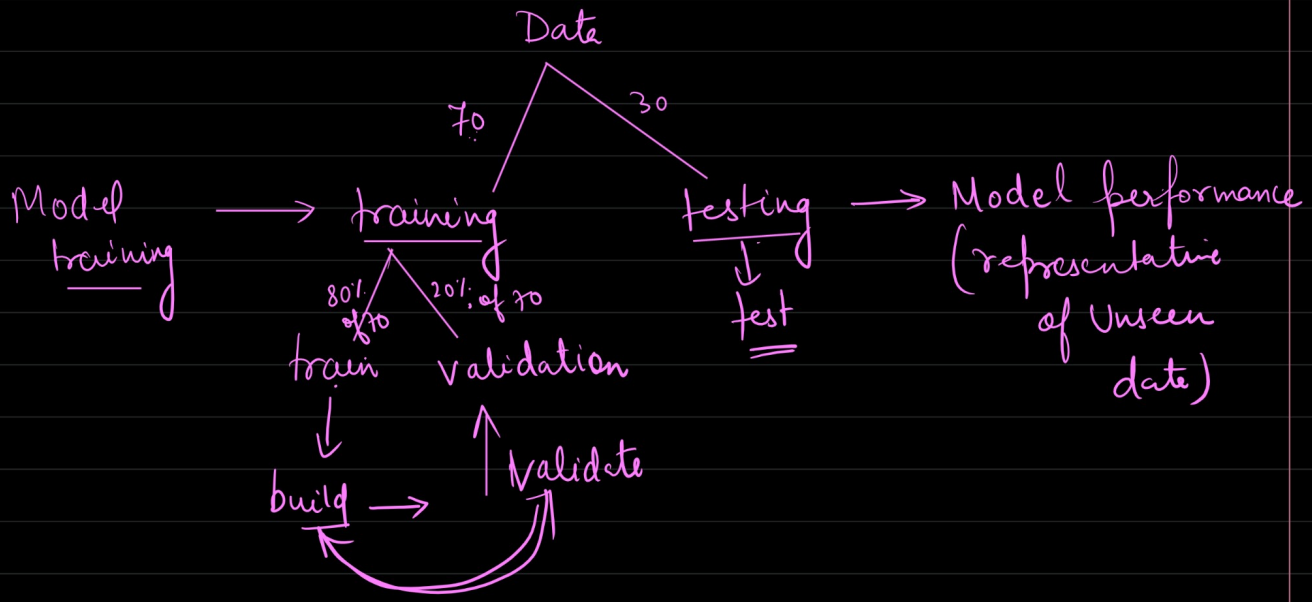
Cross validation, its types and hyper parameter tuning

$$CF = MSE + \lambda((slope)^2)$$

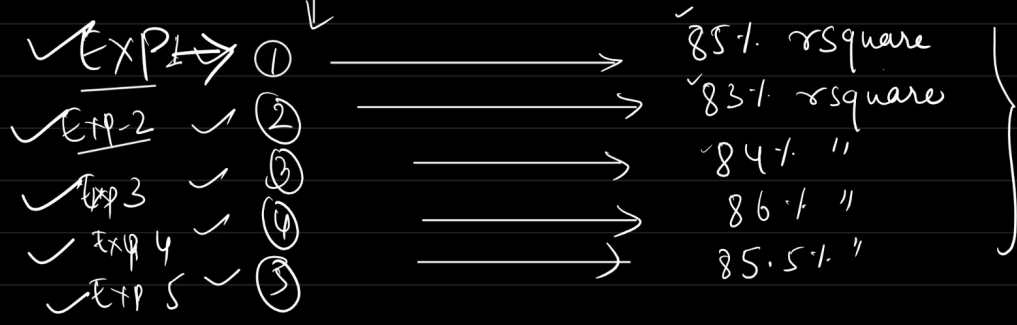


random_state = 1 ⟶ the same data will reproduced.

⟶ train_test_split (x, y)

↓

New x_train, y_train



Cross Validation ⟶ Experimenting with different arrangement of same data to build different models of same Algorithms.

training

Exp-1	train	Validate	Acc 1
Exp-2	train	Validate	Acc 2
Exp-3	"	"	Acc 3
Exp 4	"	"	Acc 4
Exp-5	"	"	Acc 5

training CV=5

↓ Mean of all accuracy at taken

* Cross-validation (train - validation data)
  * Model will be robust / is divided / Generalized

Types

① Leave one out cross validation (LOOCV)

validation training = 100 datapoints

Exp-1 [1 | ← 99 → training data] R^2 = 180% — 1st data – Validation data, 99 data → training data

Exp-2 [1 2 | —— 98 ——→ training] R^2 = 85% — 2nd dp – Validation data, 99 data – training

Validation

Exp-3 [1 | 2 | 3 | —97— traing —] R^2 = 86% — 3rd dp – validation data, 99 dp – traing data

Validation

Exp-100 [—— 99 —— traing —— | 100] $R^2_{100^{th}}$ = 87% — 100th dp – Validation data, 99 dp is training data

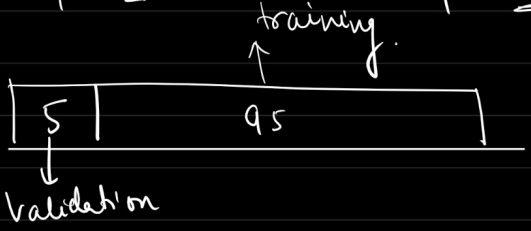
Validation

* Disadvantage (LOOCV) Final accuracy → Average of all the metric
  ① Time complexity is huge for training big data.
  ② Model overfitting ⇒ Training Accuracy ↑
  test ↓

② Leave p out Cross Validation

$p > 1$ $\quad p = 2, 3, 5, 10, 15 \ldots$

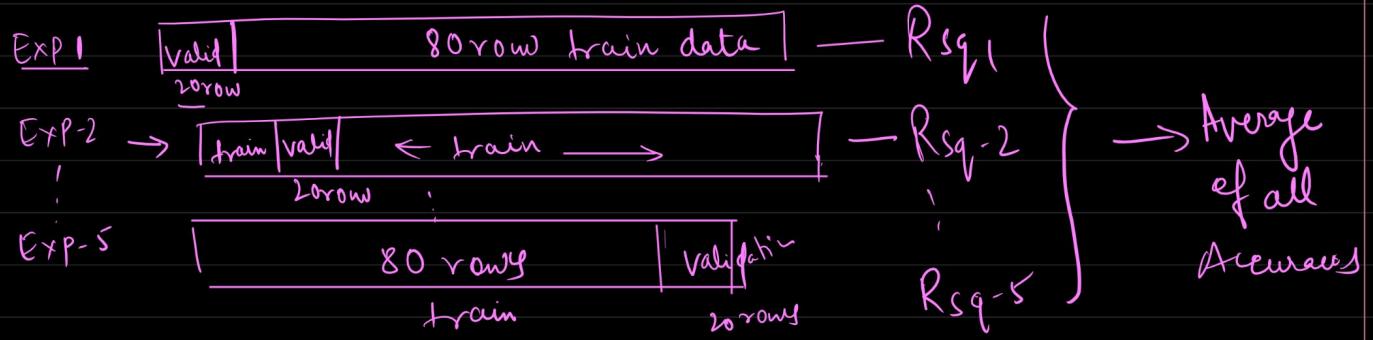
Exp-1



③ K fold Cross Validation

$K = 5$, $n = 100$ $\quad$ Validation $= \frac{100}{5} = 20$

No of groups that you want to divide the data.



④ Stratified K fold Cross validation (imbalance dataset)

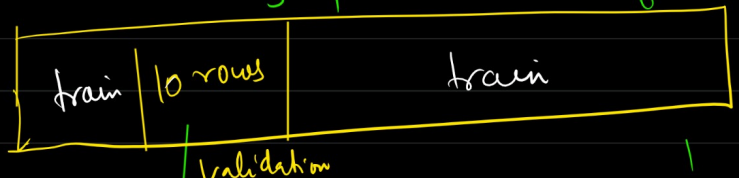
Class 1 → 350
Class 0 - 150

$K = 5$, $n = 100$

Exp-1

Valid
→ equal no of 0 class rows and class 1 rows
5 dp will be of 0th class
5 dp will be of 1st class

Exp-2



✓ 5 dp — 0th class
✓ 5 dp — 1st class

Validation
↓
50% — 0th class
50% — 1st class

Exps | 10 rows validation Acc - 5

5 dp – 0th class
5 dp – 1st class.

* Hyperparameter

↳ External configuration of a model that are not learned from the data but are set prior to training process.

$$Ridge = \sum_{i=1}^{n} (y_{act} - y_{pred})^2 + \lambda((slope)^2)$$

$y_{pred} = mx + c$

External λ dictates the behaviour of training algorithms

→ Model's performance is heavily influenced by hyperparameters.

$$m_{new} = m_{old} - \eta \frac{\partial CF}{\partial m_{old}}$$

Need of Hyperparameter tuning

↳ $\lambda \rightarrow 1, 2, 3, 4, 5, 8, 10, 100$

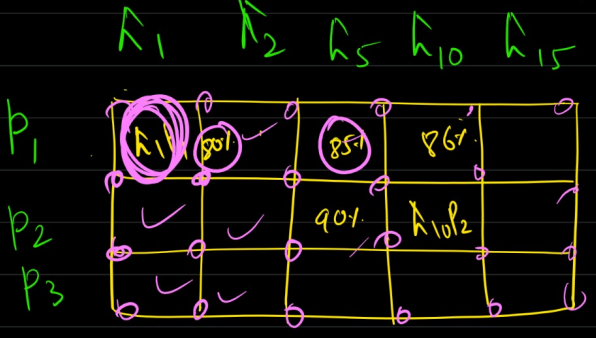
Hyperparameter tuning with cross validation

↳ finding the best parameter while training the models.

① Grid Search CV
② Randomized Search CV

① Grid search CV (Grid search + Cross Validation)

Possible combination $5 \times 3 = 15$



best model

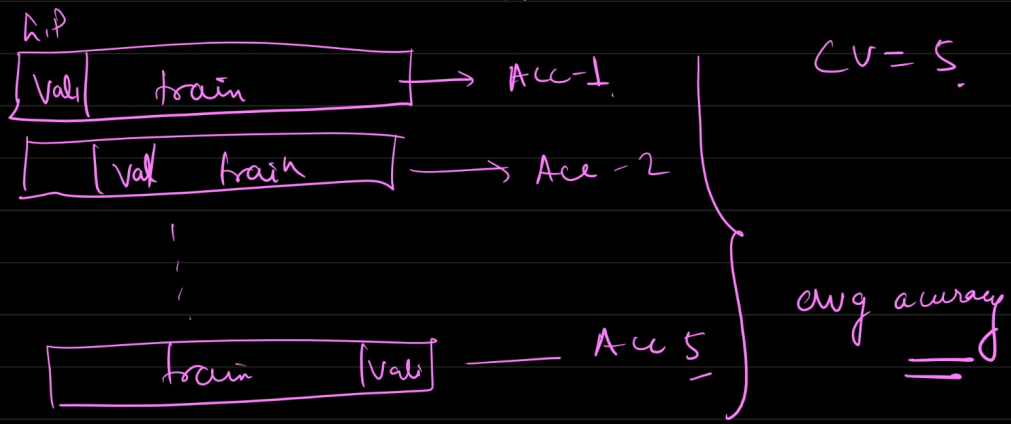
The combination of p and h which gives the maximum accuracy, that will be your best Parameter.

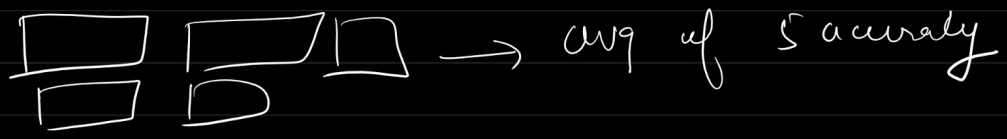
Cross validation

→ for each combination of h and P, internally Cross Validation will happen.

Whole combination

h1 P1
h1 P2
h1 P3
h2 P1
⋮
h15 P3





* for each combination of h and P K fold Cross validation will happen.

total model = $3 \times 5 \times 5$

* Disadvantage

① Time complexity increases with huge data for training the model.

② Randomized Search CV

→ we will not see all the possible combination

$n_1 p_1$
$n_1 p_2$
⋮
$n_{25} p_3$
$\Rightarrow 15$

n_iter = 5 (Select random 5 combination of n and p from all the combination)

5 * CV = 5

⇒ 25 models

⇓

Select the best parameter

* Advantage
→ Time complexity decreases

* if data is small → instead of train - test split → you can go for k-fold cross validation.