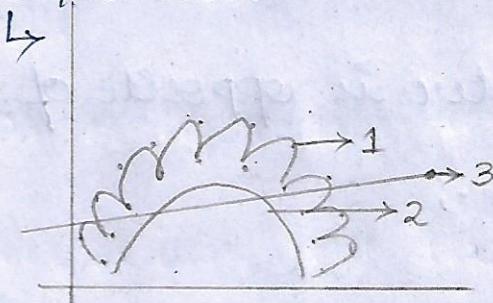
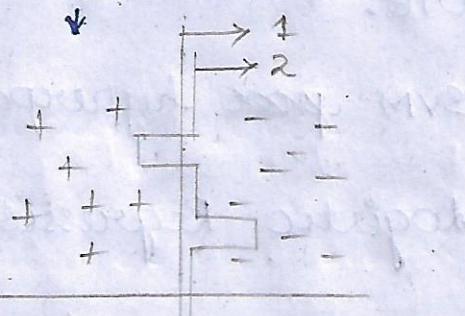


From the hyperparameters,

### REGRESSION



### CLASSIFICATION



1 → SIMPLE MODEL, 2 → COMPLEX MODEL

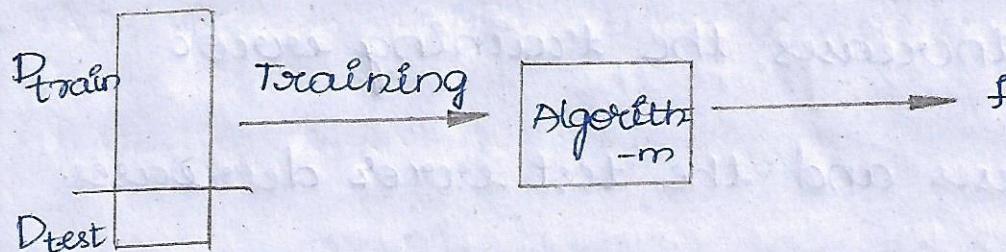
3 → OPTIMAL MODEL

### OCCAM'S RAZOR:

It states that, the model learning should be as simple as possible but not too simple.

→ The idea that the simplest and most direct solutions should be preferred, or that with the <sup>different</sup> hypothesis; the simplest one (or) the one with fewest assumptions will be best applied.

→ From the classification figure, we are getting complex model. This is because of of the training data i.e.,



↳ Because of the 'f' we get the complex model.

↳ There is "No training Error, but the Test error is more".

For Example:

→ In KNN → if  $K=1$  then,

Training error is very low & Test error is high.

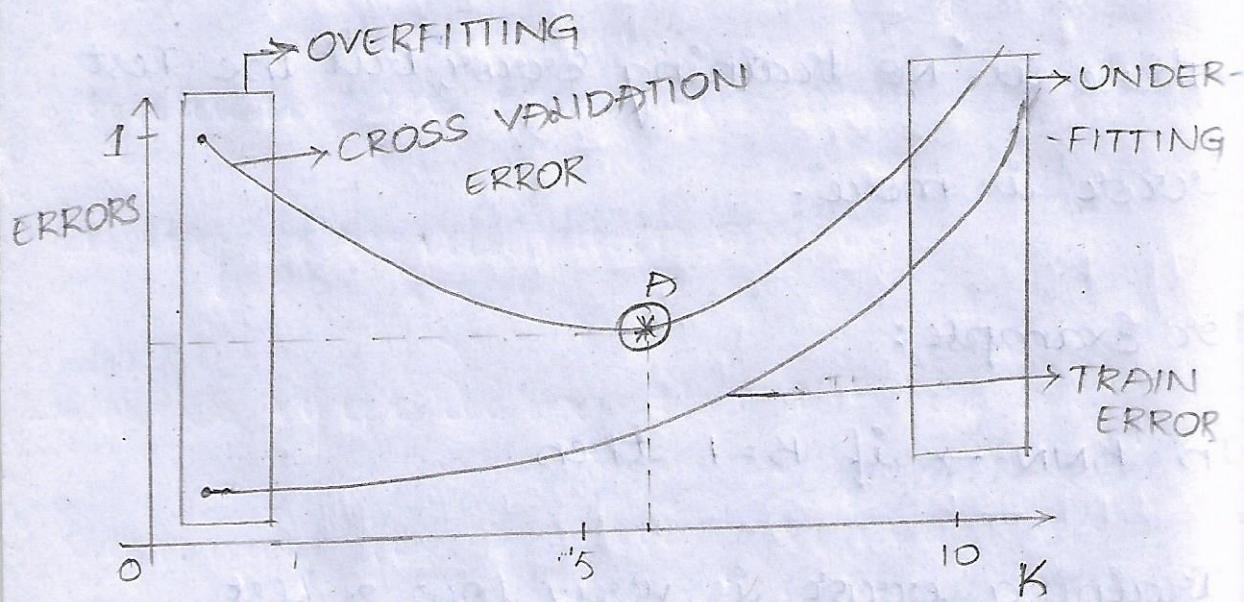
→ In KNN → if  $K=5$  then,

Training error is a bit high, & Test error tends to be very high.

In the above case i.e.,  $K=5$  & increase in  $K$  value we get a smooth line.

NOTE:

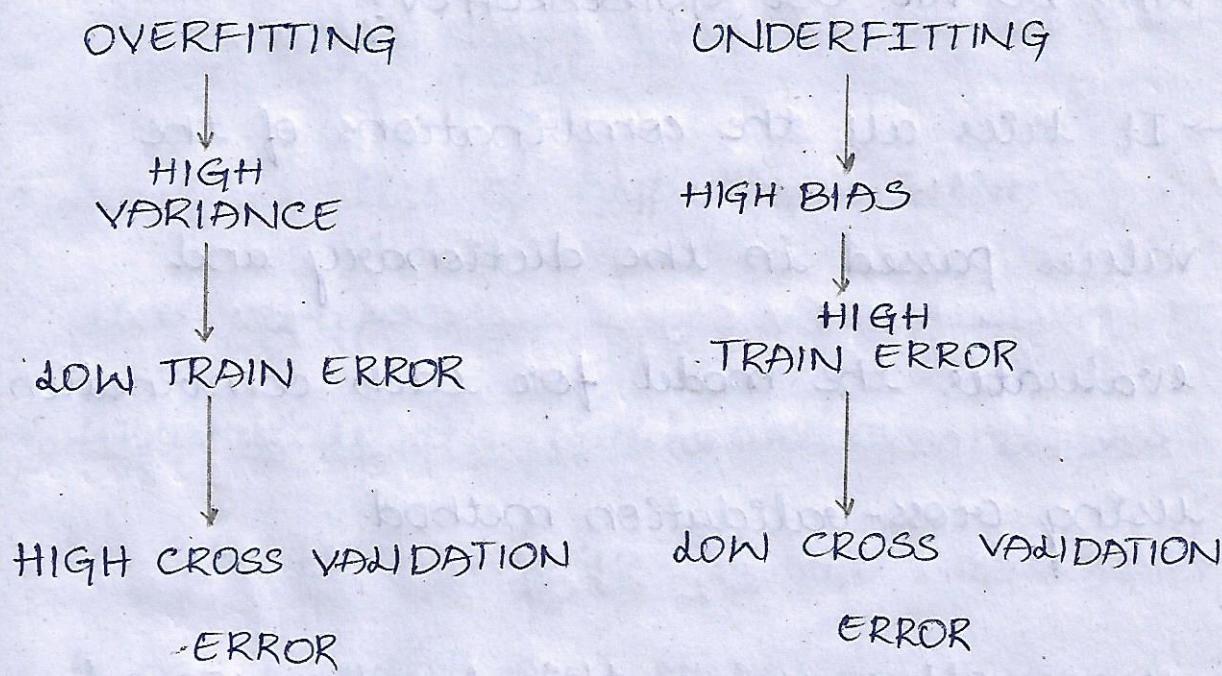
- \* As  $K$  increases, the training error increases and the test error decreases.
- \* For some extent of test error, tends to increase again.



→ We Need to take the value of cross validation error where it is <sup>very</sup> <sub>low</sub>.

→ From the above graph we can say  
that the validation accuracy is low  
i.e., the error is high.

→ Then <sup>for</sup> the training data, the accuracy is  
high.



### GRIDSEARCHCV :

It is a library function that is a member of sklearn's model selection package.

→ It helps to loop through pre-defined

hyperparameters and fit our estimator i.e., the model on our training set.

→ So, in the end, we can select the best parameters from the listed.

### WHY DO WE USE GRIDSEARCH CV?

→ It tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using cross-validation method.

→ Hence after using this function we get accuracy/loss for every combination of hyperparameters.

→ So, we can choose the <sup>one with</sup> best performance.

\* In GridSearchCV, the fold value is taken as '5' by default.

### STEPS TO PERFORM GRIDSEARCHCV:

- \* Install sklearn library. PIP install sklearn
- \* Import sklearn library
- \* Import the model
- \* Create a list of hyperparameters dictionary.
- \* Instantiate the GridsearchCV and pass in the parameters.
- \* Finally, print out the best parameters.

### RANDOMIZED SEARCHCV:

Random search is a technique, where random combinations of the hyperparameters are used to find the best solution

for the built model.

- It is similar to grid search, and yet it has proven to yield better results comparatively.

#### WHY DO WE USE RANDOMIZED SEARCHCV?

As we know that,

- GridSearchCV object searches for the best parameters and automatically fits a new model on the whole training dataset.
- RandomizedSearchCV is very useful when we have many parameters to try and the training time is very long.

## CROSS-VALIDATION :

It is a technique to evaluate the Predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

## K-FOLD CROSS-VALIDATION :

→ Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

→ The procedure has a single parameter called 'k' that refers to the number of groups that a data sample given is to be split.

→ Take the group as a hold out or test data set.

### HOW K-FOLD CROSS-VALIDATION IMPLEMENTED?

→ It is implemented by randomly dividing the set of observations into "k groups (or) folds" of approximately equal size.

→ This procedure is repeated 'k' times, each time a different group of observations is treated as a validation set.

If  $k=1$  in the cross-validation, then we get accuracies.

→ Take the mean then we'll get mean accuracy. - time complexity increases.

NOTE:

(189)

In general, we take 10 folds.