Train, Validation and Test Sets

https://machinelearningmastery.com/difference-test-validation-datasets/

A validation dataset is a sample of data **held back from training** your model that is **used to give an estimate of model skill while tuning model's hyperparameters**.

The validation dataset **is different from the test dataset** that is **also held back** from the training of the model, but is instead **used to give an unbiased estimate of the skill of the final tuned model** when comparing or selecting between final models

Generally, the term "*validation set*" is used interchangeably with the term "*test set*" and refers to a sample of the dataset held back from training the model.

The evaluation of a model skill on the training dataset would result in a biased score. Therefore the model is evaluated on the held-out sample to give an unbiased estimate of model skill. This is typically called a train-test split approach to algorithm evaluation.

> *Suppose that we would like to estimate the test error associated with fitting a particular statistical learning method on a set of observations. The validation set approach […] is a very simple strategy for this task. It involves randomly dividing the available set of observations into two parts, a training set and a validation set or hold-out set. The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set. The resulting validation set error rate — typically assessed using MSE in the case of a quantitative response—provides an estimate of the test error rate.*

— Gareth James, et al., Page 176, An Introduction to Statistical Learning: with Applications in R, 2013.

We can see the interchangeableness directly in Kuhn and Johnson's excellent text "Applied Predictive Modeling". In this example, they are clear to point out that the final model evaluation must be performed on a held out dataset that has not been used prior, either for training the model or tuning the model parameters.

> *Ideally, the model should be evaluated on samples that were not used to build or fine-tune the model, so that they provide an unbiased sense of model effectiveness. When a large amount of data is at hand, a set of samples can be set aside to evaluate the final model. The "training" data set is the general term for the samples used to create the model, while the "test" or "validation" data set is used to qualify performance.*

— Max Kuhn and Kjell Johnson, Page 67, Applied Predictive Modeling, 2013

Perhaps traditionally the dataset used to evaluate the final model performance is called the "test set". The importance of keeping the test set completely separate is reiterated by Russell and Norvig in their seminal AI textbook. They refer to using information from the test set in any way as "peeking". They suggest locking the test set away completely until all model tuning is complete.

> *Peeking is a consequence of using test-set performance to both choose a hypothesis and evaluate it. The way to avoid this is to really hold the test set out—lock it away until you are completely done with learning and simply wish to obtain an independent evaluation of the final hypothesis. (And then, if you don't like the results … you have to obtain, and lock away, a completely new test set if you want to go back and find a better hypothesis.)*

— Stuart Russell and Peter Norvig, page 709, Artificial Intelligence: A Modern Approach, 2009 (3rd edition)

Importantly, Russell and Norvig comment that the training dataset used to fit the model can be further split into a training set and a validation set, and that it is this subset of the training dataset, called the validation set, that can be used to get an early estimate of the skill of the model.

> *If the test set is locked away, but you still want to measure performance on unseen data as a way of selecting a good hypothesis, then divide the available data (without the test set) into a training set and a validation set.*

— Stuart Russell and Peter Norvig, page 709, Artificial Intelligence: A Modern Approach, 2009 (3rd edition)

This definition of validation set is corroborated by other seminal texts in the field. A good (and older) example is the glossary of terms in Ripley's book "Pattern Recognition and Neural Networks." Specifically, training, validation, and test sets are defined as follows:

> *– Training set: A set of examples used for learning, that is to fit the parameters of the classifier.*
>
> *– Validation set: A set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network.*
>
> *– Test set: A set of examples used only to assess the performance of a fully-specified classifier.*

— Brian Ripley, page 354, Pattern Recognition and Neural Networks, 1996

A good example that these definitions are canonical is their reiteration in the famous Neural Network FAQ. In addition to reiterating Ripley's glossary definitions, it goes on to discuss the common misuse of the terms "test set" and "validation set" in applied machine learning.

> *The literature on machine learning often reverses the meaning of "validation" and "test" sets. This is the most blatant example of the terminological confusion that pervades artificial intelligence research.*
>
> *The crucial point is that a test set, by the standard definition in the NN [neural net] literature, is never used to choose among two or more networks, so that the error on the test set provides an unbiased estimate of the generalization error (assuming that the test set is representative of the population, etc.).*

We can make this concrete with a pseudocode sketch:

```
# split data
data = ...
train, validation, test = split(data)

# tune model hyperparameters
parameters = ...
for params in parameters:
    model = fit(train, params)
    skill = evaluate(model, validation)

# evaluate final model for comparison with other models
model = fit(train)
skill = evaluate(model, test)
```

Cross-validation:

In their book, Kuhn and Johnson have a section titled “Data Splitting Recommendations” in which they layout the limitations of using a sole “test set” (or validation set):

> *As previously discussed, there is a strong technical case to be made against a single, independent test set:*
>
> *– A test set is a single evaluation of the model and has limited ability to characterize the uncertainty in the results.*
> *– Proportionally large test sets divide the data in a way that increases bias in the performance estimates.*
> *– With small sample sizes:*
> *– The model may need every possible data point to adequately determine model values.*
> *– The uncertainty of the test set can be considerably large to the point where different test sets may produce very different results.*
> *– Resampling methods can produce reasonable predictions of how well the model will perform on future samples.*
>
> — Max Kuhn and Kjell Johnson, Page 78, Applied Predictive Modeling, 2013

```
# split data
data = ...
train, test = split(data)

# tune model hyperparameters
parameters = ...
k = ...
for params in parameters:
    skills = list()
    for i in k:
        fold_train, fold_val = cv_split(i, k, train)
        model = fit(fold_train, params)
        skill_estimate = evaluate(model, fold_val)
        skills.append(skill_estimate)
    skill = summarize(skills)

# evaluate final model for comparison with other models
model = fit(train)
skill = evaluate(model, test)
```