# Read the data

In [1]:
```
import numpy as nps
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
T = pd.read_csv('Titanic.csv')
T.head(6)
```

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 |
| **5** | 6 | 0 | 3 | Moran, Mr. James | male | NaN | 0 | 0 | 330877 | 8.4583 | NaN | Q |

# Clean the data

In [3]:
```
T.isnull().sum()
```

Out[3]:
```
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 [4]:
```
T.columns
```

Out[4]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [5]:
```
T['Age'].fillna(T['Age'].mean())        # mean of the null values in the data.
```

Out[5]:
```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         ...
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

In [6]:
```
T = T.drop(['Name', 'PassengerId', 'Cabin', 'Ticket', 'Fare'],axis=1)    # deleting the unwanted columns
T = T.dropna()      # filling nearest values in the null values.
T.head(6)
```

Out[6]:

| | Survived | Pclass | Sex | Age | SibSp | Parch | Embarked |
|---|---|---|---|---|---|---|---|
| **0** | 0 | 3 | male | 22.0 | 1 | 0 | S |
| **1** | 1 | 1 | female | 38.0 | 1 | 0 | C |
| **2** | 1 | 3 | female | 26.0 | 0 | 0 | S |
| **3** | 1 | 1 | female | 35.0 | 1 | 0 | S |
| **4** | 0 | 3 | male | 35.0 | 0 | 0 | S |
| **6** | 0 | 1 | male | 54.0 | 0 | 0 | S |

## Feature Engineering

In [7]:
```
from sklearn.preprocessing import LabelEncoder
gen = LabelEncoder()
emb = LabelEncoder()
```

In [8]:
```
T['Sex'] = gen.fit_transform(T['Sex'])              # changing sex notation into no.s (1 for male, 0 for female)
T['Embarked'] = emb.fit_transform(T['Embarked'])     # embarked is changed to 0,1,2
T.head(6)
```

Out[8]:

| | Survived | Pclass | Sex | Age | SibSp | Parch | Embarked |
|---|---|---|---|---|---|---|---|
| **0** | 0 | 3 | 1 | 22.0 | 1 | 0 | 2 |
| **1** | 1 | 1 | 0 | 38.0 | 1 | 0 | 0 |
| **2** | 1 | 3 | 0 | 26.0 | 0 | 0 | 2 |
| **3** | 1 | 1 | 0 | 35.0 | 1 | 0 | 2 |
| **4** | 0 | 3 | 1 | 35.0 | 0 | 0 | 2 |
| **6** | 0 | 1 | 1 | 54.0 | 0 | 0 | 2 |

## Prepare the data

In [9]:
```
x = T.drop('Survived', axis=1)    # we want the no. of people who are survived in the titanic ship after the cra
y = T['Survived']
```

In [10]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
```

In [11]:
```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[11]:
```
((569, 6), (143, 6), (569,), (143,))
```

## Building model

In [15]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
```

In [16]:
```
lr.fit(x_train,y_train)
```

Out[16]:
```
LogisticRegression
LogisticRegression()
```

## Testing

In [18]:
```
y_pred = lr.predict(x_test)
y_pred
```

Out[18]:
```
array([1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1,
       1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1], dtype=int64)
```

In [20]:
```
y_test
```

Out[20]:
```
708    1
149    0
755    1
528    0
278    0
      ..
805    0
209    1
328    1
700    1
269    1
Name: Survived, Length: 143, dtype: int64
```

In [21]:
```
from sklearn.metrics import accuracy_score
print("Accuracy: ", accuracy_score(y_pred, y_test))
```

```
Accuracy:  0.7692307692307693
```

In [23]:
```
age = int(input("Enter age: "))
pclass = int(input("Enter pclas: "))
gender = int(input("enter gender (0-female, 1-male): "))
sibsp = int(input("enter sibsp: "))
parch = int(input("enter parched: "))
emba = int(input("enter embarked : (0-c, 1-q, 2-s)"))
data = [[pclass, age, gender, sibsp, parch, emba]]
pred = lr.predict(data)
print(pred)
```

```
Enter age: 25
Enter pclas: 4
enter gender (0-female, 1-male): 1
enter sibsp: 4
enter parched: 3
enter embarked : (0-c, 1-q, 2-s)2
[0]


D:\Python\Anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, bu
t LogisticRegression was fitted with feature names
  warnings.warn(
```

In [24]:
```
import joblib
joblib.dump(lr, filename="titan.pkl")
```

Out[24]:
```
['titan.pkl']
```

In [ ]:
```

```