In [25]: ▶|
```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

In [26]: ▶|
```python
df=pd.read_csv(r"C:\Users\MY HOME\Downloads\ionosphere.csv")
df
```

Out[26]:

	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	0.605
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	-0.516
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	0.545
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	-1.000
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	0.343
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	-0.045
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	0.746
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	0.000
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	0.903
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	-0.336
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	0.907

In [27]: ▶|
```python
pd.set_option('display.max_rows',10000000000)
pd.set_option('display.max_columns',10000000000)
pd.set_option('display.width',95)
```

```
In [28]: ▶  print('This DataFrame has %d rows and %d columns'%(df.shape))
```

This DataFrame has 350 rows and 35 columns

```
In [29]: ▶  features_matrix=df.iloc[:,0:34]
            target_vector=df.iloc[:,-1]
            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 [30]: ▶  features_matrix_Standardized=StandardScaler().fit_transform(features_matrix)
```

```
In [36]: ▶  algorithm=LogisticRegression(penalty=None,dual=False,tol=1e-4,C=1.0,fit_intercept=True,intercept_scaling=1,class
```

```
In [37]: ▶  Logistic_Regression_Model=algorithm.fit(features_matrix_Standardized,target_vector)
```

```
In [38]: ▶  observation=[[1,0,0.99539,-0.05889,0.854299999999999,0.02306,0.8339799999999999,-0.37708,1.0,0.0376,0.8524299999
                        -0.38223,0.8435600000000001,-0.36946,-0.4737,0.56811,-0.51171,0.410780000000003,-0.4616800000000
```

```
In [41]: ▶  predictions=Logistic_Regression_Model.predict(observation)
            print('The model predicted the observtaion to belong to class %s'%(predictions))
            print('The algorithm was trained to predict one of the two calsses : %s'%(algorithm.classes_))
```

The model predicted the observtaion to belong to class ['g']
The algorithm was trained to predict one of the two calsses : ['b' 'g']

In [43]: ▶|
```python
print("""The model says the prbability of the observation we passed belonging to calss ['b'] Is %s"""%(algorithm
print()
print("""The model says the prbability of the observation we passed belonging to calss ['g'] Is %s"""%(algorithm
```

The model says the prbability of the observation we passed belonging to calss ['b'] Is 5.923069895330535e-06

The model says the prbability of the observation we passed belonging to calss ['b'] Is 0.9999940769301047

In [ ]: ▶|