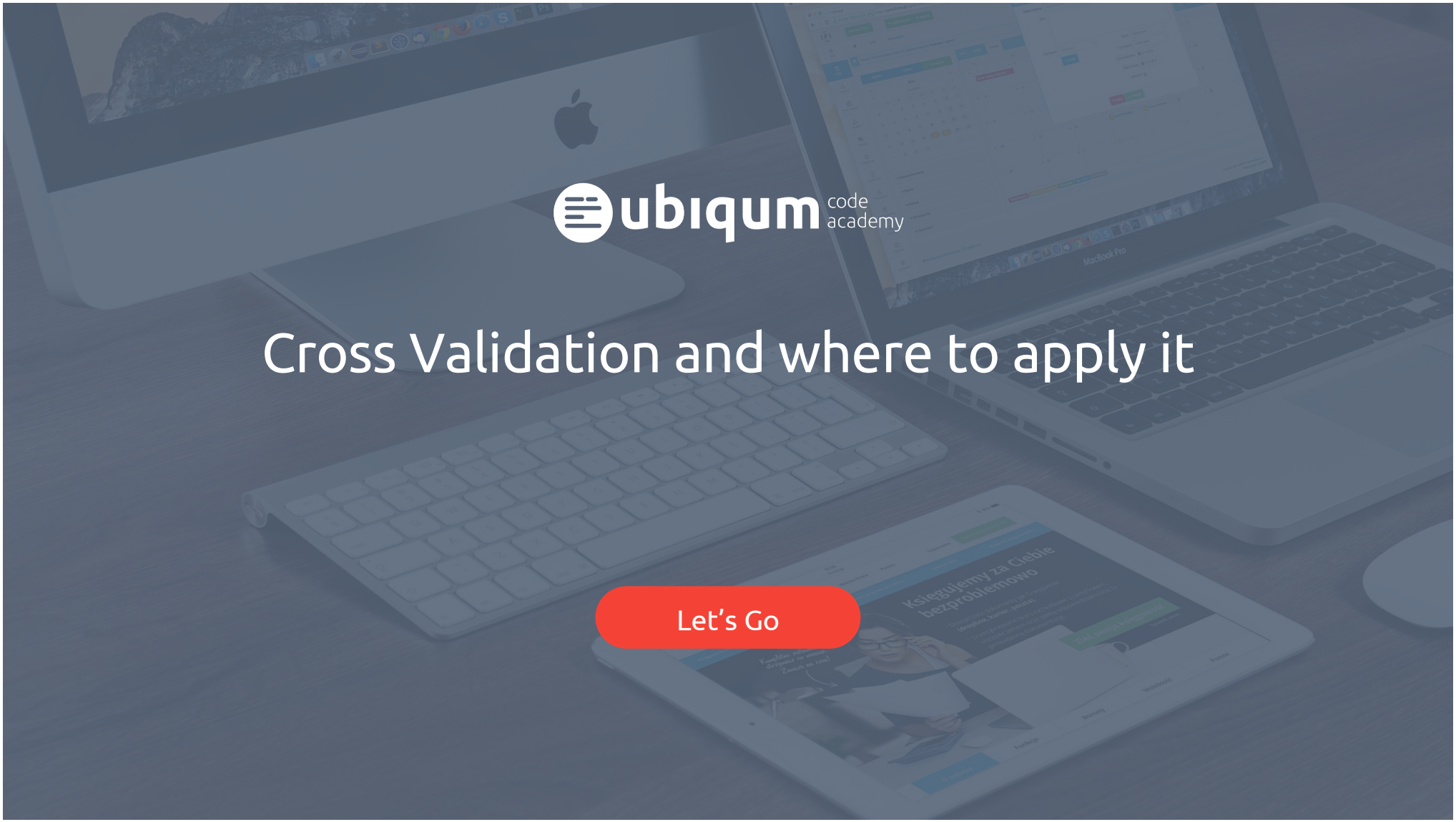

ubiqum code academy
Cross Validation and where to apply it
Let's Go

Major steps to build and employ a ML model

1. Choose an algorithm
2. Fit the model
3. Validate the model (using **CV**)
4. Tune the hyperparameters (using **CV**)
   - Grid Search
   - Random Search
5. Make predictions

Validate the Model. Why?

You can't just fit the model to your training data and hope it will work for yet-unseen data.

You need to make sure that your model has got most of the patterns from the data correct, and It's not picking up too much noise. In other words, **it has to be low in bias and variance**.

Why do Bias and Variance need to be as low as possible?

Because they are they only two somehow-controllable components that make up the prediction error, whose level must be kept as low as possible in order to achieve the best possible prediction performance.

$$\text{Prediction Error} = \text{Bias} + \text{Variance} + \text{Irreducible Error}$$

The “No Free Lunch theorem”

The “no free lunch” (NFL) theorem for supervised machine learning implies that **no single machine learning algorithm is universally the best-performing algorithm for all problems**.

In other words, all algorithms have the same error rate when averaged over all possible data generating distributions. Therefore, a certain algorithm must have a certain bias towards certain distributions and functions to be better at modelling those distributions, which would at the same time render it worse at modelling other types of distributions.

Assumptions

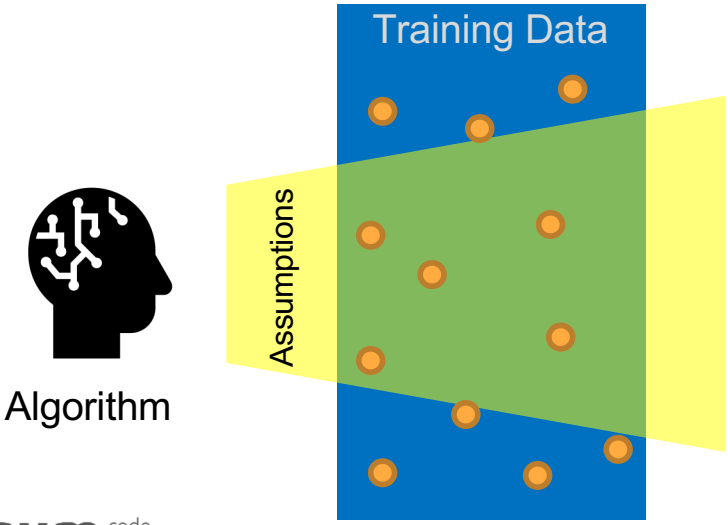
Where are we?

12020{ }52231717513

The text in red is a question and the line below is the answer to it. With no other instructions, any attempt in trying to solve this sort of riddle would involve you making some assumptions.
This is, more or less, what happens when we fit a model with our training data. The algorithm makes some assumptions based on which it estimates the target function.

Bias

As any assumption inevitably narrows the scope of what can be learned from the training set, BIAS can be thought of as the portion of the predictions that will be systematically wrong due to the specific assumptions made by the algorithm.



The dots in the blue area represents the portion of the data which the algorithm did not learn from, due to the limitations imposed by its own assumptions.

Bias

Generally, linear algorithms have a high bias, making them fast to learn and easier to understand, but generally less flexible. In turn, they have lower predictive performance on complex problems that fail to meet the simplifying assumptions of the algorithms bias.

Low Bias: Suggests less assumptions about the form of the target function.
High-Bias: Suggests more assumptions (or too simplifying assumptions) about the form of the target function.

Variance

Variance refers to the sensitivity of the learning algorithm to the specifics of the training data. This means that different training set will help build different models, which will end up making different predictions. Therefore, as Variance is an unavoidable element of any ML model, its presence, per se, should not be thought as an issue.

High level of Variance, however, should be avoided as it would indicate that:

- the algorithm is modelling the random noise in the training data, rather than the intended outputs (overfitting)
- the noise in the training data is largely influencing the number and types of the parameters use to characterize the mapping function.

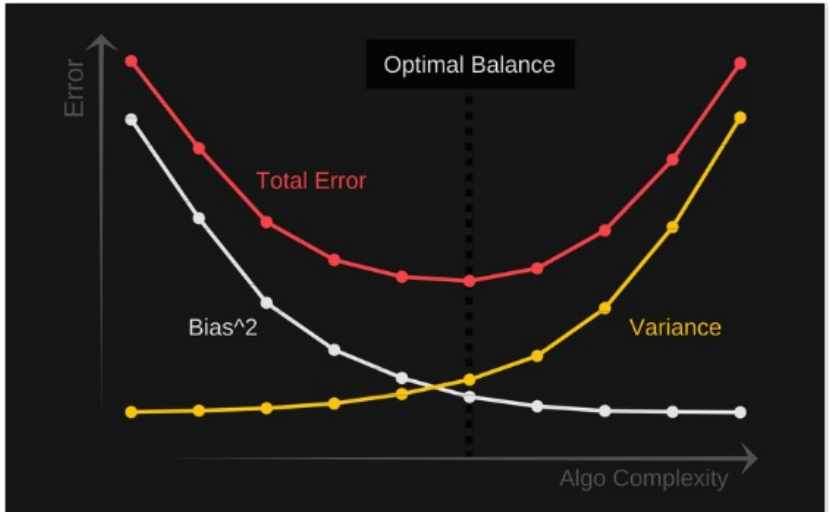
Variance

- **Low Variance**: Suggests that changes in the dataset will lead to small changes to the estimate of the target function
- **High Variance**: Suggests that changes in the dataset will lead to large changes to the estimate of the target function

Generally, nonlinear machine learning algorithms that have a lot of flexibility have a high variance. For example, decision trees have a high variance, that is even higher if the trees are not pruned before use.

Bias-Variance trade off

If a model is too simple and has very few parameters, then it may have high bias and low variance. Conversely, if it has a large number of parameters then it's going to have high variance and low bias. The key is to find the right balance without overfitting or underfitting the data.



Bias-Variance trade off. How do we handle it?

The bias-variance trade-off is a conceptual tool to think about these sources of error and how they should be kept in balance, at all times, to achieve the best possible results.

The tuning of the hyperparameters of machine learning algorithms often represents the place where the battle to balance out bias and variance takes place.

Example 1: the support vector machine algorithm has low bias and high variance, but the trade-off can be changed by increasing the C parameter that influences the number of violations of the margin allowed in the training data, which increases the bias but decreases the variance.

Example 2: in the *k*-nearest neighbours, a small *k* results in predictions with high variance and low bias. Conversely, a large *k* results in predictions with a small variance and a large bias.

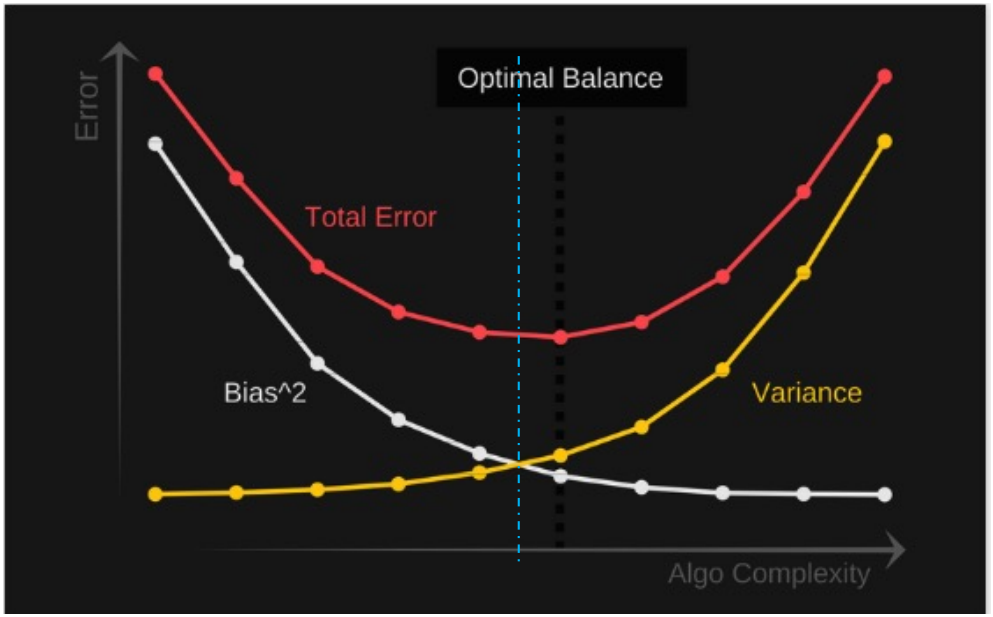
Bias-Variance trade off in Linear Algorithms

Linear ML algorithms often have low variance but high bias. Some examples are:

- Linear Regression
- Linear Discriminant Analysis
- Logistic Regression

By default, before any hyper-parameters tuning, the total error of a model based on one of the algorithms above should be represented by a point somewhere on the left side of the Total Error curve.

In such situation, you should try to move down the curve, towards the dashed blue line, by reducing the BIAS and increasing the VARIANCE.



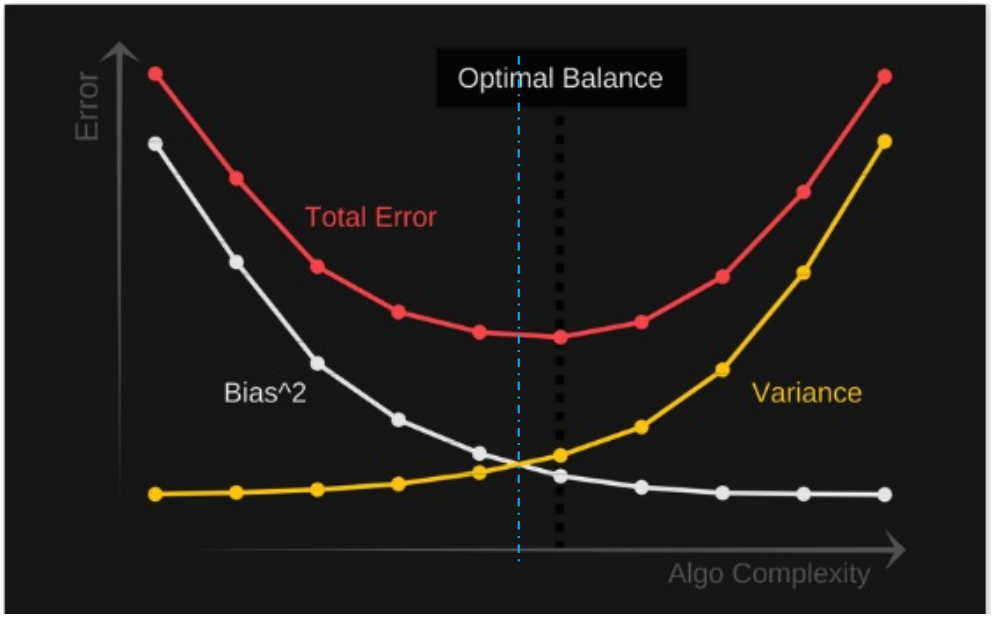
Bias-Variance trade off in non-linear algorithms

Non-linear ML algorithms often have a low bias but a high variance. Some examples are:

- Decision Trees
- k-Nearest Neighbours
- Support Vector Machines.

By default, before any hyper-parameters tuning, the total error of a model based on one of the algorithms above should be represented by a point somewhere on the **right** side of the Total Error curve.

In such situation, you should try to move down the curve, towards the dashed blue line, by reducing the VARIANCE and increasing the BIAS.



Resources:

- https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f
- https://towardsdatascience.com/cross-validation-and-hyperparameter-tuning-how-to-optimise-your-machine-learning-model-13f005af9d7d
- https://medium.com/@mandava807/cross-validation-and-hyperparameter-tuning-in-python-65cfb80ee485
- https://machinelearningmastery.com/how-to-reduce-model-variance/
- https://becominghuman.ai/machine-learning-bias-vs-variance-641f924e6c57

Thanks