7_Splitting

January 14, 2025

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

   Splitting a dataset into train and test sets is an essential step in␣
↪machine learning because
   it helps to evaluate the performance and generalization ability of a model

   1. Model Training and Learning

   Training Set: The training data is used to fit the machine learning model.
   The model "learns" from this data by adjusting its parameters (such as␣
↪weights in a neural network)
   to minimize error or maximize accuracy.
   The goal during training is to have the model learn patterns from the data.

   2. Model Evaluation and Testing

   Test Set: After training, you need to evaluate the model on unseen data to␣
↪check how well it generalizes.
   The test set acts as a proxy for how the model will perform on real-world␣
↪data.
   By evaluating on a test set, you simulate how the model will perform when␣
↪faced with new, unseen data.

   3. Avoiding Overfitting

   If you train the model on the entire dataset and evaluate it on the same␣
↪data, the model might perform well
   simply because it has "memorized" the data rather than learning general␣
↪patterns.
   This leads to overfitting, where the model performs well on the training␣
↪data but poorly on unseen data (test data).
   Splitting the dataset helps to ensure that the model generalizes well and␣
↪doesn't overfit the training data.

   4. Performance Metrics
```

```
    The test set allows you to compute metrics such as accuracy, precision,␣
 ↪recall, F1-score, etc.,
    on data the model has never seen, giving a true measure of its performance.
    These metrics provide insights into how the model will behave in a␣
 ↪real-world environment.

    5. Validation

    In some cases, a third set called the validation set is also used for␣
 ↪hyperparameter tuning during training.
    After the best model is selected, the final evaluation is done on the test␣
 ↪set to give an unbiased estimate of its performance.
'''
```

[ ]:

```
#   Common Train-Test Splits -->

#   A common split ratio is 80% training and 20% testing, or 70% training and␣
 ↪30% testing,
#   depending on the size of the dataset.
```

[1]:

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

[2]:

```
dataset = pd.read_csv('Data/Data.csv')
dataset
```

[2]:

```
   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
5   France  35.0  58000.0       Yes
6    Spain   NaN  52000.0        No
7   France  48.0  79000.0       Yes
8  Germany  50.0  83000.0        No
9   France  37.0  67000.0       Yes
```

[3]:

```
x_data = dataset.iloc[:, :-1]
y_data = dataset.iloc[:, -1]
```

[7]:

```
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size =␣
 ↪0.2, random_state = 1)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
```

```python
print(y_test.shape)
```

```
(8, 3)
(2, 3)
(8,)
(2,)
```