In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
data=pd.read_csv("/home/placement/Desktop/BhanuSiva4K8/Titanic Dataset.csv")
data
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [3]:
```
data.describe()
```

Out[3]:

	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 [4]:
```
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 [5]:
```
data.head()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [6]:
```
data.isna().sum()
```

Out[6]:
```
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 [7]:
```
data.dtypes
```

Out[7]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [8]:
```
data['Age'].unique()
```

Out[8]:
```
array([22.  , 38.  , 26.  , 35.  ,   nan, 54.  ,  2.  , 27.  , 14.  ,
        4.  , 58.  , 20.  , 39.  , 55.  , 31.  , 34.  , 15.  , 28.  ,
        8.  , 19.  , 40.  , 66.  , 42.  , 21.  , 18.  ,  3.  ,  7.  ,
       49.  , 29.  , 65.  , 28.5 ,  5.  , 11.  , 45.  , 17.  , 32.  ,
       16.  , 25.  ,  0.83, 30.  , 33.  , 23.  , 24.  , 46.  , 59.  ,
       71.  , 37.  , 47.  , 14.5 , 70.5 , 32.5 , 12.  ,  9.  , 36.5 ,
       51.  , 55.5 , 40.5 , 44.  ,  1.  , 61.  , 56.  , 50.  , 36.  ,
       45.5 , 20.5 , 62.  , 41.  , 52.  , 63.  , 23.5 ,  0.92, 43.  ,
       60.  , 10.  , 64.  , 13.  , 48.  ,  0.75, 53.  , 57.  , 80.  ,
       70.  , 24.5 ,  6.  ,  0.67, 30.5 ,  0.42, 34.5 , 74.  ])
```

In [9]:
```
data['Cabin'].unique()
```

Out[9]:
```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F33',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49', 'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A19',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91', 'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46', 'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B42',
       'C148'], dtype=object)
```

In [10]:
```
list(data)
```

Out[10]:
```
['PassengerId',
 'Survived',
 'Pclass',
 'Name',
 'Sex',
 'Age',
 'SibSp',
 'Parch',
 'Ticket',
 'Fare',
 'Cabin',
 'Embarked']
```

In [11]:
```
data=data.drop(['PassengerId','Name','Ticket','SibSp','Parch'],axis=1)
```

In [12]:
```
data
```

Out[12]:

	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked
0	0	3	male	22.0	7.2500	NaN	S
1	1	1	female	38.0	71.2833	C85	C
2	1	3	female	26.0	7.9250	NaN	S
3	1	1	female	35.0	53.1000	C123	S
4	0	3	male	35.0	8.0500	NaN	S
...	...	...	...	...	...	...	...
886	0	2	male	27.0	13.0000	NaN	S
887	1	1	female	19.0	30.0000	B42	S
888	0	3	female	NaN	23.4500	NaN	S
889	1	1	male	26.0	30.0000	C148	C
890	0	3	male	32.0	7.7500	NaN	Q

891 rows × 7 columns

In [13]:
```
data['Sex']=data['Sex'].map({'male':1,'female':0})
data
```

Out[13]:

	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked
0	0	3	1	22.0	7.2500	NaN	S
1	1	1	0	38.0	71.2833	C85	C
2	1	3	0	26.0	7.9250	NaN	S
3	1	1	0	35.0	53.1000	C123	S
4	0	3	1	35.0	8.0500	NaN	S
...	...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	NaN	S
887	1	1	0	19.0	30.0000	B42	S
888	0	3	0	NaN	23.4500	NaN	S
889	1	1	1	26.0	30.0000	C148	C
890	0	3	1	32.0	7.7500	NaN	Q

891 rows × 7 columns

In [14]:
```
data1=data.fillna(data.median())
```

In [15]:
```
data1
```

Out[15]:

	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked
0	0	3	1	22.0	7.2500	NaN	S
1	1	1	0	38.0	71.2833	C85	C
2	1	3	0	26.0	7.9250	NaN	S
3	1	1	0	35.0	53.1000	C123	S
4	0	3	1	35.0	8.0500	NaN	S
...	...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	NaN	S
887	1	1	0	19.0	30.0000	B42	S
888	0	3	0	28.0	23.4500	NaN	S
889	1	1	1	26.0	30.0000	C148	C
890	0	3	1	32.0	7.7500	NaN	Q

891 rows × 7 columns

In [16]:
```
data.fillna(35, inplace=True)
data
```

Out[16]:

	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked
0	0	3	1	22.0	7.2500	35	S
1	1	1	0	38.0	71.2833	C85	C
2	1	3	0	26.0	7.9250	35	S
3	1	1	0	35.0	53.1000	C123	S
4	0	3	1	35.0	8.0500	35	S
...	...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	35	S
887	1	1	0	19.0	30.0000	B42	S
888	0	3	0	35.0	23.4500	35	S
889	1	1	1	26.0	30.0000	C148	C
890	0	3	1	32.0	7.7500	35	Q

891 rows × 7 columns

In [17]:
```
data.isna().sum()
```

Out[17]:
```
Survived    0
Pclass      0
Sex         0
Age         0
Fare        0
Cabin       0
Embarked    0
dtype: int64
```

In [18]:
```
data1.isna().sum()
```

Out[18]:
```
Survived      0
Pclass        0
Sex           0
Age           0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

In [19]:
```
data1.fillna(35, inplace=True)
data1
```

Out[19]:

	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked
0	0	3	1	22.0	7.2500	35	S
1	1	1	0	38.0	71.2833	C85	C
2	1	3	0	26.0	7.9250	35	S
3	1	1	0	35.0	53.1000	C123	S
4	0	3	1	35.0	8.0500	35	S
...	...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	35	S
887	1	1	0	19.0	30.0000	B42	S
888	0	3	0	28.0	23.4500	35	S
889	1	1	1	26.0	30.0000	C148	C
890	0	3	1	32.0	7.7500	35	Q

891 rows × 7 columns

In [20]:
```
data1.isna().sum()
```

Out[20]:
```
Survived    0
Pclass      0
Sex         0
Age         0
Fare        0
Cabin       0
Embarked    0
dtype: int64
```

In [21]:
```
y=data1['Survived']
x=data1.drop('Survived',axis=1)
```

In [22]:
```
x
```

Out[22]:

	Pclass	Sex	Age	Fare	Cabin	Embarked
0	3	1	22.0	7.2500	35	S
1	1	0	38.0	71.2833	C85	C
2	3	0	26.0	7.9250	35	S
3	1	0	35.0	53.1000	C123	S
4	3	1	35.0	8.0500	35	S
...	...	...	...	...	...	...
886	2	1	27.0	13.0000	35	S
887	1	0	19.0	30.0000	B42	S
888	3	0	28.0	23.4500	35	S
889	1	1	26.0	30.0000	C148	C
890	3	1	32.0	7.7500	35	Q

891 rows × 6 columns

In [23]:
```
x=pd.get_dummies(x)
x
```

Out[23]:

	Pclass	Sex	Age	Fare	Cabin_35	Cabin_A10	Cabin_A14	Cabin_A16	Cabin_A19	Cabin_A20	...	Cabin_F2	Cabin_F33	Cabin_F38	Ca
0	3	1	22.0	7.2500	1	0	0	0	0	0	...	0	0	0	
1	1	0	38.0	71.2833	0	0	0	0	0	0	...	0	0	0	
2	3	0	26.0	7.9250	1	0	0	0	0	0	...	0	0	0	
3	1	0	35.0	53.1000	0	0	0	0	0	0	...	0	0	0	
4	3	1	35.0	8.0500	1	0	0	0	0	0	...	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
886	2	1	27.0	13.0000	1	0	0	0	0	0	...	0	0	0	
887	1	0	19.0	30.0000	0	0	0	0	0	0	...	0	0	0	
888	3	0	28.0	23.4500	1	0	0	0	0	0	...	0	0	0	
889	1	1	26.0	30.0000	0	0	0	0	0	0	...	0	0	0	
890	3	1	32.0	7.7500	1	0	0	0	0	0	...	0	0	0	

891 rows × 156 columns

In [24]:
```
y
```

Out[24]:
```
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 [25]:
```
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]:
```
x_train.isna().sum()
```

Out[26]:
```
Pclass         0
Sex            0
Age            0
Fare           0
Cabin_35       0
              ..
Cabin_T        0
Embarked_35    0
Embarked_C     0
Embarked_Q     0
Embarked_S     0
Length: 156, dtype: int64
```

Random Forest Model

In [27]:
```
#importing Random Forest Classifier from sklearn.ensemble
%time
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)
```

```
CPU times: user 8 µs, sys: 1e+03 ns, total: 9 µs
Wall time: 27.7 µs
```

Out[27]:
```
▸ GridSearchCV
▸ estimator: RandomForestClassifier
  ▸ RandomForestClassifier
```

In [28]:
```
RFC_cls.best_params_
```

Out[28]:
```
{'criterion': 'entropy', 'max_depth': 10, 'n_estimators': 200}
```

In [39]:
```
cls=RandomForestClassifier(n_estimators=175,criterion='entropy',max_depth=10)
```

In [40]:
```
cls.fit(x_train,y_train)
```

Out[40]:
```
▾ RandomForestClassifier
RandomForestClassifier(criterion='entropy', max_depth=10, n_estimators=175)
```

In [41]:
```
rfy_pred=cls.predict(x_test)
```

In [42]:
```
rfy_pred
```

Out[42]:
```
array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 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, 1, 0, 1, 0, 1, 1, 0, 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, 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, 0, 0, 1, 0,
       0, 1, 1, 0, 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, 1, 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, 0, 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, 1, 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 [43]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,rfy_pred)
```

Out[43]:
```
array([[160,  15],
       [ 41,  79]])
```

In [44]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,rfy_pred)
```

Out[44]:
```
0.8101694915254237
```

Logistic Regression

In [35]:
```
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()#creating object of LogisticRegression
classifier.fit(x_train,y_train)#training and fitting LR object using training data
```

Out[35]:
```
▾ LogisticRegression
LogisticRegression()
```

In [36]:
```
y_pred=classifier.predict(x_test)
y_pred
```

Out[36]:
```
array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1,
       0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0])
```

In [37]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[37]:
```
array([[150,  25],
       [ 33,  87]])
```

In [38]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[38]:
```
0.8033898305084746
```

In [ ]: