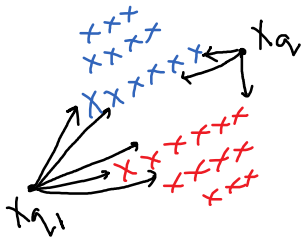
Saturday, May 4, 2024 9:38 AM

KNN $K = 3, 5, 7, 9, \dots$

\# neighbours

↳ K-Nearest Neighbours

↳ you are like your neighbours



$K=3$

$X_q = (2B, 1R) \Rightarrow$ Blue

$X_{q,1} = (2B, 3R) \Rightarrow$ Red

$K=5$

Classification

↓

voting

Regression

↳ Average

Can I take even number of neighbours?

$K=4$ $X_{q_2} = (2B, 2R) \rightarrow P(B) = \frac{1}{2}$

$\rightarrow P(R) = \frac{1}{2}$

Hyperparameters

K ⇒ number of neighbours ⇒ K↑ or K↓ ⇒ helps you in improving the model.

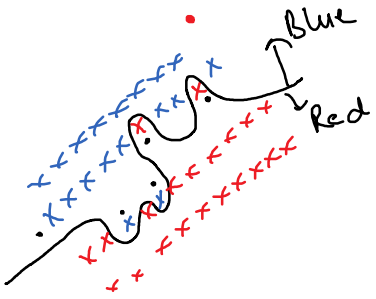
distance ⇒ distance ↓ euclidean, manhattan, minkowski

effects of K

⇒ K=1 → Non-smooth decision boundary

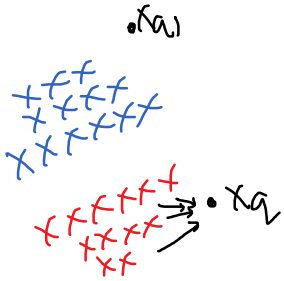
→ No mistakes

⇒ K=1



→ No mistakes
→ Perfect boundary
→ Training accuracy ↑ } overfitting
→ Testing accuracy ↓ } ↳ low bias & high variance
↓
explore?

2) K = N



X_q = (13B, 12R) ⇒ Blue

x_{q1} = (13B, 12R) = Blue

x_{q3} = (13B, 12R) = Blue

irrespective of x_q position, the answer will always be blue.

→ Training accuracy ↓ } → underfitting
→ Testing accuracy ↓

3) K = 5

x_{q1} = (4B, 1R) = Blue

x_q = (4R, 1B) = Red

Training accuracy is fine (80% - 90%)

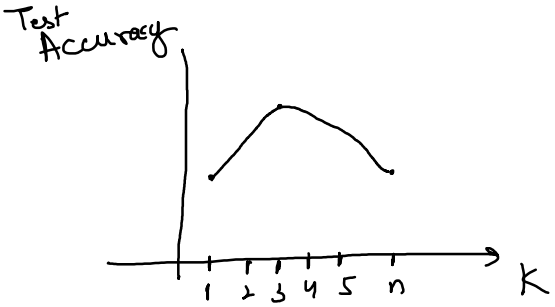
→ Training accuracy is fine (80% - 90%)

→ Test accuracy is fine (80%)

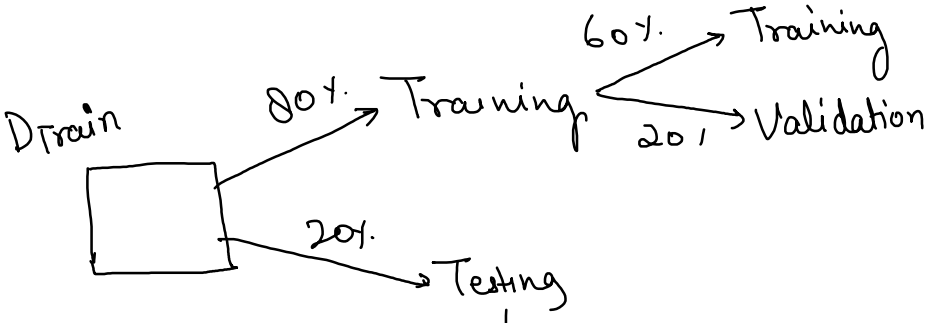
↓

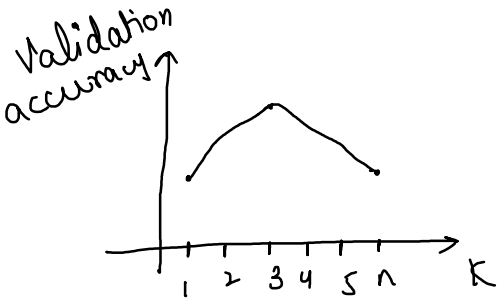
Right fit or best fit

curve of K with accuracy:



Choosing the right value of K ⇒ Cross-validation

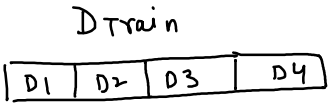




CROSS-VALIDATION ⇒ K-fold ⇒ K'-fold Thumb rule ⇒ K'=10 (industry)

K' ⇒ how many partitions will be done of your dataset

K'=4



K (# neighbours)	Training	Validation	Accuracy
1	D1 D2 D3	D4	a'_1
1	D2 D3 D4	D1	a'_2 → a'_{avg}
1	D1 D2 D4	D3	a'_3
1	D1 D3 D4	D2	a'_4
2	D1 D2 D3	D4	a^2_1
2	D2 D3 D4	D1	a^2_2 → a^2_{avg}
2	D1 D2 D4	D3	a^2_3
2	D1 D3 D4	D2	a^2_4

validation accuracy avg_list = [a'_{avg} (K=1), a^2_{avg} (K=2), a^3_{avg} (K=3), a^4_{avg} (K=4), a^5_{avg} (K=5), a^6_{avg} (K=6), a^7_{avg} (K=7)]

from the list, choose value k that has highest avg validation accuracy

Advantages:

Disadvantages:

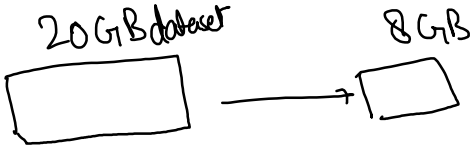
Advantages:

→ It is very easy to understand

→ No assumptions in the algorithm

Disadvantages:

→ Lazy learner
↳ does all calculations at time of execution



→ space issues

→ very slow algorithm

→ Imbalanced dataset

Application: Healthcare

Evaluation Metrics
(CLASSIFATION)

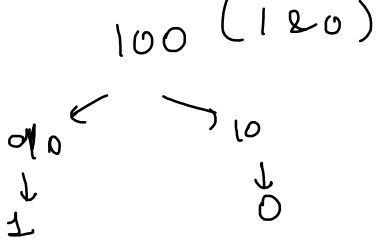
CONFUSION MATRIX

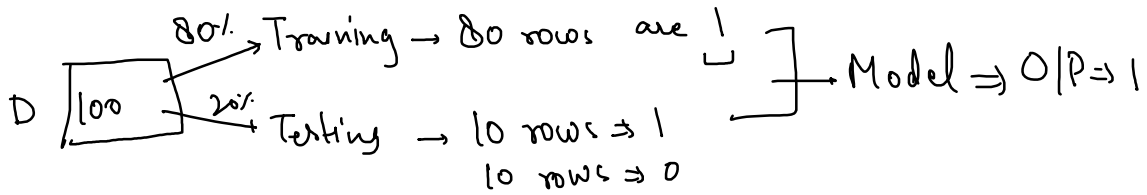
Predicted Values \ Actual Values	0	1	
0	TN	FN	⇒ Total predicted 0 (-ves)
1	FP	TP	⇒ Total predicted 1 (+ve)
	⇓ Total actual -ve (0)	⇓ Total actual positive (1)	

1) Accuracy $\Rightarrow$ $\frac{TP+TN}{TP+FP+TN+FN}$

Accuracy isn't reliable under two situations:

→ Imbalanced dataset





$\Rightarrow$ Prob $\Rightarrow$ 0.5 (complete confusion)

→ ↓Precision $\Rightarrow$ $\frac{TP}{TP+FP\uparrow}$

→ Recall ↑ $\Rightarrow$ $\frac{TP}{TP+FN\downarrow}$ (Sensitivity) (TPR)

→ FPR = $\frac{FP}{TN+FP}$

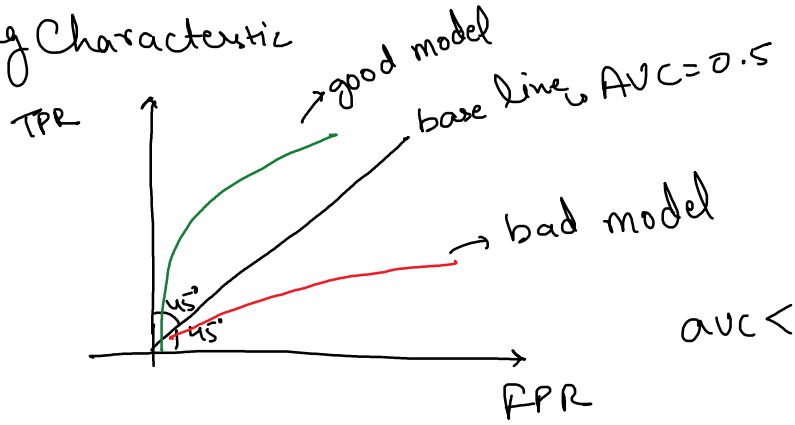
→ F1-Score ⇒ $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

→ specificity = 1 − FPR = $1 - \frac{FP}{TN+FP} = \frac{TN + FP - FP}{TN+FP} = \frac{TN}{TN+FP}$

→ ROC - AUC curve

↓

Receiver Operating Characteristic



auc < 0.5
bad model

auc > 0.7
good model