Understanding Training, Validation, and Test Datasets in Machine Learning

Authored by

Mohammed looti

November 2, 2025

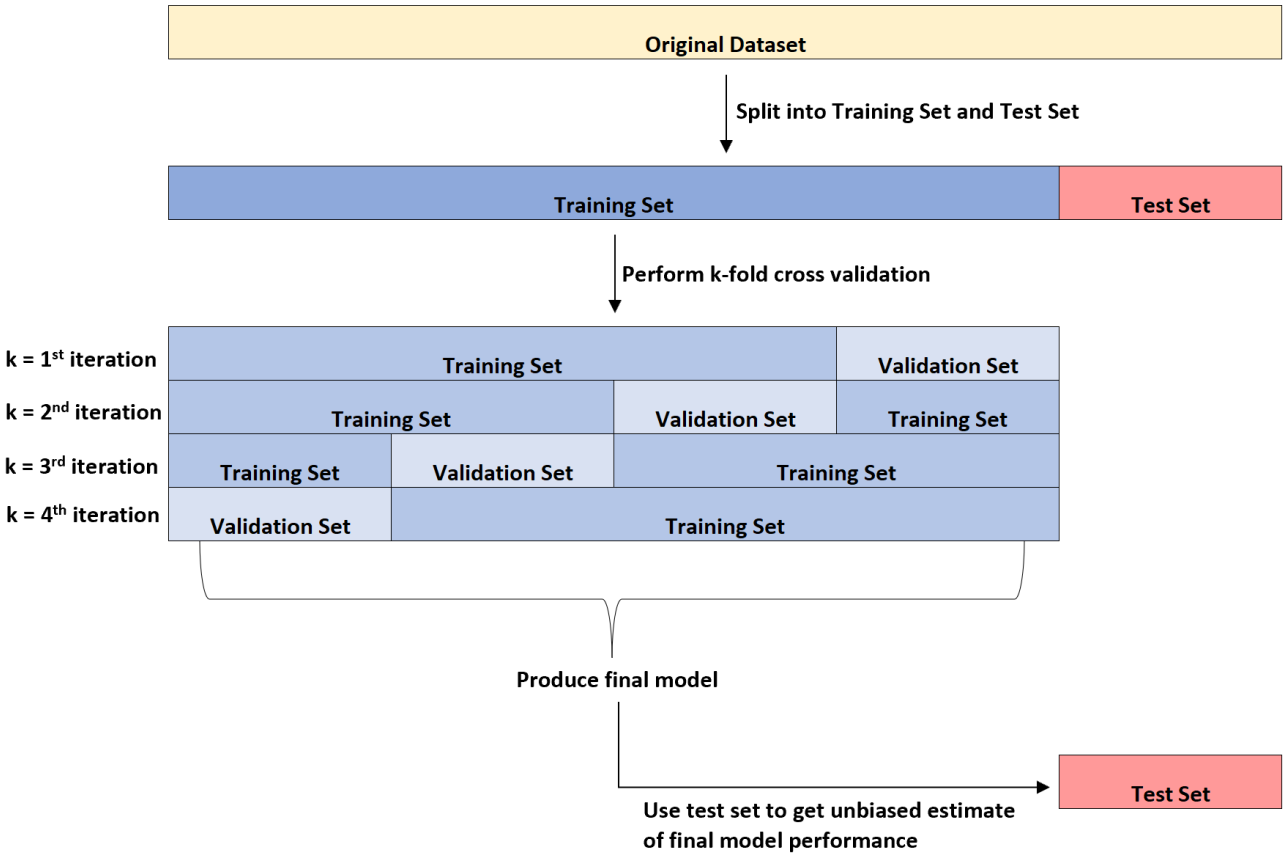
Introduction: The Necessity of Dataset Splitting in Machine Learning

In the field of data science, the development of a reliable machine learning model is fundamentally dependent on rigorous evaluation. When we set out to fit a complex algorithm to a body of data, our ultimate goal is not merely high performance on the historical data we possess, but strong generalization capabilities--the ability of the model to make accurate predictions on data it has never encountered before. Achieving this requires a strategic, methodical approach to data utilization, which necessitates splitting the available dataset into distinct, mutually exclusive partitions.

If a model is trained and evaluated using the exact same data points, it often suffers from a critical flaw known as overfitting. Overfitting occurs when the model memorizes the noise and specific anomalies within the training examples instead of learning the fundamental, underlying relationship between the features and the targets. While an overfit model might achieve near-perfect accuracy on its training set, its performance inevitably collapses when exposed to new, real-world data. To accurately diagnose and prevent this phenomenon, and to ensure the final performance metric is trustworthy, the data must be divided into three core components: the Training Set, the Validation Set, and the Test Set.

Each of these three partitions serves a unique and indispensable function in the model development lifecycle. The largest segment is dedicated to the core learning process, while the two smaller segments are dedicated to distinct phases of evaluation--one for guiding optimization and selection, and one for providing the final, unbiased assessment. Understanding the precise role and constraints of the Validation Set versus the Test Set is paramount for any practitioner seeking to build robust and deployable predictive systems. The deliberate separation of these two evaluation sets is often a point of confusion for students, but it is the cornerstone of generating reliable performance metrics.

The following diagram provides a visual explanation of how the entire dataset is typically partitioned for model development:



Deconstructing the Data Split: Training, Validation, and Test

The initial step in any supervised learning project involves defining how the available data will be allocated. This allocation dictates which data points will teach the model, which will refine it, and which will grade it. The standard allocation methodology ensures that the model is continuously challenged by new information throughout its development, mitigating the risk of reporting misleadingly optimistic performance figures. This structure separates the data into three primary roles, each interacting with the model at different stages and for different purposes, thereby preventing leakage of information and maintaining the integrity of the evaluation process.

The three components of the data split are defined by their specific utilization:

Training Set: This is the largest partition, typically comprising 60% to 80% of the total data. It is the data source used by the optimization algorithm (e.g., stochastic gradient descent) to calculate the loss function, adjust the model's internal weights and biases, and effectively learn the patterns required for prediction. The model actively modifies itself based on the errors observed on the training set.

Validation Set: This intermediate partition is reserved for frequent evaluation **during** the training process. It is used to tune the model's external settings, known as hyperparameters, and to inform

decisions regarding model selection or early stopping. The model does not directly learn from the validation set, but the researcher uses its performance metrics to guide optimization decisions.
Test Set: This set is held back until the very end of the model development pipeline. It is not used for training, nor is it used for tuning. Its sole, critical purpose is to provide a final, single, and **unbiased estimate** of the chosen model's performance on truly unseen data before deployment.

The fundamental distinction rests on usage frequency and purpose. The **validation set** is actively and iteratively used to optimize model settings and select the best candidate architecture, while the **test set** is passively reserved for one single run to provide the definitive performance guarantee. If the validation set is repeatedly used to make decisions about the model, its error rate will gradually become an optimistic assessment of the model's true capability, highlighting the necessity of the isolated test set.

The Validation Set: Tuning Hyperparameters and Mitigating Overfitting

The validation set is an indispensable tool for preventing the pitfalls of overfitting and for ensuring that the model achieves optimal generalization capacity. Its primary function is tied to the selection and tuning of hyperparameters. Hyperparameters are the configuration variables external to the data that govern the training process--examples include the learning rate, the number of epochs, the depth of a decision tree, or the regularization strength. Since these parameters cannot be learned directly from the training data, we must test various combinations and select the configuration that performs best on a separate evaluation set.

The process of using the validation set is iterative. A data scientist might train several candidate models, each employing a different set of hyperparameters (e.g., Model A with a learning rate of 0.01, Model B with 0.001, and Model C with 0.1). After each training iteration, the performance of these candidate models is measured exclusively on the validation set. The model whose performance metric (such as lowest loss or highest accuracy) is superior on the validation set is then selected as the optimal architecture. This approach ensures that the selection process is guided by performance on unseen data, rather than merely maximizing training set performance.

A second critical application of the validation set is managing the training duration via a technique known as early stopping. During training, a model's performance on the training data consistently improves. However, we often observe that performance on the validation set initially improves alongside the training set, but then begins to plateau or even degrade as the model starts to overfit the training data. The validation set acts as an early warning system; when the validation error rate stops improving for a specified number of epochs, training is halted. This prevents the model from spending excessive time learning the noise in the training data, thereby preserving its ability to generalize.

Because the validation set is used repeatedly throughout the development process to make

decisions about the model's final structure, the model is inherently biased toward performing well on this specific dataset. This bias is intentional and useful for optimization, but it means the validation error cannot be trusted as the definitive measure of performance on entirely new, future data. This is why the Test Set must remain isolated.

The Test Set: Providing an Unbiased Final Evaluation

The test set occupies the highest level of confidentiality within the machine learning workflow. It is the gold standard used to measure the true effectiveness of the finalized machine learning model. The test set must remain absolutely isolated and untouched until the very end, only after the model architecture has been chosen, the hyperparameters have been tuned using the validation set, and the final model has been trained on the combined training and validation data (or the entire primary block).

The core objective of the test set is singular: to provide an **unbiased estimate** of the model's generalization error. If the model's performance on the validation set is used for the final metric, that estimate will invariably be too optimistic due to the iterative tuning and selection process. The test set, composed of data points that have never influenced any decision about the model's configuration, guarantees that the resulting performance metric reflects the true expected error rate when the model is deployed in a production environment and encounters genuinely new data.

The integrity of the test set is so crucial that if a data scientist observes the test set performance and decides to go back and make even minor tweaks to the model or its hyperparameters, the test set immediately loses its utility as an unbiased measure. That single observation introduces data leakage, turning the test set into another iteration of a validation set. Therefore, the test set evaluation must be treated as the final, irrevocable assessment. If the performance on the test set is unacceptable, the entire development cycle must be restarted, often requiring new data or a fundamental change in modeling approach, rather than simply tweaking the current model based on the disappointing test results.

The Practical Application of K-Fold Cross-Validation

To maximize the utility of the available data and obtain a more robust estimate of performance during the optimization phase, practitioners frequently employ k-fold cross-validation. This technique is typically applied to the large block of data designated for training and validation, while the dedicated test set remains quarantined. K-fold cross-validation addresses the variability inherent in traditional single train/validation splits, where model performance can be highly sensitive to which specific data points end up in the validation set.

In k-fold cross-validation, the Training/Validation data block is segmented into 'k' equal-sized subsets, or folds. The process is then repeated 'k' times. In each iteration, one fold is temporarily

held out as the validation set, and the remaining k-1 folds are used collectively as the temporary training set. The model is trained and evaluated 'k' times, ensuring that every data point in the block serves as validation data exactly once. This rotation provides a comprehensive view of how well the model generalizes across the entire development dataset.

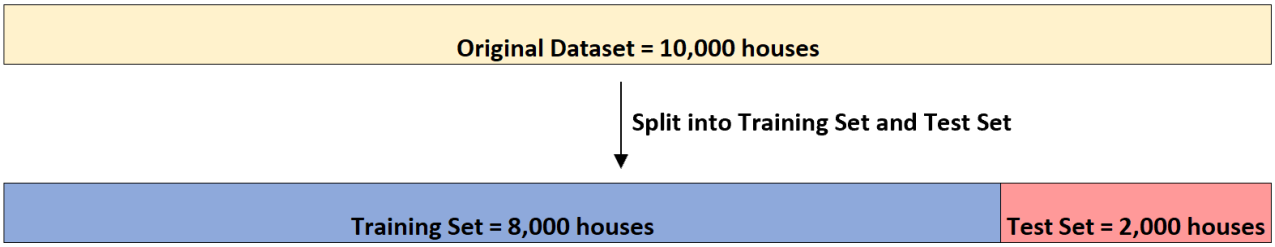
The primary benefit of k-fold cross-validation is the stabilization of the performance metric used for hyperparameter selection. Instead of relying on the error metric from a single validation split, the practitioner averages the performance across all 'k' runs. This average provides a much more reliable indicator of the optimal model configuration, reducing the chance of selecting hyperparameters that simply performed well on one specific, lucky validation split. For example, if k=10, the model selection is based on the average performance over ten different validation scenarios.

Crucially, even after the k-fold procedure identifies the best model structure, the resulting average performance metric (the cross-validation score) is still considered a biased estimate of the final generalization error, as this data was used to guide the final model selection. The true, unbiased estimate can only be obtained by applying the finalized model to the completely untouched Test Set.

Case Study: Real Estate Price Prediction Revisited

To solidify the distinction between these datasets, let us revisit the real estate prediction example. An investor uses a dataset of 10,000 houses, tracking features such as the number of bedrooms, square footage, and bathrooms, aiming to predict the selling price. The process begins with the critical separation of the data.

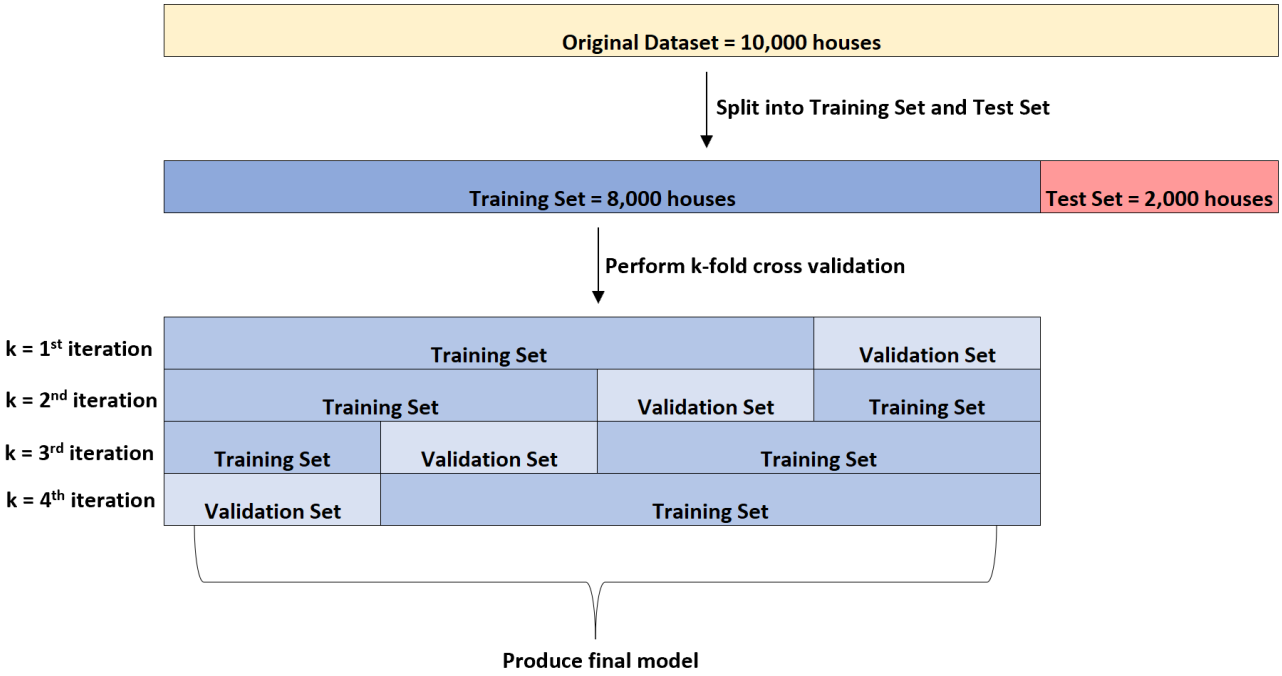
First, the investor allocates 80% (8,000 houses) to the primary Training/Validation block and reserves 20% (2,000 houses) as the final Test Set. This initial holdout is non-negotiable and ensures that 2,000 houses remain completely unseen throughout the entire hyperparameter tuning process.



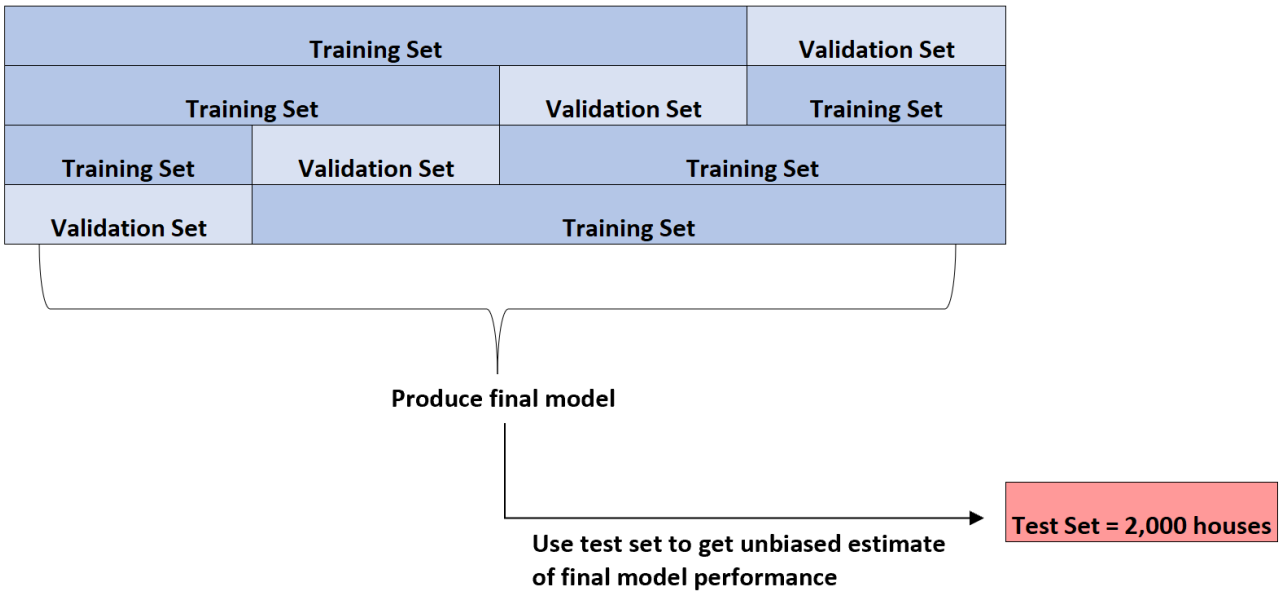
Next, the investor decides to compare several types of regression models (e.g., standard Linear Regression versus Ridge Regression) and optimize their settings using the 8,000-house block.

They employ 4-fold cross-validation. This means the 8,000 houses are split into four equal groups of 2,000 houses. In the first fold, 6,000 houses are used for training, and 2,000 houses serve as the temporary validation set. This rotation is repeated three more times, ensuring that every house in the 8,000-house block is used for validation once.

The investor performs this 4-fold CV procedure multiple times across various model types and hyperparameter settings to identify the optimal configuration--for example, the regularization strength that minimizes the average Mean Absolute Error (MAE) across the four validation folds. Suppose they identify a specific Ridge Regression configuration that results in a mean validation MAE of **$8,345**. This figure represents the estimated error based on repeated internal testing during the model selection process.



Only once the best model (the optimized Ridge Regression) is definitively selected does the investor proceed to the final evaluation. The selected model is trained one last time on the entire 8,000-house Training/Validation block using the optimal hyperparameters. This finalized model is then applied to the 2,000-house Test Set, which has been held out since the beginning. The result of this final test is the unbiased estimate of performance. If the test set evaluation yields an MAE of **$8,847**, this is the figure the investor must rely on for real-world deployment. The difference between the validation MAE ($8,345) and the test MAE ($8,847) illustrates the inherent optimism of the validation set and underscores the necessity of the isolated test set for a true generalization error rate.



Summary and Key Takeaways

The strategic differentiation between the validation set and the test set is a mandatory practice for achieving scientific rigor in machine learning. The validation set is an active, iterative tool essential for the optimization phase, guiding the search for the best hyperparameters and preventing premature overfitting. Because it participates in model selection, the performance measured on the validation set is inherently optimistic and biased.

In contrast, the test set is a passive, single-use resource that must remain isolated to provide the final, unbiased metric of model performance. This metric serves as the definitive prediction of the model's real-world error rate on truly unseen data. Adhering to the rule of absolute separation for the test set is the most critical safeguard against data leakage and the reporting of unreliable performance statistics. By correctly utilizing the training, validation, and test partitions, practitioners ensure that their finalized machine learning model is robust, generalizable, and trustworthy when deployed in production.

Additional Resources

How to Perform K-Fold Cross Validation in Python