



**BITS Pilani**

Pilani | Dubai | Goa | Hyderabad

# S2-20\_DSECFZC415

## Classification and Prediction



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

# Classification

# Rule-based Classification

# Classification Techniques

---

- Decision Tree based Methods
- Rule-based Methods
- Neural Networks
  - computational networks that simulate the decision process in neurons (networks of nerve cell)
- Naïve Bayes and Bayesian Belief Networks
  - uses the probability theory to find the most likely of the possible classifications
- Support Vector Machines
  - fits a boundary to a region of points that are all alike; uses the boundary to classify a new point

# Rule-Based Classifier

- Classify records by using a collection of “if...then...” rules
- Rule:  $(\text{Condition}) \rightarrow y$  where
  - Condition is a conjunctions of attributes
  - $y$  is the class label
  - LHS: rule antecedent or condition
  - RHS: rule consequent
  - Examples of classification rules:
    - $(\text{Blood Type}=\text{Warm}) \wedge (\text{Lay Eggs}=\text{Yes}) \rightarrow \text{Birds}$
    - $(\text{Taxable Income} < 50\text{K}) \wedge (\text{Refund}=\text{Yes}) \rightarrow \text{Evade}=\text{No}$

# 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)  $\rightarrow$  Birds

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

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

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

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians

# 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)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

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

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

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

R5: (Live in Water = sometimes)  $\rightarrow$  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  $\Rightarrow$  Bird

The rule R3 covers the grizzly bear  $\Rightarrow$  Mammal



# Rule Coverage and Accuracy

- Coverage of a rule:
  - Fraction of records that satisfy the antecedent of a rule
- Accuracy of a rule:
  - Fraction of records that satisfy both the antecedent and consequent of a rule

(Status=Single) → No

Coverage = 40%, Accuracy = 50%

<i>Tid</i>	Refund	Marital Status	Taxable Income	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

# How does Rule-based Classifier Work?

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

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

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

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

R5: (Live in Water = sometimes)  $\rightarrow$  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

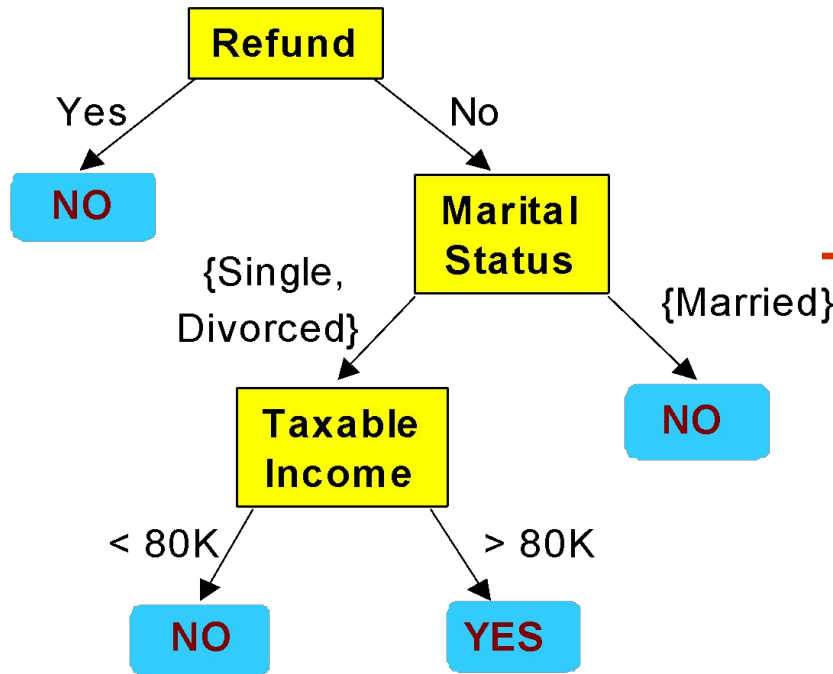
A dogfish shark triggers none of the rules

# Characteristics of Rule-Based Classifier

---

- Mutually exclusive rules
  - Classifier contains mutually exclusive rules if the rules are independent of each other
  - Every record is covered by at most one rule
- Exhaustive rules
  - Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
  - Each record is covered by at least one rule

# From Decision Trees To Rules



## Classification Rules

(Refund=Yes) ==> No

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

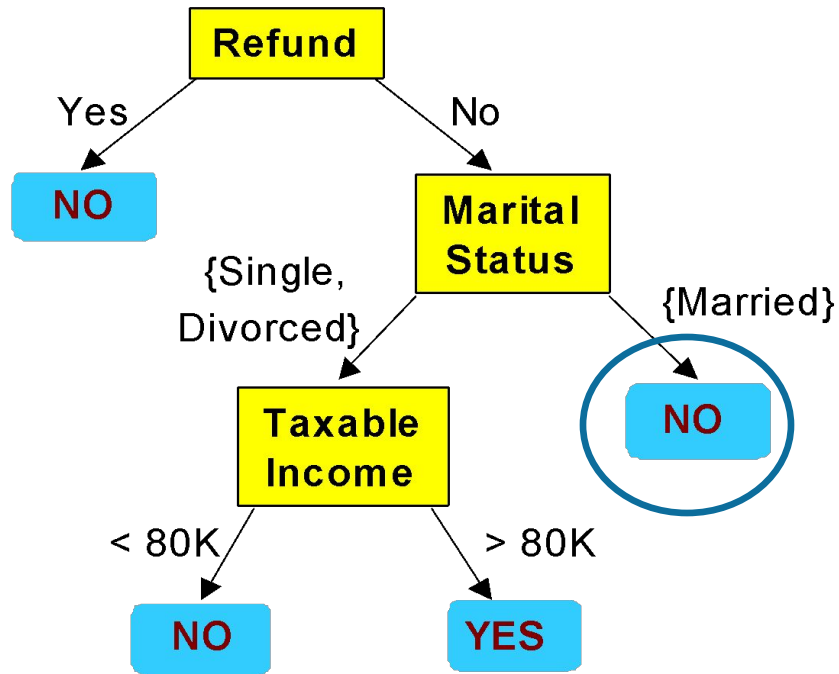
(Refund=No, Marital Status={Single, Divorced}, Taxable Income>80K) ==> Yes

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

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

# Rules Can Be Simplified



<i>Tid</i>	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

Initial Rule:  $(\text{Refund}=\text{No}) \wedge (\text{Status}=\text{Married}) \rightarrow \text{No}$

Simplified Rule:  $(\text{Status}=\text{Married}) \rightarrow \text{No}$

# Further Characterizing Rules

---

- Rules are not mutually exclusive
  - A record may trigger more than one rule
  - Solution?
    - Ordered rule set
    - Unordered rule set – use voting schemes
- Rules are not exhaustive
  - A record may not trigger any rules
  - Solution?
    - Use a default class

# Ordered Rule Set

- 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

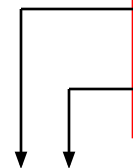
R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

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

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

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

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians



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

# Rule Ordering Schemes

- Rule-based ordering
  - Individual rules are ranked based on their quality
- Class-based ordering
  - Rules that belong to the same class appear together

## Rule-based Ordering

(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

## Class-based Ordering

(Refund=Yes) ==> No

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

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

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



# How to Evaluate Learnt Rule?

- Start with the most general rule possible: condition = empty
- Adding new attributes by adopting a greedy depth-first strategy
  - Picks the one that most improves the rule quality
- Rule-Quality measures: consider both coverage and accuracy
  - Foil-gain (in FOIL & RIPPER): assesses info\_gain by extending condition
    - favors rules that have high accuracy and cover many positive tuples

$$FOIL\_Gain = pos' \times (\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg})$$

- Rule pruning based on an independent set of test tuples
  - Pos/neg are # of positive/negative tuples covered by R.
  - If FOIL\_Prune is higher for the pruned version of R, prune R

$$FOIL\_Prune(R) = \frac{pos - neg}{pos + neg}$$

# How to Evaluate Learnt Rule?

- We can use Likelihood Ratio Statistic, which confirms that effect of rule is not attributed to chance, but represents correlation between attribute value and classes.

$$\text{Likelihood\_Ratio} = 2 * \sum_{j=1}^m f_i \log_2\left(\frac{f_i}{e_i}\right)$$

- m is the number of classes
- For tuples satisfying the rule,  $f_i$  is the observed frequency of each class among tuples,  $e_i$  is the expected frequency if the rule made random predictions
- Higher the Likelihood Ratio, better the rule is.
- Used by CN2

# Building Classification Rules

---

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

# Direct Method: Sequential Covering

---

- Sequential covering algorithm: Extracts rules directly from training data
- Typical sequential covering algorithms: FOIL, AQ, CN2, RIPPER
- Rules are learned *sequentially*, each for a given class  $C_i$  will cover many tuples of  $C_i$  but none (or few) of the tuples of other classes

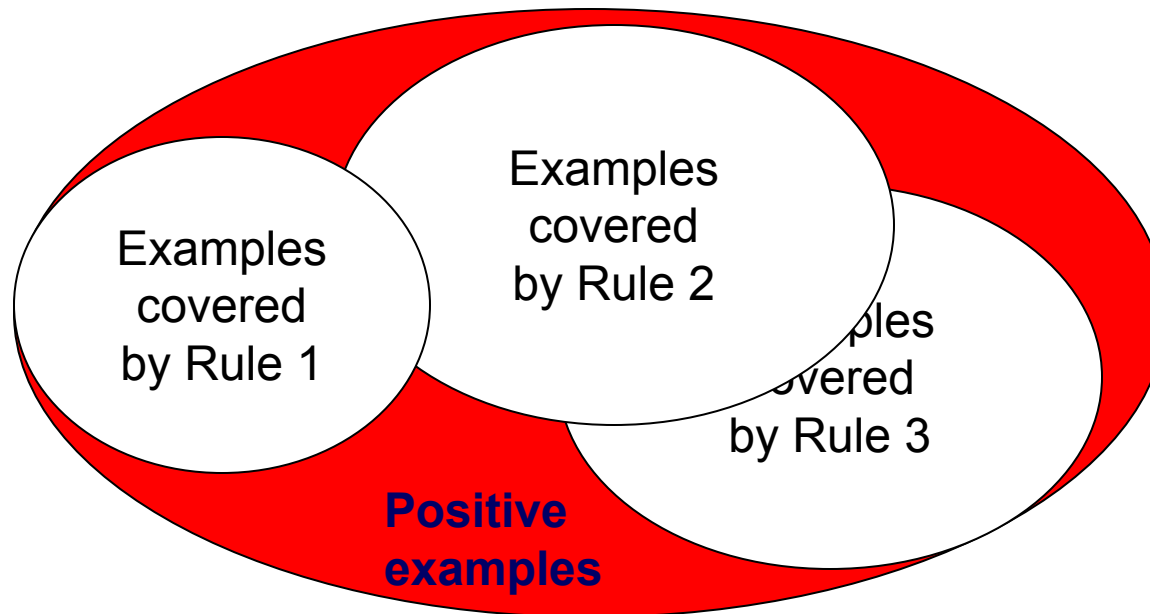
# Rule Induction: Sequential Covering Method

---

- Start with an empty rule set
- Steps:
  - Rules are learned one at a time
  - Each time a rule is learned, the tuples covered by the rules are removed
  - Repeat the process(above steps) on the remaining tuples
    - until *termination condition*, e.g., when no more training examples or when the quality of a rule returned is below a user-specified threshold
- Comparison with decision-tree induction: learning a set of rules *simultaneously*

# Sequential Covering Algorithm

**while** (enough target tuples left)  
generate a rule  
remove positive target tuples satisfying this rule



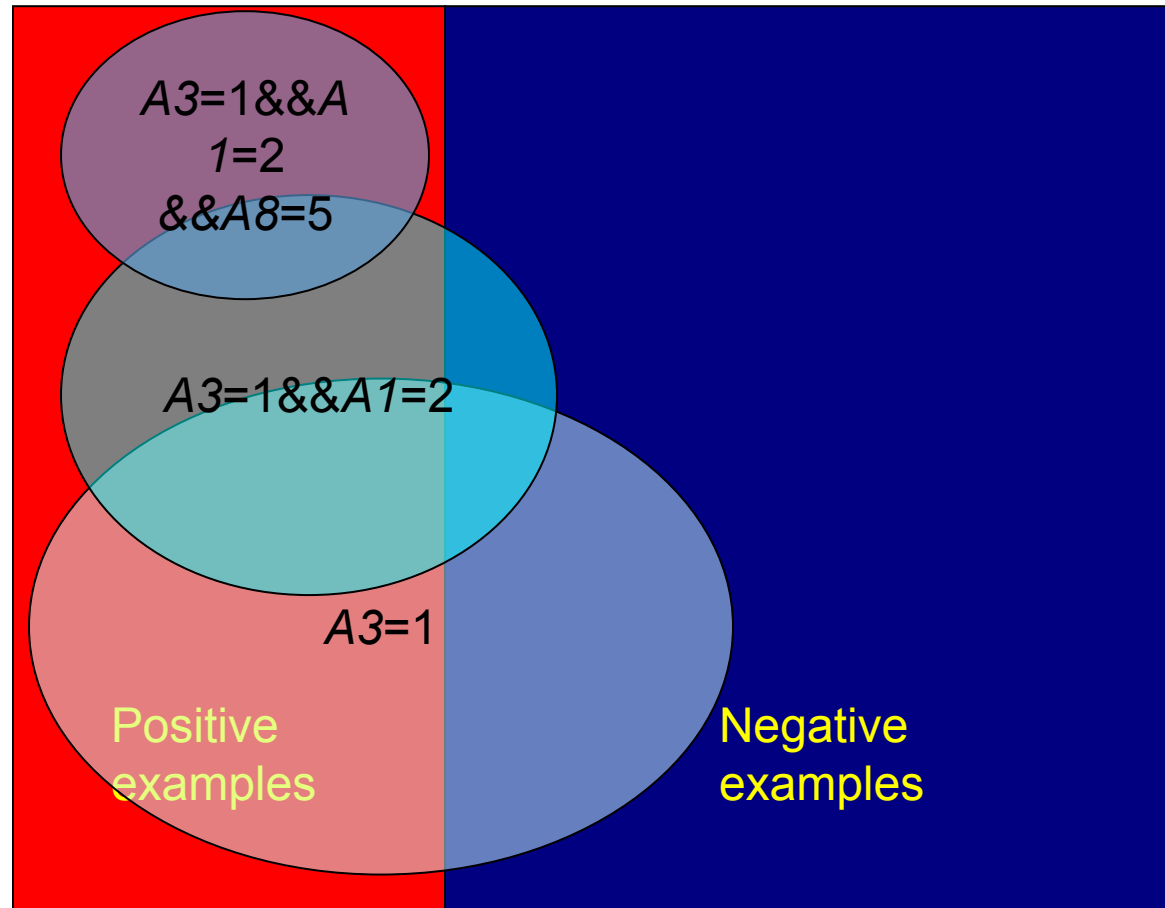
# Rule Generation

---

- To generate a rule
  - while**(true)
    - find the best predicate  $p$
    - if** foil-gain( $p$ ) > threshold **then** add  $p$  to current rule
    - else** break

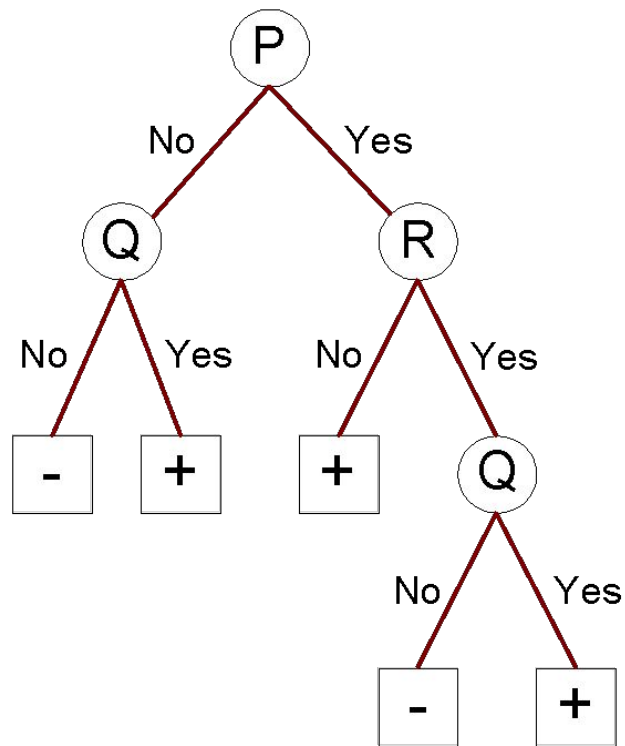
Predicates considered may be independent of each other (as in previous slide) or progressively restrictive (as in the next slide)

# Rule Generation





# Indirect Methods



## Rule Set

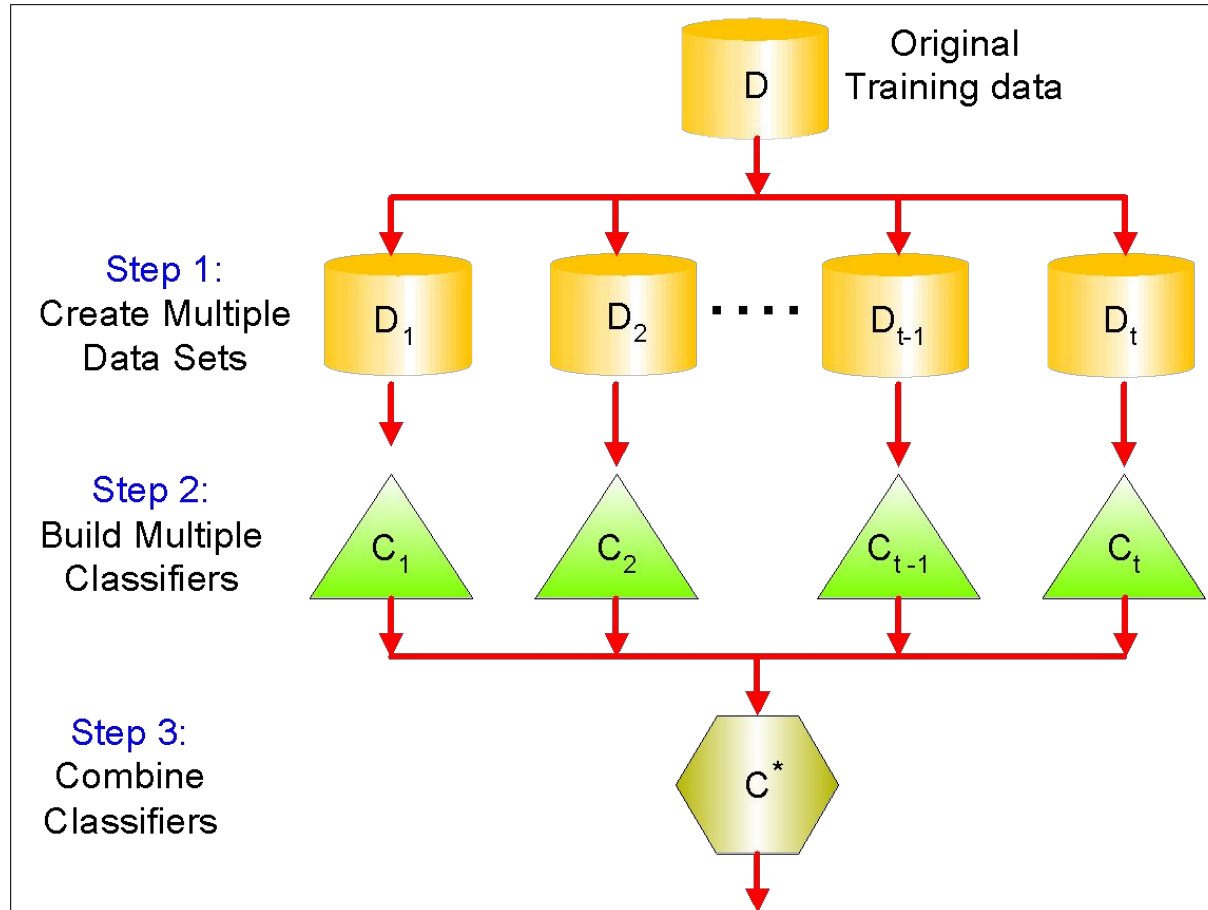
r1: (P=No,Q=No) ==> -  
 r2: (P=No,Q=Yes) ==> +  
 r3: (P=Yes,R=No) ==> +  
 r4: (P=Yes,R=Yes,Q=No) ==> -  
 r5: (P=Yes,R=Yes,Q=Yes) ==> +

# Ensemble Methods

---

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers

# General Idea



# Why does it work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate,  $\varepsilon = 0.35$
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1 - \varepsilon)^{25-i} = 0.06$$

# When error rate differs...

- Suppose there are  $k$  base classifiers
  - Each classifier has different error rate,  $\epsilon_i$
  - Again, assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:
    - Majority of classifiers have to make wrong prediction
    - Compute the probability for each combination that can make wrong prediction (brute force method)
    - Sum up for all possible combinations

## Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers

---

# Thank You