3) partial materialization:

selectively compute a proper subset of the whole set of possible cuboids it consider three factors

i) Identify the subset of cuboids or subcubes to materialize.
ii) use the materialized cuboids during query processing
iii) efficiently update the materialize cuboid during load and refresh.

Association Rule:

Frequent + pattern.

→ support → 2% mobile → Tempered glass
→ confidence → 60%

Ranges:

i) min-max nor : [new_min$_A$, new_max$_A$]
ii) z-score nor : [$-\infty$, ∞]
iii) " using mean absolu deviation : [0, ∞]
iv) Decimal scaling : [-1 to 1]
v) Normalization : [0, 1] & [-1 +1]

Association Rules:

$A \rightarrow B$

I_1	I_2	I_3	----------	I_n
1	0	0	1 1 0 0 0	

Support = 2% confidence = 60%

Let, $I = \{I_1, I_2, \ldots I_m\}$ be a set of items.

T → Transaction

⇒ Let 'A' be a set of items in 'T' is said to contain A if and only if $A \subseteq T$

⇒ An Association rule $A \rightarrow B$, where $A \subset I$, $B \subset I$ & $A \cap B = \phi$ let $A \rightarrow B$ holds in transaction set (T)

support $(A \rightarrow B) = P(A \cup B)$

confidence $(A \rightarrow B) = P(B/A)$

$$\text{confidence} = \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

$$= \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$$

closed frequent itemset:

* An item set X is closed in dataset S if there exist no proper super item set Y such that Y has the same support_count as X in S

* An item set X is **maximal** if there exists NO super itemset Y such that $X \subset Y$ and Y is frequent in S. (IA an ite)

Q) suppose that the database has only 2 transactions. $\{(a_1, a_2, a_3, \ldots a_{100}), (a_1, a_2, a_3, a_4, \ldots a_{50})\}$

check solution

Multi-level Association Rules:

⇒ suppose 'x' is a variable, representing a customer

buys ('x', "computer") ⟶ buys ('x', "Hp printer")

buys ('x', "laptop computer") ⟶ buys ("x", "Hp printa")

In this multilevel association rules, rules are reffered at different level of association

⇒ If the items are attribute in an association rule refer only one dimension that it is single dimension

If the rule refers two or more dimension such as age, income, buys etc then it is multi-D Association rule.

Apriori Algorithm:

→ 1994

→ R. Agarwal & R. Srikant

→ It uses level-wise search, where K items are used to explore K+1 itemset.

→ It is used for mining boolean frequent itemsets

→ It uses prior knowledge of frequent itemset

→ First, the set of frequent 1-itemset is found by scanning the database to accumulate count for each item and collecting those items that satisfy minimum support. The resulting set is denoted by L_1

→ Next, L_1 is used to find L_2 and so on...

→ Finding each L_K requires a full scan of database

Apriori property:

⇒ All non-empty subsets of a frequent itemset must also be frequent.

⇒ If an item A is added to the itemset I then the resulting itemset I∪A cannot occur more frequent than I.

⇒ This propery belongs to special category of propert called Antimonotong in the sense that if the set cannot pass a test all of it's superset will fail in the same test as well.

⇒ Apriori L_{K-1} is used to find L_K for $K \geq 2$. It is a two-step process followed by joined on prune actions

1) Join step: To find L_K a set of candidates K-itemsets is generated by joining L_{K-1} with itself. The candidates are denoted by C_K

2) Prune step: C_K is a super set of L_K. A scan of database to determine sub-count of each candidate in C_K would result in the determination of L_K

C_k can be used, to reduce the size of C_k apriori property is used. any (k-1) itemset i.e not frequent can't be a subset of frequent itemset.

Ex: $T_1 \rightarrow I_1, I_2, I_5$

$T_2 \rightarrow I_2, I_4$

$T_3 \rightarrow I_2, I_3$

$T_4 \rightarrow I_1, I_2, I_4$

$T_5 \rightarrow I_1, I_3$

$T_6 \rightarrow I_2, I_3$

$T_7 \rightarrow I_1, I_3$

$T_8 \rightarrow I_1, I_2, I_3, I_5$

$T_9 \rightarrow I_1, I_2, I_3$

C_1

I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

Scan D for count of each candidate

compare → Support-count with minimum support count

L_1

I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

Generate C_2 from L_1

C_2

I_1, I_2	4
I_1, I_3	44
I_1, I_4	1
I_1, I_5	2
I_2, I_3	4
I_2, I_4	2
I_2, I_5	2
I_5, I_4	0
I_3, I_5	1
I_4, I_5	0

compare with minimum support

L_2

I_1, I_2	4
I_1, I_3	4
I_1, I_5	2
I_2, I_3	4
I_2, I_4	2
I_2, I_5	2

Generate C_3 from L_2

$I_1\ I_2\ I_3$	2
$I_1\ I_2\ I_5$	2
$I_1\ I_2\ I_4$	1
$I_1\ I_3\ I_5$	1
$I_1\ I_3\ I_4$	0
$I_1\ I_5\ I_3$	1
$I_1\ I_5\ I_4$	0
$I_2\ I_3\ I_4$	0
$I_2\ I_3\ I_5$	1
$I_2\ I_4\ I_5$	0

compare → support-count

L_3

$I_1\ I_2\ I_3$	2
$I_1\ I_2\ I_5$	2

→ Generating rules from frequent itemset:

$$\text{Confidence}(A \rightarrow B) = \frac{\text{support count}(A \cup B)}{\text{support\_count}(A)}$$

⇒ For each frequent item set 'l', generate all non empty subsets of l

⇒ For every non-empty subset 's', of 'l' output the rule $s \rightarrow (l - s)$ where $\frac{\text{support\_count}(l)}{\text{support\_count}(s)} \geq$ min_confidence.

Suppose the data contains frequent itemset $l = \{I_1, I_2, I_5\}$ then subset of l are $\{I_1, I_2\}$, $\{I_1, I_5\}$ $\{I_2, I_5\}$ $\{I_1\}$ $\{I_2\}$ $\{I_5\}$.

$I_1 \wedge I_2 \rightarrow I_5$ conf $= \frac{2}{4} = 50\%$.

$I_1 \wedge I_5 \rightarrow I_2$ conf $= \frac{2}{2} = 100\%$

(FP Growth (FP-Tree);)

Apriori Algo suffer from 2 main problems:

1) It may need to generate a huge no. of candidate set.

2) It may need to repeatedly scan the database

FP Growth: (FP-Tree)

In this the first scan of the DB is same as apriori which derives the set of frequent items and they support_count

* Let the min support_count is 2
* The set of frequent items are sorted in the descend order of support_count the resulting set or list is denoted by 'L' and thus

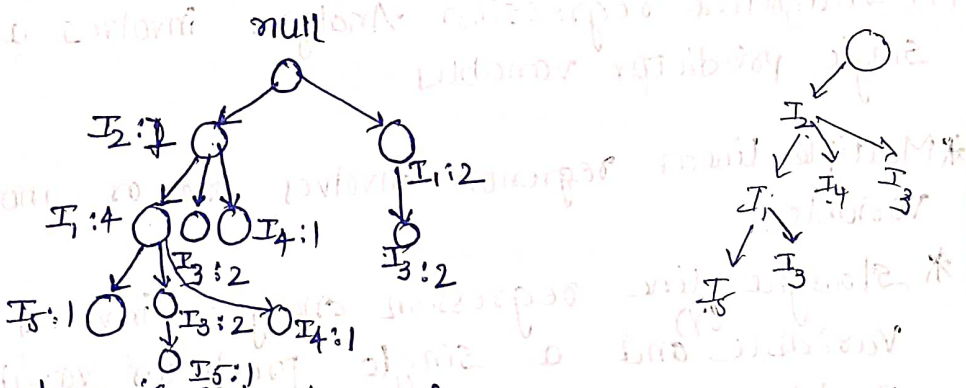
Tid	Itemsets	
T_1	I_1, I_2, I_5	I_2, I_1, I_5
T_2	I_2, I_4	I_2, I_4
T_3	I_2, I_3	I_2, I_3
T_4	I_1, I_2, I_4	I_2, I_1, I_4
T_5	I_1, I_3	I_1, I_3
T_6	I_2, I_3	I_2, I_3
T_7	I_1, I_3	I_1, I_3
T_8	I_1, I_2, I_3, I_5	I_2, I_1, I_3, I_5
T_9	I_1, I_2, I_3	I_2, I_1, I_3

$L = \{\{I_2:7\}, \{I_1:6\}\}, \{I_3:6\}, \{I_4:2\}, \{I_5:2\}\}$

* An FP tree is constructed as follows.

1) Firstly, create root of the tree labelled with null.

2) Scan the DB 'D', for the second time the items in each transaction are processed, in the 'L' order i.e, sorted a/c to decreasing support term and the branch is created for each transaction.



3) An FP tree is mined as follows

⇒ Start from each frequent one. pattern (as an initial suffix pattern), construct it's conditional pattern base then construct it's conditional FP-tree and perform mining recursively in such a tree.

⇒ The pattern growth is acheived by contactination of the suffix pattern with the frequent pattern generated from the conditional FP-tree.

Item	Conditional patten Base	Conditional Fp tree	frequent patt generate
I_5	$\{I_2, I_1:1\}$ $\{I_2, I_1, I_3:1\}$	$\{I_2:2, I_1:2\}$	$\{I_2, I_5:2\}$ $\{I_2, I_1, I_5:2\}$
I_4	$\{I_2:1\}$ $\{I_2, I_1:1\}$	$\{I_2:2\}$	$\{I_2, I_4:2\}$
I_3	$\{I_2, I_1:2\}$ $\{I_2:2\}$ $\{I_1:2\}$	$\{I_2:4, I_1:2\}$	$\{I_2, I_3:4\}$ $\{I_2, I_3:4\}$ $\{I_2, I_1, I_3:2\}$
I_1	$\{I_2:4\}$	$\{I_2:4\}$	$\{I_2, I_1:4\}$

Regression:

* Regression can be used to model the relationship b/w one or more independent or predictor variables and a dependent are response variables predictor variables describes the tuples (making up the attribute vector)
* In general, the values of predictor variables are known Response variable is what we want to predict
* A straightline regression Analysis involves a state single predictor variables
* Multiple linear regresion involves one or more predictor variables.
* Straight line regression analysis involves a response variable (y) and a single predictor variable (x). it models y as a linear function of x.

$$y = b + wx$$

where b, w are regression co-efficients.

* Regression coefficients can be thought of as a weight so we can equivalently write $y = w_0 + w_1 x$
* These co-efficients can be solved by the method of least square, which estimates the best fit line which minimises the error between the actual data and the estimated of line.

* Regression co-efficient can be estimated by

$$\Rightarrow w_1 = \frac{\sum_{i=1}^{|D|} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|} (x_i - \bar{x})^2}$$

Write on desk

$$w_0 = \bar{y} - w_1 \bar{x}$$

x year Experience	Salary (k€)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Find the Salary of person with 10 year of experience.

$$\bar{x} = \frac{3+8+9+13+3+6+11+21+1+16}{10}$$

$$= 9.1$$

$$\bar{y} = \frac{30+57+64+72+36+43+59+90+20+83}{10}$$

$$= 55.4$$

$$w_1 = \frac{(3-9.1)(30-55.4) + (8-9.1)(57-55.4) + (9-9.1)(64-55.4) + (13-9.1)(72-55.4) + (3-9.1)(36-55.4) + (6-9.1)(43-55.4) + (11-9.1)(59-55.4) + (21-9.1)(90-55.4) + (1-9.1)(20-55.4) + (16-9.1)(83-55.4)}{(3-9.1)^2 + (8-9.1)^2 + (9-9.1)^2 + (13-9.1)^2 + (3-9.1)^2 + (6-9.1)^2 + (11-9.1)^2 + (21-9.1)^2 + (1-9.1)^2 + (16-9.1)^2}$$

$$w_1 = 3.5$$

$\Rightarrow w_0 = \bar{y} - w_1 \bar{x}$

$= 55.4 - (3.5)(9.1)$

$= 55.4 - (3.5 \times 9.1)$

$= 23.5$

$y = 23.5 + 55.4\,(10)$

$= 23.5 + 554$

$= 577.5$

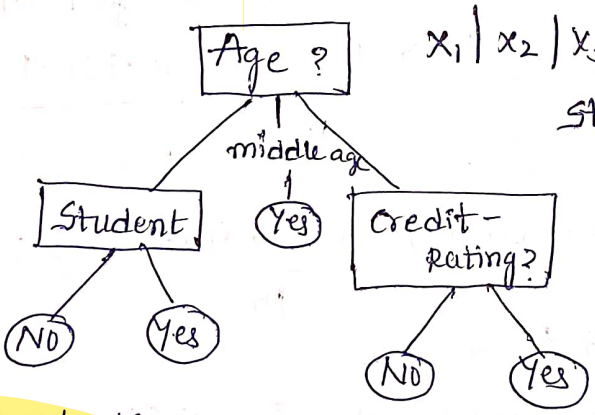
554.
23.5
577.5

Decision tree:

A decision tree is a flowchart by a tree structure where each internal node denotes a test or attribute, each branch represents an outcome of test an each leaf node (terminal node) holds a class label. the top most node in the tree is a root node.

Internal nodes represented by Rectangle

Leaf nodes " " ovels.



x_1	x_2	x_3	...	Class label	age
		Student		credit Rating	

Attribute selection methods:

- Information gain
- Gain ratio
- Gini Index

* Attribute selection measure determines how the tuple at a given node are to be split
* the attribute having the best score for measure is choosen as the splitting attribute for the given tuple.

Information Gain:

* The node with the highest info gain is choosen as the splitting attribute for node (N)
* Expected information needed to classify a tuple is given by

$$\text{Info}(D) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

where p_i = probability that an arbitrary tuple belongs to class C_i

p_i is estimated by $\frac{|C_i, D|}{|D|}$ Info D is also known as Entropy of D

Write on desk

* $\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info}(D_j)$

where $|D_j|$ is estimated as weight of Jth partition

* $\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$. (Information Gain)

Q)

Age (u), Income (high) student (NO) Credit rating Class (NO)

Age	Income	Student	Credit rating	Class Buy-computer
Youth	High	NO	Fair	NO
Youth	High	NO	Excellent	NO
Middle	High	NO	Excellent	NO
Senior	Medium	NO	Fair	Yes
Senior	Low	Yes	Fair	Yes
Senior	Low	Yes	Fair	Yes
Middle	Low	Yes	Excellent	NO
Youth	Medium	NO	Excellent	yes.
Youth	Low	Yes	Fair	NO
Senior	medium	Yes	Fair	Yes
Youth	medium	Yes	Fair	Yes
Middle	medium	NO	Excellent	Yes
Middle	High	Yes	Excellent	Yes
			Fair	Yes

Senior	medium	NO	excellent	NO

$$\text{Info}(D) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log\frac{5}{14}\right).$$

$$\text{Info}_{age}(D) = \frac{5}{14}\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14}\left(-\frac{4}{4}\log_2 * \right.$$

$$\left. - \frac{0}{4}\log_2\frac{0}{4}\right) + \frac{5}{14}\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$

$$\simeq 0.694$$

$$\text{Gain (Age)} = \text{Info}(D) - \text{Info}_{age}(D).$$

$$= (-0.694 + 0.94)$$

$$= 0.246$$

Gain (Income) = 0.029

Gain ratio:

It is an extension of Information gain it attempts to overcome this bias it applies a kind of normalization to info gain using split information

$$\Rightarrow \text{Split info}_A(D) = \sum_{j=1}^{v} \frac{D_j}{D}\log_2\frac{|D_j|}{|D|}$$

minus

write on desk

Gain ratio is defined as

$$\Rightarrow \text{Gain ration}(A) = \frac{\text{Gain}(A)}{\text{Split Info}(A)}$$

The attribute with the max. Gain ratio is selected as the splitting attribute

$$\text{Split Info}(D) = -\frac{4}{14}\log_2\frac{4}{14} - \frac{6}{14}\log_2\frac{4}{14} - \frac{4}{14}\log_2\frac{4}{14}$$

Income $\simeq 1.57$

Gain Income = 0.029.

$$\text{Gain ratio (Income)} = \frac{0.029}{1.58}$$

Gini Index:

* It is used in CART, gini index measures the impurity of data

$$\text{Gini}(D) = 1 - \sum_{i=1}^{n} p_i^2$$

where, p_i is the probability that a tuple in D belongs to the class C_i
it is estimated by $\frac{C_i}{D}$

* The gini index considers a binary split of each attribute if a binary split on attribute A partions (D) into D_1 and D_2. The gini index of D is calculated as

$$\Rightarrow Gini_A(D) = \frac{|D_1|}{D} Gini(D_1) + \frac{|D_2|}{D} Gini(D_2)$$

The reduction in impurity is calculated as

$$\Rightarrow \boxed{\Delta Gini(A) = Gini(D) - Gini_A(D)}$$

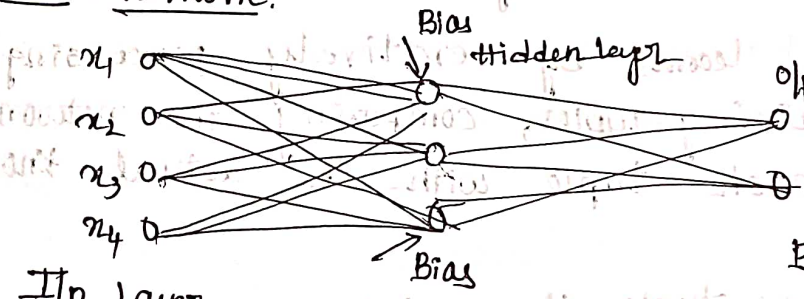
$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2$$

$$= 0.459.$$

$$Gini_{credit\_rating}(D) = \frac{8}{14}\left(1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2\right) + \frac{6}{14}\left(1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2\right)$$

$$\simeq 0.428$$

Neural network:



⇒ NN is a set of connected input output units in which each connection has a weight associated with it

⇒ During the learning phase the network learns by adjusting weight associated with it so as to predict the corr class labels of input labels.

⇒ Neural network layer is also referred to as Connectionist learning due to the connection b/w units. Neural network involves

⇒ NN involves long training time and are therefore more suitable for applications where this is feasible.

⇒ The most popular Neural network Algo is back propagation.

⇒ The back propagation Algo. performs learning on a multi layer feed forward neural network.

⇒ It iteratively learns a set of weights for prediction of class label of tuples.

⇒ A multi-layer feed forward NN consists of an input layer one or more hidden layer and an Output layer.

⇒ Each layer is made up of units the input to the network corresponds to attribute measured for each training tuple.

⇒ The Input are fed simultaneously into the units making up the input layer.

⇒ Back propagation learns by iteratively processing a dataset of training tuples, comparing the network prediction of each tuple with the actual known target value.

⇒ For each training tuple the weights are modified so as to minimize the mean-squared error b/w the network prediction and the actual known target value.

⇒ These modification are made in the backward direction i.e from the O/p layer through each hidden layer upto the 1st hidden layer hence it is called back propagation.

$$I_j = \sum w_{ij} O_i + \theta_j$$

w_{ij} = weight of connection from unit i in previous layer to unit j.

O_i = Output of the unit i from the previous layer

θ_j = Bias of the unit

used to vary the activity of unit.

⇒ Each unit in the hidden layer and the o/p layer takes it's net input and then apply the activation function to it.

⇒ The sigmoid func is generally used for this purpose given the net input I_j to unit j then O_J the Output of unit J is computed as

$$O_J = \frac{1}{1+e^{-I_J}}$$

⇒ This function also referred to as squashing coefficient function. Because it maps large into a smaller range (from 0 to 1).

Error is calculated as follows

Last layer: $Err_J = O_J(1-O_J)(T_J - O_J)$

T_J = known target value.

Hidden layer $Err_J = O_J(1-O_J)\sum Err_K w_{JK}$

w_{JK} = weight of the connection from unit J to K

⇒ The weights and bias are updated to reflect the propagated error. weights are updated by the following equation.

$$\Delta w_{ij} = l\, Err_J O_i$$
$$w_{ij} = w_{ij} - \Delta w_{ij}$$

where l = learning rate that technical varries (0.02 to 1.0).

⇒ Bias are updated by the following equation

$$\Delta \theta_J = (l)\, Err_j$$
$$\theta_J = \theta_J - \Delta \theta_J$$

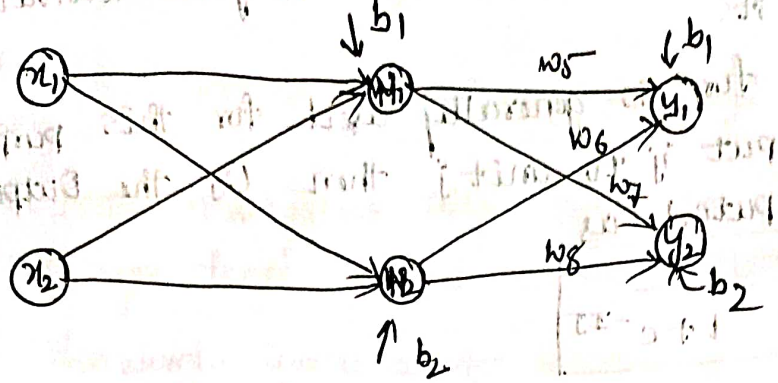
Chain Rule: $y = f(x) = \frac{1}{1+e^{-x}}$

$$f'(x) = \frac{e^{-x}}{(1+e^{-x})^2} \quad \left[\frac{dy}{dx} = \frac{u'v - v'u}{v^2}\right]$$

$$f(x)\cdot(1-f(x)) = \frac{e^{-x}}{(1+e^{-x})^2}$$

$$f'(x) = f(x)(1 - f(x))$$

Q)



$x_1 = 0.05$ $\quad b_1 = 0.35$ $\quad T_1 = 0.01$

$x_2 = 0.10$ $\quad b_2 = 0.6$ $\quad T_2 = 0.99$

$w_1 = 0.15$ $\quad w_5 = 0.4$

$w_2 = 0.2$ $\quad w_6 = 0.45$

$w_3 = 0.25$ $\quad w_7 = 0.5$

$w_4 = 0.3$ $\quad w_8 = 0.55$

$$H_1(in) = x_1 w_1 + x_2 w_2 + b_1 \simeq 0.377$$

$$H_1(out) = \frac{1}{1+e^{-H_1(in)}} \simeq 0.5932$$

$$H_2 = 0.596$$

$$y_1(in) = w_5 H_1(out) + w_6 H_2(out) + b_2$$

$$y_1(out) = \frac{1}{1+e^{-y(output)}}$$

$$E_{total} = \frac{1}{2} \Sigma (Target - Actual\ o/p)^2$$

$$= \frac{1}{2}(T_1 - y_{1out})^2 + \frac{1}{2}(T_2 - y_{2out})^2$$

$$E_{total} = E_1 + E_2$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_1}{\partial w_5} + \frac{\partial E_2}{\partial w_5}$$

$$\frac{\partial E_1}{\partial w_5} = \frac{\partial E_1}{\partial y_1(out)} \times \frac{\partial y_1(out)}{\partial y_1(in)} \times \frac{\partial y_1(in)}{\partial w_5}$$

$$= (y_{1\,output} - T_1)\,[y_1(out)(1 - y_1(out))]$$

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_1}{\partial w_1} + \frac{\partial E_2}{\partial w_2}$$

$$\frac{\partial E_1}{\partial w_1} = \frac{\partial E_1}{\partial y_1(out)} \times \frac{\partial y_1(out)}{\partial y_1(in)} \times \frac{\partial y_1(in)}{\partial H_1(out)} \times \frac{\partial H_1(out)}{\partial H_1(in)} \times \frac{\partial H_1(in)}{\partial w_1}$$

Assign

[Diff b/w OLAP and OLTP?]

Baye's classifier:

It is a supervised classification technique the model assumes all input attributes are of equal importance and independent of one another.

Baye's classifier is based on the bayes theorem which is stated as $P(H|E) = \frac{P(E|H)\,P(H)}{P(E)}$

H is a hypothesis to be tested

E is the evidence associated with H

$P(\frac{A}{B})$ it is the probability of A given that B already occurs $P(B) > 0$.

P(H) = is a prior probability which denotes the prob of hypothesis before presentation of any evidence.

Q

Magazine promotion	watch promotion	Life Insurance	credit card	Gender class
Yes	NO	NO	NO	M
Yes	Yes	Yes	Yes	F
NO	NO	NO	NO	M
Yes	Yes	Yes	Yes	M
Yes	NO	Yes	NO	F
NO	NO	NO	NO	F
Yes	Yes	Yes	Yes	M
NO	NO	NO	NO	M
Yes	NO	NO	NO	M
Yes	Yes	Yes	NO	F

Which the above given data find gender of tuple where magazine promotion = Yes, W→S, L→N, C→N

$$\left(\frac{P(\text{Gender}=\text{Male})}{E}\right) = \frac{P(E \mid \text{Gender}=\text{Male})\; P(\text{Gender}=\text{male})}{P(E)}$$

$\Rightarrow$ $P(MP=\text{yes} \mid \text{Gender}=\text{male}) = \frac{4}{6}$

$\Rightarrow$ $P(WP=\text{yes} \mid \text{Gender}=\text{male}) = \frac{2}{6}$

$\Rightarrow$ $P(LI=\text{No} \mid G=M) = \frac{4}{6}$

$\Rightarrow$ $P(CC=\text{No} \mid G=M) = \frac{4}{6}$

$$\left(\frac{P(G=M)}{E}\right) = \frac{4}{6} \times \frac{2}{6} \times \frac{4}{6} \times \frac{4}{6} \times \frac{6}{10} = 0.059$$

$$\Rightarrow \left(\frac{P(\text{Gender}=\text{female})}{E}\right) = \frac{P(E \mid G=F)\; P(G=M)}{P(E)}$$

$P(MP=\text{Yes} \mid G=F) = \frac{3}{4}$

$P(WP=\text{Yes} \mid G=F) = \frac{2}{4}$

$P(LI=\text{No} \mid G=F) = \frac{1}{4}$

$P(CC=\text{No} \mid G=F) = \frac{3}{4}$

$\Rightarrow \frac{3}{4} \times \frac{2}{4} \times \frac{1}{4} \times \frac{3}{4} \times \frac{4}{10} = 0.281$

Male has high probability, Gender = "Male"

Q)

Outlook	temperature	humidity	windy	play class.
sunny	85	85	F	No
sunny	80	90	T	No
overcast	83	86	F	Yes
rainy	70	96	F	Yes
rainy	68	80	F	Yes
rainy	65	70	T	No
overcast	64	65	T	Yes
sunny	72	95	F	No
sunny	69	70	F	Yes
rainy	75	80	F	Yes
sunny	75	70	T	Yes
overcast	72	90	T	Yes
overcast	81	75	F	Yes
rainy	71	91	T	No

Find the values of the class tuple where attribute values are as follows Output Look = overcast, temperature = 16, humidity = 62 windy = false.

write on desk

Ans $$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

μ - temp - Yes = 73
μ - hum - Yes = 79.1
σ - tem - Yes = 6.2
σ - hum - Yes = 10.2

μ - temp - NO = 74.6
μ - hum - NO = 8.0
σ - tem - yes = 8.0
σ - hum - yes = 9.7

$$f(\text{temp} = 60 \mid \text{Yes}) = \frac{1}{6.2\sqrt{2\pi}} e^{-\frac{(60-73)^2}{2(6.2)^2}}$$

$$f(\text{temp} = 60 \mid \text{NO}) = \frac{1}{8\sqrt{2\pi}} e^{-\frac{(60-74.6)^2}{2(8)^2}}$$

$f(\text{hum} = 62 \mid \text{Yes}) = 0.0096$
$f(\text{hum} = 62 \mid \text{NO}) = 0.0018$

$P(\text{Yes}) = \frac{9}{14}$ $P(\text{NO}) = \frac{5}{14}$

$P(\text{Outlook} = \text{overcast} \mid \text{Yes}) = \frac{4}{9}$
$P(\text{Outlook} = \text{overcast} \mid \text{NO}) = \frac{0}{5}$
$P(\text{windy} = \text{false} \mid \text{Yes}) = \frac{6}{9}$
$P(\text{windy} = \text{false} \mid \text{NO}) = \frac{2}{5}$

$\frac{4+1}{9+3}$
$\frac{0+1}{5+3}$
} laplace correction.

"clustering" is the process of grouping the data into classes of cluster, so that objects within a cluster have high similarity in comparison to another, but of varying dissimilar to objects in other clusters

Measurement of similarity:

* categorical variable
* Oblique nominal variable.

⇒ Consider 2 instances x and y which consists of n nomical attributes then the distance b/w is measured as follows

$$d(x,y) = \frac{n-m}{n}$$ where m = no. of matches

⇒ Consider 2 instances x and y that consists of n-numeric attributes. it can be represented as vector for $x_1, x_2, x_3 \dots x_n$

$y_1, y_2, y_3 \dots y_n$

The distance b/w x and y can be determined by $d(x,y) = (|x_1-y_1|^q + |x_2-y_2|^q + \dots |x_n-y_n|^q)^{1/q}$

cosine base similarity in the vector form

$$\cos(x,y) = \frac{x \cdot y}{|x||y|}$$

Clustering methods:

→ partitioning methods
- → K-means
- → K-mediods / PAM. (partitioning Around mediods)

K-means : It works only for numeric attributes

Squared error is used to determine clustering quality.

id	name	Height	weight
x_1	Ram	64	60
x_2	Shyam	60	61
x_3	Gita	59	70
x_4	Mohan	68	71

$d(x_3, x_1) = \sqrt{(59-64)^2 + (70-60)^2} = 11.18$

$d(x_3, x_2) = 9.06$

$d(x_4, x_1) = 11.7$

$d(x_4, x_2) = 12.81$

$E = d(x_2, x_3)^2 + d(x_1, x_4)^2 \simeq 219$

Iteration 2

$C_1 \rightarrow \left(\frac{64+68}{2}, \frac{60+71}{2}\right)$ A

$C_2 \rightarrow \left(\frac{60+59}{2}, \frac{61+70}{2}\right)$ B.

A = 66, 65.5

B = 59.5, 65.5

$d(x_1, A) = 8.5.85$

$d(x_1, B) = 7.1$

$d(x_2, A) = 7.5$

$d(x_2, B) = 4.53$

$d(x_3, A) = 8.32$

$d(x_3, B) = 4.53$

$d(x_4, A) = 5.85$

$d(x_4, B) = 10.12$

$E = d(x_1, A)^2 + d(x_4, A)^2 + d(x_2, B)^2 + d(x_3, B)^2$

$= 109.49.$

K-medoids:

⇒ This algo ensures that it selects initial k prototypes arbitrarily the absolute error criteria is used to determine the clustering quality

⇒ In each iteration the prototype of each cluster is resigned to an actual dataset of each point that minimizes the absolute error criteria.

Iteration 1:

$E = d(x_2, x_3) + d(x_1, x_4) = 20.72$

Iteration 2: The Algo. selects new midoids in place of existing midoids. i.e, either x_3 or x_4 selected in place of x_1 or x_2

⇒ consider the case where x_4 is selected in place x_1

x_2 x_1

x_4 x_3

$d(x_1, x_2)$ *

$d(x_1, x_4)$

$d(x_3, x_2)$

$d(x_3, x_4)$ *

$E = d(x_1, x_2) + d(x_3, x_4).$

⇒ In this the first step to create similarity matrix
⇒ The similarity matrix is the pair of table of all pair wise distances or degree of similarity between clusters. Initially it contains pair-wise distances between individual pair of records.

⇒ Distance between the cluster can be found using 3 different approaches

1) Single linkage
2) complete linkage
3) Centroid linkage.

Q)

	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	2	8	0

→

	35	1	2	4
35	0			
1	11	0		
2	10	9	0	
4	9	6	5	0

→

	35	24
35	0	
24	10	0
1	11	9

→

	35	241
35	0	
241	11	0