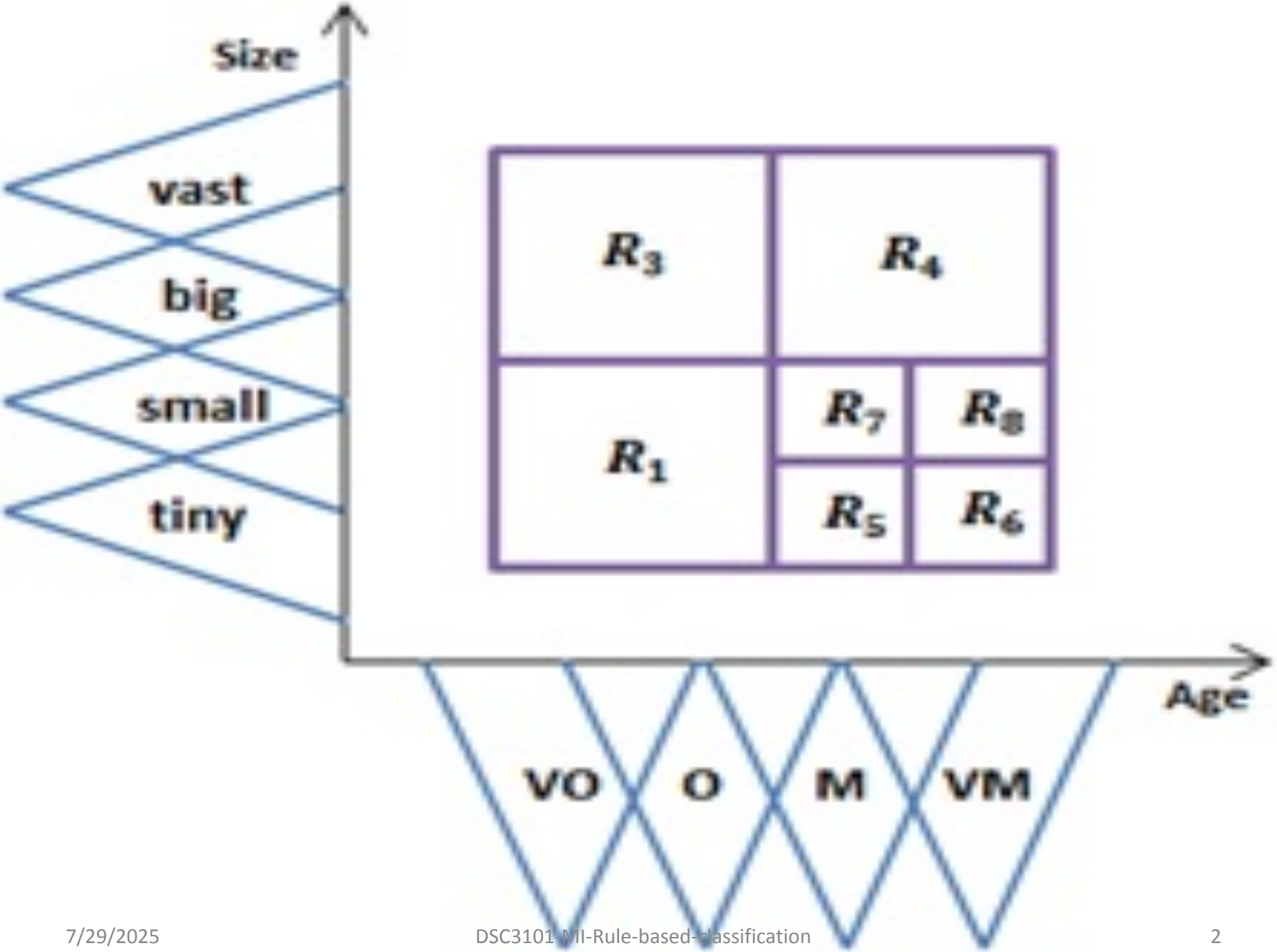
DSC3101 – MI - Rule-Based Classification

Working, Rule Ordering, Construction, and Rule Extraction

Nilina Bera | July 2025


Size
vast
big
small
tiny
R_3
R_4
R_1
R_7
R_8
R_5
R_6
Age
VO
O
M
VM

Course Name: Introduction to Data Mining					
Course Code: DSC3101					
Contact Hours per week:	L	T	P	Total	Credit points
	3	0	0	3	3

1. Course Outcomes

After completion of the course, students will be able to:

DSC3101.1 Remember different terminologies in respect of data mining techniques.

DSC3101.2 Understand and apply the various data preprocessing methods as and when required.

DSC3101.3 Understand and apply different classification, clustering algorithms to solve various real life problems.

DSC3101.4 Analyze various methods for mining the frequent patterns in different real life situations.

DSC3101.5 Apply several ensemble techniques, like bagging, boosting, random forests etc. as and when required.

DSC3101.6 Evaluate various data mining techniques to solve real-world problems.

2. Detailed Syllabus

Module 1 [9L]

Introduction: Basics of Data Mining? Why do we need data mining? Data mining Architecture, Data mining goals and techniques. Challenges in Data Mining.

Data pre-processing: Data cleaning, Data transformation and Data reduction. Applications

Rule-based Classification: How a rule-based classifier works, rule-ordering schemes, how to build a rule-based classifier, direct and indirect methods for rule extraction.

What is Rule-Based Classification?

- • A classification approach that uses IF-THEN rules to classify data.
- • Rules are human-readable and interpretable.
- • Each rule typically has the form:
- IF <condition> THEN <class label>

Rule-Based Classifier

A rule-based classifier is a technique for classifying records using a collection of "if . . .then. . ." rules. Table 1 shows an example of a model generated by a rule-based classifier for the *vertebrate classification problem*. The rules for the model are represented in a *disjunctive normal form*, $R = (r_1 \vee r_2 \vee \ldots r_k)$, where R is known as the rule set and r_i's are the classification rules or disjuncts.

Table 1. Example of a rule set for the vertebrate classification problem.

r_1: (Gives Birth = no) $\wedge$ (Aerial Creature = yes) $\longrightarrow$ Birds

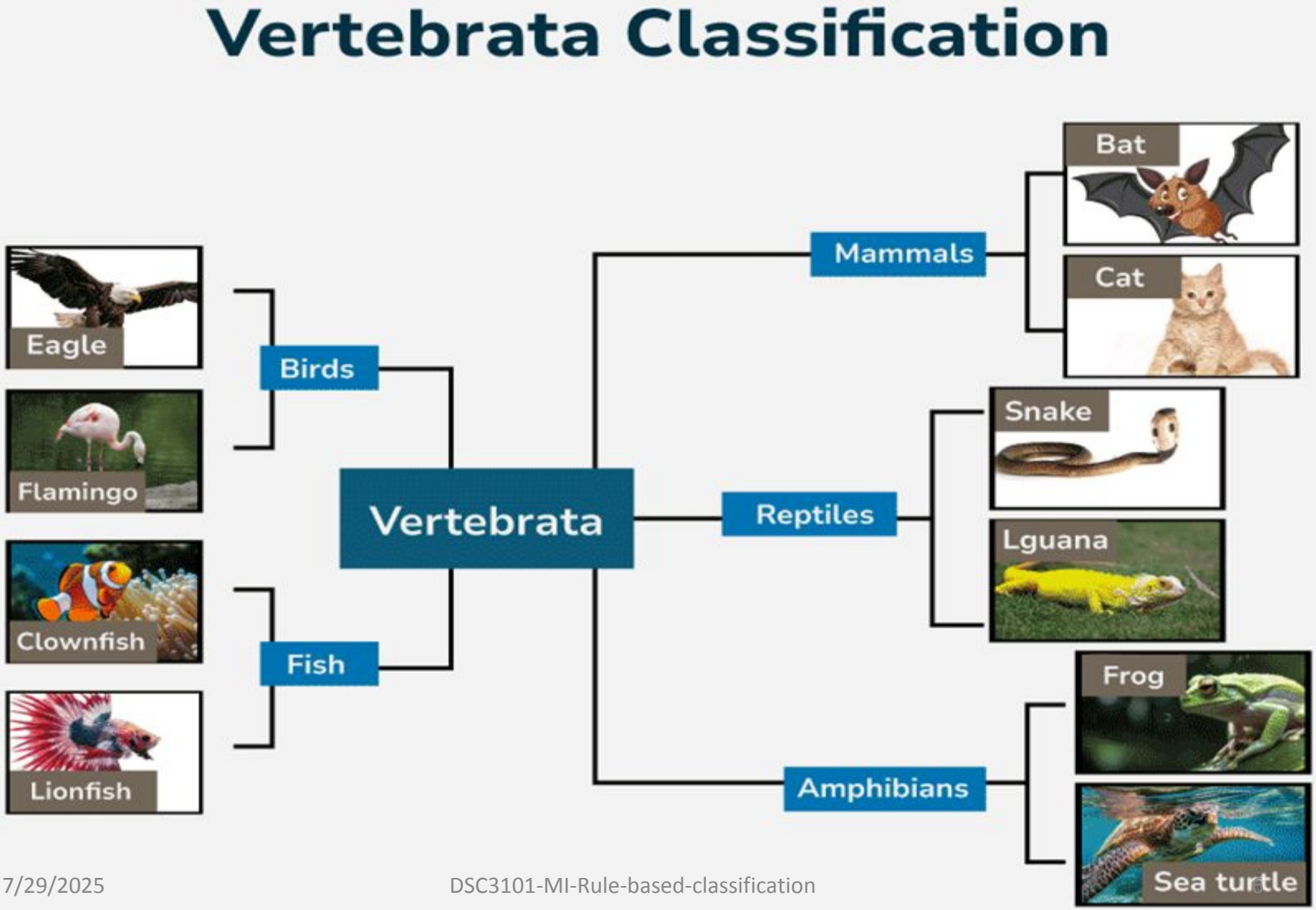
r_2: (Gives Birth = no) $\wedge$ (Aquatic Creature = yes) $\longrightarrow$ Fishes

r_3: (Gives Birth = yes) $\wedge$ (Body Temperature = warm-blooded) $\longrightarrow$ Mammals

r_4: (Gives Birth = no) $\wedge$ (Aerial Creature = no) $\longrightarrow$ Reptiles

r_5: (Aquatic Creature = semi) $\longrightarrow$ Amphibians

Vertebrate classification involves grouping animals with backbones (vertebrates) into different categories based on shared characteristics



Rule-based Classifier (Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) → Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) → Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

What about Penguin? Not a Bird? Platypus? Mammals?

Rule-Based Classifier

Each classification rule can be expressed in the following way:

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) → Birds

$$r_i : \ (Condition_i) \longrightarrow y_i. \qquad (1)$$

The left-hand side of the rule is called the **rule antecedent** or **precondition**. It contains a conjunction of attribute tests:

$$Condition_i = (A_1 \ op \ v_1) \wedge (A_2 \ op \ v_2) \wedge \ldots (A_k \ op \ v_k), \qquad (2)$$

where (A_j, v_j) is an attribute-value pair and op is a logical operator chosen from the set $\{=, \neq, <, >, \leq, \geq\}$. Each attribute test $(A_j \ op \ v_j)$ is known as a conjunct. The right-hand side of the rule is called the **rule consequent**, which contains the predicted class y_i.

Application of Rule-Based Classifier

- A rule *r* covers an instance **x** if the attributes of the instance satisfy the condition of the rule

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk □ Bird

The rule R3 covers the grizzly bear □ Mammal

How a Rule-Based Classifier Works

- • Applies a set of IF-THEN rules to assign class labels.
- • Process:
- 1. Match instance against rules in sequence.
- 2. First matching rule assigns the class label.
- 3. If no rule matches, use a default rule or fallback strategy.

How does Rule-based Classifier Work?

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5, Since the classes predicted by the rules are contradictory (reptiles versus amphibians), their conflicting classes must be resolved.

A dogfish shark triggers none of the rules

Rule Ordering Schemes

- 1. **Ordered Rule Set**:
- - Rules are applied in order.
- - First matching rule is used.
- 2. **Unordered Rule Set**:
- - All matching rules are considered.
- - Use voting or confidence scoring to assign class.

Characteristics of Rule Sets

Ordered Rules In this approach, the rules in a rule set are ordered in decreasing order of their priority, which can be defined in many ways (e.g., based on accuracy, coverage, total description length, or the order in which the rules are generated). An ordered rule set is also known as a decision list. When a test record is presented, it is classified by the highest-ranked rule that covers the record. This avoids the problem of having conflicting classes predicted by multiple classification rules.

Unordered Rules This approach allows a test record to trigger multiple classification rules and considers the consequent of each rule as a vote for a particular class. The votes are then tallied to determine the class label of the test record. The record is usually assigned to the class that receives the highest number of votes. In some cases, the vote may be weighted by the rule's accuracy. Using unordered rules to build a rule-based classifier has both advantages and disadvantages. Unordered rules are less susceptible to errors caused by the wrong rule being selected to classify a test record (unlike classifiers based on ordered rules, which are sensitive to the choice of rule ordering criteria). Model building is also less expensive because the rules do not have to be kept in sorted order. Nevertheless, classifying a test record can be quite an expensive task because the attributes of the test record must be compared against the precondition of every rule in the rule set.

Ordered Rule Set : we will focus on rule-based classifiers that use ordered rules.

- Rules are rank ordered according to their priority
  - An ordered rule set is known as a decision list
- When a test record is presented to the classifier
  - It is assigned to the class label of the highest ranked rule it has triggered
  - If none of the rules fired, it is assigned to the default class

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds
R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

Class-Based Ordering Scheme

- In this approach, rules that belong to the same class appear together in the rule set R. The rules are then collectively sorted on the basis of their class information. The relative ordering among the rules from the same class is not important; as long as one of the rules fires, the class will be assigned to the test record. This makes rule interpretation slightly easier. However, it is possible for a high-quality rule to be overlooked in favour of an inferior rule that happens to predict the higher-ranked class.
- Most of the well-known rule-based classifiers (such as C4.5rules and RIPPER) employ the class-based ordering scheme.

Rule Ordering Schemes

Rule-Based Ordering

(Skin Cover=feathers, Aerial Creature=yes) ==> Birds

(Body temperature=warm-blooded, Gives Birth=yes) ==> Mammals

(Body temperature=warm-blooded, Gives Birth=no) ==> Birds

(Aquatic Creature=semi)) ==> Amphibians

(Skin Cover=scales, Aquatic Creature=no) ==> Reptiles

(Skin Cover=scales, Aquatic Creature=yes) ==> Fishes

(Skin Cover=none) ==> Amphibians

Class-Based Ordering

(Skin Cover=feathers, Aerial Creature=yes) ==> Birds

(Body temperature=warm-blooded, Gives Birth=no) ==> Birds

(Body temperature=warm-blooded, Gives Birth=yes) ==> Mammals

(Aquatic Creature=semi)) ==> Amphibians

(Skin Cover=none) ==> Amphibians

(Skin Cover=scales, Aquatic Creature=no) ==> Reptiles

(Skin Cover=scales, Aquatic Creature=yes) ==> Fishes

Figure 1. Comparison between rule-based and class-based ordering schemes.

Building a Rule-Based Classifier

- • Steps:
- 1. Data Preparation and Discretization
- 2. Rule Generation from Training Data
- 3. Rule Pruning to remove noisy/overfitting rules
- 4. Rule Ordering (if ordered system is used)
- 5. Evaluation using a test set

How to Build a Rule-Based Classifier

- To build a rule-based classifier, we need to extract a set of rules that identifies key relationships between the attributes of a data set and the class label. There are two broad classes of methods for extracting classification rules:

(1) direct methods, which extract classification rules directly from data, and

(2) indirect methods, which extract classification rules from other classification models, such as decision trees and neural networks.

- Direct methods partition the attribute space into smaller subspaces so that all the records that belong to a subspace can be classified using a single classification rule.
- Indirect methods use the classification rules to provide a succinct description of more complex classification models.

Rule Extraction Methods

- 1. **Direct Methods**:
- - Extract rules directly from training data.
- - Example: RIPPER, CN2, PART algorithms.
- 2. **Indirect Methods**:
- - Derive rules from existing models (e.g., decision trees or neural nets).
- - Improves model interpretability.

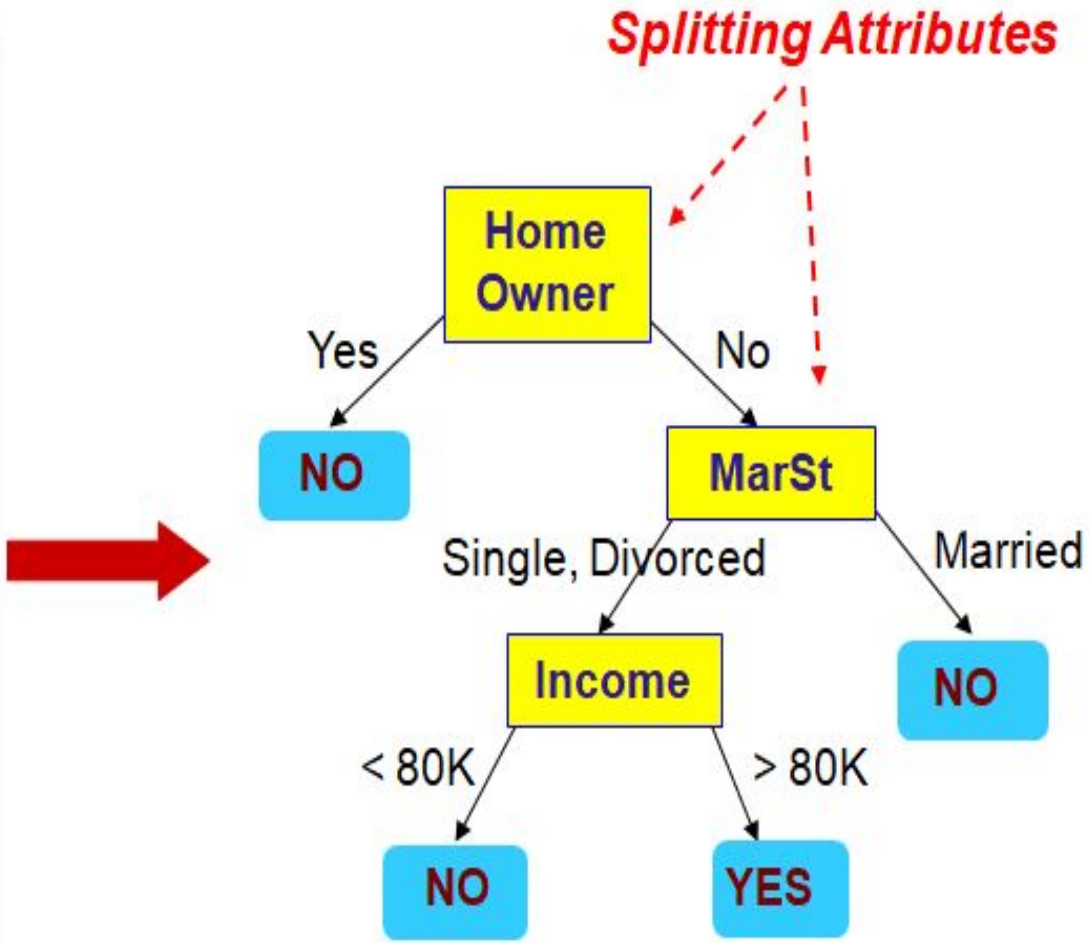
How to Build a Rule-Based Classifier

- Direct Method:
  - Extract rules directly from data
  - Examples: RIPPER, CN2, Holte's 1R
- Indirect Method:
  - Extract rules from other classification models (e.g. decision trees, neural networks, etc).
  - Examples: C4.5 rules

Example of a Decision Tree

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
		categorical	categorical	continuous	class
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

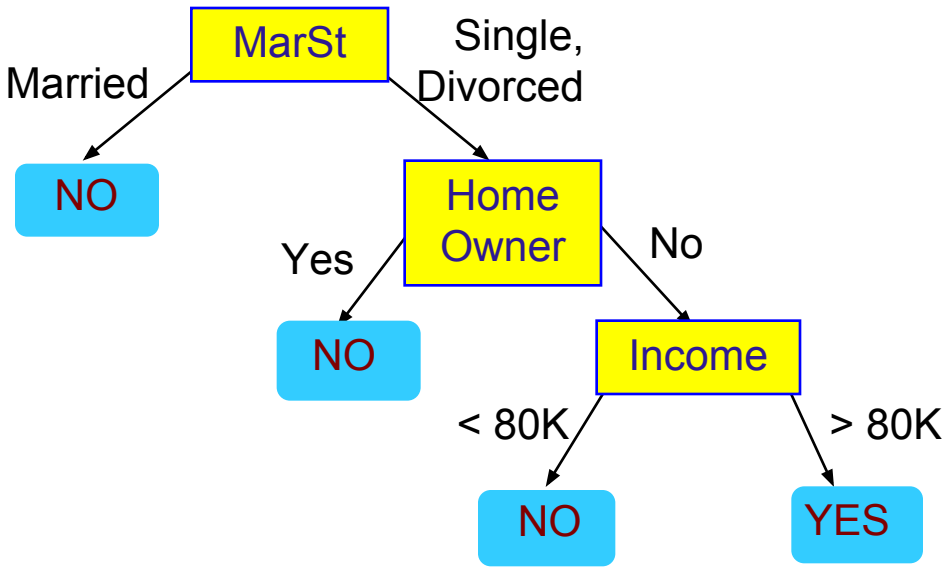
Training Data



Model: Decision Tree

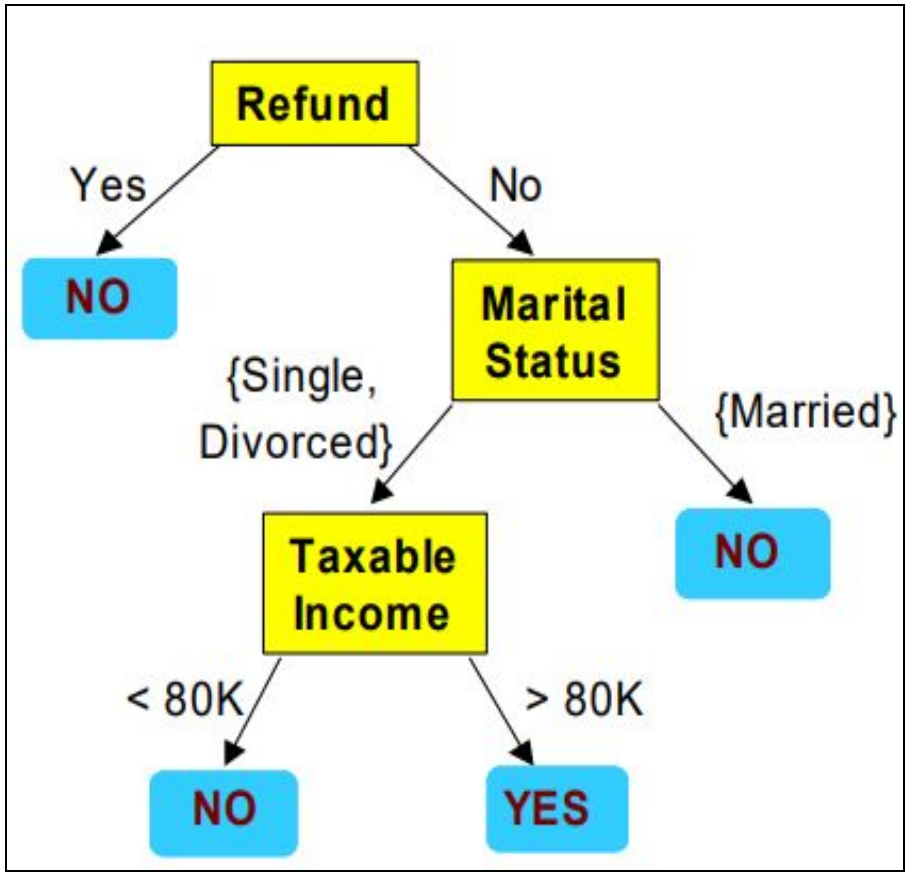
Another Example of Decision Tree

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
	categorical	categorical	continuous	class
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!

From Decision Trees To Rules



Classification Rules

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Rules are mutually exclusive and exhaustive Rule set contains as much information as the tree

Direct Methods for Rule Extraction : Learn-One-Rule Function

- The objective of the Learn-One-Rule function is to extract a classification rule that covers many of the positive examples and none (or very few) of the negative examples in the training set.
- However, finding an optimal rule is computationally expensive given the exponential size of the search space.
- The Learn-One-Rule function addresses the exponential search problem by growing the rules in a greedy fashion.
- It generates an initial rule r and keeps refining the rule until a certain stopping criterion is met.
- The rule is then pruned to improve its generalization error.

1. **Which of the following is the correct format of a rule in rule-based classification?**
   a) WHILE condition DO action
   b) IF condition THEN class
   c) WHEN condition THEN result
   d) SELECT condition FROM class

2. **In an ordered rule-based classifier, how is a rule selected?**
   a) Random selection
   b) First matching rule is fired
   c) All rules are fired
   d) Majority vote among rules

3. **Which technique converts decision tree paths into classification rules?**
   a) Rule pruning
   b) Rule optimization
   c) Indirect rule extraction
   d) Sequential covering

4. **The process of simplifying rules by removing unnecessary conditions is called:**

a) Rule training

b) Rule boosting

c) Rule pruning

d) Rule evaluation

5. **In which scheme do all matching rules vote for a class label?**

a) Ordered rule set

b) Random forest

c) Majority voting

d) Confidence boosting

.What is a rule-based classifier?

Answer:

A rule-based classifier uses a set of IF-THEN rules to assign a class label to an input data instance based on matching conditions.

2. List any two rule-ordering schemes.

Answer:

1. Ordered Rule Set (Rule Priority)
2. Unordered Rule Set (Majority Voting or Weighted Voting)

3. What is rule pruning and why is it used?

Answer:

Rule pruning removes redundant or overly specific conditions from a rule to improve generalization and avoid overfitting.

4. Name one direct and one indirect rule extraction method.

Answer:

Direct: Sequential Covering Algorithm

Indirect: Conversion of decision trees into rules

5. What does the sequential covering algorithm do?

Answer:

It builds one rule at a time to cover a subset of data and removes the covered instances, repeating the process until all data is classified.

1. **Explain how a rule-based classifier works with a suitable example.**

   Answer:

   A rule-based classifier assigns class labels using IF-THEN rules. For example:

   ```
   IF (age > 50 AND smoker = yes) THEN class = high risk
   ```

   When a data instance satisfies this condition, it is classified as "high risk." The classifier checks rules sequentially or based on a voting scheme to determine the most appropriate class label.

2. **Describe different rule-ordering schemes used in rule-based classification.**

   Answer:

   - **Ordered Rule Set:** Rules are applied in a specific sequence. The first matching rule is selected, and others are ignored.
   - **Unordered Rule Set (Majority Voting):** All matching rules vote for a class; the class with the most votes is chosen.
   - **Weighted Voting:** Each rule has a weight or confidence; votes are weighted accordingly, and the class with the highest score is chosen.

3. **Explain the steps to build a rule-based classifier.**

Answer:

- **Step 1: Rule Generation** – Extract rules from data to cover instances of a particular class.
- **Step 2: Rule Pruning** – Simplify rules by removing unnecessary conditions.
- **Step 3: Rule Ordering/Conflict Resolution** – Arrange rules or use voting to resolve conflicts and improve classification accuracy.

4. **Differentiate between direct and indirect methods of rule extraction. Give examples.**

Answer:

- **Direct Methods:** Extract rules directly from the data using techniques like sequential covering. These are straightforward and interpretable.
  Example: IF (temperature > 100) THEN class = 'fever'
- **Indirect Methods:** Convert models like decision trees into rule sets. Each path from the root to a leaf becomes a rule.
  Example: A decision tree node path becomes: IF (outlook = sunny AND humidity = high) THEN class = no.

Summary

- • Rule-based classifiers use IF-THEN logic to classify data.
- • Rule-ordering affects classification results.
- • Built using direct or indirect rule extraction methods.
- • Simple yet powerful, especially in interpretable systems.