OVERFITTING & UNDERFITTING

Train-Test split

Usually while building a Machine Learning model, you would split your data into (1) Train, Test and Validation datasets or simply (2) Train using Cross-validation and Test sets.

Note that the purpose of Validation set in (1) and Test set in (2) is to give an unbiased estimate of the skill of the final tuned model on unseen data i.e. the data the model has never seen before

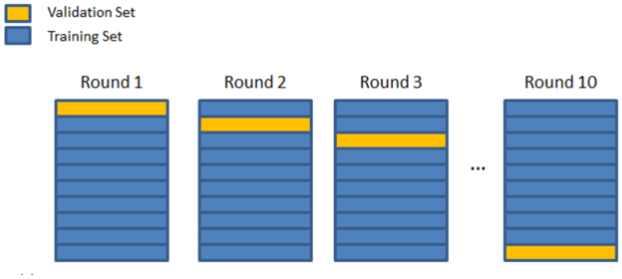
In supervised learning **Training** a model simply means **learning** (determining) good values for all the weights and the bias from labeled examples.

The **training** data is used to make sure the **machine** recognizes patterns in the data, the cross-validation data (or the test data in (1)) is used to ensure better accuracy and efficiency of the algorithm used to **train** the **machine**.

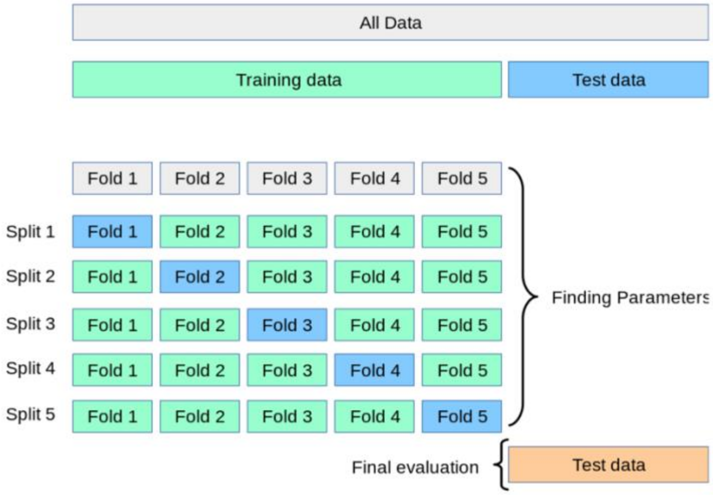
What is Cross Validation?

Cross-validation is a technique used to protect against overfitting in a predictive model, particularly in a case where the amount of data may be limited. In **cross-validation**, **you** make a fixed number of folds (or partitions) of the data, run the analysis on each fold, and then average the overall error estimate.There are different types of Cross Validation Techniques but the overall concept remains the same:

- To partition the data into a number of subsets
- Hold out a set at a time and train the model on remaining set
- Test model on hold out set
- Repeat the process for each subset of the dataset



k-Fold Cross Validation



What is Stratification?

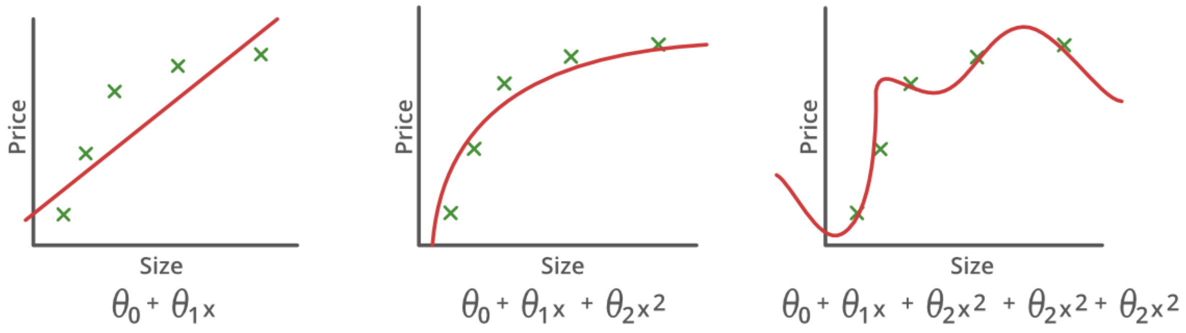
Stratification is the process of rearranging the data as to ensure each fold is a good representative of the whole. For example in a binary classification problem where each class comprises 50% of the data, it is best to arrange the data such that in every fold, each class comprises around half the instances.

Stratified sampling aims at splitting a data set so that each split is similar with respect to something. In a classification setting, it is often chosen to ensure that the train and test sets have approximately the same percentage of samples of each target class as the complete set

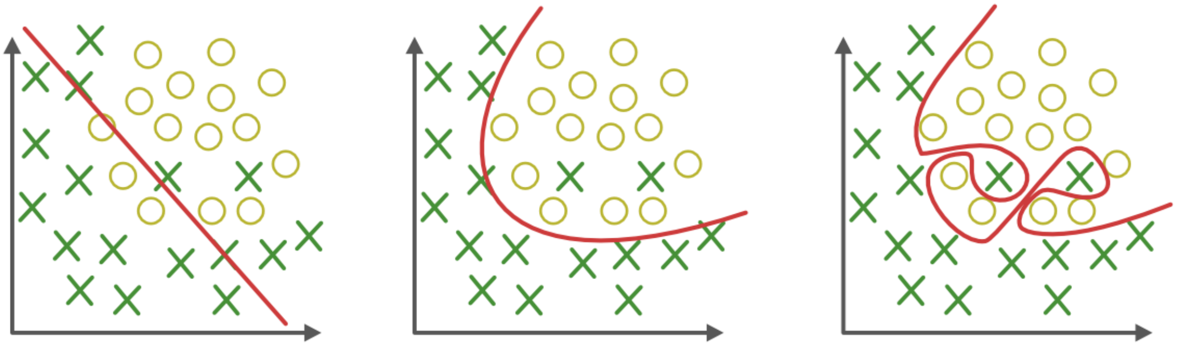
If you want to apply stratification to Cross validation check stratified k-fold cross validation.

What is underfitting and overfitting?

[[Citation: **Thanks to Andrew NG for these great images**]]



For the Regression model above, image (1) Underfit (2) Just the right fit (3) Overfit



Similarly, for the Binary classification model above, image (1) Underfit (2) Just the right fit (3) Overfit

Overfitting is a modeling error which occurs when a function is too closely fit to a limited set of data points. **Underfitting** refers to a model that can neither model the training data nor generalize to new data.

Some ways of tacking overfitting (or High Variance):

(1) Get more data
(2) Regularization
(3) Simplify the model by selecting fewer features

Some ways of tackling underfitting (or High Bias)

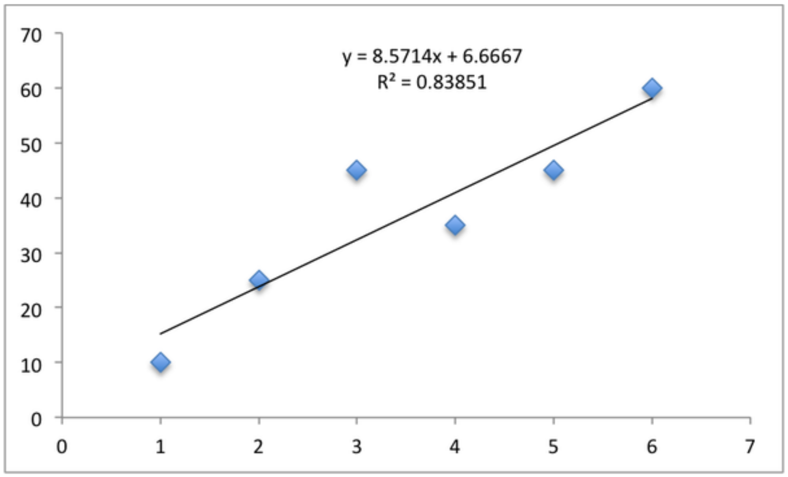
(1) Selecting more powerful model, with more parameters
(2) Improved feature engineering
(3) Reducing constraints on the model (for example reducing regularization)
(4) Note that getting more data might not help (it depends) here, as you might have lot of data and say for example you are fitting a Linear Regression where the data is highly non-linear which would result in underfitting.

A great answer on the intuition of Overfitting on Quora, answered by Jessica Su:

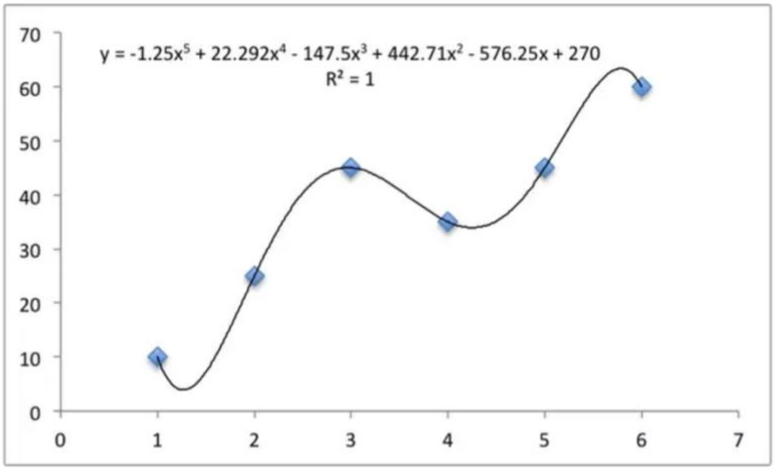
https://www.quora.com/What-is-an-intuitive-explanation-of-over-fitting-particularly-with-a-small-sample-set-What-are-you-essentially-doing-by-over-fitting-How-does-the-over-promise-of-a-high-R%C2%B2-low-standard-error-occur

Overfitting is when you're trying to model how a system works, and you build the model using some data. But then you realize your model was so tuned to the quirks of your data that it didn't really explain anything about the system.

Suppose you have some data that's linearly related, but has a little bit of noise in it. You try to fit your sample to a linear model, and see that it doesn't match exactly right:

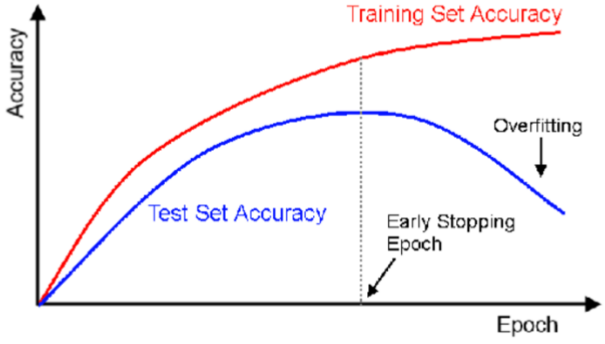


But you really want to match each datapoint perfectly, so you end up with a model like this:



This model performs great on the data you have, but when you ask it to make predictions about the rest of the data, it falls flat on its face. That's because you made your model so flexible that it tuned itself to the random noise in your dataset. It would have been better to take the linear model, which didn't perform as well on your small sample but would have made better predictions on new data points.

Overfitting detection: If your model has high accuracy on training dataset but poor validation accuracy, its overfitting.



Underfitting:

Cause: Your model has too few parameters for the machine to generalize. For example, your machine was to predict if a certain object is a ball, but your model has only one parameter – balls are round. This is an underfitting scenario where your machine would wrongly generalize all round objects into balls. It is highly biased towards that single parameter.

Underfitting Detection: If your machine performs poorly on both training data and unseen data, i.e. it has poor accuracy and validation accuracy, the machine's underfitting.

References:

Scikit-learn Cross Validation: https://scikit-learn.org/stable/modules/cross_validation.html

Overfitting on Quora, answered by Jessica Su: https://www.quora.com/What-is-an-intuitive-explanation-of-over-fitting-particularly-with-a-small-sample-set-What-are-you-essentially-doing-by-over-fitting-How-does-the-over-promise-of-a-high-R%C2%B2-low-standard-error-occur

Andrew NG: for the great overfitting/underfitting diagrams