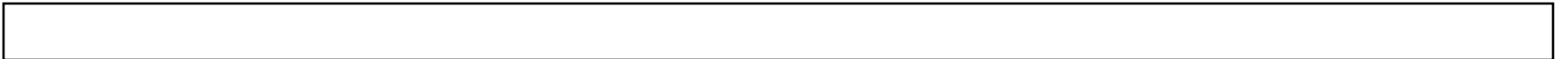# Classification problem

Python for AI

Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.

- Find a *model* for class attribute as a function of the values of other attributes.

- Goal: <u>previously unseen</u> records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.
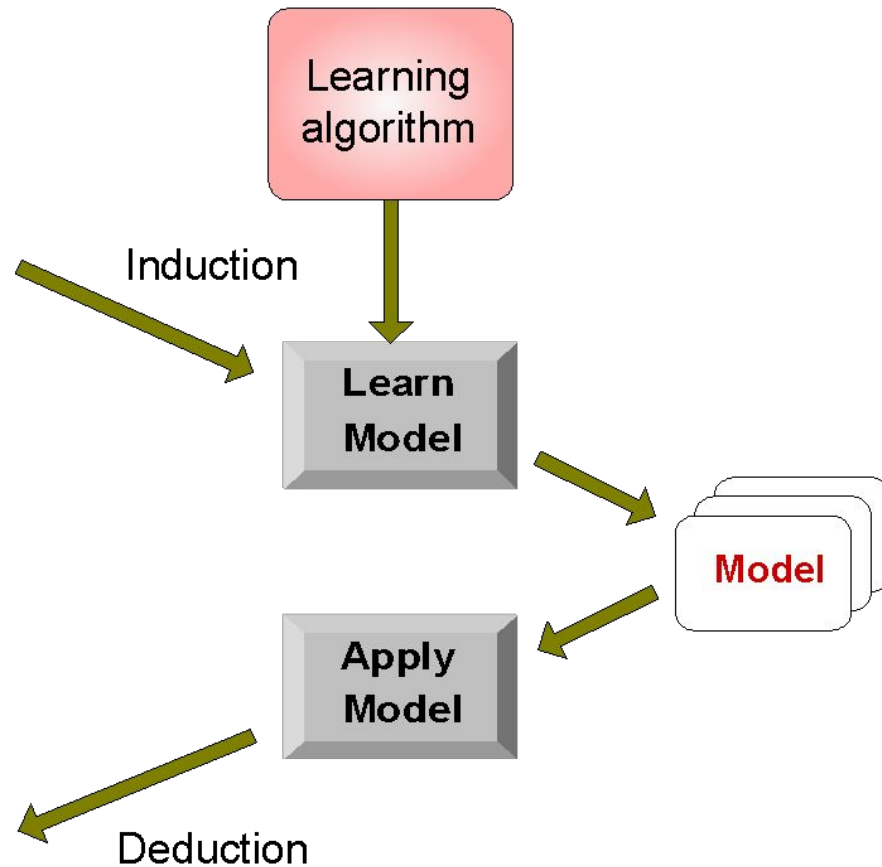
Illustrating Classification Task

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

Learning algorithm

Induction

Learn Model

Model

Apply Model

Deduction

| 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

Examples of Classification Task

- Predicting tumor cells as benign or malignant

- Classifying credit card transactions as legitimate or fraudulent

- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil

- Categorizing news stories as finance, weather, entertainment, sports, etc
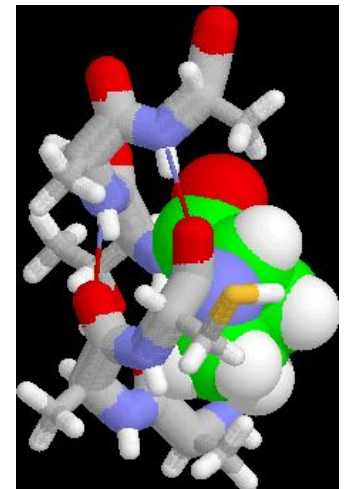
Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
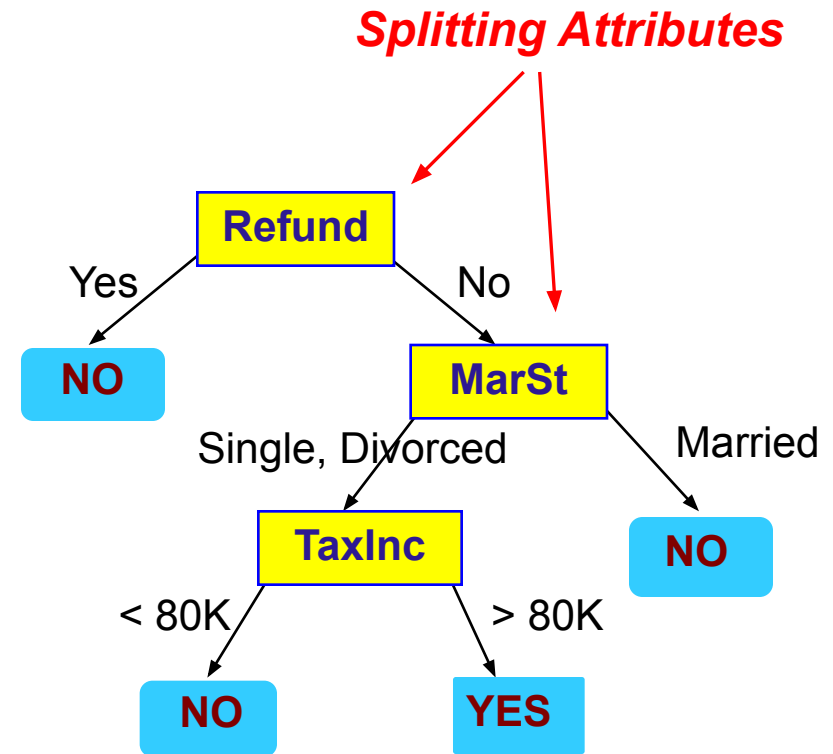- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines
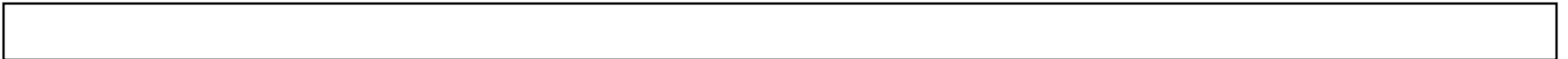
Example of a Decision Tree

	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

Splitting Attributes

Refund
- Yes → NO
- No → **MarSt**
  - Single, Divorced → **TaxInc**
    - < 80K → NO
    - > 80K → YES
  - Married → NO

Model: Decision Tree

Another Example of Decision Tree

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

categorical categorical continuous class

MarSt

Married → NO

Single, Divorced → Refund

Refund:
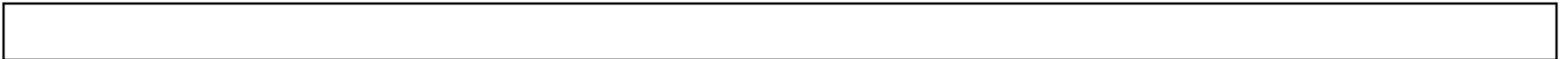Yes → NO
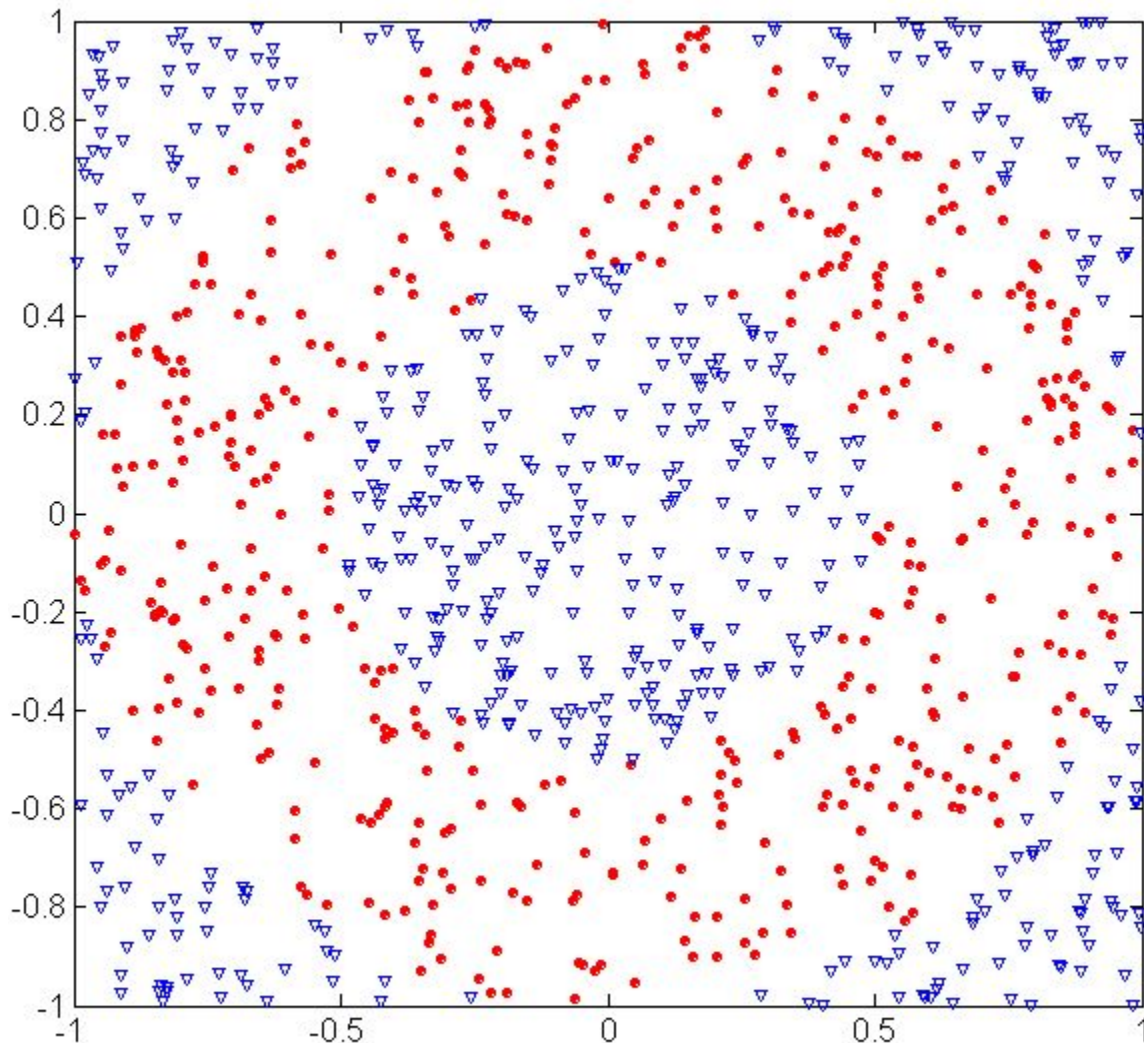No → TaxInc

TaxInc:
< 80K → NO
> 80K → YES
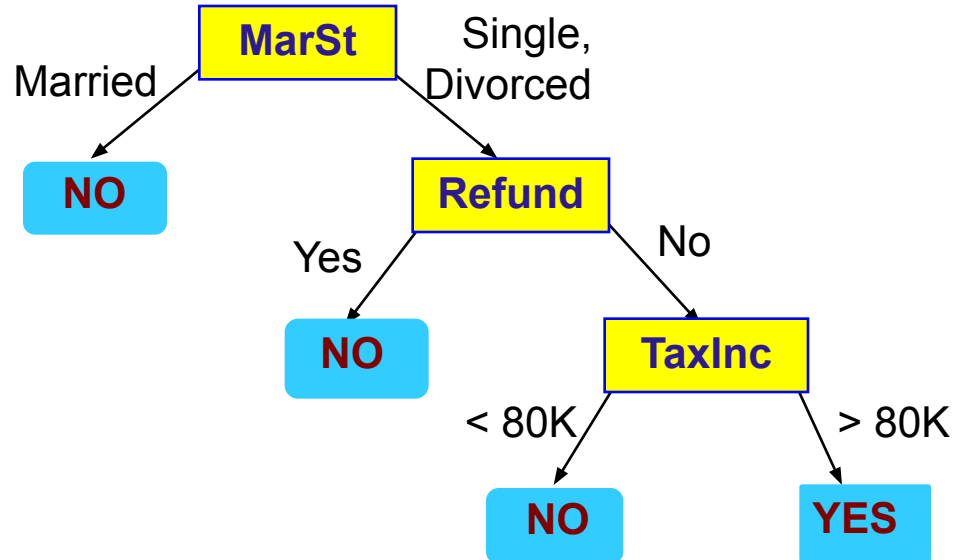
There could be more than one tree that fits the same data!

Practical Issues of Classification

- Underfitting and Overfitting

- Missing Values

- Costs of Classification

Underfitting and Overfitting (Example)

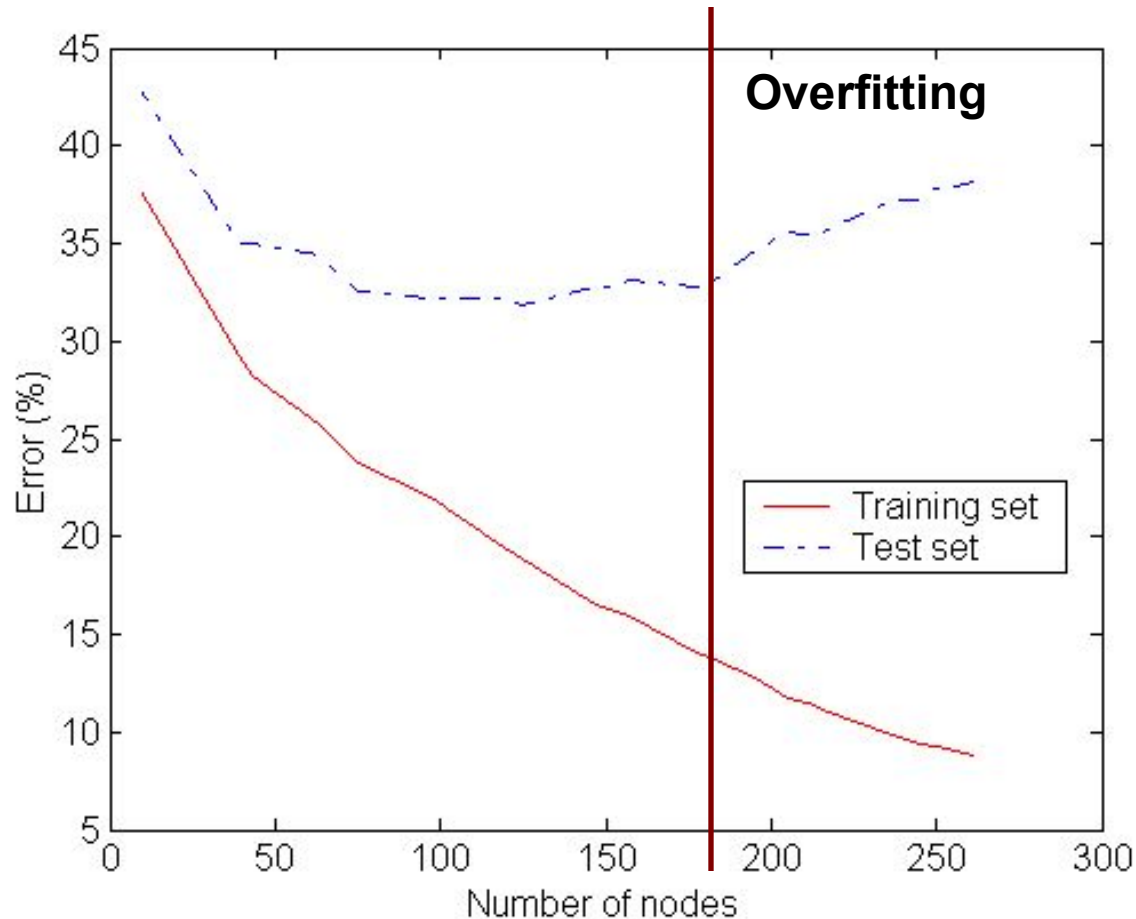
500 circular and 500 triangular data points.

Circular points:
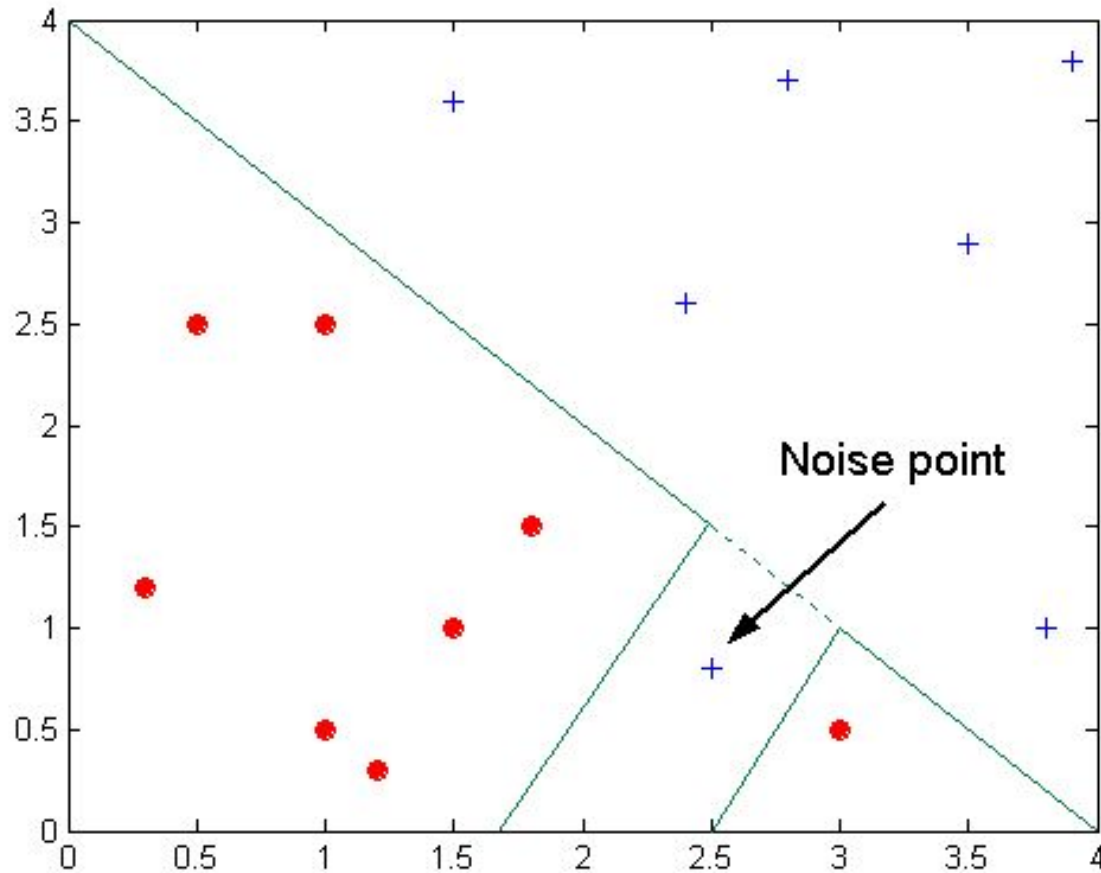
$0.5 \leq \sqrt{x_1^2 + x_2^2} \leq 1$

Triangular points:

$\sqrt{x_1^2 + x_2^2} > 0.5$ or

$\sqrt{x_1^2 + x_2^2} < 1$

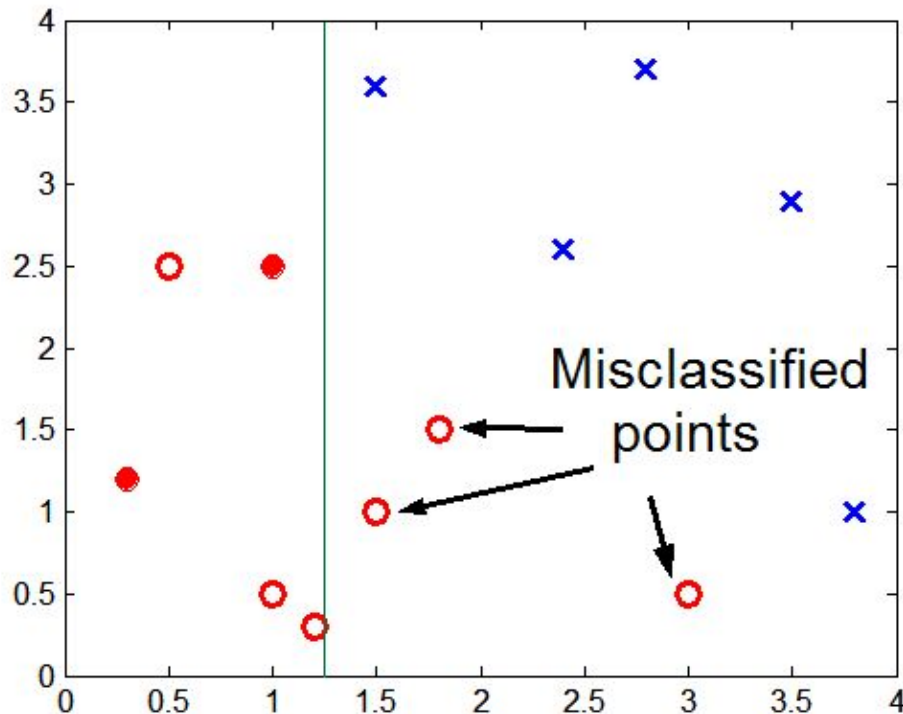# Underfitting and Overfitting

Overfitting

Underfitting: when model is too simple, both training and test errors are large

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

Decision Boundary
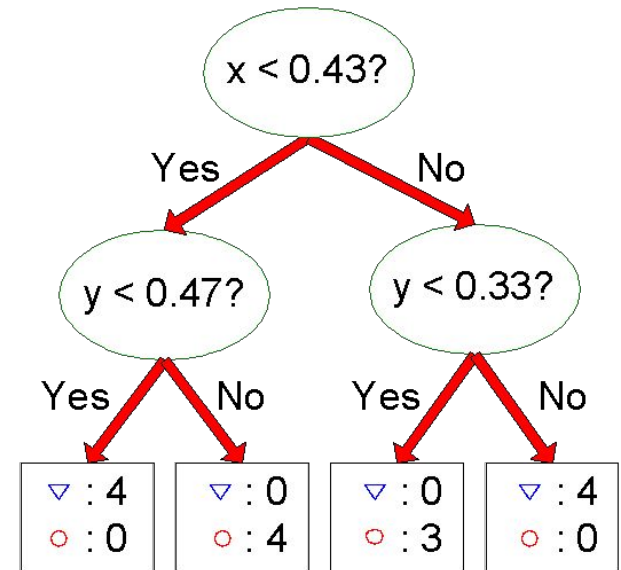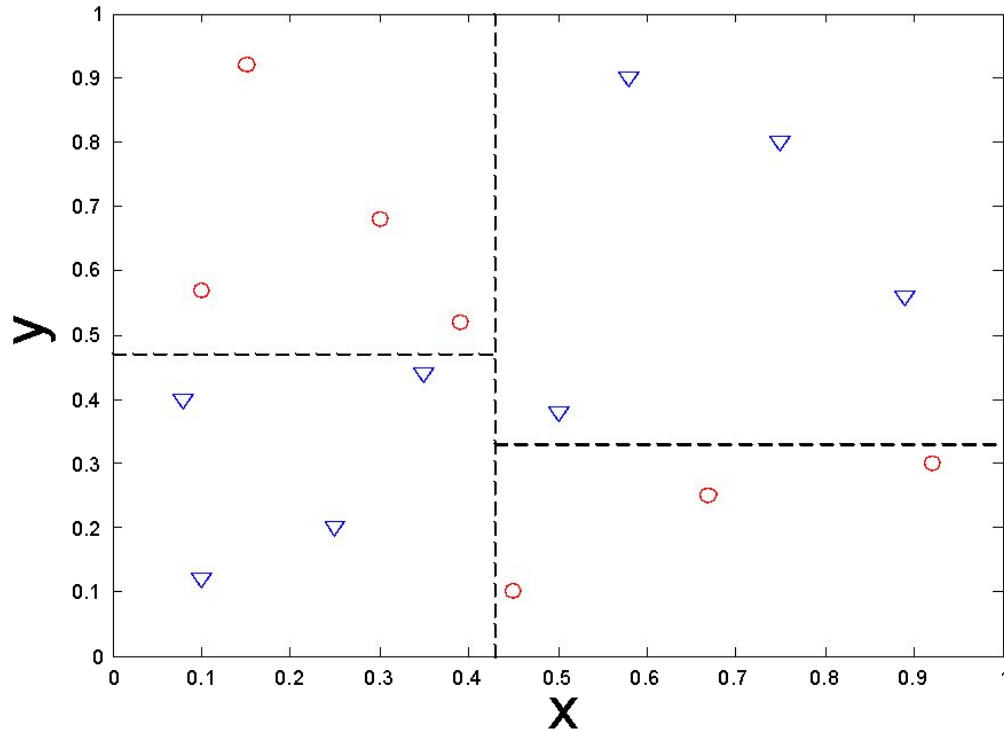
- **Border line between two neighboring regions of different classes is known as decision boundary**

- **Decision boundary is parallel to axes because test condition involves a single attribute at-a-time**

Oblique Decision Trees
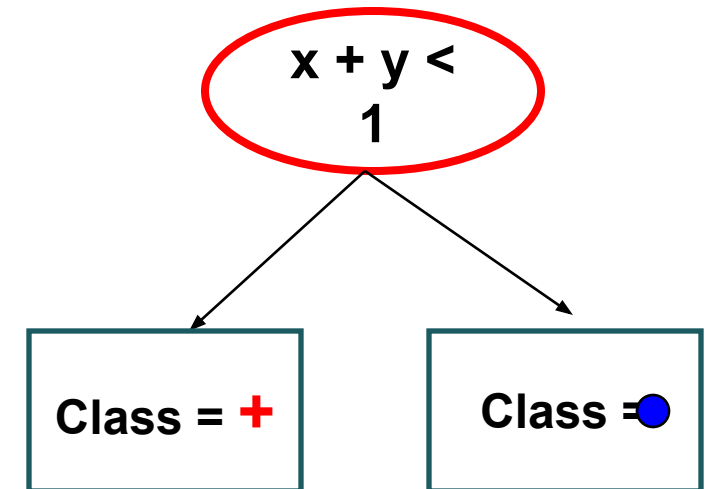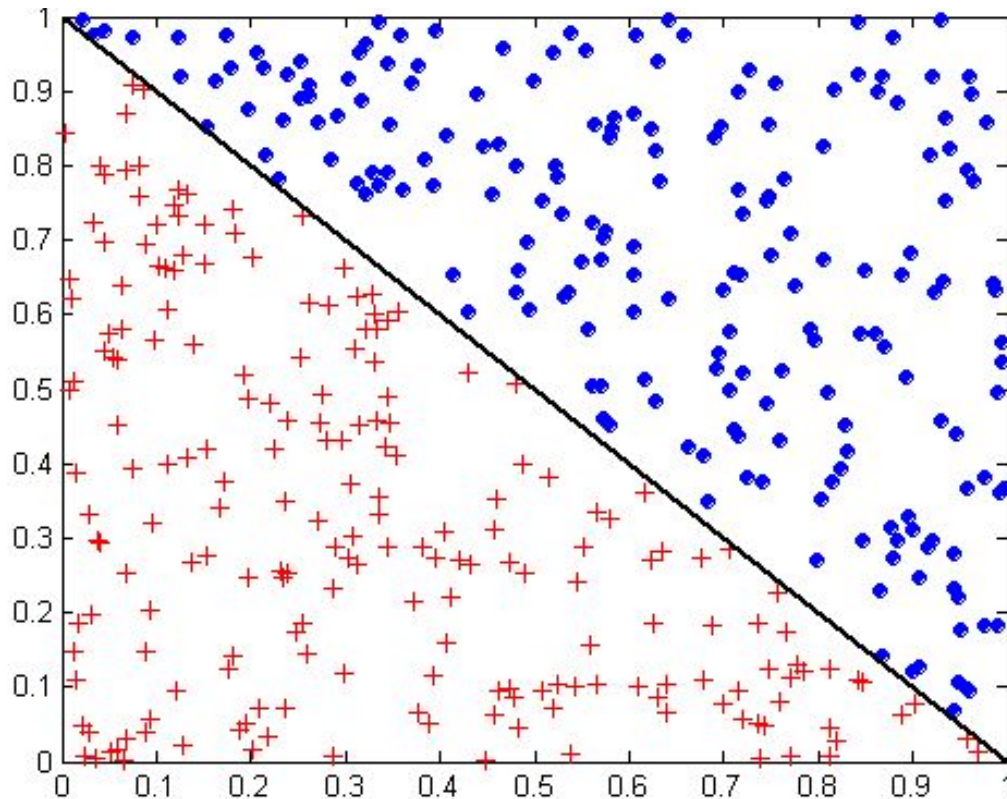
- **Test condition may involve multiple attributes**

- **More expressive representation**

- **Finding optimal test condition is computationally expensive**

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.

- Confusion Matrix:

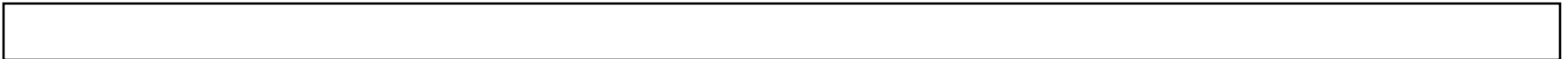|	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation…

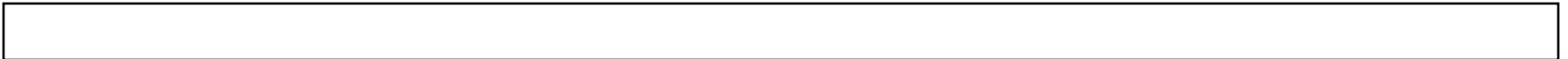|		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Đô chính xác} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Cost Matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j): Cost of misclassifying class j example as class i
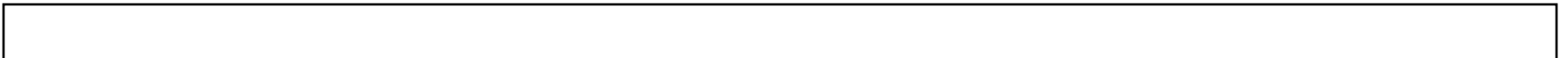
Computing Cost of Classification

Cost Matrix	PREDICTED CLASS		
	C(i|j)	**+**	**-**
ACTUAL CLASS	**+**	-1	100
	-	1	0

Model M$_1$	PREDICTED CLASS		
		+	**-**
ACTUAL CLASS	**+**	150	40
	-	60	250

Model M$_2$	PREDICTED CLASS		
		+	**-**
ACTUAL CLASS	**+**	250	45
	-	5	200

Accuracy = 80%

Cost = 3910

Accuracy = 90%

Cost = 4255

Cost vs Accuracy

Count	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	a	b
Class=No	c	d

Accuracy is proportional to cost if
1. $C(Yes|No)=C(No|Yes) = q$
2. $C(Yes|Yes)=C(No|No) = p$

$$N = a + b + c + d$$

$$Accuracy = (a + d)/N$$

Cost	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	p	q
Class=No	q	p

$$\begin{aligned} Cost &= p\,(a + d) + q\,(b + c) \\ &= p\,(a + d) + q\,(N - a - d) \\ &= q\,N - (q - p)(a + d) \\ &= N\,[q - (q-p) \times Accuracy] \end{aligned}$$

Cost-Sensitive Measures

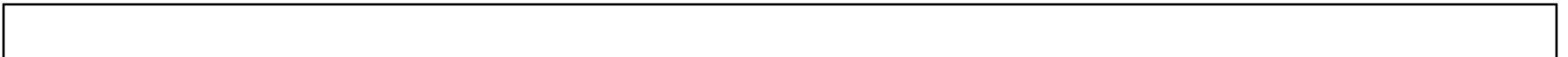$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Example and code

- Download code in the classroom
- On class: follow a step by step tutorial