In [17]:

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

In [18]:

```
df = pd.read_csv(r"C:\Users\smb06\Downloads\archive (1).zip")
df
```

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
5	1	0	0.97588	-0.10602	0.94601	-0.20800	0.92806	-0.28350	0.85996	-0.27342	0.79766	-0.47929	0.78225	-0.50764	0.74628	-0.61436
6	0	0	0.00000	0.00000	0.00000	0.00000	1.00000	-1.00000	0.00000	0.00000	-1.00000	-1.00000	0.00000	0.00000	0.00000	0.00000
7	1	0	0.96355	-0.07198	1.00000	-0.14333	1.00000	-0.21313	1.00000	-0.36174	0.92570	-0.43569	0.94510	-0.40668	0.90392	-0.46381
8	1	0	-0.01864	-0.08459	0.00000	0.00000	0.00000	0.00000	0.11470	-0.26810	-0.45663	-0.38172	0.00000	0.00000	-0.33656	0.38602
9	1	0	1.00000	0.06655	1.00000	-0.18388	1.00000	-0.27320	1.00000	-0.43107	1.00000	-0.41349	0.96232	-0.51874	0.90711	-0.59017
10	1	0	1.00000	-0.54210	1.00000	-1.00000	1.00000	-1.00000	1.00000	0.36217	1.00000	-0.41119	1.00000	1.00000	1.00000	-1.00000
11	1	0	1.00000	-0.16316	1.00000	-0.10169	0.99999	-0.15197	1.00000	-0.19277	0.94055	-0.35151	0.95735	-0.29785	0.93719	-0.34412
12	1	0	1.00000	-0.86701	1.00000	0.22280	0.85492	-0.39896	1.00000	-0.12090	1.00000	0.35147	1.00000	0.07772	1.00000	-0.14767
13	1	0	1.00000	0.07380	1.00000	0.03420	1.00000	-0.05563	1.00000	0.08764	1.00000	0.19651	1.00000	0.20328	1.00000	0.12785
14	1	0	0.50932	-0.93996	1.00000	0.26708	-0.03520	-1.00000	1.00000	-1.00000	0.43685	-1.00000	0.00000	0.00000	-1.00000	-0.34265
15	1	0	0.99645	0.06468	1.00000	-0.01236	0.97811	0.02498	0.96112	0.02312	0.99274	0.07808	0.89323	0.10346	0.94212	0.05269
16	0	0	0.00000	0.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000

In [20]:

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

In [21]:

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

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

In [22]:

```
df.head()
```

Out[22]:

	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	0.60536
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
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
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
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
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

In [23]:

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

In [24]:

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

In [25]:

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

```
The Features Matrix Has 350 Rows And 34 columns(s)
The Target Matrix Has 350 Rows And 1 Columns(s)
```

In [26]:

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

In [34]:

```
algorithm=LogisticRegression(penalty=None,dual=False,tol=1e-4,C=1.0,fit_intercept=True,intercept_scaling=1,class_weight=
                            random_state=None,solver='lbfgs',max_iter=1000,multi_class='auto',verbose=0,warm_start=False
                             n_jobs=None,l1_ratio=None)
```

In [39]:

```
logistic_Regression_Model
l=algorithm.fit(features_matrix_standardized,target_vector)
```

In [42]:

```
observation=[[1,0,0.99539,-0.5889,0.8524299999999999,0.02306,0.833979999999999,-0.37708,1.0,0.0376,0.8524-299999999999,-0
             0.59755,-0.44945,0.60536,-0.38223,0.8435600000000001,-0.38542,0.58212,-0.32192,0.56971,-0.29674,0.36946,-0
             0.56811,-0.51171,0.4107800000000003,-0.4616800000000003,0.21256,-0.3409,0.112267,-0.54487,0.18641,-0.453]
```

In [43]:

```
predictions=logistic_Regression_Model.predict(observation)
print('The Model predicted the observation to belog to class %s'%(predictions))
```

```
The Model predicted the observation to belog to class ['g']
```

In [44]:

```
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 [45]:

```
print("""The model says the probability of the obserbvation we passedbelonging to class['b']is %s"""
      %(algorithm.predict_proba(observation)[0][0]))
print()
print("""The model says the probability of the observation we passed belonging to class['g']is %s"""
     %(algorithm.predict_proba(observation)[observation[0][1]]))
```

```
The model says the probability of the obserbvation we passedbelonging to class['b']is 0.0

The model says the probability of the observation we passed belonging to class['g']is [0. 1.]
```

In [ ]: