

```
In [1]: 1 from sklearn.datasets import load_breast_cancer
        2 ds = load_breast_cancer()
        3
```

```
In [2]: 1 X=ds.data
        2
```

```
In [3]: 1 X.shape
```

```
Out[3]: (569, 30)
```

```
In [4]: 1 X[0]
```

```
Out[4]: array([1.799e+01, 1.038e+01, 1.228e+02, 1.001e+03, 1.184e-01, 2.776e-01,
               3.001e-01, 1.471e-01, 2.419e-01, 7.871e-02, 1.095e+00, 9.053e-01,
               8.589e+00, 1.534e+02, 6.399e-03, 4.904e-02, 5.373e-02, 1.587e-02,
               3.003e-02, 6.193e-03, 2.538e+01, 1.733e+01, 1.846e+02, 2.019e+03,
               1.622e-01, 6.656e-01, 7.119e-01, 2.654e-01, 4.601e-01, 1.189e-01])
```

```
In [5]: 1 ds.feature_names
```

```
Out[5]: array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
               'mean smoothness', 'mean compactness', 'mean concavity',
               'mean concave points', 'mean symmetry', 'mean fractal dimension',
               'radius error', 'texture error', 'perimeter error', 'area error',
               'smoothness error', 'compactness error', 'concavity error',
               'concave points error', 'symmetry error',
               'fractal dimension error', 'worst radius', 'worst texture',
               'worst perimeter', 'worst area', 'worst smoothness',
               'worst compactness', 'worst concavity', 'worst concave points',
               'worst symmetry', 'worst fractal dimension'], dtype='<U23')
```

```
In [6]: 1 y=ds.target
```

```
In [7]: 1 ds.target_names
```

```
Out[7]: array(['malignant', 'benign'], dtype='<U9')
```

```
In [8]: 1 y.shape
```

```
Out[8]: (569,)
```

```
In [9]: 1 from sklearn.model_selection import train_test_split
```

```
In [10]: 1 xtrain,xtest,ytrain,ytest = train_test_split(X,y,test_size=0.3, random_
```

```
In [11]: 1 xtrain.shape,xtest.shape,ytrain.shape,ytest.shape
```

```
Out[11]: ((398, 30), (171, 30), (398,), (171,))
```

```
In [12]: 1 from sklearn.linear_model import LogisticRegression
```

```
In [13]: 1 lrc=LogisticRegression()
```

```
In [14]: 1 lrc.fit(xtrain,ytrain)
```

C:\Users\CompLab05\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

```
Out[14]: LogisticRegression()
```

```
In [15]: 1 preds=lrc.predict(xtest)
```

```
In [16]: 1 preds
```

```
Out[16]: array([1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1,
                0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
                1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1,
                1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1,
                0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
                1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1,
                1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1])
```

```
In [17]: 1 pred_probs=lrc.predict_proba(xtest)
```

```
In [18]: 1 pred_probs
```

```
Out[18]: array([[4.98744039e-01, 5.01255961e-01],
                [6.00247640e-01, 3.99752360e-01],
                [2.23885739e-03, 9.97761143e-01],
                [9.99430237e-01, 5.69763491e-04],
                [8.08535937e-01, 1.91464063e-01],
                [9.99091815e-01, 9.08184877e-04],
                [9.99997101e-01, 2.89905413e-06],
                [9.95098004e-01, 4.90199566e-03],
                [2.14139728e-02, 9.78586027e-01],
                [1.50701737e-02, 9.84929826e-01],
                [2.36266499e-02, 9.76373350e-01],
                [9.51473277e-01, 4.85267230e-02],
                [8.70018969e-01, 1.29981031e-01],
                [1.09074397e-02, 9.89092560e-01],
                [2.58993391e-01, 7.41006609e-01],
                [1.08845557e-01, 8.91154443e-01],
                [4.10997806e-02, 9.58900219e-01],
                [2.17466950e-02, 9.78253305e-01],
                [2.99171260e-04, 9.99700829e-01],
                [5.00000000e-01, 5.00000000e-01])
```

```
In [19]: 1 pred_probs[:,0]
```

```
Out[19]: array([4.98744039e-01, 6.00247640e-01, 2.23885739e-03, 9.99430237e-01,
      8.08535937e-01, 9.99091815e-01, 9.99997101e-01, 9.95098004e-01,
      2.14139728e-02, 1.50701737e-02, 2.36266499e-02, 9.51473277e-01,
      8.70018969e-01, 1.09074397e-02, 2.58993391e-01, 1.08845557e-01,
      4.10997806e-02, 2.17466950e-02, 2.99171260e-04, 9.99999932e-01,
      6.87006650e-03, 5.12532333e-02, 9.99548102e-01, 2.21391408e-02,
      8.99964996e-01, 1.25849955e-01, 2.57408292e-02, 9.99999841e-01,
      1.00000000e+00, 9.98746092e-01, 1.00000000e+00, 2.44646705e-03,
      9.99909598e-01, 9.50154309e-01, 1.80650613e-02, 2.98047825e-02,
      9.99778876e-01, 6.70694356e-02, 6.39114569e-01, 4.57688202e-03,
      1.58889958e-03, 2.05005148e-01, 7.49618758e-02, 8.41852031e-03,
      1.93003032e-02, 8.86028034e-01, 3.40328289e-03, 1.98314941e-03,
      3.39349518e-01, 9.98247052e-01, 9.99988040e-01, 1.00000000e+00,
      2.29868018e-03, 3.21740831e-03, 1.51049230e-02, 4.80151429e-02,
      1.03953249e-02, 9.53749995e-01, 1.26949235e-02, 8.78700924e-04,
      6.31005043e-03, 9.26550026e-01, 2.59370093e-01, 8.20169189e-01,
      2.55054815e-02, 6.91499302e-02, 7.63388335e-02, 9.99999989e-01,
      2.56359924e-03, 4.47847330e-02, 1.75711336e-03, 9.53354077e-04,
      4.70258930e-01, 9.99663583e-01, 2.48257622e-03, 9.99999796e-01,
      1.07940951e-01, 1.84406921e-01, 3.09802777e-03, 9.99123484e-01,
      1.94185619e-03, 9.99999740e-01, 1.68781649e-02, 9.96619767e-01,
      5.14148245e-03, 6.49051803e-03, 5.59247280e-01, 6.78086184e-02,
      9.92702074e-01, 2.74409459e-03, 3.14838368e-02, 9.99861105e-01,
      1.56287165e-03, 3.65349492e-04, 9.99638937e-01, 6.95242934e-01,
      6.14503396e-04, 1.28759746e-02, 5.76332688e-03, 2.62340823e-02,
      5.38160899e-02, 7.02626635e-03, 3.90641047e-03, 3.91689797e-01,
      2.38001054e-02, 2.73554655e-02, 7.12090596e-04, 3.39812990e-03,
      9.97403261e-01, 9.80048549e-01, 1.40695065e-01, 4.80057024e-03,
      6.09673538e-01, 8.04638217e-03, 9.98045303e-01, 9.99995766e-01,
      1.02952474e-03, 1.00138430e-02, 1.52894850e-03, 1.18119993e-02,
      5.61341627e-02, 9.99998228e-01, 9.99999614e-01, 3.32290724e-03,
      7.60584728e-03, 1.00000000e+00, 9.99314723e-01, 9.73705761e-01,
      1.00000000e+00, 9.99999966e-01, 2.09778859e-01, 2.01846311e-02,
      1.53346103e-02, 1.00000000e+00, 1.43552956e-04, 9.92630235e-01,
      9.99040651e-01, 3.46567125e-01, 7.09308837e-03, 6.77515307e-01,
      9.40342987e-01, 9.99803900e-01, 8.12544215e-03, 9.96658727e-01,
      2.91808769e-01, 9.99660427e-01, 1.32836587e-03, 1.74750177e-02,
      4.25906588e-03, 9.99997216e-01, 8.08970602e-02, 2.34407574e-03,
      6.69774658e-01, 2.24744500e-01, 5.32406442e-02, 5.56928074e-03,
      7.87074906e-03, 1.57054547e-01, 1.62020113e-02, 3.96030294e-03,
      6.34681602e-03, 6.61828496e-03, 8.61382070e-01, 3.88171854e-01,
      3.54705755e-02, 5.87579198e-02, 9.99998511e-01, 9.99999831e-01,
      9.99974059e-01, 1.62149081e-02, 8.96948292e-03])
```

In [20]: 1 pred_probs[:,1]

Out[20]: array([5.01255961e-01, 3.99752360e-01, 9.97761143e-01, 5.69763491e-04,
1.91464063e-01, 9.08184877e-04, 2.89905413e-06, 4.90199566e-03,
9.78586027e-01, 9.84929826e-01, 9.76373350e-01, 4.85267230e-02,
1.29981031e-01, 9.89092560e-01, 7.41006609e-01, 8.91154443e-01,
9.58900219e-01, 9.78253305e-01, 9.99700829e-01, 6.78604068e-08,
9.93129933e-01, 9.48746767e-01, 4.51898326e-04, 9.77860859e-01,
1.00035004e-01, 8.74150045e-01, 9.74259171e-01, 1.58839282e-07,
7.23381203e-20, 1.25390750e-03, 3.91778449e-11, 9.97553533e-01,
9.04018960e-05, 4.98456910e-02, 9.81934939e-01, 9.70195218e-01,
2.21123783e-04, 9.32930564e-01, 3.60885431e-01, 9.95423118e-01,
9.98411100e-01, 7.94994852e-01, 9.25038124e-01, 9.91581480e-01,
9.80699697e-01, 1.13971966e-01, 9.96596717e-01, 9.98016851e-01,
6.60650482e-01, 1.75294763e-03, 1.19602334e-05, 1.20789100e-13,
9.97701320e-01, 9.96782592e-01, 9.84895077e-01, 9.51984857e-01,
9.89604675e-01, 4.62500048e-02, 9.87305076e-01, 9.99121299e-01,
9.93689950e-01, 7.34499736e-02, 7.40629907e-01, 1.79830811e-01,
9.74494518e-01, 9.30850070e-01, 9.23661166e-01, 1.05353716e-08,
9.97436401e-01, 9.55215267e-01, 9.98242887e-01, 9.99046646e-01,
5.29741070e-01, 3.36417399e-04, 9.97517424e-01, 2.03738004e-07,
8.92059049e-01, 8.15593079e-01, 9.96901972e-01, 8.76516125e-04,
9.98058144e-01, 2.60479654e-07, 9.83121835e-01, 3.38023290e-03,
9.94858518e-01, 9.93509482e-01, 4.40752720e-01, 9.32191382e-01,
7.29792607e-03, 9.97255905e-01, 9.68516163e-01, 1.38895434e-04,
9.98437128e-01, 9.99634651e-01, 3.61063223e-04, 3.04757066e-01,
9.99385497e-01, 9.87124025e-01, 9.94236673e-01, 9.73765918e-01,
9.46183910e-01, 9.92973734e-01, 9.96093590e-01, 6.08310203e-01,
9.76199895e-01, 9.72644535e-01, 9.99287909e-01, 9.96601870e-01,
2.59673914e-03, 1.99514508e-02, 8.59304935e-01, 9.95199430e-01,
3.90326462e-01, 9.91953618e-01, 1.95469689e-03, 4.23415761e-06,
9.98970475e-01, 9.89986157e-01, 9.98471051e-01, 9.88188001e-01,
9.43865837e-01, 1.77158381e-06, 3.85549276e-07, 9.96677093e-01,
9.92394153e-01, 7.50672006e-18, 6.85276841e-04, 2.62942392e-02,
2.53046639e-11, 3.40911214e-08, 7.90221141e-01, 9.79815369e-01,
9.84665390e-01, 3.07497033e-10, 9.99856447e-01, 7.36976527e-03,
9.59349422e-04, 6.53432875e-01, 9.92906912e-01, 3.22484693e-01,
5.96570131e-02, 1.96099991e-04, 9.91874558e-01, 3.34127277e-03,
7.08191231e-01, 3.39572567e-04, 9.98671634e-01, 9.82524982e-01,
9.95740934e-01, 2.78447703e-06, 9.19102940e-01, 9.97655924e-01,
3.30225342e-01, 7.75255500e-01, 9.46759356e-01, 9.94430719e-01,
9.92129251e-01, 8.42945453e-01, 9.83797989e-01, 9.96039697e-01,
9.93653184e-01, 9.93381715e-01, 1.38617930e-01, 6.11828146e-01,
9.64529425e-01, 9.41242080e-01, 1.48887717e-06, 1.69144772e-07,
2.59412168e-05, 9.83785092e-01, 9.91030517e-01])

In [21]: 1 from sklearn.metrics import accuracy_score, confusion_matrix

In [22]: 1 accuracy_score(ytest, preds)

Out[22]: 0.9298245614035088

In [23]: 1 confusion_matrix(ytest, preds)

Out[23]: array([[57, 6],
[6, 102]], dtype=int64)

In []:

1