In [1]: `import pandas as pd`

In [2]: `data=pd.read_csv("/home/placement/Desktop/naren/TelecomCustomerChurn.csv")`

In [3]: `data`

Out[3]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DevicePro
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	...	
...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	

7043 rows × 21 columns
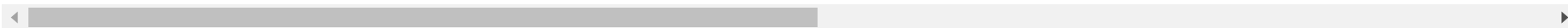
In [6]: `data.info()`

```
<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      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

In [5]: `data.isna().sum()`

Out[5]:
```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

In [7]: `data.describe()`

Out[7]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

In [8]: `data=data.drop("customerID",axis=1)`

In [9]: `data`

Out[9]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProte
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	
...	...	...	...	...	...	...	...	...	...	...	
7038	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	No	
7039	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	Yes	
7040	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	No	
7041	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	No	
7042	Male	0	No	No	66	Yes	No	Fiber optic	Yes	No	

7043 rows × 20 columns

In [10]: `data["Churn"]=data["Churn"].map({"Yes":1,"No":0})`

In [11]: `data`

Out[11]:

OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCha
Yes	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	
No	Yes	No	No	No	One year	No	Mailed check	56.95	1
Yes	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	1
No	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	18
No	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	1
...	...	...	...	...	...	...	...	...	
No	Yes	Yes	Yes	Yes	One year	Yes	Mailed check	84.80	1
Yes	Yes	No	Yes	Yes	One year	Yes	Credit card (automatic)	103.20	7
No	No	No	No	No	Month-to-month	Yes	Electronic check	29.60	3
No	No	No	No	No	Month-to-month	Yes	Mailed check	74.40	
No	Yes	Yes	Yes	Yes	Two year	Yes	Bank transfer (automatic)	105.65	6

In [12]: `data['TotalCharges']=pd.to_numeric(data['TotalCharges'],errors='coerce')`

In [13]: `data`

Out[13]:

OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCha
Yes	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	
No	Yes	No	No	No	One year	No	Mailed check	56.95	18
Yes	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	1
No	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	18
No	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	1
...	...	...	...	...	...	...	...	...	
No	Yes	Yes	Yes	Yes	One year	Yes	Mailed check	84.80	19
Yes	Yes	No	Yes	Yes	One year	Yes	Credit card (automatic)	103.20	73
No	No	No	No	No	Month-to-month	Yes	Electronic check	29.60	3
No	No	No	No	No	Month-to-month	Yes	Mailed check	74.40	3
No	Yes	Yes	Yes	Yes	Two year	Yes	Bank transfer (automatic)	105.65	68

In [14]: `data['TotalCharges']=data['TotalCharges'].fillna(data['TotalCharges'].median())`

In [15]: `data`

Out[15]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProte
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	
...	...	...	...	...	...	...	...	...	...	...	
7038	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	No	
7039	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	Yes	
7040	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	No	
7041	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	No	
7042	Male	0	No	No	66	Yes	No	Fiber optic	Yes	No	

7043 rows × 20 columns

In [16]: `data["SeniorCitizen"]=data["SeniorCitizen"].map({0:"No",1:"Yes"})`

In [17]: `data`

Out[17]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProte
0	Female	No	Yes	No	1	No	No phone service	DSL	No	Yes	
1	Male	No	No	No	34	Yes	No	DSL	Yes	No	
2	Male	No	No	No	2	Yes	No	DSL	Yes	Yes	
3	Male	No	No	No	45	No	No phone service	DSL	Yes	No	
4	Female	No	No	No	2	Yes	No	Fiber optic	No	No	
...	...	...	...	...	...	...	...	...	...	...	
7038	Male	No	Yes	Yes	24	Yes	Yes	DSL	Yes	No	
7039	Female	No	Yes	Yes	72	Yes	Yes	Fiber optic	No	Yes	
7040	Female	No	Yes	Yes	11	No	No phone service	DSL	Yes	No	
7041	Male	Yes	Yes	No	4	Yes	Yes	Fiber optic	No	No	
7042	Male	No	No	No	66	Yes	No	Fiber optic	Yes	No	

7043 rows × 20 columns

In [18]: `x=data.drop(['Churn'],axis=1)`

In [19]: `y=data['Churn']`

In [20]: `x=pd.get_dummies(x,dtype=int)`

In [21]: `x.head()`

Out[21]:

	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	SeniorCitizen_No	SeniorCitizen_Yes	Partner_No	Partner_Yes	Dependents
0	1	29.85	29.85	1	0	1	0	0	1	
1	34	56.95	1889.50	0	1	1	0	1	0	
2	2	53.85	108.15	0	1	1	0	1	0	
3	45	42.30	1840.75	0	1	1	0	1	0	
4	2	70.70	151.65	1	0	1	0	1	0	

5 rows × 46 columns

In [22]:
```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 [26]:
```python
rn.model_selection import GridSearchCV #GridSearchCV is for parameter tuning
rn.ensemble import RandomForestClassifier
ForestClassifier()
rs=[25,50,75,100,125,150,175,200] #number of decision trees in the forest, default = 100
['gini','entropy'] #criteria for choosing nodes default = 'gini'
[3,5,10] #maximum number of nodes in a tree default = None (it will go till all possible nodes)
={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth} #this will undergo 8*2*3 = 48 ite
GridSearchCV(cls, parameters)
t(x_train,y_train)
```

Out[26]:

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

In [27]: `RFC_cls.best_params_`

Out[27]: `{'criterion': 'entropy', 'max_depth': 10, 'n_estimators': 50}`

In [29]:
```python
cls=RandomForestClassifier(n_estimators=50,criterion='entropy',max_depth=10)
```

In [30]:
```python
cls.fit(x_train,y_train)
```

Out[30]:
```
                                 ▼                RandomForestClassifier

RandomForestClassifier(criterion='entropy', max_depth=10, n_estimators=50)
```

In [34]:
```python
rfy_pred=cls.predict(x_test)
```

In [35]:
```python
rfy_pred
```

Out[35]: `array([1, 0, 0, ..., 1, 0, 0])`

In [36]:
```python
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,rfy_pred)
```

Out[36]:
```
array([[1549,  148],
       [ 300,  328]])
```

In [37]:
```python
from sklearn.metrics import accuracy_score
accuracy_score(y_test,rfy_pred)
```

Out[37]: `0.8073118279569892`

In [ ]: