

Logistic Regression

```
In [74]: import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

```
In [75]: df=pd.read_csv(r"C:\Users\sneha\Downloads\ionosphere.csv")
df
```

Out[75]:

	column_a	column_b	column_c	column_d	column_e	column_f	column_g	column_h	column_i	column_j
0	True	False	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.00000	0.03760
1	True	False	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549
2	True	False	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198
3	True	False	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000
4	True	False	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399
5	True	False	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.00000
6	True	False	0.97588	-0.10602	0.94601	-0.20800	0.92806	-0.28350	0.85996	-0.04549
7	False	False	0.00000	0.00000	0.00000	0.00000	1.00000	-1.00000	0.00000	0.00000
8	True	False	0.96355	-0.07198	1.00000	-0.14333	1.00000	-0.21313	1.00000	-0.04549
9	True	False	-0.01864	-0.08459	0.00000	0.00000	0.00000	0.00000	0.11470	-0.04549

```
In [76]: pd.set_option('display.max_rows',1000000000)
pd.set_option('display.max_columns',1000000000)
pd.set_option('display.width',95)
```

```
In [77]: print('this DataFrame has %d Rowws and %d columns'%(df.shape))
```

this DataFrame has 351 Rowws and 35 columns

```
In [78]: df.head()
```

Out[78]:

	column_a	column_b	column_c	column_d	column_e	column_f	column_g	column_h	column_i	column_j
0	True	False	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.00000	0.03760
1	True	False	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549
2	True	False	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198
3	True	False	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000
4	True	False	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399

```
In [79]: features_matrix=df.iloc[:,0:34]
```

```
In [80]: target_vector=df.iloc[:,-1]
```

```
In [81]: print('The Features Matrix Has %d Rows and %d Column(s)'%(features_matrix.shape))
print('The Target Matrix Has %d Rows and %d Column(s)'
      %(np.array(target_vector).reshape(-1,1).shape))
```

The Features Matrix Has 351 Rows and 34 Column(s)
The Target Matrix Has 351 Rows and 1 Column(s)

```
In [82]: features_matrix_standardized=StandardScaler().fit_transform(features_matrix)
```

```
In [83]: algorithm=LogisticRegression(penalty=None,dual=False,tol=1e-4,C=1.0,fit_intercept=True,
intercept_scaling=1,class_weight=None,random_state=None,
solver='lbfgs',max_iter=1000,multi_class='auto',verbose=0,
warm_start=False,n_jobs=None,l1_ratio=None)
```

```
In [84]: logistic_Regression_Mode=algorithm.fit(features_matrix_standardized,target_vector)
```

```
In [85]: observation=[[1,0,0.99539,-0.5889,0.8524299999999999,0.02306,0.8339799999999999,-0.37708,
1.0,0.0376,0.8524-2999999999999999,-0.17755,0.59755,-0.44945,0.60536,-0.38223
,0.84356000000000001,-0.38542,0.58212,-0.32192,0.56971,-0.29674,0.36946,
-0.47357,0.56811,-0.51171,0.41078000000000003,-0.46168000000000003,0.21256,
-0.3409,0.112267,-0.54487,0.18641,-0.453]]
```

```
In [86]: predictions=logistic_Regression_Mode.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 [87]: 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 [88]: print("""The model says the probability of the observstion we passed belonging to
class['b']is %s"""%(algorithm.predict_proba(observation)[0][0]))
print("""The model says the probability of the observstion we passed belonging to
class['g']is %s""%(algorithm.predict_proba(observation)[observation[0][1]]))
```

The model says the probability of the observstion we passed belonging to
class['b']is 0.0
The model says the probability of the observstion we passed belonging to
class['g']is [0. 1.]

```
In [ ]:
```

```
In [ ]:
```