3_Dataset

January 14, 2025

[ ]:
```
'''
   Dataset -->

   A dataset is a collection of data, typically organized in a structured␣
↪format, which is used for analysis,
   training machine learning models, or conducting experiments. In a dataset,␣
↪each data point (also known as an observation,
   sample, or instance) contains information about a specific entity, and this␣
↪information is often stored in a tabular format
'''
```

[ ]:
```
'''
   Key Components of a Dataset -->

   Features (Attributes or Variables) -->

   Definition : Features are the individual measurable properties or␣
↪characteristics of the data.
   These can be inputs to a model in machine learning.
   Example : In a dataset of housing prices, features could include house␣
↪size, number of rooms, location, etc.

   Samples (Rows, Records, or Observations) -->

   Definition : Each sample or observation represents a single entry in the␣
↪dataset, which is characterized
   by the values of the features.
   Example : In a dataset of car sales, each row might represent a specific␣
↪car and contain information like
   the make, model, year, price, etc.

   Target Variable (Label or Output) -->

   Definition : In supervised learning tasks, the target variable is the value␣
↪that the model is supposed to predict or classify.
   It is the output of the model.
```

```
   Example : In a dataset for predicting house prices, the target variable␣
↪could be the actual price of the house.

   Structure -->

   Tabular : Most common format, where data is arranged in rows and columns␣
↪(similar to a spreadsheet).
   Non-tabular : Datasets may also come in formats such as images, audio, or␣
↪text.
'''
```

[ ]:
```
'''
   Example of a Tabular Dataset -->

       ID            House Size (sq ft)  Bedrooms   Price ($)
       1                   2000               3          300,000
       2                   1500               2          200,000
       3                   2500               4          400,000

   Features --> "House Size" and "Bedrooms"
   Target Variable --> "Price"
   Samples --> Each row represents one house
'''
```

[ ]:
```
'''
   Types of Datasets -->

   Training Dataset -->
   Used to train a machine learning model by feeding it both features and the␣
↪target variable.
   Example : A dataset of images and their corresponding labels for␣
↪classifying objects in images.

   Testing Dataset -->
   Used to evaluate the performance of a trained machine learning model.
   The model makes predictions on this dataset, and the results are compared␣
↪to the actual values to measure accuracy.

   Validation Dataset -->
   Used to fine-tune the hyperparameters of the model during training to avoid␣
↪overfitting.

   Unlabeled Dataset -->
   Contains only input data (features) without the corresponding output␣
↪(target variable).
```

```
   Commonly used in unsupervised learning tasks like clustering or anomaly␣
↪detection.

   Labeled Dataset -->
   Contains both input data and the corresponding output, used in supervised␣
↪learning tasks.
'''
```

[ ]:
```
'''
   Formats of Datasets -->

   CSV (Comma-Separated Values) : A common format for tabular data.
   Excel : Data stored in spreadsheets, often in .xls or .xlsx formats.
   Image files : Datasets of images, often stored as .jpg, .png, etc.
   JSON (JavaScript Object Notation) : Used for semi-structured data.
   SQL Databases : Data stored in relational databases.
'''
```

[ ]:
```
'''
   Types of Data in a Dataset -->

   Numerical Data -->
   Continuous : Data that can take any value within a range (e.g., house␣
↪price, temperature).
   Discrete : Data that can only take specific values (e.g., number of␣
↪bedrooms).

   Categorical Data -->
   Data that represents categories or groups (e.g., car make, gender).
   Can be nominal (no order) or ordinal (with a meaningful order).

   Textual Data -->
   Data in the form of text (e.g., product reviews, tweets).

   Image Data -->
   Images, often represented as arrays of pixel values.
'''
```

[ ]:
```
'''
   Importing a dataset -->

   In Python, datasets can be imported from various sources such as files␣
↪(CSV, Excel, etc.),
   databases, or even directly from web URLs. Below are common ways to import␣
↪datasets using
   popular libraries like pandas and numpy
```

```
    Importing Datasets using pandas -->
    df = pd.read_csv('file')

    Importing Datasets from sklearn.datasets (Toy Datasets) -->
    from sklearn.datasets import load_database
    many datasets are available on sklearn you can use any
'''
```

[1]:
```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
```

[3]:
```
pd_data = pd.read_csv('Data/Data.csv')
sk_data = load_iris()
```

[4]:
```
pd_data.head()
```

[4]:

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes

[ ]:
```
# sklearn data is not a DataFrame so we can't do head

sk_data
```

[ ]:
```
{'data': array([[5.1, 3.5, 1.4, 0.2],
        [4.9, 3. , 1.4, 0.2],
        [4.7, 3.2, 1.3, 0.2],
        [4.6, 3.1, 1.5, 0.2],
        [5. , 3.6, 1.4, 0.2],
        [5.4, 3.9, 1.7, 0.4],
        [4.6, 3.4, 1.4, 0.3],
        [5. , 3.4, 1.5, 0.2],
        [4.4, 2.9, 1.4, 0.2],
        [4.9, 3.1, 1.5, 0.1],
        [5.4, 3.7, 1.5, 0.2],
        [4.8, 3.4, 1.6, 0.2],
        [4.8, 3. , 1.4, 0.1],
        [4.3, 3. , 1.1, 0.1],
        [5.8, 4. , 1.2, 0.2],
        [5.7, 4.4, 1.5, 0.4],
        [5.4, 3.9, 1.3, 0.4],
        [5.1, 3.5, 1.4, 0.3],
        [5.7, 3.8, 1.7, 0.3],
```

```
[5.1, 3.8, 1.5, 0.3],
[5.4, 3.4, 1.7, 0.2],
[5.1, 3.7, 1.5, 0.4],
[4.6, 3.6, 1. , 0.2],
[5.1, 3.3, 1.7, 0.5],
[4.8, 3.4, 1.9, 0.2],
[5. , 3. , 1.6, 0.2],
[5. , 3.4, 1.6, 0.4],
[5.2, 3.5, 1.5, 0.2],
[5.2, 3.4, 1.4, 0.2],
[4.7, 3.2, 1.6, 0.2],
[4.8, 3.1, 1.6, 0.2],
[5.4, 3.4, 1.5, 0.4],
[5.2, 4.1, 1.5, 0.1],
[5.5, 4.2, 1.4, 0.2],
[4.9, 3.1, 1.5, 0.2],
[5. , 3.2, 1.2, 0.2],
[5.5, 3.5, 1.3, 0.2],
[4.9, 3.6, 1.4, 0.1],
[4.4, 3. , 1.3, 0.2],
[5.1, 3.4, 1.5, 0.2],
[5. , 3.5, 1.3, 0.3],
[4.5, 2.3, 1.3, 0.3],
[4.4, 3.2, 1.3, 0.2],
[5. , 3.5, 1.6, 0.6],
[5.1, 3.8, 1.9, 0.4],
[4.8, 3. , 1.4, 0.3],
[5.1, 3.8, 1.6, 0.2],
[4.6, 3.2, 1.4, 0.2],
[5.3, 3.7, 1.5, 0.2],
[5. , 3.3, 1.4, 0.2],
[7. , 3.2, 4.7, 1.4],
[6.4, 3.2, 4.5, 1.5],
[6.9, 3.1, 4.9, 1.5],
[5.5, 2.3, 4. , 1.3],
[6.5, 2.8, 4.6, 1.5],
[5.7, 2.8, 4.5, 1.3],
[6.3, 3.3, 4.7, 1.6],
[4.9, 2.4, 3.3, 1. ],
[6.6, 2.9, 4.6, 1.3],
[5.2, 2.7, 3.9, 1.4],
[5. , 2. , 3.5, 1. ],
[5.9, 3. , 4.2, 1.5],
[6. , 2.2, 4. , 1. ],
[6.1, 2.9, 4.7, 1.4],
[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
```

```
[5.6, 3. , 4.5, 1.5],
[5.8, 2.7, 4.1, 1. ],
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
[6. , 2.9, 4.5, 1.5],
[5.7, 2.6, 3.5, 1. ],
[5.5, 2.4, 3.8, 1.1],
[5.5, 2.4, 3.7, 1. ],
[5.8, 2.7, 3.9, 1.2],
[6. , 2.7, 5.1, 1.6],
[5.4, 3. , 4.5, 1.5],
[6. , 3.4, 4.5, 1.6],
[6.7, 3.1, 4.7, 1.5],
[6.3, 2.3, 4.4, 1.3],
[5.6, 3. , 4.1, 1.3],
[5.5, 2.5, 4. , 1.3],
[5.5, 2.6, 4.4, 1.2],
[6.1, 3. , 4.6, 1.4],
[5.8, 2.6, 4. , 1.2],
[5. , 2.3, 3.3, 1. ],
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
[5.8, 2.7, 5.1, 1.9],
[7.1, 3. , 5.9, 2.1],
[6.3, 2.9, 5.6, 1.8],
[6.5, 3. , 5.8, 2.2],
[7.6, 3. , 6.6, 2.1],
[4.9, 2.5, 4.5, 1.7],
[7.3, 2.9, 6.3, 1.8],
[6.7, 2.5, 5.8, 1.8],
[7.2, 3.6, 6.1, 2.5],
[6.5, 3.2, 5.1, 2. ],
[6.4, 2.7, 5.3, 1.9],
[6.8, 3. , 5.5, 2.1],
```

```
        [5.7, 2.5, 5. , 2. ],
        [5.8, 2.8, 5.1, 2.4],
        [6.4, 3.2, 5.3, 2.3],
        [6.5, 3. , 5.5, 1.8],
        [7.7, 3.8, 6.7, 2.2],
        [7.7, 2.6, 6.9, 2.3],
        [6. , 2.2, 5. , 1.5],
        [6.9, 3.2, 5.7, 2.3],
        [5.6, 2.8, 4.9, 2. ],
        [7.7, 2.8, 6.7, 2. ],
        [6.3, 2.7, 4.9, 1.8],
        [6.7, 3.3, 5.7, 2.1],
        [7.2, 3.2, 6. , 1.8],
        [6.2, 2.8, 4.8, 1.8],
        [6.1, 3. , 4.9, 1.8],
        [6.4, 2.8, 5.6, 2.1],
        [7.2, 3. , 5.8, 1.6],
        [7.4, 2.8, 6.1, 1.9],
        [7.9, 3.8, 6.4, 2. ],
        [6.4, 2.8, 5.6, 2.2],
        [6.3, 2.8, 5.1, 1.5],
        [6.1, 2.6, 5.6, 1.4],
        [7.7, 3. , 6.1, 2.3],
        [6.3, 3.4, 5.6, 2.4],
        [6.4, 3.1, 5.5, 1.8],
        [6. , 3. , 4.8, 1.8],
        [6.9, 3.1, 5.4, 2.1],
        [6.7, 3.1, 5.6, 2.4],
        [6.9, 3.1, 5.1, 2.3],
        [5.8, 2.7, 5.1, 1.9],
        [6.8, 3.2, 5.9, 2.3],
        [6.7, 3.3, 5.7, 2.5],
        [6.7, 3. , 5.2, 2.3],
        [6.3, 2.5, 5. , 1.9],
        [6.5, 3. , 5.2, 2. ],
        [6.2, 3.4, 5.4, 2.3],
        [5.9, 3. , 5.1, 1.8]]),
 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
 'frame': None,
 'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10'),
```

```
 'DESCR': '.. _iris_dataset:\n\nIris plants
dataset\n--------------------\n\n**Data Set Characteristics:**\n\n:Number of
Instances: 150 (50 in each of three classes)\n:Number of Attributes: 4 numeric,
predictive attributes and the class\n:Attribute Information:\n    - sepal length
in cm\n    - sepal width in cm\n    - petal length in cm\n    - petal width in
cm\n    - class:\n            - Iris-Setosa\n            - Iris-Versicolour\n
- Iris-Virginica\n\n:Summary Statistics:\n\n============== ==== ==== =======
===== ====================\n                Min  Max   Mean    SD   Class
Correlation\n============== ==== ==== ======= ===== ====================\nsepal
length:   4.3  7.9   5.84   0.83    0.7826\nsepal width:    2.0  4.4   3.05
0.43   -0.4194\npetal length:   1.0  6.9   3.76   1.76    0.9490  (high!)\npetal
width:    0.1  2.5   1.20   0.76    0.9565  (high!)\n============== ==== ====
======= ===== ====================\n\n:Missing Attribute Values: None\n:Class
Distribution: 33.3% for each of 3 classes.\n:Creator: R.A. Fisher\n:Donor:
Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n:Date: July, 1988\n\nThe famous
Iris database, first used by Sir R.A. Fisher. The dataset is taken\nfrom
Fisher\'s paper. Note that it\'s the same as in R, but not as in the
UCI\nMachine Learning Repository, which has two wrong data points.\n\nThis is
perhaps the best known database to be found in the\npattern recognition
literature.  Fisher\'s paper is a classic in the field and\nis referenced
frequently to this day.  (See Duda & Hart, for example.)  The\ndata set contains
3 classes of 50 instances each, where each class refers to a\ntype of iris
plant.  One class is linearly separable from the other 2; the\nlatter are NOT
linearly separable from each other.\n\n|details-
start|\n**References**\n|details-split|\n\n- Fisher, R.A. "The use of multiple
measurements in taxonomic problems"\n  Annual Eugenics, 7, Part II, 179-188
(1936); also in "Contributions to\n  Mathematical Statistics" (John Wiley, NY,
1950).\n- Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene
Analysis.\n  (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page
218.\n- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System\n
Structure and Classification Rule for Recognition in Partially Exposed\n
Environments".  IEEE Transactions on Pattern Analysis and Machine\n
Intelligence, Vol. PAMI-2, No. 1, 67-71.\n- Gates, G.W. (1972) "The Reduced
Nearest Neighbor Rule".  IEEE Transactions\n  on Information Theory, May 1972,
431-433.\n- See also: 1988 MLC Proceedings, 54-64.  Cheeseman et al"s AUTOCLASS
II\n  conceptual clustering system finds 3 classes in the data.\n- Many, many
more …\n\n|details-end|\n',
 'feature_names': ['sepal length (cm)',
  'sepal width (cm)',
  'petal length (cm)',
  'petal width (cm)'],
 'filename': 'iris.csv',
 'data_module': 'sklearn.datasets.data'}
```