In [25]:

```
import pandas as pd
```

In [26]:

```
df = pd.read_csv('exp4.csv')
df.head()
```

Out[26]:

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

In [27]:

```
df.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Cabin', 'Embarked'], axis='col
df.head()
```

Out[27]:

	Survived	Pclass	Sex	Age	Fare
0	0	3	male	22.0	7.2500
1	1	1	female	38.0	71.2833
2	1	3	female	26.0	7.9250
3	1	1	female	35.0	53.1000
4	0	3	male	35.0	8.0500

In [28]:

```
inputs = df.drop('Survived', axis='columns')
inputs.head()
```

Out[28]:

	Pclass	Sex	Age	Fare
0	3	male	22.0	7.2500
1	1	female	38.0	71.2833
2	3	female	26.0	7.9250
3	1	female	35.0	53.1000
4	3	male	35.0	8.0500

In [29]:

```
inputs.Sex = inputs.Sex.map({'male':1, 'female':2})
inputs
```

Out[29]:

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

891 rows × 4 columns

In [30]:

```
inputs.Age = inputs.Age.fillna(inputs.Age.mean())
inputs
```

Out[30]:

	Pclass	Sex	Age	Fare
0	3	1	22.000000	7.2500
1	1	2	38.000000	71.2833
2	3	2	26.000000	7.9250
3	1	2	35.000000	53.1000
4	3	1	35.000000	8.0500
...	...	...	...	...
886	2	1	27.000000	13.0000
887	1	2	19.000000	30.0000
888	3	2	29.699118	23.4500
889	1	1	26.000000	30.0000
890	3	1	32.000000	7.7500

891 rows × 4 columns

In [31]:

```
target = df.Survived
target.head()
```

Out[31]:

```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [32]:

```
from sklearn.model_selection import train_test_split
```

In [33]:

```
x_train, x_test, y_train, y_test = train_test_split(inputs, target, test_size=0.3)
print(x_train)
```

```
     Pclass  Sex        Age      Fare
269       1    2  35.000000  135.6333
875       3    2  15.000000    7.2250
472       2    2  33.000000   27.7500
625       1    1  61.000000   32.3208
422       3    1  29.000000    7.8750
..      ...  ...        ...       ...
431       3    2  29.699118   16.1000
816       3    2  23.000000    7.9250
433       3    1  17.000000    7.1250
121       3    1  29.699118    8.0500
677       3    2  18.000000    9.8417

[623 rows x 4 columns]
```

In [34]:

```
from sklearn.linear_model import LogisticRegression
```

In [35]:

```
model = LogisticRegression()

model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

In [36]:

```
from sklearn.metrics import accuracy_score
```

In [37]:

```
accu = accuracy_score(y_test, y_pred)
print('Accuracy = ', accu)
```

```
Accuracy =  0.7574626865671642
```

In [38]:

```
print(x_test)
print(y_test)
```

```
     Pclass  Sex        Age     Fare
596       2    2  29.699118  33.0000
447       1    1  34.000000  26.5500
512       1    1  36.000000  26.2875
876       3    1  20.000000   9.8458
477       3    1  29.000000   7.0458
..      ...  ...        ...      ...
448       3    2   5.000000  19.2583
835       1    2  39.000000  83.1583
279       3    2  35.000000  20.2500
384       3    1  29.699118   7.8958
461       3    1  34.000000   8.0500

[268 rows x 4 columns]
596    1
447    1
512    1
876    0
477    0
      ..
448    1
835    1
279    1
384    0
461    0
Name: Survived, Length: 268, dtype: int64
```

In [39]:

```
print(model.predict([[3, 1, 29.699118, 21.6792]]))
```

```
[0]

C:\Users\manth\AppData\Local\Programs\Python\Python310\lib\site-packages\skl
earn\base.py:450: UserWarning: X does not have valid feature names, but Logi
sticRegression was fitted with feature names
  warnings.warn(
```