



Classification

Rahmad Mahendra, M.Sc

with credit to
Samuel Louvan, M.Sc
Meganingrum Arista Jiwanggi, M.Kom

Pusilkom UI

Data Mining for Big Data

16-20 Juli 2018







Classification



A mapping f from input data x (drawn from instance space X) to a label y from some enumerable output space Y

X = set of all fruits

$Y = \{\text{orange, apple, banana}\}$

$$f \left[\begin{array}{c} \text{banana} \\ x \end{array} \right] = \begin{array}{c} \text{banana} \\ y \end{array}$$



As humans how do you
discriminate between each of
the instance?

x



f

```
if x.color == "orange":  
    y = "orange"  
elif x.color == "red":  
    y = "apple"  
elif x.color == "yellow":  
    y = "banana"
```



y

orange



Recognizing a Classification Problem

- Can you formulate your question as a choice among some universe of possible classes?
- Can you create (or find) labeled data that marks that choice for a bunch of examples? Can you make that choice?
- Can you create features that might help in distinguishing those classes?

Problems ?

x



f

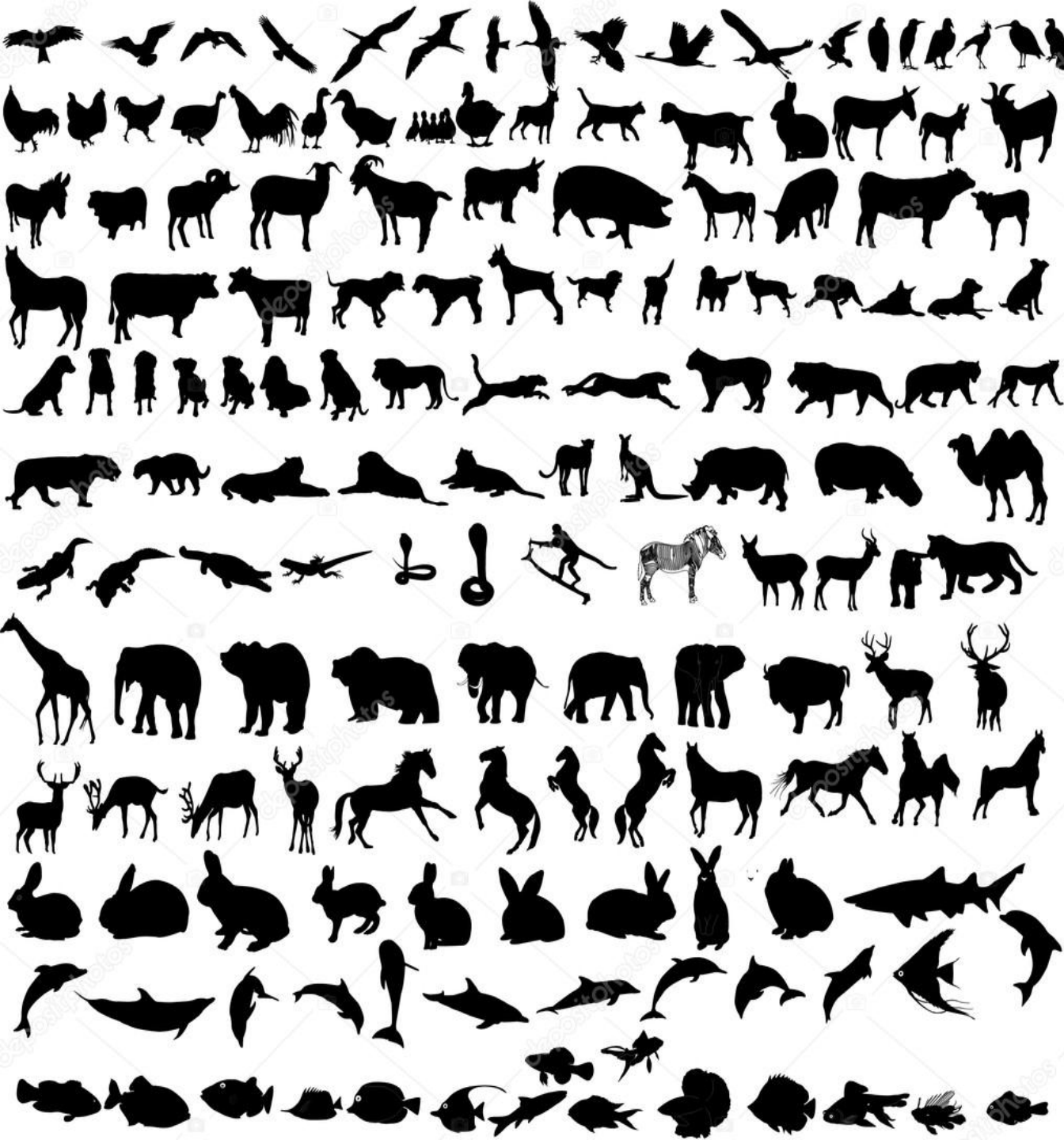
```
if x.color == "orange":  
    y = "orange"  
elif x.color == "red":  
    y = "apple"  
elif x.color == "yellow":  
    y = "banana"
```



?



Problems ?

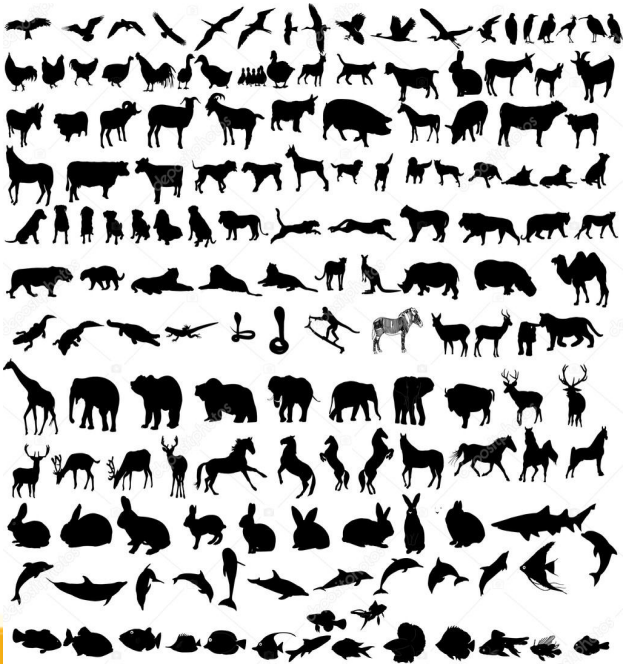


Problems ?

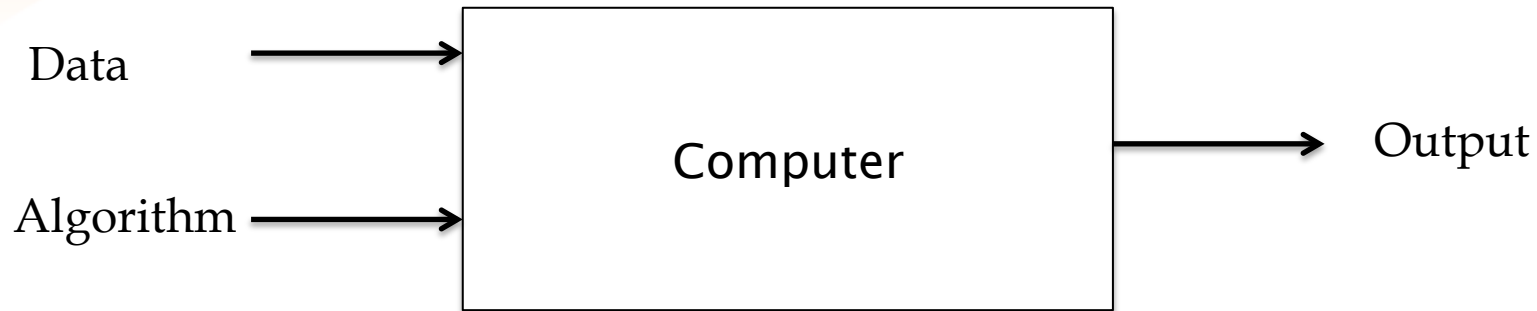


Manually creating rules
are not scalable

Instead, can we let
computers to *learn* the
rules automatically from
data?



Traditional Programming

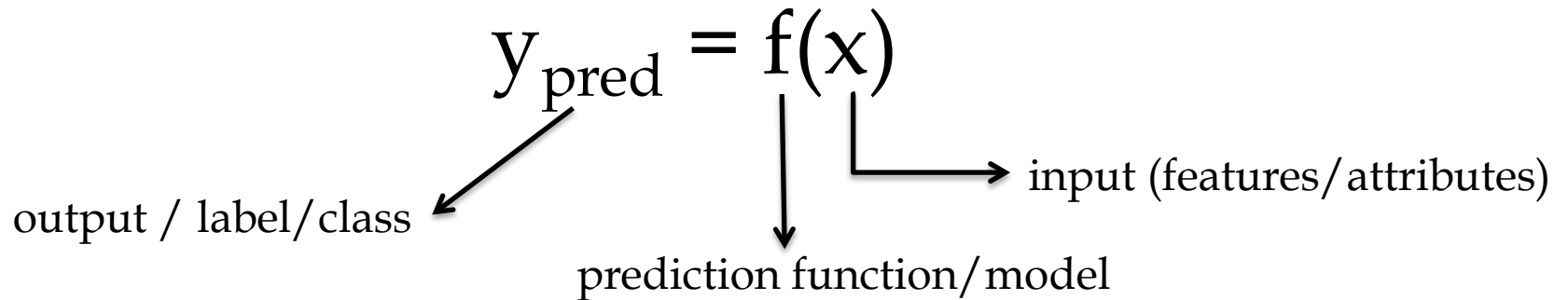


Machine Learning (ML)



[Pedros Domingos]

ML Framework



To learn the function f , you need to *train* it






Apply f to a *never before seen test* example x and output the predicted value $y_{\text{pred}} = f(x)$

ML Framework - Training (Supervised Learning)

Training data

x

y

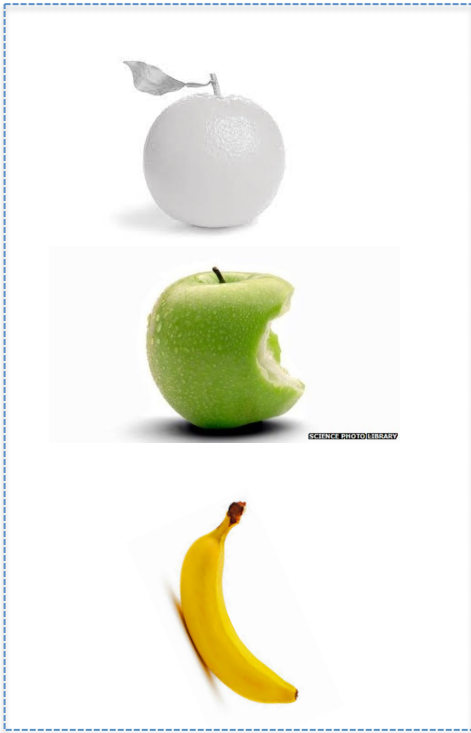
x_1		[color = ... , shape = ..., texture = ...]	orange	y_1
x_2		[color = ... , shape = ..., texture = ...]	banana	y_2
x_3		[color = ... , shape = ..., texture = ...]	apple	y_3
x_4		[color = ... , shape = ..., texture = ...]	banana	y_4
x_5		[color = ... , shape = ..., texture = ...]	apple	y_5



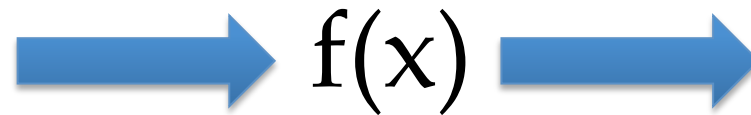
feature vector representation

ML Framework - Testing

Unseen data



x



$f(x)$



y_{pred}

Learning Algorithms

Decision Tree

k-NN

Naïve Bayes

Support Vector
Machine

Logistic
Regression

Neural
Network

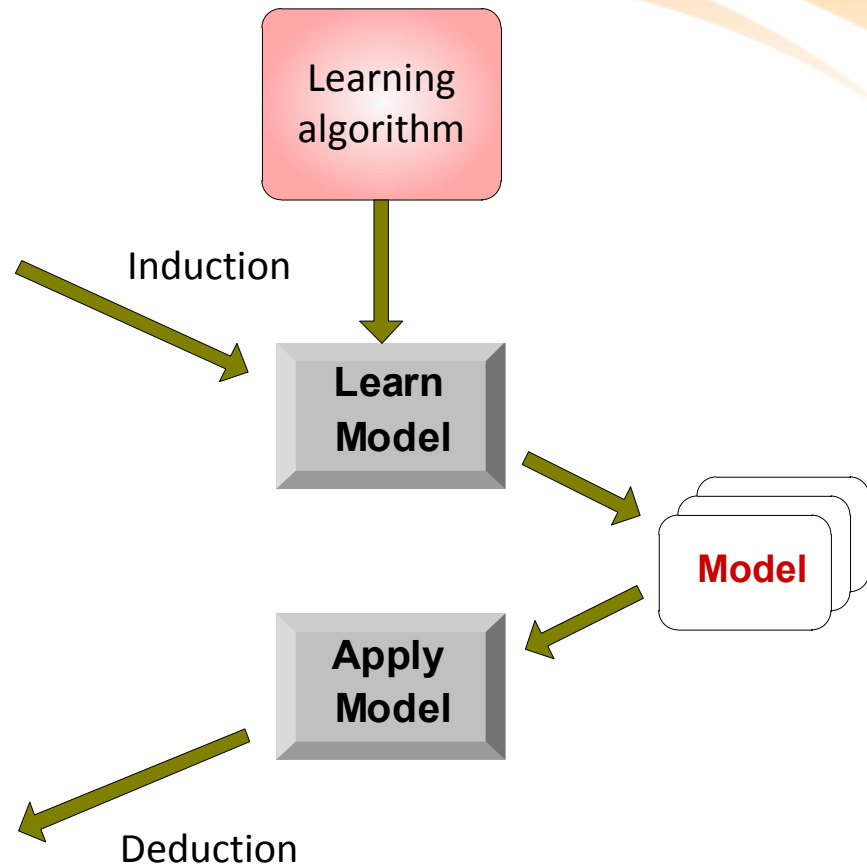
.....

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



k - NN

Instance-Based Classifiers

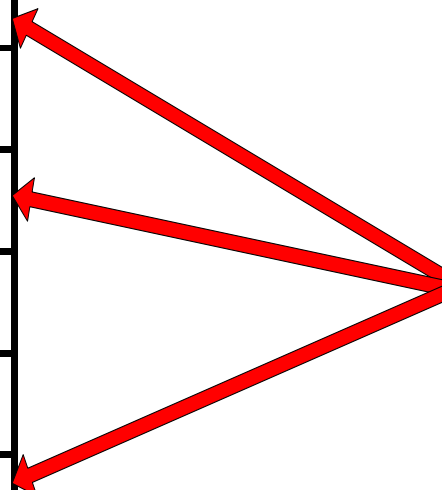
Set of Stored Cases

Atr1	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

- Store the training records
- Use training records to predict the class label of unseen cases

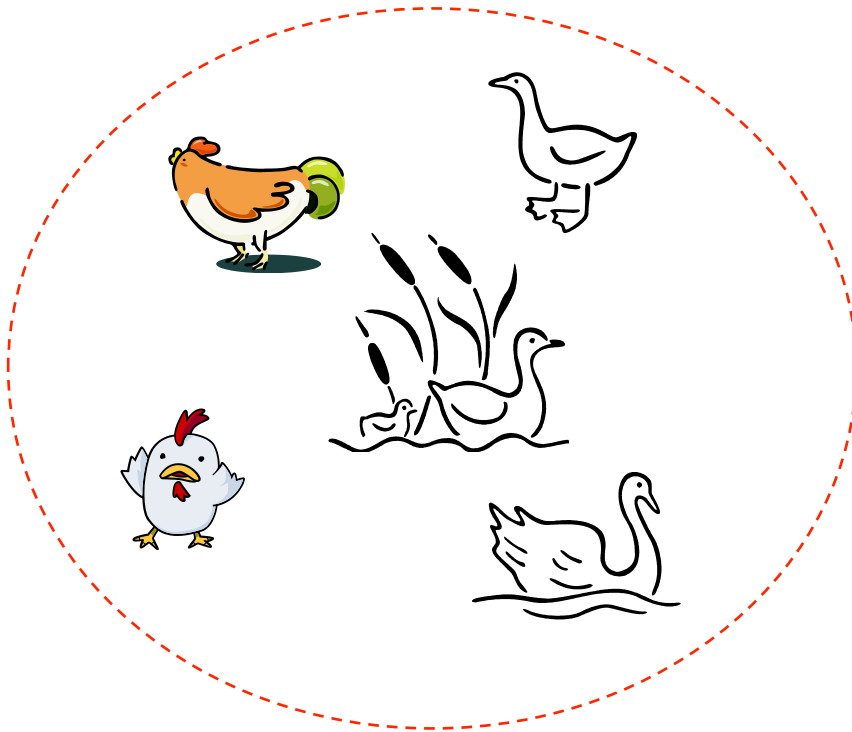
Unseen Case

Atr1	AtrN

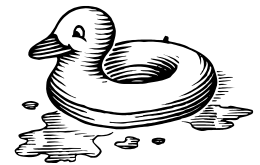


Basic idea:

“If it walks like a duck, quacks like a duck, then it’s probably a duck”

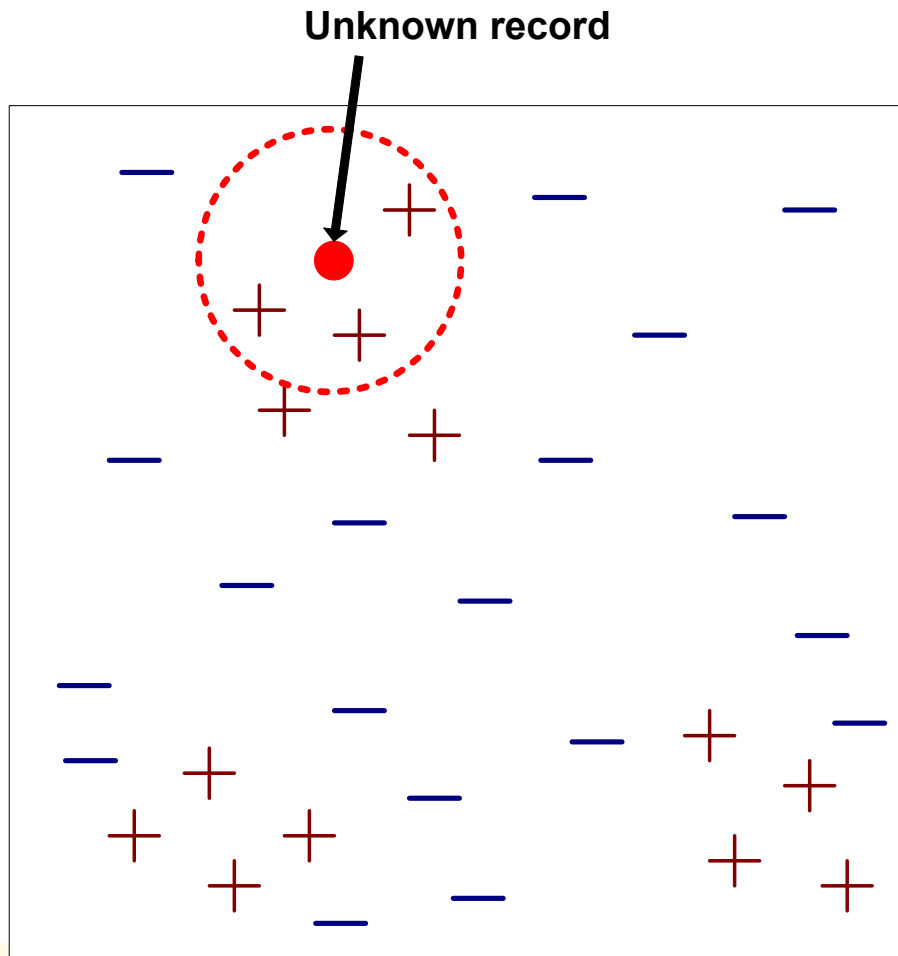


Training
Data



Test Data

Nearest-Neighbor Classifiers



- Requires three things
 - The set of **stored records**
 - **Distance Metric** to compute distance between records
 - The value of k , the **number of nearest neighbors** to retrieve
- To classify an unknown record:
 1. **Compute distance** to other training records
 2. Identify k **nearest neighbors**
 3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking **majority vote**)

Nearest Neighbor Classification

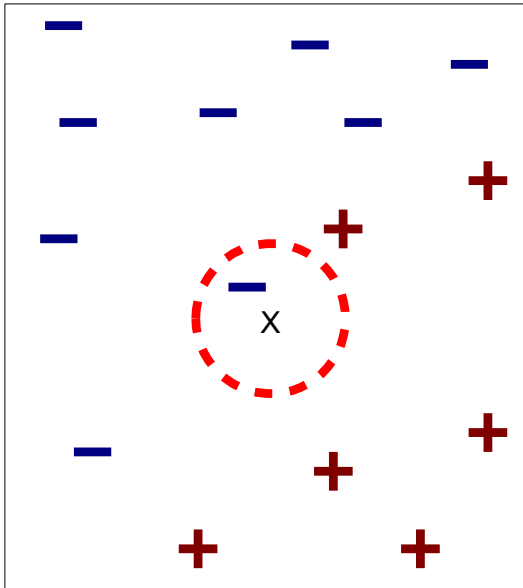
- Compute distance between points:

- Euclidean distance

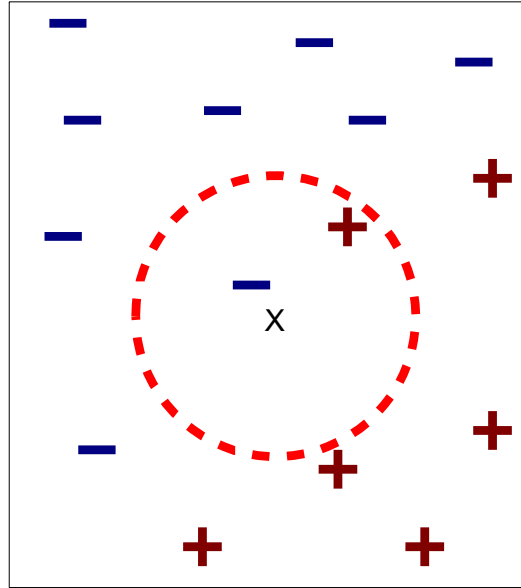
$$d(p, q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from the nearest neighbor list
 - Take the majority class labels among the k-nearest neighbors
 - Weight the vote according to the distance

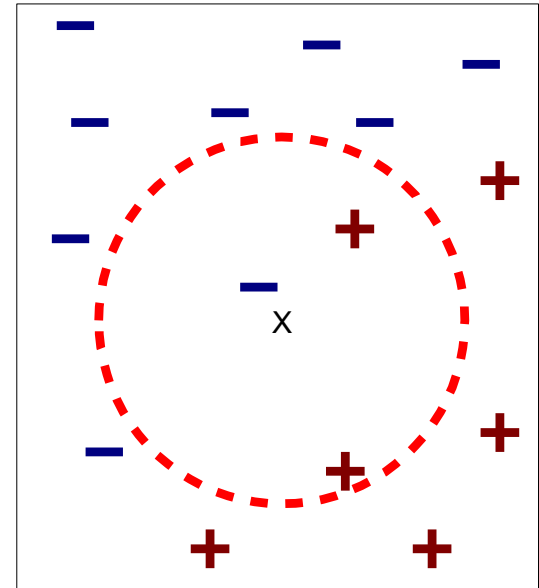
Definition of Nearest Neighbor



(a) 1-nearest neighbor



(b) 2-nearest neighbor

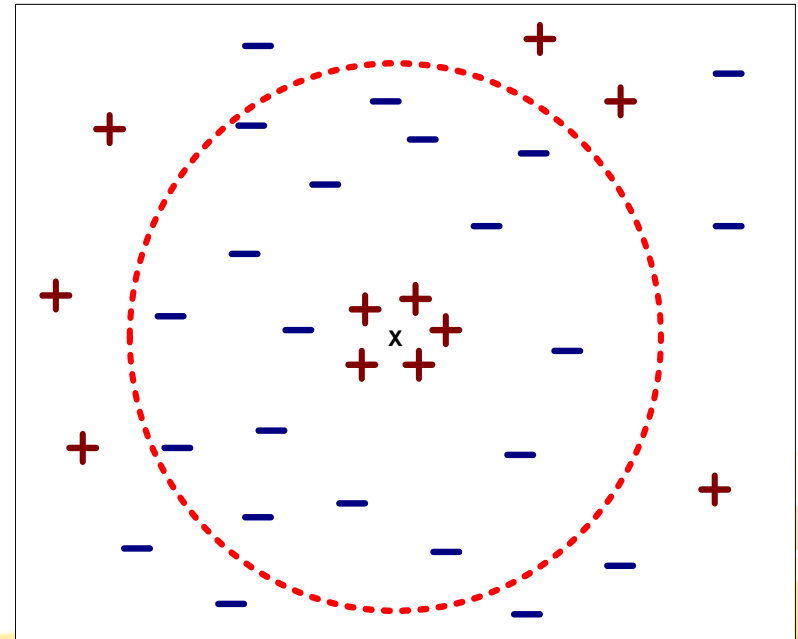


(c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Nearest Neighbor Classification...

- Choosing the value of k :
 - If k is too small, sensitive to noise points
 - If k is too large, neighborhood may include points from other classes

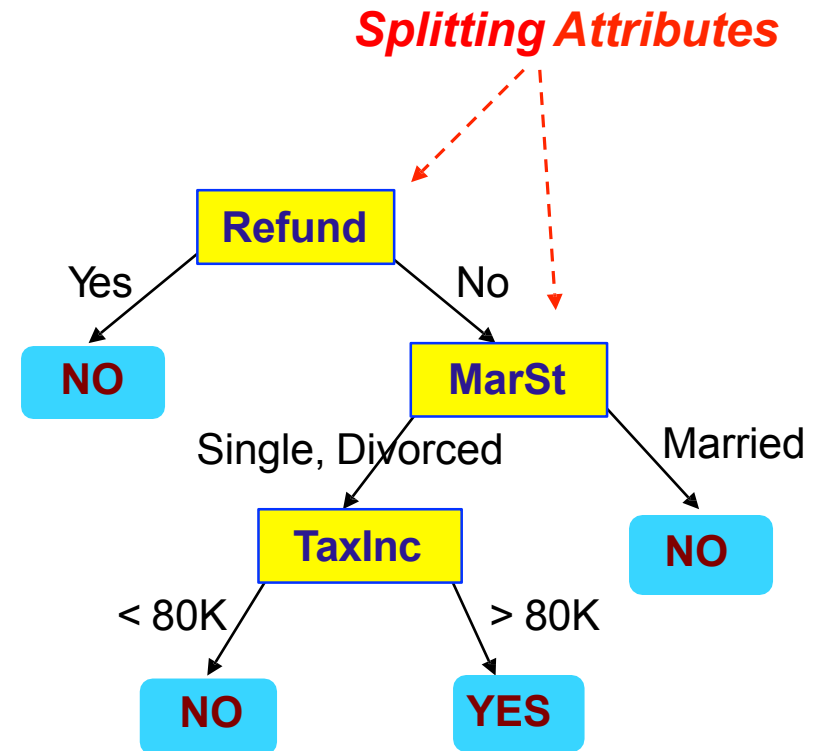


Decision Tree

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



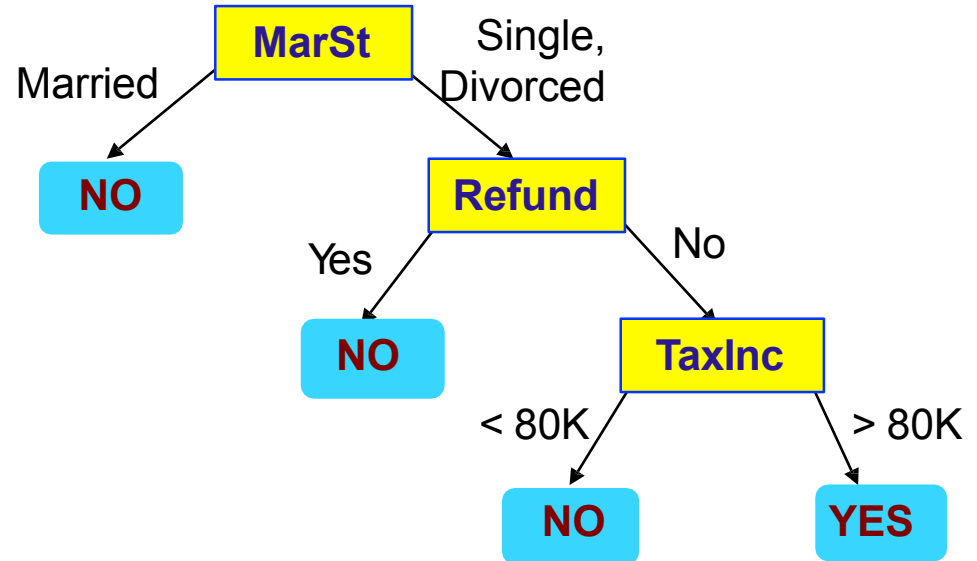
Training Data

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!

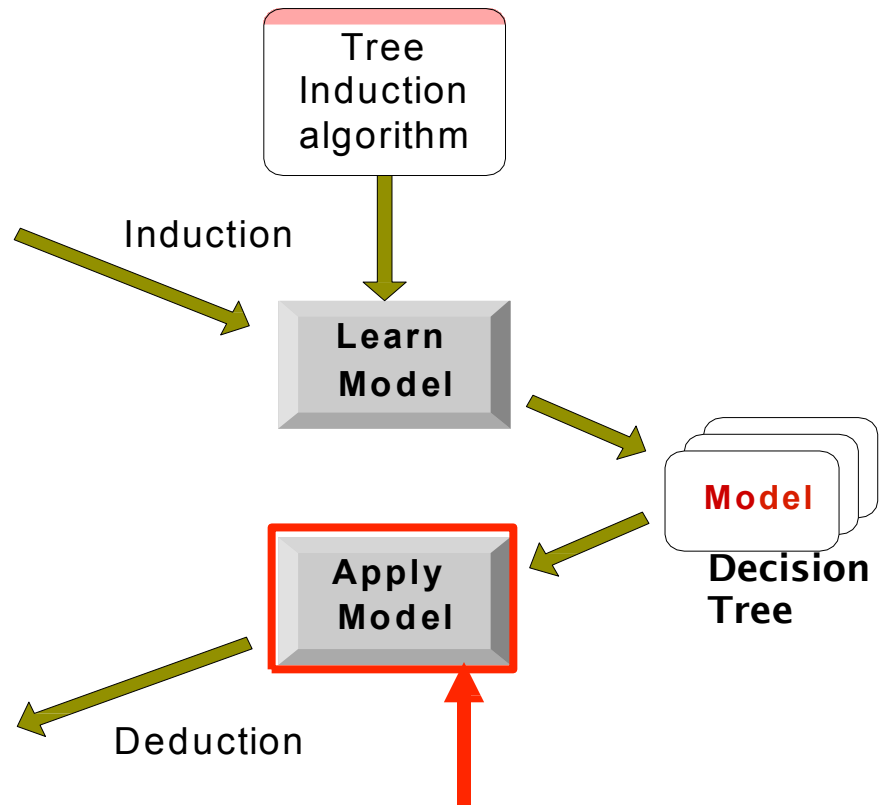
Decision Tree Classification Task

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

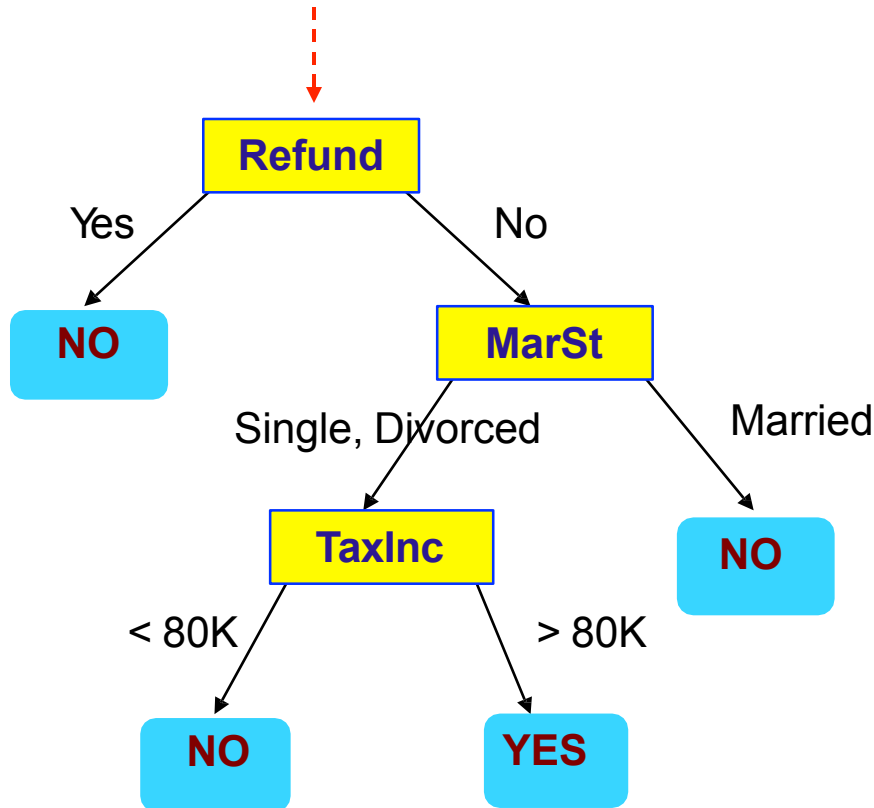


Apply Model to Test Data

Test Data

Start from the root of tree.

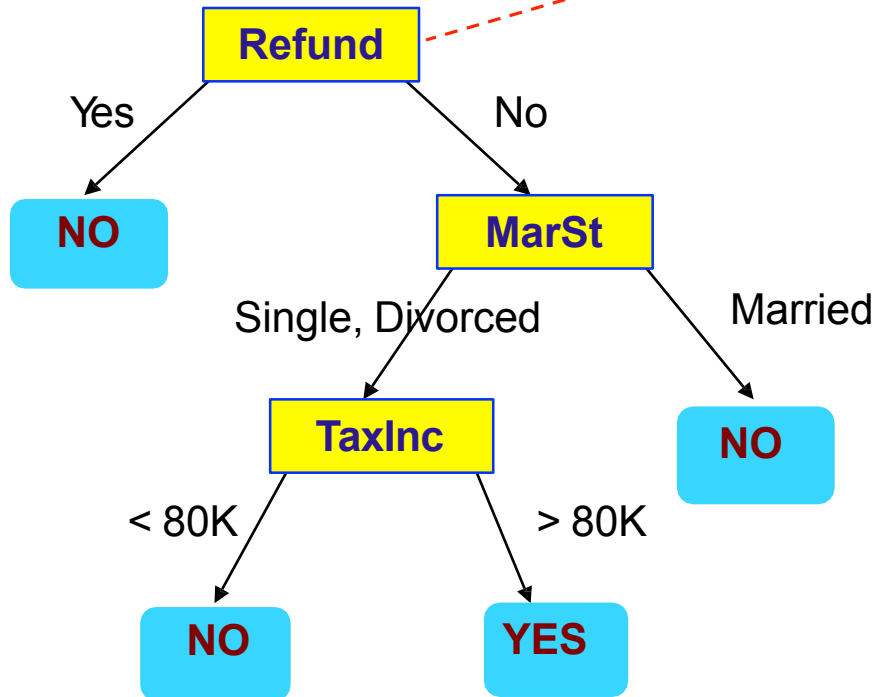
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

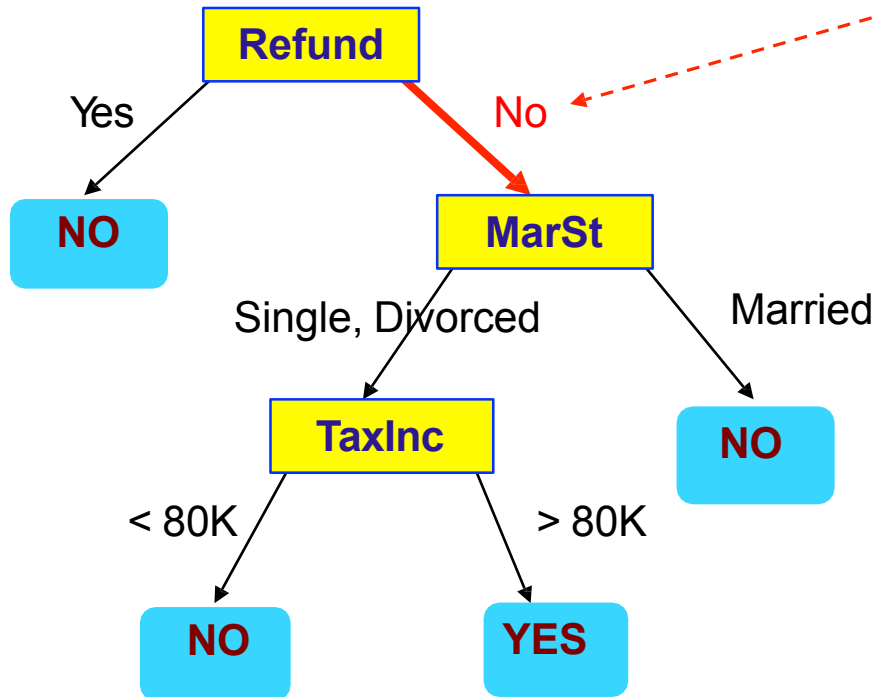
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

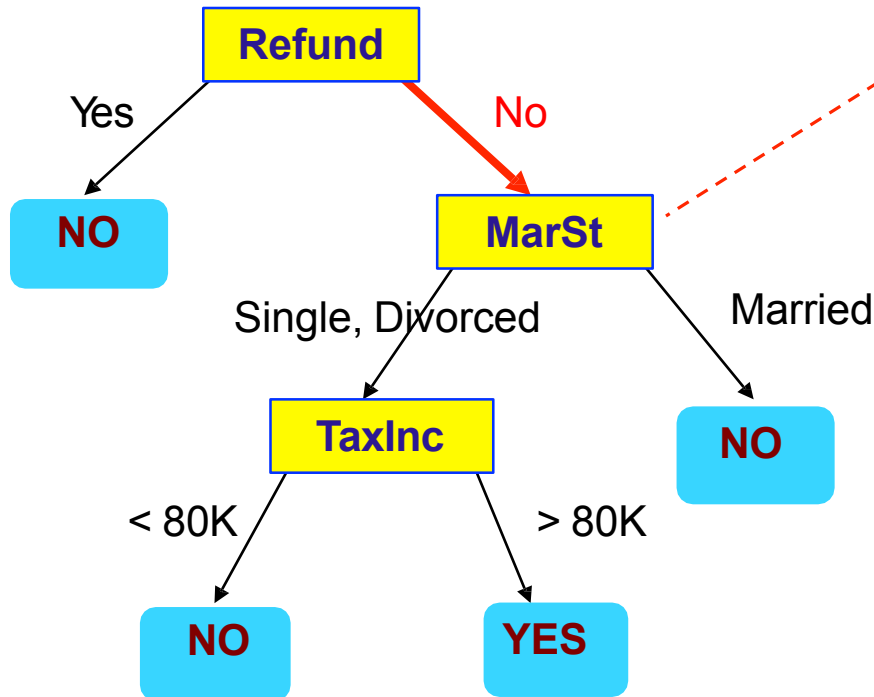
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

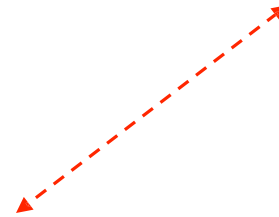
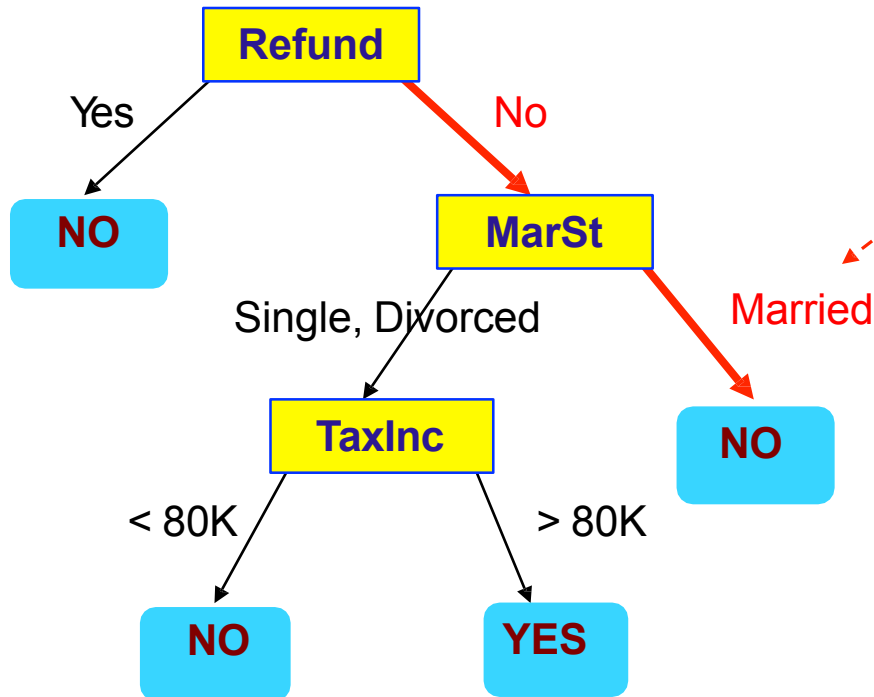
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

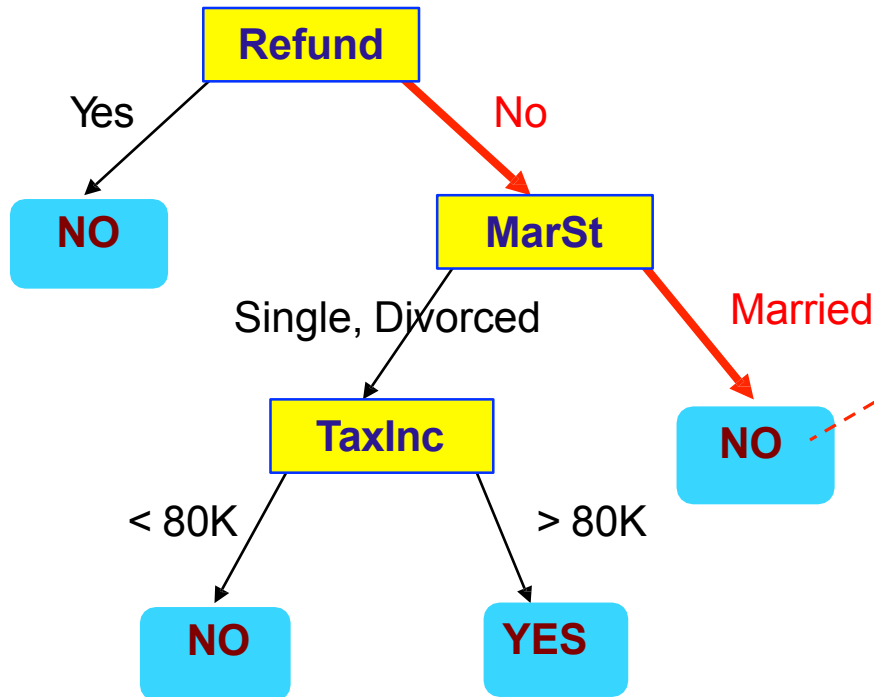
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"

Training examples: **9 yes / 5 no**

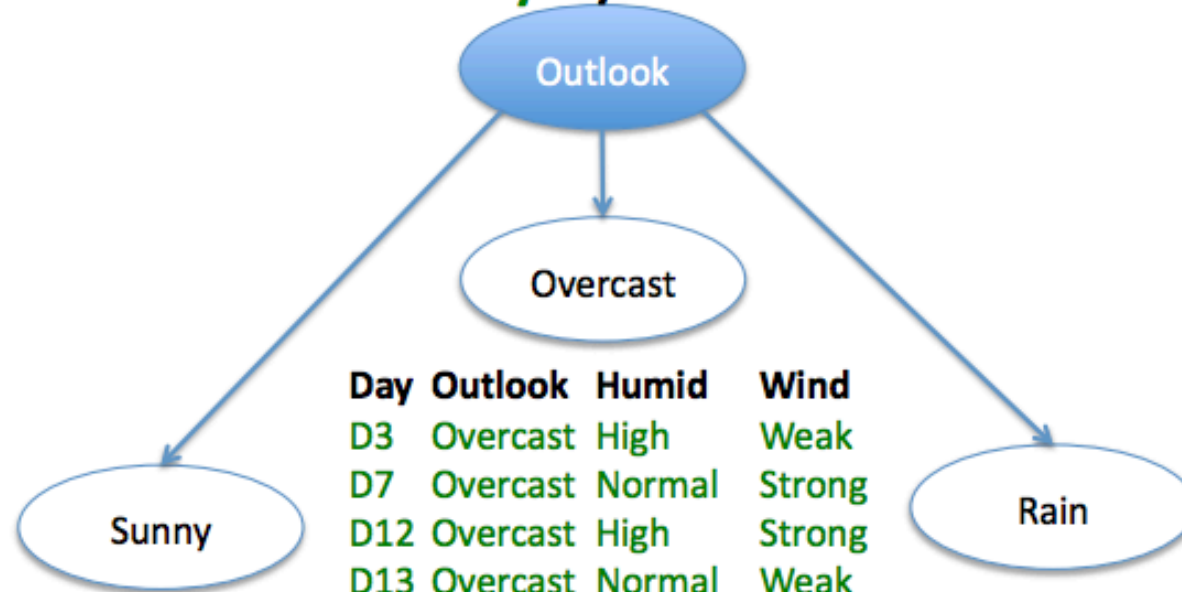
Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

New data:

D15	Rain	High	Weak	?
-----	------	------	------	---



9 yes / 5 no



Day	Outlook	Humid	Wind
D1	Sunny	High	Weak
D2	Sunny	High	Strong
D8	Sunny	High	Weak
D9	Sunny	Normal	Weak
D11	Sunny	Normal	Strong

2 yes / 3 no
split further

4 yes / 0 no
pure subset

Day	Outlook	Humid	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 yes / 2 no
split further

9 yes / 5 no

Outlook

Overcast

Sunny

Humidity

High

Normal

Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

Rain

4 yes / 0 no
pure subset

Day	Outlook	Humid	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

Day	Humid	Wind
D1	High	Weak
D2	High	Strong
D8	High	Weak

Day	Humid	Wind
D9	Normal	Weak
D11	Normal	Strong

3 yes / 2 no
split further

New data:

Day	Outlook	Humid	Wind	
D15	Rain	High	Weak	→ Yes

9 yes / 5 no

Outlook

Overcast

Sunny

Humidity

High

Normal

Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

Rain

Wind

Weak

Strong

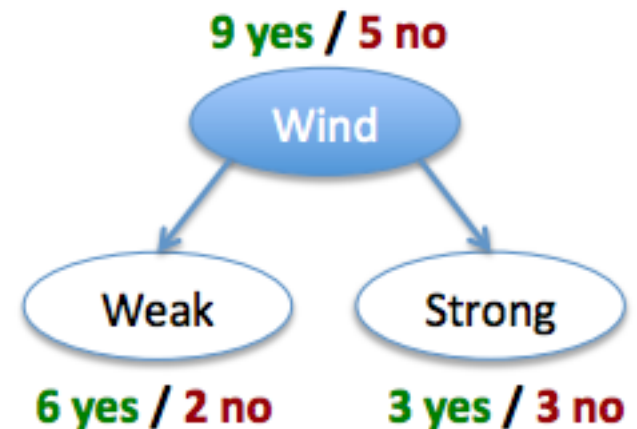
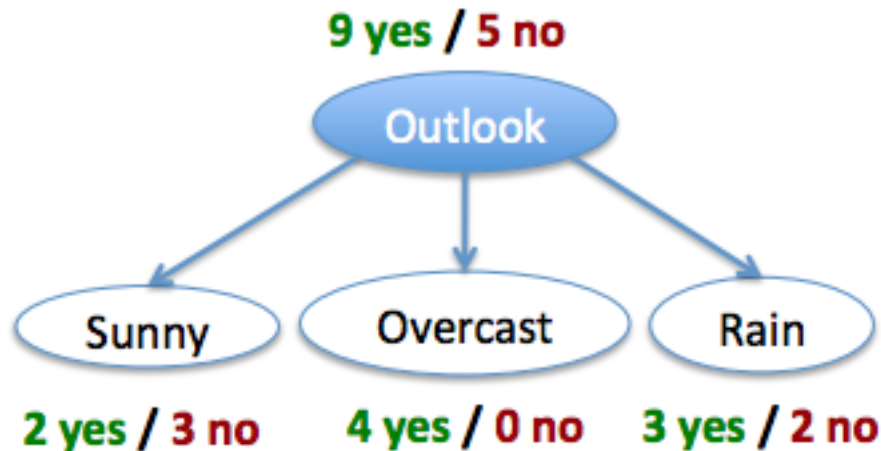
Day	Humid	Wind
D1	High	Weak
D2	High	Strong
D8	High	Weak

Day	Humid	Wind
D9	Normal	Weak
D11	Normal	Strong

Day	Humid	Wind
D4	High	Weak
D5	Normal	Weak
D10	Normal	Weak

Day	Humid	Wind
D6	Normal	Strong
D14	High	Strong

Which attribute to split on?



- Want to measure “purity” of the split
 - more certain about Yes/No after the split
 - pure set (4 yes / 0 no) => completely certain (100%)
 - impure (3 yes / 3 no) => completely uncertain (50%)

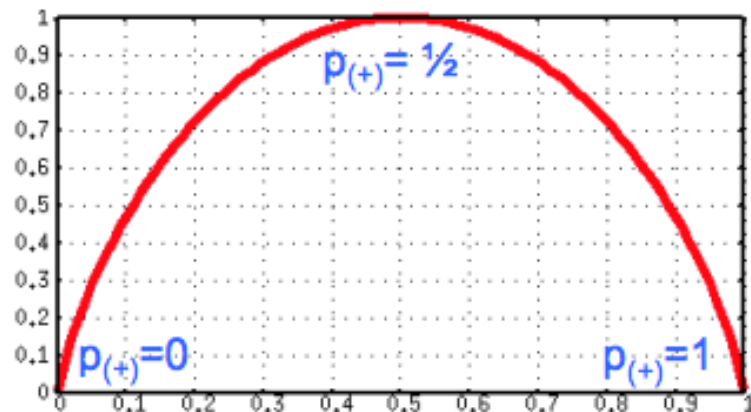
Entropy

- Entropy: $H(S) = -p_{(+)} \log_2 p_{(+)} - p_{(-)} \log_2 p_{(-)}$ bits
 - S ... subset of training examples
 - $p_{(+)} / p_{(-)}$... % of positive / negative examples in S
- Interpretation: assume item X belongs to S
 - how many bits need to tell if X positive or negative
- impure (3 yes / 3 no):

$$H(S) = -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} = 1 \text{ bits}$$

- pure set (4 yes / 0 no):

$$H(S) = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} = 0 \text{ bits}$$



Information Gain

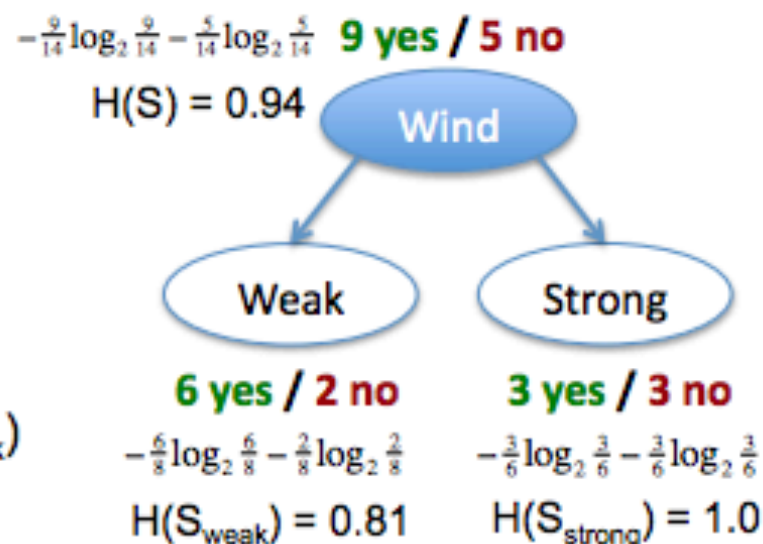
- Want many items in pure sets
- Expected drop in entropy after split:

$$Gain(S, A) = H(S) - \sum_{V \in Values(A)} \frac{|S_V|}{|S|} H(S_V)$$

V ... possible values of A
 S ... set of examples $\{X\}$
 S_V ... subset where $X_A = V$

- Mutual Information
 - between attribute A and class labels of S

$$\begin{aligned}
 Gain(S, Wind) &= H(S) - \frac{8}{14} H(S_{weak}) - \frac{6}{14} H(S_{strong}) \\
 &= 0.94 - \frac{8}{14} * 0.81 - \frac{6}{14} * 1.0 \\
 &= 0.049
 \end{aligned}$$



Naïve Bayes

Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:

$$P(C|A) = \frac{P(A,C)}{P(A)}$$

$$P(A|C) = \frac{P(A,C)}{P(C)}$$

- Bayes theorem:
$$P(C|A) = \frac{P(A|C)P(C)}{P(A)}$$

Example of Bayes Theorem

- Given:
 - A doctor knows that meningitis causes stiff neck 50% of the time
 - Prior probability of any patient having meningitis is $1/50,000$
 - Prior probability of any patient having stiff neck is $1/20$
- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M | S) = \frac{P(S | M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes (A_1, A_2, \dots, A_n)
 - Goal is to predict class C
 - Specifically, we want to find the value of C that maximizes $P(C \mid A_1, A_2, \dots, A_n)$
- Can we estimate $P(C \mid A_1, A_2, \dots, A_n)$ directly from data?

Bayesian Classifiers

- Approach:
 - compute the posterior probability $P(C \mid A_1, A_2, \dots, A_n)$ for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \dots A_n) = \frac{P(A_1 A_2 \dots A_n \mid C) P(C)}{P(A_1 A_2 \dots A_n)}$$

- Choose value of C that maximizes $P(C \mid A_1, A_2, \dots, A_n)$
- Equivalent to choosing value of C that maximizes $P(A_1, A_2, \dots, A_n \mid C) P(C)$

- How to estimate $P(A_1, A_2, \dots, A_n \mid C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
 - $P(A_1, A_2, \dots, A_n | C) = P(A_1 | C_j) P(A_2 | C_j) \dots P(A_n | C_j)$
 - Can estimate $P(A_i | C_j)$ for all A_i and C_j .
 - New point is classified to C_j if $P(C_j) \prod P(A_i | C_j)$ is maximal.

How to Estimate Probabilities from Data?

- Class: $P(C) = N_c / N$

– e.g., $P(\text{No}) = 7/10$,
 $P(\text{Yes}) = 3/10$

Tid	Refund	Marital Status	Taxable Income	Evade
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

For discrete attributes:

$$P(A_i \mid C_k) = |A_{ik}| / N_{C_k}$$

– where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k

– Examples:

$$P(\text{Status}=\text{Married} \mid \text{No}) = 4/7$$

$$P(\text{Refund}=\text{Yes} \mid \text{Yes})=0$$

How to Estimate Probabilities from Data?

- For continuous attributes:
 - **Discretize** the range into bins
 - one ordinal attribute per bin
 - violates independence assumption
 - **Two-way split:** $(A < v)$ or $(A > v)$
 - choose only one of the two splits as new attribute

k

How to Estimate Probabilities from Data?

Taxable Income	Evade
125K	No
100K	No
70K	No
120K	No
95K	Yes
60K	No
220K	No
85K	Yes
75K	No
90K	Yes

- Normal distribution:

- – One for each (A_i, c_i) pair
- For (Income, Class=No):
 - If Class=No
 - sample mean = 110
 - sample variance = 2975

$$\frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(A_i - \mu_{ij})^2}{2 \sigma_{ij}^2}}$$

$$P(A | c) = e$$

$$P(\text{Income} = 120 | \text{No}) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Example of Naïve Bayes Classifier

Given a Test Record:

$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$

naive Bayes Classifier:

$P(\text{Refund}=\text{Yes}|\text{No}) = 3/7$
 $P(\text{Refund}=\text{No}|\text{No}) = 4/7$
 $P(\text{Refund}=\text{Yes}|\text{Yes}) = 0$
 $P(\text{Refund}=\text{No}|\text{Yes}) = 1$
 $P(\text{Marital Status}=\text{Single}|\text{No}) = 2/7$
 $P(\text{Marital Status}=\text{Divorced}|\text{No}) = 1/7$
 $P(\text{Marital Status}=\text{Married}|\text{No}) = 4/7$
 $P(\text{Marital Status}=\text{Single}|\text{Yes}) = 2/7$
 $P(\text{Marital Status}=\text{Divorced}|\text{Yes}) = 1/7$
 $P(\text{Marital Status}=\text{Married}|\text{Yes}) = 0$

- $P(X|\text{Class}=\text{No}) = P(\text{Refund}=\text{No}|\text{Class}=\text{No})$
 $\times P(\text{Married}|\text{Class}=\text{No})$
 $\times P(\text{Income}=120\text{K}|\text{Class}=\text{No})$
 $= 4/7 \times 4/7 \times 0.0072 = 0.0024$
- $P(X|\text{Class}=\text{Yes}) = P(\text{Refund}=\text{No}|\text{Class}=\text{Yes})$
 $\times P(\text{Married}|\text{Class}=\text{Yes})$
 $\times P(\text{Income}=120\text{K}|\text{Class}=\text{Yes})$
 $= 1 \times 0 \times 1.2 \times 10^{-9} = 0$

Since $P(X|\text{No})P(\text{No}) > P(X|\text{Yes})P(\text{Yes})$

Therefore $P(\text{No}|X) > P(\text{Yes}|X)$

$\Rightarrow \text{Class} = \text{No}$

Naïve Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original : } P(A_i | C) = \frac{N_{ic}}{N_c}$$

c: number of classes

$$\text{Laplace : } P(A_i | C) = \frac{N_{ic} + 1}{N_c + c}$$

p: prior probability

m: parameter

$$\text{m - estimate : } P(A_i | C) = \frac{N_{ic} + mp}{N_c + m}$$

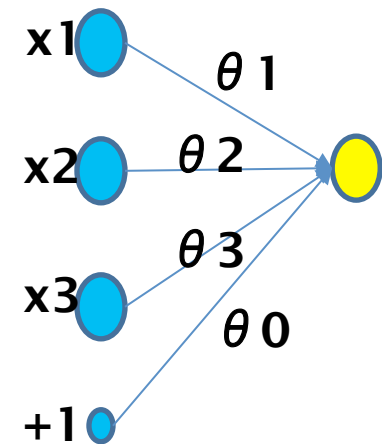
Logistic Regression

- Can be visualized as a “single neuron”.

$$P(y = 1 | x; \theta) = \sigma(\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n) = \sigma(\theta_0 + \sum_{i=1}^n \theta_i x_i)$$

$$P(y = 0 | x; \theta) = 1 - P(y = 1 | x; \theta)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Using the **sigmoid function** as the **activation function**

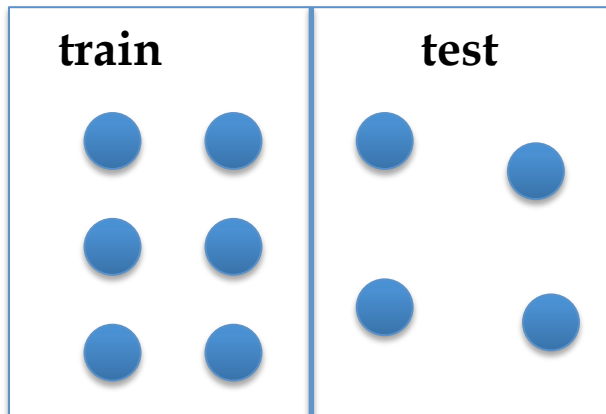
Evaluation

- For all supervised problems, it's important to understand how well your model is performing
- What we try to estimate is how well you *will* perform in the future, on new data also drawn from X

X










labeled data

X

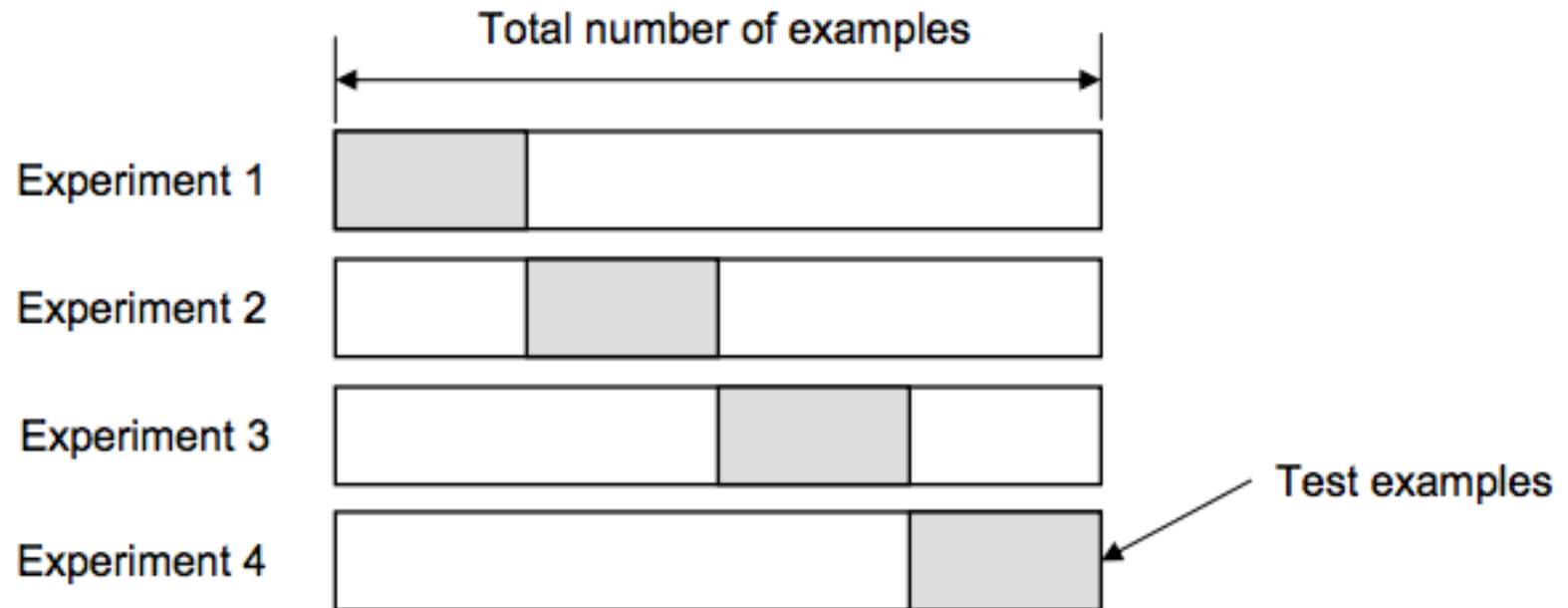


Problems?

X

train	dev	test
		
		
		
		
		

K-fold cross validation



$K = 4$

Evaluation Metric

Binary Classification

Confusion Matrix:

	PREDICTED (y_{pred})		
		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d
ACTUAL CLASS (y_{true})			

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

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

Precision

Precision: proportion of predicted class that are actually that class. i.e., if a class prediction is made, should you trust it?

$$\text{Precision} = \frac{a}{a + c} = \frac{TP}{TP + FP}$$

	Predicted (y_{pred})		
		Class=Yes	Class=No
	Actual class (y_{true})		
	Class=Yes	48	70
	Class=No	0	10347

Recall

Recall: proportion of true class actually predicted to be that class

$$\text{Recall} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

	Predicted (y_{pred})		
		Class=Yes	Class=No
	Actual class (y_{true})		
	Class=Yes	48	70
	Class=No	0	10347

Precision, Recall, F1

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

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

$$\text{F-measure (F)} = \frac{1}{\left(\frac{1/r + 1/p}{2} \right)} = \frac{2rp}{r + p} = \frac{2a}{2a + b + c} = \frac{2TP}{2TP + FP + FN}$$

Multiclass classification

	Apple	Orange	Banana
Precision	?	?	?
Recall	?	?	?

Predicted









True label



100	2	15
0	104	30
30	40	70

Multiclass classification

	Apple	Orange	Banana
Precision	0.769	0.712	0.609
Recall	0.855	0.776	0.500

		Predicted		
				
True label		100	2	15
		0	104	30
		30	40	70

Summary

- Classification task
- Some learning algorithms
- Evaluation