45.0
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	0	0	1	123.0	27.4
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0	0	0	114.0	41.3
7	84.0	0.0	299.4	61.9	196.9	6.6	2.0	0	1	0	71.0	50.9
9	75.0	0.0	166.7	148.3	186.9	10.1	3.0	0	1	0	113.0	28.3

In [446]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 17 columns):
Account Length      3333 non-null float64
VMail Message       3333 non-null float64
Day Mins            3333 non-null float64
Eve Mins            3333 non-null float64
Night Mins          3333 non-null float64
Intl Mins           3333 non-null float64
CustServ Calls      3333 non-null float64
Churn               3333 non-null object
Intl Plan           3333 non-null object
VMail Plan          3333 non-null object
Day Calls           3333 non-null float64
Daily Charges MV    3333 non-null float64
Eve Calls           3333 non-null float64
Night Calls         3333 non-null float64
Night Charge        3333 non-null float64
Intl Calls          3333 non-null float64
Intl Charge         3333 non-null float64
dtypes: float64(14), object(3)
memory usage: 468.7+ KB
```

In [447]:

```
def treat_outliers (x,q=0.05):
    upper=x.quantile(1-q)
    lower=x.quantile(q)
    mask=(x<upper) & (x>lower)
    return mask
mask=treat_outliers(data1,0.05)
```

In [448]:

```
mask
```

Out[448]:

	Account Length	Churn	CustServ Calls	Daily Charges MV	Day Calls	Day Mins	Eve Calls	Eve Mins	Intl Calls	Intl Charge	Intl Mins	P
1	True	False	True	True	True	True	True	True	True	True	True	Fa
3	True	False	True	True	True	True	True	True	True	True	True	Fa
5	True	False	False	True	True	True	True	True	True	True	True	Fa
7	True	False	True	False	True	False	True	False	True	True	True	Fa
9	True	False	True	True	True	True	True	True	True	True	True	Fa
...	...	...	...	...	...	...	...	...	...	...	...	
6657	False	False	True	True	True	True	True	True	True	True	True	Fa
6659	True	False	True	True	False	True	False	True	True	True	True	Fa
6661	False	False	True	True	True	True	False	False	True	True	True	Fa
6663	False	False	True	True	True	True	True	True	False	False	False	Fa
6665	True	False	False	True	True	True	True	True	True	True	True	Fa

3333 rows × 17 columns

In [449]:

```
#for i in data1:
    Q1=data1[i].quantile(0.25)
    print(Q1)
    Q3=data1[i].quantile(0.75)
    print(Q3)
    IQR=Q3-Q1
    print(IQR)
    data1=data1[(data1[i]>=(Q1-1.5*IQR))&(data1[i]<=(Q3+1.5*IQR))]
#k=data.loc[filter]
#k
    sns.boxplot(data1[i])
    plt.show()
```

```
  File "<ipython-input-449-59f57e121bcd>", line 2
    Q1=data1[i].quantile(0.25)
    ^
IndentationError: unexpected indent
```

In [450]:

```
y=data1["Churn"]
y
```

Out[450]:

```
1       0
3       0
5       0
7       0
9       0
       ..
6657    0
6659    0
6661    0
6663    0
6665    0
Name: Churn, Length: 3333, dtype: object
```

In [451]:

```
data1=data1.drop(["Churn"],axis=1)
data1
```

Out[451]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Intl Plan	VMail Plan	Day Calls	Daily Charges MV	C
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	0	1	110.0	45.07	9
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	0	1	123.0	27.47	1(
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0	0	114.0	41.38	1
7	84.0	0.0	299.4	61.9	196.9	6.6	2.0	1	0	71.0	50.90	8
9	75.0	0.0	166.7	148.3	186.9	10.1	3.0	1	0	113.0	28.34	1:
...	...	...	...	...	...	...	...	...	...	...	...	
6657	192.0	36.0	156.2	215.5	279.1	9.9	2.0	0	1	77.0	26.55	1:
6659	68.0	0.0	231.1	153.4	191.3	9.6	3.0	0	0	57.0	39.29	5
6661	28.0	0.0	180.8	288.8	191.9	14.1	2.0	0	0	109.0	30.74	5
6663	184.0	0.0	213.8	159.6	139.2	5.0	2.0	1	0	105.0	36.35	8
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0	1	113.0	39.85	8

3333 rows × 16 columns

In [452]:

```
x=data1
x
```

Out[452]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Intl Plan	VMail Plan	Day Calls	Daily Charges MV	C
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	0	1	110.0	45.07	9
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	0	1	123.0	27.47	1
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	0	0	114.0	41.38	1
7	84.0	0.0	299.4	61.9	196.9	6.6	2.0	1	0	71.0	50.90	8
9	75.0	0.0	166.7	148.3	186.9	10.1	3.0	1	0	113.0	28.34	1
...	...	...	...	...	...	...	...	...	...	...	...	
6657	192.0	36.0	156.2	215.5	279.1	9.9	2.0	0	1	77.0	26.55	1
6659	68.0	0.0	231.1	153.4	191.3	9.6	3.0	0	0	57.0	39.29	5
6661	28.0	0.0	180.8	288.8	191.9	14.1	2.0	0	0	109.0	30.74	5
6663	184.0	0.0	213.8	159.6	139.2	5.0	2.0	1	0	105.0	36.35	8
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	0	1	113.0	39.85	8

3333 rows × 16 columns

In [453]:

```
y
```

Out[453]:

```
1       0
3       0
5       0
7       0
9       0
       ..
6657    0
6659    0
6661    0
6663    0
6665    0
Name: Churn, Length: 3333, dtype: object
```

In [454]:

```
x["Intl Plan"]=x["Intl Plan"].astype(object)
```

In [455]:

```
x["VMail Plan"]=x["VMail Plan"].astype(object)
```

In [456]:

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 16 columns):
Account Length      3333 non-null float64
VMail Message       3333 non-null float64
Day Mins            3333 non-null float64
Eve Mins            3333 non-null float64
Night Mins          3333 non-null float64
Intl Mins           3333 non-null float64
CustServ Calls      3333 non-null float64
Intl Plan           3333 non-null object
VMail Plan          3333 non-null object
Day Calls           3333 non-null float64
Daily Charges MV    3333 non-null float64
Eve Calls           3333 non-null float64
Night Calls         3333 non-null float64
Night Charge        3333 non-null float64
Intl Calls          3333 non-null float64
Intl Charge         3333 non-null float64
dtypes: float64(14), object(2)
memory usage: 442.7+ KB
```

In [457]:

```
x.nunique()
```

Out[457]:

```
Account Length       212
VMail Message         46
Day Mins            1667
Eve Mins            1611
Night Mins          1591
Intl Mins            162
CustServ Calls        10
Intl Plan              2
VMail Plan             2
Day Calls            119
Daily Charges MV    1649
Eve Calls            123
Night Calls          120
Night Charge         933
Intl Calls            21
Intl Charge          162
dtype: int64
```

In [458]:

```
x=pd.get_dummies(x)
x
```

Out[458]:

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Day Calls	Daily Charges MV	Eve Calls	Night Calls
1	128.0	25.0	265.1	197.4	244.7	10.0	1.0	110.0	45.07	99.0	91.0
3	107.0	26.0	161.6	195.5	254.4	13.7	1.0	123.0	27.47	103.0	103.0
5	137.0	0.0	243.4	121.2	162.6	12.2	0.0	114.0	41.38	110.0	104.0
7	84.0	0.0	299.4	61.9	196.9	6.6	2.0	71.0	50.90	88.0	89.0
9	75.0	0.0	166.7	148.3	186.9	10.1	3.0	113.0	28.34	122.0	121.0
...	...	...	...	...	...	...	...	...	...	...	...
6657	192.0	36.0	156.2	215.5	279.1	9.9	2.0	77.0	26.55	126.0	83.0
6659	68.0	0.0	231.1	153.4	191.3	9.6	3.0	57.0	39.29	55.0	123.0
6661	28.0	0.0	180.8	288.8	191.9	14.1	2.0	109.0	30.74	58.0	91.0
6663	184.0	0.0	213.8	159.6	139.2	5.0	2.0	105.0	36.35	84.0	137.0
6665	74.0	25.0	234.4	265.9	241.4	13.7	0.0	113.0	39.85	82.0	77.0

3333 rows × 18 columns

In [459]:

```
len(x)
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3333 entries, 1 to 6665
Data columns (total 18 columns):
Account Length      3333 non-null float64
VMail Message       3333 non-null float64
Day Mins            3333 non-null float64
Eve Mins            3333 non-null float64
Night Mins          3333 non-null float64
Intl Mins           3333 non-null float64
CustServ Calls      3333 non-null float64
Day Calls           3333 non-null float64
Daily Charges MV    3333 non-null float64
Eve Calls           3333 non-null float64
Night Calls         3333 non-null float64
Night Charge        3333 non-null float64
Intl Calls          3333 non-null float64
Intl Charge         3333 non-null float64
Intl Plan_0.0       3333 non-null uint8
Intl Plan_1.0       3333 non-null uint8
VMail Plan_0.0      3333 non-null uint8
VMail Plan_1.0      3333 non-null uint8
dtypes: float64(14), uint8(4)
memory usage: 403.6 KB
```

In [460]:

```
len(y)
y=y.astype('int')
y
```

Out[460]:

```
1       0
3       0
5       0
7       0
9       0
       ..
6657    0
6659    0
6661    0
6663    0
6665    0
Name: Churn, Length: 3333, dtype: int32
```

In [461]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=10)
```

In [407]:

```
clf=LogisticRegression(class_weight='balanced')
x.shape
```

Out[407]:

```
(3333, 18)
```

In [408]:

```
clf.fit(x_train,y_train)
```

```
C:\Users\admin\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:
432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Speci
fy a solver to silence this warning.
  FutureWarning)
```

Out[408]:

```
LogisticRegression(C=1.0, class_weight='balanced', dual=False,
                   fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                   max_iter=100, multi_class='warn', n_jobs=None, penalty='l
2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [411]:

```
preds = clf.predict(x_test)
preds
```

Out[411]:

```
array([0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
       1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0,
       1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0,
       0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 1, 0, 0, 0])
```

In [412]:

```
confusion_matrix(y_test,preds)
```

Out[412]:

```
array([[441, 131],
       [ 24,  71]], dtype=int64)
```

In [413]:

```
print(classification_report(y_test, preds))
```

```
              precision    recall  f1-score   support

           0       0.95      0.77      0.85       572
           1       0.35      0.75      0.48        95

    accuracy                           0.77       667
   macro avg       0.65      0.76      0.66       667
weighted avg       0.86      0.77      0.80       667
```

In [416]:

```
preds1=clf.predict_proba(x_test)
preds1
```

Out[416]:

```
array([[0.53110796, 0.46889204],
       [0.73799755, 0.26200245],
       [0.16260804, 0.83739196],
       ...,
       [0.61829884, 0.38170116],
       [0.83750809, 0.16249191],
       [0.83467317, 0.16532683]])
```

Decision tree

In [468]:

```
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier()
```

In [469]:

```
clf=clf.fit(x_train,y_train)
```

In [472]:

```
y_predict=clf.predict(x_test)
y_predict
```

Out[472]:

```
array([0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0])
```

In [475]:

```
print("Accuracy:",metrics.accuracy_score(y_test,y_predict))
```

```
Accuracy: 0.9130434782608695
```

Optimisation of Decision Tree

In [476]:

```
clf=DecisionTreeClassifier(criterion="entropy",max_depth=3)
```

In [477]:

```
clf=clf.fit(x_train,y_train)
```

In [479]:

```
y_pred=clf.predict(x_test)
```

In [480]:

```
y_pred
```

Out[480]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0])
```

In [483]:

```
print("Accuracy:",metrics.accuracy_score(y_test,y_pred))
```

```
Accuracy: 0.9010494752623688
```

Random Forest

In [484]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [485]:

```
clf=RandomForestClassifier(n_estimators=100)
```

In [486]:

```
clf.fit(x_train,y_train)
```

Out[486]:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=N
one,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

In [487]:

```
y_predict=clf.predict(x_test)
```

In [488]:

```
print("Acuuracy:",metrics.accuracy_score(y_test,y_predict))
```

```
Acuuracy: 0.9535232383808095
```

Type *Markdown* and LaTeX: α^2