## K-Fold cross validation

Scuffle Dato	set		\		
PORT THE	0.140.00	ol varification	, Ja	him	
Split Dotasetin		Split Training	10.	use (k-1) fold	1
Training and 7		dataset into	13/10/1	Port	
- markeritis is mais	Ti , BC Will >	K-folds	Garage C	Training	
continued to the second	as a dort	is a complexit	I I	a Interior	
mitigation of home posterior and by the days and assimilar					
	area may a	ail pulling age	3/19/19	14, 001	,
7	Test Training 1 Training 1				
2 Training 2 Training 3 Test 3 Ing					
5. Training 5 Test					
April en le l'est distribute de la					
ENGERT A FOR TO JOB (DE TOET AND ENCAPERATE ENCAPE					
Always leave 1	Take c	are of all	Find the	e accuracu	
fold for Test	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	ormation >	of each	e accuracy	
	in the	Pold		ί <sub>7</sub>	
					,
	ч				
		-			

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.

promote of their line is given the little work squared in the second squared in the seco

## **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 30 36 42 48 60 1
                                         2
                                            3 14 15 16 19 20 21 49 53
     Train: [12 18 24 36 42 48 54 60 2 3 4 13 14 15 19 20 49 53 52
     Train: [ 6 24 30 42 48 54 60
                                   1 2 3 4 13 14 16 19 20 21 49 52
     Train: [ 6 12 18 24 30 36 54
                                   1 4 13 14 15 16 19 20 21 49 53 52
     Train: [ 6 12 18 24 30 36 42 48 54 60
                                               2
                                                  3
                                            1
                                                     4 13 15 16 21 53
                         + Code
                                     + Text
```