

```
In [61]: import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

```
In [62]: data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

```
In [63]: data.describe()
```

```
Out[63]:
```

	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 [64]: `data.head()`

Out[64]:

	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 [65]: 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 [66]: data.shape

Out[66]: (7043, 21)

```
In [67]: list(data)
```

```
Out[67]: ['customerID',  
          'gender',  
          'SeniorCitizen',  
          'Partner',  
          'Dependents',  
          'tenure',  
          'PhoneService',  
          'MultipleLines',  
          'InternetService',  
          'OnlineSecurity',  
          'OnlineBackup',  
          'DeviceProtection',  
          'TechSupport',  
          'StreamingTV',  
          'StreamingMovies',  
          'Contract',  
          'PaperlessBilling',  
          'PaymentMethod',  
          'MonthlyCharges',  
          'TotalCharges',  
          'Churn']
```

```
In [68]: data.isna().sum()
```

```
Out[68]: 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 [69]: data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')
```

```
In [70]: 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 [71]: data['TotalCharges']=data['TotalCharges'].fillna(data['TotalCharges'].median())
```

```
In [72]: data.isna().sum()
```

```
Out[72]: 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 [73]: data['SeniorCitizen']=data['SeniorCitizen'].map({0:'No',1:'Yes'})
```

```
In [74]: x=data.drop(['customerID','Churn'],axis=1)  
y=data['Churn']
```

```
In [75]: x=pd.get_dummies(x)
```

In [76]: `x.head()`

Out[76]:

	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	SeniorCitizen_No	SeniorCitizen_Yes	Partner_No	Partner_Yes	Dependent
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 [77]: `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 [78]: `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[78]: `GridSearchCV(estimator=RandomForestClassifier(),`
`param_grid={'criterion': ['gini', 'entropy'],`
`'max_depth': [3, 5, 10],`
`'n_estimators': [25, 50, 75, 100, 125, 150, 175, 200]})`

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
 On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [79]: data

Out[79]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DevicePro
0	7590-VHVEG	Female	No	Yes	No	1	No	No phone service	DSL	No	...	
1	5575-GNVDE	Male	No	No	No	34	Yes	No	DSL	Yes	...	
2	3668-QPYBK	Male	No	No	No	2	Yes	No	DSL	Yes	...	
3	7795-CFOCW	Male	No	No	No	45	No	No phone service	DSL	Yes	...	
4	9237-HQITU	Female	No	No	No	2	Yes	No	Fiber optic	No	...	
...	
7038	6840-RESVB	Male	No	Yes	Yes	24	Yes	Yes	DSL	Yes	...	
7039	2234-XADUH	Female	No	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	
7040	4801-JJAZL	Female	No	Yes	Yes	11	No	No phone service	DSL	Yes	...	
7041	8361-LTMKD	Male	Yes	Yes	No	4	Yes	Yes	Fiber optic	No	...	
7042	3186-AJIEK	Male	No	No	No	66	Yes	No	Fiber optic	Yes	...	

7043 rows × 21 columns



In [80]: RFC_cls.best_params_

Out[80]: {'criterion': 'gini', 'max_depth': 10, 'n_estimators': 200}

In [82]: cls=RandomForestClassifier(n_estimators=200,criterion='entropy',max_depth=10)

```
In [84]: cls.fit(x_train,y_train)
```

```
Out[84]: RandomForestClassifier(criterion='entropy', max_depth=10, n_estimators=200)
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
In [85]: rfy_pred=cls.predict(x_test)
```

```
In [86]: rfy_pred
```

```
Out[86]: array(['Yes', 'No', 'No', ..., 'Yes', 'No', 'No'], dtype=object)
```

```
In [87]: from sklearn.metrics import confusion_matrix  
confusion_matrix(y_test,rfy_pred)
```

```
Out[87]: array([[1548,  149],  
               [ 295,  333]])
```

```
In [88]: from sklearn.metrics import accuracy_score  
accuracy_score(y_test,rfy_pred)
```

```
Out[88]: 0.8090322580645162
```

```
In [89]: import warnings  
warnings.filterwarnings("ignore")  
from sklearn.linear_model import LogisticRegression  
classifier= LogisticRegression()  
classifier.fit(x_train,y_train)
```

```
Out[89]: LogisticRegression()
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
In [90]: y_pred=classifier.predict(x_test)
```

```
In [91]: y_pred
```

```
Out[91]: array(['Yes', 'No', 'No', ..., 'Yes', 'No', 'No'], dtype=object)
```

```
In [92]: from sklearn.metrics import confusion_matrix  
confusion_matrix(y_test,rfy_pred)
```

```
Out[92]: array([[1548, 149],  
               [ 295, 333]])
```

```
In [93]: from sklearn.metrics import accuracy_score  
accuracy_score(y_test,rfy_pred)
```

```
Out[93]: 0.8090322580645162
```

```
In [ ]:
```