In [1]:

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

In [4]:

```
df=pd.read_csv(r"C:\Users\user\Downloads\archive (3).zip")
df
```

Out[4]:

	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	feature9	featur
0	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.00000	0.037
1	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04
2	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01
3	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00
4	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16
...	...	...	...	...	...	...	...	...	...	...
346	1	0	0.83508	0.08298	0.73739	-0.14706	0.84349	-0.05567	0.90441	-0.04
347	1	0	0.95113	0.00419	0.95183	-0.02723	0.93438	-0.01920	0.94590	0.01
348	1	0	0.94701	-0.00034	0.93207	-0.03227	0.95177	-0.03431	0.95584	0.02
349	1	0	0.90608	-0.01657	0.98122	-0.01989	0.95691	-0.03646	0.85746	0.00
350	1	0	0.84710	0.13533	0.73638	-0.06151	0.87873	0.08260	0.88928	-0.09

351 rows × 35 columns

In [5]:

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

In [6]:

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

```
This DataFrame has 351 Rows and 35 Columns
```

In [7]:

```
df.head()
```

Out[7]:

	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature1
0	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.00000	0.0376
1	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.0454
2	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.0119
3	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.0000
4	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.1639

In [8]:
```
features_matrix=df.iloc[:,0:34]
```

In [9]:
```
target_vector=df.iloc[:,-1]
```

In [12]:
```
print('The features matrix Has %d Rows And %d Columns(s)'%(features_matrix.shap
print('The Target matrix has %d Rows And %d Columns(s)'%(np.array(target_vector
```

```
The features matrix Has 351 Rows And 34 Columns(s)
The Target matrix has 351 Rows And 1 Columns(s)
```

In [13]:
```
features_matrix_Standardized=StandardScaler().fit_transform(features_matrix)
```

In [14]:
```
algorithm=LogisticRegression(max_iter=1000)
```

In [23]:
```
Logistic_Regression_Model=algorithm.fit(features_matrix_Standardized,target_vec
```

In [ ]: