The Goal of the Workflow: Generalization

21.06.2025

* The single most important goal of a supervised ML model is generalization.

A real-world analogy: Studying for an exam

* Imagine you are teaching a student for a final exam.
  * You give the student a textbook, homework problems, and practice examples to study. This is the training data.
  * If you gave the student a final exam with the exact same questions from the homework, they might get 100%. But would that score tell you if they actually learned the subject? No. It would only tell you if they could memorize the specific answers you showed them.
  * To truly know if they learned, you must give them a final exam with new questions they have never seen before, but that covers the same topics. Their score on this "unseen" exam is the true measure of their knowledge and their ability to generalize what they've learned. ML is exactly the same:

Connecting the analogy to ML

* Generalization is the model's ability to make accurate predictions on new, unseen data; not just the data it was trained on.
* A model that forms well on its training data but fails on new data has only "memorized" the answers. This failure to generalize is one of the most common problems in ML, and it's called overfitting
* The entire modeling workflow is designed to prevent this. We create a fair "final exam" for our model to ensure we are accurately measuring its ability to generalize.

* The primary goal of the ML workflow is to build a model that generalizes well to new, unseen data.

Training Set and Testing Set

21.06.2025

* To accurately measure a model's ability to generalize, we split our original dataset into two independent sets:

1. The Training Set

* This is the majority of your dataset, typically 70-80% of the total data.
* This is the data that the model is allowed to "see" and learn from. The `.fit()` method is only ever called on the training set.
* This is the textbook, homework problems, and practice exams. The model analyzes the features (`X_train`) and their corresponding correct answers (`y_train`) in this set to learn the underlying patterns and relationships.

2. The Testing Set (a.k.a. Hold-out Set)

* This is the smaller portion of the dataset (the remaining 20-30%) that is "held out" and kept completely separate during the entire training and model selection process.
* The testing set is only used once, at the very end of the project, to get a final, unbiased evaluation of the champion model's performance on data it has never seen before.
* This is the final, proctored exam. The questions (`X_test`) are given to the trained model, and its predictions are compared against the official answer key (`y_test`) to generate the final reported score.
* The cardinal rule of ML: The model must never learn from the test set. Any information from the test set that influences the training process (called "data leakage") will invalidate your results and give you a false sense of confidence in your model.

* To achieve and measure the model's ability to generalize, the data is split into a training set, used exclusively for teaching the model, and a testing set, which serves as a "held-back "final exam" for an unbiased evaluation.

The Validation Set and Cross Validation

21.06.2025

* If we can't touch the test set during development, how do we tune our model's hyperparameters (like k in KNN)? How do we know k=5 is better than k=11 without getting a grade on the final exam?

The validation set

* This is the answer. This is a subset of your data that you carve from the training set. It acts as a "proxy" or "practice exam" test set that you can use during the development and tuning phase.
* If the training set is the textbook chapters and homework problems, validation set is the end-of-chapter quizzes or a practice final exam. The student takes the quizzes to see which study strategy (hyperparameters) works best. They get immediate feedback and can adjust their strategy without "using up" the real final exam.

A better approach: Cross Validation

* A single validation set is good but it has the same potential problem as a single test set: you might have gotten a "lucky" or "unlucky" split. A more robust and standard technique is K-Fold Cross Validation.
* Instead of making one static split, we systematically rotate which part of the training data is used for validation. We split the full training set into "K" sections (or "folds"), then we train and validate the model "K" times, using a different fold for validation each time, and averaging the results.

* For tuning model hyperparameters without contaminating the test set, a validation set (or more robustly cross validation).

The Core Actions of the Workflow

21.06.2025

* Once we have our data split into Training and Testing sets (and potentially a validation set), the modeling process involves three core actions. These are the "verbs" of the ML process:

1. Model Fitting

* This is the "learning" or "training" phase. It's the process where the model's algorithm iterates through the training set (X_train, y_train) to learn underlying patterns and relationships between the features and the target.
* This is the main study session. The model reads the textbook and does all the homework problems (X_train, y_train) to build its knowledge.
* The result of fitting is a "trained model" that has learned its internal parameters (e.g., the slope and intercept in a linear regression).

2. Prediction

* Once the model is trained, it can be used to make predictions on new, unseen data. This is often called "inference".
* This is like giving the student the final exam questions (X_test). They must use their learned knowledge to produce their own answers.
* You provide the fitted model with a set of features for which you do know the target and it outputs it predictions (y_pred).

* The entire process three core actions: ① Fitting the model to learn patterns, ② Using the trained model to make predictions. ③ Evaluating those predictions against the true values to measure performance

3. Evaluation

* This is the final step where we measure how well the model performed by comparing its predictions to the true, known outcomes.
* This is the grading phase. We take the student's exam answers (y_pred) and compare them to the official answer key (y_test) to calculate a final score.
* The result is one or more numerical evaluation metrics (like accuracy, R-squared, etc.) that quantify the model's performance and tell us how much we can trust it.

* This structured process ensures that the model's final evaluation score is a reliable estimate of its real-world performance.