→ Relationship Diagram

Linear Regression

Simple L.R (one-one)

$y = \alpha_0 + \alpha_1 x_1 + \text{error}$

α_0 ↓ , α_1 ↓ regression coefficient

Actual value - Predicted value

If there is one independent variable that affect the dependant variable then the regression is called Simple L.R.

Multiple L.R (many-one)

$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_m x_m + \text{error}$

α_0 ↓

Many Independent variable
One Dependent variable.

When we have more than one independent variable that effect one dependent variable is called multiple linear Regression.

• Linear Regression with least square criterion.

→ Best fit line equation:

$$\hat{y}^{(i)} = w_0 + w_1 x_1^{(i)} + \cdots + w_n x_n^{(i)} \; ; \; i = 1 \ldots N$$

Where $w_0, w_1, \ldots, w_n$ are the regression coefficient.

The method of linear regression is to choose the coefficients w_0 to w_n to minimize the residual sum of squares of these differences over all the N training instances.

The performance criteria is the sum of error squares.

Residual error : $e^{(i)} = y^{(i)} - \hat{y}^{(i)} = y^{(i)} - \sum_{j=0}^{n} w_j x_j^{(i)} \; ; \; x_0^{(i)} = 1$

Residual sum of error square:

$$E = \sum_{i=1}^{N} (e^{(i)})^2 = \sum_{i=1}^{N} \left(y^{(i)} - \sum_{j=0}^{n} w_j x_j^{(i)} \right)^2 \; ; \; x_0^{(i)} = 1$$

Minimal Sum of Error Square

$$X = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(N)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(N)} \\ \vdots & \vdots & & \vdots \\ x_n^{(1)} & x_n^{(2)} & & x_n^{(N)} \end{vmatrix} = [\bar{x}^{(1)} \bar{x}^{(2)} \cdots \bar{x}^{(N)}]$$

$$y = [y^{(1)} \; y^{(2)} \; y^{(3)} \cdots y^{(N)}]^T$$

$$\bar{\omega} = [\omega_0 \; \omega_1 \; \omega_2 \cdots \omega_n]^T$$

$$\bar{x} = [1 \; x_1 \; x_2 \cdots x_n]^T$$

$$y^{(1)} = \bar{\omega}^T \bar{x}^{(1)}$$

$$\vdots$$

$$y^{(N)} = \bar{\omega}^T \bar{x}^{(N)}$$

Therefore,

$$[y^{(1)} y^{(2)} y^{(3)} \cdots y^{(N)}] = \bar{\omega}^T [\bar{x}^{(1)} \bar{x}^{(2)} \cdots \bar{x}^{(N)}] = \bar{\omega}^T X$$

or $$\hat{y} = (\bar{\omega}^T X)^T$$

The vector of residual or becomes

$$\text{error} = y - (\bar{\omega}^T X)^T$$

Hence the error function can be written as

$$E(\bar{\omega}) = [y - (\bar{\omega}^T X)^T]^T [y - (\bar{\omega}^T X)^T]$$

$$= \bar{\omega}^T [X X^T] \bar{\omega} - 2\bar{\omega}^T X y + y^T y$$

→ In the least square estimation the objective is to find the optimum value of ω which is written as $\bar{\omega}^*$ which minimizes the error function.

$$\frac{\partial E(\omega)}{\partial \omega} = 2(xx^T)\bar{\omega} - 2xy = 0$$

This gives

$$\bar{\omega}^* = (xx^T)^{-1} xy$$

The fitted output values at the training data are

$$\hat{y} = x^T \bar{\omega}^* = x^T (xx^T)^{-1} xy$$

$$\bar{\omega}^* = x^+ y$$

Q)

y	3	2	4	5	8
x_1	2	3	5	7	8
x_2	1	5	3	6	7

find the estimated value of y.

y	x_1	x_2	equation
3	2	1	$3 = 1\omega_0 + 2\omega_1 + 1\omega_2 + \varepsilon_1$
2	3	5	$2 = 1\omega_0 + 3\omega_1 + 5\omega_2 + \varepsilon_2$
4	5	3	$4 = 1\omega_0 + 5\omega_1 + 3\omega_2 + \varepsilon_3$
5	7	6	$5 = 1\omega_0 + 7\omega_1 + 6\omega_2 + \varepsilon_4$
8	8	7	$8 = 1\omega_0 + 8\omega_1 + 7\omega_2 + \varepsilon_5$

(error)

It is to be calculated for determining the matrix properly

$$\underbrace{\begin{pmatrix} 3 \\ 2 \\ 4 \\ 5 \\ 8 \end{pmatrix}}_{y} = \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 1 & 3 & 5 \\ 1 & 5 & 3 \\ 1 & 7 & 6 \\ 1 & 8 & 7 \end{bmatrix}}_{X} \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix}}_{\beta} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{pmatrix}}_{\epsilon}$$

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 3 & 5 & 7 & 8 \\ 1 & 5 & 3 & 6 & 7 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 1 & 3 & 5 \\ 1 & 5 & 3 \\ 1 & 7 & 6 \\ 1 & 8 & 7 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 5 & 25 & 22 \\ 25 & 151 & 130 \\ 22 & 130 & 120 \end{bmatrix}_{3\times3}$$

$$\Rightarrow (X^T X)^{-1} = \frac{1}{|X^T X|} \text{Adj} \, |X^T X|$$

$$= \begin{bmatrix} 1.201 & -0.138 & -0.071 \\ -0.138 & 0.114 & -0.098 \\ -0.071 & -0.098 & 0.128 \end{bmatrix}$$

$$= (X^T X)^{-1} X Y =$$

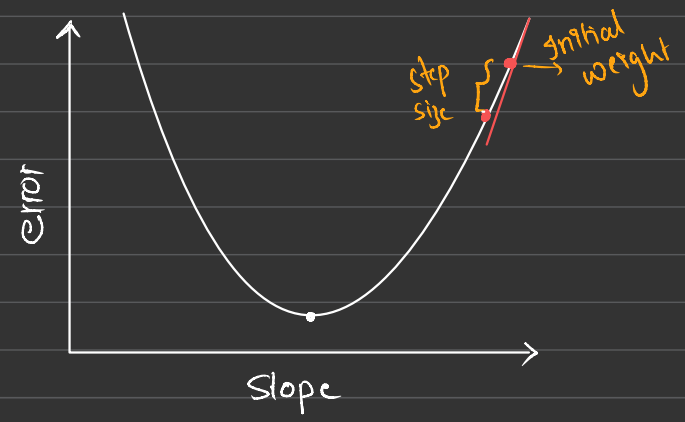
$$X^T y = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 5 & 7 & 8 \\ 1 & 5 & 3 & 6 & 7 \end{pmatrix}_{3\times5} \begin{pmatrix} 3 \\ 2 \\ 4 \\ 5 \\ 8 \end{pmatrix}$$

$$w^* = (X^T X)^{-1} X^T y = X^\dagger y = \begin{pmatrix} 22 \\ 131 \\ 111 \end{pmatrix}_{3\times1}$$

$$\begin{pmatrix} 0.50 \\ 1 \\ -0.25 \end{pmatrix} \qquad y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$y = 0.5 + 1 \cdot x_1 - 0.25 x_2$$

• Gradient Descent method (Error Detection)



Gradient → Derivative of slope.

(Minimization method)

→ It is a first order iterative optimization algorithm for finding the minimum of the error function. to achieve this goal it perform two steps iteratively

Step 1: Compute the slope ie the 1st order derivative of the current point

Step 2: Move in opposite direction of the slope starting from the current point by the computed amount

- Step size
- learning rate (η)

Two methods to check.

1) Batch Gradient Descent
2) Stochastic Gradient Descent

→ In Batch gradient D. all the training data are taken into account to take a single step, then we take the average the of the gradient to update the weight parameter.

→ Stocastic: Here we update the parameter of ea every single observation & everytime the weights are updated which is known as Iteration.

Batch	Stocastic
→ Used for Smaller Data set	→ Used for Larger Dataset.

* Derivation of Gradient Descend method.

W* to minimize the error function

$$\underbrace{\bar{w}_{k+1}}_{\text{new value.}} = \underbrace{\bar{w}_k}_{\text{old value}} - \eta \left.\frac{\partial E}{\partial \bar{w}}\right|_k$$

$$\left[w_{new} = w_{old} - \underbrace{\eta}_{\text{learning rate}} * \overbrace{\left.\frac{\partial E}{\partial w}\right|_k}^{\text{step size}} \right] \Big\} \text{Training rule of the gradient Descent.}$$

In this method we start with an arbitary initial weight vector then the weight vector parameters are updated after each iteration till the global minimum error is attained.

The gradient with respect to weight w_j

$$\frac{\partial E}{\partial w_j} = \frac{\partial}{\partial w_j}\left[\frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \left(\sum_{j=1}^{n} w_j x_j^{(i)} + w_0\right)\right)^2\right]$$

Error $e^{(i)}$ for the i^{th} sample is given by

$$e^{(i)} = y^{(i)} - \left(\sum_{j=1}^{n} w_j x_j^{(i)} + w_0\right)$$

$$\therefore \quad \frac{1}{2}\sum_{i=1}^{N}\left(\frac{\partial}{\partial w}(e^{(i)})^2\right] = \sum_{i=1}^{N} e^{(i)}\frac{\partial e^{(i)}}{\partial w_j}$$

$$\& \quad \frac{\partial e^{(i)}}{\partial w_j} = -x_j^{(i)}$$

$$\Rightarrow \quad \frac{\partial E}{\partial w_j} = \frac{\partial}{\partial w_j}\left[\frac{1}{2}\sum_{i=1}^{N}(e^{(i)})^2\right]$$

$$= -\sum_{i=1}^{N}\frac{1}{2}\times 2\, e_i \times \frac{\partial e_i}{\partial w_j}$$

$$\frac{\partial E}{\partial w_j} = -\sum_{i=1}^{N} e_i x_j$$

$$= -\sum_{i=1}^{N}\left(y^{(i)} - \left(\sum_{j=1}^{n} w_j x_j^{(i)} + w_0\right)\right) x_j$$

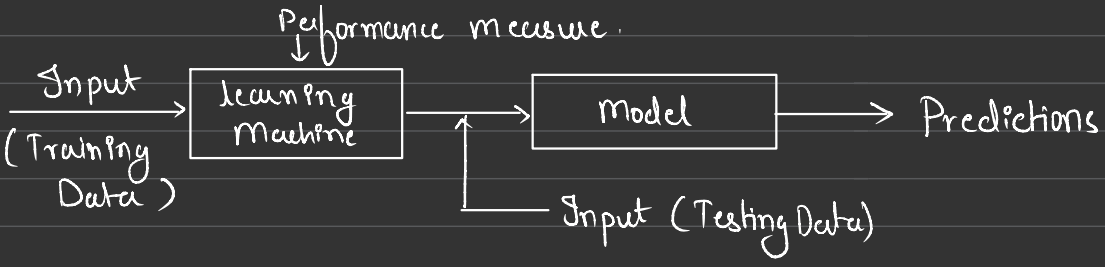
$$\frac{\partial E}{\partial w_0} = -\sum_{i=1}^{N} e^{(i)} = -\sum_{i=1}^{N}\left[y^{(i)} - \left(\sum_{j=1}^{n} w_j x_j^{(i)} + w_0\right)\right]$$

$$w_j \leftarrow w_j + \eta\sum_{i=1}^{N}\left(y^{(i)} - \left(\sum_{j=1}^{n} w_j x_j^{(i)} + w_0\right)\right) x_j$$

$$w_0 \leftarrow w_0 + \eta\sum_{i=1}^{N}\left(y^{(i)} - \left(\sum_{j=1}^{n} w_j x_j^{(i)} + w_0\right)\right)$$

Unit - 4 (Generalisation)

•> Generalisation :



→ What is Generalisation / Concept of Generalisation:

⇒ Generalisation refers to the model's ability to adopt properly to new data or previously new data drawn from the same distribution.

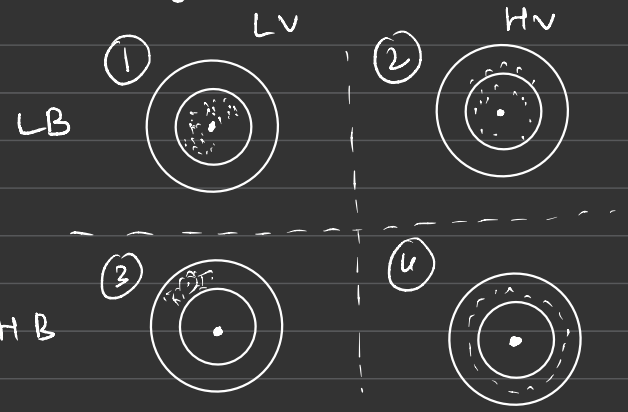
⇒ The generalization means how good our model is at learning from our given data and applying the learned information elsewhere.

•> Poor Generalisation

→ Factors

1) Bias & Variance
2) Underfitting & Overfitting.

1) Bias & Variance



① is the ideal case

Bias: Technically called as error.

Variance: How much data points are scattered between eachother.

• function used for representing Bias & Variance

⇒ h(x) is called as hypothesis function. It is the funct that best maps input to output it is also called function approximation & mapping function.

h(x) = Mapping function / approximation function.
f(x) = True function.

$$error_D[h] = E_D\{[h(x; D_j) - f(x)^2]\}$$

E_D represents the expected value or the arithemic mean of the large no. of independent values

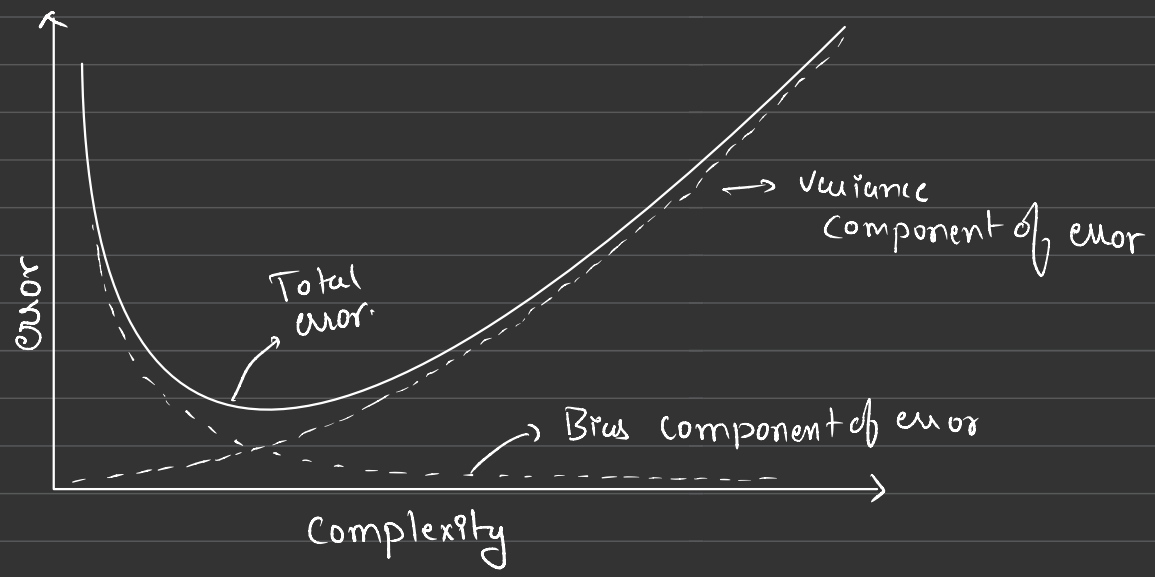
can be split into two func^n.

$$\Rightarrow \underbrace{[E_D\{h(x; D_j)\} - f(x)]^2}_{(bais)^2} + \underbrace{E_D\{[h(x; D_j) - E_D(x; D_j)]^2\}}_{Variance}$$

⇒ In the figure A in linear hypothesis function slope & intercept are two adjustable function if the experiment is repeated multiple times the estimates are scattered around 10.81, while the theoritical value is scattered around 9 for $x = 3$

⇒ for second case for a 5th order polynomial as there are many adjustable parameters the distribution of estimates $\bar{y}$ is scattered while the theoretical value y is scattered around g. So it is understand that for a simple model bias is high & variance is low but for complex model bias is low & variance is high. This is know as Bias - variance tradeoff.

•) Graph of Bias - Variance Tradeoff.

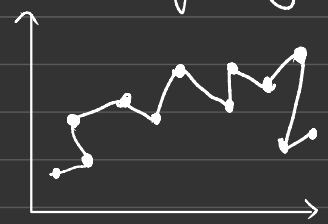


•) Occam's Razor Principle: The simpler explainations are more reasonable, and any unnecessary complexity should be shaved off.

→ It suggests hypothesis function that avoid overfitting of the training data.

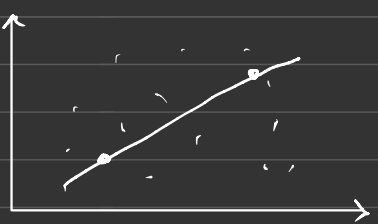
→ 'Simple' may imply needing lesser parameters, lesser training time, fewer attributes for data representation & lesser computational complexity

Overfitting



→ High variance
→ Overcomplicates
→ Covers all / maxm of the actual data

Underfitting



→ High bias.
→ Oversimplifies
→ line is not fitting properly.

• Regularization

→ Regularization Technique optimizes hypothesis complexity for a given training data set.

→ Here we look for the weights that both decrease the error & shrink words zero.

Ex = $y = 0.9 + 2x_1 + 20x_2 + 39x_2$

↓

Has predominant effect on y

* → Ridge regularization (L_2 norm)

→ Lasso regularization (L_1 norm)

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

RSS ↙ Residual sum of squares

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \text{ ----- } + \beta_p x_p$

• Ridge Regression:

→ Equation for ridge regression

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

λ = shrinkage penalty

When $\lambda = 0$, RSS will produce OLS (ordinary least square) estimate

When $\lambda \to \infty$, the impact of shrinkage penalty grows & the ridge regression coefficient estimates will approach towards zero.

Ridge Vs OLS

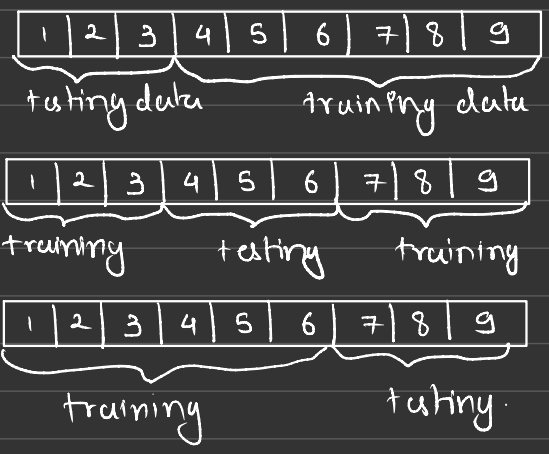
$$L_{ridge}(\hat{\beta}) = \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{m} \hat{\beta}_j^2 = ||y - x\hat{\beta}||^2 + \lambda ||\vec{\beta}||^2$$

solving

$$\hat{\beta} = (x'x + \lambda I)^{-1} (x'y)$$

$$\bar{\beta}^* = \underbrace{(xx^T)^{-1} xy)}_{\text{pseudomatrix}}$$

- Cross Validation (λ)



⇒ A commonly used technique for evaluating the success rate of a learning method taking into account a fixed data sample is called the K-Fold Cross validation.

⇒ Here training & testing is done K times. The given is divided into K subsets or folds in Iteration K one partition is set aside for testing and other partitions are employed to train the model.

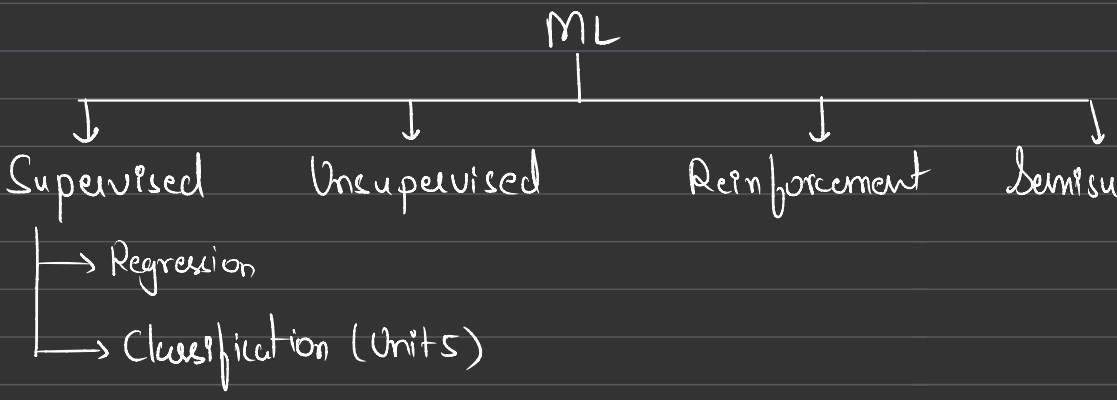
⇒ Ultimatly the K error estimate recieved from K Iterations, are averaged to give rise to an overall error estimation.

- Lasso Regression.
  - ↳ least absolute Shrinkage & selection Operator

→ Has feature selection capability.

•> Unit - 5 (Classification)

→



P(A/B) : Conditional Probability.

P(A/B) → Prob of event A given that Prob of B is already given.

$$\Rightarrow \frac{P(A \cap B)}{P(B)}$$

Ex: 40 students belong to set A who likes CI. then 30 student belongs to set B who likes ML 20 students like both CL & ML Calculate P(A/B)

$$\frac{20}{30} \Rightarrow \frac{2}{3} = 0.67$$

Bayes Theorem.

$$\therefore P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$\therefore P(A|B) \cdot P(B) = P(A \cap B) \quad -(1)$$

Similarly $P(B|A) = \frac{P(B \cap A)}{P(A)}$

$$\therefore P(B|A) \cdot P(A) = P(B \cap A) \quad -(2)$$

$$\therefore P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

posterior = $\frac{\text{Prior} \times \text{likelihood}}{\text{Evidence}}$

Here A is hypothesis & B is Evidence (data)

Q) Prob (King | face) : find out that my given card is a king and it was given that it is a face card.

$$\Rightarrow \frac{4}{12} = \frac{1}{3} \Rightarrow \frac{\text{Prob (Face | King)} \times P(K)}{P(\text{Face})} = \frac{1 \times 4/52}{12/52}$$

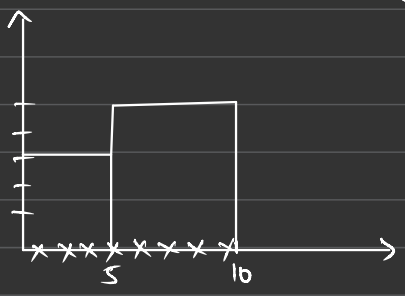
$$= 0.33$$

• Probability Density function

→ A function that represents a continous probability distribution is called as Probability Density function.

→ KNN methods estimates the density value at point x based on the distance between x and its k^{th} nearest neighbour.

• Histogram density estimation.



→ Origin = 0

→ Bin width = 5

Consider a dataset
[2, 3, 5, 7, 9, 4, 6, 8]

→ The Bin represts the input space is divided into equal size intervals named as Bins.

from 0-5 , 2 3 4
from 5-10 , 5 , 6 , 7 , 8 , 9

$$\hat{P}(x) = \frac{x^t \text{ in th same bin}}{N \times h} = \frac{\text{how many data points are in the same bin}}{\text{Total dataset} \times \text{bin width}}$$

→ The general formulation for density estimation is

$$\hat{p}(x) = \frac{k}{N V_k(x)}$$

→ k represents the nearest observation to x among the neighbors.

→ V_k represents the volume of the neighborhood around x.

The class conditional density function

$$p(x|y_q) = \frac{k_q}{N_q V_k(x)}$$

⇒ k_q represents the number of neighbours out of k nearest that belongs to class y_q

N_q represents no of samples that belongs to Class y_q

V_k represents volume of hyperspace.

$$P(y_q|x) = \frac{p(x|y_q) \cdot p(y_q)}{p(x)}$$

$$\therefore \frac{\frac{k_q}{N_q \times V_k(x)} \times \frac{N_q}{N}}{\frac{k}{N V_k(x)}} = \frac{k_q}{k}$$

$$P(y_q|x) = \frac{k_q}{k}$$

The posterior probability of an unknown sample x belong to a specific class q can be calculated as the ratio of k_q to k

Q) Given a dataset with two classes where 1st class [0,2,3] and 2nd class are [6,9,10] estimate the class label for $x = 5$.

use KNN method with $K = 3$.

$$\Rightarrow P(y_1|x=5) = \frac{K_1}{K} = \frac{2}{3} = 0.67$$

$$P(y_2|x=5) = \frac{K_2}{K} = \frac{1}{3} = 0.33$$

$\therefore$ $x = 5$ belongs to 1st class for $K = 3$.
$x = 5$ belongs to 2nd class for $K = 1$.

This is because classification in KNN method depends on the samples in the feature space.

If the samples are scattered or if the features are high dimentional the density estimation with KNN becomes problematic.

If K varied prob will vary

• KNN classification:

Problem: $K = 3$

	Age	loan	Default	Distance		Xs Dist
1	25	40000	N	102000		0.7652
2	35	60000	N	82000		0.5200
3	45	80000	N	62000		0.3160
4	20	20000	N	122000		0.9245
5	35	120000	N	22000	2	0.3428
6	52	18000	N	124000		0.6220
7	23	95000	Y	47000		0.6669
8	40	62,000	Y	80000		0.4437
9	60	100000	Y	42000	3	0.3650
10	48	220000	Y	78000		0.3861
11	33	150000	Y	8000	1	0.3771
	48	142000	?			

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} = \sqrt{(48-33)^2 + (142000-150000)^2} = 8000$$

$K = 3$ we have 2 Y & 1 N

[∴ prediction = Y]

• While calculating the distance Directly from the training set where variables have different measurement scale there is a influence on the calculated distance.

⇒ The solution is to standardize the training dataset.

• Method of standardization

$$X_s = \frac{X - Min}{Max - Min}$$ } standardize.

$$= \frac{150000 - 18000}{220000 - 18000} = 0.65$$

Unit 6: Clustering:

Unsupervised learning

⇒ Unsupervised machine learning learns from the unlabled data. It automatically finds the pattern between the training dataset and put the similar dataset in one group and another dataset in another group.

⇒ There is no specific methodology.

Clustering	Association
1) It is an unsupervised task which involves the grouping of the similar datapoints.	It is an unsupervised task that is used to find the imp relation ship b/w data points

• Clustring By K-means Approach.

⇒

Sl no.	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76

Q4 Divide the training dataset into two clusters where cluster C1 is given (185, 72) 1st row & C2 (170, 56) 2nd row by using k-means approach by taking Euclidean distance.

→ Step 1 :- Check that the remaining data points are belonging to cluster 1 or cluster 2 by using euclidean distance.

$$\therefore \quad \sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

x_0 = Observed value of height.
x_c = Centroid value of particular cluster (185, 72) & (170, 5
y_0 = Observed value of weight
y_c = Centroid value of weight

ED for 3rd row.

$$C_1 = \sqrt{(168-185)^2 + (60-72)^2}$$
$$= 20.8$$

$$C_2 = \sqrt{(168-170)^2 + (60-56)^2}$$
$$= 4.47$$

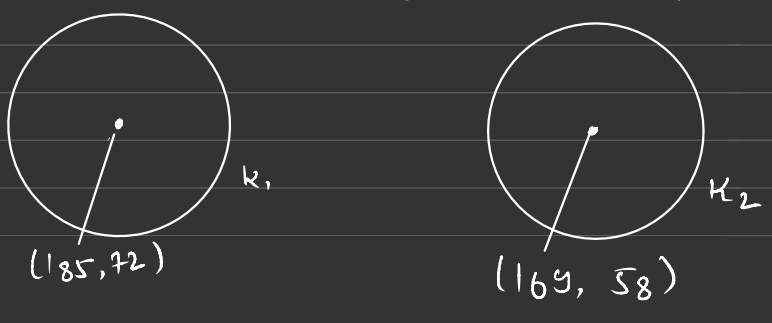
The shortest distance decides the cluster.

∴ 3rd row belongs to C2

Once a row goes to a particular cluster, always we have to compute new cluster.

Step 2 :- Re computing of centroid

$$\text{New centroid for } k_2 = \left(\frac{170+168}{2}, \frac{56+60}{2}\right)$$
$$k_2 = (169, 58)$$



Euclidian Distance for 4th row :-

$$K_1 = \sqrt{(179-185)^2 + (68-72)^2}$$
$$= 7.21$$

$$K_2 = \sqrt{(179-169)^2 + (68-58)^2}$$
$$= 14.14$$

4th row belongs to cluster 1 ∴ New centroid for cluster 1

$$\left(\frac{185+179}{2}, \frac{72+68}{2} \right)$$

Centroid for C_1 = (182, 70) { $K_1 = [1, 4]$ }

Centroid for C_2 = (169, 58) { $K_2 = [2, 3]$ }

ED for 5th row,

$$K_1 = \sqrt{(182-182)^2 + (72-70)^2} = 2$$

$$K_2 = \sqrt{(182-169)^2 + (72-58)^2} = 19.10$$

5th row belongs to cluster 1

⋮

Repeat these steps

Summary:

1) Decides the cluster.
2) Initialize the centroid
3) Calculate other datasets distance wrt the clusters.
4) Recomputing the centroid.
5) Stopping Criteria : Here K-means clustering stops when there is no change in centroid.