

```
In [23]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
In [24]: data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

```
In [25]: data.head(10)
```

```
Out[25]:
```

	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	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	...	
6	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	...	
7	6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL	Yes	...	
8	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	...	
9	6388-TABGU	Male	0	No	Yes	62	Yes	No	DSL	Yes	...	

10 rows × 21 columns



```
In [26]: data.shape
```

```
Out[26]: (7043, 21)
```

```
In [27]: list(data)
```

```
Out[27]: ['customerID',  
          'gender',  
          'SeniorCitizen',  
          'Partner',  
          'Dependents',  
          'tenure',  
          'PhoneService',  
          'MultipleLines',  
          'InternetService',  
          'OnlineSecurity',  
          'OnlineBackup',  
          'DeviceProtection',  
          'TechSupport',  
          'StreamingTV',  
          'StreamingMovies',  
          'Contract',  
          'PaperlessBilling',  
          'PaymentMethod',  
          'MonthlyCharges',  
          'TotalCharges',  
          'Churn']
```

```
In [28]: data.isnull().sum()
```

```
Out[28]: 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 [29]: data1=data.drop(['customerID'],axis=1)
```

```
In [30]: data1=pd.get_dummies(data)
```

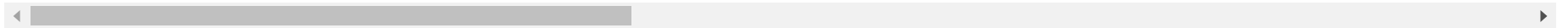
```
In [ ]:
```

```
In [31]: data1.head()
```

```
Out[31]:
```

	SeniorCitizen	tenure	MonthlyCharges	customerID_0002-ORFBO	customerID_0003-MKNFE	customerID_0004-TLHLJ	customerID_0011-IGKFF	customerID_0013-EXCHZ	customerID_0014-EXCHZ
0	0	1	29.85	0	0	0	0	0	0
1	0	34	56.95	0	0	0	0	0	0
2	0	2	53.85	0	0	0	0	0	0
3	0	45	42.30	0	0	0	0	0	0
4	0	2	70.70	0	0	0	0	0	0

5 rows × 13620 columns



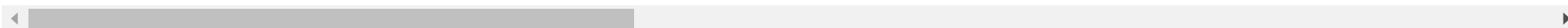
```
In [32]: data1['SeniorCitizen']=data1['SeniorCitizen'].map({'Yes':1,'No':0})
```

In [33]: data1

Out[33]:

	SeniorCitizen	tenure	MonthlyCharges	customerID_0002-ORFBO	customerID_0003-MKNFE	customerID_0004-TLHLJ	customerID_0011-IGKFF	customerID_0013-EXCHZ	custom
0	NaN	1	29.85	0	0	0	0	0	
1	NaN	34	56.95	0	0	0	0	0	
2	NaN	2	53.85	0	0	0	0	0	
3	NaN	45	42.30	0	0	0	0	0	
4	NaN	2	70.70	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
7038	NaN	24	84.80	0	0	0	0	0	
7039	NaN	72	103.20	0	0	0	0	0	
7040	NaN	11	29.60	0	0	0	0	0	
7041	NaN	4	74.40	0	0	0	0	0	
7042	NaN	66	105.65	0	0	0	0	0	

7043 rows × 13620 columns



In [34]: `y=data['Churn']`  
`x=data.drop(columns='Churn')`

In [35]: `x=pd.get_dummies(x)`

In [36]: `x.head()`

Out[36]:

	SeniorCitizen	tenure	MonthlyCharges	customerID_0002-ORFBO	customerID_0003-MKNFE	customerID_0004-TLHLJ	customerID_0011-IGKFF	customerID_0013-EXCHZ	customerID_0014-EXCHZ
0	0	1	29.85	0	0	0	0	0	0
1	0	34	56.95	0	0	0	0	0	0
2	0	2	53.85	0	0	0	0	0	0
3	0	45	42.30	0	0	0	0	0	0
4	0	2	70.70	0	0	0	0	0	0

5 rows × 13618 columns

In [37]: `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 [38]: `x_test.head(5)`

Out[38]:

	SeniorCitizen	tenure	MonthlyCharges	customerID_0002-ORFBO	customerID_0003-MKNFE	customerID_0004-TLHLJ	customerID_0011-IGKFF	customerID_0013-EXCHZ	customerID_0014-EXCHZ
185	0	1	24.80	0	0	0	0	0	0
2715	0	41	25.25	0	0	0	0	0	0
3825	0	52	19.35	0	0	0	0	0	0
1807	0	1	76.35	0	0	0	0	0	0
132	0	67	50.55	0	0	0	0	0	0

5 rows × 13618 columns

```
In [39]: x_test.shape
```

```
Out[39]: (2325, 13618)
```

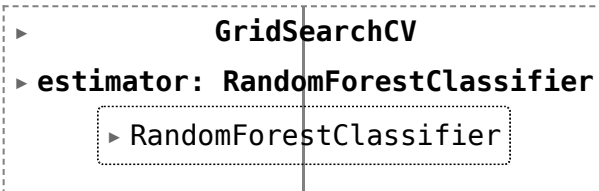
```
In [40]: y_test.head(5)
```

```
Out[40]: 185      Yes
2715     No
3825     No
1807     Yes
132      No
Name: Churn, dtype: object
```

## RandomForestClassifier

```
In [41]: 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[41]:
```



```
  ▶ GridSearchCV
  ▶ estimator: RandomForestClassifier
    ▶ RandomForestClassifier
```

```
In [42]: RFC_cls.best_params_
```

```
Out[42]: {'criterion': 'gini', 'max_depth': 3, 'n_estimators': 25}
```



```
In [43]: y_pred=RFC_cls.predict(x_test)
```

```
In [44]: y_pred
```

```
Out[44]: array(['No', 'No', 'No', ..., 'No', 'No', 'No'], dtype=object)
```

```
In [45]: from sklearn.metrics import confusion_matrix  
confusion_matrix(y_test,y_pred)
```

```
Out[45]: array([[1697,    0],  
               [ 628,    0]])
```

```
In [46]: from sklearn.metrics import accuracy_score  
accuracy_score(y_test,y_pred)
```

```
Out[46]: 0.7298924731182795
```