1

00:00:00,530 --> 00:00:03,740

[Auto-generated transcript. Edits may have been applied for clarity.]

Validation is a very common concept in machine learning.

2

00:00:04,010 --> 00:00:07,580

During training, it's typical to split the data into three sets.

3

00:00:07,730 --> 00:00:10,010

Training, validation, and test.

4

00:00:10,520 --> 00:00:18,170

There's also an algorithm called cross validation that builds on this idea to reduce variability and improve model evaluation.

5

00:00:18,920 --> 00:00:23,450

Let's say you divide your data set into training, validation, and test sets.

6

00:00:23,660 --> 00:00:29,600

And after training your model achieves a performance metric, say 98% accuracy.

7

00:00:30,140 --> 00:00:35,660

If you randomly reshuffle the samples in each group and retrain, you might get different results.

8

00:00:35,960 --> 00:00:40,430

96%, 94%, or even 99%.

9

00:00:40,970 --> 00:00:46,520

These fluctuations show how sensitive your evaluation can be to how the data is split.

10

00:00:46,910 --> 00:00:54,080

Cross validation helps address this by providing a more reliable average performance metric across different splits.

11

00:00:54,590 --> 00:01:02,660

Although it's called cross-validation, a more accurate name might be cross testing since it doesn't necessarily require a validation set.

12

00:01:03,020 --> 00:01:06,470

When training and testing, you face two challenges.

13

00:01:06,860 --> 00:01:11,840

First, you're not using the full data set for training because you're holding out a test set.

14

00:01:12,050 --> 00:01:17,210

Second, if you allocate more data for training, the test set becomes smaller.

15

00:01:18,110 --> 00:01:24,350

Cross-validation solves this by dividing the data set into equal parts, called folds.

16

00:01:25,070 --> 00:01:33,530

For example, in five fold cross-validation, the data is split into five equal parts, labeled one through five.

17

00:01:33,920 --> 00:01:38,450

You perform five rounds of training and testing in each round.

18

00:01:38,570 --> 00:01:42,590

Four folds are used for training and one fold is used for testing.

19

00:01:43,220 --> 00:01:51,560

Second, the test fold rotates each round, so every part of the data is used once for testing and multiple times for training.

20

00:01:52,130 --> 00:02:00,140

This ensures that every data point is used for both training and testing, providing a more robust estimate of model performance.

21

00:02:00,440 --> 00:02:06,980

At the end, you average the performance metrics from each round to obtain a final evaluation score.

22

00:02:07,580 --> 00:02:15,200

You can optionally include a validation set in each fold for tasks like early stopping to avoid overfitting, but it's not required.

23

00:02:15,740 --> 00:02:19,340

Cross-validation can be used with or without validation sets.

24

00:02:19,790 --> 00:02:26,120

It helps address the challenges of data partitioning and gives a better understanding of the model's ability to generalize.

25

00:02:26,750 --> 00:02:34,460

For example, if you have 200 data points and divide them into five folds, each fold will contain 40 data points.

26

00:02:34,640 --> 00:02:41,630

In each round, 160 data points are used for training and 40 for testing across the five rounds.

27

00:02:41,840 --> 00:02:46,130

All 200 points are eventually used for both training and testing.

28

00:02:46,430 --> 00:02:52,040

Suppose your performance metrics across the five folds are fold 198%.

29

00:02:52,250 --> 00:03:00,950

Full 295% fold 399% fold 489% fold 571%.

30

00:03:01,520 --> 00:03:09,290

Averaging these results gives a more reliable measure of your model's performance than relying on a single random train test split.