* Hyperparameter Tuning *

- Hyperparameter:- 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 SearchCV
2. Randomized searchCV

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:

	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
count	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 combina^{n} of param values:

count = 0.4

rate = 0.3

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

2) RandomizedSearchCV :-

↳ Solve GridSearchCV drawback, as goes through only fixed num. of hyperparameter 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 sklear.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_
```