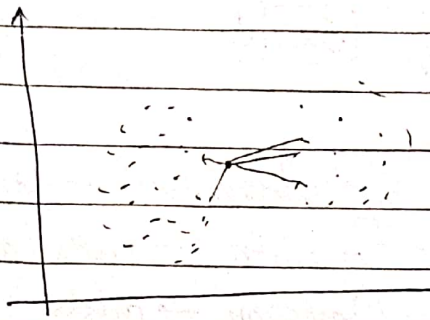
Non-parametric Learning Algo → opposite for Parametric

(Training data required for inferencing (learning faster, influence stored)

KNN (K-Nearest Neighbours)

Usually K is odd.



Calculate all dist. → arrange in ascending order → consider the first K values. → majority of their labels → our prediction

→ euclidean distance = $\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ (ellipse in 2D, spherical shape in 2D)

for 2 pts → $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

→ manhattan distance = $\sum_{i=1}^{n} |x_i - y_i|$ → for 2 pts 2D. $D = |x_2-x_1| + |y_2-y_1|$

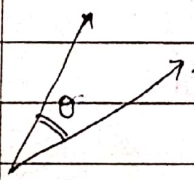
minkowski distance = $\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$

p = 1 : Manhattan Distance

p = 2 : Euclidean Distance

p → infinity : Chebychev Distance

cosine distance / similarity:



$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_{i}^{n} a_i b_i}{\sqrt{\sum_{i}^{n} a_i^2} \sqrt{\sum_{i}^{n} b_i^2}}$

Hamming Distance = 01101010 ⊛ 11011011 → 01110001 → 4 1s, (differ at 4 places).

d(01101010, 11011011) = 4

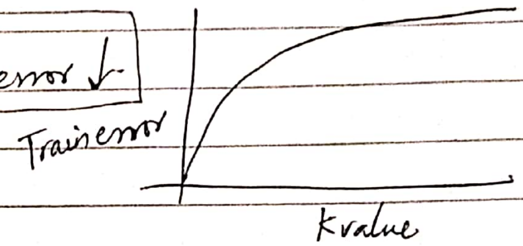
Assumptions in traditional KNN:

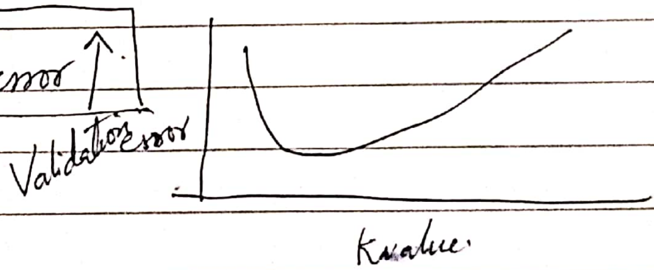
- Equal weight to all attributes (to compute distance)
- Scale of attribute range - same
- Equal variance
- Classes - spherical.
- Attributes - very less noise
- Equal importance

Impact of K:

K less - Training error ↓

K more - Training error ↑





→ Distance-weighted KNN

- ~~[illegible]~~

if K is small - algo sensitive to outliers

$$Pred_{test} = \frac{\sum_{i=0}^{k} w_i \cdot class_i}{\sum_{i=0}^{k} w_i}$$

$$w_K = \frac{1}{Dist(x_K, x_{test})}$$

K large - may include unnecessary values from other classes.

→

Weighted distance function.

$$D(x_i, x_j) = \sqrt{\sum_{i=1}^{m} w_i (x_{i,m} - x_{j,m})^2}$$

every feature has difference weight

$w_i \rightarrow$ high

$w_i \rightarrow$ low

$w_i = 0$

→ Mean Absolute Error

$$MAE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y}_i|$$

does a poor job when scale of the data is high

→ Mean Squared Error

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2$$

Squaring +ve value, sum won't be zero

" emphasizes larger differences

→ Median Absolute Error.

$$MedAE(y, \hat{y}) = median(|y_1 - \hat{y}_1|, \ldots |y_n - \hat{y}_n|)$$

↳ does not support multioutput

Median AE prevents outliers contributing more error to model evaluation

Scale of prediction too high logs scales it down

→ Mean Squared Logarithmic Error.

$$MSLE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (\log_e(1 + y_i) - \log_e(1 + \hat{y}_i))^2$$

when targets have exponential growth → population counts, average sales of commodity over years

Method penalizes an under predicted estimate greater than over predicted estimate

regression + classification metrics

→ Max Error.

$$\text{MaxError}(y, \hat{y}) = \max(|y_i - \hat{y}_i|)$$

aims to find max error made by model for single sample → finding best model that covers all samples.

Similarity Measures.

→ Explained Variance Score. $(y, \hat{y})$ Best possible score → 1.0 lower values are worse

$$= 1 - \frac{\text{Var}\{y - \hat{y}\}}{\text{Var}\{y\}}$$

Estimates deviation between prediction and actual values

→ R^2 score, the coeff of determination

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \epsilon_i^2$$

calculates unadjusted R^2 without correcting for bias in sample variance of y

Error / Dissimilar Measure (less the better)

Accuracy / Similar Measure (more the better)

Classification Metrics

	actual	predicted	
True +ve	1	1	11 → 1
False +ve	0	1	01 → 0
False -ve	1	0	10 → 0
True -ve	0	0	00 → 1

$A\bar{B} + \bar{A}B$ - EXOR.

$AB + \bar{A}\bar{B}$ EXNOR

→ $\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$ } all true values / total

→ $\text{Precision} = \frac{TP}{TP + FP}$ ↑ how precision ↑ fraction of predicted +ves that are actually +ve

Precision Recall / Confusion matrix

Recall (Sensitivity) = $\frac{TP}{TP + FN}$ ← correct; → 0 then recall max; → all count

Total days - 30 (one month)
actual rain - 10
Model - 9 days (5 ✓, 4 ×)
Precision = 5/9 ; Recall = 5/10

Accuracy :- (TP + TN) / total → TP+FP+TN+FN

True +ve Rate :- TP / actual yes → FN+TP
(Sensitivity / Recall)

False +ve Rate :- FP / actual no → TN+FP

True -ve Rate :- TN / actual no → TN+FP
(Specificity)

Precision :- TP / predicted yes → FP+TP

	Predicted NO	Predicted YES
Actual NO	TN	FP
Actual YES	FN	TP

Misclassified Rate :- FP + FN / Total → TP+FP+TN+FN

False +ve Rate (miss rate) = FN / actual yes → FN+TP

→ $F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * (precision * recall)}{precision + recall}$

50% 50% focus

→ F_β score = $\frac{(1+\beta^2) \cdot (precision \cdot recall)}{(\beta^2 \cdot precision + recall)}$

$\beta = 0$ P ↑

$\beta = \infty$ R ↑

- $F_{0.5}$ (beta = 0.5) :- Precision ↑ Recall ↓
- F_1 (beta = 1.0) :- Precision 50% Recall 50%
- F_2 (beta = 2.0) :- Precision ↓ Recall ↑

ROC AUC curve

ROC Curve and ROC AUC Score

helps in understanding balance b/w TPR & FPR

Overfitting/underfitting

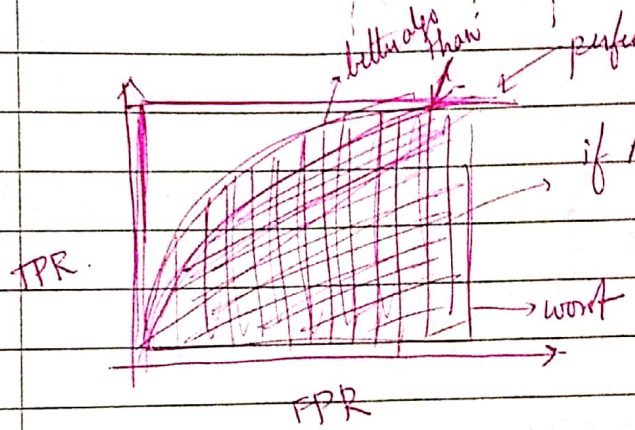
- thresholds → all unique prediction probabilities in 'descending order'
- FPR = (FP / (FP + TN)) } for each threshold
- TPR = (TP / (TP + FN)) }

Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s.

eg: $th_1 > 0.3$ | 1 else 0

$th_2 > 0.41$ | 1 else 0

	fps	tps
th_4	x_4	y_4
th_2	x_2	y_2
th_3	x_3	y_3



Overfitting and underfitting

$x_1 \propto y$ $\quad x_2 \propto \frac{1}{y}$ $\quad x_3 \not\propto$ (no impact) y

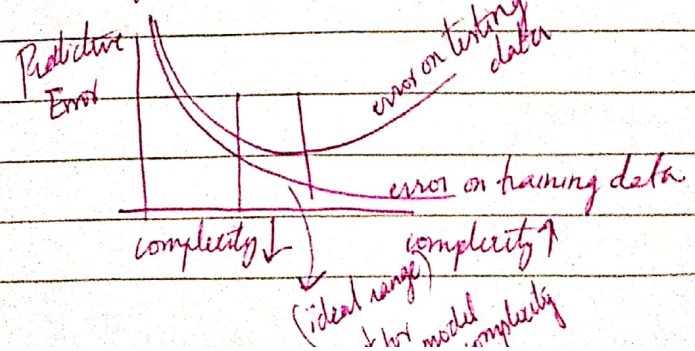
$\uparrow w_1$ $\quad w_2 \downarrow$ $\quad w_3 = 0$

always tries to get +ve impact on 'y'

Training Error	Testing Error
Training Loss	Testing Loss
Underfit	Overfit
Bias	Variance

→ A complex model: y = high order polynomial in X

→ A true (simpler) model: $y = aX + b +$ noise



training acc	test acc	
60	58	HB LV
60	40	HB HV
98	70	LB HV
98	96	LB LV