

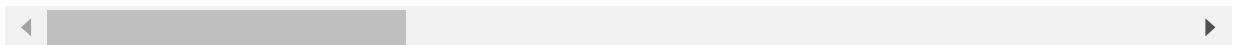
```
In [1]: import pandas as pd
cancer=pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/Cancer.csv')
```

```
In [2]: cancer.head()
```

```
Out[2]:
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840
1	842517	M	20.57	17.77	132.90	1326.0	0.08474
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960
3	84348301	M	11.42	20.38	77.58	386.1	0.14250
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030

5 rows × 33 columns



```
In [3]: cancer.columns
```

```
Out[3]: Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
              'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
              'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
              'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
              'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
              'fractal_dimension_se', 'radius_worst', 'texture_worst',
              'perimeter_worst', 'area_worst', 'smoothness_worst',
              'compactness_worst', 'concavity_worst', 'concave points_worst',
              'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
              dtype='object')
```

```
In [12]: y=cancer['diagnosis']
x=cancer[['radius_mean', 'texture_mean', 'perimeter_mean',
          'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
          'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
          'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
          'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
          'fractal_dimension_se', 'radius_worst', 'texture_worst',
          'perimeter_worst', 'area_worst', 'smoothness_worst',
          'compactness_worst', 'concavity_worst', 'concave points_worst',
          'symmetry_worst', 'fractal_dimension_worst']]
```

```
In [13]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=2529)
```

```
In [20]: from sklearn.linear_model import LogisticRegression
model=LogisticRegression(max_iter=5000)
```

```
In [21]: model.fit(x_train,y_train)
```

```
Out[21]: LogisticRegression(max_iter=5000)
```

```
In [22]: y_pred=model.predict(x_test)
         y_pred
```

```
Out[22]: array(['B', 'M', 'M', 'B', 'M', 'B', 'M', 'B', 'M', 'B', 'B', 'M', 'B',
                'M', 'B', 'B', 'M', 'B', 'M', 'B', 'B', 'B', 'B', 'M',
                'B', 'B', 'M', 'B', 'M', 'B', 'B', 'B', 'B', 'M', 'B', 'B',
                'M', 'M', 'M', 'M', 'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B',
                'B', 'B', 'B', 'B', 'B', 'M', 'M', 'M', 'B', 'M', 'B', 'M',
                'M', 'M', 'B', 'M', 'M', 'B', 'M', 'B', 'M', 'B', 'M', 'B',
                'M', 'M', 'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'B',
                'B', 'B', 'B', 'M', 'B', 'M', 'B', 'B', 'M', 'B', 'M', 'B',
                'B', 'M', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'B',
                'M', 'B', 'B', 'M', 'B', 'M', 'B', 'M', 'M', 'B', 'B', 'B',
                'M', 'B', 'B', 'M', 'M', 'B', 'B', 'M', 'B', 'M', 'B', 'M',
                'M', 'B', 'M', 'M', 'B', 'B', 'M', 'B', 'M', 'B', 'M', 'B'],
              dtype=object)
```

```
In [27]: from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
         accuracy_score(y_test, y_pred)
```

```
Out[27]: 0.965034965034965
```

```
In [28]: confusion_matrix(y_test, y_pred)
```

```
Out[28]: array([[80,  3],
                [ 2, 58]], dtype=int64)
```

```
In [30]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
B	0.98	0.96	0.97	83
M	0.95	0.97	0.96	60
accuracy			0.97	143
macro avg	0.96	0.97	0.96	143
weighted avg	0.97	0.97	0.97	143

```
In [ ]:
```