In [11]:

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

In [12]:

```
df=pd.read_csv(r"C:\Users\RAMADEVI SURIPAKA\Downloads\Inosphere.csv")
df
```

Out[12]:

	column_a	column_b	column_c	column_d	column_e	column_f	column_g	column_h
0	True	False	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708
1	True	False	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597
2	True	False	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062
3	True	False	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000
4	True	False	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255
...	...	...	...	...	...	...	...	...
346	True	False	0.83508	0.08298	0.73739	-0.14706	0.84349	-0.05567
347	True	False	0.95113	0.00419	0.95183	-0.02723	0.93438	-0.01920
348	True	False	0.94701	-0.00034	0.93207	-0.03227	0.95177	-0.03431
349	True	False	0.90608	-0.01657	0.98122	-0.01989	0.95691	-0.03646
350	True	False	0.84710	0.13533	0.73638	-0.06151	0.87873	0.08260

351 rows × 35 columns

In [13]:

```
pd.set_option('display.max_row',10000000000)
pd.set_option('display.max_columns',10000000000)
pd.set_option('display.width',95)
print('This dataFrame has %d rows and %d columns'%(df.shape))
```

```
This dataFrame has 351 rows and 35 columns
```

In [14]:

```
df.head()
```

Out[14]:

	column_a	column_b	column_c	column_d	column_e	column_f	column_g	column_h	c
0	True	False	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	
1	True	False	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	
2	True	False	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	
3	True	False	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	
4	True	False	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	

In [15]:

```
features_matrix=df.iloc[:,0:34]
target_vector=df.iloc[:,-1]
print('The Features Matrix Has %d Rows And %d Columns(s)'%(features_matrix.shape))
print('The Features Matrix Has %d Rows And %d Columns(s)'%(np.array(target_vector).resha
```

```
The Features Matrix Has 351 Rows And 34 Columns(s)
The Features Matrix Has 351 Rows And 1 Columns(s)
```

In [16]:

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

In [17]:

```
algorithm=LogisticRegression(penalty='l2',dual=False,tol=1e-4,C=1.0,fit_intercept=True,i
```

In [18]:

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

In [20]:

```
observation=[[1,0,0.99539,-0.05889,0.854299999999999,0.02306,0.8339799999999999,-0.37708
```

In [21]:

```
predictions=Logistic_Regression_Model.predict(observation)
```

In [22]:

```
print('The Model Predicted The Obsevation To Belong To Class %s'%(predictions))
print('The Algorithm Was Trained To Predict One Of The Two Classes:%s'%(algorithm.classe
```

```
The Model Predicted The Obsevation To Belong To Class ['g']
The Algorithm Was Trained To Predict One Of The Two Classes:['b' 'g']
```

In [23]:

```
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
Class['b'] Is 0.007759545690606995

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

In [ ]: