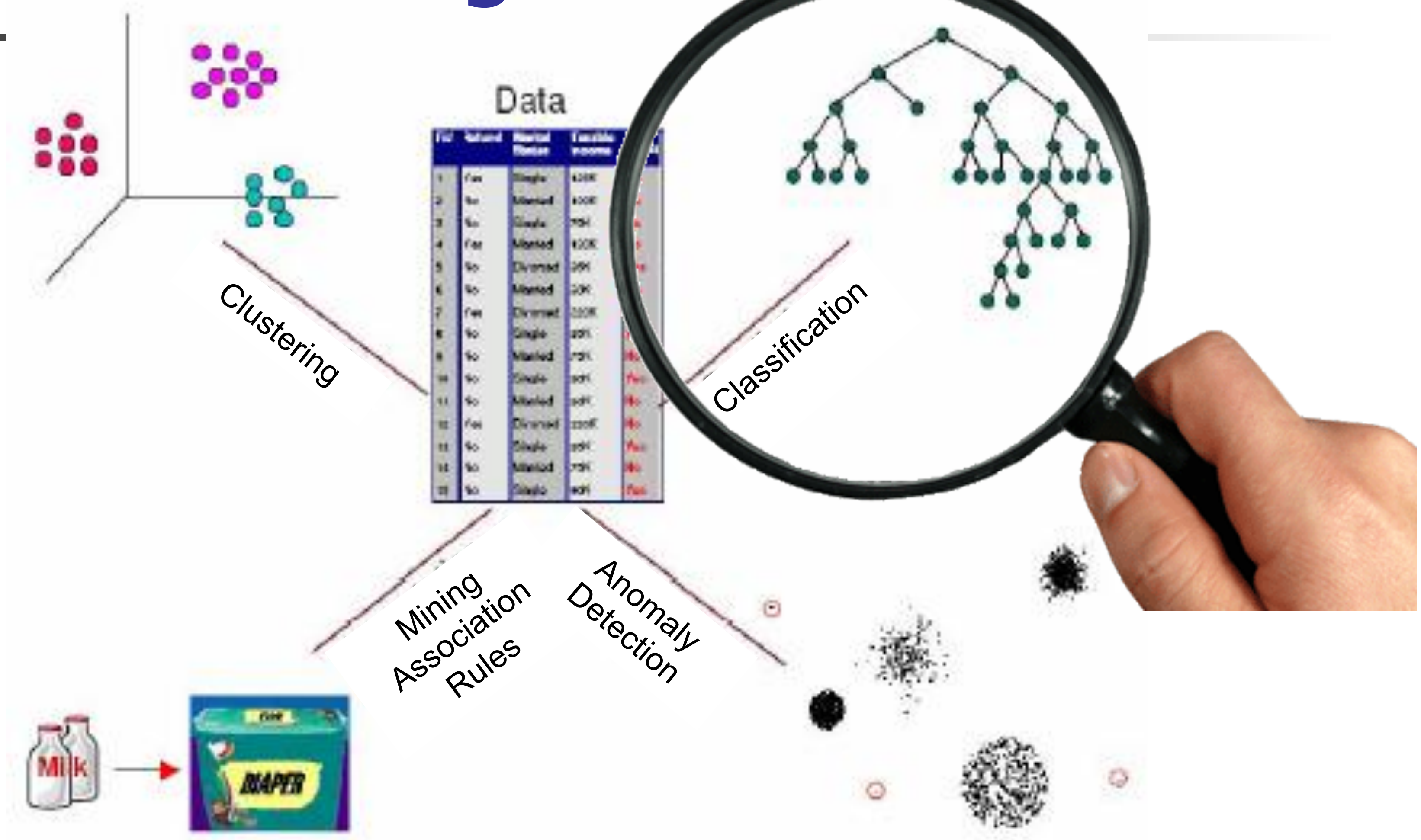
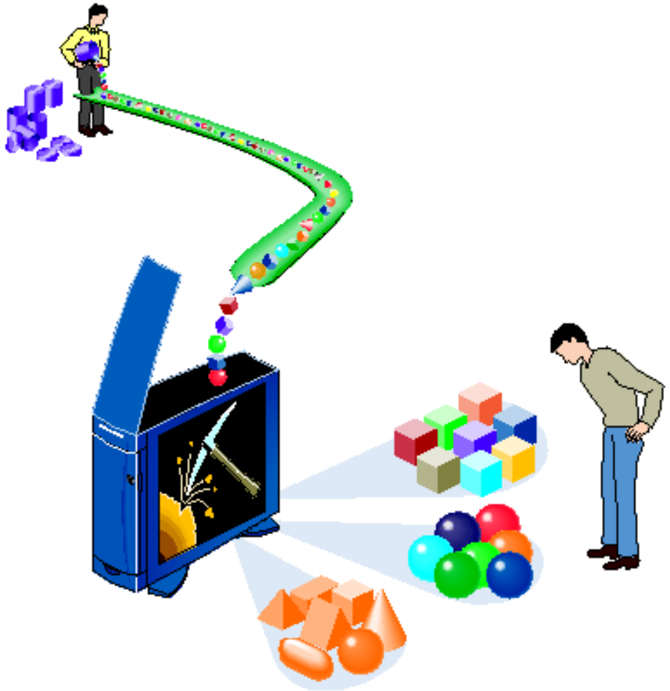
Data Mining Tasks



Data Mining

- Classification Algorithms (I)



A Motivating Example

- A simple classification problem…
  - I know there is Salmon in this river
  - When I pick up a fish from this river, can you tell me whether this fish is Salmon?
- Assume you do not know how a Salmon looks like
  - Then… How to solve this problem?

A Motivating Example

- Since you know nothing about Salmon or Tuna, the first thing you need to do is...

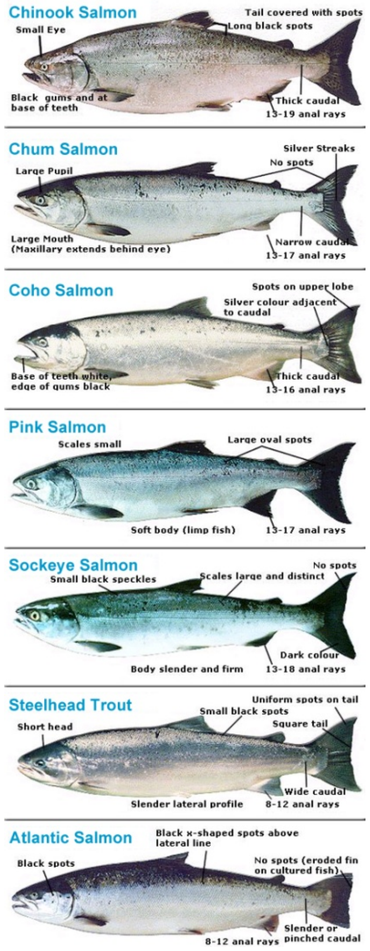
LEARN!

Different Kinds of Learning

- Two types of learning
  1. Passive learning
  2. Active learning

Different Kinds of Learning

- **Passive learning**
  - Find an expert
  - The expert tells you all the characteristics of Salmon
  - You simply memorize and apply what you have learned



Different Kinds of Learning

- **Active learning**
  - Find an expert
  - The expert catches a lot of Fish
  - The expert only tells you which of them are Salmon, but does not tell you their characteristics
  - You identify the characteristics of salmon by yourself

salmon

pinkish in color and have spots on their fins and back, blah blah blah…

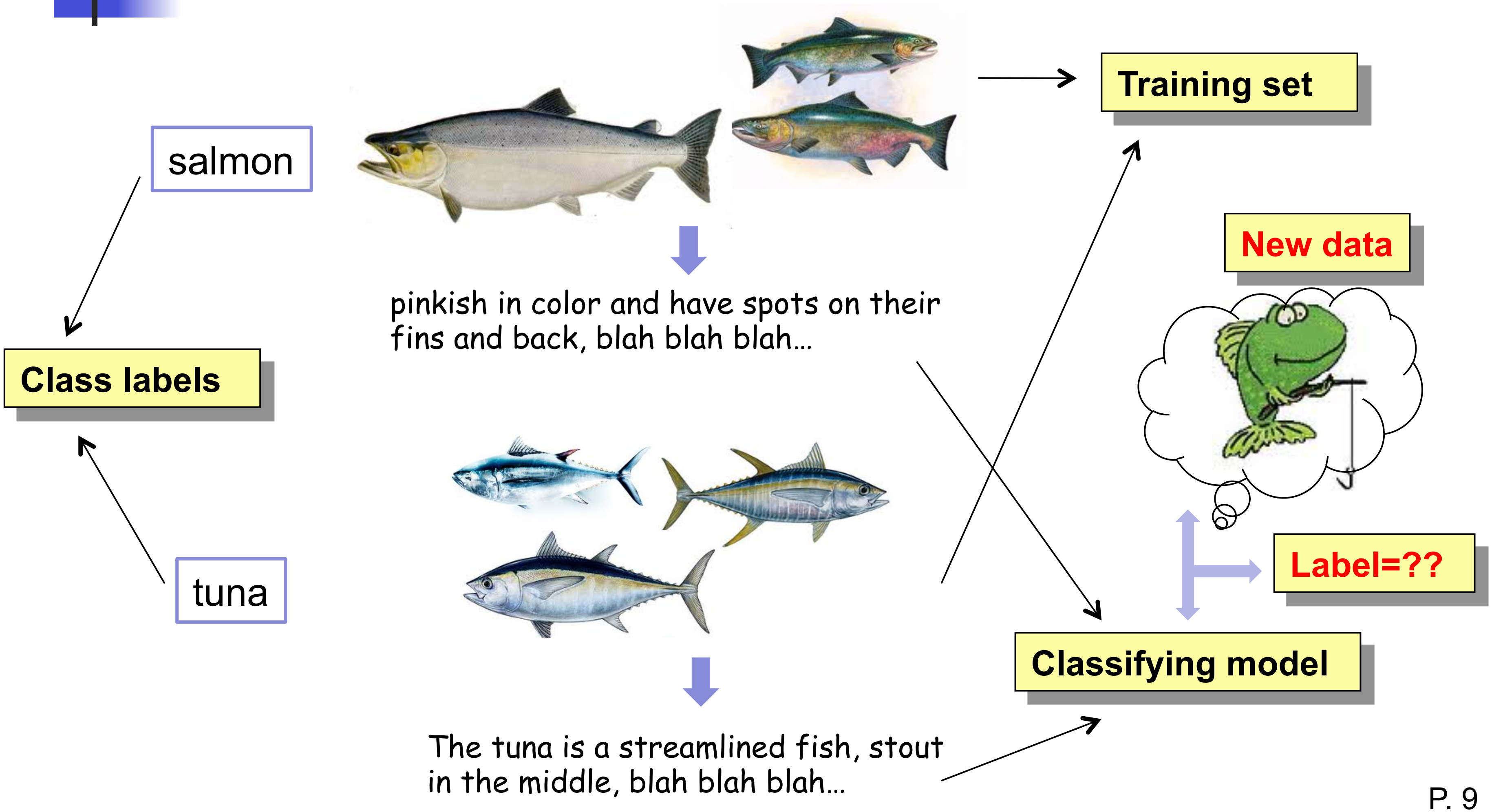
tuna

The tuna is a streamlined fish, stout in the middle, blah blah blah…

Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains:
    - A set of *attributes* (i.e., characteristics), and
    - One *class* attribute (i.e., class label)
- 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

Classification: Example



salmon

pinkish in color and have spots on their fins and back, blah blah blah…

Class labels

tuna

The tuna is a streamlined fish, stout in the middle, blah blah blah…

Training set

New data

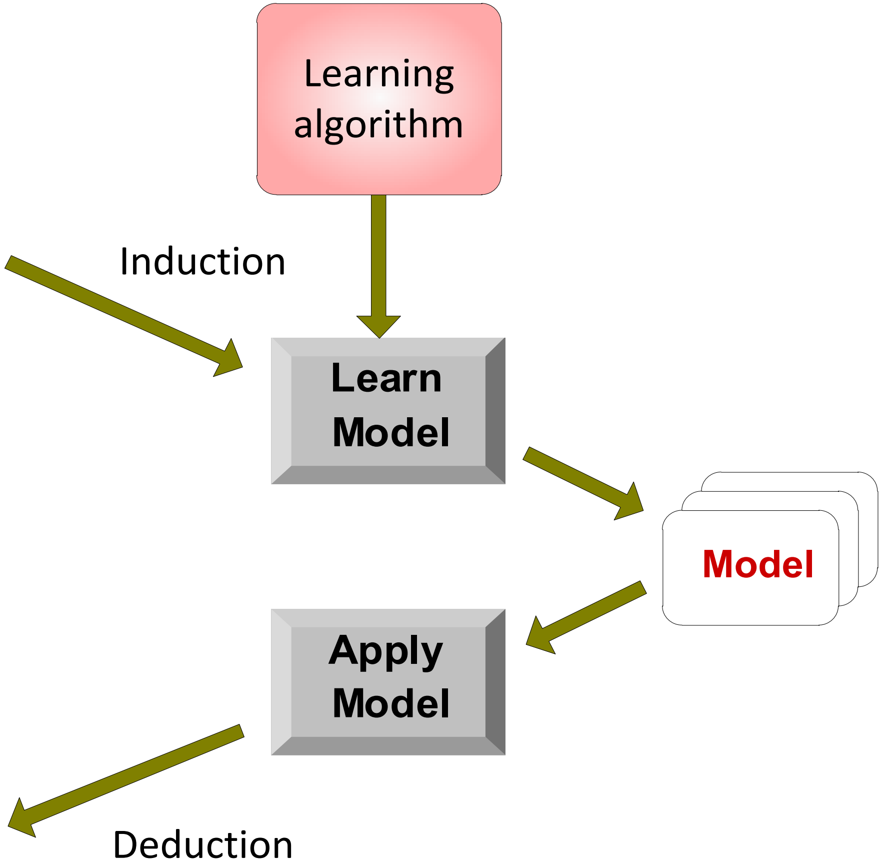
Label=??

Classifying model

Classification: Example

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

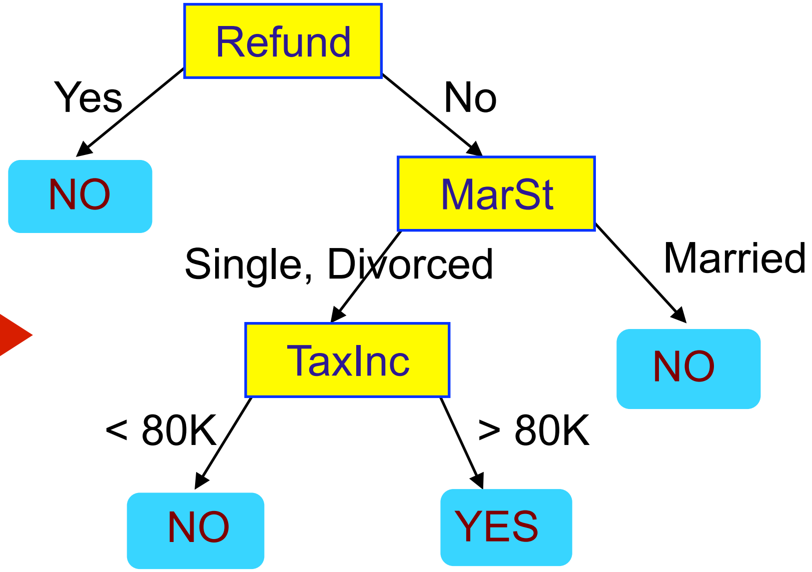
Deduction

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	**?**

Example of a 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**

Training Data

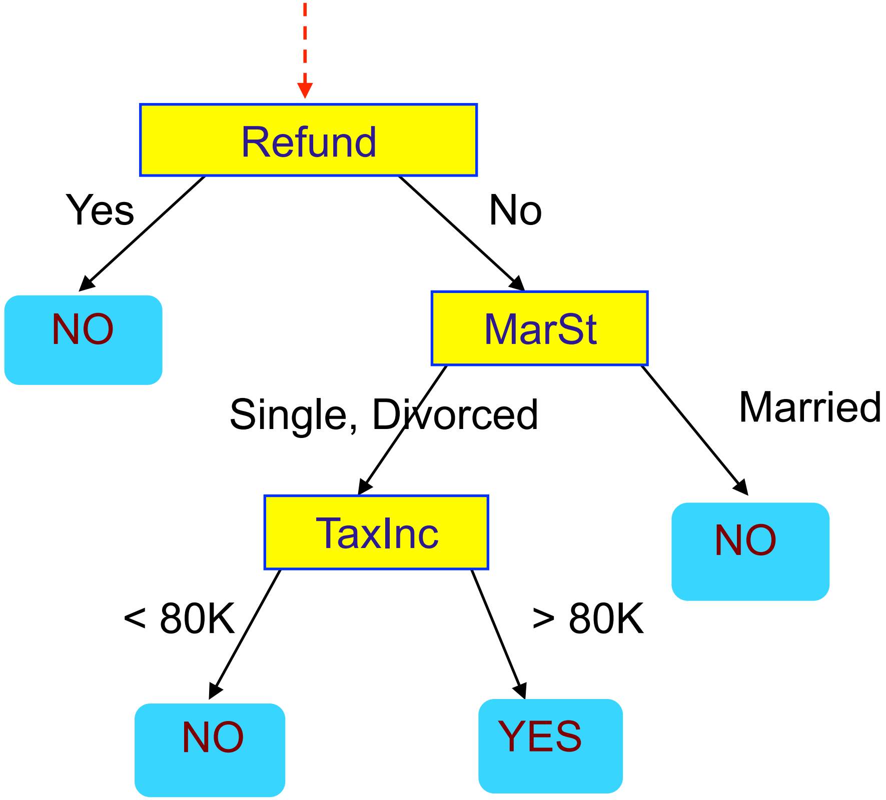


Model: Decision Tree

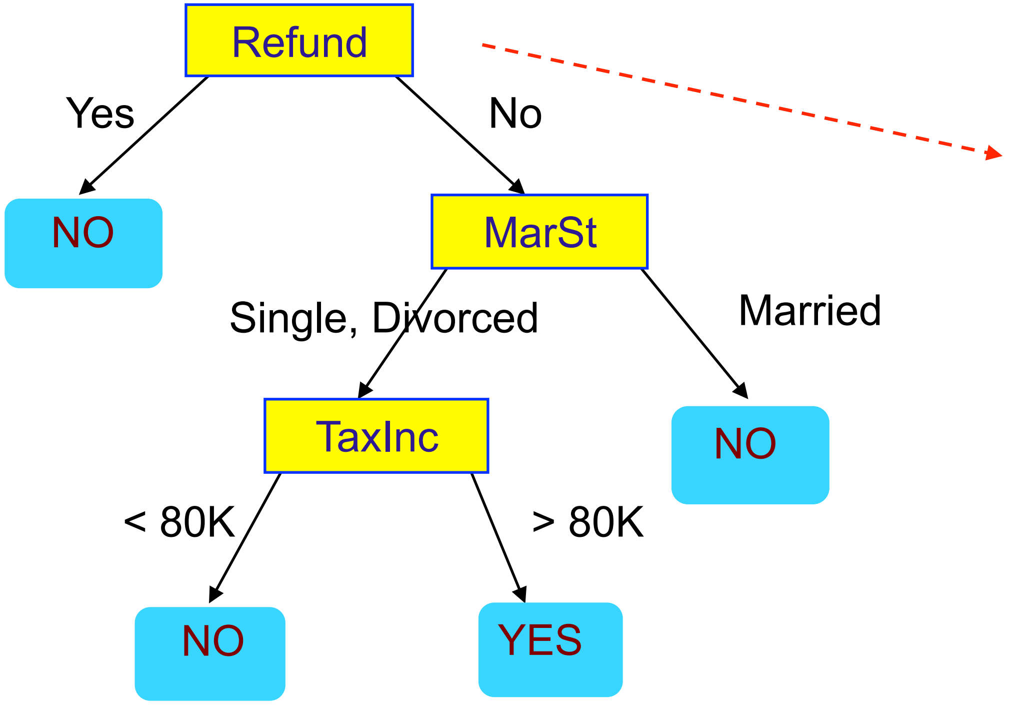
Apply Model to Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Start from the root of tree.

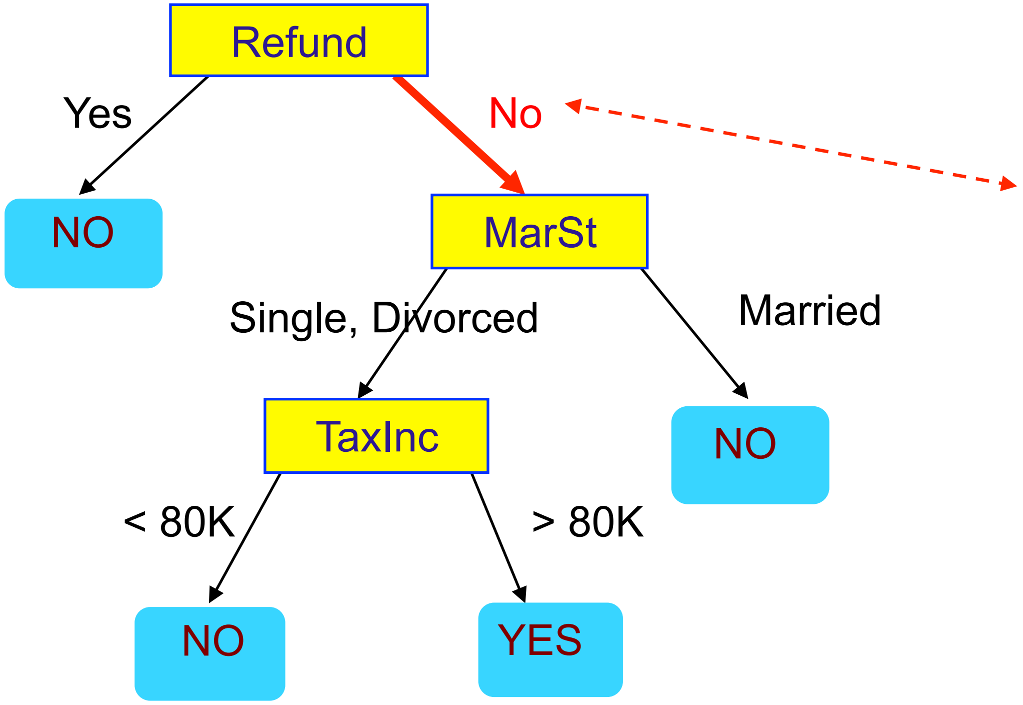


Apply Model to Test Data



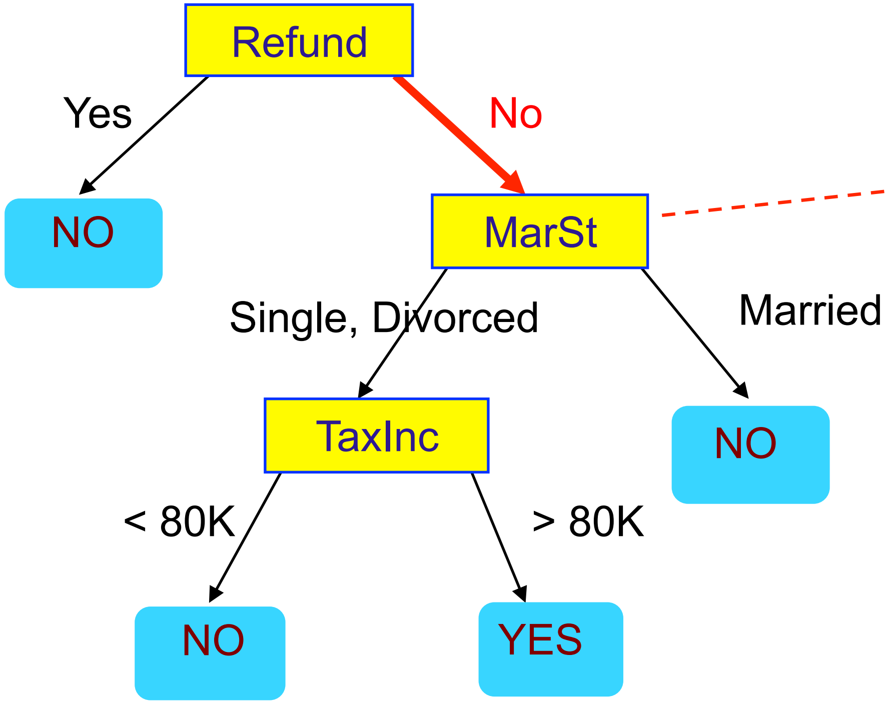
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

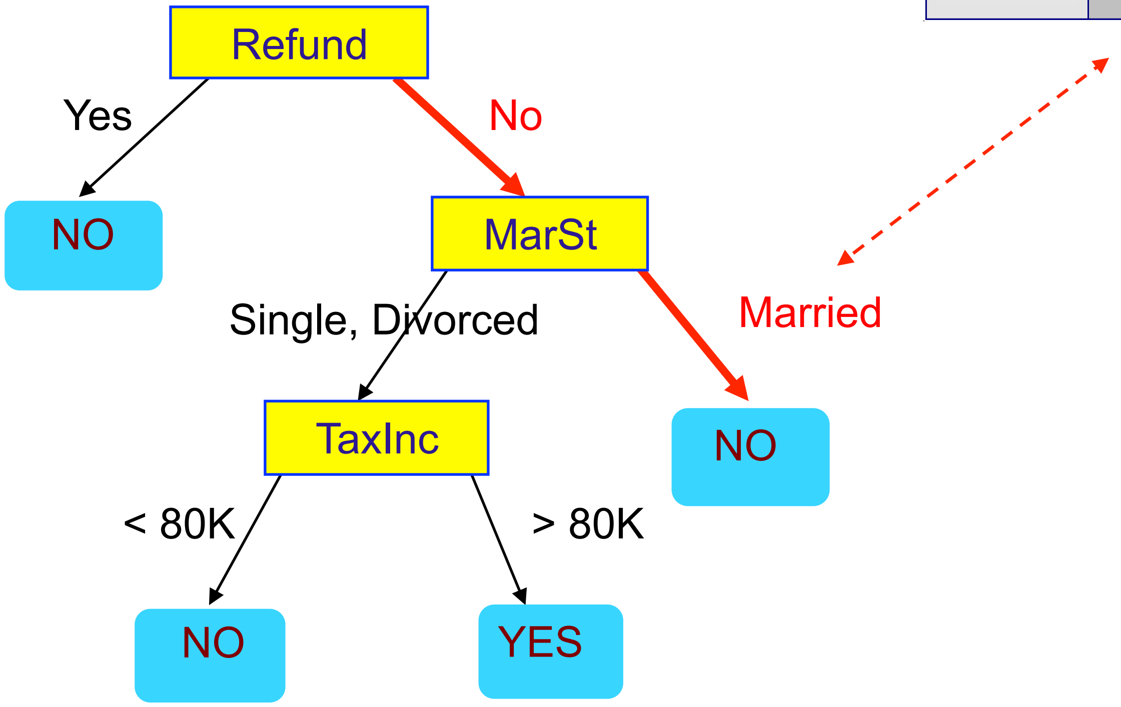
Apply Model to Test Data



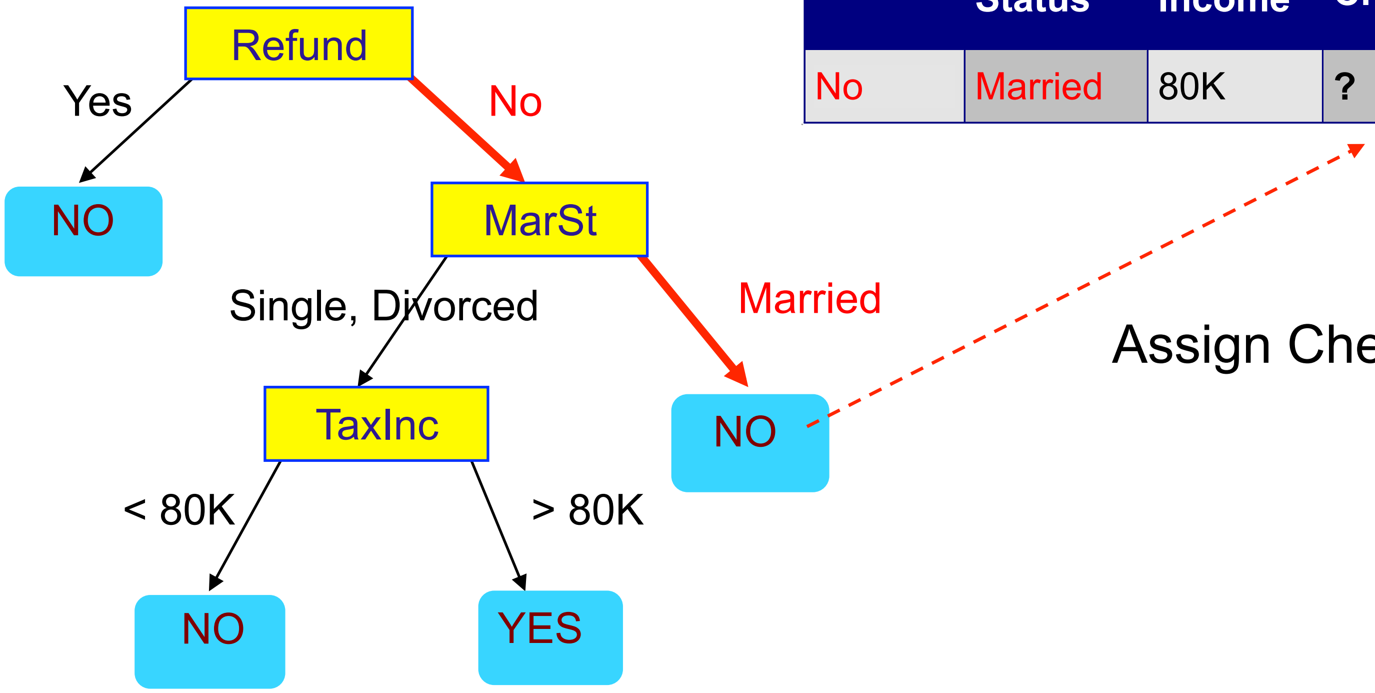
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data



Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assign Cheat to “No”

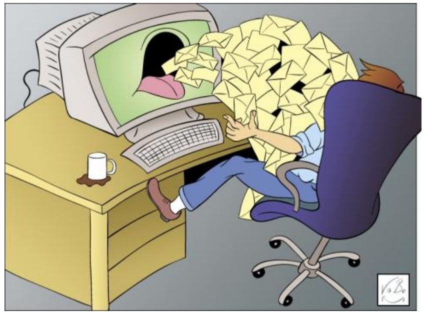
Classification in Data Mining

- In data mining, we are always interested in active learning
  - You are an expert
  - You catch a lot of Fish
  - You only tell the model which of them are Salmon, but do not tell the characteristics
  - The model identifies the characteristics by itself

- Question:
  - As long as you are an expert, why don't you simply tell the characteristics of Salmon to the model?

Classification in Data Mining

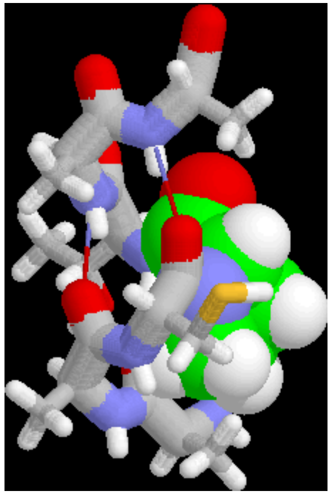
- Answer:
  - Even an expert may sometimes find it difficult to **generalize/ extract/identify** the characteristics of some observations…



- An example:
  - You receive lots of emails. You must know which of them are spam and which of them are not spam
    - Yet, can you list **ALL** the characteristics of spam emails?
  - For active learning, you only need to tell the model which of them are spam, and which are not

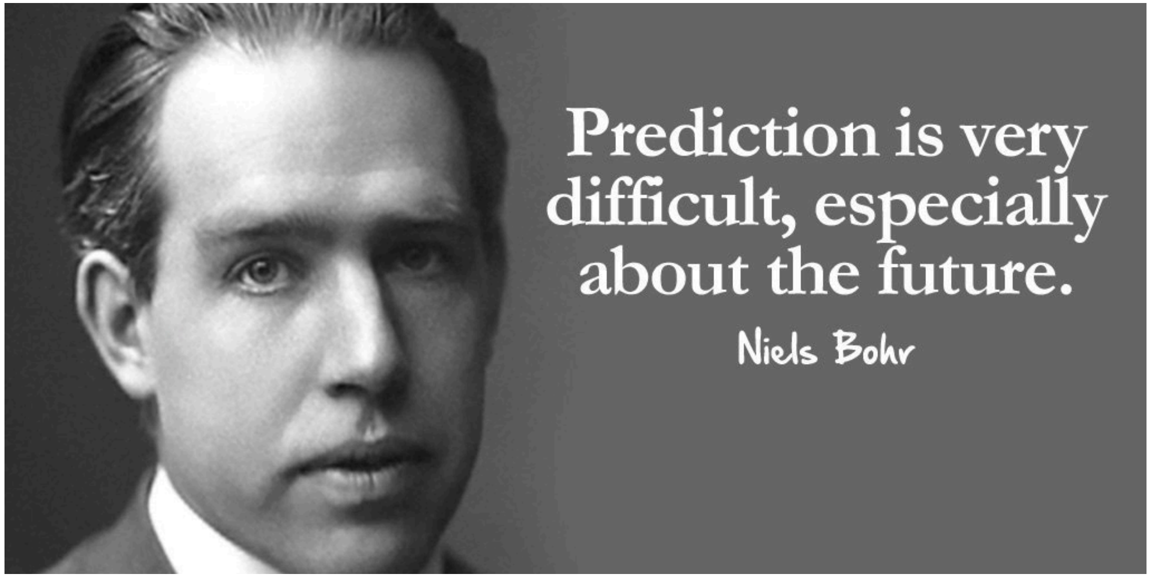
Examples of Classification Task

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Categorizing news stories as finance, weather, entertainment, sports, etc



Always Remember…

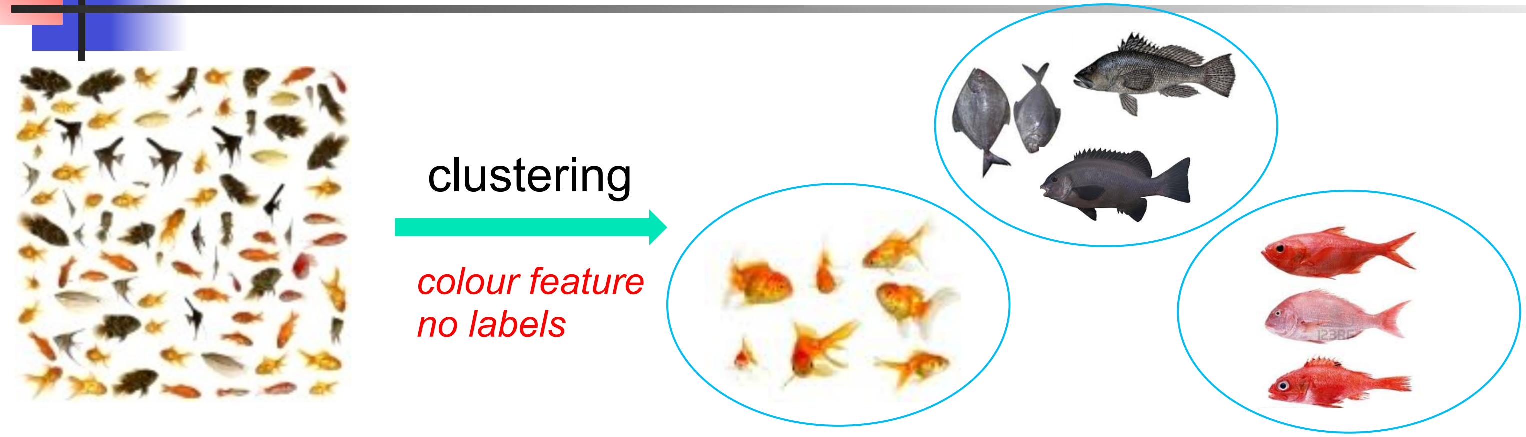
- From the data mining point of view…
  - Classification ≈ **Prediction** ≈ **Forecasting**
  - This is because the techniques are the same



Always Remember…

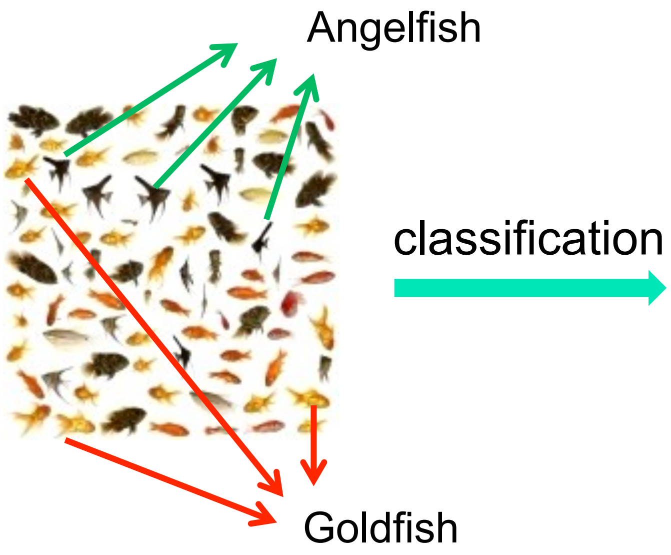
- Classification is also known as **"Supervised Learning"**
  - There must be an "expert" (you) to "supervise" the model
  - In contrast, **Clustering** is known as "Unsupervised Learning"
    - *In later lectures…*

Classification vs. Clustering



clustering

colour feature
no labels



Angelfish

classification

Goldfish

Angelfish:
Up to 6inches or 15cm. Their bodies are very thin, yet tall, their profile rounded, almost disc-shaped.

Salmon:
pinkish in color and have spots on their fins and back

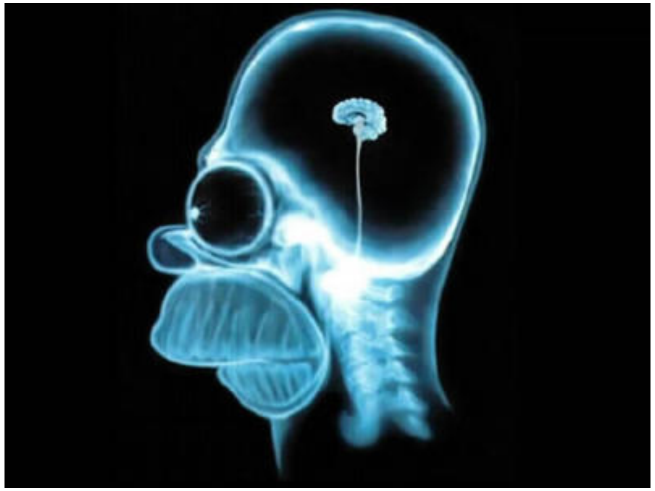
Tuna:
The tuna is a streamlined fish, stout in the middle

Classification—A Two-Step Process

1. Model construction
2. Model usage

Classification—A Two-Step Process

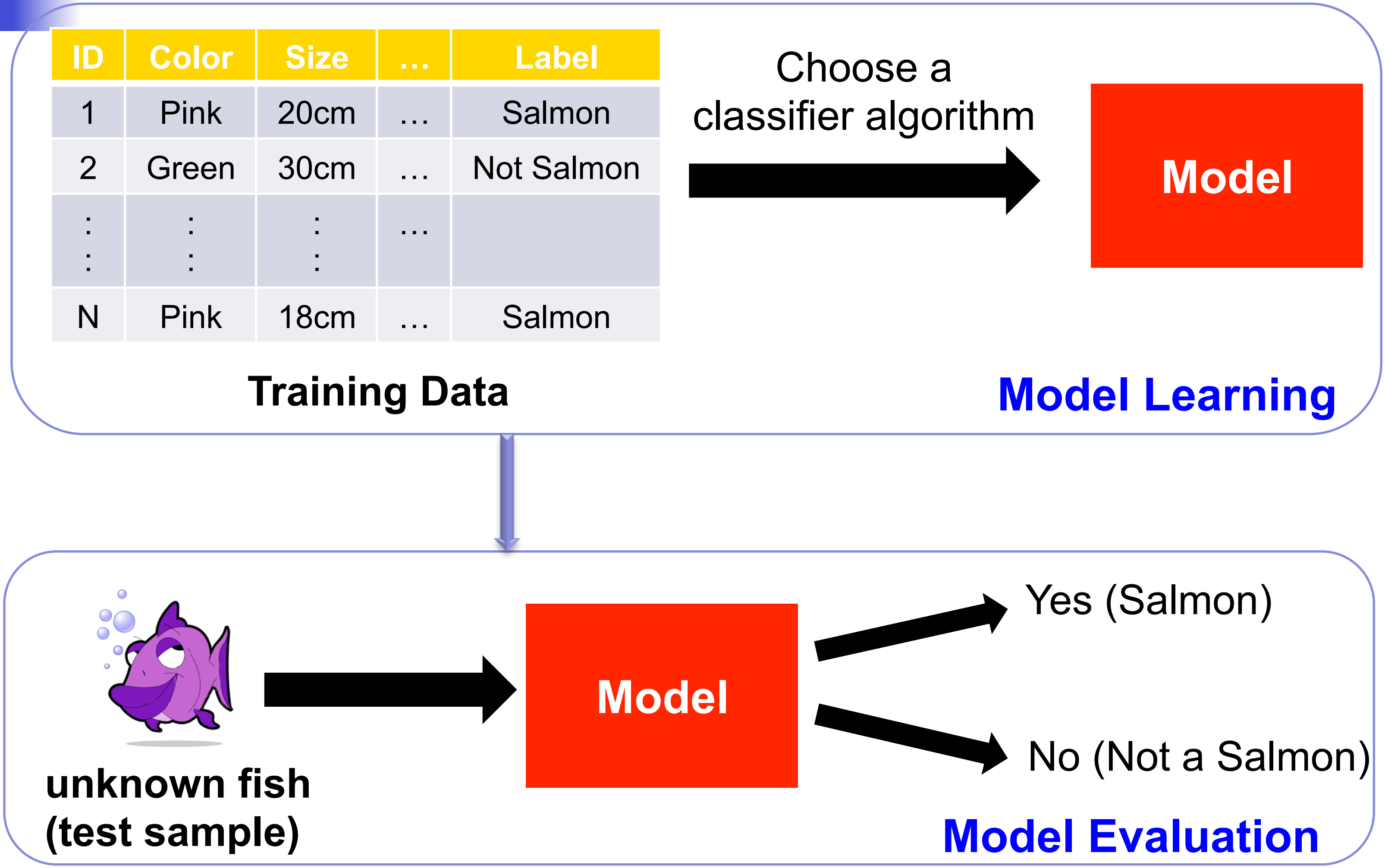
- **Model construction**: describes a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction is a training set
  - The model is represented as:
    - classification rules,
    - decision trees,
    - mathematical formulae, or
    - …



Classification—A Two-Step Process

- **Model usage:** for classifying future or unknown objects
  - Estimate **accuracy** of the model
    - The known labels of **test sample** is compared against the classified result from the model
    - Accuracy rate is the percentage of testing set samples that are **correctly** classified by the model
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Learning and Operation



ID	Color	Size	…	Label
1	Pink	20cm	…	Salmon
2	Green	30cm	…	Not Salmon
: :	: :	: :	…	
N	Pink	18cm	…	Salmon

Training Data

Choose a classifier algorithm

Model

Model Learning

unknown fish (test sample)

Model

Yes (Salmon)

No (Not a Salmon)

Model Evaluation

Classification Algorithms

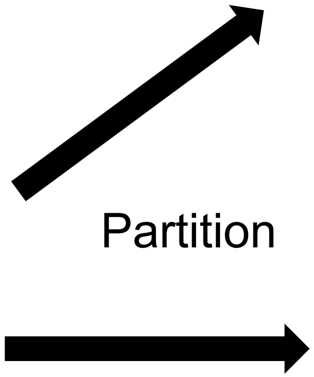
- Nearest Neighbor
- Naïve Bayes
- Decision Tree
- ...

- *But first, …*

Testing

- Prepare the training data and testing data

ID	Color	Size	…	Label
1	Pink	20cm	…	Salmon
2	Green	30cm	…	Not Salmon
: :	: :	: :	… …	: :
N	Pink	18cm	…	Salmon



Partition

ID	Color	Size	…	Label
1	Pink	20cm	…	Salmon
3	Green	32cm	…	Salmon
: :	: :	: :	… …	: :
K	Black	24cm	…	Not Salmon

Training Data

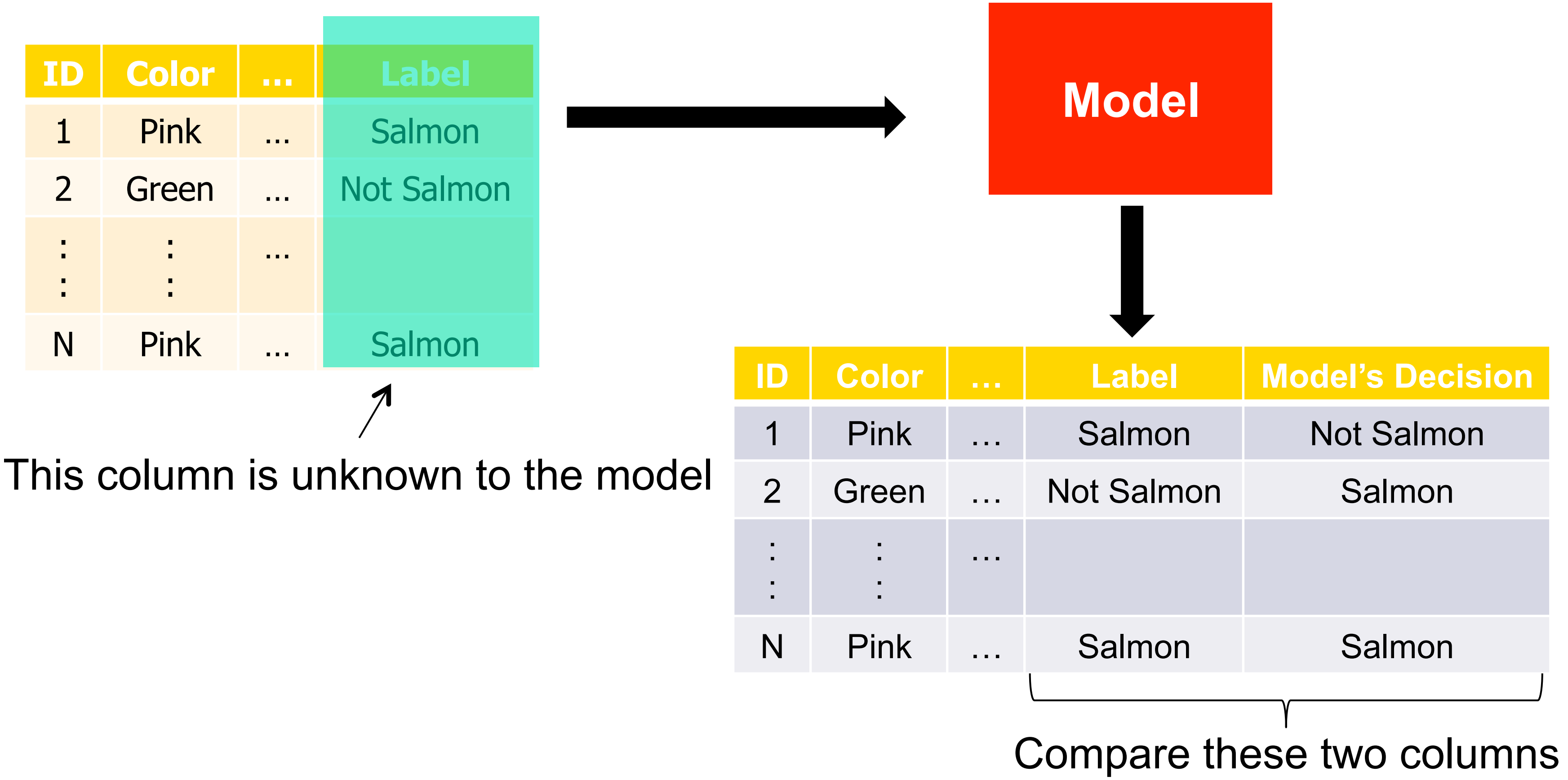
ID	Color	Size	…	Label
2	Green	30cm	…	Not Salmon
6	Grey	12cm	…	Not Salmon
: :	: :	: :	… …	: :
M	Pink	18cm	…	Salmon

Testing Data

data partitioning is discussed shortly…

Testing

- Testing process



ID	Color	...	Label
1	Pink	...	Salmon
2	Green	...	Not Salmon
: :	: :	...	
N	Pink	...	Salmon

This column is unknown to the model

ID	Color	…	Label	Model's Decision
1	Pink	…	Salmon	Not Salmon
2	Green	…	Not Salmon	Salmon
: :	: :	…		
N	Pink	…	Salmon	Salmon

Compare these two columns

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?

- Methods for Performance Evaluation
  - How to obtain reliable estimates?
    - how to partition the data?

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?

- Methods for Performance Evaluation
  - How to obtain reliable estimates?
    - how to partition the data?

Performance Evaluation

- Confusion Matrix:

		Prediction	
		Salmon	Not Salmon
Actual Class	Salmon	A	B
	Not Salmon	C	D

A: TP (true positive)

B: FN (false negative)

C: FP (false positive)

D: TN (true negative)

$$\text{Accuracy} = \frac{A+D}{A+B+C+D} = \frac{TP+TN}{TP+TN+FP+FN}$$

An Example

🔴	Red
🔴	Not Red
🟢	Not Red
🔴	Red
🔵	Red
🔴	Red
🔴	Not Red
🔴	Red
🟣	Not Red
🔵	Not Red

True Positive: 4 (Red, Red)

True Negative:

False Positive:

False Negative:

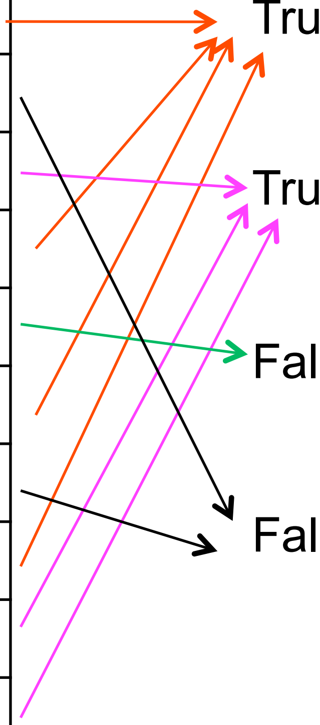
Accuracy = ?

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

		Prediction	
		Salmon	Not Salmon
Actual Class	Salmon	TP	FN
	Not Salmon	FP	TN

An Example

🔴	Red
🔴	Not Red
🟢	Not Red
🔴	Red
🔵	Red
🔴	Red
🔴	Not Red
🔴	Red
🟣	Not Red
🔵	Not Red



True Positive:	4	(Red, Red)
True Negative:	3	(Not, Not)
False Positive:	1	(Not, Red)
False Negative:	2	(Red, Not)

Accuracy = (4 + 3) / (4 + 3 + 1 + 2) = 70%

Limitation of Accuracy

- Consider…
  - The Total number of fish in the testing sample= 10,000
  - Number of Non-Salmon = 9990
  - Number of Salmon = 10

- If a model predicts everything to be class non-salmon:
  - Accuracy is 9990/10000 = 99.9 %!!!
  - Accuracy could be misleading because this model cannot detect any Salmon!

Precision and Recall

- **Precision:** exactness – what % of tuples that the classifier labeled as positive are actually positive?
- **Recall:** completeness – what % of positive tuples did the classifier label as positive?
- Perfect score is 1.0
- Usually, there is an Inverse relationship between the two

Precision and Recall

- Measuring the quality (effectiveness) of the model:

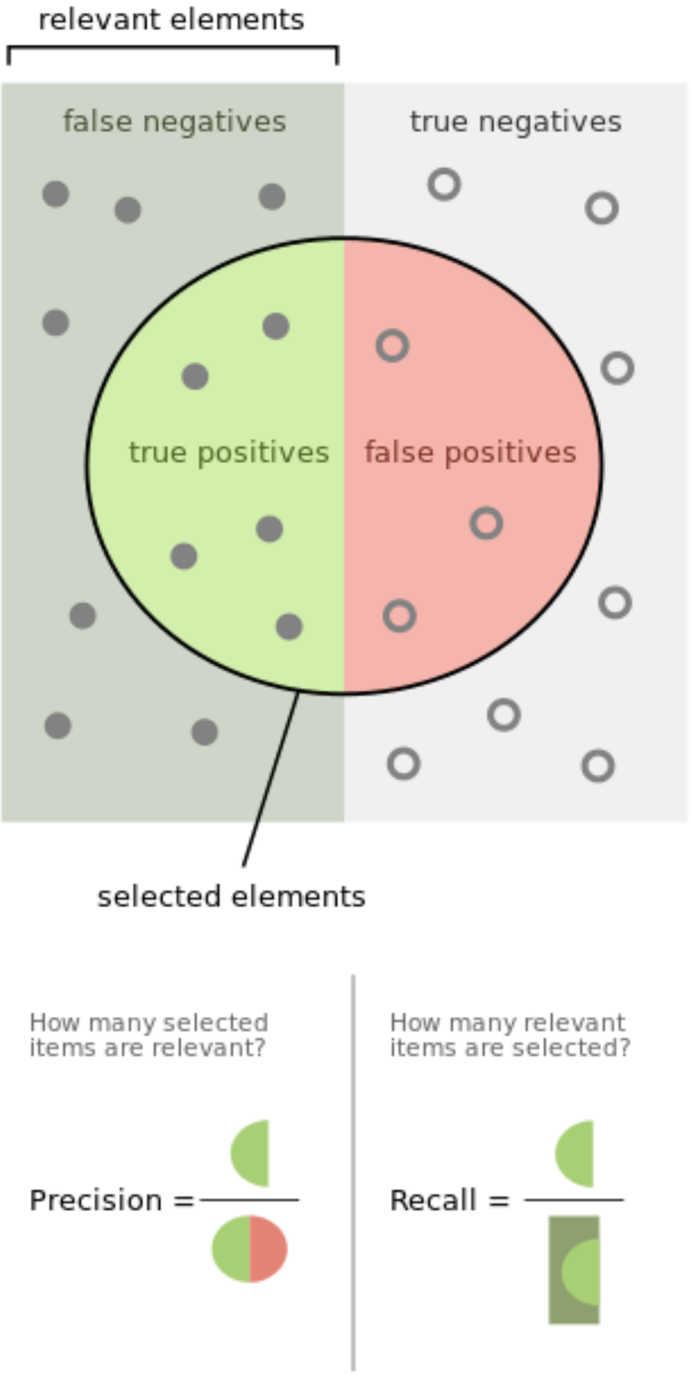
$$\text{Precision (P)} = \frac{A}{A+C}$$

$$\text{Recall } (R) = \frac{A}{A+B}$$

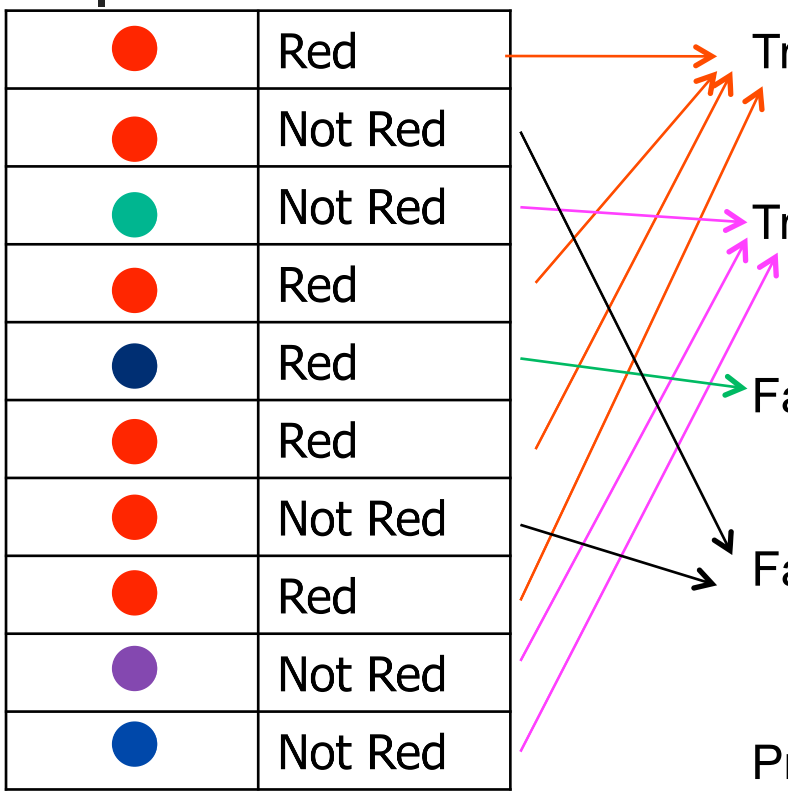
A: TP (true positive) B: FN (false negative)

C: FP (false positive) D: TN (true negative)

		Prediction	
		Salmon	Not Salmon
Actual Class	Salmon	A	B
	Not Salmon	C	D


relevant elements
false negatives
true negatives
true positives
false positives
selected elements
How many selected items are relevant?
Precision =
How many relevant items are selected?
Recall =

An Example



●	Red
●	Not Red
●	Not Red
●	Red
●	Red
●	Red
●	Not Red
●	Red
●	Not Red
●	Not Red

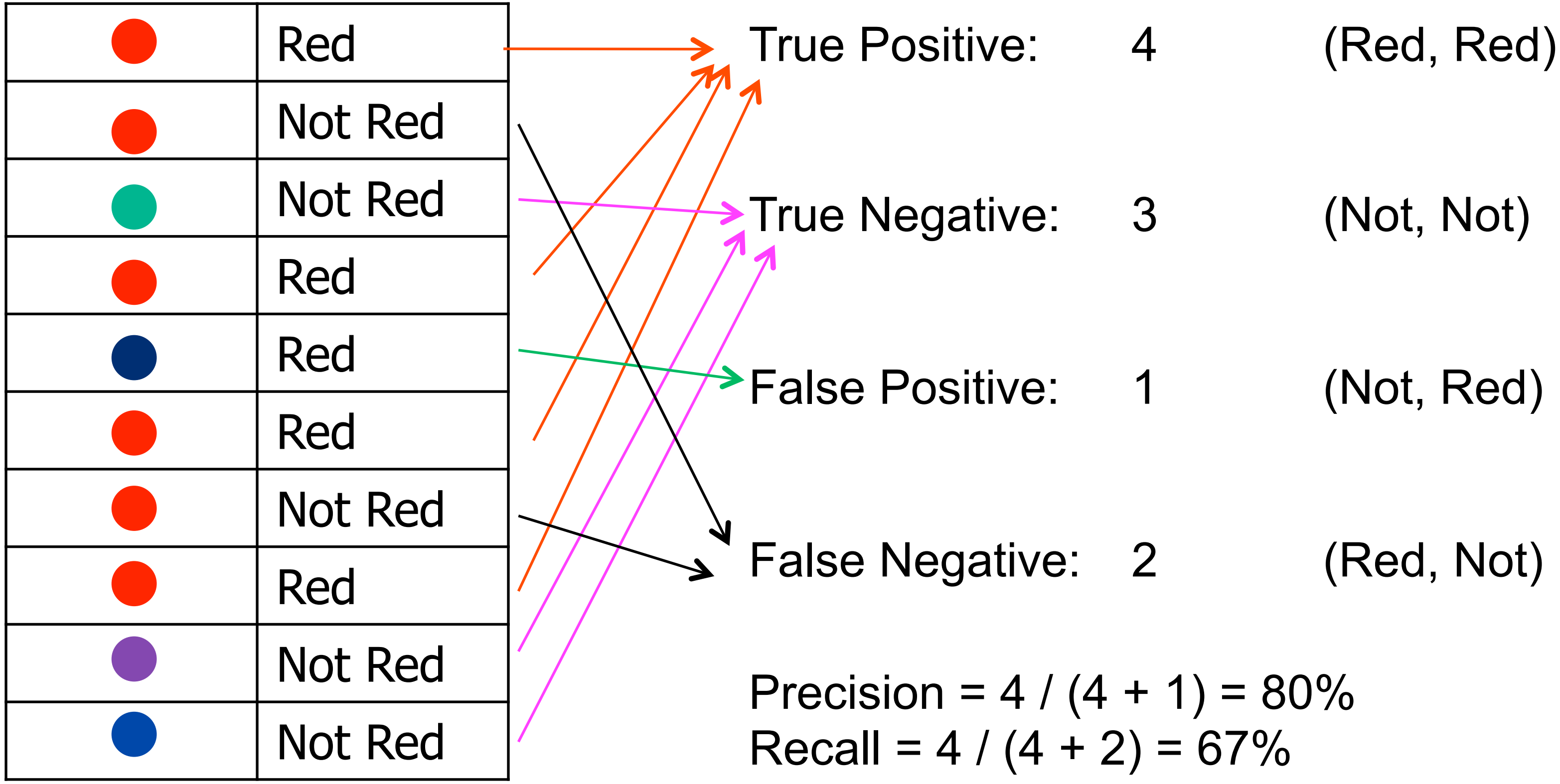
True Positive:	4	(Red, Red)
True Negative:	3	(Not, Not)
False Positive:	1	(Not, Red)
False Negative:	2	(Red, Not)

Precision = ?
Recall = ?

$$Precision, p = \frac{A}{A+C}$$

$$Recall, r = \frac{A}{A+B}$$

An Example



🔴	Red
🔴	Not Red
🟢	Not Red
🔴	Red
🔵	Red
🔴	Red
🔴	Not Red
🔴	Red
🟣	Not Red
🔵	Not Red

True Positive:	4	(Red, Red)
True Negative:	3	(Not, Not)
False Positive:	1	(Not, Red)
False Negative:	2	(Red, Not)

Precision = 4 / (4 + 1) = 80%
Recall = 4 / (4 + 2) = 67%

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?

- Methods for Performance Evaluation
  - How to obtain reliable estimates?
    - how to partition the data?

Methods of Estimation (I)

- **Holdout**
  - Randomly take 70% of the examples as training and the remaining 30% as testing
  - Repeat for several times (e.g. 10)
  - used for data set with large number of samples

Methods of Estimation (II)

- **Cross validation**
  - Randomly partition the data into k mutually exclusive subsets (D_1,D_2, .., D_k), each approximately equal size
  - At i-th iteration, use D_i as test set and others as training set
  - for data set with moderate size

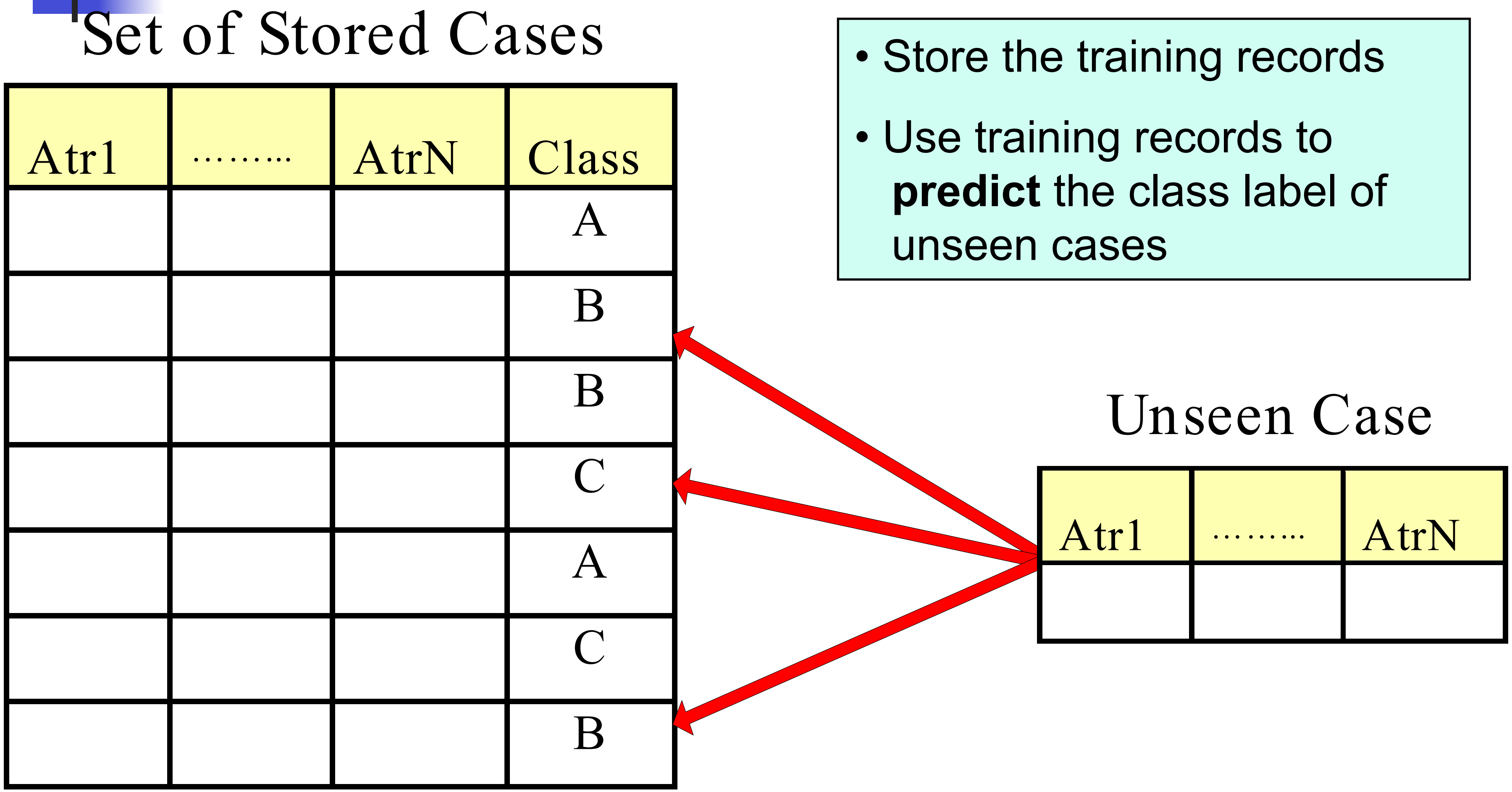
Classification Algorithms

Classification Algorithms

- **Nearest Neighbor**
- Naïve Bayes
- Decision Tree
- …

Instance-Based Classifiers

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



Set of Stored Cases

Atr1	………	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

Unseen Case

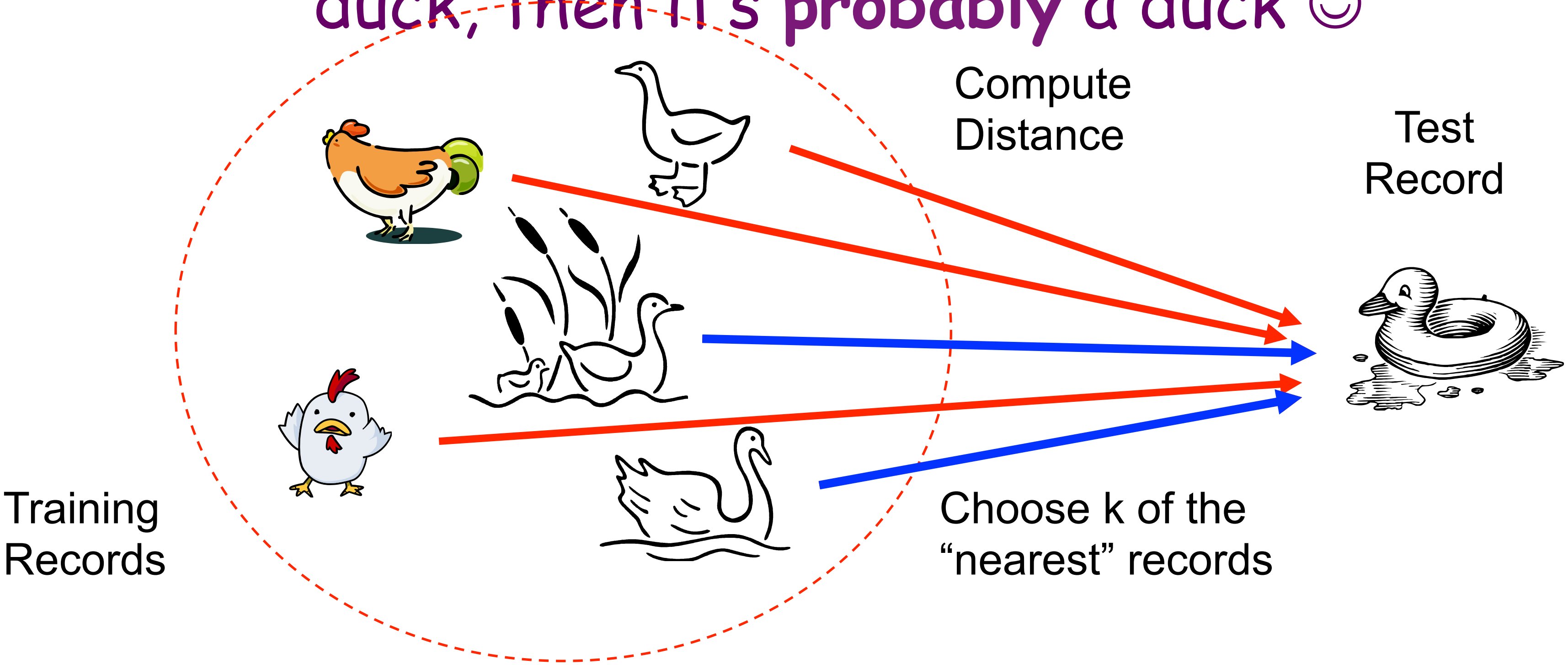
Atr1	………	AtrN

Instance-Based Classifiers

- Examples:
  - Rote-learner
    - Memorizes entire training data and performs classification only if attributes of record **match** one of the training examples exactly
  - **Nearest neighbor**
    - Uses k "closest" points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's **probably** a duck ☺



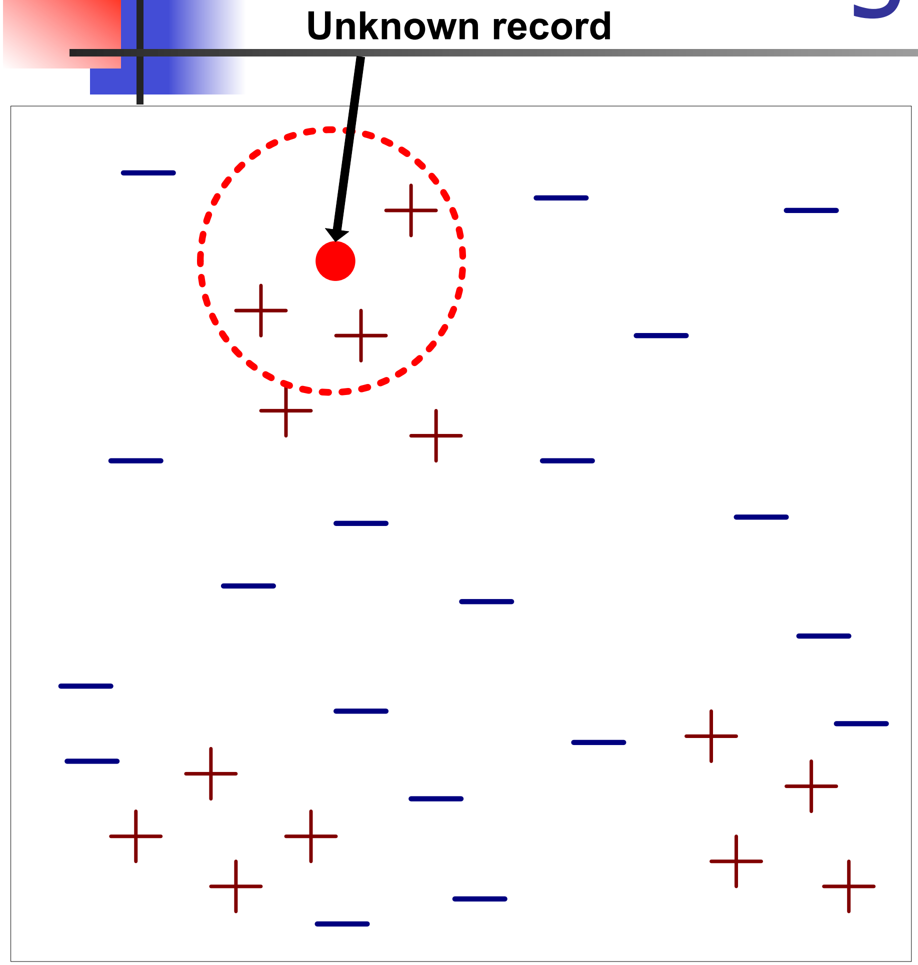
Compute Distance

Test Record

Training Records

Choose k of the "nearest" records

Nearest-Neighbor Classifiers



- Requires three things
  1. The set of stored records
  2. **Distance Metric** to compute distance between records
  3. **The value of *k***, the number of nearest neighbors to retrieve