In [5]:

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

In [6]:

```
df=pd.read_csv(r"C:\Users\MSI\Downloads\archive.zip")
df
```

Out[6]:

	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.1	0.03760	...	-0.51171	0.4107
0	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	...	-0.26569	-0.2046
1	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	...	-0.40220	0.5898
2	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	...	0.90695	0.5161
3	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	...	-0.65158	0.1329
4	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637	...	-0.01535	-0.0324
...	...	...	...	...	...	...	...	...	...	...	...	...	.
345	1	0	0.83508	0.08298	0.73739	-0.14706	0.84349	-0.05567	0.90441	-0.04622	...	-0.04202	0.8347
346	1	0	0.95113	0.00419	0.95183	-0.02723	0.93438	-0.01920	0.94590	0.01606	...	0.01361	0.9352
347	1	0	0.94701	-0.00034	0.93207	-0.03227	0.95177	-0.03431	0.95584	0.02446	...	0.03193	0.9248
348	1	0	0.90608	-0.01657	0.98122	-0.01989	0.95691	-0.03646	0.85746	0.00110	...	-0.02099	0.8914
349	1	0	0.84710	0.13533	0.73638	-0.06151	0.87873	0.08260	0.88928	-0.09139	...	-0.15114	0.8114

350 rows × 35 columns

In [7]:

```
pd.set_option('display.max_rows',10000000000)
pd.set_option('display.max_columns',10000000000)
pd.set_option('display.Width',95)
```

In [8]:

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

```
This DataFrame has 350 Rows and 35 Columns
```

In [9]:

```
df.head()
```

Out[9]:

	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.1	0.03760	0.85243.1	-0.17755	0.5
0	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	0.50874	-0.67743	0.3
1	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	0.73082	0.05346	0.8
2	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	0.00000	0.00000	0.0
3	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	0.52798	-0.20275	0.5
4	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637	0.03786	-0.06302	0.0

In [10]:

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

In [11]:

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

In [12]:

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

```
The Features Matrix Has 350 Rows And 34 columns
The Features Matrix Has 350 Rows And 1 Columns
```

In [13]:

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

In [15]:

```
1,class_weight=None,random_state=None,solver='lbfgs',max_iter=1000,multi_class='auto',warm_start=Fals
```

In [16]:

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

In [17]:

```
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, -0.38
             0.56971, -0.29674, 0.36946, -0.47357, 0.56811, -0.51171, 0.4107800000000003, -0.4616800
```

In [18]:

```
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 [34]:

```
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 [36]:

```
 Says The Probability of The Observation We Passed Belonging To class['b'] is %s"""%(algorithm.predic

 Says The Probability of The Observation We Passed Belonging To class['g'] is %s"""%(algorithm.predic
```

```
The Model Says The Probability of The Observation We Passed Belonging To class['b'] is
2.0076200191287974e-05

The Model Says The Probability of The Observation We Passed Belonging To class['g'] is
0.9999799237998087
```

In [ ]: