

MACHINE LEARNING



Disusun Oleh:

Nama : Ilpan

NPM : 41155050210046

Kelas : A – 2

TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS LANGLANGBUANA

2024

Load dataset

```
[1]: from sklearn.datasets import load_breast_cancer
     print (load_breast_cancer().DESCR)
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
-----

**Data Set Characteristics:**

: Number of Instances: 569

: Number of Attributes: 30 numeric, predictive attributes and the class

: Attribute Information:
  - radius (mean of distances from center to points on the perimeter)
  - texture (standard deviation of gray-scale values)
  - perimeter
  - area
  - smoothness (local variation in radius lengths)
  - compactness (perimeter^2 / area - 1.0)
```

```
[2]: #Load_breast_cancer?
     X, y = load_breast_cancer(return_X_y=True)
     X.shape
```

```
[2]: (569, 30)
```

```
[3]: load_breast_cancer?
     #X, y = load_breast_cancer(return_X_y=True)
     #X.shape

Signature: load_breast_cancer(*, return_X_y=False, as_frame=False)
Docstring:
Load and return the breast cancer wisconsin dataset (classification).

The breast cancer dataset is a classic and very easy binary classification
dataset.

=====
Classes                2
Samples per class      212(M), 357(B)
Samples total          569
Dimensionality          30
Features               real, positive
=====

The copy of UCI ML Breast Cancer Wisconsin (Diagnostic) dataset is
downloaded from:
https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

Read more in the :ref:`User Guide <breast_cancer_dataset>`.

Parameters
-----
return_X_y : bool, default=False
    If True, returns ``(data, target)`` instead of a Bunch object
```

Training & Testing

```
[5]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.2,
                                                    random_state=0)

print(f'X_train shape {X_train.shape}')
print(f'X_test shape {X_test.shape}')

X_train shape (455, 30)
X_test shape (114, 30)
```

[]:



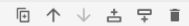
Naïve Bayes dengan Scikit Learn

```
[7]: from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

model = GaussianNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy_score(y_test, y_pred)
```

[7]: 0.9298245614035088

[]:



```
[9]: model.score(X_test, y_test)
```

[9]: 0.9298245614035088

[]:

