__/__/____

DATA MINING

Ch-4 Classification

→ Classification is the task of assigning objects to one of several predefined categories

→ It is the task of learning a target function f (Classification model) that maps each attribute set x to predefined class label y.

• Uses of classification model:

① Descriptive Modelling: It serves as an explanatory tool to distinguish b/w objects of different classes. For eg. what features define a vertebrate as a mammal, reptile etc.

② Predictive Modeling: It can be used to predict the class label of unknown records

• General Approach to Solving a Classification Problem:

→ A learning algorithm is used to generate a model

→ The model generated by a learning ~~algorithm~~ algo should fit the input data well & correctly predict the class labels of records it has never seen before

→ First, a training set consisting of records whose class labels are known, must be provided. This is used to build a classification model which is then applied to test set, which consists of records with unknown class label.

→ The counts of test records correctly & incorrectly predicted are tabulated in a confusion matrix

		Predicted Class	
		class = 1	class = 0
Actual Class	class = 1	f_{11}	f_{10}
	class = 0	f_{01}	f_{00}

Eg. f_{11} shows from class 1 predicted as class 1
f_{10} shows from class 1 " " class 0

__/__/__

→ Performance ~~measure~~ metric is used to compare models using accuracy and error rate

$$\text{Accuracy} = \frac{\text{No. of correct predictions}}{\text{Total no. of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

$$\text{Error Rate} = \frac{\text{No. of wrong predictions}}{\text{Total no. of predictions}} = \frac{f_{01} + f_{10}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

- **Decision Tree Induction**

→ We can solve a classification problem by asking a series of carefully crafted questions. These & their possible answers can be ~~org~~ organised in the form of a decision tree.

→ The tree has 3 types of nodes:

(1) A root node that has no incoming edges & zero or more outgoing edges

(2) Internal nodes each of which has exactly one incoming edge and two or more outgoing edges

(3) Leaf or terminal nodes each of which has exactly one incoming edge & no outgoing edges.

- **Hunt's Algorithm**

→ Let D_t be the set of training records associated with node t and $y = \{y_1, y_2, \ldots y_c\}$ be the class labels.

(1) If all records in D_t belongs to the same class y_t, then t is a leaf node labelled as y_t

(2) If it belongs to more than one class, an attribute test condition is selected to partition the records into smaller subsets.

A child node is created for each outcome of the test condition and the records in D_t are distributed to the children based on the outcomes. The algo is then recursively applied.

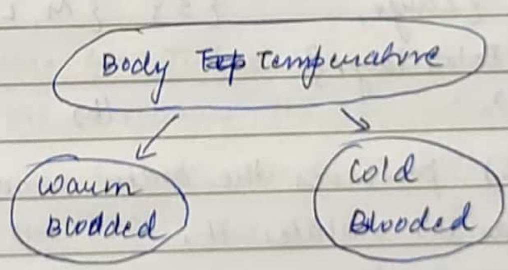
* In step 2, if all records associated with D_t have identical attribute values (except for class label), then it is not possible to split these records any further. In this case, the node is declared a leaf node with same class label as majority of training records.

- Design Issues with decision tree induction

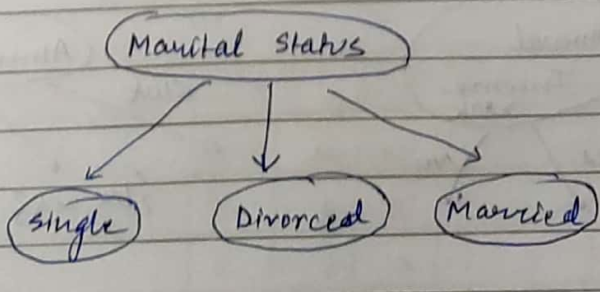
(1) How should the training record be split: There must be a test condition as well as an objective measure for evaluating the goodness of each test condition

(2) How to stop the splitting procedure: We can continue expanding the node until either all records belong to the same class or all records have identical attribute values

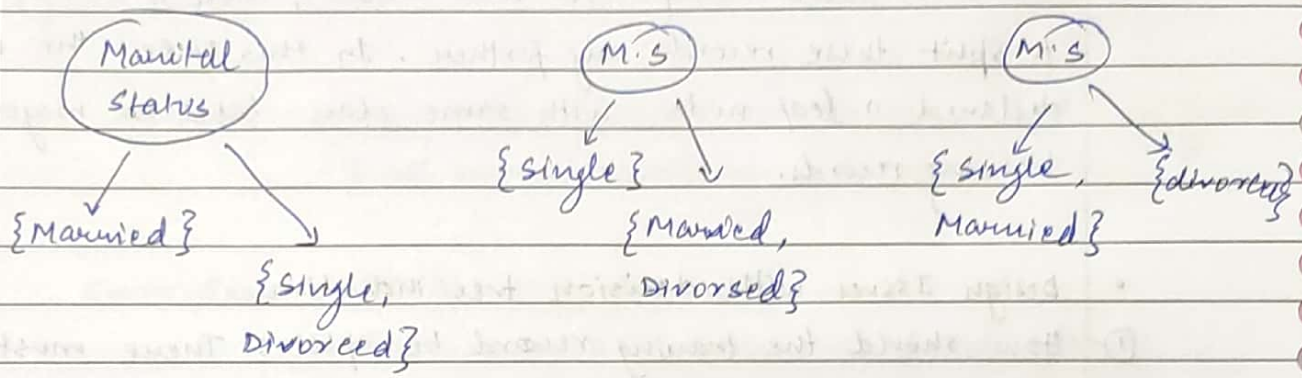
- Binary Attribute: The test condition for a binary attribute generates two potential outcomes



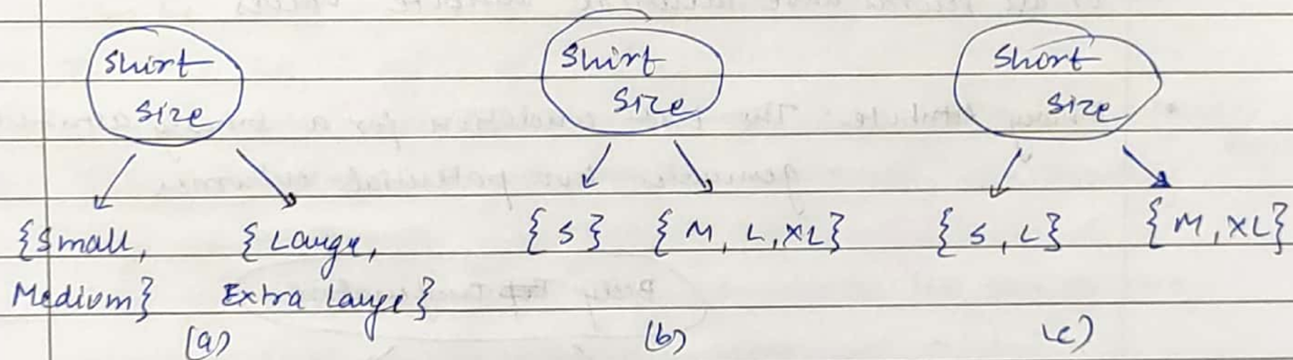
- Nominal Attributes: For a <u>multiway split</u>, the no. of outcomes depends on the distinct values of corresponding attribute



For Binary split, we can group the attribute values

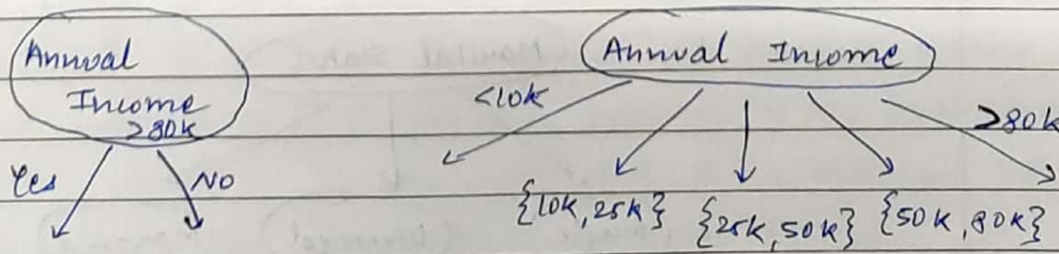


- Ordinal Attributes: They can be splitted in binary or multiway. They can be grouped as long as the grouping does not violate the order property of attribute value.



→ (a) & (b) preserves the order among the attribute values, whereas (c) violates it.

- Continuous Attributes: The test condition can be expressed as a comparison set $(A<v)$ or $(A \geq v)$ with binary outcomes, or a range query with outcomes of the form $v_i \leq A < v_{i+1}$ ...



- **Measures of selecting the Best split**

→ A node with class distribution (0, 1) has zero impurity, whereas a node with uniform class distribution (0.5, 0.5) has the highest impurity.

Node N_1	Count
Class = 0	0
Class = 1	6

$Gini = 1 - (0/6)^2 - (6/6)^2 = 0$

$Entropy = -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$

$Error = 1 - \max[0/6, 6/6] = 0$

Node N_2	Count
Class = 0	1
Class = 1	5

$Gini = 1 - (1/6)^2 - (5/6)^2 = 0.278$

$Entropy = -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.65$

$Error = 1 - \max[1/6, 5/6] = 0.167$

Node N_3	Count
Class = 0	3
Class = 1	3

$Gini = 1 - (3/6)^2 - (3/6)^2 = 0.5$

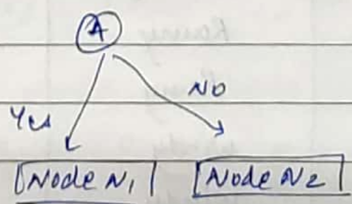
$Entroply = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$

$Error = 1 - \max[3/6, 3/6] = 0.5$

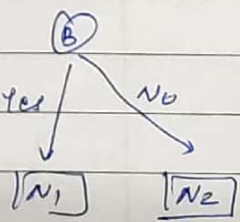
- **Splitting Attributes using Gini Index**

① Binary Attribute: Before splitting, the Gini index is 0.5 because there are an equal no. of records in both classes. If att. A is choosen to split the data, the GI for N_1 is 0.4898 & for N_2 it is 0.480. The weighted GI both nodes is $\frac{7}{12} \times 0.4898 + \frac{5}{12} \times 0.480 = 0.486$. Similarly for B it is 0.375. Since B has smaller GI it is preferred over attribute A.

	Parent
C_0	6
C_1	6
Gini = 0.500	



	N_1	N_2
C_0	4	2
C_1	3	3
Gini = 0.486		



	N_1	N_2
C_0	1	5
C_1	4	2
Gini = 0.375		

__/__/____

(2) Nominal Attribute: For binary splitting it is similar as before. For multiway ~~split~~ split, the Gini Index is computed for every attribute value. Since gini({Family}) = 0.375, Gini ({Sports}) = 0 & gini({Luxury}) = 0.219, the overall Gini Index for the multiway split is -

$$\frac{4}{20} \times 0.375 + \frac{8}{20} \times 0 + \frac{8}{20} \times 0.219 = \underline{0.163}$$

(3) Continuous Attribute: A brute force method to find v is to consider every value of the attribute in N records as a candidate split position. We compute the GI for each candidate & choose the one that gives the lowest value.
This approach is computationally expensive because it required O(N) operations. Since there are N candidates the overall complexity is O(N2). To reduce the complexity the training records are sorted on the basis of annual income, now it req. O(N log N) time

★ **Decision Tree using Gini Index**

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

__/__/____

- Gini Index of overall training samples

$$1-\left[\left(\frac{\text{Cinema}}{\text{Total}}\right)^2+\left(\frac{\text{Tennis}}{\text{Total}}\right)^2+\left(\frac{\text{Stay In}}{\text{Total}}\right)^2+\left(\frac{\text{Shopping}}{\text{Total}}\right)^2\right]$$

$$1-\left[\left(\frac{6}{10}\right)^2+\left(\frac{2}{10}\right)^2+\left(\frac{1}{10}\right)^2+\left(\frac{1}{10}\right)^2\right]$$

$$1-\left[\frac{36}{100}+\frac{4}{100}+\frac{1}{100}+\frac{1}{100}\right] = \underline{\underline{0.58}}$$

- Gini Index of Money Attribute

Money = Poor

$$1-\left[\left(\frac{3}{3}\right)^2+0+0+0\right] = 1-1 = \underline{\underline{0}}$$

Money = Rich

$$1-\left[\left(\frac{3}{7}\right)^2+\left(\frac{2}{7}\right)^2+\left(\frac{1}{7}\right)^2+\left(\frac{1}{7}\right)^2\right]^2 = \underline{\underline{0.694}}$$

weighted avg. of Money $= \frac{3}{10}\times 0+\frac{7}{10}\times 0.694 = \boxed{0.486}$

- Gini Index of Parents Attribute

Parents = Yes

$$1-\left[\left(\frac{5}{5}\right)^2+0+0+0\right] = 1-1 = \underline{\underline{0}}$$

Parents = No

$$1-\left[\left(\frac{2}{5}\right)^2+\left(\frac{1}{5}\right)^2+\left(\frac{1}{5}\right)^2+\left(\frac{1}{5}\right)^2\right] = \underline{\underline{0.72}}$$

weighted avg. of Parents $= 0\times\frac{5}{10}+\frac{5}{10}\times 0.72 = \boxed{0.36}$

_/_/_

- Gini Index of weather Attribute

Weather = Sunny

$$1-\left[\left(\frac{2}{3}\right)^2+\left(\frac{1}{3}\right)^2+0+0\right] = 0.444$$

Weather = Rainy

$$1-\left[\left(\frac{2}{3}\right)^2+\left(\frac{1}{3}\right)^2+0+0\right] = 0.444$$

Weather = Windy

$$1-\left[\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2+0+0\right] = 0.375$$

Weighted Average for Weather $= \frac{3}{10}\times 0.444 + \frac{3}{10}\times 0.444 + \frac{4}{10}\times 0.375$

$$= \boxed{0.416}$$

→ Since the gini Index for Attribute Parents is the lowest, so we select it as the split attribute

→ For Parents = NO, we have to further find the best att. to split

- Gini for Parents = NO & Weather = Sunny

$$1-\left[\left(\frac{2}{2}\right)^2\right] = 0$$

Parents = NO | Weather = Rainy

$$1-\left[\left(\frac{1}{1}\right)^2\right] = 0$$

Parents = NO | Weather = Windy

$$1-\left[\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right] = 0.5$$

weighted Avg. for Parents = NO | weather $= \frac{2}{5}\times 0 + \frac{1}{5}\times 0 + \frac{2}{5}\times 0.5$

$$= 0.2$$

- Gini Index for Parents = NO | Money

  Parents = NO | Money = Rich

  $$1 - \left[\left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right] = 0.625$$

  Parents = NO | Money = Poor

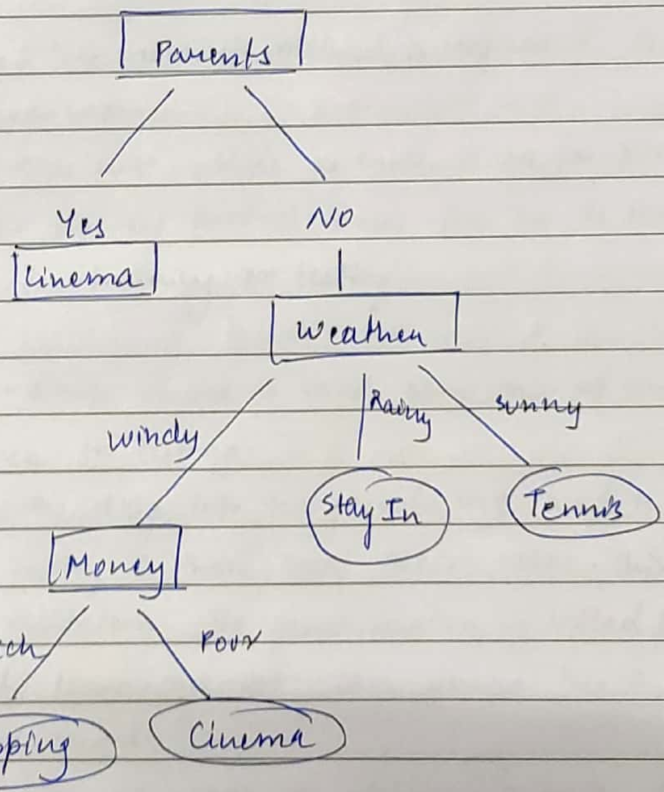
  $$1 - \left[\left(\frac{1}{1}\right)^2\right] = 0$$

  Weighted Avg. for Parents = NO | Money $= \frac{4}{5} \times 0.625 + \frac{1}{5} \times 0$

  $$= \underline{\underline{0.5}}$$

→ Since GI for weather is lower we select it as split attribute

→ we have to split for Parent = NO | weather = windy using Money attribute.



__/__/__

- Evaluating the performance of a classifier:

① Holdout Method: Original data is divided into two disjoint sets called traing & test set. A classification model is then induced from the training set & evaluated on test set.

→ Limitation:

- Fewer labelled eg's are available for training because some are withheld for testing.
- It may be highly dependent on the composition of traing & test set. Smaller the training set, larger the variance. or if the training set is too large, then the estimated accuracy computed from the smaller test set is less reliable.
- The training & test set are no longer independent of each other. A class that is overrepresented in one subset will be over represented in the other.

② Random Subsampling: The holdout method is repeated several times to improve performance.

→ Limitation: Still does not utilize as much data as possible for training. It also has no control over the no. of times each record is used for training or testing.

③ Cross Validation: Each record is used same no. of times for training and exactly once for testing.

- If we partition the data into two equal size and the use both for training & testing & then swap their roles. This is called two fold cross validation. Its generalisation is called k-fold cross validation.
- It has a special leave-one-out case where the test set has only one record.
- This approach has the advantage of utilizing as much data as possible.
- Limitation: computationally expensive to repeat N times
  Variance of estimated performance metric is high