⇒ Correlation Coefficient (Pearson's Product moment Coefficient)

$$\gamma_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n \sigma_A \sigma_B}$$

$$\sigma_A = \sqrt{\frac{\Sigma(A-\bar{A})^2}{n}} \quad , \quad \sigma_B = \sqrt{\frac{\Sigma(B-\bar{B})^2}{n}}$$

⇒ Covariance (Numeric Data)

$$Cov(A,B) = \sum_{i=1}^{n} \frac{(a_i - \bar{A})(b_i - \bar{B})}{n}$$

$$\gamma_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$$

⇒ Correlation Analysis (Nominal Data)
(Chi-square (χ^2) test)

$$\chi^2 = \sum \frac{(\text{observed} - \text{Expected})^2}{\text{Expected}}$$

⇒ Normalization (Min-Max Normalization)

$$v_i' = \frac{v_i - min_A}{Max_A - Min_A} (new\text{-}max_A - new\text{-}min_A) + new\text{-}min_A$$

⇒ Z-score Normalization

$$v' = \frac{v - \mu_A}{\sigma_A}$$

μ_A = mean of A
σ_A = Standard deviation.

⇒ Normalization By Decimal Scaling

$$v' = \frac{v}{10^j}$$

$j \Rightarrow (Max|v'|) < 1$

{99, 9}
w
↓
2
$j = 2$

(10^2)

⇒ Binning :- By Equi width Partitioning

$$\text{width} = \frac{\max - \min}{K} \quad K \rightarrow \text{intervals of same size}$$

⇒ By Equi depth Partitioning → Number of values in each bin K = 3

Equidepth

⇒ Smoothing by Bin Means :- Replace each value in bin with mean of that bin

⇒ Equidepth (by Median) :- Replace each value with the middle value of given range.

⇒ Equidepth by boundaries) :- Replace each value in bin with nearest boundary value.

⇒ Mean (Measuring Central tendency)

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \Rightarrow \frac{x_1 + x_2 \; - - - x_n}{N}$$

⇒ weighted Arithmetic Mean

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

⇒ Trimmed Mean :- Mean obtained after chopping off values at high and low extremes.

⇒ Median :- Middle value if odd number of values.
→ Average of middle two values if even number of values.

⇒ for grouped data :-

$$\text{Median} = L_1 + \left(\frac{\frac{n}{2} - (\Sigma \text{freq})_l}{\text{freq}_{\text{median}}}\right) \times \text{width}$$

$L_1 \rightarrow$ lower boundary of median interval

$N \rightarrow$ Number of values in data set.

$\Sigma \text{freq}_L \rightarrow$ Sum of frequencies of all intervals that are lower than the median interval.

$\text{freq}_{\text{median}} \rightarrow$ freq of median interval, width $\rightarrow$ width of median interval.

⇒ Mode:- Value that occurs most frequently in data.

For unimodal

mean - mode = 3 × (mean - median)

⇒ Range:- Difference between Max and Min values in set of observations

⇒ Quantiles :- $Q_1 = 25^{th}$ percentile, Q_3 (75^{th} percentile)

$Q_2 \rightarrow 50^{th}$ percentile (Median)

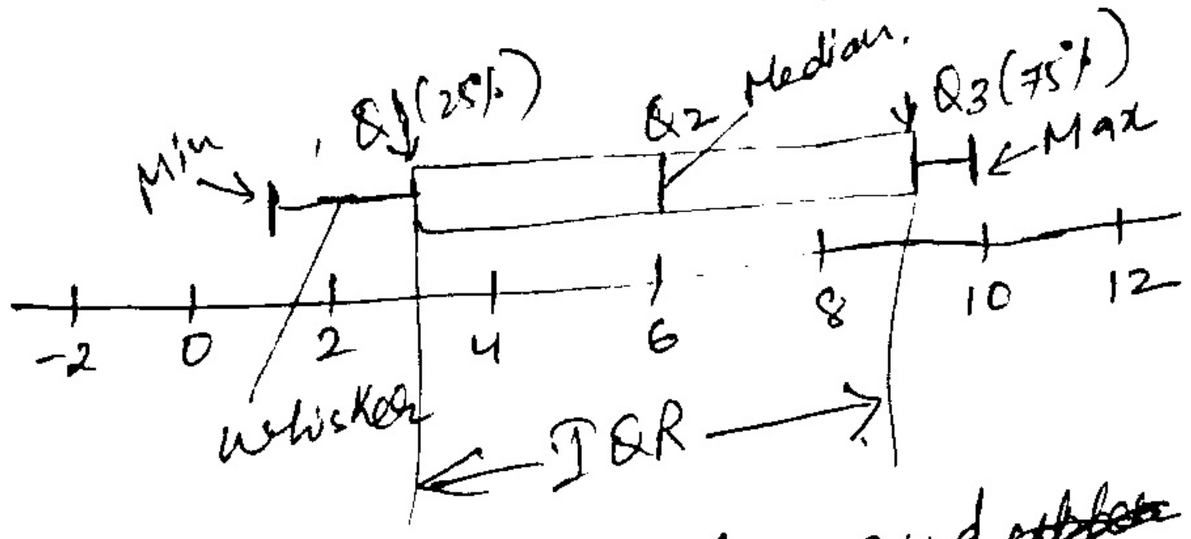
⇒ InterQuantile Range :- $IQR = Q3 - Q1$

⇒ Five Number Summary :- min, Q_1, median, Q_3, max

$Q_1 = 25^{th}\% ile = \frac{25}{100}(n-1)$ th term element

$Q3 = 75^{th}\% = \frac{75}{100}(n-1)$ th term element

⇒ BoxPlot



Outliers → datapoints → below and ~~above~~ above the lower and upper limit:- lower = Q1 - 1.5 IQR

upper = Q3 + 1.5 × IQR

⇒ Variance $V = \sigma^2 = \frac{1}{n}\sum_{i=1}^{N}(x_i - \bar{x})^2$ [$\bar{x} \rightarrow$ mean]

⇒ Standard Deviation :- $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(x_i - \bar{x})^2}$

$\sigma = \sqrt{V}$

⇒ Dissimilarity Matrix

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ \vdots & \vdots & \vdots & \\ d(n,1) & d(n,2) & \cdots & 0 \end{bmatrix}$$

$d(i,j) = d(j,i)$

$Sim(i,j) = 1 - d(i,j)$

⇒ Proximity measure of nominal attributes

$$d(i,j) = \frac{P-m}{P}$$ ⇒ P → Total Number of attributes

M → No. of matches

Id	Type 1	Type 2
1	10	A
2	30	B
3	10	A
4	20	C

$d(2,1) = \frac{2-0}{2} = 1$

$d(3,1) = \frac{2-2}{0} = 0$

$d(4,1) = \frac{2-0}{2} = 1$

$d(3,2) = \frac{2-0}{2} = 1$

$d(4,2) = \frac{2-0}{2} = 1$

$d(4,3) = \frac{2-0}{2} = 1$

$$\text{dissimilarity Matrix} = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

→ d(3,1) are similar.

(3,1)

$$Sim(i,j) = 1 - d(i,j) = \frac{m}{p}$$

⇒ Proximity Measure of Binary attributes

⇒ For Asymetric Binary data

$$d(i,j) = \frac{r+s}{q+r+s}$$

⇒ For Symmetric Binary data

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

Jacquard coefficient

⇒ Similarity $S(i,j) = \frac{q}{q+r+s}$ for assymetric

$$\text{Coherence}(i,j) = \frac{q}{(q+r)+(q+s)-q}$$

Name	T_1	T_2	T_3	T_4	T_5	T_6
X	1	0	1	0	0	0
Y	1	0	1	0	1	0
Z	1	1	0	0	0	0

object i \ object j	1	0
1	q	r
0	s	t

q → {i=1, n=1}

r → {i=1, j=0}

s → {i=0, j=1}

t = {i=0, j=0}

i → x, j → y

q → 2, r → 0, s → 1

$d(X,Y) = \frac{r+s}{q+r+s} = \frac{0+1}{2+0+1} = 0.33$

d(X,Z) = 0.67

d(Y,Z) = 0.75

d(X,Y) are more similar

→ Minkowski Distance :-

$$d(i,j) = \sqrt[h]{|x_{i1}-x_{j1}|^h + |x_{i2}-x_{j2}|^h + \cdots + |x_{ip}-x_{jp}|^h}$$

where $i = (x_{i1}, x_{i2}, \ldots x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots x_{jp})$ are, two p dimensional data objects, 'h' is the order (the distance so defined is also called h-h norm)

→ $d(i,j) > 0$ if $i \neq j$ and $d(i,i) = 0$

→ $d(i,j) = d(j,i)$ → Symmetry

→ $d(i,j) \leq d(i,k) + d(k,j)$

∴ $h = 1$:- Manhattan (City block, L_1 norm) distance

$$d(i,j) = |x_{i1}-x_{j1}| + |x_{i2}-x_{j2}| + \cdots + |x_{ip}-x_{jp}|$$

$h = 2$:- (L_2 norm) → Elucidien distance

$$d(i,j) = \sqrt{(|x_{i1}-x_{j1}|^2 + |x_{i2}-x_{j2}|^2 + \cdots + |x_{ip}-x_{jp}|^2}$$

$h \to \infty$:- Supremum (L_{max} norm, L_∞ norm) → Max difference

$$d(i,j) = \lim_{h\to\infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{1/h}$$

$$= \max_{f}^{p} |x_{if} - x_{jf}|$$

Example

Point	attribute1	attribute2
x_1	1	2
x_2	3	5
x_3	2	0
x_4	4	6

Manhattan (L_1)

L_1	x_1	x_2	x_3	x_4
x_1	0			
x_2	5	0		
x_3	3	6	0	
x_4	6	1	7	0

Euclidean (L_2)

L_2	x_1	x_2	x_3	x_4
x_1	0			
x_2	3.61	0		
x_3	2.24	5.1	0	
x_4	4.24	1	5.39	0

Supremum

L_∞	x_1	x_2	x_3	x_4
x_1	0			
x_2	3	0		
x_3	2	5	0	
x_4	3	1	5	0

⇒ Proximity Measure of ordinal attributes

Obj	T_1
1	High
2	low
3	medium
4	high

High ⇒ 1
Medium ⇒ 2
low ⇒ 3
Normalize thorough [0, 1]

$Z_{if} = \frac{r_{if} - 1}{M_f - 1}$

→ $M_f \Rightarrow 3$

for high = $\frac{1-1}{3-1}$

$= 0/3 = 0$

1 → 0.0, 2 → 0.5, 3 → 1.0

Dissimalarity Matrix $d(2,1) = (1-0) = 1$, $d(3,1) = (0.5-0) = 0.5$

$d(4,1) = (0-0) = 0$, $d(3,2) = (0.5-1) = 0.5$

$d(4,2) = (1-0) = 1$

$d(4,3) = (0.5-0) = 0.5$

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1 & 0.5 & 0 \end{bmatrix}$$

⇒ Proximity Measure of Mixed attributes:-

⇒ A database may contain all attribute types of Nominal, Symmetric binary asymmatric binary numeric, ordinal.

Combined → $d(i,j) = \frac{\sum_{f=1}^{p} \delta_{if}^{(f)} d_{if}^{(f)}}{\sum_{f=1}^{p} \delta_{ff}^{(f)}}$

→ f → is binary or nominal
$d_{ij}(f) = 0$ if $x_{if} = x_{jf}$
or $d_{ij}(f) = 1$ otherwise

→ f is numeric → use the normalized distance

→ f is ordinal:-
Compute ranks r_{if} and
Treat z_{if} as interval-scaled

$$Z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

Example :- Based on the information given, find most similar and most dissimilar persons among them, Apply min-max normalization on income to obtain [0, 1] range. Consider profession and mother tongue as nominal consider native place as ordinal variable with ranking order of [village, small town, Suburban, Metropolitan], Give equal weight to each attribute

Name	Income	Profession	Mother tongue	Native Place
Ram	70000	Doctor	Bengali	village
Balram	50000	Data Scient	Hindi	small town
Bharat	60000	Carpenter	Hindi	Suburban
Kishan	80000	Doctor	Bhojpuri	Metropolitan

→ After normalizing and quantifying Native Place

Name	Income	Prof	Mother tongue	Native
Ram	0.67	Doctor	Bengali	1
Balram	0	Data Sci	Hindi	2
Bharat	0.33	Carpenter	Hindi	3
Kishan	1	Doctor	Bhojpuri	4

$d(Ram, Balram) = 0.67+1+1+\frac{(2-1)}{(4-1)} = 3$,

$d(Ram, Bharat) = 0.33+1+1+\frac{(3-1)}{4-1} = 3$

$d(Ram, Kishan) = 0.33+0+1+\frac{(4-1)}{(4-1)} = 2.33$,

$d(Balram, Bharat) = 0.33+1+0+\frac{(3-2)}{(4-1)} = 1.67$

$d(Balram, Kishan) = 1+1+1+\frac{(4-2)}{(4-1)} = 3.67$

$d(Bharat, Kishan) = 0.67+1+1+\frac{(4-3)}{4-1} = 3$

Most Similar → Balram and Bharat

Most dissimilar → Balram and Kishan

→ Cosine Similarity :- $Cos(d_1, d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \|d_2\|}$

Ex:- $d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$

$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$

$d_1 \cdot d_2 = 5\times3 + 0\times0 + 3\times2 + 0\times0 + 2\times1 + 0\times1 + 0\times0 + 2\times1 + 0\times0 + 0\times1$

$\|d_1\| = \sqrt{5^2+0+3^2+0+2^2+0+0+2^2+0+0} = 6.481$

$\|d_2\| = \sqrt{3^2+0+2^2+0+1^2+1^2+0+1^2+0+1^2} = 4.12$

$Cos(d_1, d_2) = 0.94$

Classification

⇒ GINI INDEX

At any node(t) $GINI(t) = 1 - \sum_j [p(j/t)]^2$

$p(j/t)$ → relative frequency of class j at node t.

→ Maximum $(1-1/n_c)$ when records are equall distributes among all classes implying least interesting information.

→ Minimum (0,0) when all records belong to one class implying most interesting information.

Ex:-

C_1	0
C_2	6

$GINI = 1 - \sum_j [p(j/t)]^2$

$P(C1) = 0/6 = 0,\ P(C_2) = 6/6 = 1$

$GINI = 1 - (P(C1))^2 + P(C_2)^2$

$= 1 - 0 - 1 = 0$

C_1	1
C_2	5

$P(1) = \frac{1}{6}, P(2) = 5/6$

$GINI = 1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 = 0.278$

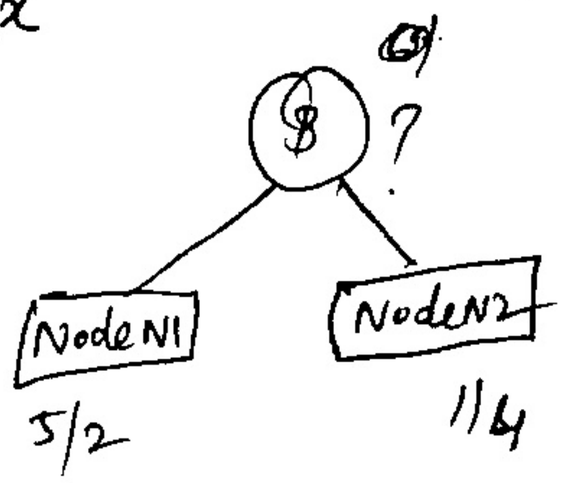
Splitting Based on GINI

$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} (GINI(i))$ → when a node P is split into to k partitions (children)

→ n_i → number of records at child i

→ n = number of records at node P

Computing Gini index



	Parent
C_1	6
C_2	6
GINI = 0.500	

$GINI(N1) = 1 - \left(\frac{5}{7}\right)^2 - \left(\frac{2}{7}\right)^2 = 0.408$

$GINI(N2) = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2 = 0.32$

$GINI(children) = \frac{7}{12} \times 0.408 + \frac{5}{12} \times 0.32 = 0.37$

	N1	N2
C1	5	1
C2	2	4
GINI = 0.37		

⇒ Example :- Which attribute should be choosen as first splitting attribute (b/w Q_1 & Q_2) according to GINI INDEX?

Instance	Q_1	Q_2	Q_3	target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−

Sol :- Contingency table for Q_1 & Q_2

Q_1

Class	Q_1=T	Q_1=F
+	3	1
−	1	4

Q_2

Class	Q_2=T	Q_2=F
+	2	2
−	3	2

$$GINI(D) = 1 - \sum_{i=j}^{m} [P(i/t)]^2$$

Overall GINI Before splitting = $1 - \left(\frac{4}{9}\right)^2 - \left(\frac{5}{9}\right)^2 = 1 - 0.1197 = 0.495$

$$GINI(Split) = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

For Q_1 GINI of Q_1 = $GINI(Q_1) = \frac{4}{9}\left[1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right] + \frac{5}{9}\left[1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2\right]$

$GINI(Q_1) = 0.3441$

$\Delta Gini(Q_1) = 0.495 - 0.3441 = 0.151$

For Q_2 :- $Gini(Q_2) = \frac{5}{9} Gini(i_1) + \frac{4}{9} Gini(i_2)$

$= \frac{5}{9}\left(1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2\right] + \frac{4}{9}\left[1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right]$

$= 0.4884$

$\Delta Gini(Q_2) = 0.495 - 0.488 = 0.007$

The attribute that maximises the reduction in impurity (or has minimum Gini index) is selected as splitting attribute.

Hence, Q_1 will be selected as the splitting attribute.

⇒ Classification Error [splitting Criteria]

Classification Error at node t :- i = class

$$Error(t) = 1 - \max[P(i|t)]$$

Error before splitting :-

Error orig :- $1 - \max\left(\frac{4}{9}, \frac{5}{9}\right) = 1 - \max(0.444, 0.555)$

$= 1 - 0.555 = 0.444.$

Error after splitting on q1

$$Error_{q1} = 1 - \frac{4}{9}\left[1 - \max\left[\frac{3}{4}, \frac{1}{4}\right]\right] - \frac{5}{9}\left[1 - \max\left(\frac{1}{5}, \frac{4}{5}\right)\right]$$

$$= 1 - \frac{4}{9}\left[1 - \max(0.75, 0.25)\right] - \frac{5}{9}\left[1 - \max(0.2, 0.8)\right].$$

$$= 1 - 0.444(1 - 0.75) - 0.555(1 - 0.8)$$

$$= 1 - 0.111 - 0.111$$

$$= 0.778$$

Error after splitting on q2

$$Error_{q2} = 1 - \frac{5}{9}\left[1 - \max\left(\frac{3}{5}, \frac{3}{5}\right)\right] - \frac{4}{9}\left[1 - \max\left(\frac{2}{4}, \frac{2}{4}\right)\right]$$

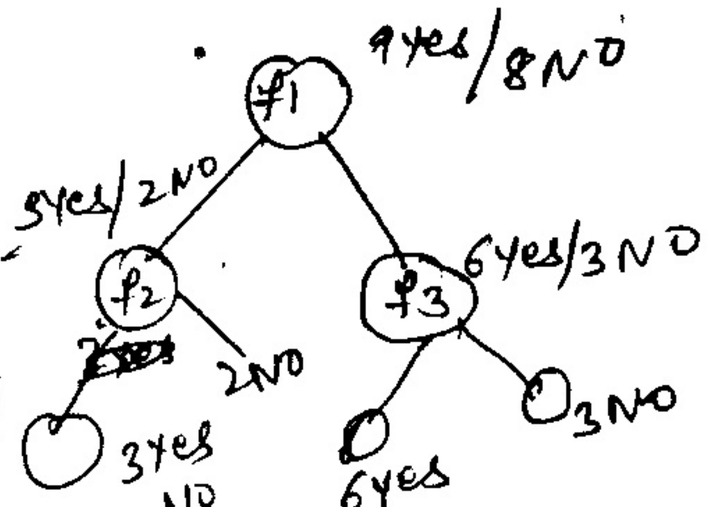
$$= 1 - 0.555\left[1 - \max(0.4, 0.6)\right] - 0.444\left[1 - \max(0.5, 0.5)\right]$$

$$= 1 - 0.222 - 0.222 = 0.556$$

The attribute that minimises classification error will be selected as splitting attribute, Hence q2 will be selected as splitting attribute.

⇒ ENTROPY :-

→ To select right attribute/feature for splitting, we need to calculate Entropy. How can we select the right node, If we select correctly we can reach leaf ~~node~~ node quickly



Suppose features f1, f2, f3

O/P - Yes
NO

⇒ Entropy helps to measure the purity of split.

→ Pure split whenever we get, that becomes leaf node.

$$\text{Entropy } H(S) = -P(+)\log_2(P) - P(-)\log_2 P(-)$$

$P+/P_-$ = % of +ve class / % of -ve class

S - Subset of training Example

$$\text{Entropy}(t) = -\sum_j P(j/t)\log_2 P(j/t)$$

⇒ When we have complete impure split like 3Yes and 3No, the entropy will be 1 (Worst). Entropy will range from 0 to 1, lesser the entropy value, better the split

⇒ Ex:-

C1	0
C2	6

$$P(C1) = -\frac{0}{6}\log\frac{0}{6} = 0$$

$$P(C2) = \frac{6}{6} = 1$$

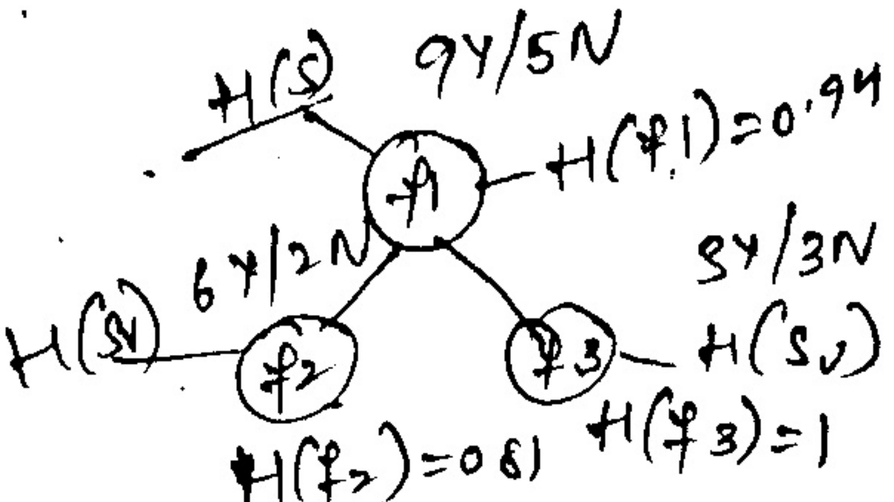
$$\text{Entropy } H(S) = 0\log 0 - 1\log 1 = 0$$

⇒ Information Gain

$$\text{Gain}(S,A) = H(S) - \sum \frac{|Sv|}{|S|} H(Sv)$$

$$\text{Gain}(S, f_1) = H(S) - \frac{8}{14}H(f_2) - \frac{6}{14}H(f_3)$$

$$= 0.94 - \frac{8}{14} \times 0.81 - \frac{6}{14} \times 1 = 0.049$$



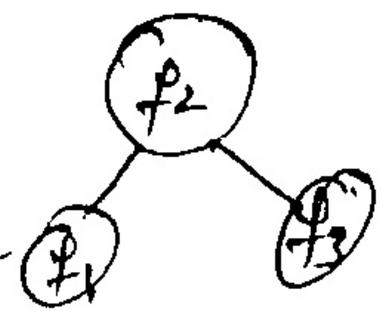
Higher the Information gain, better split

→ Select the attribute with highest Information Gain

→ Let P_i be the probability that an arbitary tuple in D belongs to class C_i, estimated by $|C_i, D|/|D|$

→ Expected information (entropy) needed to classify a tuple in D:- $\text{Info}(D) = -\sum_{i=1}^{m} P_i \log_2(P_i)$

→ Information needed (after using A to split D into v partitions) to classify D:- $\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|Dj|}{|D|} \times \text{Info}(Dj)$



Information Gained $\overline{\text{Gain}(A) - \text{Info}(D) - \text{Info}_A(D)}$

Sol Example:- Using Information Gain As impurity measure

Contigency table q1 →

Class	q1=T	q1=F
+	3	1
−	1	4

Class	q2=T	q2=F
+	2	2
−	3	2

⇒ Gives All information needed to classify a tuple D is

$$Info_A(D) = \sum_{j=1}^{k} \frac{|Dj|}{|D|} \times Inf(Dj)$$

$$Info(D) = -\sum_{i=1}^{m} Pi \log_2 Pi)$$

$$Gain(A) = Info(D) - Info_A(D)$$

⇒ Overall information needed to classify a tuple

$$Info(D) = -\frac{4}{9}\log_2 4/9 - \frac{5}{9}\log_2 5/2 = 01989$$

⇒ Expected Information needed to classify a tuple D acc to q1

$$Info_{q_1}(D) = \frac{4}{9}\left[-\frac{3}{4}\log_2 3/4 - 1/4 \log_2 1/4\right] + \frac{5}{9}\left[\frac{-1}{5}\log_2 1/5 - 4/5 \log_2 4/5\right]$$

$$= 0.760$$

⇒ Information Gain (q1) = Info(D) − $Info_{q_1}(D)$ = 0.989 − 0.760 = 0.229

⇒ Expected Information needed to classify tuple D acc to q2

$$Info_{q_2}(D) = \frac{4}{9}\left[-\frac{2}{5} \cdot \frac{8}{8}\log_2 2/5 - \frac{3}{5}\log_2 3/5\right] + \frac{5}{9}\left[-\frac{2}{4}\log_2 2/2 - 2/4 \log_2 2/4\right]$$

$$= 0.985$$

Information Gain(q2) = Info(D) − $Info_{q_1}(D)$ = 0.989 − 0.985 = 0.004

⇒ Gain(q2) is more for q2, it will be selected as first splitting attribute.

⇒ Gain Ratio

→ Information Gain measure is biased towards attributes with large number of values. C4.5 (GD3 successor) uses gain ratio to overcome the problem (normalization of Information gain)

$$\text{Gain Ratio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfo}(A)}$$

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2 \frac{|D_j|}{|D|}$$

Sol :- $\text{SplitInfo}_{q_1}(D) = -\frac{4}{9}\log_2 4/9 - \frac{5}{9}\log_2 5/9 = 0.1989$

$$\text{SplitInfo}_{q_2}(D) = -\frac{5}{9}\log_2 5/9 - \frac{4}{9}\log_2 4/9$$

$$= -0.555 * (-0847) - 0.444\,(-1.169) = 0.1989$$

$$\text{Gain Ratio}(q_1) = \frac{\text{Gain}(q_1)}{\text{SplitInfo}_{q_1}(D)} = \frac{0.229}{0.1989} = 0.231$$

$$\text{Gain Ratio}(q_2) = \frac{\text{Gain}(q_2)}{\text{SplitInfo}_{q_2}(D)} = \frac{0.004}{0.1989} = 0.004$$

the attribute with maximum gain ratio is selected as splitting attribute, Hence q_1 will be selected as first splitting attribute

Example !-

age	income	student	credit-rating	buys-computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31-40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31-40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31-40	medium	no	excellent	yes
31-40	high	yes	fair	yes
>40	medium	no	excellent	no

→ Class P: - buys-computer 'yes'
→ Class N: - buys-computer = "NO"

age	Pi	ni	I(Pi,ni)
<=30	2	3	0.971
31-40	4	0	0
>40	3	2	0.971

$\Rightarrow$ $Info(D) = I(9,5) = \frac{9}{14}\log_2 9/14 - \frac{5}{14}\log_2 5/14$

$I(9,5) = 0.940$

$\Rightarrow Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2)$

$= 0.694$

$\frac{5}{14} I(2,3)$ → means "age<=30" → has 5 out of 14 samples with 2 yes'es and 3 NO's Hence

$Gain(age) = Info(D) - Info_{age}(D)$

$= 0.940 - 0.694$

$= 0.246$

Similarly Gain(income) = 0.029
Gain(student) = 0.151
Gain(credit_rating) = 0.048

Gain Ratio :-

$SplitInfo_{income}(D) = \frac{-4}{14}\log\frac{4}{14} - \frac{6}{14}\log(6/14) - \frac{4}{14}\log_2\left(\frac{4}{14}\right)$

$= 1.557$

$gain\_ratio(income) = \frac{0.029}{1.557} = 0.019$

The attribute with maximum gain ratio is selected as the splitting attribute.

⇒ Rule based Classification

⇒ Given a tuple 'X', from a class labeled data set D,
let n_{covers} be the number of tuples covered by R
$n_{correct}$ be the number of tuples correctly classified by R
|D| → be the number of tuples in D, then

$$Coverage(R) = \frac{n_{covers}}{|D|}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}$$

⇒ let (Pos) → be the Positive and (neg) be the negative tuples covered by R
let (Pos') → be the Positive and (neg') be the negative tuples covered by R', FOIL asses the information gained by extending condition as

$$FOIL\_Gain = Pos' \times \left(\log_2 \frac{Pos'}{Pos'+neg'} - \log_2 \frac{Pos}{Pos+neg} \right)$$

$$FOIL\_PRUNE(R) = \frac{pos-neg}{pos+neg}$$ [if Foil-Prune value is higher, then we Prune R]

⇒ Likelihood_Ratio = $2 * \sum_{i=1}^{m} f_i \log_2 \left(\frac{f_i}{e_i}\right)$ → Higher → better

→ m → number of classes
→ f_i → observed freq of each class among tuples
→ e_i → expected freq if the rule made random predictions
→ used by CN2

Example :- Derive the Possible rules for classification of bank customer into defaulter or not defaulter classes based on the following data set :

Tid	Home-Owner	MaritalStatus	Defaulter
1	Yes	Single	NO
2	NO	Married	NO
3	NO	Single	NO
4	Yes	Married	NO
5	NO	Divorced	Yes
6	NO	Married	NO
7	Yes	Single	Yes
8	NO	Married	NO
9	NO	Single	Yes
10			

few direct possible Rules by looking at data set

r_1 = (Home-owner = Yes) $\Rightarrow$ (Defaulter = NO)

r_2 :- (Marital-status = married) $\Rightarrow$ (Defaulter=NO)

r_3 : (Home-Owner = NO) & (Marital-status = Married) $\Rightarrow$ (Defaulter-NO)

r : A $\rightarrow$ y $\rightarrow$ A is rule antecedent
y is rule consequent

$$\text{Coverage}(r_1) = \frac{|A|}{|D|} = \frac{n_{covers}}{|D|} = \frac{3}{10} = 0.3 = 30\%$$

$$\text{Accuracy}(r_1) = \frac{n_{correct}}{|n_{covers}|} = \frac{3}{3} = 1 = 100\%$$

$\therefore$ Coverage $(r_2) = \frac{4}{10} = 0.4 = 40\%$; Accuracy $(r_2) = \frac{4}{4} = 1 = 100\%$

Coverage $(r_3) = \frac{2}{10} = 0.2 = 20\%$; Accuracy $(r_3) = \frac{2}{2} = 1 = 100\%$.

$\rightarrow$ r_2 $\rightarrow$ has high accuracy with high coverage $\rightarrow$ stronger

⇒ Example :- Consider a training dataset that contains 60 +ve samples & 100 -ve samples. Suppose following two candidate rules are mind-

r_1 :→ Covers 50 +ve samples & 5 -ve samples

r_2 :→ Covers 2 +ve samples & 2 -ve samples. , which rull is better. determine Information gain.

Sol :- Coverage $(r_1) = \frac{55}{160} = 0.343 = 34.3\%$, Accuracy$(r_1) = \frac{50}{55} = 0.909$
$= 90.9\%$

Coverage $(r_2) = \frac{2}{160} = 0.0125 = 1.25\%$ / Accuracy $(r_2) = \frac{2}{2} = 100\%$

r_1 is better despite its lower accuracy, the high accuracy for r_2 is potentially superious because of low coverage of rule-

$$\text{FOIL\_Info gain } (r_1) = P_1 \times \left[\log_2 \frac{P_1}{P_1+n_1} - \log_2 \frac{P_0}{P_0+n_0} \right]$$

$$\text{Foil-gain}(r_1) = 50 \times \left[\log_2 \frac{50}{50+55} - \log_2 \frac{60}{60+100} \right]$$

$$= 50 \left[-0.137 + 1.415 \right] = 63.9$$

$$\text{Foil-gain}(r_2) = 2 \times \left[\log_2 \frac{2}{2} - \log_2 \frac{60}{60+160} \right] = 2.83$$

As Foils gain for r_1 is better than r_2, r_1 is better rule.

Model Evaluation and Selection :-

⇒ Confusion Matrix

Predicted class	C_1	$-C_1$
Actual class		
C_1	True Positives (TP)	False Negatives (FN)
$-C_1$	False Positives (FP)	True Negatives (TN)

⇒ Accuracy ⇒ $\frac{TP+TN}{P+N}$

⇒ Error rate ⇒ 1 − accuracy

OR :- $\frac{(FP+FN)}{P+N}$

⇒ Sensitivity = TP/P [True Positive recognition rate]

⇒ Specificity = TN/N [True Negative recognition rate]

TP ⇒ no. of True Positive

TN ⇒ No. of True Negative

FP → false Positive

P → Total Positive

N → Total Negative

A/P	C	−C	
C	TP	FN	P
−C	FP	TN	N
	P'	N'	P+N

⇒ Precision :- $\frac{TP}{TP+FP}$ (exactness)

⇒ Recall = $\frac{TP}{TP+FN} = \frac{TP}{P}$ (Completness)

Perfect Score = 1.0

⇒ F measure (F1 or f score) = $F = \frac{2 \times Precision \times recall}{Precision + recall}$

(harmonic mean of Precision & Recall)

⇒ $F_\beta = \frac{(1+\beta^2) \times Precision \times recall}{\beta^2 \times Precision + recall}$ → weight measure

$\beta^2 = (1-\alpha)/\alpha$

Example :-

Predicted class → / Actual class ↓	Y	N	
Y	900	100	1000
N	200	800	1000
	1100	900	2000

TP = 900, FP = 200, FN = 100, TN = 800

$$\text{Precision } P = \frac{TP}{TP+FP} = \frac{900}{900+200} = 0.818$$

$$\text{Recall } R = \frac{TP}{TP+FN} = \frac{900}{900+100} = 0.9$$

$$F_{Score} = \frac{2 \times TP}{2 \times TP+FP+FN} = \frac{2 \times R \times P}{R+P} = \frac{2 \times 900}{2 \times 900+200+100} = 0.85$$

$$\text{Sensitivity (True +ve Rate)} = \frac{TP}{TP+fN} = \frac{900}{900+100} = 0.9$$

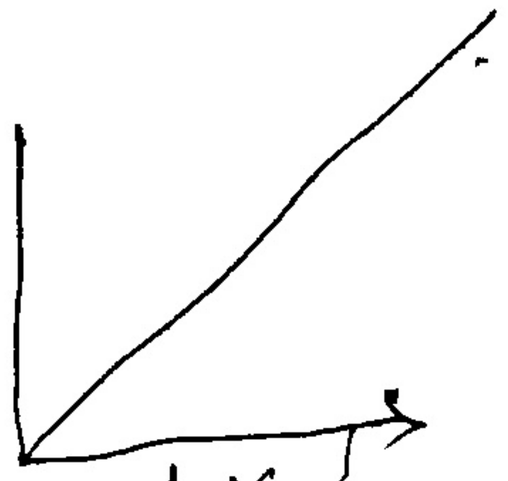
$$\text{Specificity (True -ve Rate)} = \frac{TN}{TN+FP} = \frac{800}{800+200} = 0.8$$

$$\text{False +ve Rate} = \frac{FP}{TN+FP} = \frac{200}{800+200} = 0.2$$

$$\text{False -ve Rate} = \frac{FN}{TP+FN} = \frac{100}{900+100} = 0.1$$

Simple Linear Regression

$Y = w_0 + w_1 X$ → w_0 – Y-intercept, w_1 → slope



$$w_0 = \frac{(\Sigma Y)(\Sigma X^2) - (\Sigma X)(\Sigma XY)}{n(\Sigma X^2) - (\Sigma X)^2}$$

$$w_1 = \frac{n(\Sigma XY) - (\Sigma X)(\Sigma Y)}{n(\Sigma X^2) - (\Sigma X)^2}$$

n → number of tuples.

X	Y	XY	X^2	P	Error
1	3	3	1	2.8	0.2
2	4	8	4	4.1	0.1
3	5	15	9	5.4	0.4
4	7	28	16	6.7	0.3
$\Sigma 10$	$\Sigma 19$	$\Sigma 54$	$\Sigma 30$		

Predicte

$$w_0 = \frac{(19)(30) - (10)(54)}{(4)(30) - 100} = 1.5$$

$$w_1 = \frac{(4)(54) - (10)(19)}{(4)(30) - 100} = 1.3$$

$\Rightarrow$ $Y = 1.3X + 1.5$

For any given x → find y

Association Rule Analysis

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Beer
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Itemset :- { Milk, Bread, Diaper }

k-itemset
→ Itemset that contains k items

Support count(σ) :- frequency of an occurrence of Itemset

Eg:- σ({ Milk, Bread, Diaper }) = 2

Support :- fraction of transactions that contain an itemset

Eg:- S(Milk, Bread, Diaper) = 2/5

Frequent Itemset :

- An itemset whose support is greater than or equal to minsup threshold.

Association Rule $X \rightarrow Y$

where X & Y are Itemsets

Ex:- {Milk, Diaper} → Butter.

- **Support(s)** :- fraction of transactions that contain both X & Y

$$S = \frac{\sigma(\text{Milk, Diaper, Butter})}{T} = \frac{2}{5} = 0.4$$

- **Confidence (c)** :- Measure how often items in Y appear in transactions that contain X

$$C = \frac{\sigma(\text{Milk, Diaper, Butter})}{\sigma(\text{Milk, Diaper})} \Rightarrow \frac{2}{3} = 0.67$$

→ Support ≥ minsup threshold
confidence ≥ minconf threshold

⇒ Apriori Algorithm

Apriori employs an iterative approach known as level-wise search where K-itemsets are used to explore (K+1)-itemsets First set of frequent 1-itemset is found by scanning DB to accumulate the count of 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, the set of frequent 2-itemsets, which is used to find L_3 and so, on, until no more frequent K-item sets can be found! -

Example :- Min support = 50%
Threshold confidence 7%.

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

↓

L_1

Itemset	Support
1	2/4 = 50%
2	3/4 = 75%
3	3/4 → 75%
(4)	1/4 → 25% → This can be dropped
5	3/4 → 75%

L_1 Itemset ⇒ → 1, 2, 3, 5

(II) L_2

Itemset	Support
{1,2}	1/4 → 25% → X
{1,3}	2/4 → 50%
{1,5}	1/4 → 25% → X
{2,3}	2/4 → 50%
{2,5}	3/4 → 75%
{3,5}	2/4 → 50%

(III) L_3

Itemset	Support
{1,3,5}	1/4 = 25% → X
{2,3,5}	2/4 = 50%
{1,2,3}	1/4 = 25% → X

→ {2, 3, 5}

Example :- Min Sup = 2

TID	Items List
1	I_1, I_2, I_5
2	I_2, I_4
3	I_1, I_2, I_4
4	I_1, I_3
5	I_2, I_3
6	I_1, I_3
7	I_1, I_2, I_3, I_5
8	I_1, I_2, I_3
9	

Step1 :- Get frequency support count

$I_1 - 6$
$I_2 - 7$
$I_3 - 6$
$I_4 - 2$
$I_5 - 2$

⇐ (C1)
. Condidate set of C1
min support count = 2
Nothing will be removed

L1

Step2 :-

Itemset	Suppcount
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_4\}$	1 x
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2
$\{I_3, I_4\}$	0 x
$\{I_3, I_5\}$	1 x
$\{I_4, I_5\}$	0 x

→ Compare Candidate support count with min support count

⇒

Itemset	supcount
$\{I_1, I_2\}$	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

L2

Step3 :-

Itemset	Suppcount
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2
$\{I_1, I_2, I_4\}$	1 x
$\{I_1, I_3, I_5\}$	1 x
$\{I_2, I_3, I_4\}$	0 x
$\{I_2, I_3, I_5\}$	1 x
$\{I_2, I_4, I_5\}$	0 x

⇒ L_3

Itemset	Suppl.
$\{I_1, I_2 I_3\}$	2
$\{I_1, I_2, I_5\}$	2