

Training, Validation, and Test sets

The core idea

You never want to evaluate a model on the same data it learned from. That only tells you how well it memorized, not how well it generalizes. The full dataset gets split into three parts, each serving a different purpose.

Training Set

- **Purpose:** This is what the model actually learns from. The algorithm adjusts its internal parameters (weights, coefficients, etc.) to minimize error on this data.
- **Typical size:** ~60–80% of the total data.
- **Analogy:** The practice problems a student works through, answer key included.
- **Note:** You still test on this set, to ensure you model indeed learns. Poor performance on the training set indicates a serious flaw with the network architecture of the hyper-parameters.

Validation Set

- **Purpose:** Used *during* development of the model to tune hyperparameters (learning rate, tree depth, number of layers, regularization strength, etc.) and to make decisions like when to stop training or which model architecture to pick.
- **Key point:** The model doesn't learn from this data directly, but you *do* use it to steer decisions. Repeatedly checking validation performance and adjusting can lead to overfitting to the validation set itself.
- **Typical size:** ~10–20%.
- **Analogy:** A practice quiz; the student uses to figure out what to study more, before the real exam.

Test Set

- **Purpose:** The final, untouched check. Used *once*, at the end, to estimate how the model will perform on genuinely new, unseen data.
- **Key point:** It should not influence any modeling decisions. If you tune anything based on test performance, it stops being a fair estimate of generalization.
- **Typical size:** ~10–20%.
- **Analogy:** The actual final exam; no do-overs.

Why the distinction matters

Without separating validation from test, you risk **overfitting to your own evaluation process**. In other words, you would be tuning the model until it looks great on the data you keep checking. This does not guarantee it will perform well in the real world. The test set is your best (unbiased) proxy for real-world performance, which is why it's held back until the very end.