Training Set vs Validation Set vs Test Set

This article teaches the importance of splitting a data set into training, validation and test sets.

Testing Our Model

Supervised machine learning algorithms are amazing tools capable of making predictions and classifications. However, it is important to ask yourself how accurate those predictions are. After all, it's possible that every prediction your classifier makes is actually wrong! Luckily, we can leverage the fact that supervised machine learning algorithms, by definition, have a dataset of pre-labeled datapoints. In order to test the effectiveness of your algorithm, we'll split this data into:

- **training set**
- **validation set**
- **test set**

Training Set vs Validation Set

The training set is the data that the algorithm will learn from. Learning looks different

Next Get Help

Nearest Neighbors, the points in the training set are the points that could be the neighbors.

After training using the training set, the points in the validation set are used to compute the accuracy or error of the classifier. The key insight here is that we know the true labels of every point in the validation set, but we're temporarily going to pretend like we don't. We can use every point in the validation set as input to our classifier. We'll then receive a classification for that point. We can now peek at the true label of the validation point and see whether we got it right or not. If we do this for every point in the validation set, we can compute the validation error!

Validation error might not be the only metric we're interested in. A better way of judging the effectiveness of a machine learning algorithm is to compute its precision, recall, and F1 score.

How to Split

Figuring out how much of your data should be split into your validation set is a tricky question. If your training set is too small, then your algorithm might not have enough data to effectively learn. On the other hand, if your validation set is too small, then your accuracy, precision, recall, and F1 score could have a large variance. You might happen to get a really lucky or a really unlucky split! In general, putting 80% of your data in the training set, and 20% of your data in the validation set is a good place to start.

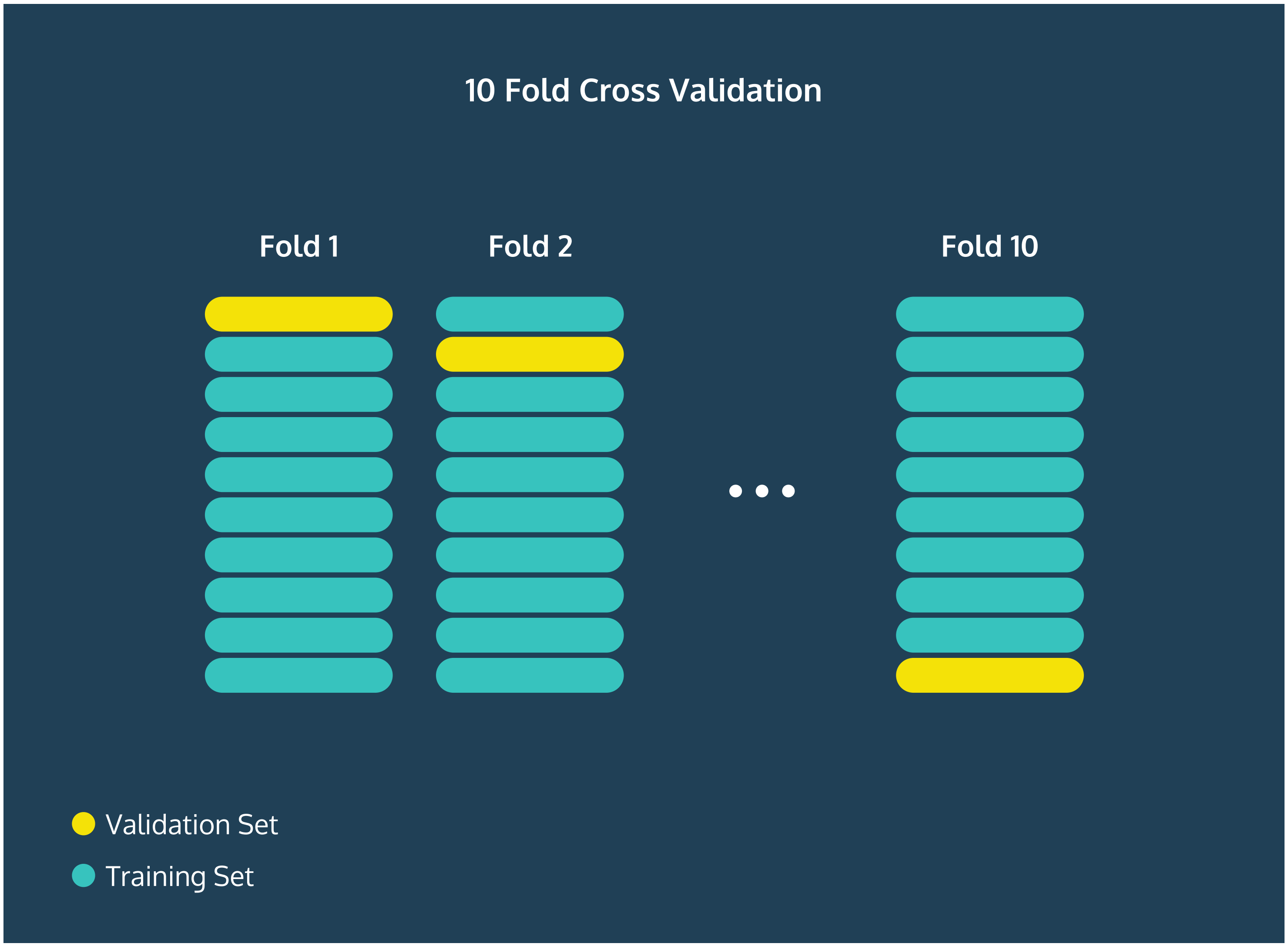
N-Fold Cross-Validation

Sometimes your dataset is so small, that splitting it 80/20 will still result in a large amount of variance. One solution to this is to perform **N-Fold Cross-Validation**. The central idea here is that we're going to do this entire process N times and average the

Next Get Help

make the validation set the second 10% of the data and calculate these statistics again. We can do this process 10 times, and every time the validation set will be a

different chunk of the data. If we then average all of the accuracies, we will have a better sense of how our model does on average.



Changing The Model / Test Set

Understanding the accuracy of your model is invaluable because you can begin to tune the parameters of your model to increase its performance. For example, in the K-

Next

Get Help

Once you’re happy with your model’s performance, it is time to introduce the test set. This is part of your data that you partitioned away at the very start of your experiment. It’s meant to be a substitute for the data in the real world that you’re actually interested in classifying. It functions very similarly to the validation set, except you never touched this data while building or tuning your model. By finding the accuracy, precision, recall, and F1 score on the test set, you get a good understanding of how well your algorithm will do in the real world.

Next Get Help