

* Hyperparameter Tuning *

• Hyperparameters:- Params explicitly defined by user to control learning process.

↳ Adjust hyperparameters in ML algo that are not learned during training, but can be set before training begins.

↳ Used to improve model learning & values set before starting learning process of model.

↳ Values cannot be changed during training process.

↳ 2 best strategies for tuning hyperparameters for ML models:-

- ① Grid Search CV
- ② Randomized Search CV

1) GridsearchCV :-

↳ ML model evaluated for range of values provided to hyperparameters.

↳ called Gridsearch CV, searches for best set of hyperparam from grid of hyperparam values.

- Example: consider 2 params: count & rate.

count = [0.1, 0.2, 0.3, 0.4, 0.5]

rate = [0.1, 0.2, 0.3, 0.4]

Performance for all combinations in grid are:

count	0.5	0.42	0.68	0.78	0.79
	0.4	0.40	0.67	0.825	0.81
	0.3	0.45	0.71	0.75	0.80
	0.2	0.35	0.72	0.62	0.61
	0.1	0.42	0.59	0.61	0.78
		0.1	0.2	0.3	0.4
		rate			

Best performance = 0.825 (Highest).

Best combinaⁿ of param values:

count = 0.4

rate = 0.3

- Drawbacks - Go through all intermediate combinations of hyperparameters leads to computational expensive.

2) RandomizedSearchCV :-

↳ solve GridsearchCV drawback, as goes through only fixed num. of hyperparameters setting.

↳ Moves within grid in random fashion to find best set of hyperparams.

↳ Reduce unnecessary computation.

• Implementation steps:

① create hyperparameter grid -

```
param-dist = {"max-depth": [3, None],  
              "max-features": randint(1, 9),  
              "min-samples-leaf": randint(1, 9),  
              "criterion": ["gini", "entropy"]}
```



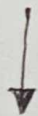
② Instantiate Decision Tree Classifier -

```
sklearn.tree => DecisionTreeClassifier  
tree = DTC()
```



③ Instantiate RandomizedSearchCV object -

```
from sklearn.model-selection import RSCV  
tree-cv = RSCV(tree, param-dist, cv=5)  
tree-cv.fit(X, y)
```



④ Tuned Model -

```
best-params = tree-cv.best-params-  
best-model = tree-cv.best-estimator-  
best-score = tree-cv.best-score-
```