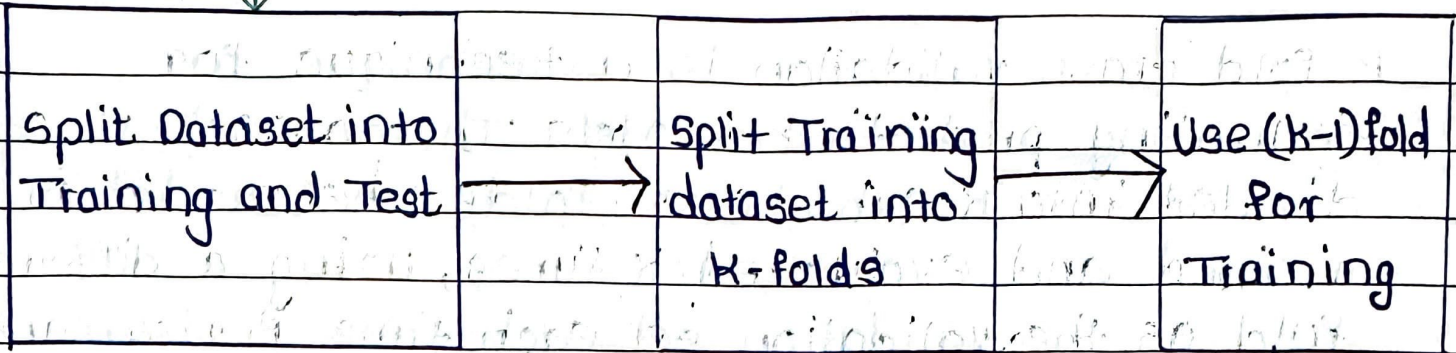
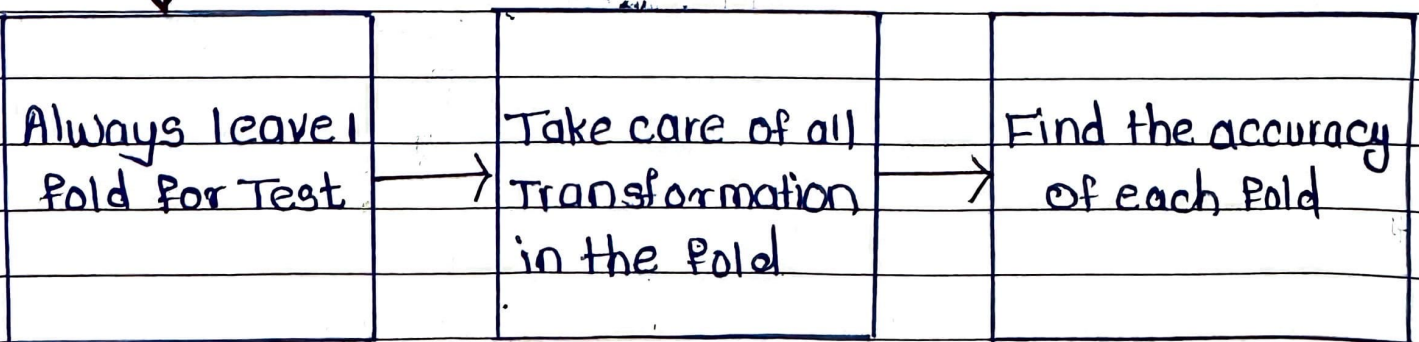


K-Fold Cross Validation

Scuffle Dataset



| | | | | | | | | | |
|---|----------|---|----------|---|----------|---|----------|----------|-------|
| 1 | Test | 1 | Training | 1 | Training | 1 | 1 | | |
| 2 | Training | 2 | Test | 2 | Test | 2 | Training | 2 | Train |
| 3 | | 3 | Training | 3 | | 3 | 3 | ing | |
| 4 | Training | 4 | Training | 4 | Training | 4 | Test | 4 | |
| 5 | | 5 | | 5 | | 5 | 5 | Training | 5 |



K-fold cross validation is a technique for evaluating predictive models. The dataset is divided into K subsets or folds. The model is trained and evaluated K times, using a different fold as the validation set each time. Performance metrics from each fold are averaged to estimate the model's generalization performance.

Let's have a generalised K value. If $K=5$, it means, in the given dataset we are splitting into 5 folds and running the Train and Test. During each run, one fold is considered for testing and the rest will be for training and moving on with iterations.

K-Fold Cross Validation

I am creating a simple array, defining the K size as 5 and splitting my array. Using the simple loop and printing the Train and Test portions. Here we could see clearly that the data points in Training buckets and Test buckets are unique in each cycle.

```
from sklearn.model_selection import KFold

import numpy as np

data=np.array([6,12,18,24,30,36,42,48,54,60,1,2,3,4,13,14,15,16,19,20
kfold=KFold(5,shuffle = True)
for train,test in kfold.split(data):
    print("Train:" ,data[train], ",Test:", data[test])
```

```
➞ Train: [ 6 12 18 24 30 36 42 48 54 60  1  2  3 14 15 16 19 20 21 49 53 54]
Train: [12 18 24 36 42 48 54 60  2  3  4 13 14 15 19 20 49 53 54]
Train: [ 6 24 30 42 48 54 60  1  2  3  4 13 14 16 19 20 21 49 54]
Train: [ 6 12 18 24 30 36 54  1  4 13 14 15 16 19 20 21 49 53 54]
Train: [ 6 12 18 24 30 36 42 48 54 60  1  2  3  4 13 15 16 21 53]
```

[+ Code](#)
[+ Text](#)