

```
In [285]: import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

```
In [286]: data=pd.read_csv("/home/placement/Desktop/divyasri/Titanic Dataset.csv")
```

```
In [287]: data.describe()
```

Out[287]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
In [288]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     891 non-null    int64
1   Survived        891 non-null    int64
2   Pclass         891 non-null    int64
3   Name            891 non-null    object
4   Sex             891 non-null    object
5   Age            714 non-null    float64
6   SibSp          891 non-null    int64
7   Parch          891 non-null    int64
8   Ticket         891 non-null    object
9   Fare           891 non-null    float64
10  Cabin          204 non-null    object
11  Embarked       889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
In [289]: list(data)
```

```
Out[289]: ['PassengerId',
           'Survived',
           'Pclass',
           'Name',
           'Sex',
           'Age',
           'SibSp',
           'Parch',
           'Ticket',
           'Fare',
           'Cabin',
           'Embarked']
```

```
In [290]: data.isna().sum()
```

```
Out[290]: PassengerId      0
          Survived        0
          Pclass          0
          Name            0
          Sex             0
          Age            177
          SibSp           0
          Parch           0
          Ticket          0
          Fare            0
          Cabin          687
          Embarked        2
          dtype: int64
```

```
In [291]: data1=data.drop(['Name', 'PassengerId', 'Ticket', 'Cabin', 'SibSp', 'Parch'],axis=1)
```

```
In [292]: data1
```

```
Out[292]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	male	22.0	7.2500	S
1	1	1	female	38.0	71.2833	C
2	1	3	female	26.0	7.9250	S
3	1	1	female	35.0	53.1000	S
4	0	3	male	35.0	8.0500	S
...
886	0	2	male	27.0	13.0000	S
887	1	1	female	19.0	30.0000	S
888	0	3	female	NaN	23.4500	S
889	1	1	male	26.0	30.0000	C
890	0	3	male	32.0	7.7500	Q

891 rows × 6 columns

```
In [293]: data1['Sex']=data1['Sex'].map({'male':1,'female':0})
```

```
In [294]: data1
```

```
Out[294]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	0	38.0	71.2833	C
2	1	3	0	26.0	7.9250	S
3	1	1	0	35.0	53.1000	S
4	0	3	1	35.0	8.0500	S
...
886	0	2	1	27.0	13.0000	S
887	1	1	0	19.0	30.0000	S
888	0	3	0	NaN	23.4500	S
889	1	1	1	26.0	30.0000	C
890	0	3	1	32.0	7.7500	Q

891 rows × 6 columns

```
In [295]: data1=data1.fillna(data1.median())
```

```
In [296]: data1
```

```
Out[296]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	0	38.0	71.2833	C
2	1	3	0	26.0	7.9250	S
3	1	1	0	35.0	53.1000	S
4	0	3	1	35.0	8.0500	S
...
886	0	2	1	27.0	13.0000	S
887	1	1	0	19.0	30.0000	S
888	0	3	0	28.0	23.4500	S
889	1	1	1	26.0	30.0000	C
890	0	3	1	32.0	7.7500	Q

891 rows × 6 columns

```
In [297]: data1.isna().sum()
```

```
Out[297]: Survived    0
Pclass      0
Sex         0
Age         0
Fare        0
Embarked    2
dtype: int64
```

```
In [298]: data1['Pclass']=data1['Pclass'].map({1:'FC',2:'SC',3:'TC'})
```

In [299]: data1

Out[299]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	TC	1	22.0	7.2500	S
1	1	FC	0	38.0	71.2833	C
2	1	TC	0	26.0	7.9250	S
3	1	FC	0	35.0	53.1000	S
4	0	TC	1	35.0	8.0500	S
...
886	0	SC	1	27.0	13.0000	S
887	1	FC	0	19.0	30.0000	S
888	0	TC	0	28.0	23.4500	S
889	1	FC	1	26.0	30.0000	C
890	0	TC	1	32.0	7.7500	Q

891 rows × 6 columns

```
In [300]: data1=pd.get_dummies(data1)
data1
```

Out[300]:

	Survived	Sex	Age	Fare	Pclass_FC	Pclass_SC	Pclass_TC	Embarked_C	Embarked_Q	Embarked_S
0	0	1	22.0	7.2500	0	0	1	0	0	1
1	1	0	38.0	71.2833	1	0	0	1	0	0
2	1	0	26.0	7.9250	0	0	1	0	0	1
3	1	0	35.0	53.1000	1	0	0	0	0	1
4	0	1	35.0	8.0500	0	0	1	0	0	1
...
886	0	1	27.0	13.0000	0	1	0	0	0	1
887	1	0	19.0	30.0000	1	0	0	0	0	1
888	0	0	28.0	23.4500	0	0	1	0	0	1
889	1	1	26.0	30.0000	1	0	0	1	0	0
890	0	1	32.0	7.7500	0	0	1	0	1	0

891 rows × 10 columns

```
In [301]: y=data1['Survived']
x=data1.drop(['Survived'],axis=1)
```


In [302]:

y

Out[302]:

```
0      0
1      1
2      1
3      1
4      0
```

```
..
886    0
887    1
888    0
889    1
890    0
```

Name: Survived, Length: 891, dtype: int64

In [303]:

x

Out[303]:

	Sex	Age	Fare	Pclass_FC	Pclass_SC	Pclass_TC	Embarked_C	Embarked_Q	Embarked_S
0	1	22.0	7.2500	0	0	1	0	0	1
1	0	38.0	71.2833	1	0	0	1	0	0
2	0	26.0	7.9250	0	0	1	0	0	1
3	0	35.0	53.1000	1	0	0	0	0	1
4	1	35.0	8.0500	0	0	1	0	0	1
...
886	1	27.0	13.0000	0	1	0	0	0	1
887	0	19.0	30.0000	1	0	0	0	0	1
888	0	28.0	23.4500	0	0	1	0	0	1
889	1	26.0	30.0000	1	0	0	1	0	0
890	1	32.0	7.7500	0	0	1	0	1	0

891 rows × 9 columns

```
In [304]: from sklearn.model_selection import train_test_split #splitting of training and testing
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

```
In [305]: y_train.head(5)
```

```
Out[305]: 6      0
718     0
685     0
73      0
882     0
Name: Survived, dtype: int64
```

```
In [306]: x_train.head(5)
```

```
Out[306]:
```

	Sex	Age	Fare	Pclass_FC	Pclass_SC	Pclass_TC	Embarked_C	Embarked_Q	Embarked_S
6	1	54.0	51.8625	1	0	0	0	0	1
718	1	28.0	15.5000	0	0	1	0	1	0
685	1	25.0	41.5792	0	1	0	1	0	0
73	1	26.0	14.4542	0	0	1	1	0	0
882	0	22.0	10.5167	0	0	1	0	0	1

```
In [307]: y_test.head(5)
```

```
Out[307]: 709     1
439     0
840     0
720     1
39      1
Name: Survived, dtype: int64
```

```
In [308]: x_test.head(5)
```

```
Out[308]:
```

	Sex	Age	Fare	Pclass_FC	Pclass_SC	Pclass_TC	Embarked_C	Embarked_Q	Embarked_S
709	1	28.0	15.2458	0	0	1	1	0	0
439	1	31.0	10.5000	0	1	0	0	0	1
840	1	20.0	7.9250	0	0	1	0	0	1
720	0	6.0	33.0000	0	1	0	0	0	1
39	0	14.0	11.2417	0	0	1	1	0	0

```
In [309]: x.isna().sum()
```

```
Out[309]: Sex          0
Age            0
Fare           0
Pclass_FC      0
Pclass_SC      0
Pclass_TC      0
Embarked_C     0
Embarked_Q     0
Embarked_S     0
dtype: int64
```

```
In [310]: 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[310]: 

▸ GridSearchCV
  ▸ estimator: RandomForestClassifier
    ▸ RandomForestClassifier


```

```
In [324]: RFC_cls.best_params_
```

```
Out[324]: {'criterion': 'entropy', 'max_depth': 5, 'n_estimators': 125}
```

```
In [325]: cls=RandomForestClassifier(n_estimators=125,criterion='entropy',max_depth=5)
```

```
In [326]: cls.fit(x_train,y_train)
```

```
Out[326]: 

▾ RandomForestClassifier
    RandomForestClassifier(criterion='entropy', max_depth=5, n_estimators=125)


```

```
In [327]: rfy_pred=cls.predict(x_test)
```

```
In [328]: rfy_pred
```

```
Out[328]: array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 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, 1, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1,
                0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
                1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0,
                0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
                0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
                0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0,
                1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0,
                0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
                1, 0, 1, 0, 0, 0, 1, 1, 0])
```

```
In [329]: from sklearn.metrics import confusion_matrix #confusion_matrix
          confusion_matrix(y_test,rfy_pred)
```

```
Out[329]: array([[158,  17],
                [ 42,  78]])
```

```
In [323]: from sklearn.metrics import accuracy_score
          accuracy_score(y_test,rfy_pred)
```

```
Out[323]: 0.7898305084745763
```

```
In [ ]:
```