

[← Go Back to Machine Learning](#)

Course Content

Summary: GridsearchCV

In the previous lecture video, we have already learned about what is **Pre-Pruning using gridsearchCV** and why we need to prune our Decision-Tree models. Let's have a look at the key-take aways -

GridSearch

Grid search is a process that identifies the best combination of the hyperparameters from a predefined set of values.

- In Grid search, we create a parameter space (also known as 'search space') using a certain number of hyperparameters
- These hyperparameters are associated with the model and can take fixed values like int, float, or boolean within a specified range.

For example, **max_depth** defines the maximum depth of the tree and it can take **int values only**. If we provide `max_depth = 3.5` or `max_depth = True`, it will make no sense.

Let's see what a parameter space looks like -

```
param_grid = {"criterion" : [ "Gini", "Entropy"],  
             "max_depth" : [4, 7] }
```

Here *criterion* and *max_depth* are the hyperparameters. For each of them, we have given 2 values (*it is not necessary to have a fixed number of parameters*). From this set of hyperparameters, we can get $(2*2) = 4$ combinations of them -

(criterion = Gini, max_depth = 4), (criterion = Gini, max_depth = 7), (criterion = Entropy, max_depth = 4), (criterion = Entropy, max_depth = 7)

Now, the Grid search will check each combination of the hyperparameters and try to find the best combination among them.

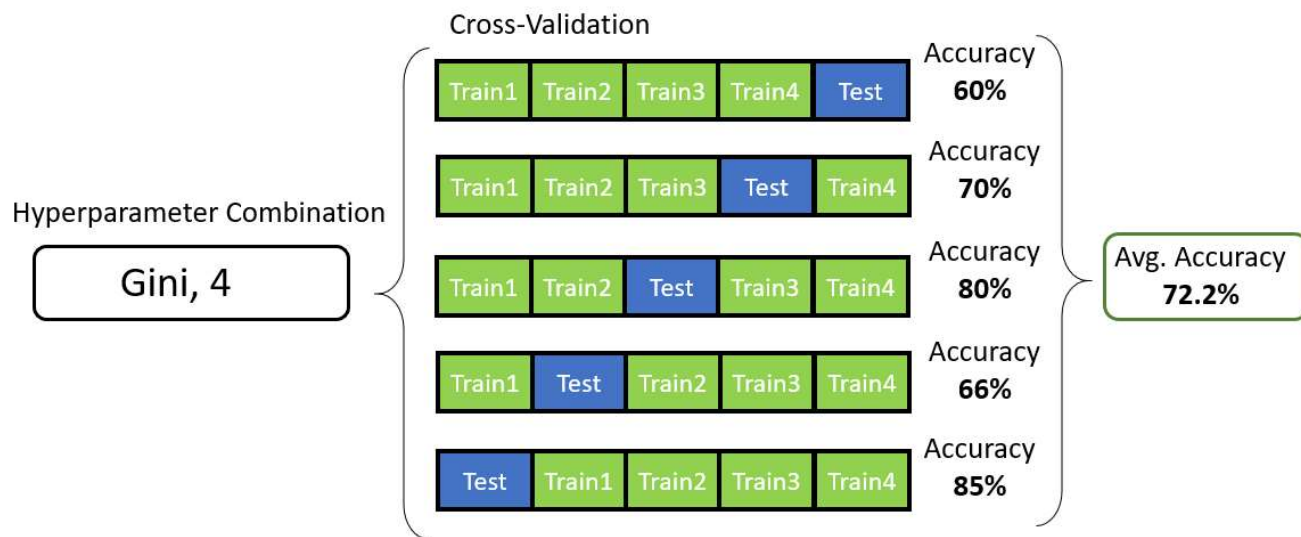
How do we decide which is the best combination of the hyperparameters?

The best combination of hyperparameters is chosen simply based on the performance - which combination of the parameter gives the best performance. Let's see how this performance is obtained -

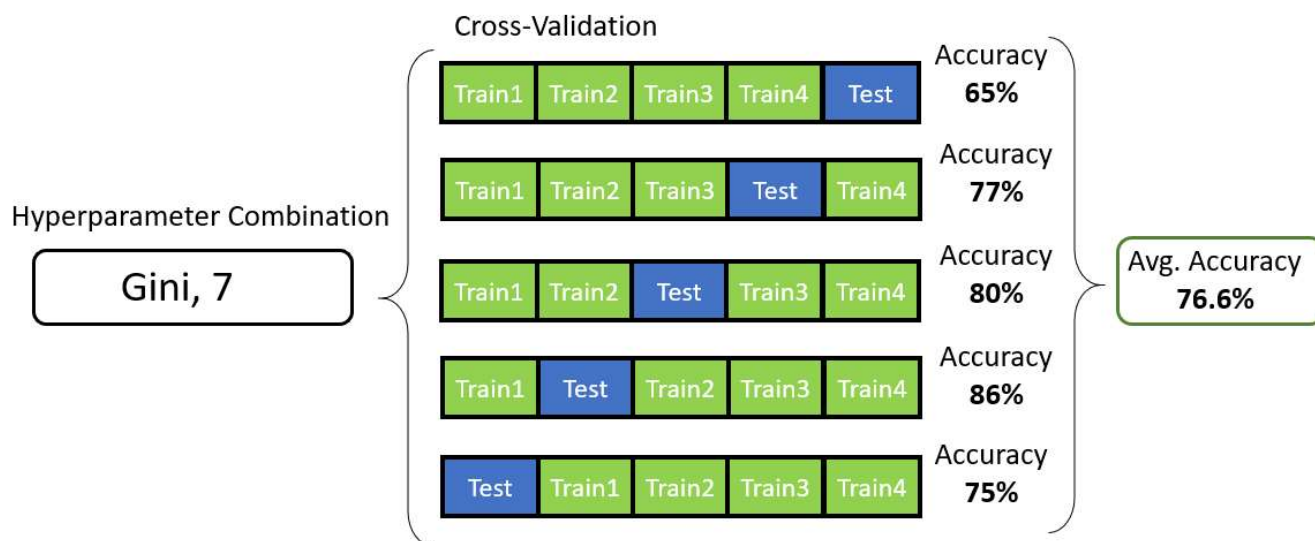
We know that in Cross-validation, the data will be divided into folds and each fold will be used for training and testing. Suppose, in our case, $CV = 5$.

Therefore, the data will be split so that each iteration uses 4 folds for training and 1 fold for testing for each combination of the hyperparameters.

For example, for the first combination -



Similarly for the second combination -



and so on.

In this way, the average performance of all possible hyperparameter combinations is assessed, and the combination that results in the best performance is chosen.

Thus, using pre-pruning, we prevent the tree from growing any further and minimize the chance of overfitting.

Proprietary content.©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.