In [1]:
```
import pandas as pd
from sklearn.linear_model import LogisticRegression
```

In [8]:
```
from sklearn.preprocessing import StandardScaler
```

In [9]:
```
df=pd.read_csv("1_ionosphere.csv")
df
```

Out[9]:

| | 1 | 0 | 0.99539 | -0.05889 | 0.85243 | 0.02306 | 0.83398 | -0.37708 | 1.1 | 0.03760 | ... | -0.51171 | 0.41078 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 1.00000 | -0.18829 | 0.93035 | -0.36156 | -0.10868 | -0.93597 | 1.00000 | -0.04549 | ... | -0.26569 | -0.20468 |
| **1** | 1 | 0 | 1.00000 | -0.03365 | 1.00000 | 0.00485 | 1.00000 | -0.12062 | 0.88965 | 0.01198 | ... | -0.40220 | 0.58984 |
| **2** | 1 | 0 | 1.00000 | -0.45161 | 1.00000 | 1.00000 | 0.71216 | -1.00000 | 0.00000 | 0.00000 | ... | 0.90695 | 0.51613 |
| **3** | 1 | 0 | 1.00000 | -0.02401 | 0.94140 | 0.06531 | 0.92106 | -0.23255 | 0.77152 | -0.16399 | ... | -0.65158 | 0.13290 |
| **4** | 1 | 0 | 0.02337 | -0.00592 | -0.09924 | -0.11949 | -0.00763 | -0.11824 | 0.14706 | 0.06637 | ... | -0.01535 | -0.03240 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **345** | 1 | 0 | 0.83508 | 0.08298 | 0.73739 | -0.14706 | 0.84349 | -0.05567 | 0.90441 | -0.04622 | ... | -0.04202 | 0.83479 |
| **346** | 1 | 0 | 0.95113 | 0.00419 | 0.95183 | -0.02723 | 0.93438 | -0.01920 | 0.94590 | 0.01606 | ... | 0.01361 | 0.93522 |
| **347** | 1 | 0 | 0.94701 | -0.00034 | 0.93207 | -0.03227 | 0.95177 | -0.03431 | 0.95584 | 0.02446 | ... | 0.03193 | 0.92489 |
| **348** | 1 | 0 | 0.90608 | -0.01657 | 0.98122 | -0.01989 | 0.95691 | -0.03646 | 0.85746 | 0.00110 | ... | -0.02099 | 0.89147 |
| **349** | 1 | 0 | 0.84710 | 0.13533 | 0.73638 | -0.06151 | 0.87873 | 0.08260 | 0.88928 | -0.09139 | ... | -0.15114 | 0.81147 |

350 rows × 35 columns

In [12]:
```
x=df.iloc[:,:10]
y=df.iloc[:,-1]
f=StandardScaler().fit_transform(x)
lo=LogisticRegression()
lo.fit(f,y)
```

Out[12]:
```
LogisticRegression()
```

In [15]:
```
val=[[6,45,234,534,6,3456,345,4,53,45]]
lo.predict(val)
```

Out[15]:
```
array(['g'], dtype=object)
```

In [16]:
```
lo.score(f,y)
```

Out[16]:
```
0.8885714285714286
```

In [ ]:
```

```