In [63]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

In [64]:
```
df=pd.read_csv(r"C:\Users\chinta pavani\Downloads\ionosphere.csv")
df
```

Out[64]:

	atr1	atr2	atr3	atr4	atr5	atr6	atr7	atr8	atr9	atr10
0	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.00000	0.03760
1	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549
2	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198
3	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000
4	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399
5	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637
6	1	0	0.97588	-0.10602	0.94601	-0.20800	0.92806	-0.28350	0.85996	-0.27342
7	0	0	0.00000	0.00000	0.00000	0.00000	1.00000	-1.00000	0.00000	0.00000
8	1	0	0.96355	-0.07198	1.00000	-0.14333	1.00000	-0.21313	1.00000	-0.36174
9	1	0	-0.01864	-0.08459	0.00000	0.00000	0.00000	0.00000	0.11470	-0.26810
10	1	0	1.00000	0.06655	1.00000	-0.18388	1.00000	-0.27320	1.00000	-0.43107

In [65]:
```
pd.set_option('display.max_rows',10000000000)
pd.set_option('display.max_columns',10000000000)
pd.set_option('display.width',95)
```

In [66]:
```
print('This Dataframe has % d Rows and % d Columns'%(df.shape))
```

```
This Dataframe has  351 Rows and  35 Columns
```

In [67]:
```
df.head()
```

Out[67]:

	atr1	atr2	atr3	atr4	atr5	atr6	atr7	atr8	atr9	atr10	
0	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.00000	0.03760	0.8
1	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	0.5
2	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	0.7
3	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	0.0
4	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	0.5

In [68]:
```
features_matrix=df.iloc[:,0:34]
```

In [69]:
```
target_vector=df.iloc[:,-1]
```

In [70]:
```
print('The features Matrix Has % d Rows and % d Columns'%(features_matrix.
print('The Target Matrix Has % d Rows and % d Columns)'%(np.array)(target_
```

```
The features Matrix Has  351 Rows and  34 Columns
The Target Matrix Has  351 Rows and  1 Columns)
```

In [75]:
```
features_matrix_standardized=StandardScaler().fit_transform(features_matri
```

In [76]:
```
algorithm=LogisticRegression(penalty='l2',dual=False,tol=1e-4,C=1.0,fit_in
```

In [77]:
```
Logistic_Regression_Model=algorithm.fit(features_matrix_standardized,targe
```

In [80]:
```
observation=[[1,0,0.99539,-0.05889,0.854299999999999,0.02386,0.83399979999
```

In [81]:
```
predictions=Logistic_Regression_Model.predict(observation)
print('The Model Predicted The Obsevation To belong to class % s'%(predict
```

```
The Model Predicted The Obsevation To belong to class ['g']
```

In [82]:
```
print('The Algorithm Was Trained To Predict One Of The Two Classes:%s'%(al
```

```
The Algorithm Was Trained To Predict One Of The Two Classes:['b' 'g']
```

In [83]:
```
print("""The Model Says The Probability Of The Observation We Passed Belon
print()
print("""The Model Says The Probability Of The Observation We Passed Belon
```

```
The Model Says The Probability Of The Observation We Passed Belonging To
Class['b']Is 0.007753470575315724

The Model Says The Probability Of The Observation We Passed Belonging To
Class['b']Is 0.9922465294246843
```

In [ ]: