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\manis\Downloads\ionosphere_data.csv")
df
```

Out[3]:

	column_a	column_b	column_c	column_d	column_e	column_f	column_g	colum
0	True	False	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37
1	True	False	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93
2	True	False	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12
3	True	False	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00
4	True	False	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23
...	...	...	...	...	...	...	...	
346	True	False	0.83508	0.08298	0.73739	-0.14706	0.84349	-0.05
347	True	False	0.95113	0.00419	0.95183	-0.02723	0.93438	-0.01
348	True	False	0.94701	-0.00034	0.93207	-0.03227	0.95177	-0.03
349	True	False	0.90608	-0.01657	0.98122	-0.01989	0.95691	-0.03
350	True	False	0.84710	0.13533	0.73638	-0.06151	0.87873	0.08

351 rows × 35 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 351 Rows and 35 Columns
```

In [6]:
```
df.head()
```

Out[6]:

	column_a	column_b	column_c	column_d	column_e	column_f	column_g	column_
0	True	False	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.3770
1	True	False	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.9359
2	True	False	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.1206
3	True	False	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.0000
4	True	False	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.2325

In [7]:
```
features_matrix=df.iloc[:, 0:34]
```

In [8]:
```
target_vector=df.iloc[:, -1]
```

In [9]:
```
print('The Featues Matrix Has %d Rows adn %d Columns(s)'%(features_matrix.shape)
print('The Target Matrix Has %d Rows adn %d Columns(s)'%(np.array(target_vector)
```

```
The Featues Matrix Has 351 Rows adn 34 Columns(s)
The Target Matrix Has 351 Rows adn 1 Columns(s)
```

In [10]:
```
features_matrix_standardized=StandardScaler().fit_transform(features_matrix)
```

In [11]:
```
algorithm=LogisticRegression(penalty='l2',dual=False,tol=1e-4,C=1.0,fit_intercep
```

In [12]:
```
Logistic_Regression_Model=algorithm.fit(features_matrix_standardized,target_vect
```

In [13]:
```
observation=[[1,0,0.99539,-0.05889,0.8524299999999999,0.02306,0.8339799999999999
              -0.17755,0.59755,-0.44945,0.60536,-0.38223,0.8435600000000001,-0.3
              0.36946,-0.47357,0.56811,-0.51171,0.41078000000000003,-0.46168000
              -0.54487,0.18641,-0.453]]
```

In [15]:
```
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 [16]:
```
print('The Algorithm Was Trained To Predict One Of The Two Classes:%s'%(algorith
```

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

In [17]:
```
print("""The Model Says The Probability Of The Observation We Passed Belonging T
print()
print("""The Model Says The Probability Of The Observation We Passed Belonging T
```

```
The Model Says The Probability Of The Observation We Passed Belonging To Class
['b'] Is 0.0077739316001405045

The Model Says The Probability Of The Observation We Passed Belonging To Class
['g'] Is 0.9922260683998595
```

In [ ]:
```

```