

In [28]:

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

In [30]:

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

| | | | | | | | | | | | | | | | | |
|-----|---|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 195 | 1 | 0 | 1.00000 | -1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | -0.50000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 |
| 196 | 1 | 0 | 1.00000 | 0.23058 | 1.00000 | -0.78509 | 1.00000 | -0.10401 | 1.00000 | 0.15414 | 1.00000 | 0.27820 | 0.98120 | -0.06861 | 1.00000 | 0.06111 |
| 197 | 1 | 0 | 0.36876 | -1.00000 | -1.00000 | -1.00000 | -0.07661 | 1.00000 | 1.00000 | 0.95041 | 0.74597 | -0.38710 | -1.00000 | -0.79313 | -0.09677 | 1.00000 |
| 198 | 1 | 0 | 0.79847 | 0.38265 | 0.80804 | -0.16964 | 1.00000 | -0.07653 | 0.98151 | -0.07398 | 0.70217 | 0.20663 | 0.99745 | 0.02105 | 0.98214 | 0.02105 |
| 199 | 0 | 0 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 |
| 200 | 1 | 0 | 1.00000 | -0.28428 | 1.00000 | -0.25346 | 0.94623 | -0.35094 | 1.00000 | -0.30566 | 0.92736 | -0.49057 | 0.90818 | -0.44119 | 0.75723 | -0.58119 |
| 201 | 1 | 0 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | -0.85000 | -1.00000 | 0.00000 | 0.00000 | 1.00000 | -1.00000 | 0.00000 | 0.00000 |
| 202 | 1 | 0 | 1.00000 | 0.09091 | 0.95455 | -0.09091 | 0.77273 | 0.00000 | 1.00000 | 0.00000 | 0.95455 | 0.00000 | 1.00000 | 0.04545 | 0.90909 | -0.04545 |
| 203 | 0 | 0 | 0.00000 | 0.00000 | -1.00000 | 1.00000 | 1.00000 | 1.00000 | -1.00000 | -1.00000 | 0.00000 | 0.00000 | -1.00000 | -1.00000 | -1.00000 | -0.31818 |
| 204 | 1 | 0 | 0.91176 | -0.08824 | 0.97059 | 0.17647 | 0.82353 | 0.08824 | 0.91176 | -0.02941 | 0.97059 | -0.17647 | 0.97059 | 0.14706 | 0.94118 | 0.02941 |
| 205 | 1 | 0 | -1.00000 | 1.00000 | -1.00000 | 0.15244 | 0.28354 | 1.00000 | -1.00000 | 1.00000 | -1.00000 | -1.00000 | 1.00000 | 1.00000 | -1.00000 | -0.23529 |
| 206 | 1 | 0 | 0.47368 | -0.10526 | 0.83781 | 0.01756 | 0.83155 | 0.02615 | 0.68421 | -0.05263 | 0.68421 | 0.00000 | 0.79856 | 0.05028 | 0.78315 | 0.05028 |
| 207 | 1 | 0 | 0.00641 | 0.50000 | 0.00000 | 0.00000 | 0.01823 | 1.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |

In [31]:

```
pd.set_option('display.max_rows',1000000000)
pd.set_option('display.max_columns',1000000000)
pd.set_option('display.width',95)
```

In [32]:

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

This DataFrame has 350 Rows And 35 Columns

In [33]:

```
df.head()
```

Out[33]:

| | | | | | | | | | | | | | | | |
|---|---|---|---------|----------|----------|----------|----------|----------|---------|----------|-----------|----------|---------|----------|--------|
| | 1 | 0 | 0.99539 | -0.05889 | 0.85243 | 0.02306 | 0.83398 | -0.37708 | 1.1 | 0.03760 | 0.85243.1 | -0.17755 | 0.59755 | -0.44945 | 0.605 |
| 0 | 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 | -0.516 |
| 1 | 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 | 0.545 |
| 2 | 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 | -1.000 |
| 3 | 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 | 0.343 |
| 4 | 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 | -0.045 |

In [35]:

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

In [36]:

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

In [37]:

```
print('The Features Matrix Has %d Rows And %d Columns(s)'%(features_matrix.shape))
```

The Features Matrix Has 350 Rows And 34 Columns(s)

In [42]:

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

In [43]:

```
algorithm=LogisticRegression(max_iter=1000)
```

In [44]:

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

In [45]:

```
observation=[[1,0,0.99539,-0.05889,0.8524299999999999,0.02306,0.8339799999999999,
-0.37708,1.0,0.0376,0.8524299999999999,-0.17755,0.59755,-0.44945,0.60536,
-0.38223,0.8435600000000001,-0.38542,0.58212,-0.32192,0.56971,-0.29674,0.36946,
-0.47357,0.56811,-0.51171,0.4107800000000003,-0.4616800000000003,0.21256,-0.3409,0.112267,-0.54487,0.18641,-0.453]
```

In [46]:

```
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 [47]:

```
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 [48]:

```
el says the probability of the observation we passed belonging to class['b']is %s""%(algorithm.predict_proba(observation)[0][0])
el 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 0.009471289688510742
The model says the probability of the observation we passed belonging to class['g']is 0.9905287103114893

In []: