```python
In [2]:   1  import pandas as pd
          2  import numpy as np
          3  from sklearn.linear_model import LogisticRegression
          4  from sklearn.preprocessing import StandardScaler
```

```python
In [3]:   1  df=pd.read_csv(r"C:\Users\HP\OneDrive\Documents\Ionospear.csv")
          2  df
```

Out[3]:

	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.1	0.0376	...	-0.51171	0.41078	-0.46168	0.21266	-0.3409	0.42267	-0.54487	0.18641	-0.453	g
0	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	...	-0.26569	-0.20468	-0.18401	-0.19040	-0.11593	-0.16626	-0.06288	-0.13738	-0.02447	b
1	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	...	-0.40220	0.58984	-0.22145	0.43100	-0.17365	0.60436	-0.24180	0.56045	-0.38238	g
2	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	...	0.90695	0.51613	1.00000	1.00000	-0.20099	0.25682	1.00000	-0.32382	1.00000	b
3	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	...	-0.65158	0.13290	-0.53206	0.02431	-0.62197	-0.05707	-0.59573	-0.04608	-0.65697	g
4	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637	...	-0.01535	-0.03240	0.09223	-0.07859	0.00732	0.00000	0.00000	-0.00039	0.12011	b
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
345	1	0	0.83508	0.08298	0.73739	-0.14706	0.84349	-0.05567	0.90441	-0.04622	...	-0.04202	0.83479	0.00123	1.00000	0.12815	0.86660	-0.10714	0.90546	-0.04307	g
346	1	0	0.95113	0.00419	0.95183	-0.02723	0.93438	-0.01920	0.94590	0.01606	...	0.01361	0.93522	0.04925	0.93159	0.08168	0.94066	-0.00035	0.91483	0.04712	g
347	1	0	0.94701	-0.00034	0.93207	-0.03227	0.95177	-0.03431	0.95584	0.02446	...	0.03193	0.92489	0.02542	0.92120	0.02242	0.92459	0.00442	0.92697	-0.00577	g
348	1	0	0.90608	-0.01657	0.98122	-0.01989	0.95691	-0.03646	0.85746	0.00110	...	-0.02099	0.89147	-0.07760	0.82983	-0.17238	0.96022	-0.03757	0.87403	-0.16243	g
349	1	0	0.84710	0.13533	0.73638	-0.06151	0.87873	0.08260	0.88928	-0.09139	...	-0.15114	0.81147	-0.04822	0.78207	-0.00703	0.75747	-0.06678	0.85764	-0.06151	g

350 rows × 35 columns

```python
In [4]:   1  pd.set_option('display.max_rows',10000000000)
          2  pd.set_option('display.max_columns',10000000000)
          3  pd.set_option('display.width',95)
```

```python
In [5]:   1  print('The DataFrame has %d Rows and %d columns'%(df.shape))
```

The DataFrame has 350 Rows and 35 columns

```python
In [6]:   1  df.head()
```

Out[6]:

	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.1	0.0376	0.85243.1	-0.17755	0.59755	-0.44945	0.60536	-0.38223	0.84356	-0.38542	0.58212	-0.32192	0.56971	-0.29674	0.36946	-0.47357	0.56811	-0.51
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	-0.51685	-0.97515	0.05499	-0.62237	0.33109	-1.00000	-0.13151	-0.45300	-0.18056	-0.35734	-0.20332	-0.26
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	0.54591	0.00299	0.83775	-0.13644	0.75535	-0.08540	0.70887	-0.27502	0.43385	-0.12062	0.57528	-0.40
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	-1.00000	0.14516	0.54094	-0.39330	-1.00000	-0.54467	-0.69975	1.00000	0.00000	0.00000	1.00000	0.90
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	0.34395	-0.27457	0.52940	-0.21780	0.45107	-0.17813	0.05982	-0.35575	0.02309	-0.52879	0.03286	-0.65
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	-0.04572	-0.15540	-0.00343	-0.10196	-0.11575	-0.05414	0.01838	0.03669	0.01519	0.00888	0.03513	-0.01

```python
In [27]:  1  feature_matrix=df.iloc[:,0:34]
          2  target_vector=df.iloc[:,-1]
```

```python
In [28]:  1  print('The Features matrix has %d Rows and %d column(s)'%(feature_matrix.shape))
          2  print('The target matrix has %d Rows and %d column(s)'%(np.array(target_vector).reshape(-1,1).shape))
```

The Features matrix has 350 Rows and 34 column(s)
The target matrix has 350 Rows and 1 column(s)

```python
In [35]:  1  feature_matrix_standardized = StandardScaler().fit_transform(feature_matrix)
```

```python
In [67]:  1  Algorithm = LogisticRegression(penalty='l2',dual=False,tol=1e-4,C=1.0,fit_intercept=True,intercept_scaling=1,class_weight=None,random_state=None,solver='lbfgs',max_iter=100,multi_class
          2
```

```python
In [68]:  1  Logistic_Regression_Model = Algorithm.fit(feature_matrix_standardized,target_vector)
```

```python
In [69]:  1  observation = [[1,0,0.99539,-0.05889,0.8524299999999999,0.02306,
          2                  0.8339799999999999,-0.37708,1.0,0.0376,0.8524299999999999,
          3                  -0.17755,0.59755,-0.44945,0.60536,-0.38223,0.8435600000000001,
          4                  -0.38542,0.58219,-0.32192,0.56971,-0.29674,0.36946,-0.47357,
          5                  0.56811,-0.51171,0.4107800000000003,-0.4616800000000003,0.21266,
          6                  -0.3409,0.42267,-0.54487,0.18641,-0.453]]
```

```python
In [70]:  1  predictions = Logistic_Regression_Model.predict(observation)
          2  print('The Model predicted The observation to belong to class %s'%(predictions))
```

The Model predicted The observation to belong to class ['g']

```python
In [71]:  1  print('Algorithm was Trained To predict one of the two classes : %s'%(Algorithm.classes_))
```

Algorithm was Trained To predict one of the two classes : ['b' 'g']

```python
In [72]:  1  print("""The Model ssays the probability of the observation we passed Belonging To class['b'] Is %s"""%(Algorithm.predict_proba(observation)[0][0]))
          2  print()
          3  print("""The Model ssays the probability of the observation we passed Belonging To class['g'] Is %s"""%(Algorithm.predict_proba(observation)[0][1]))
```

The Model ssays the probability of the observation we passed Belonging To class['b'] Is 0.008044378633976335

The Model ssays the probability of the observation we passed Belonging To class['g'] Is 0.9919556213660237

```python
In [ ]:   1
```