Cross validation in Machine Learning

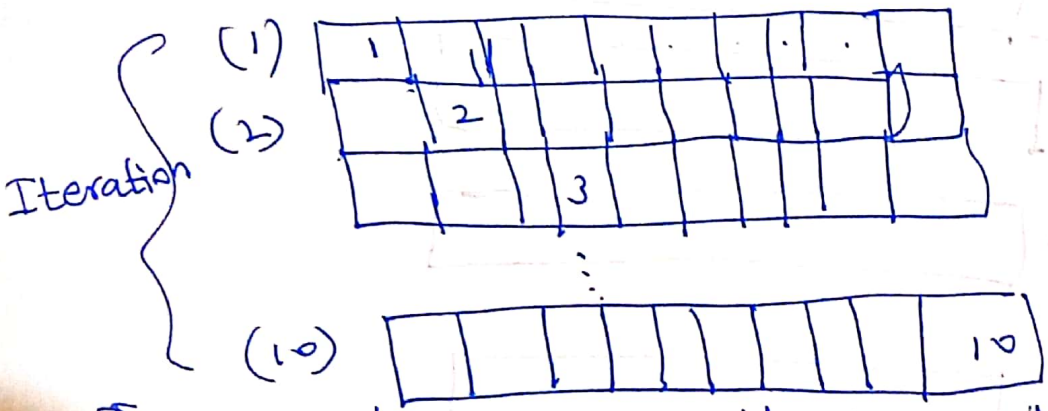
* the idea of Cross validation arises because of the problems with train test model (or) Train test validation model
* It basically wants to guarantee that the score of our model does not depend on the way we ~~pice~~ picked the train set and test set

Types of ~~ed~~ Cross validation:

* k- fold cross validation
* stratified k- fold
* Leave one out cross validation
* Leave p- out cross validation

k- fold Cross validation:

In k- fold Cross validation we divide the entire data set into k folds. suppose k = 10. We divide the entire data set into 10 folds.

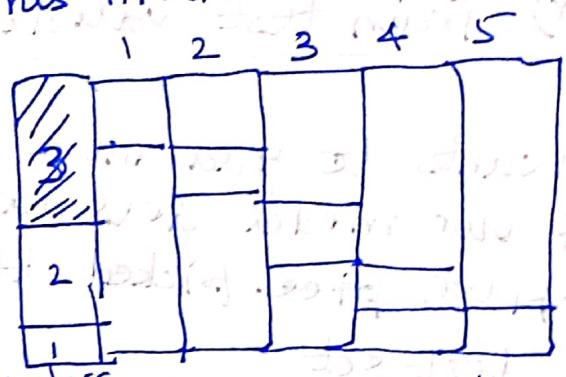


* In each case 1 folder is used for testing and the remaining 9 folders are used for

training.

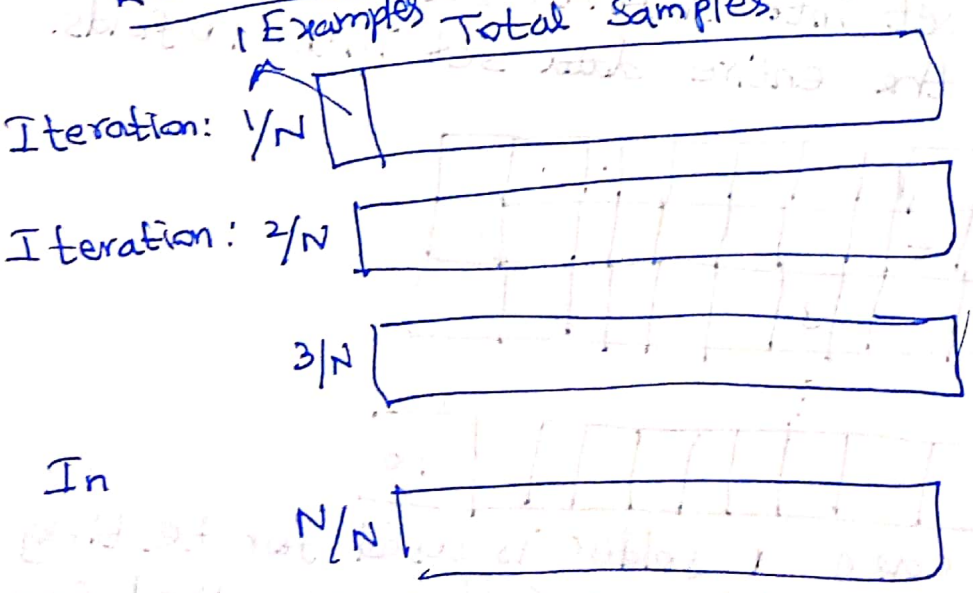
Average performance for these iterations are calculate.

This performance is final performance of this model.

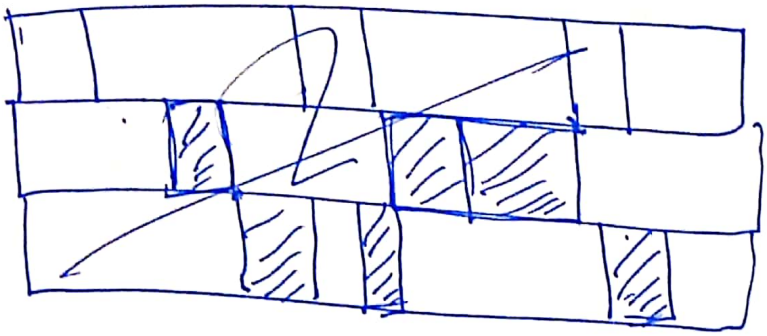


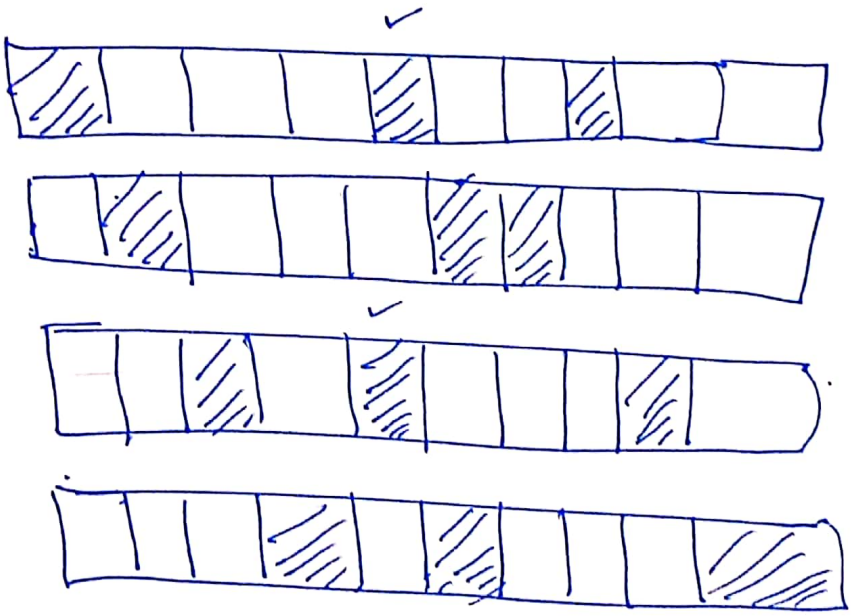
In K-fold cross validation, the selected class cannot have equal representation But, In class, there may not be equal representation. There shall be bias. Hence to avoid this stratified cross validation technique is used. so that each-class will have equal representation.

Leave one out cross valution:



Leave p- out Cross out validation





$p = 3$

Leave One out Cross Validation

Let us consider ~~n observat~~ n sample data set points.

(x_1, y_1), (x_2, y_2), ... (x_n, y_n).

Take $\{(x_1, y_1)\}$ as validation set and $\{(x_2, y_2); (x_3, y_3), \ldots (x_n, y_n)\}$ as training set. [(n-1) -observations]

We have to predict the value of y_1. name this value as $\hat{y}_1$.

$\hat{y}_1 - y_1$ will the predicted error.

~~(i.e) MSE~~ $_1$

step	Validation set / Predicted value		Training set	Error
1	(x_1, y_1)	$\hat{y}_1$	$(x_2, y_2)(x_3, y_3) \ldots (x_n, y_n)$	$MSE_1 = y_1 - \hat{y}_1$
2	(x_2, y_2)	$\hat{y}_2$	$(x_1, y_1), (x_3, y_3) \ldots (x_n, y_n)$	$MSE_2 = y_2 - \hat{y}_2$
3	(x_3, y_3)	$\hat{y}_3$	$(x_1, y_1)(x_2, y_2), (x_4, y_4) \ldots (x_n, y_n)$	$MSE_3 = y_3 - \hat{y}_3$
⋮	⋮			
n	(x_n, y_n)		$(x_1, y_1), (x_2, y_2) \ldots (x_{n-1}, y_{n-1})$	$MSE_n = y_n - \hat{y}_n$

$$\text{LOOCV error} = \frac{1}{n} \cdot \sum_{i=1}^{n} MSE_i$$

$$LOOCV_{(n)}$$

This ~~LOOCV(n)~~ $LOOCV_{(n)}$ is the mean MSE. and this value is the estimate for the test error.

Advantages of LOOCV

1. It is less biased because we use (n-1) observations out of n observation.
2. Not much over estimation of test error.
3. It gives consistant result because there is no random selection among training validation set values.

Disadvantage:

1. It is Computationally expensive, because we have to find MSE's 'n' times for 'n' observations.