In [52]:
```python
import pandas as pd
```

In [53]:
```python
data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

In [54]:
```python
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')
```

In [55]:
```python
data.describe()
```

Out[55]:

|       | SeniorCitizen | tenure | MonthlyCharges | TotalCharges |
|-------|---------------|--------|----------------|--------------|
| count | 7043.000000 | 7043.000000 | 7043.000000 | 7032.000000 |
| mean | 0.162147 | 32.371149 | 64.761692 | 2283.300441 |
| std | 0.368612 | 24.559481 | 30.090047 | 2266.771362 |
| min | 0.000000 | 0.000000 | 18.250000 | 18.800000 |
| 25% | 0.000000 | 9.000000 | 35.500000 | 401.450000 |
| 50% | 0.000000 | 29.000000 | 70.350000 | 1397.475000 |
| 75% | 0.000000 | 55.000000 | 89.850000 | 3794.737500 |
| max | 1.000000 | 72.000000 | 118.750000 | 8684.800000 |

In [56]: `data.head()`

Out[56]:

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | ... | DeviceProtec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7590-VHVEG | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | ... | |
| 1 | 5575-GNVDE | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | ... | |
| 2 | 3668-QPYBK | Male | 0 | No | No | 2 | Yes | No | DSL | Yes | ... | |
| 3 | 7795-CFOCW | Male | 0 | No | No | 45 | No | No phone service | DSL | Yes | ... | |
| 4 | 9237-HQITU | Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | ... | |

5 rows × 21 columns

In [57]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7032 non-null   float64
 20  Churn             7043 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

In [58]: `list(data)`

Out[58]: 
```
['customerID',
 'gender',
 'SeniorCitizen',
 'Partner',
 'Dependents',
 'tenure',
 'PhoneService',
 'MultipleLines',
 'InternetService',
 'OnlineSecurity',
 'OnlineBackup',
 'DeviceProtection',
 'TechSupport',
 'StreamingTV',
 'StreamingMovies',
 'Contract',
 'PaperlessBilling',
 'PaymentMethod',
 'MonthlyCharges',
 'TotalCharges',
 'Churn']
```

In [59]: `data=data.fillna(data.median())`

```
/tmp/ipykernel_12281/495656529.py:1: FutureWarning: The default value of numeric_only in DataFrame.median i
s deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  data=data.fillna(data.median())
```

In [60]: `data.shape`

Out[60]: (7043, 21)

In [61]: `data.dtypes`

Out[61]:
```
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
TotalCharges       float64
Churn               object
dtype: object
```

In [62]:
```python
#from sklearn.model_selection import GridSearchCV #GridSearchCV is for parameter tuning
#from sklearn.ensemble import RandomForestClassifier
#cls=RandomForestClassifier()
#n_estimators=[25,50,75,100,125,150,175,200] #number of decision trees in the forest, default = 100
#criterion=['gini','entropy'] #criteria for choosing nodes default = 'gini'
#max_depth=[3,5,10] #maximum number of nodes in a tree default = None (it will go till all possible nodes)
#parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth} #this will undergo 8*
#RFC_cls = GridSearchCV(cls, parameters)
#RFC_cls.fit(X_train,y_train)
```

In [63]: `databackup=data.copy()`

In [64]: 
```python
x=data.drop(['customerID','Churn'],axis=1)
y=data['Churn']
```
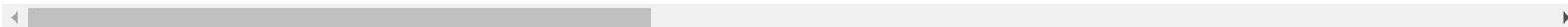
In [65]: 
```python
x=pd.get_dummies(x)
```

In [66]: 
```python
x.head()
```

Out[66]:

| | SeniorCitizen | tenure | MonthlyCharges | TotalCharges | gender_Female | gender_Male | Partner_No | Partner_Yes | Dependents_No | Dependents_Yes |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 29.85 | 29.85 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 34 | 56.95 | 1889.50 | 0 | 1 | 1 | 0 | 1 | 0 |
| 2 | 0 | 2 | 53.85 | 108.15 | 0 | 1 | 1 | 0 | 1 | 0 |
| 3 | 0 | 45 | 42.30 | 1840.75 | 0 | 1 | 1 | 0 | 1 | 0 |
| 4 | 0 | 2 | 70.70 | 151.65 | 1 | 0 | 1 | 0 | 1 | 0 |

5 rows × 45 columns

In [67]: 
```python
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.33,random_state=42)
```

In [68]:
```python
from sklearn.model_selection import GridSearchCV #GridSearchCV is for parameter tuning
from sklearn.ensemble import RandomForestClassifier
cls=RandomForestClassifier()
n_estimators=[25,50,75,100,125,150,175,200] #number of decision trees in the forest, default = 100
criterion=['gini','entropy'] #criteria for choosing nodes default = 'gini'
max_depth=[3,5,10] #maximum number of nodes in a tree default = None (it will go till all possible nodes)
parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth} #this will undergo 8*2
RFC_cls = GridSearchCV(cls, parameters)
RFC_cls.fit(x_train,y_train)
```

Out[68]:

```
▸        GridSearchCV
▸ estimator: RandomForestClassifier
     ▸ RandomForestClassifier
```

In [70]:
```python
RFC_cls.best_params_
```

Out[70]: {'criterion': 'entropy', 'max_depth': 10, 'n_estimators': 75}

In [72]:
```python
cls=RandomForestClassifier(n_estimators=75,criterion='entropy',max_depth=10)
```

In [73]:
```python
cls.fit(x_train,y_train)
```

Out[73]:

```
▾                        RandomForestClassifier
RandomForestClassifier(criterion='entropy', max_depth=10, n_estimators=75)
```

In [74]:
```python
rfy_pred=cls.predict(x_test)
rfy_pred
```

Out[74]: array(['Yes', 'No', 'No', ..., 'Yes', 'No', 'No'], dtype=object)

In [76]:
```python
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,rfy_pred)
```

Out[76]:
```
array([[1540,  157],
       [ 293,  335]])
```

In [77]:
```python
from sklearn.metrics import accuracy_score
accuracy_score(y_test,rfy_pred)
```

Out[77]:
```
0.8064516129032258
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: