

inosphere

In [1]:

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

In [2]:

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

Out[2]:

	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 [3]:

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

In [4]:

```
print('This DataFrame has %d Rows and %d columns'%(df.shape))
```

This DataFrame has 351 Rows and 35 columns

In [5]:

```
df.head()
```

Out[5]:

	column_a	column_b	column_c	column_d	column_e	column_f	column_g	column_h	column_i
0	True	False	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	0.00000
1	True	False	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	0.00000
2	True	False	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.00000
3	True	False	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000
4	True	False	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.00000

In [6]:

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

In [7]:

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

In [8]:

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

The Features Matrix Has 351 Rows And 34 columns(s)

In [9]:

```
print('The Target Matrix Has %d Rows And %d Columns(s)'%(np.array(target_vector).reshape(1, -1).shape))
```

The Target Matrix Has 351 Rows And 1 Columns(s)

In [10]:

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

In [11]:

```
algorithm = LogisticRegression(penalty=None, dual=False, tol=1e-4, C=1.0, fit_intercept=True,
                               class_weight=None, random_state=None, solver='lbfgs', max_iter=10000,
                               multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
```

In [12]:

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

In []: