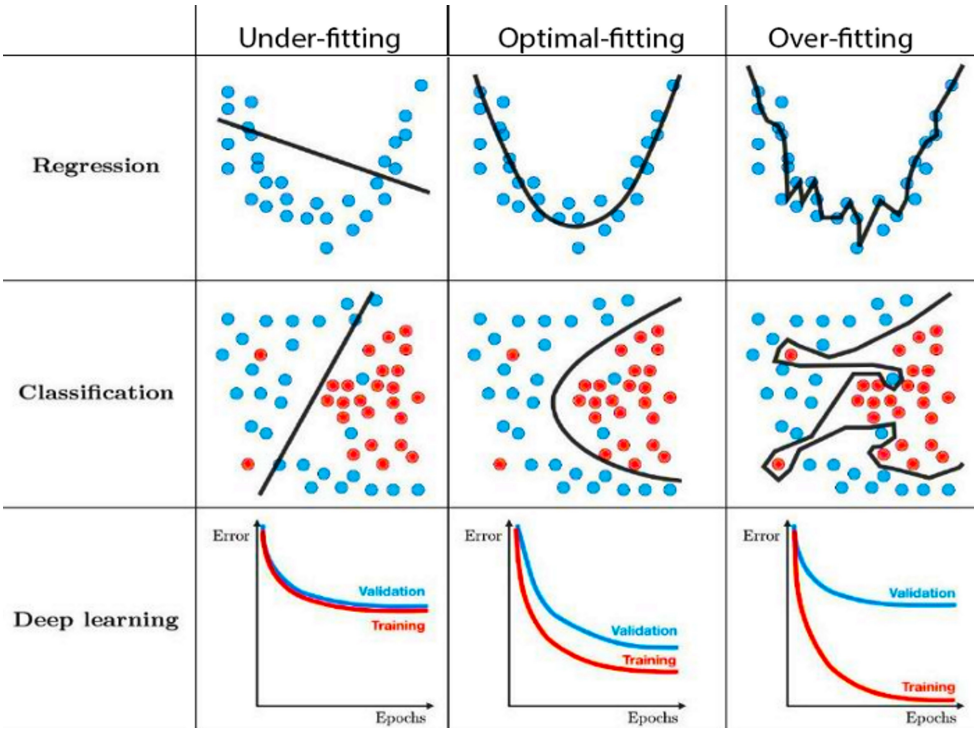
Underfitting & Overfitting

Tuesday, July 27, 2021 8:25 PM



Variance :

- High Variance means small change in training data cause high change in model, it is trying to overfit in data. Model becomes very complex. We have to choose model neither too complex nor too simple. (According to Occam's Razor).

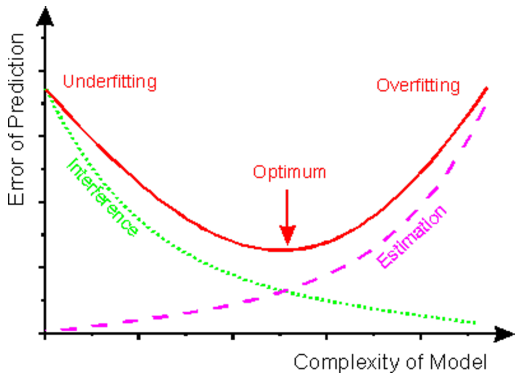
Bias :

- If you take unnecessary simplified assumptions for training data, we get high bias.
- In KNN, if we take very high K value then the algorithm gives value of only that class label which is high.

Underfitting : It have high Train Error/ Misclassifications as well as High Test Error / Misclassifications. We are using simpler model which is unable to good fit/ separate the data.

Overfitting : It gets very low Training Error/ Misclassifications but high Test Error / Misclassifications. We are using very complex model which try to fit/separate every point in data.

We need to find that optimum position where we don’t have high bias and high variance .



Regularization :

It is used to minimize the effect of Overfitting and reduce variance in data.

Two types of Regularizer :

- Lasso Regularizer : used Manhattan Distance(L1)

  w* = $arg_{m,c}$ **min {** $(y_{actual} - Y_{predicted})^2 + \lambda \sum_i^d |w_i|$ **}**

- Ridge Regularizer : used Euclidean Distance (L2)

  w* = $arg_{m,c}$ **min {** $(y_{actual} - Y_{predicted})^2 + \lambda \sum_i^d (w_i)^2$ **}**

In Sklearn Library, LinearRegression() doesn't contain regularizer, we have to import it separately. While LogisticRegression() it contains inside class already.

In Decision Tree for Overfitting :

a) Pruning : In Pruning, they try to overfit the data as good as possible and then cut the branches and transfer the class labels to lower branches.

b) Tune the depth and terminal nodes

c) Use Random Forest.

Lasso vs Ridge Regularizer :

Lasso uses Absolute Value which take more computational resources than Ridge. But Lasso do Feature Selection internally by make coefficient = 0 of unnecessary Features.

Parameters and Hyperparameters :

Parameters are outputs of your Algorithm while Hyperparameters are used to tune or set the Algorithm while Modelling.

Hyperparameters for different Algorithms:

- Linear / Logistic Regression : $\lambda = 0$, then function will overfit

$\lambda \uparrow$	Underfitting	$\lambda \downarrow$	Overfitting

We have to keep λ not too low and too high.

- Support Vector Machine : If c = 0 , the Soft Margin will look like Hard Margin and it will take unnecessary simple assumption of linearly classifiable data. So the model will be underfitted.

C ↑	Overfit	C ↓	Underfit

We have to take C in good range.

RBF Kernel = $-\exp\{-\frac{|x_i - x_j|}{2\sigma^2}\}$ and $\gamma = \frac{1}{2\sigma^2}$ if γ = 0 then RBF kernel become 0

And it will Linear SVM.

$\gamma\downarrow$	Underfit	$\gamma\uparrow$	Overfit

- KNN : If K = 1 , it will take only one neighbor and Overfit.

K↓	Overfit	K↑	Underfit

P = {1,2, … } , Minkowski metric, we have to do experimentation

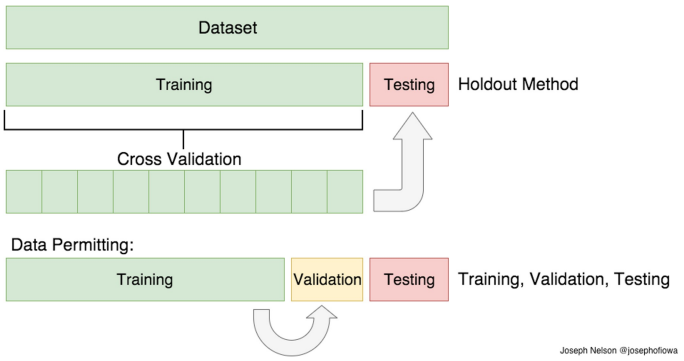
- Decision Tree : If Depth increase a lot, it causes overfitting and decreases a lot means Underfitting.

Depth↑	Overfit	Depth↓	Underfit

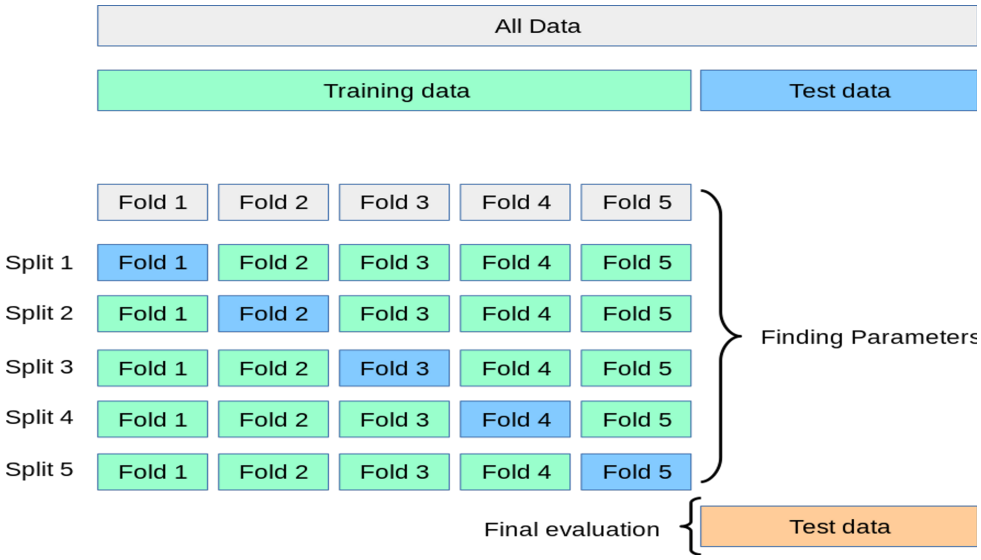
If No of leaf increases a lot, it causes underfitting and vice versa.

- Gradient Descent : η

Cross Validation :



Data is split into :- Train, Validation and Test . Different Hyperparameters are used in training of Train data are validated by Validation Data. The Model which gives best accuracy / R2 score is then choose to use on Test Data for Final Testing.
But splitting of data causes loss of data which results in improper training of Model, That's where K-fold Cross Validation came in.



In Cross Validation, Training data is split into different subset. Take each subset at a time as a Validation data and train the rest data. Compare the trained data with Validation data to choose the best Hyperparameters for Model.

Here for example we use KNN, K = {3,5,7,9,13} and P = {1,2,3}
Then we get 5*3*5 = 75 Models

Best Model will test with Testing Data.

K - fold Cross Validation :

- Grid Search CV : It uses every combination of Hyperparameters to check the best Model. It uses more computational resources.
- Randomized Search CV : It take some random pairs and perform CV. It is little fast.