In [32]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

In [33]:
```
df=pd.read_csv(r"C:\Users\manasa\Downloads\ionosphere.csv")
df
```

Out[33]:

	atr1	atr2	atr3	atr4	atr5	atr6	atr7	atr8	atr9	atr10	atr11	atr12	atr13	atr14	atr15	atr16	at
0	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.00000	0.03760	0.85243	-0.17755	0.59755	-0.44945	0.60536	-0.38223	0.843
1	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	0.50874	-0.67743	0.34432	-0.69707	-0.51685	-0.97515	0.054
2	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	0.73082	0.05346	0.85443	0.00827	0.54591	0.00299	0.837
3	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	-1.00000	0.14516	0.540
4	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	0.52798	-0.20275	0.56409	-0.00712	0.34395	-0.27457	0.529
5	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637	0.03786	-0.06302	0.00000	0.00000	-0.04572	-0.15540	-0.003
6	1	0	0.97588	-0.10602	0.94601	-0.20800	0.92806	-0.28350	0.85996	-0.27342	0.79766	-0.47929	0.78225	-0.50764	0.74628	-0.61436	0.579
7	0	0	0.00000	0.00000	0.00000	0.00000	1.00000	-1.00000	0.00000	0.00000	-1.00000	-1.00000	0.00000	0.00000	0.00000	0.00000	1.000
8	1	0	0.96355	-0.07198	1.00000	-0.14333	1.00000	-0.21313	1.00000	-0.36174	0.92570	-0.43569	0.94510	-0.40668	0.90392	-0.46381	0.983
9	1	0	-0.01864	-0.08459	0.00000	0.00000	0.00000	0.00000	0.11470	-0.26810	-0.45663	-0.38172	0.00000	0.00000	-0.33656	0.38602	-0.371

In [34]:
```
pd.set_option('display.max_rows',10000000000)
pd.set_option('display.max_columns',10000000000)
pd.set_option('display.width',95)
```

In [35]:
```
print('This Dataframe has % d Rows and % d columns' %(df.shape))
```

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

In [36]:
```
df.head()
```

Out[36]:

	atr1	atr2	atr3	atr4	atr5	atr6	atr7	atr8	atr9	atr10	atr11	atr12	atr13	atr14	atr15	atr16	atr17	at
0	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.00000	0.03760	0.85243	-0.17755	0.59755	-0.44945	0.60536	-0.38223	0.84356	-0.385
1	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	0.50874	-0.67743	0.34432	-0.69707	-0.51685	-0.97515	0.05499	-0.622
2	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	0.73082	0.05346	0.85443	0.00827	0.54591	0.00299	0.83775	-0.136
3	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	-1.00000	0.14516	0.54094	-0.393
4	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	0.52798	-0.20275	0.56409	-0.00712	0.34395	-0.27457	0.52940	-0.217

In [37]:
```
features_matrix=df.iloc[:,0:34]
```

In [38]:
```
target_vector=df.iloc[:,-1]
```

In [39]:
```
print('The features Matrix has % d Rows and % d columns'%(features_matrix.shape))
print('The Target Matrix has % d Rows and % d columns)'%(np.array)(target_vector).reshape(-1,1).shape)
```

```
The features Matrix has  351 Rows and  34 columns
The Target Matrix has  351 Rows and  1 columns)
```

In [40]:
```
features_matrix_standardized=StandardScaler().fit_transform(features_matrix)
```

In [41]:
```
_weight=None,random_state=None,solver='lbfgs',max_iter=100,multi_class='auto',verbose=0,warm_start=False,n_jobs=None,l1_ratio=None
```

In [42]:
```
Logistic_Regression_Model=algorithm.fit(features_matrix_standardized,target_vector)
```

In [46]:
```
.29674,0.36946,-0.47357,0.56811,-0.51171,0.4107800000000001,-0.4616800000000003,0.21266,-0.3409,0.42267,-0.54487,0.18641,-0.453]]
```

In [47]:
```
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 ['g']
```

In [48]:
```
print('The Algorithm was Trained To predict One Of The Two Classes:%s'%(algorithm.classes_))
```

```
The Algorithm was Trained To predict One Of The Two Classes:['b' 'g']
```

In [55]:
```
l Says The Probability Of The Observation We Passed Belonging To Class['b']Is %s"""%(algorithm.predict_proba(observation)[0][0]))

l Says The Probability Of The Observation We Passed Belonging To Class['g']Is %s"""%(algorithm.predict_proba(observation)[0][1]))
```

```
The Model Says The Probability Of The Observation We Passed Belonging To Class['b']Is 0.007759545690606995

The Model Says The Probability Of The Observation We Passed Belonging To Class['g']Is 0.992240454309393
```

In [ ]: