7. Data Science – Machine Learning – Train & Test Datasets

Contents

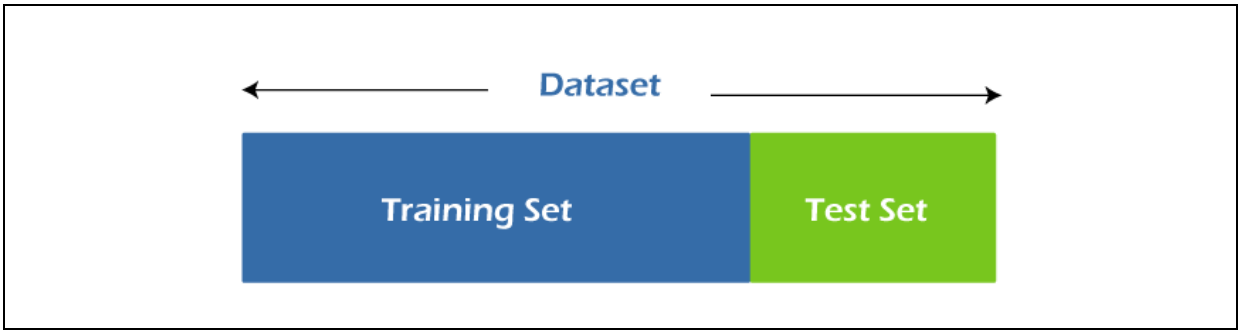
7. Data Science – Machine Learning – Train & Test Datasets

1. Types of Datasets in machine learning

- ✓ There are mainly 3 types of datasets used in Machine learning,
  - Train dataset
  - Test dataset
  - Validation dataset

Note

- ✓ Here validation dataset is an optional but training and test datasets are mandatory



2. Train dataset

- ✓ Train dataset is used to train the models.
- ✓ This is the part of dataset which is used to train the model.
- ✓ Typically, training set contains about 60-70% of total dataset.
- ✓ First step is, model should get train with training dataset, during training model learns the parameters and underlying concepts from dataset.

3. Test dataset

- ✓ Test dataset is used to test the models.
- ✓ Once model training is done then we need to test the model with test dataset.
- ✓ During testing the model we will understand about model performance either good or not.
- ✓ Size of Test set is about 15-30% of total dataset.

4. How to decide size of these 3 sets?

- ✓ There is no thumb rule about choosing the size of these sets, but according to the experts' 70-30 or 60-40 is a good size for train and test set respectively.

5. Creating array

- ✓ We can create an array and split that array

Program Name: Creating an array
demo1.py

```
import numpy as np

dataset = np.arange(10)

print(dataset)
```

Output

```
[0 1 2 3 4 5 6 7 8 9]
```

6. train_test_split(p) function

- ✓ train_test_split(p) is a predefined function in sklearn.model_selection package
- ✓ We need to access this function from sklearn package.
- ✓ By using this function we can split the dataset into train dataset and test dataset.

Program Name: Creating an array and splitting dataset
demo2.py

```
import numpy as np
from sklearn.model_selection import train_test_split

dataset = np.arange(10)

result = train_test_split(dataset)

print(dataset)
print(result)
```

Output

```
C:\Users\Nireekshan\Desktop\PROGRAMS>py demo1.py
[0 1 2 3 4 5 6 7 8 9]
[array([0, 1, 8, 9, 2, 5, 7]), array([3, 6, 4])]

C:\Users\Nireekshan\Desktop\PROGRAMS>py demo1.py
[0 1 2 3 4 5 6 7 8 9]
[array([3, 2, 6, 4, 1, 5, 9]), array([7, 8, 0])]

C:\Users\Nireekshan\Desktop\PROGRAMS>py demo1.py
[0 1 2 3 4 5 6 7 8 9]
[array([7, 8, 4, 3, 0, 6, 2]), array([5, 1, 9])]
```

Program Name: Creating an array and splitting
demo3.py

```
import numpy as np
from sklearn.model_selection import train_test_split

dataset = np.arange(10)

X_train, X_test = train_test_split(dataset)

print(dataset)
print()
print(X_train)
print(X_test)
```

Output

```
[0 1 2 3 4 5 6 7 8 9]

[0 6 5 1 4 9 7]
[8 3 2]
```

Program Name: Creating an array and splitting
demo4.py

```
import numpy as np
from sklearn.model_selection import train_test_split

dataset = np.arange(10)

X_train, X_test = train_test_split(dataset, test_size = 4)

print(dataset)
print()
print(X_train)
print(X_test)
```

Output

```
[0 1 2 3 4 5 6 7 8 9]

[6 5 7 9 3 1]
[2 0 4 8]
```

Program Name: Creating an array and splitting
demo5.py

```
import numpy as np
from sklearn.model_selection import train_test_split

dataset = np.arange(10)

X_train, X_test = train_test_split(dataset, train_size = 6)

print(dataset)
print()
print(X_train)
print(X_test)
```

Output

```
[0 1 2 3 4 5 6 7 8 9]

[1 9 4 3 2 5]
[7 8 6 0]
```

Program Name: Creating an array and splitting
demo6.py

```
import numpy as np
from sklearn.model_selection import train_test_split

dataset = np.arange(10)

X_train, X_test = train_test_split(dataset, test_size = 0.4)

print(dataset)
print()
print(X_train)
print(X_test)
```

Output

```
[0 1 2 3 4 5 6 7 8 9]

[9 5 2 0 8 7]
[1 4 3 6]
```

Program Name: Creating an array and splitting
demo7.py

```
import numpy as np
from sklearn.model_selection import train_test_split

dataset = np.arange(10)

X_train, X_test = train_test_split(dataset, train_size = 0.6)

print(dataset)
print()
print(X_train)
print(X_test)
```

Output

```
[0 1 2 3 4 5 6 7 8 9]

[9 6 4 8 3 5]
[2 1 0 7]
```

Program Name: Creating an array and splitting
demo8.py

```
import numpy as np
from sklearn.model_selection import train_test_split

dataset = np.arange(10)

X_train, X_test = train_test_split(dataset, train_size = 4, test_size
= 4)

print(dataset)
print()
print(X_train)
print(X_test)
```

Output

```
[0 1 2 3 4 5 6 7 8 9]
[0 9 3 6]
[4 7 2 5]
```

Program Name: Creating an array and splitting
demo9.py

```
import numpy as np
from sklearn.model_selection import train_test_split

dataset = np.arange(10)

X_train, X_test = train_test_split(dataset, train_size = 4, test_size
= 10)

print(dataset)
print()
print(X_train)
print(X_test)
```

Output

```
ValueError: test_size=10 should be either positive and smaller than the number of samples 10 or a float
 in the (0, 1) range
```

Program Name: Creating an array and splitting
demo10.py

```
import numpy as np

dataset = np.arange(20)

print(dataset)
```

Output

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
```

Program Name: Creating an array and splitting
demo11.py

```
import numpy as np

dataset = np.arange(20).reshape(2, 10)

print(dataset)
```

Output

```
[[ 0  1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18 19]]
```

Program Name: Creating an array and splitting
demo12.py

```
import numpy as np

dataset = np.arange(20).reshape(2, 10).T

print(dataset)
```

Output

```
[[ 0 10]
 [ 1 11]
 [ 2 12]
 [ 3 13]
 [ 4 14]
 [ 5 15]
 [ 6 16]
 [ 7 17]
 [ 8 18]
 [ 9 19]]
```

Program Name: Creating an array and splitting
demo13.py

```
import numpy as np

X = np.arange(20).reshape(2, 10).T

y = np.arange(10)

print(X)
print()
print(y)
```

Output

```
[[ 0 10]
 [ 1 11]
 [ 2 12]
 [ 3 13]
 [ 4 14]
 [ 5 15]
 [ 6 16]
 [ 7 17]
 [ 8 18]
 [ 9 19]]

[0 1 2 3 4 5 6 7 8 9]
```

Program Name: Creating an array and splitting
demo14.py

```
import numpy as np

X = np.arange(20).reshape(2, 10).T

y = np.arange(10)

print(X)
print()
print(y)
```

Output

```
[[ 0 10]
 [ 1 11]
 [ 2 12]
 [ 3 13]
 [ 4 14]
 [ 5 15]
 [ 6 16]
 [ 7 17]
 [ 8 18]
 [ 9 19]]

[0 1 2 3 4 5 6 7 8 9]
```

Program Name: Creating an array and splitting
demo15.py

```
import numpy as np
from sklearn.model_selection import train_test_split

X = np.arange(20).reshape(2, 10).T

y = np.arange(10)

X_train, X_test, y_train, y_test = train_test_split(X, y)

print(X_train)
print()
print(X_test)
print()
print(y_train)
print()
print(y_test)
```

Output

```
[[ 9 19]
 [ 6 16]
 [ 2 12]
 [ 4 14]
 [ 3 13]
 [ 8 18]
 [ 7 17]]

[[ 5 15]
 [ 1 11]
 [ 0 10]]

[9 6 2 4 3 8 7]

[5 1 0]
```

7. train_test_split(p, random_state = 0) function

- ✓ train_test_split(p, random_state = 0) is a predefined function in sklearn.model_selection package
- ✓ We need to access this function from sklearn package.
- ✓ By using this function we can split the dataset into train dataset and test dataset.
- ✓ We will get the same train and test datasets across different executions.

Program Name: Creating an array and splitting
demo16.py

```
import numpy as np
from sklearn.model_selection import train_test_split

X = np.arange(20).reshape(2, 10).T

y = np.arange(10)

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

print(X_train)
print()
print(X_test)
print()
print(y_train)
print()
print(y_test)
```

Output

```
[[ 9 19]
 [ 1 11]
 [ 6 16]
 [ 7 17]
 [ 3 13]
 [ 0 10]
 [ 5 15]]

[[ 2 12]
 [ 8 18]
 [ 4 14]]

[9 1 6 7 3 0 5]

[2 8 4]
```