Introduction to Artificial Intelligence

Decision Tree and Random Forest

Lionel Fillatre

2021-2022

Outline of the Lecture

- Introduction
- Tree Induction
- Best Split
- Practical Issues
- Random Forest
- Conclusion

Introduction

Classification Task

Given:

- $\mathcal{X}$ is an instance space
- X is an instance defined as $X = (X_1, \dots, X_M)$ where X_i is a discrete/continuous variable (attributes).
- $\mathcal{Y}$ is a finite class set of labels.
- Y is a label.
- Training data $D \subseteq \mathcal{X} \times \mathcal{Y} = \{(x, y) : x \in X, y \in Y\}$.

Find:

- Class $y \in Y$ of a test instance $x \in X$.

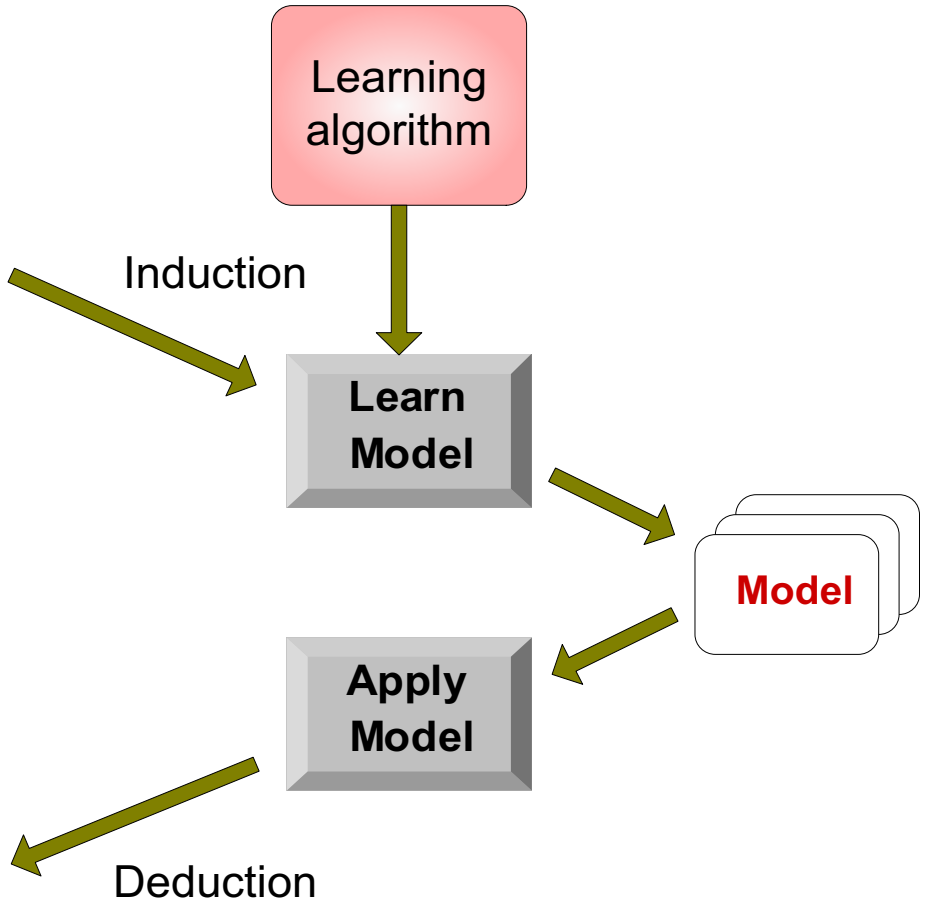
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

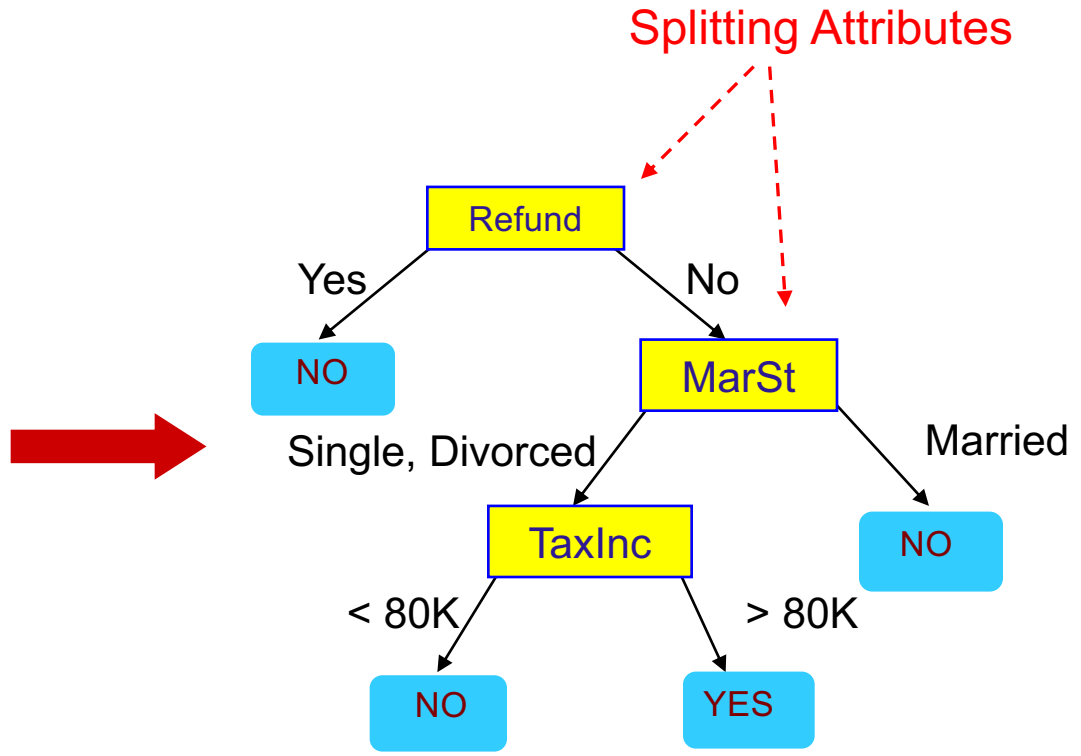
Test Set



Example of a Decision Tree

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

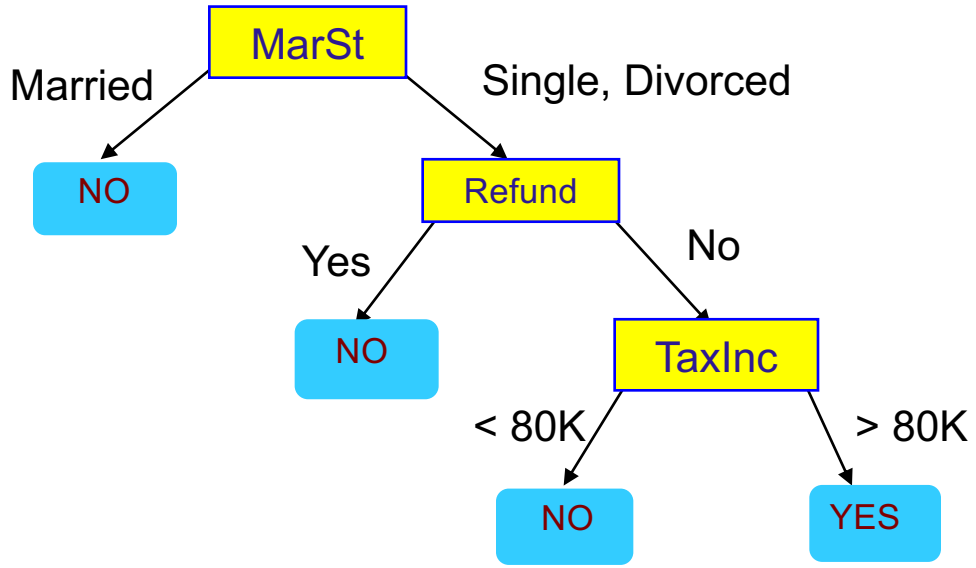
Training Data



Model: Decision Tree

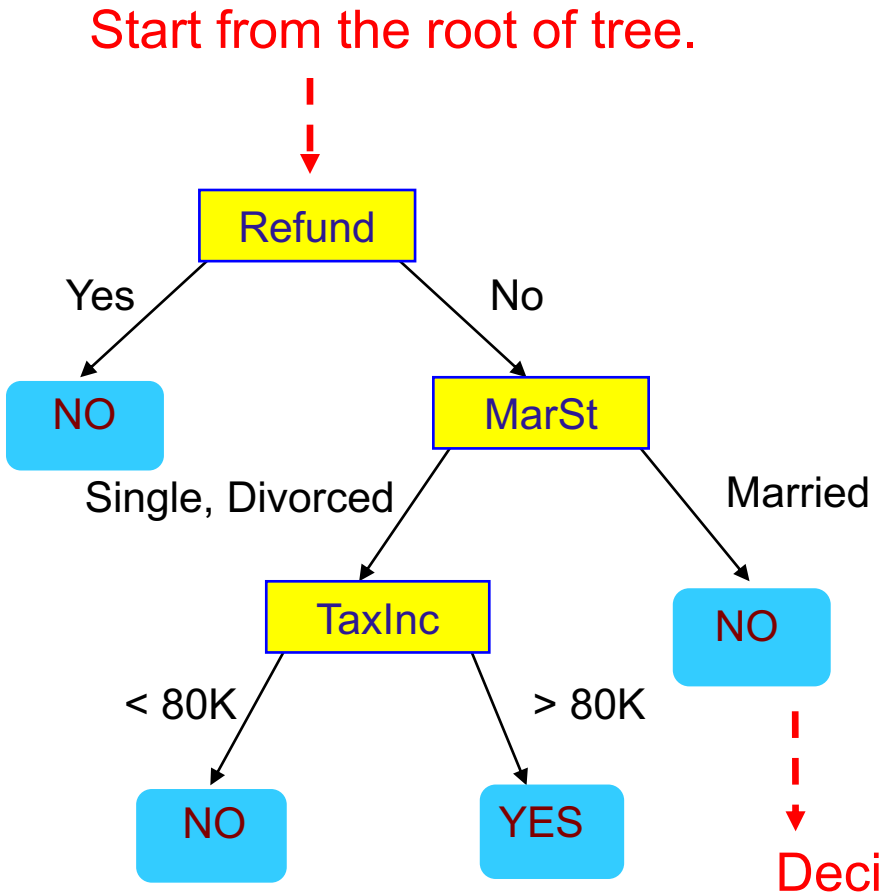
Another Example of Decision Tree

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**



There could be more than one tree that fits the same data!

Apply Model to Test Data

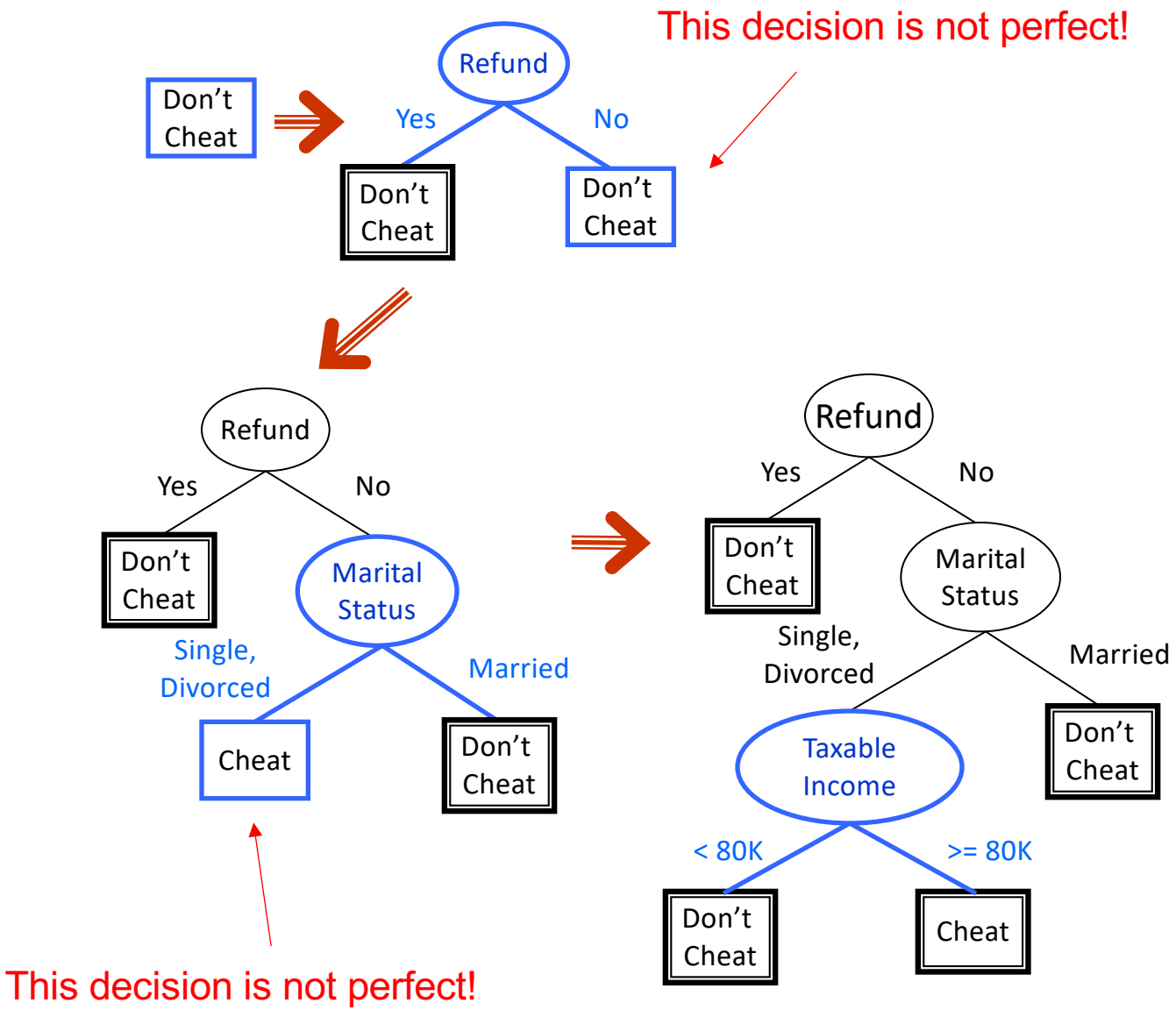


Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Tree Induction

Hunt's Algorithm



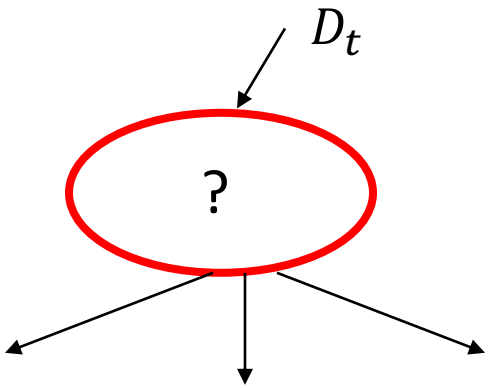
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t and $\{y_1, \dots, y_{n_c}\}$ be the class labels.

- General Procedure:
  - If D_t contains records that belong to the same class y_t, then t is a leaf node labeled as y_t
  - If D_t is an empty set, then t is a leaf node labeled by the label of its parent class
  - If all records in D_t have identitical attributes (except for the class label), then t is a leaf node labeled by the same class label as the majority class of training records associated with this node.
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets.
  - Recursively apply the procedure to each subset.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Tree Induction

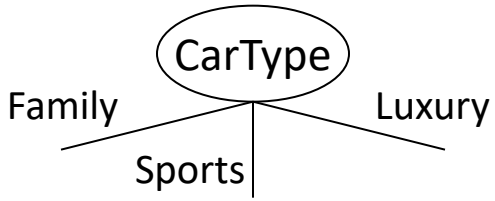
- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

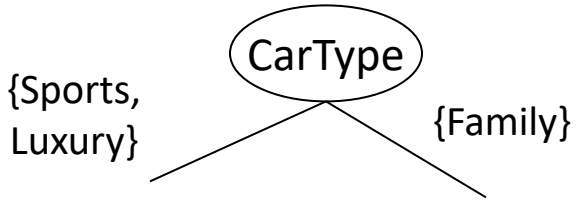
Best Split

Splitting Based on Nominal Attributes

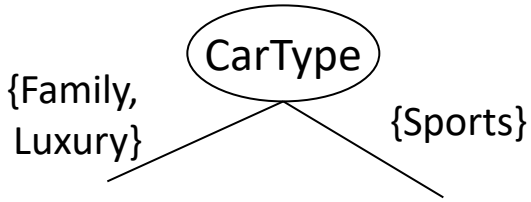
- Multi-way split: Use as many partitions as distinct values.



- Binary split:
  - Divides values into two subsets.
  - Need to find optimal partitioning.

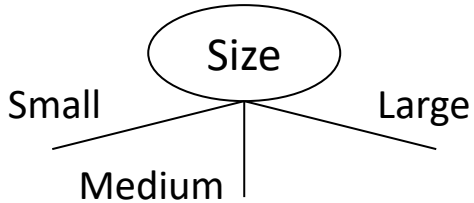


OR

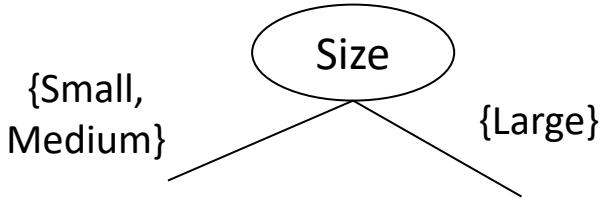


Splitting Based on Ordinal Attributes

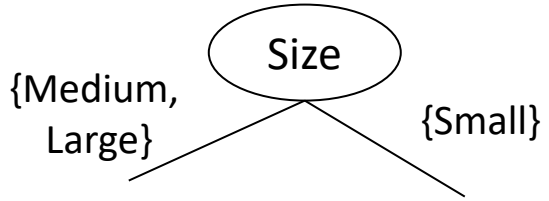
- Multi-way split: Use as many partitions as distinct values.



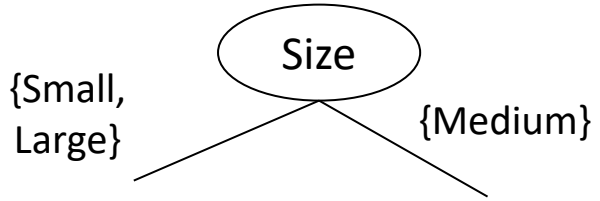
- Binary split:
  - Divides values into two subsets.
  - Need to find optimal partitioning.



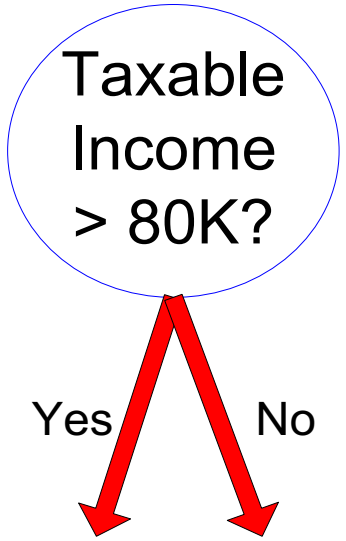
OR



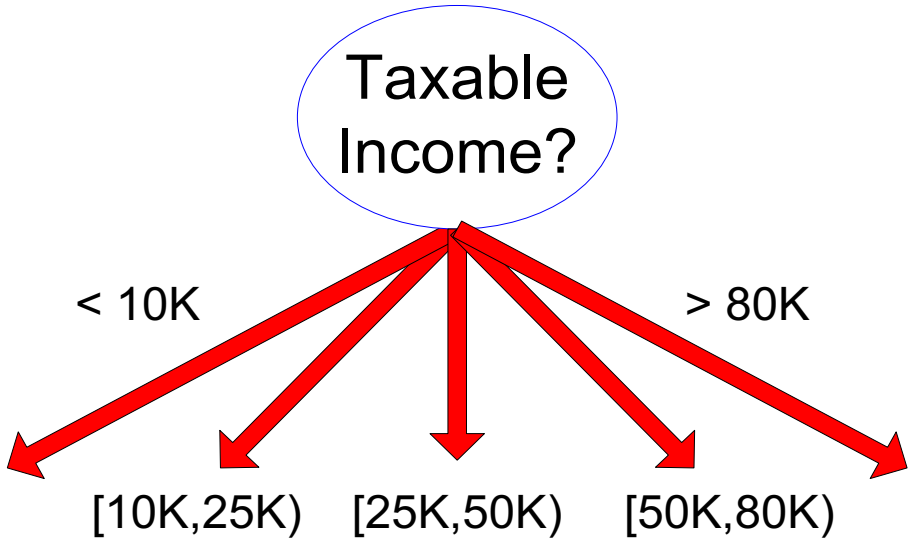
- What about this split?



Splitting Based on Continuous Attributes



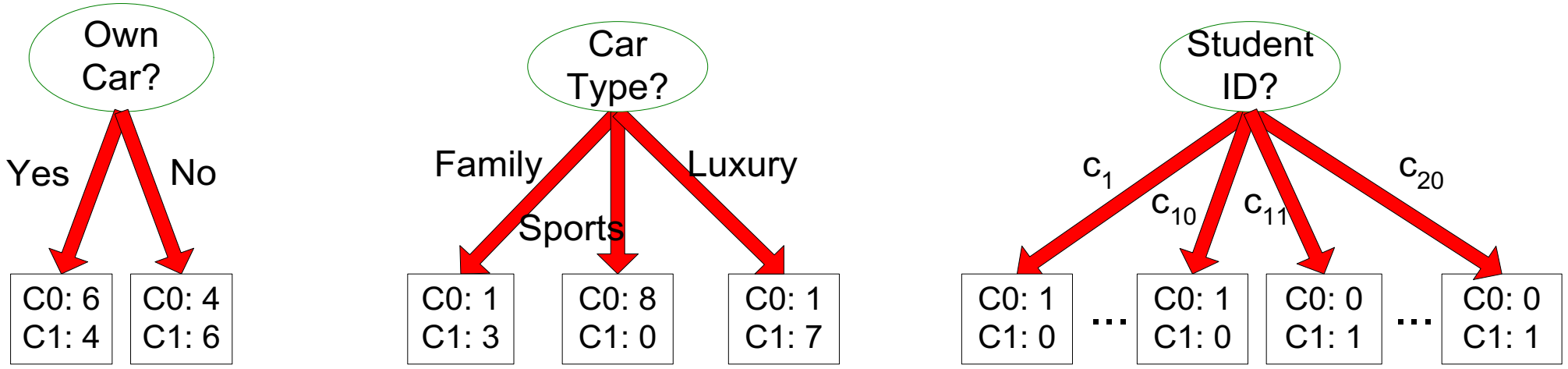
(i) Binary split



(ii) Multi-way split

How to determine the Best Split

- Before Splitting:
  - 10 records of class 0,
  - 10 records of class 1



- Which test condition is the best?

How to determine the Best Split

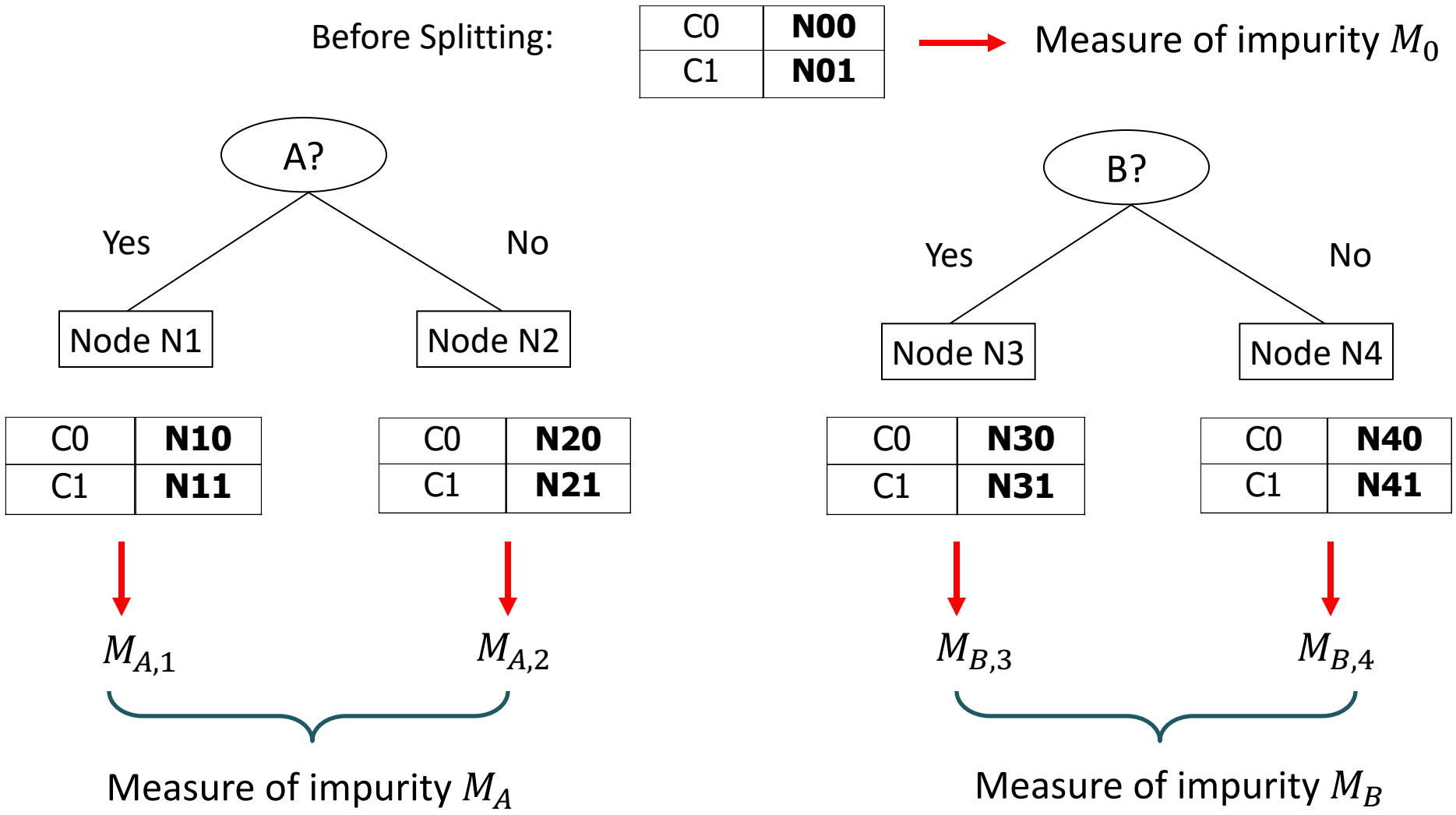
- Greedy approach:
  - Nodes with homogeneous class distribution are preferred

- Need a measure of node impurity:

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous,	Homogeneous,
High degree of impurity	Low degree of impurity

- Some measures:
  - Gini Index
  - Entropy
  - Misclassification error

How to Find the Best Split



Gain: $M_0 - M_A$ versus $M_0 - M_B$.

Measure of Impurity: GINI

- **Gini Index** for a given node t:

$$GINI(t) = 1 - \sum_{j=1}^{n_c} [p(j|t)]^2$$

where $p(j|t)$ is the relative frequency of class j at node t.

  - Maximum $(1 - 1/n_c)$ when records are equally distributed among all classes, implying least interesting information

  - Minimum (0.0) when all records belong to one class, implying most interesting information

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Examples for computing GINI

$$GINI(t) = 1 - \sum_{j=1}^{n_c} [p(j|t)]^2$$

Node t

C1	**0**
C2	**6**

$p(j = C1|t) = \frac{0}{6} = 0$ $\quad$ $p(j = C2|t) = \frac{6}{6} = 1$

$GINI(t) = 1 - p(j = C1|t)^2 - p(j = C2|t)^2 = 1 - 0 - 1 = 0$

C1	**1**
C2	**5**

$p(j = C1|t) = \frac{1}{6}$ $\quad$ $p(j = C2|t) = \frac{5}{6}$

$GINI(t) = 0.278$

C1	**3**
C2	**3**

$p(j = C1|t) = \frac{3}{6} = \frac{1}{2}$ $\quad$ $p(j = C2|t) = \frac{3}{6} = \frac{1}{2}$

$GINI(t) = 0.5$

Splitting Based on GINI

- Used in CART, SLIQ, SPRINT.
- When a node t is split into k partitions (children), the quality of split is computed as a weighed sum of GINI Indices:

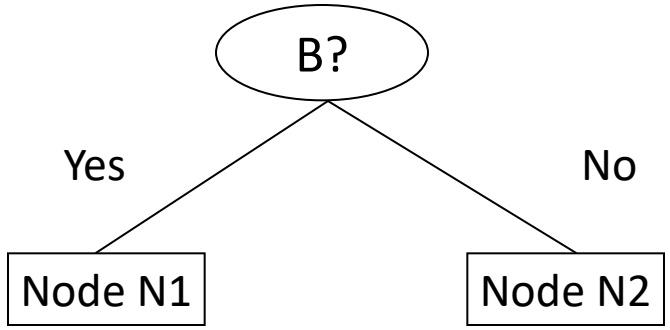
$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where,

- n_i = number of records at child i,
- n = number of records at node t.

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.



	Parent
C1	**6**
C2	**6**
Gini = 0.500	

	N1	N2
C1	**5**	**1**
C2	**2**	**4**
Gini=0.333		

Gini(N1)
$= 1 - (5/7)^2 - (2/7)^2$
$= 0.408$

Gini(N2)
$= 1 - (1/5)^2 - (4/5)^2$
$= 0.320$

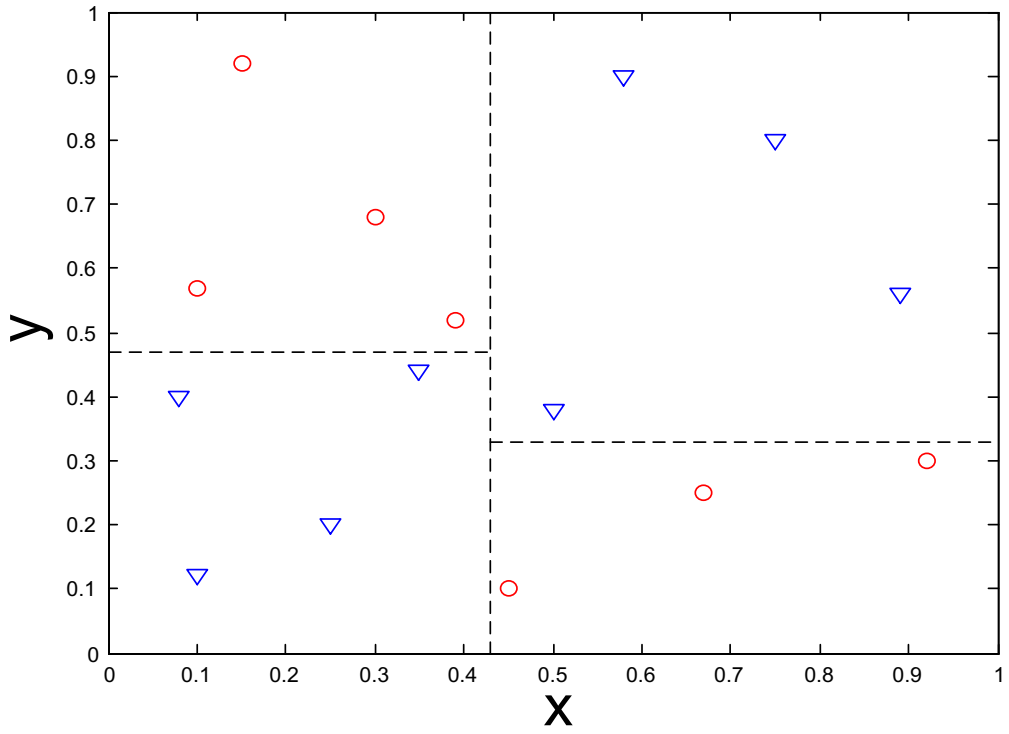
$$GINI_{split} = 7/12 * 0.408 + 5/12 * 0.320 = 0.371$$

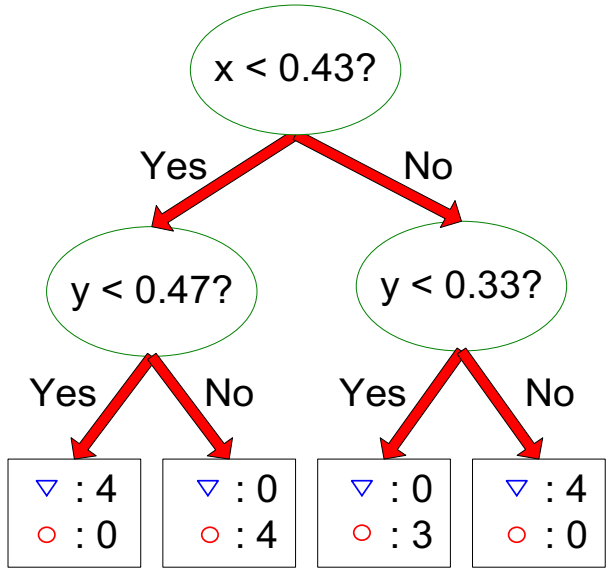
Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination

Practical Issues

Decision Boundary

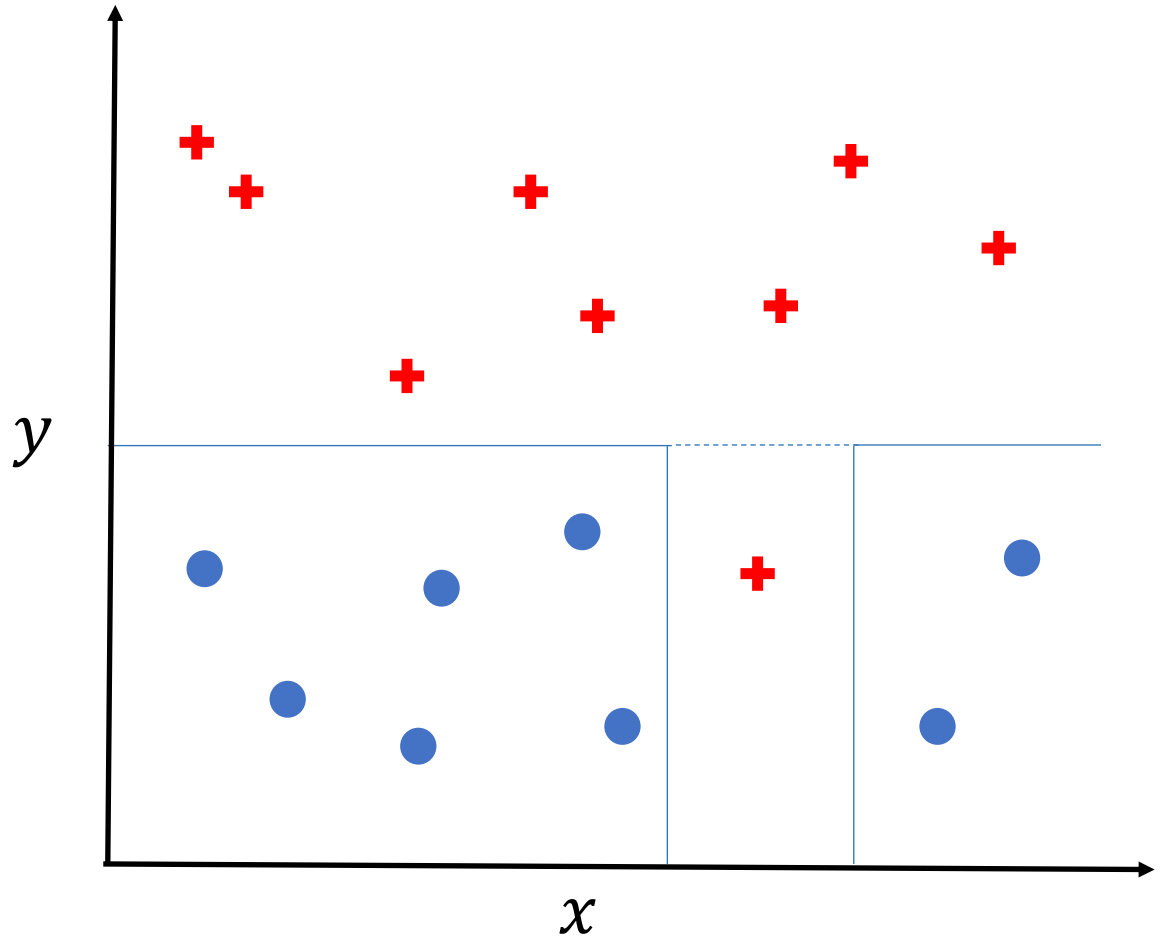




- Border line between two neighboring regions of different classes is known as decision boundary
- Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

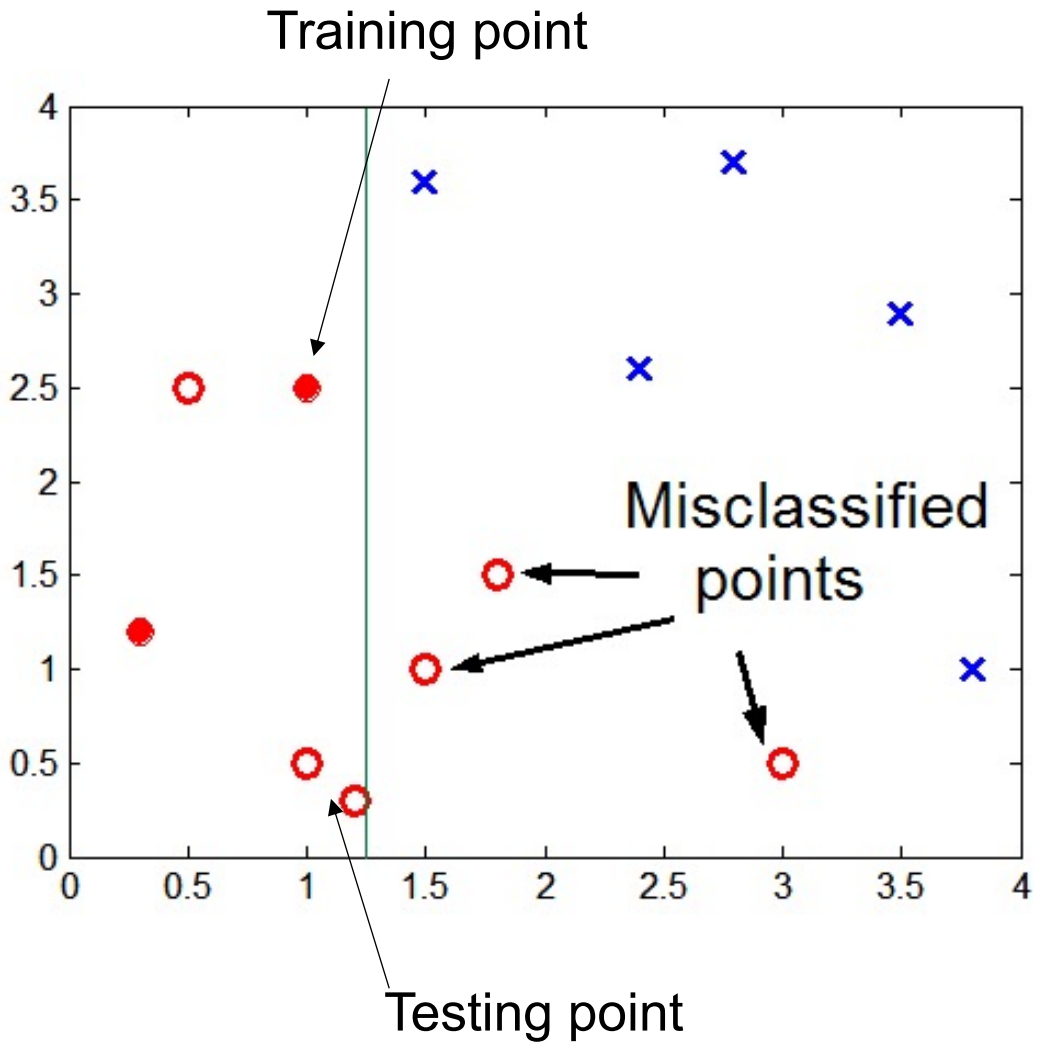
Overfitting due to Noise

- Decision boundary is distorted by noise point!



Overfitting due to Insufficient Examples

- Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels of that region
- Insufficient number of training records in the region causes the decision tree to predict the test examples using other training records that are irrelevant to the classification task



Random Forest

Random Forest

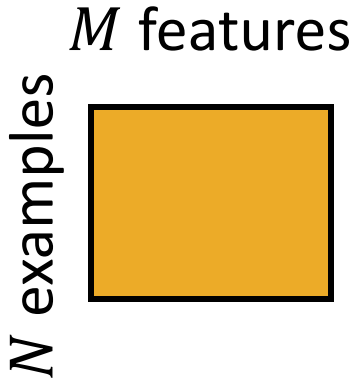
- A problem with decision trees like CART is that they are greedy. They choose which variable to split on using a greedy algorithm that minimizes error.

- Combining predictions from multiple trees should work better.

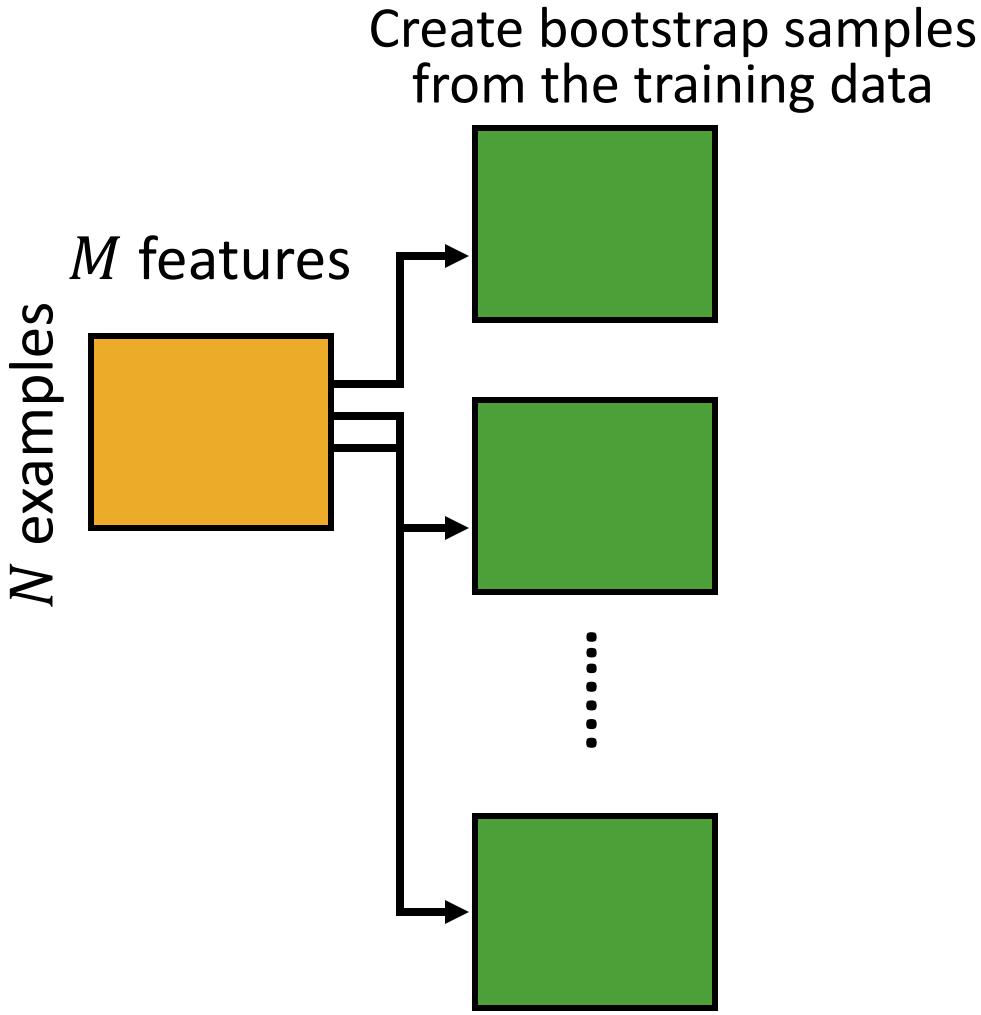
- Random forest changes the algorithm for the way that the sub-trees are learned so that the resulting predictions from all of the subtrees have less correlation.

Random Forest Classifier

Training Data

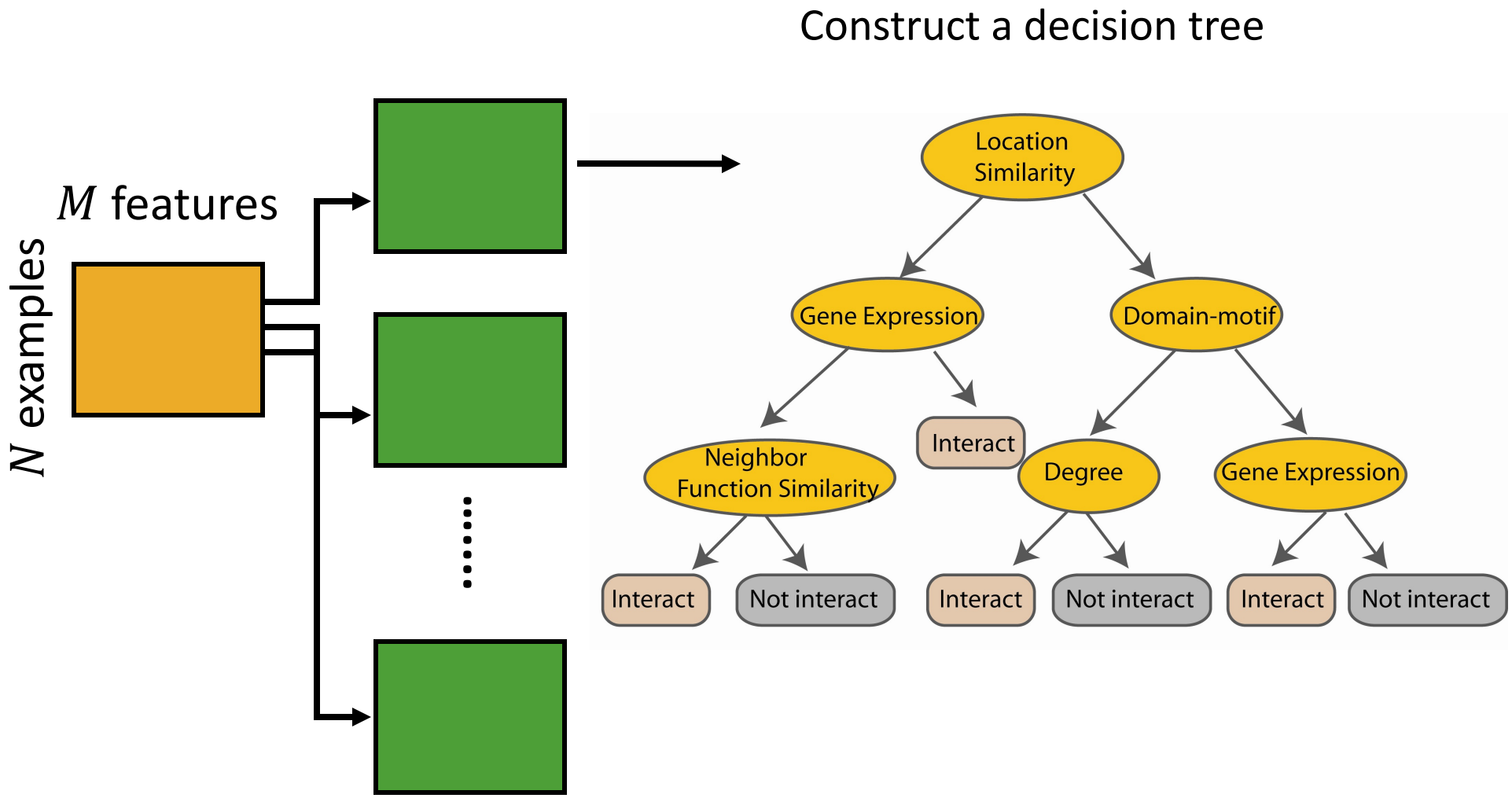


Random Forest Classifier



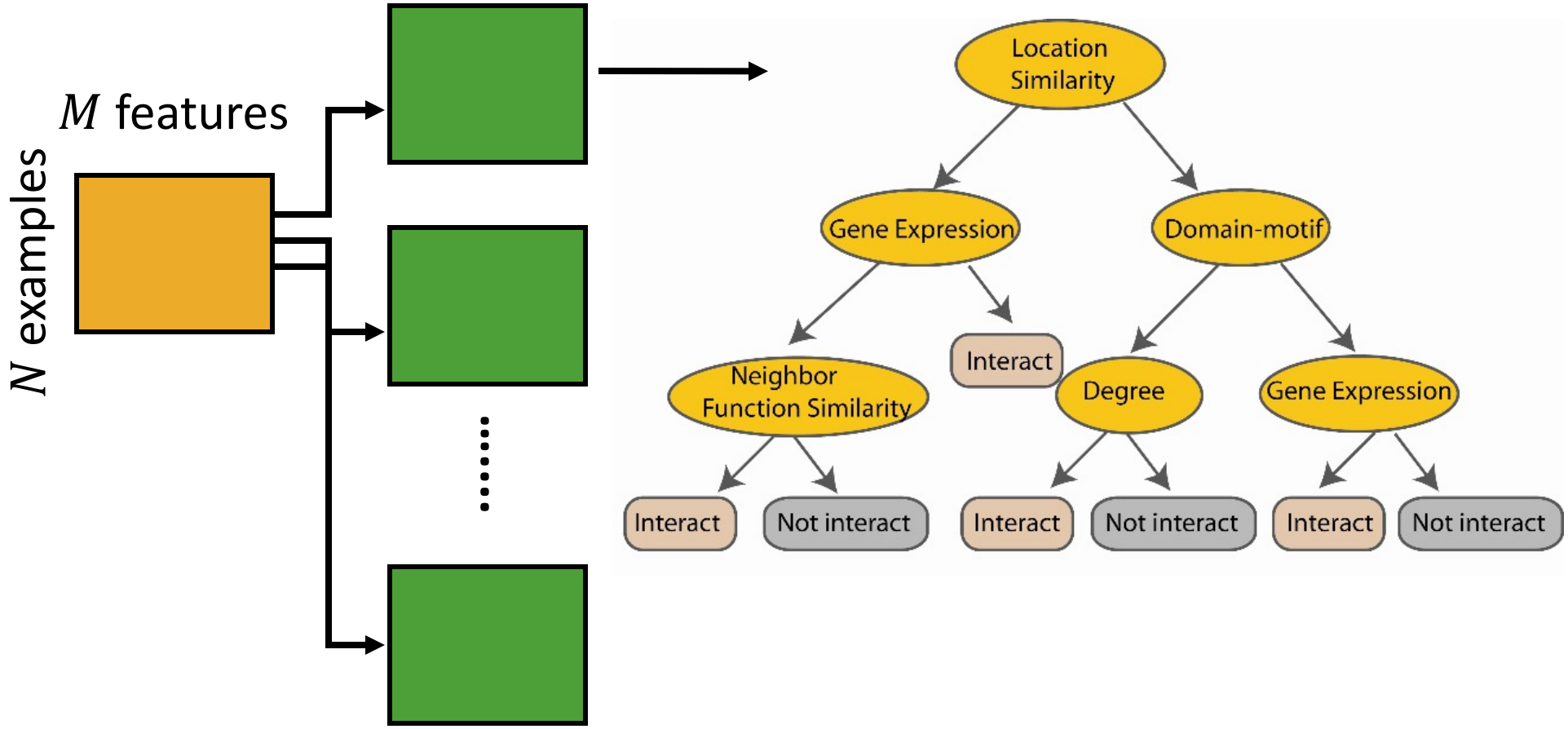
Bootstrap samples: create many (e.g. 100) random sub-samples of the training dataset with replacement (meaning we can select the same value multiple times).

Random Forest Classifier

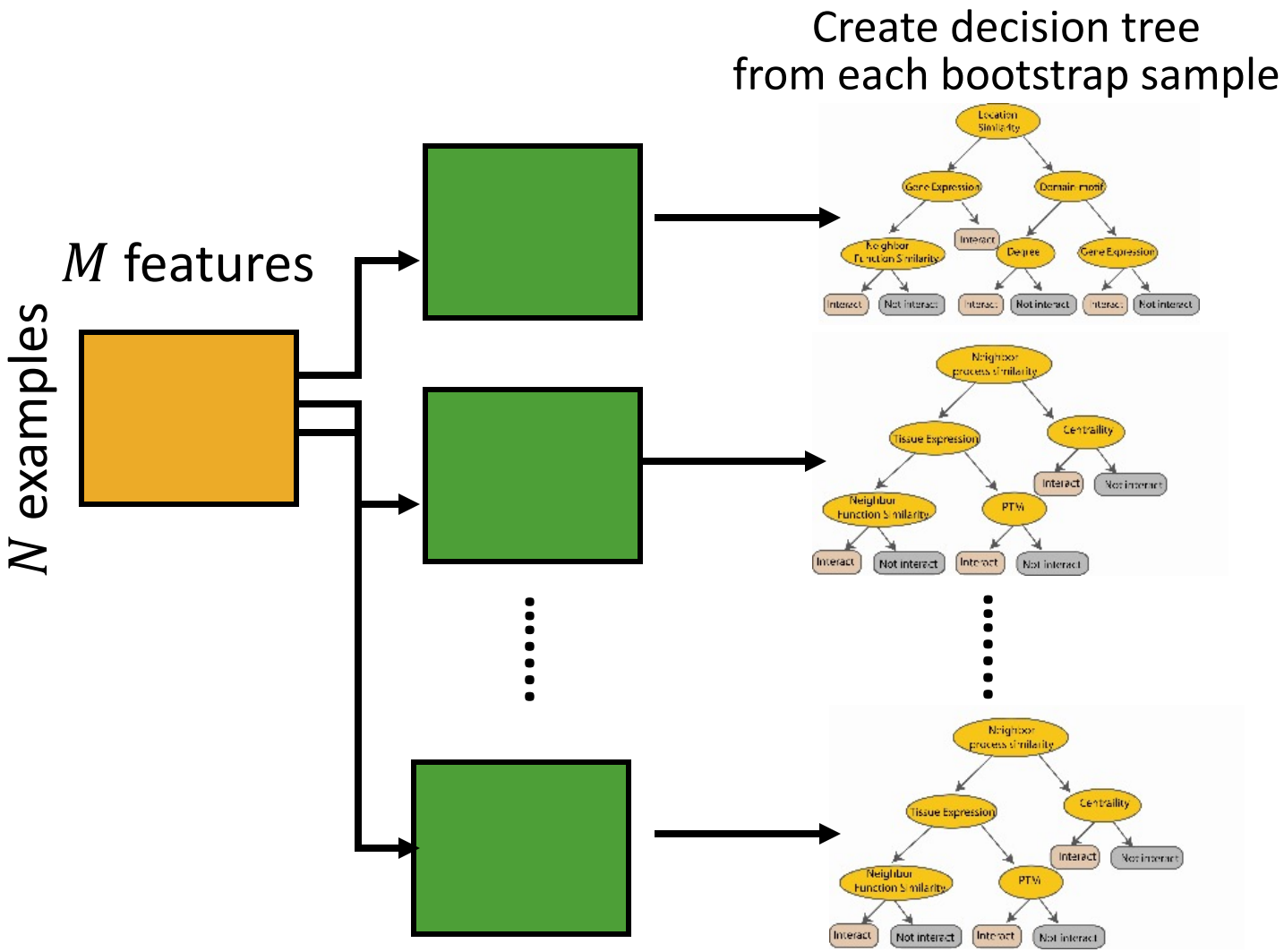


Random Forest Classifier

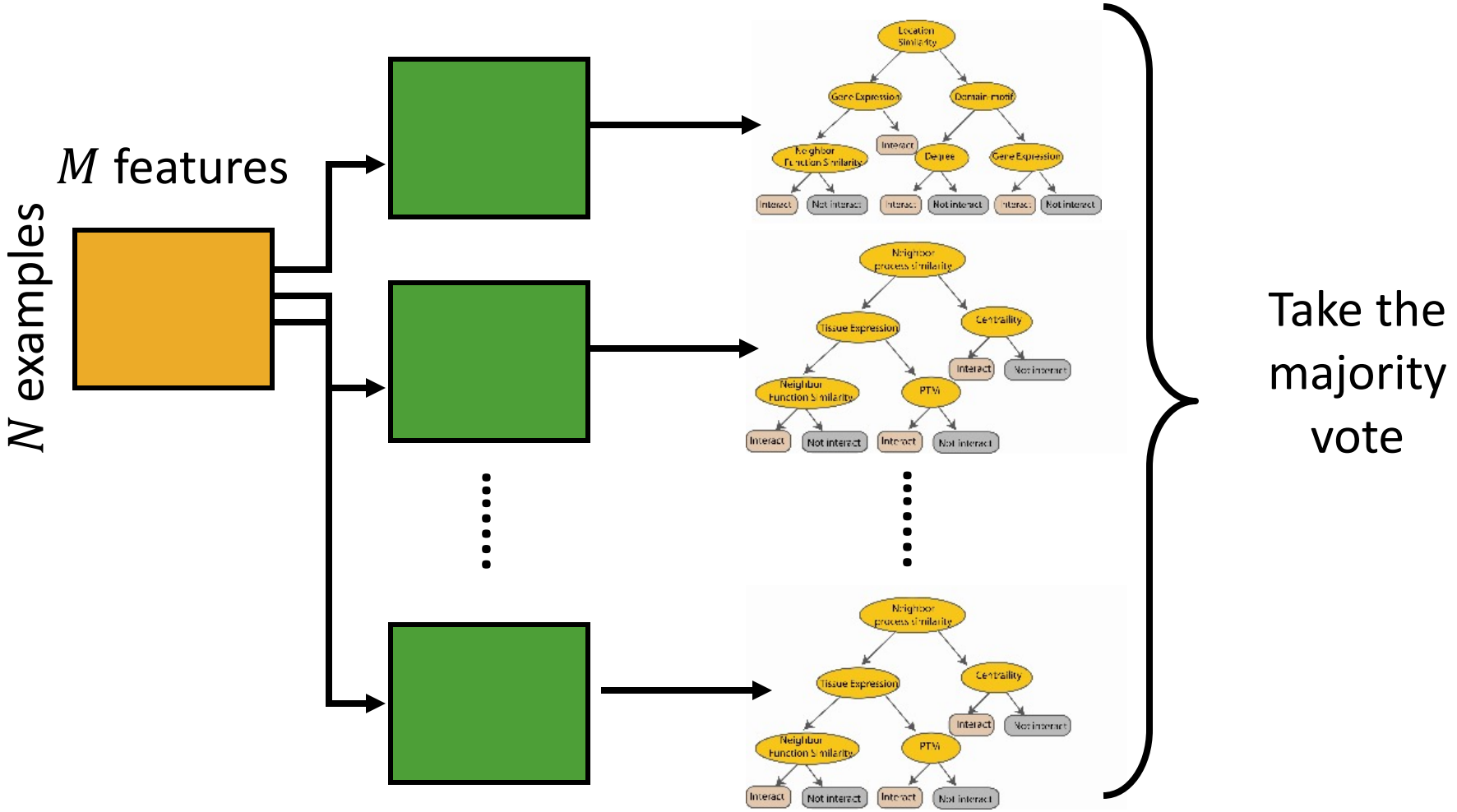
At each node,
choose the split feature at random among $m < M$ features



Random Forest Classifier



Random Forest Classifier



Conclusion

Conclusion

- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Accuracy is comparable to other classification techniques for many simple data sets
- Random forest: a famous classification method that is based strongly on an ensemble of decision trees
- Many libraries in many programming languages
- Tuning is important (overfitting, etc.)