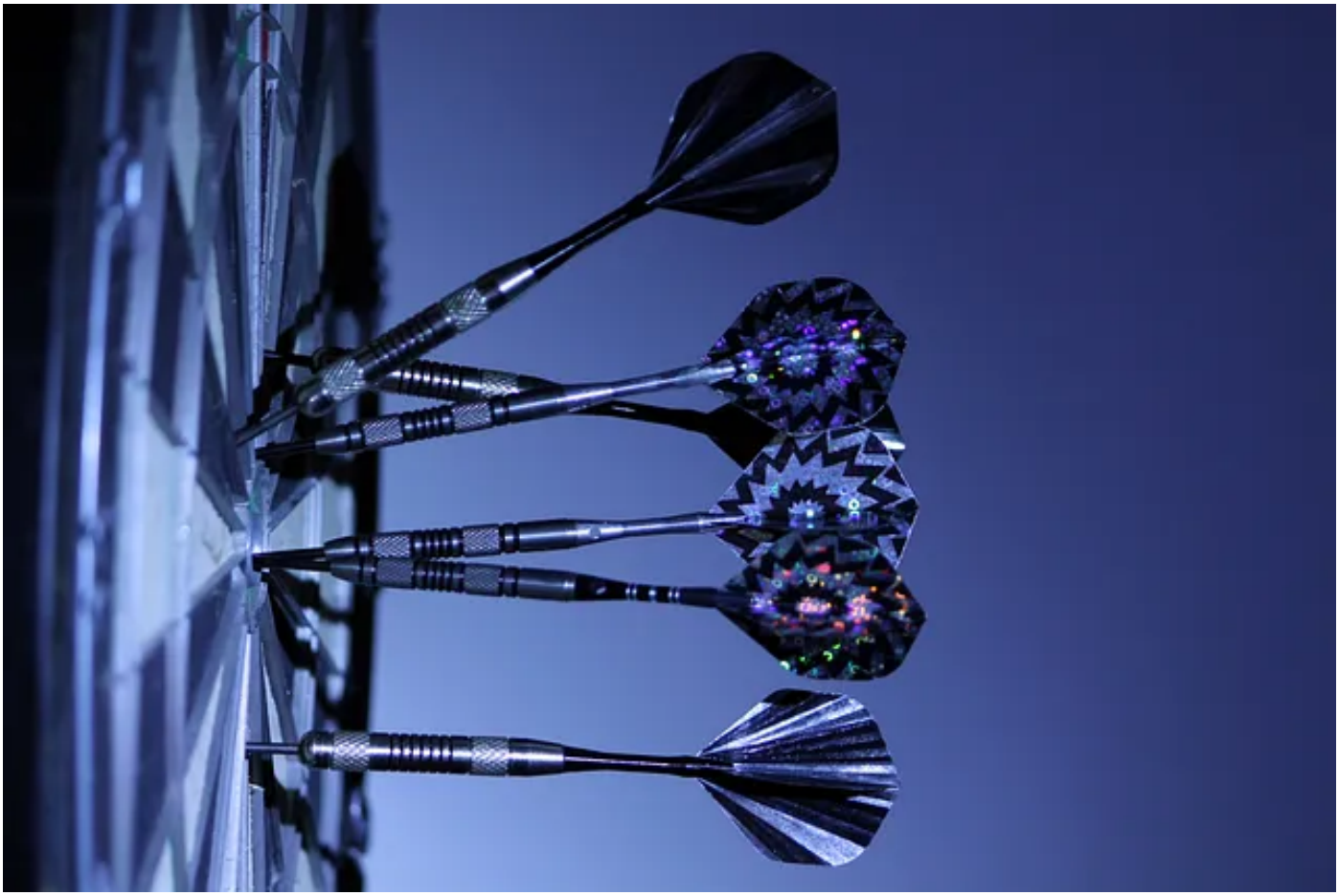
About Train, Validation and Test Sets in Machine Learning

Tarang Shah



This is aimed to be a short primer for anyone who needs to know the difference between the various dataset splits while training Machine Learning models.

For this article, I would quote the base definitions from Jason Brownlee's excellent article on the same topic, it is quite comprehensive, do check it out for more details.

Training Dataset

> ***Training Dataset****: The sample of data used to fit the model.*

The actual dataset that we use to train the model (weights and biases in the case of a Neural Network). The model *sees* and *learns* from this data.

Validation Dataset

> ***Validation Dataset****: The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.*

The validation set is used to evaluate a given model, but this is for frequent evaluation. We, as machine learning engineers, use this data to fine-tune the model hyperparameters. Hence the model occasionally *sees* this data, but never does it "*Learn*" from this. We use the validation set results, and update higher level hyperparameters. So the validation set affects a model, but only indirectly. The validation set is also known as the Dev set or the Development set. This makes sense since this dataset helps during the "development" stage of the model.

Test Dataset

> ***Test Dataset****: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.*

The Test dataset provides the gold standard used to evaluate the model. It is only used once a model is completely trained(using the train and validation

sets). The test set is generally what is used to evaluate competing models (For example on many Kaggle competitions, the validation set is released initially along with the training set and the actual test set is only released when the competition is about to close, and it is the result of the the model on the Test set that decides the winner). Many a times the validation set is used as the test set, but it is not good practice. The test set is generally well curated. It contains carefully sampled data that spans the various classes that the model would face, when used in the real world.

A visualization of the splits

About the dataset split ratio

Now that you know what these datasets do, you might be looking for recommendations on how to split your dataset into Train, Validation and Test sets.

This mainly depends on 2 things. First, the total number of samples in your data and second, on the actual model you are training.

Some models need substantial data to train upon, so in this case you would optimize for the larger training sets. Models with very few hyperparameters will be easy to validate and tune, so you can probably reduce the size of your validation set, but if your model has many hyperparameters, you would want to have a large validation set as well(although you should also consider cross validation). Also, if you happen to have a model with no hyperparameters or

ones that cannot be easily tuned, you probably don't need a validation set too!

All in all, like many other things in machine learning, the train-test-validation split ratio is also quite specific to your use case and it gets easier to make judge ment as you train and build more and more models.

Note on Cross Validation: Many a times, people first split their dataset into 2 — Train and Test. After this, they keep aside the Test set, and randomly choose X% of their Train dataset to be the actual ***Train*** *set and the remaining (100-X)% to be the* ***Validation*** *set, where X is a fixed number(say 80%), the model is then iteratively trained and validated on these different sets. There are multiple ways to do this, and is commonly known as Cross Validation. Basically you use your training set to generate multiple splits of the Train and Validation sets. Cross validation avoids over fitting and is getting more and more popular, with K-fold Cross Validation being the most popular method of cross validation. Check this out for more.*

Let me know in the comments if you want to discuss any of this further. I'm also a learner like many of you, but I'll sure try to help whatever little way I can 🙂

Originally found at
http://tarangshah.com/blog/2017-12-03/train-validation-and-test-sets/