Decision Trees

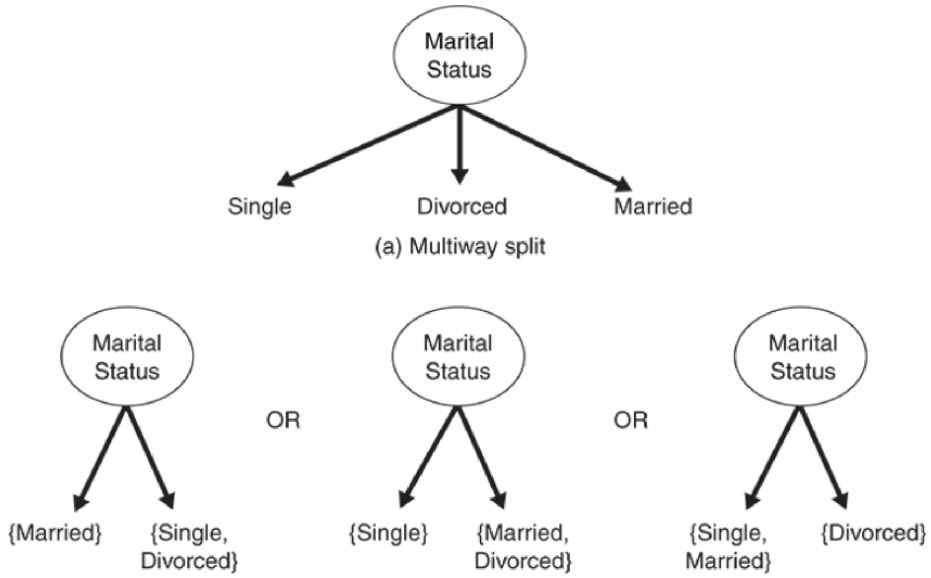
Splitting Nodes

- How should we split a node?
  - What criteria should we use?
  - Can we measure if one split is better than another?
- When do we stop splitting?
  - Stop when all instances in a node have the same class
  - Stop when all instances in a node have the same values for their attributes
    - Label node as majority class
  - Other reasons to stop (Early termination)?

Different Test Conditions for Splitting

- Depends on attributes
  - Binary
    - Home Owner -> (Yes, No)
  - Nominal
    - Marital Status -> (Single, Divorced, Married)
  - Ordinal
    - Shirt Size -> (Small, Medium, Large, X-Large)
  - Continuous
    - Annual Income -> ($<$ 45k, 45k - 75k, $>$ 75k)
- Depends on the number of ways to split
  - 2-way split (Two Paths)
  - Multi-way split (More than two paths)

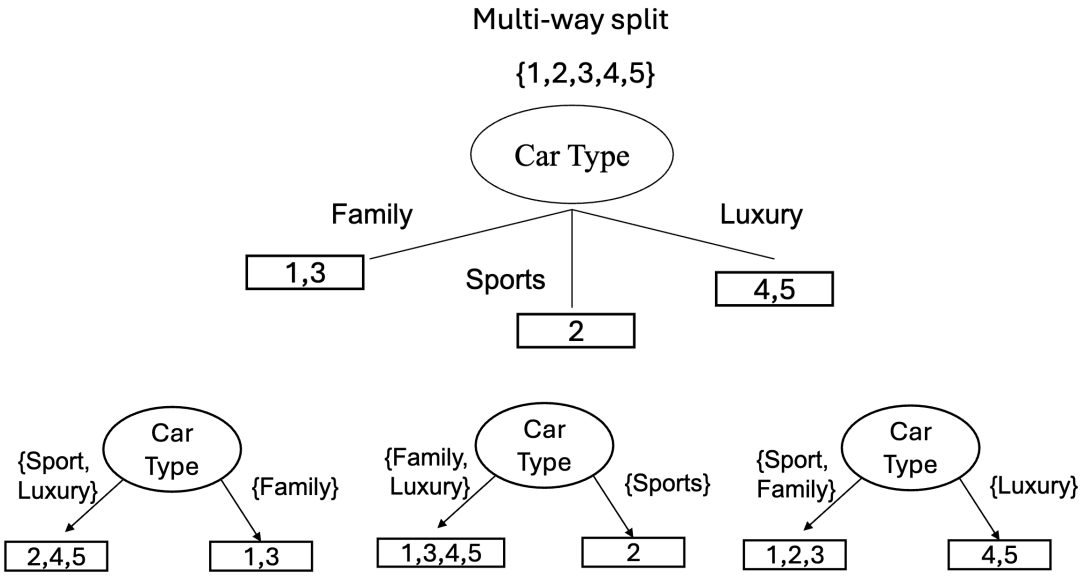
Binary vs. Multiway Split



Nominal Attribute

- Consider the data:

ID	CarType	Size	Price
1	Family	Medium	100
2	Sports	Large	200
3	Family	Small	150
4	Luxury	Large	500
5	Luxury	Medium	200



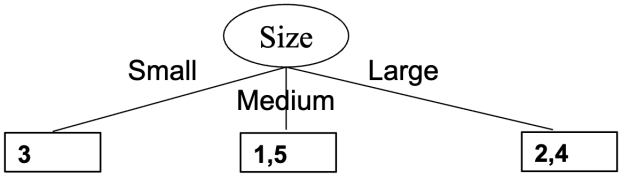
Test Condition: Ordinal

- The test condition for an ordinal attribute can also be expressed in two ways: Multi-way split and binary split

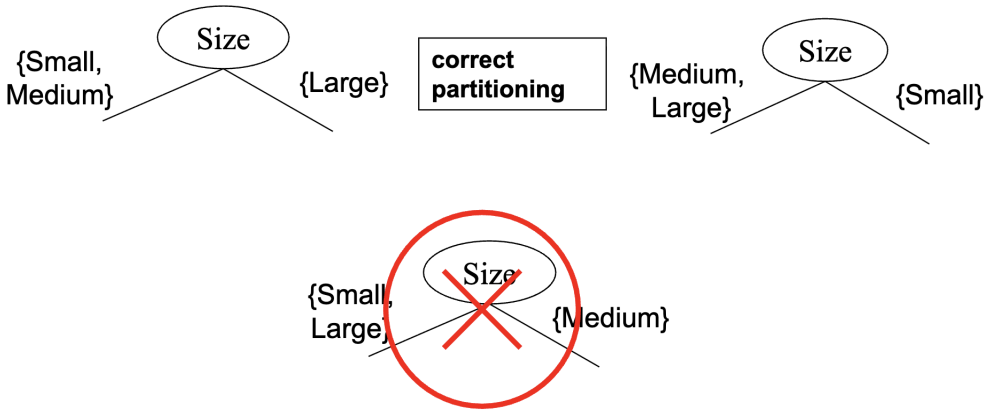
- Consider the data:

ID	CarType	Size	Price
1	Family	Medium	100
2	Sports	Large	200
3	Family	Small	150
4	Luxury	Large	500
5	Luxury	Medium	200

- For the multi-way split, it is the same as that of a nominal attribute



- For a binary split, we must maintain the order of our attributes



Test Condition: Continuous

- To specify a test condition for a continuous attribute, we need to use discretization
  - Transforming a continuous attribute to an ordinal discrete attribute
- To discretize a continuous attribute, we need to complete two sub-tasks:
  - Decide how many categories to have?
  - How to map the values to these categories?

Discretization

- The number of categories depends on the specific task
- Suppose we want the continuous attribute divided into n categories (intervals)
- $n - 1$ split points are required

- All the values in the one interval are mapped to the same categorical value. The result can be represented as:

$$\{[v_1, v_2), [v_2, v_3), \ldots, [v_n, v_{n+1})\}$$

where v_1 and v_{n+1} denote the minimum and maximum value of the attribute respectively, v_i, $i = 2, \ldots, n$ denotes the split points

- Consider the data:

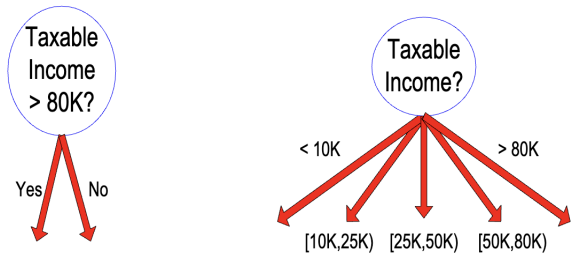
ID	Age
2204	10
2301	21
2518	23
3201	40

- Suppose Age is in $[10, 40]$, and we want 2 categories.
- Common approaches:

1. The equal width approach: each interval has the same width. $[10, 25), [25, 40]$. (Both have a width of 15)
2. The equal frequency approach: Each interval has the same number of objects. $[10, 22), [22, 40]$. (Both have 2 objects)
3. It is also called "binary split" when the attribute is transformed to two categories

Test Conditions: Continuous (Continued)

- Specify the test condition for a continuous attribute
  - The test condition for a continuous attribute can be expressed in two ways: multi-way split and binary split
  - n-categories discretization produces an n-way outcome
  - Binary discretization produces two outcomes



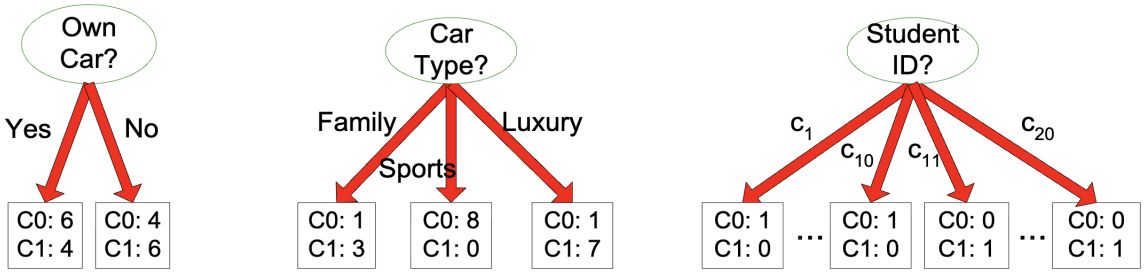
(i) Binary split (ii) Multi-way split

- In the left case, we use "80" as the split point to discretize the taxable income into two categories ($\leq$ 80, $>$ 80). Accordingly, the test condition produces two outcomes.

- In the right case, we use 10k, 25k, 50k, and 80k as the split points to discretize the taxable income into five categories. Accordingly, the test condition produces five outcomes.

How to determine the best split

- Suppose that we have a dataset and it has 20 objects that belong to two classes, class 0 and class 1
- Before splitting, suppose that each class has 10 records
  - 10 records of class 0
  - 10 records of class 1
- Let C0 and C1 denote the class labels of class 0 and class 1 respectively



- First Picture: The test condition of "Own Car" splits the dataset into two subsets, and each subset has 10 objects
  - Left subset: 6 records belong to C0 and 4 records belong to C1
  - Right subset: 4 records belong to C0 and 6 records belong to C1
- Second Picture: The test condition of "Car Type" splits the dataset into three subsets (4 objects, 8 objects, 8 objects)
  - First subset: 1 record belongs to C0 and 3 records belong to C1
  - Second subset: 8 record belongs to C0 and no records belong to C1
  - Third subset: 1 record belongs to C0 and 7 records belong to C1
- Third Picture: The test condition of "Student ID" splits the dataset into 20 subsets and each subset contains just 1 object
  - In each subset, the object belongs to either C0 or C1
- Can we evaluate the goodness of a split for the classification task? How?

Determining the best splits (goodness)

- Terminology: The class distribution: $p(i \mid t)$ for node t
  - Probability of class i given that we are in node t (often written as $p(i)$
  - Fraction of records belonging to class i at a node t

  - In a two-class problem, the class distribution is written as: (p_0, p_1), where $p_1 = 1 - p_0$

- Example:
  - C0: 9, C1: 1
  - In this node, there are 10 objects
  - 9 out of 10 objects belong to C0, so $p\,(\text{C0}) = 0.9$
  - 1 out of 10 objects belong to C1, so $p\,(\text{C1}) = 0.1$
  - Two-class, so the class distribution can be written as $(0.9, 0.1)$
- What is the class distribution of the following three nodes?

C0 : 0 C1 : 10	C0 : 12 C1 : 8	C0 : 5 C1 : 5
$(0, 1)$	$(0.6, 0.4)$	$(0.5, 0.5)$

- How does that help us?
  - Used to evaluate the impurity of a node

Node Impurity

- Node impurity is a measure of how mixed the classes are, and can be viewed as related to how unlikely we are to randomly select an instance from a node and guess the class
- The smaller the degree of impurity, the more skewed the class distribution
- Used to evaluate a split
- For two classes:
  - A node with class distribution $(0, 1)$ has zero impurity
  - A node with class distribution $(0.5, 0.5)$ has the highest impurity
- How do we measure / calculate the impurity of a node?

Measuring Node Impurity

- c is the number of different classes in a node
- $p_i\,(t)$ is the distribution of a given class i in node t (probability of seeing i in node t)

1. Gini Index
$$\text{Gini Index} = 1 - \sum_{i=0}^{c-1} p_i\,(t)^2$$
2. Entropy
$$\text{Entropy} = -\sum_{i=0}^{c-1} p_i\,(t)\, log_2\,(p_i\,(t))$$
3. Misclassification Error, also called Classification Error
$$\text{Classification Error} = 1 - \max\,(p_i\,(t))$$