TUGAS

MACHINE LEARNING

Disusun oleh :

Nama : Ichsan Haryadi Putra

NPM : 41155050210042

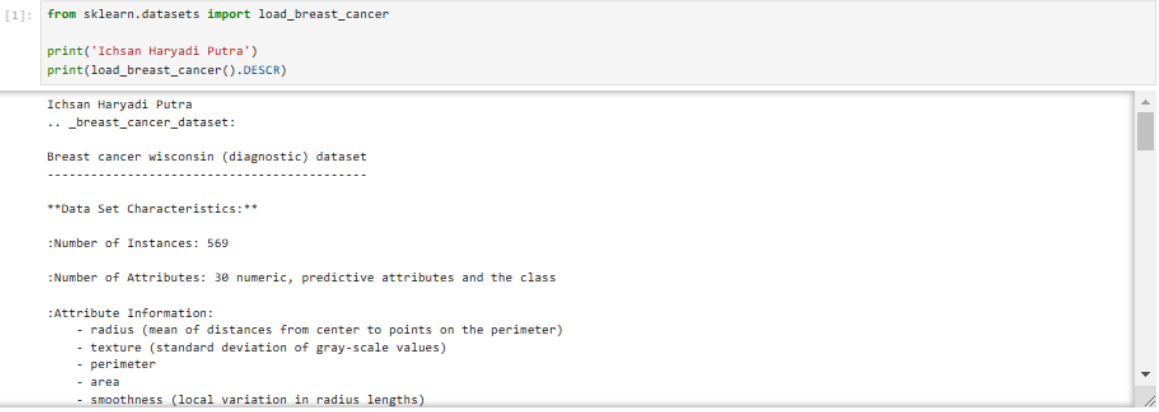
Kelas : Teknik Informatika B

UNIVERSITAS LANGLANGBUANA

BANDUNG

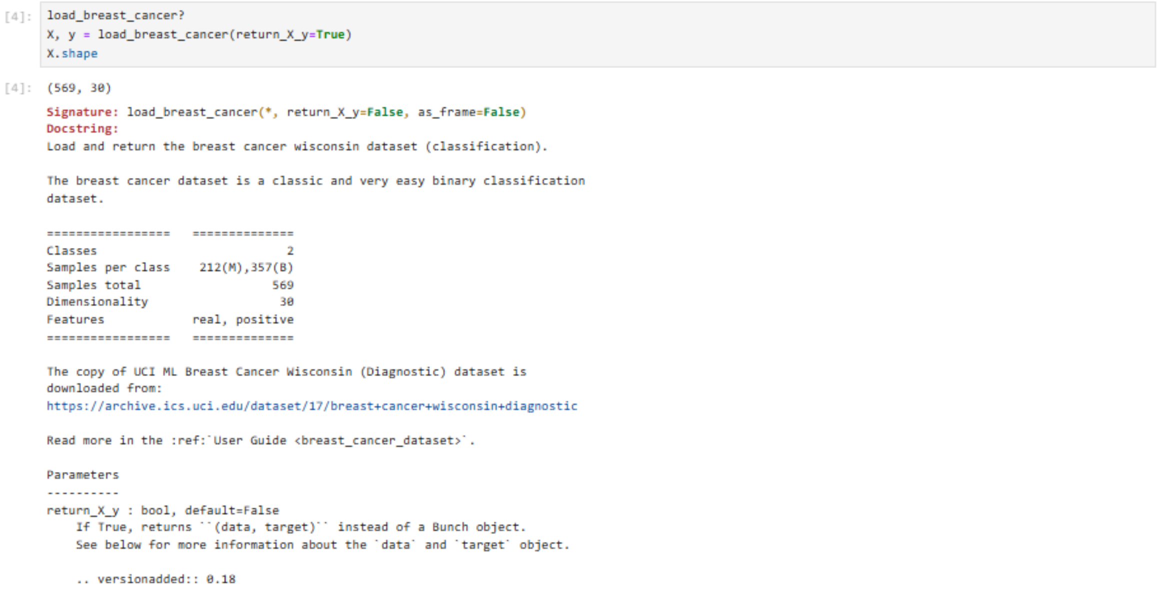
Tugas Petermuan 4

1. Persiapan Datasets



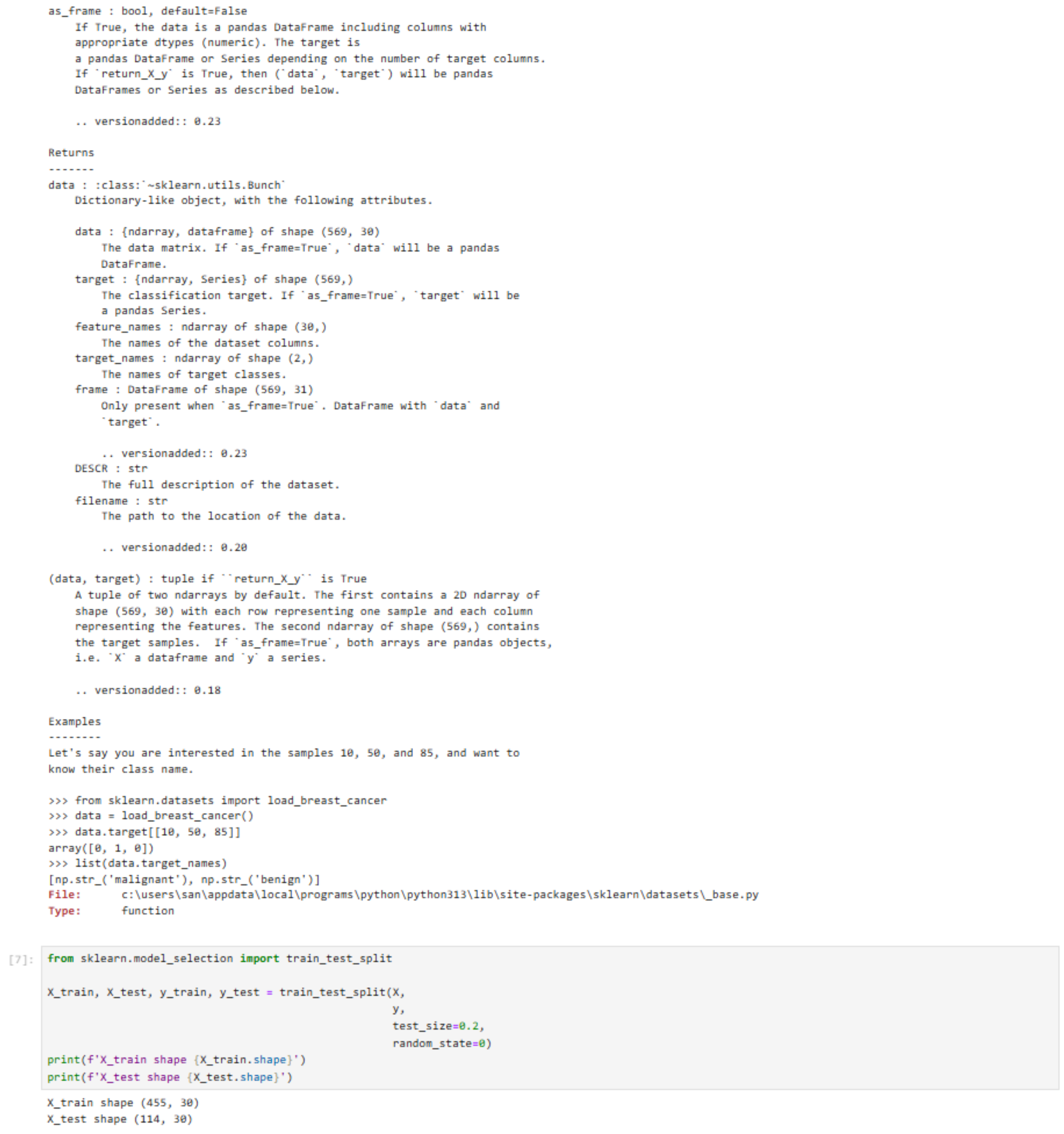
```
[1]: from sklearn.datasets import load_breast_cancer

     print('Ichsan Haryadi Putra')
     print(load_breast_cancer().DESCR)
```

```
Ichsan Haryadi Putra
.. _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)
```



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

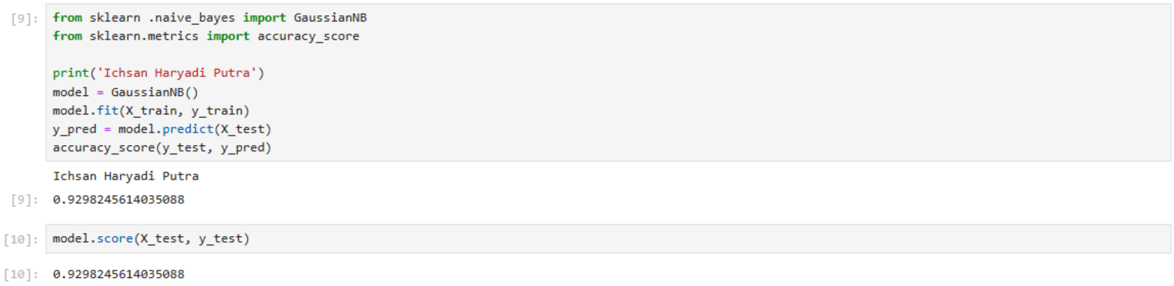
```
[4]: (569, 30)

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.
    See below for more information about the `data` and `target` object.

    .. versionadded:: 0.18
```



```
as_frame : bool, default=False
    If True, the data is a pandas DataFrame including columns with
    appropriate dtypes (numeric). The target is
    a pandas DataFrame or Series depending on the number of target columns.
    If `return_X_y` is True, then (`data`, `target`) will be pandas
    DataFrames or Series as described below.

    .. versionadded:: 0.23

Returns
-------
data : :class:`~sklearn.utils.Bunch`
    Dictionary-like object, with the following attributes.

    data : {ndarray, dataframe} of shape (569, 30)
        The data matrix. If `as_frame=True`, `data` will be a pandas
        DataFrame.
    target : {ndarray, Series} of shape (569,)
        The classification target. If `as_frame=True`, `target` will be
        a pandas Series.
    feature_names : ndarray of shape (30,)
        The names of the dataset columns.
    target_names : ndarray of shape (2,)
        The names of target classes.
    frame : DataFrame of shape (569, 31)
        Only present when `as_frame=True`. DataFrame with `data` and
        `target`.

        .. versionadded:: 0.23
    DESCR : str
        The full description of the dataset.
    filename : str
        The path to the location of the data.

        .. versionadded:: 0.20

(data, target) : tuple if ``return_X_y`` is True
    A tuple of two ndarrays by default. The first contains a 2D ndarray of
    shape (569, 30) with each row representing one sample and each column
    representing the features. The second ndarray of shape (569,) contains
    the target samples.  If `as_frame=True`, both arrays are pandas objects,
    i.e. `X` a dataframe and `y` a series.

    .. versionadded:: 0.18

Examples
--------
Let's say you are interested in the samples 10, 50, and 85, and want to
know their class name.

>>> from sklearn.datasets import load_breast_cancer
>>> data = load_breast_cancer()
>>> data.target[[10, 50, 85]]
array([0, 1, 0])
>>> list(data.target_names)
[np.str_('malignant'), np.str_('benign')]
File:      c:\users\san\appdata\local\programs\python\python313\lib\site-packages\sklearn\datasets\_base.py
Type:      function
```

```
[7]: 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)
```

2. Implementasi Naive Bayes



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

     print('Ichsan Haryadi Putra')
     model = GaussianNB()
     model.fit(X_train, y_train)
     y_pred = model.predict(X_test)
     accuracy_score(y_test, y_pred)

     Ichsan Haryadi Putra
[9]: 0.9298245614035088

[10]: model.score(X_test, y_test)

[10]: 0.9298245614035088
```