```
from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer()

cancer.keys()
```

```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR',
'feature_names', 'filename', 'data_module'])
```

```
cancer['data'].shape
```

```
(569, 30)
```

```
X = cancer['data']
y = cancer['target']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
# Fit only to the training data
scaler.fit(X_train)
```

```
StandardScaler()
```

```
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

from sklearn.neural_network import MLPClassifier

mlp = MLPClassifier(hidden_layer_sizes=(30,30,30))

mlp.fit(X_train,y_train)
```

```
C:\Users\MGM\anaconda3\Lib\site-packages\sklearn\neural_network\
_multilayer_perceptron.py:690: ConvergenceWarning: Stochastic
Optimizer: Maximum iterations (200) reached and the optimization
hasn't converged yet.
  warnings.warn(
```

```
MLPClassifier(hidden_layer_sizes=(30, 30, 30))
```

```
predictions = mlp.predict(X_test)

from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predictions))
```

```
[[45  4]
 [ 2 92]]
```

```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support
```

```
           0       0.96      0.92      0.94        49
           1       0.96      0.98      0.97        94

    accuracy                           0.96       143
   macro avg       0.96      0.95      0.95       143
weighted avg       0.96      0.96      0.96       143
```

```
len(mlp.coefs_)
```

```
4
```

```
len(mlp.coefs_[0])
```

```
30
```

```
len(mlp.intercepts_[0])
```

```
30
```

```
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

```
from sklearn.neural_network import MLPClassifier
```

```
mlp = MLPClassifier(hidden_layer_sizes=(30,40,50))
```

```
mlp.fit(X_train,y_train)
```

```
MLPClassifier(hidden_layer_sizes=(30, 40, 50))
```

```
predictions = mlp.predict(X_test)
```

```
from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predictions))
```

```
[[38 11]
 [13 81]]
```

```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.75      0.78      0.76        49
           1       0.88      0.86      0.87        94

    accuracy                           0.83       143
   macro avg       0.81      0.82      0.82       143
weighted avg       0.83      0.83      0.83       143
```

```
len(mlp.coefs_)
```

```
4
```

```
len(mlp.coefs_[0])
```

```
30
```

```
len(mlp.intercepts_[0])
```

```
30
```