

Data Mining

– Classification Algorithms (I)



A Motivating Example

- A simple classification problem...
 - I know there are Salmon in this river.
 - When I pick up a fish from this river, can you tell me whether this fish is Salmon?
- Assume that 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 of course...LEARNING!





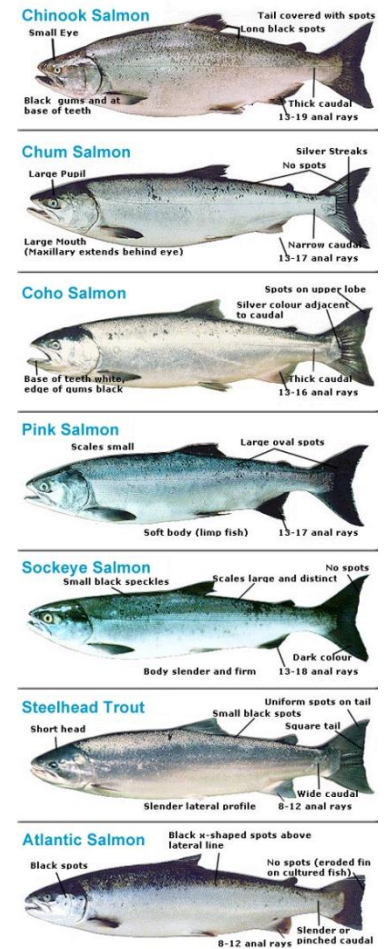
Different Kinds of Learning

- 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 need to identify their characteristics by yourself by observing their features.

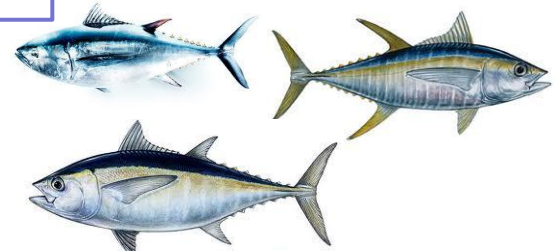


salmon



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

tuna

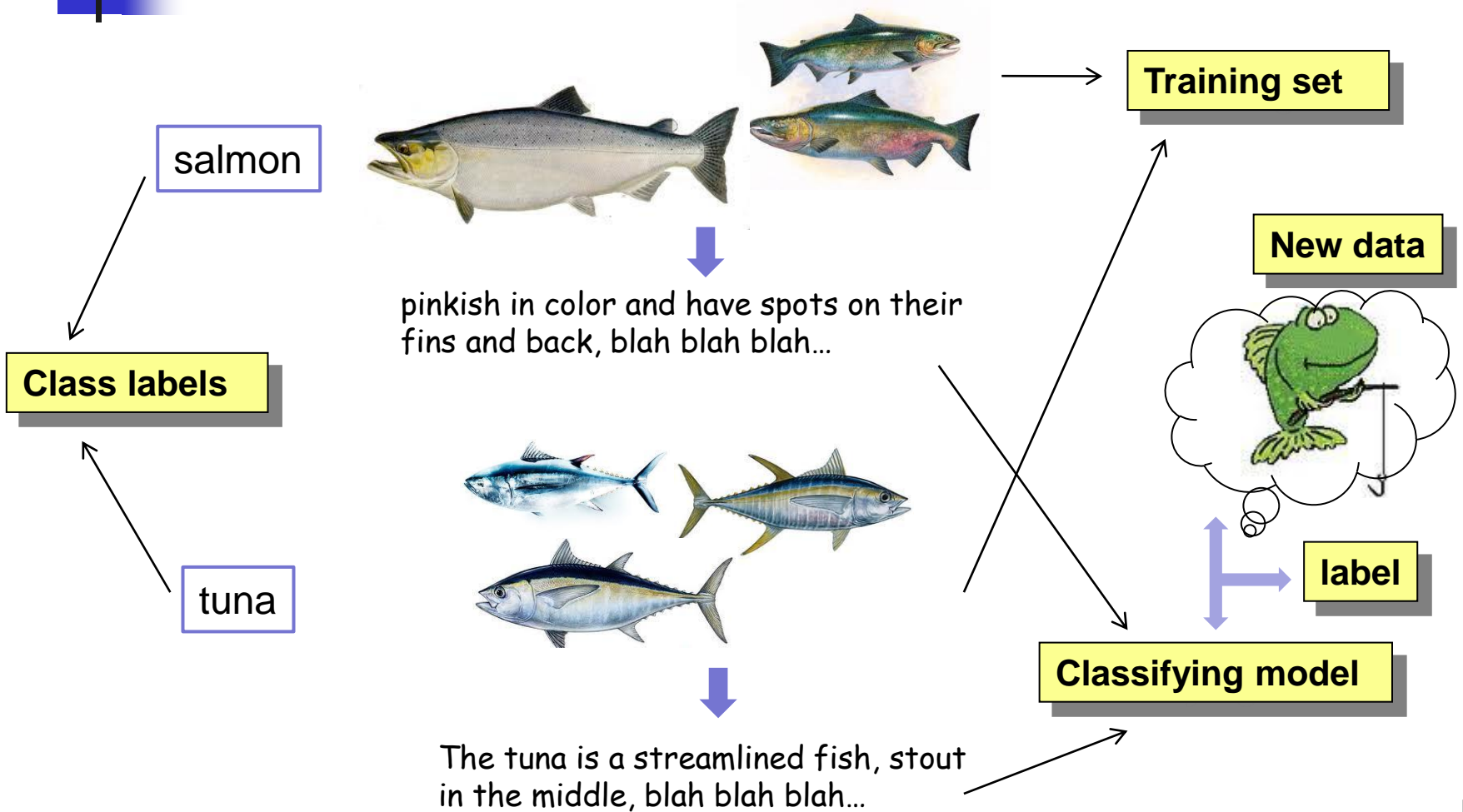
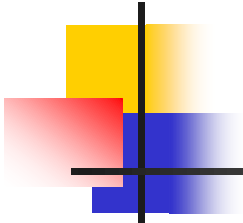


The tuna is a streamlined fish, stout in the middle, blah blah blah...



Classification in Data Mining

- predicts categorical class labels
- classifies data (constructs a model) based on the training set and the values (**class labels**) in a classifying attribute
- uses it in classifying new data





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 computer which of them are Salmon, but do not tell the computer their characteristics.
 - The computer identifies its characteristics by itself.

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

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 have a lot of emails. You must know which of them are spam and which of them are not spam.
 - Yet, can you list ALL characteristics of the spam emails?
 - For active learning, you only need to tell the computer which of them are spam, and which of them are not.
 - The computer identifies their characteristics by itself by observing their differences.
 - So, you save lots of time in fact!

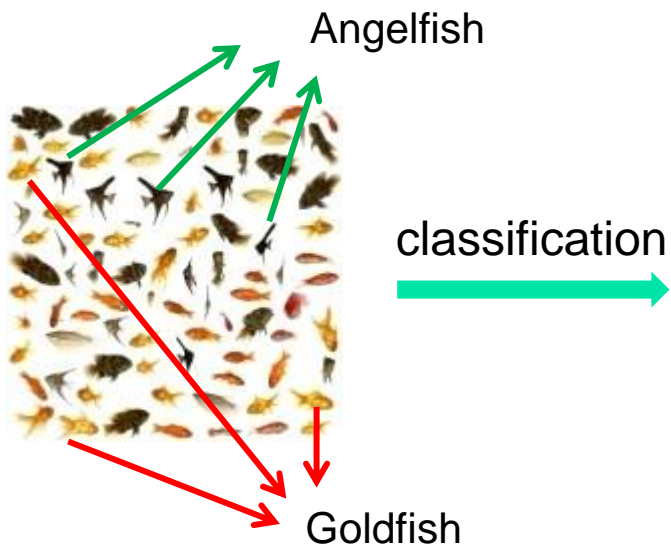
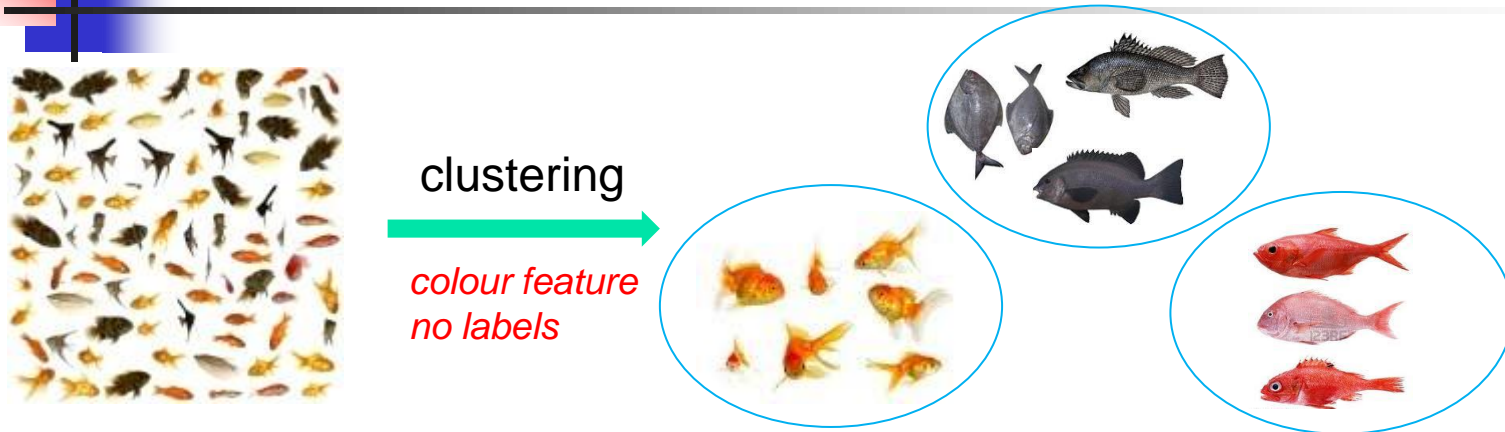




Always Remember...

- From the data mining point of view...
 - Classification \approx Prediction \approx Forecasting
 - This is because the techniques are the same
- Classification is also known as “Supervised Learning”
 - There must be an “expert” (you) to “supervise” the computer.
 - In contrast, Clustering is known as “Unsupervised Learning”. We will discuss it in the later lectures.

Classification vs. Clustering



Angelfish:

Up to 6 inches 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 Process

- Recall:

- You catch a lot of Fish.
- You tell the computer which of them are Salmon.
- The computer identifies their characteristics by itself.

- Terminologies:

- Examples – The fish that you have caught.
- Class – “Salmon” and “Not Salmon”.
- Positive examples – Fish that belong to the class Salmon.
- Negative examples – Fish that do not belong to the class Salmon.
- Model – What the computer has learned. The accuracy of the model depends on the learning algorithm.

Classification—A Two-Step Process



- **Model construction:** describing 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 **training set**
 - The model is represented as classification rules, decision trees, or mathematical formulae
- **Model usage:** for classifying future or unknown objects
 - Estimate accuracy of the model
 - The known label of test sample is compared with the classified result from the model
 - Accuracy rate is the percentage of testing set samples that are correctly classified by the model
 - Testing set is independent of training set
 - 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

1. Archive Training Data

2. Choose an learning algorithm



Model

Learning

Model Evaluation



An unknown fish



Model



Yes (Salmon)



No (Not a Salmon)

Operation



Binary-Class vs. Multi-Class

- Binary-Class Classification

- Only two classes exists.
 - "Salmon" / "Not a Salmon"
 - "Cat" / "Dog"
 - "Sheep" / "Tiger"

- Multi-Class Classification

- More than two classes.
 - "Salmon", "Tuna", "Shark", "Gold Fish"
- Every multi-class classification problem can be solved by formulating a series of binary-class classification model.
 - How?



Binary-Class vs. Multi-Class

- Pay attention when formulating a model!!!
 - When classifying fish...
 - If there are only two kinds of fish ("Salmon" and "Tuna")
 - It can be formulated as a simple binary-class classification problem
 - Trivial...
 - When classifying books...
 - If there are only two kinds of books ("Statistics" and "Algorithm")...
 - If a book can belong to either "Statistics" or "Algorithm" only...
 - A simple binary-class classification problem
 - If a book can belong to both "Statistics" and "Algorithm"...
 - Two binary-class classification problem
 - Parallel structure



Data Preparation

- Data cleaning
 - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
 - Remove the irrelevant or redundant attributes
- Data transformation
 - Generalize and/or normalize data



Major Classification Algorithms

- In this course, we will discuss the following major learning algorithms:
 - Nearest Neighbor
 - Naïve Bayes
 - Decision Tree



Classifier Committee (Ensemble Classifier)

- As the name implies, the decision is made by a set of classifiers.
 - When a task involve expert's judgment, then the decision made by N experts together is usually better than only one, if their decisions are properly combined.
- General idea
 - do not learn a *single classifier* but learn a *set of classifiers*
 - combine the predictions of multiple classifiers

Two Simple Combination Techniques

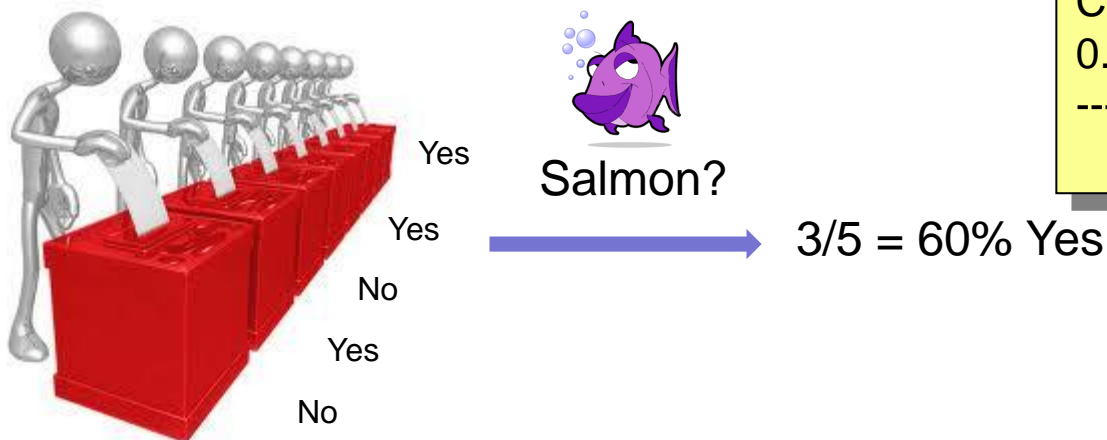
- Majority Vote.

- Simple voting! This strategy always performs surprisingly good!
 - Suppose there are 25 independent classifiers
 - Each classifier has error rate, $\varepsilon = 0.35$
 - Probability that the ensemble classifier makes a wrong prediction:

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

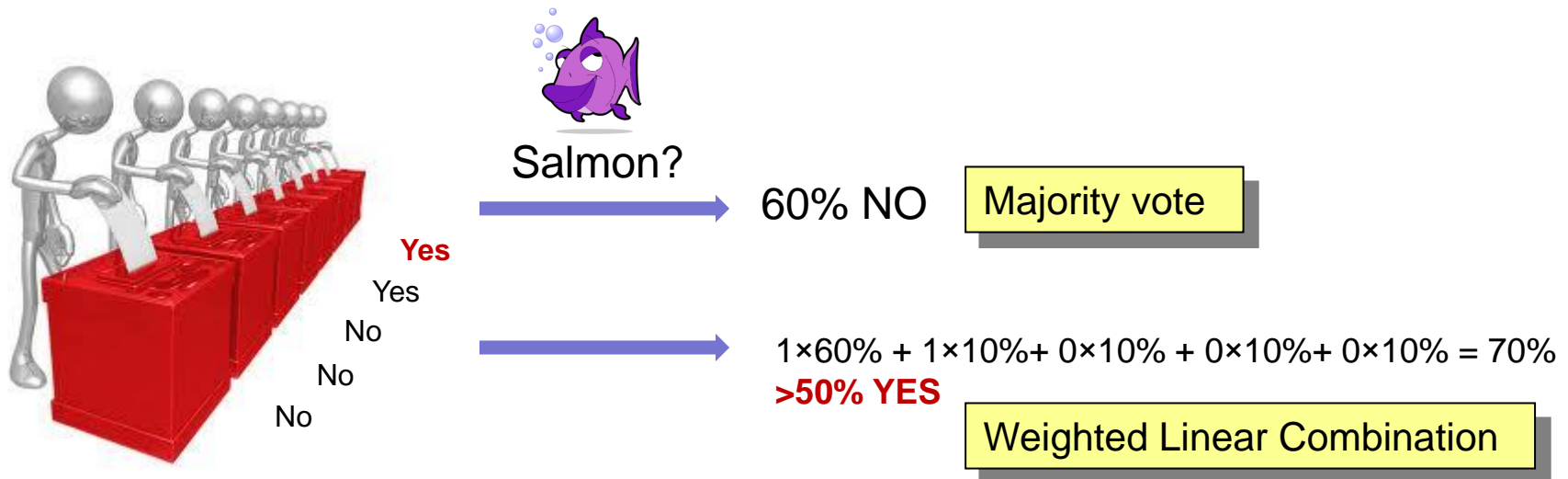
Only 3 classifiers,
 Error rate $\varepsilon = 0.35$
 Case1: 2 classifiers make mistake
 $3 \times 0.35^2 \times 0.65^1 = 0.239$
 Case2: 3 classifiers make mistake
 $0.35^3 = 0.043$

 $\Sigma = 0.282$



Two Simple Combination Techniques (cont')

- Weighted Linear Combination.
 - If a classifier is more reliable, then we value its decision higher.
 - We will discuss how to compute the reliability of a classifier shortly.
 - Usually performs even better than Majority Vote.





Model Evaluation

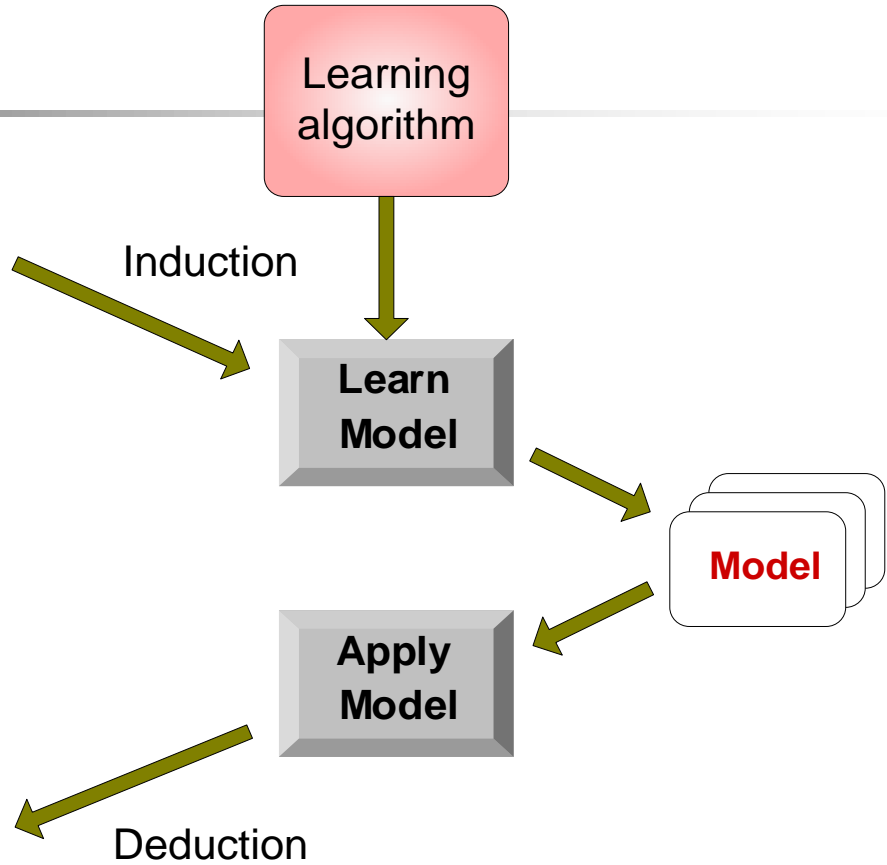
- After Learning and before using the model (operation), we need to test it first!
 - We need to “test” the model to see whether it “really learned” something.
 - To see how good (reliable) the model is.

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	?

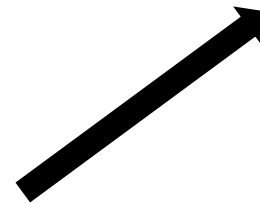
Test Set



Testing

- Prepare the training data and testing data
 - Training Data and Testing Data will NEVER overlapped
 - Why?

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

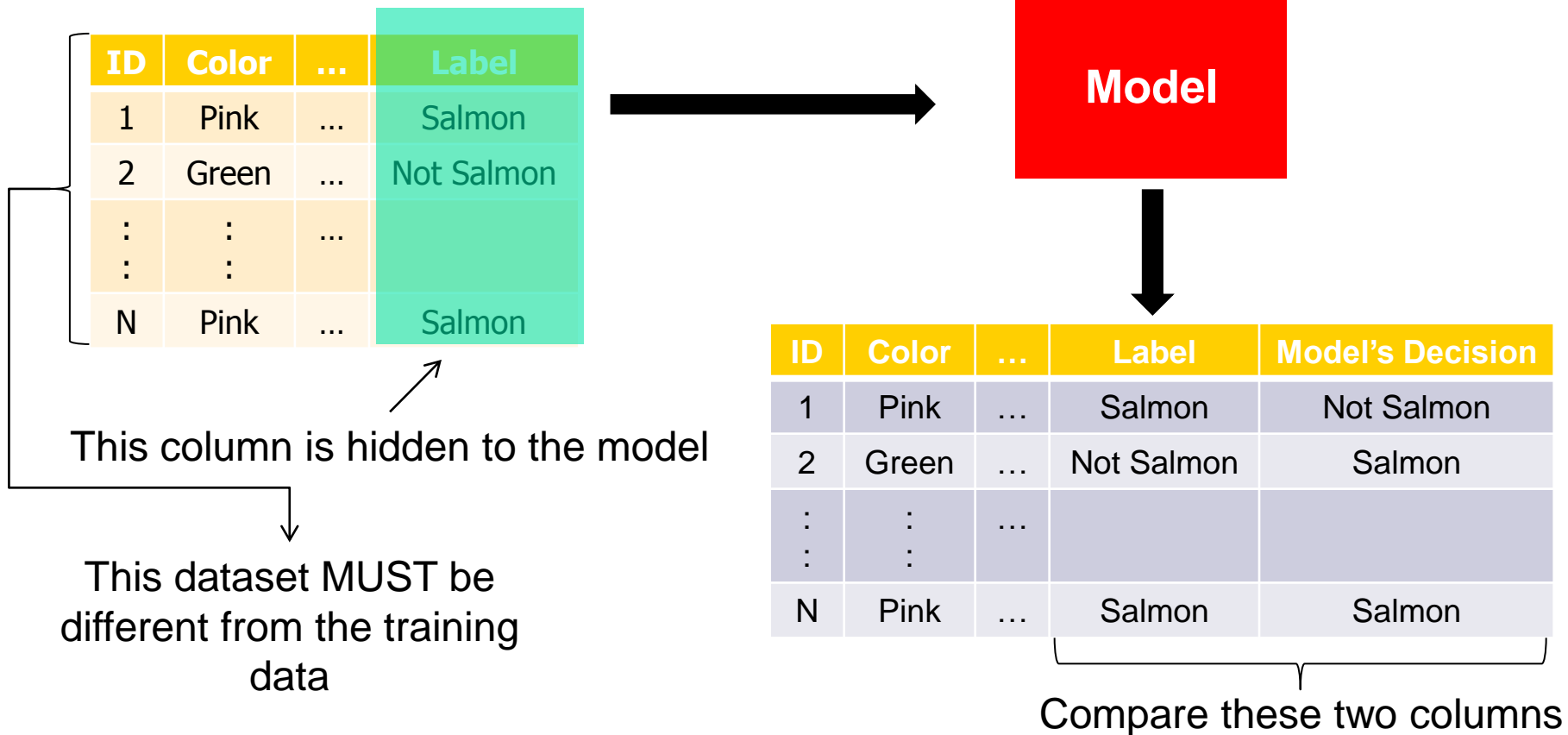
ID	Color	Size	...	Label
2	Green	30cm	...	Not Salmon
6	Grey	12cm	...	Not Salmon
⋮	⋮	⋮	...	⋮
⋮	⋮	⋮	...	⋮
M	Pink	18cm	...	Salmon

Testing Data

We will discuss how to partition shortly in the later slides

Testing

■ Testing process





Model Evaluation

- Metrics for Performance Evaluation
 - How to evaluate the performance of a model?
- Methods for Performance Evaluation
 - How to obtain reliable estimates?



Model Evaluation

- Metrics for Performance Evaluation
 - How to evaluate the performance of a model?
- Methods for Performance Evaluation
 - How to obtain reliable estimates? (i.e. how to partitioning the data?)

Metrics for 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)

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



Limitation of Accuracy

- Consider...
 - Total number of fish in the testing examples = 10,000
 - Number of Non-Salmon = 9990
 - Number of Salmon = 10
- If model predicts everything to be class non-salmon, the accuracy is $9990/10000 = 99.9\%$!!!
 - Accuracy is misleading because model cannot detect any Salmon.

Precision, Recall and F-Measure

- Measuring the quality (effectiveness) of the model:

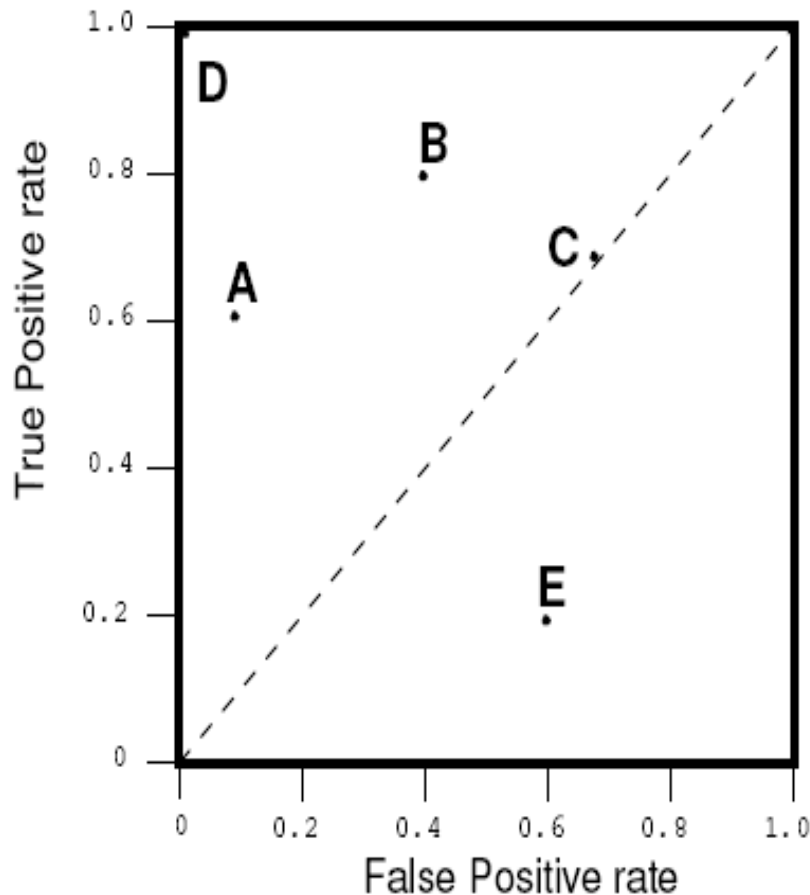
$$\text{Precision, } p = \frac{A}{A + C}$$

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

$$\text{F-measure} = \frac{2rp}{r + p}$$

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

Evaluation of Classifiers: Receiver Operating Characteristics (ROC)



$$tp \text{ rate} \approx \frac{\text{Positives correctly classified}}{\text{Total positives}}$$

$$fp \text{ rate} \approx \frac{\text{negatives incorrectly classified}}{\text{total negatives}}$$

Point (0,0) issues no positive classifications.

Point (1,1) issues positive classifications all the time.

Point D (0,1) is perfect.

Point (1,0) is the worst but can be reversed.

Point C (on the line of $x=y$) indicates a random guess.

So good classifiers appear at the upper left area of this graph.

A basic ROC graph showing five discrete classifiers.



Model Evaluation

- Metrics for Performance Evaluation
 - How to evaluate the performance of a model?
- Methods for Performance Evaluation
 - How to obtain reliable estimates? (i.e. 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 the above procedure for several times (e.g. 10)
- used for data set with large number of samples



Methods of Estimation (II)

- Cross validation
 - Partition data into k disjoint subsets
 - Train on $(k-1)$ partitions, test on the remaining one
 - Repeat for all different combinations
 - for data set with moderate size



Methods of Estimation (III)

- Leave-one-out estimation
 - Assume we have N examples.
 - Take $(N-1)$ examples as training, and the last one as testing
 - Repeat the experiment N times.
 - for small size data



Moreover...

- Speed
 - time to construct the model
 - time to use the model
- Robustness
 - handling noise and missing values
- Scalability
 - availability of model construction on large amounts of data
- Interpretability:
 - understanding and insight provided by the model
- Goodness of rules
 - decision tree size
 - compactness of classification rules

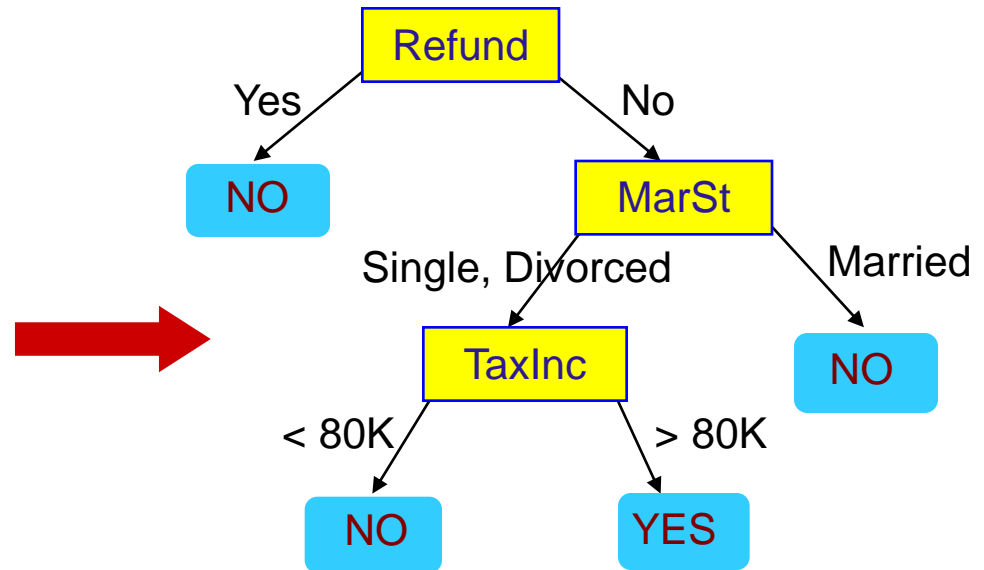


Classification Visualization

Example of a Decision Tree

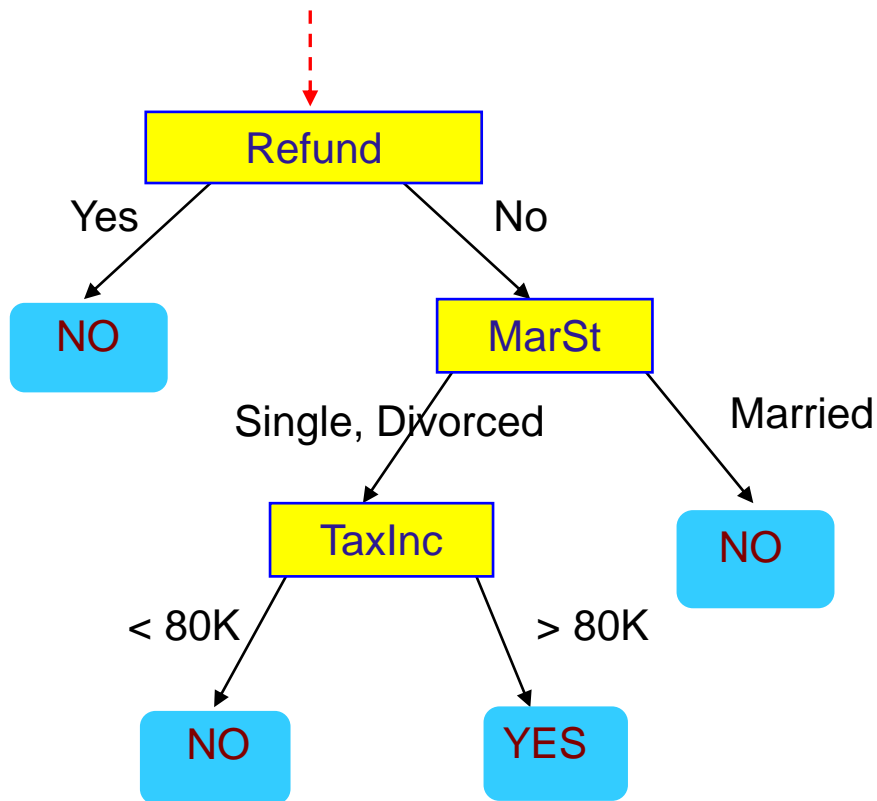
<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

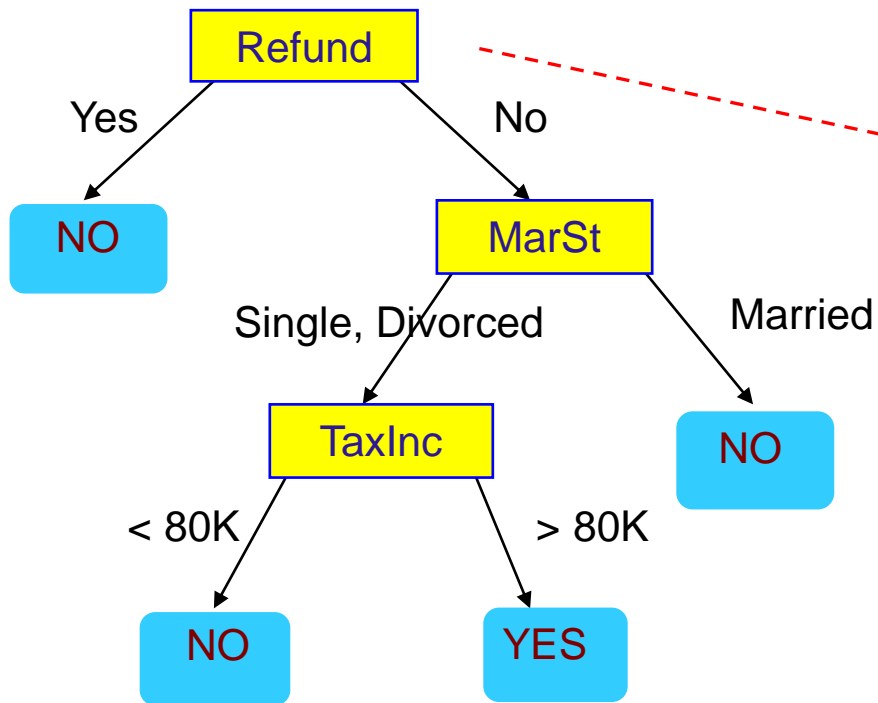
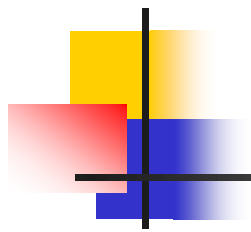


Model: Decision Tree

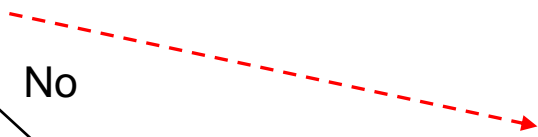
Start from the root of tree.

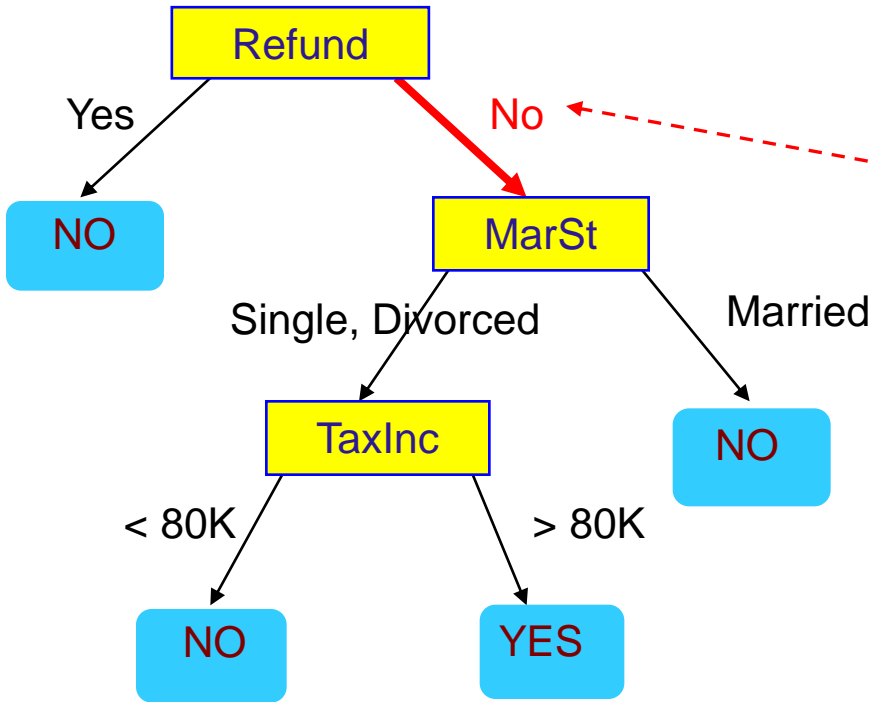
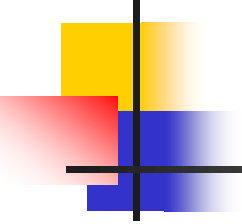


Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

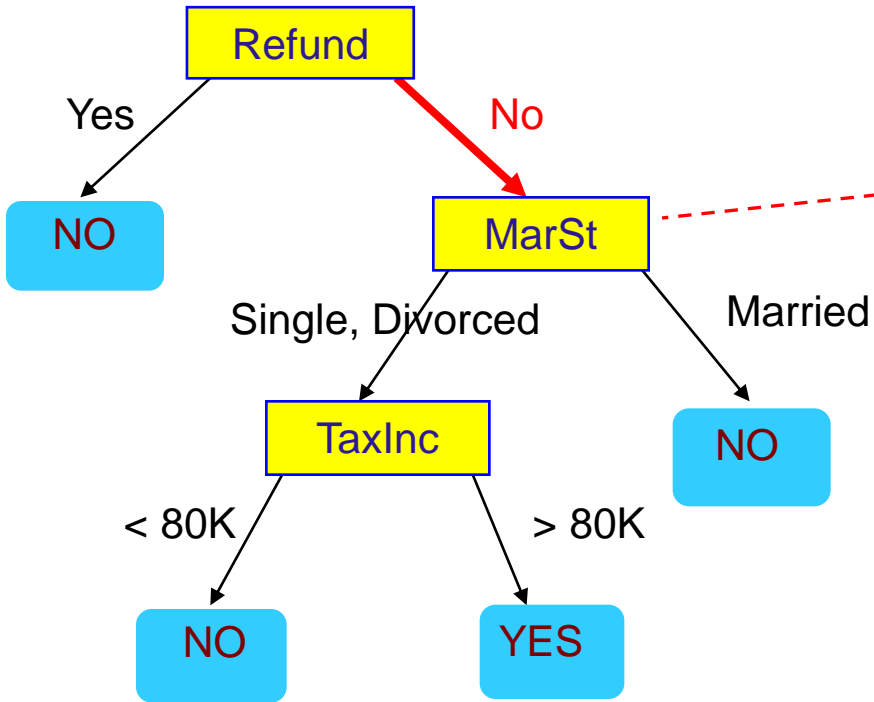
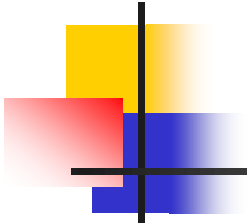


Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

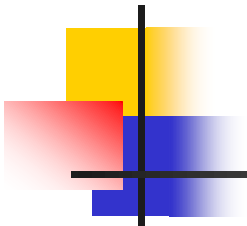




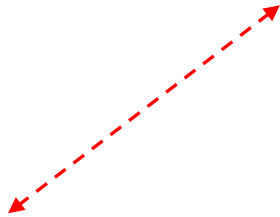
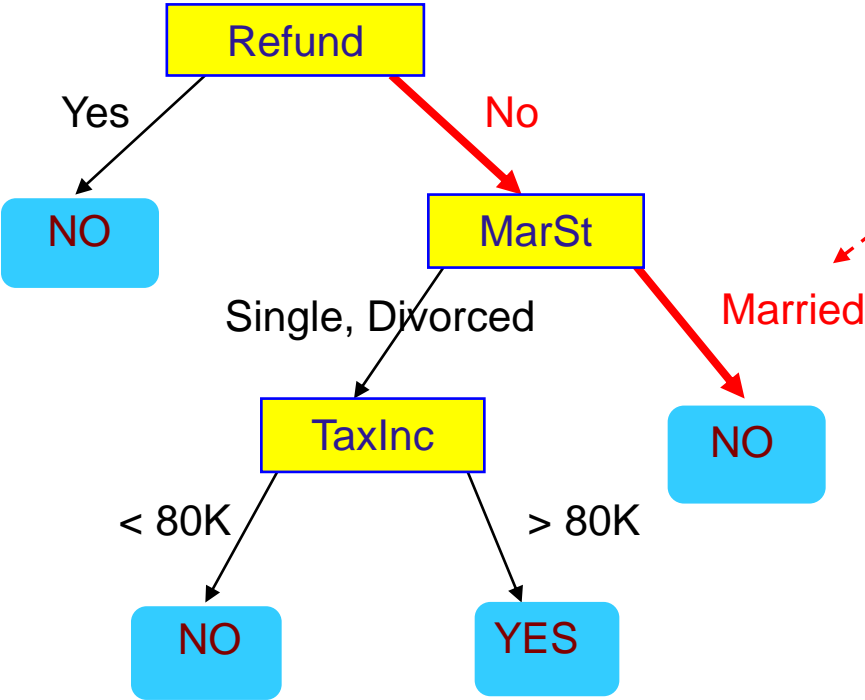
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

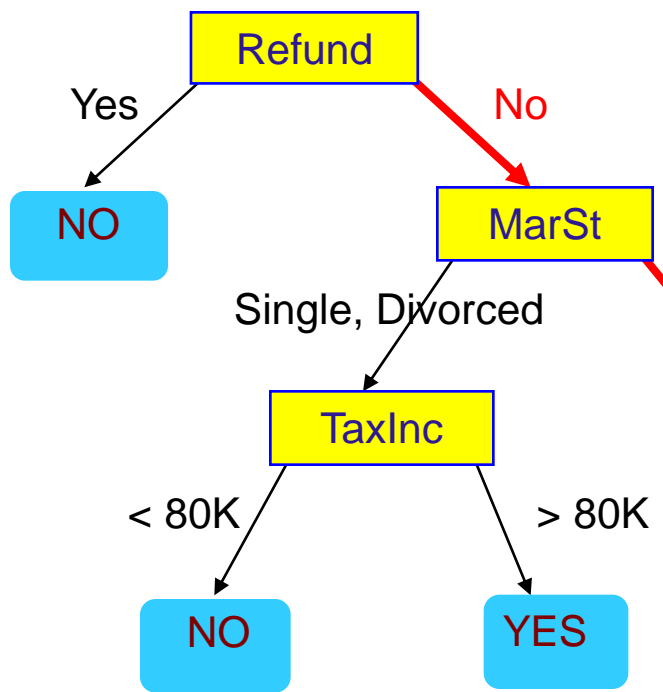
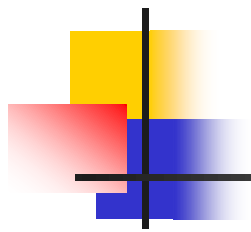


Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

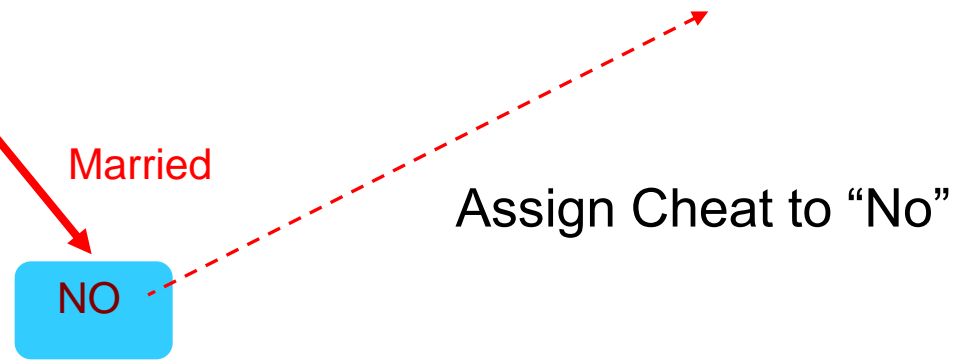


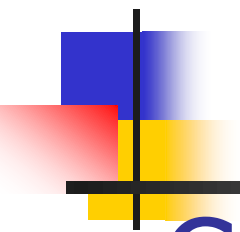
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?





Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



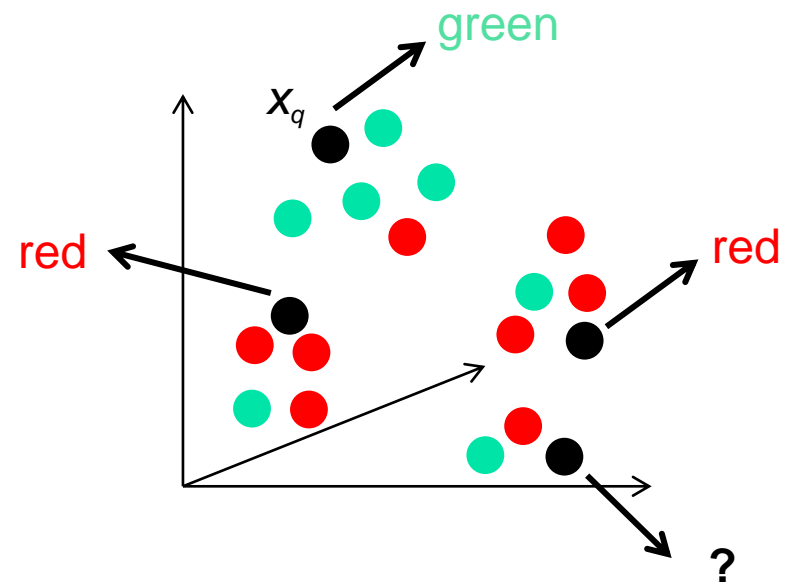
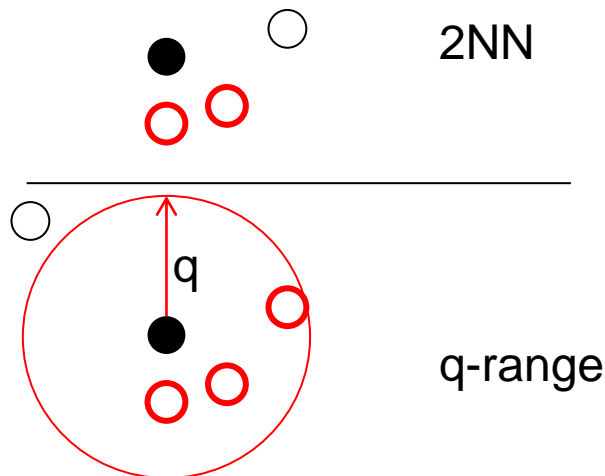


Classification Algorithms

K-Nearest Neighbour Algorithm

- All instances correspond to points in the high-d space
- Find the nearest neighbors of the test example
- Return the most common value among the k training examples nearest to x_q
- Lazy Learning

○





K-Nearest Neighbour

- Lazy vs. eager learning
 - **Lazy learning** (e.g., instance-based learning): Simply stores training data (or only minor processing) and waits until it is given a test tuple
 - **Eager learning** (the above discussed methods): Given a set of training tuples, constructs a classification model before receiving new (e.g., test) data to classify
- Lazy: less time in training but more time in predicting
- Accuracy
 - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form an implicit global approximation to the target function
 - Eager: must commit to a single hypothesis that covers the entire instance space

Discussion on the k -NN Algorithm

- k -NN for real-valued prediction for a given unknown tuple
 - Returns the mean values of the k nearest neighbors
- Distance-weighted nearest neighbor algorithm
 - Weight the contribution of each of the k neighbors according to their distance to the query x_q
 - Give greater weight to closer neighbors $w \equiv \frac{1}{d(x_q, x_i)^2}$
- Robust to noisy data by averaging k -nearest neighbors
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant attributes
 - To overcome it, axes stretch or elimination of the least relevant attributes



Bayesian Theorem: Basics

- Let \mathbf{X} be a data sample (“*evidence*”): class label is unknown
 - Let H be a *hypothesis* that X belongs to class C
 - Classification is to **determine** $P(H|\mathbf{X})$, (*posteriori probability*), the probability that the hypothesis holds given the observed data sample \mathbf{X}
-
- $P(H)$ (*prior probability*), the initial probability
 - E.g., \mathbf{X} will buy computer, regardless of age, income, ...
 - $P(\mathbf{X})$: probability that sample data is observed
 - $P(\mathbf{X}|H)$ (*likelihood*), the probability of observing the sample \mathbf{X} , given that the hypothesis holds
 - E.g., Given that \mathbf{X} will buy computer, the prob. that X is 31..40, medium income



Bayesian Theorem

- Given training data \mathbf{X} , *posteriori probability of a hypothesis* H , $P(H|\mathbf{X})$, follows the Bayes theorem

$$P(H | \mathbf{X}) = \frac{P(\mathbf{X} | H)P(H)}{P(\mathbf{X})} = P(\mathbf{X} | H) \times P(H) / P(\mathbf{X})$$

- Informally, this can be written as
posteriori = likelihood x prior/evidence
- Predicts \mathbf{X} belongs to C_i iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the k classes
- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Towards Naïve Bayesian Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n -D attribute vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$
- Suppose there are m classes C_1, C_2, \dots, C_m .
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since $P(\mathbf{X})$ is constant for all classes, only $P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$ needs to be maximized

Derivation of Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^n P(x_k | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \dots \times P(x_n | C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution
- If A_k is categorical, $P(x_k | C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_i, D|$ (# of tuples of C_i in D)
- If A_k is continuous-valued, $P(x_k | C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

and $P(x_k | C_i)$ is

$$P(\mathbf{X} | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

Not required

Naïve Bayesian Classifier: Training Dataset

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data sample

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayesian Classifier: An Example

- $P(C_i)$: $P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$
 $P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$

2 classes: Yes / No
prior

- Compute $P(X|C_i)$ for each class

$$P(\text{age} = \text{"<=30"} \mid \text{buys_computer} = \text{"yes"}) = 2/9 = 0.222$$

$$P(\text{age} = \text{"<= 30"} \mid \text{buys_computer} = \text{"no"}) = 3/5 = 0.6$$

$$P(\text{income} = \text{"medium"} \mid \text{buys_computer} = \text{"yes"}) = 4/9 = 0.444$$

$$P(\text{income} = \text{"medium"} \mid \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$$

$$P(\text{student} = \text{"yes"} \mid \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{student} = \text{"yes"} \mid \text{buys_computer} = \text{"no"}) = 1/5 = 0.2$$

$$P(\text{credit_rating} = \text{"fair"} \mid \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit_rating} = \text{"fair"} \mid \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$$

likelihood

- **$X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$**

$$P(X|C_i) : P(X|\text{buys_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$

$$P(X|\text{buys_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

posteriori

$$P(X|C_i) * P(C_i) : P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) = 0.028$$

$$P(X|\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) = 0.007$$

Therefore, X belongs to class ("buys_computer = yes")

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income=medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
 - *Adding 1 to each case*
 - Prob(income = low) = 1/1003
 - Prob(income = medium) = 991/1003
 - Prob(income = high) = 11/1003
 - The “corrected” prob. estimates are close to their “uncorrected” counterparts

Naïve Bayesian Classifier: Comments



- Advantages
 - Easy to implement
 - Good results obtained in most of the cases
- Disadvantages
 - Assumption: class conditional independence, therefore loss of accuracy
 - Practically, dependencies exist among variables
 - E.g., hospitals: patients: Profile: age, family history, etc.
Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
 - Dependencies among these cannot be modeled by Naïve Bayesian Classifier



In this Lecture

- General procedure for constructing a classification model (classifier).
 - Partition the data into Training and Testing.
 - Train the classifier.
 - Test the classifier before using it.
 - Accuracy is not appropriate
 - Precision, Recall, F-Measure
- Combine the decisions of different classifiers
 - Majority vote
 - Linear Weighted Combination
- Binary-class vs. Multi-class
- Nearest Neighbor Algorithm
- Naïve Bayesian Classifier



In Summary...

- Classification is an **extensively studied** problem (mainly in statistics, machine learning & neural networks)
- Classification is probably one of the most **widely used** data mining techniques with a lot of extensions
- **Scalability** is still an important issue for database applications: thus combining classification **with database techniques** should be a promising topic
- Research directions: classification of **non-relational data**, e.g., text, spatial, multimedia, etc..