In [2]:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

In [3]:

```
df=pd.read_csv(r"C:\Users\raja\Downloads\gender_submission.csv")
df
```

Out[3]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1
...	...	...
413	1305	0
414	1306	1
415	1307	0
416	1308	0
417	1309	0

418 rows × 2 columns

In [4]:

```
pd.set_option('display.max_rows',10000000000)
pd.set_option('display.max_columns',10000000000)
pd.set_option('display.width',95)
```

In [5]:

```
print('This DataFrame has %d Rows and %d Columns'%(df.shape))
```

```
This DataFrame has 418 Rows and 2 Columns
```

In [6]:

```
df.head()
```

Out[6]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

In [16]:

```
features_matrix=df.iloc[:,0:2]
```

In [17]:

```
target_vector=df.iloc[:,-1]
```

In [18]:

```
print('The Features Matrix has %d Rows and %d Columns(s)'%(features_matrix.shape))
print('The Target Matrix has %d Rows and %d Columns(s)'%(np.array(target_vector).reshape
```

```
The Features Matrix has 418 Rows and 2 Columns(s)
The Target Matrix has 418 Rows and 1 Columns(s)
```

In [19]:

```
features_matrix_standardized=StandardScaler().fit_transform(features_matrix)
```

In [20]:

```
algorithm=LogisticRegression(max_iter=10000)
```

In [21]:

```
Logistic_regression_model=algorithm.fit(features_matrix_standardized,target_vector)
```

In [25]:

```
observation=[[1,0]]
```

In [26]:

```
print('the algorithm was trained to predict one of the two classes:%s'%(algorithm.classe
```

```
the algorithm was trained to predict one of the two classes:[0 1]
```

In [27]:

```
print("""the model says the probability of the observation we passed belonging to class[
print()
print("""the model says the probability of the observation we passed belonging to class[
```

```
the model says the probability of the observation we passed belonging to c
lass['0'] is 0.8238872695984016

the model says the probability of the observation we passed belonging to c
lass['1'] is 0.17611273040159833
```

In [28]:

```
predictions=Logistic_regression_model.predict(observation)
print('The Model Predicted the observation to Belong to class %s'%(predictions))
```

```
The Model Predicted the observation to Belong to class [0]
```

In [ ]: