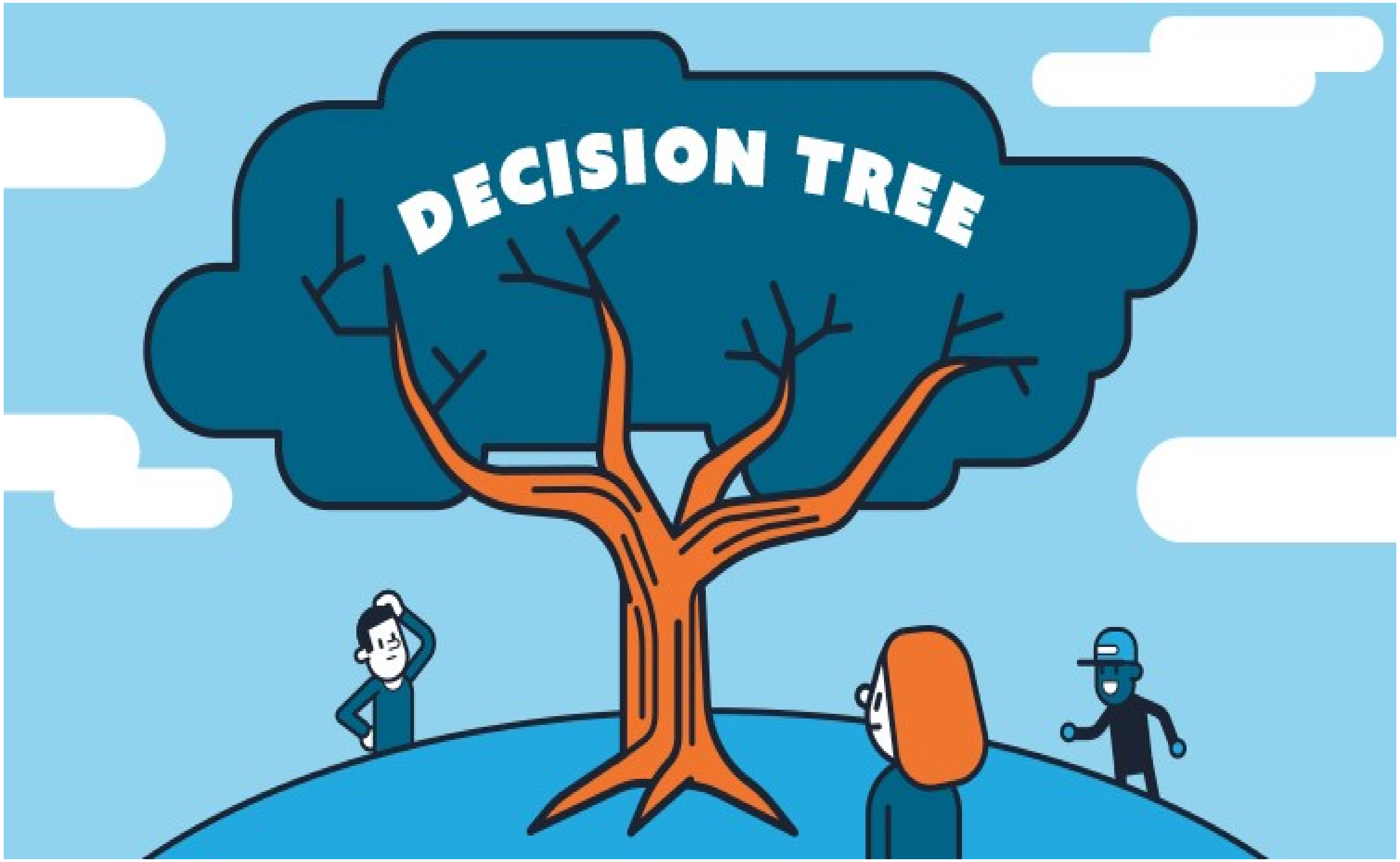
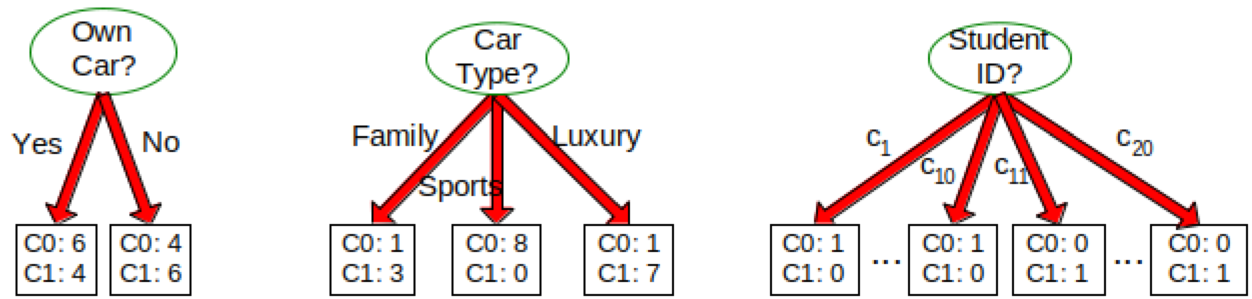
Decision Tree Classification



Gain Ratio

- The information gain measure is biased toward tests with many outcomes.
- It prefers to select attributes having a large number of values.



- Comparing the first test condition, *OwnCar*, with the second,*Car Type:*
- *Car Type* provide a better way of splitting the data since it produces purer descendent nodes.
- However, if we compare both conditions with Customer ID, the latter appears to produce purer partitions.
- Yet Customer ID is not a predictive attribute because its value is unique for each record.

Gain Ratio

- Two strategies for overcoming this problem.
  - The first strategy is to restrict the test conditions to binary splits only.
  - Used in algorithm CART
- Another strategy is to modify the splitting criterion to take into account the number of outcomes produced by the attribute test condition.
- In the C4.5 decision tree algorithm, a splitting criterion known as gain ratio is used to determine
- Normalization to information gain using a “split information” value

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

Gain Ratio

- For each outcome, it considers the number of tuples having that outcome with respect to the total number of tuples in D.
- It differs from information gain, which measures the information with respect to classification that is acquired based on the same partitioning

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

- The attribute with the **maximum gain ratio** is selected as the splitting attribute.
- Note, however, that as the split information approaches 0, the ratio becomes unstable.
- A constraint is added to avoid this, whereby the information gain of the test selected must be large—at least as great as the average gain over all tests examined

Gain Ratio : Example

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Outlook	
Info	0.693
Gain: 0.940-0.693	0.247
Split info: info ([5,4,5])	1.577
Gain ratio: 0.247/1.577	0.156

Humidity	
Info	?
Gain: 0.940-0.788	?
Split info: info ([7,7])	?
Gain ratio: 0.152/1	?

Temperature	
Info	?
Gain: 0.940-0.911	?
Split info: info ([4,6,4])	?
Gain ratio: 0.029/1.362	?

Windy	
Info	?
Gain: 0.940-0.892	?
Split info: info ([8,6])	?
Gain ratio: 0.048/0.985	?

Gain Ratio : Example

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Outlook	
Info	0.693
Gain: 0.940-0.693	0.247
Split info: info ([5,4,5])	1.577
Gain ratio: 0.247/1.577	0.156

Class : P-9 N-5

Outlook:

Sunny : P - 2 N - 3
Overcast : P - 4 N - 0
rain : P - 3 N - 2

Gain Ratio : Example

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Outlook	
$Info_{Outlook}$	**0.693**
Gain: 0.940-0.693	0.247
Split info: info ([5,4,5])	1.577
Gain ratio: 0.247/1.577	0.156

Class : P-9 N-5

Outlook:

Sunny : P - 2 N - 3
Overcast : P - 4 N - 0
rain : P - 3 N - 2

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

$$= (5/14)[- 2/5 \log_2(2/5) - 3/5 \log_2(3/5)] + (4/14)[- 4/4 \log_2(4/4)] + (5/14)[- 3/5 \log_2(3/5) - 2/5 \log_2(2/5)]$$

= 0.693

Gain Ratio : Example

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Outlook	
Info	0.693
Gain: **0.940**-0.693	0.247
Split info: info ([5,4,5])	1.577
Gain ratio: 0.247/1.577	0.156

Class : P-9 N-5

Outlook:

Sunny : P - 2 N - 3
Overcast : P - 4 N - 0
rain : P - 3 N - 2

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

$$= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$

$$= \mathbf{0.940}$$

Gain Ratio : Example

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Outlook	
Info	0.693
Gain: 0.940-0.693	0.247
Split info: info ([5,4,5])	**1.577**
Gain ratio: 0.247/1.577	0.156

Class : P-9 N-5

Outlook:

Sunny : P - 2 N - 3
Overcast : P - 4 N - 0
rain : P - 3 N - 2

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

= -2* [(5/14)$\log_2$(5/14)] -(4/14)$\log_2$(4/14)

= **1.577**

Gain Ratio : Example

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Outlook	
Info	0.693
Gain: 0.940-0.693	0.247
Split info: info ([5,4,5])	1.577
Gain ratio: 0.247/1.577	**0.156**

Class : P-9 N-5

Outlook:

Sunny : P - 2 N - 3
Overcast : P - 4 N - 0
rain : P - 3 N - 2

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

= 0.247 / 1.577
= **0.156**

Gain Ratio : Example

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Outlook	
Info	0.693
Gain: 0.940-0.693	0.247
Split info: info ([5,4,5])	1.577
Gain ratio: 0.247/1.577	**0.156**

Humidity	
Info	0.788
Gain: 0.940-0.788	0.152
Split info: info ([7,7])	1
Gain ratio: 0.152/1	**0.152**

Temperature	
Info	0.911
Gain: 0.940-0.911	0.029
Split info: info ([4,6,4])	1.362
Gain ratio: 0.029/1.362	**0.021**

Windy	
Info	0.892
Gain: 0.940-0.892	0.048
Split info: info ([8,6])	0.985
Gain ratio: 0.048/0.985	**0.049**

Gini Index

- The coefficient ranges from 0 (or 0%) to 1 (or 100%), with 0 representing perfect equality and 1 representing perfect inequality.
- The Gini index is used in CART.
- Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified
- Impurity of D, a data partition or set of training tuples:

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

- p_i is the probability that a tuple in D belongs to class C_i and is estimated by $|C_{i,D}| / |D|$.
- The sum is computed over m classes

Gini Index

- The Gini index considers a binary split for each attribute.
- Let's first consider the case where A is a discrete-valued attribute having v distinct values, {a 1 , a 2 ,..., a v }, occurring in D.
- To determine the best binary split on A, we examine all the possible subsets that can be formed using known values of A.
- If A has v possible values, then there are 2^v possible subsets.
- For example, if income has three possible values, namely {low, medium, high}, then the possible subsets are {low, medium, high}, {low, medium}, {low, high}, {medium, high}, {low}, {medium}, {high}, and {}.
- We exclude the power set, {low, medium, high}, and the empty set from consideration since, conceptually, they do not represent a split.
- Therefore, there are 2 ^ v − 2 possible ways to form two partitions of the data, D, based on a binary split on A

Gini Index

- When considering a binary split, we compute a weighted sum of the impurity of each resulting partition.
- For example, if a binary split on A, partitions D into D 1 and D 2 , the Gini index of D given that partitioning is:

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

- For each attribute, each of the possible binary splits is considered.
- For a discrete-valued attribute, the subset that gives the minimum Gini index for that attribute is selected as its splitting subset.
- For continuous-valued attributes, each possible split-point must be considered
- The strategy is similar to that described earlier for information gain, where the midpoint between each pair of (sorted) adjacent values is taken as a possible split-point.
- The point giving the minimum Gini index for a given (continuous-valued) attribute is taken as the split-point of that attribute

Gini Index

- The reduction in impurity that would be incurred by a binary split on a discrete-or continuous-valued attribute A is :

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

- The attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute.
- This attribute and either its splitting subset (for a discrete-valued splitting attribute) or split-point (for a continuous-valued splitting attribute) together form the splitting criterion

Gini Index: Example

Class	Var1	Var2
A	0	33
A	0	54
A	0	56
A	0	42
A	1	50
B	1	55
B	1	31
B	0	-4
B	1	77
B	0	49

For Var1

Var1 has 4 instances (4/10) where it's equal to 1 and

6 instances (6/10) when it's equal to 0.

For Var1 == 1 & Class == A: 1 / 4 instances

For Var1 == 1 & Class == B: 3 / 4 instances

Gini Index here is 1-((1/4)^2 + (3/4)^2) = 0.375

For Var1 == 0 & Class== A: 4 / 6 instances

For Var1 == 0 & Class == B: 2 / 6 instances

Gini Index here is 1-((4/6)^2 + (2/6)^2) = 0.4444

We then weight and sum each of the splits based on the proportion of the data each split takes up.

4/10 * 0.375 + 6/10 * 0.444 = **0.41667**

Gini Index: Example

Class	Var1	Var2
A	0	33
A	0	54
A	0	56
A	0	42
A	1	50
B	1	55
B	1	31
B	0	-4
B	1	77
B	0	49

For Var2 (Let's Threshold T>=32)

Var2 has 8 instances (8/10) where it's >= 32 and

2 instances (2/10) when it's < 32.

For Var2 >=32 & Class == A: 5 / 8 instances

For Var2 >=32 & Class == B: 3 / 8 instances

Gini Index here is 1-((5/8)^2 + (3/8)^2) = 0.46875

For Var2 < 32 & Class== A: 0 / 2 instances

For Var1 < 32 & Class == B: 2 / 2 instances

Gini Index here is 1-((0/2)^2 + (2/2)^2) = 0

We then weight and sum each of the splits based on the proportion of the data each split takes up.

8/10 * 0.46875 + 2/10 * 0 = **0.375**

Gini Index: Example

Class	Var1	Var2
A	0	33
A	0	54
A	0	56
A	0	42
A	1	50
B	1	55
B	1	31
B	0	-4
B	1	77
B	0	49

For Var1 = 0.41667

For Var2 (T>=32) = **0.375**

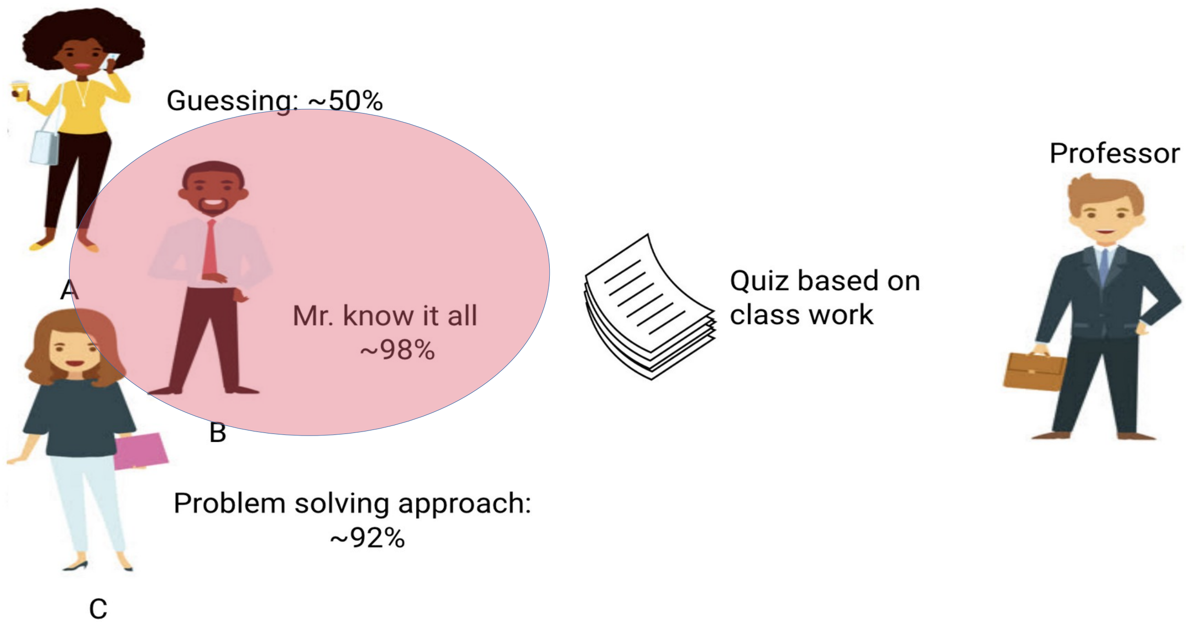
Based on these results, Var2>=32 is selected as the split.(since its weighted Gini Index is smallest)

The next step would be to take the results from the split and further partition.

Comparison of ID3, C4.5 and CART

Characteristic / Algorithm	Splitting Criteria	Attribute Type	Missing Values	Pruning Strategy
ID3	Information gain	only Categorical	Do not handle	No Pruning
C4.5	Gain Ratio	Categorical and Numeric	Handle	Error based pruning
CART	Gini Index with binary split	Categorical and Numeric	Handle	Cost-Complexity pruning

Overfitting :



- Model that models the training data too well
- This means that the noise or random fluctuations in the training data is pinked up and learned as concepts by the model.
- The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize

Solution to Overfitting

- The solution can be grouped into two classes:

  1. Pre Pruning: approaches that stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data

  2. Post Pruning: approaches that allow the tree to overfit the data, and then post-prune the tree

Overfitting Solution: Prepruning

- By halting its construction early (e.g., by deciding not to further split or partition the subset of training tuples at a given node).
- Upon halting, the node becomes a leaf.
- The leaf may hold the most frequent class among the subset tuples
- If partitioning the tuples at a node would result in a split that falls below a pre-specified threshold, then further partitioning of the given subset is halted.
- There are difficulties, however, in choosing an appropriate threshold.
  - High thresholds : oversimplified trees
  - low thresholds : very little simplification.