Introduction to Classification, aka Machine Learning

Classification: Definition

- Given a collection of examples (*training set*)
  - Each example is represented by a set of *features*, sometimes called *attributes*
  - Each example is to be given a label or class
- Find a *model* for the label as a function of the values of features.
- Goal: previously unseen examples should be assigned a label 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.

Supervised vs. Unsupervised Learning

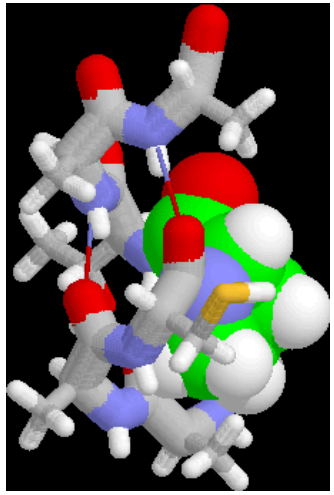
- Supervised learning (**classification)**
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (includes **clustering)**
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Examples of Classification Tasks

- 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

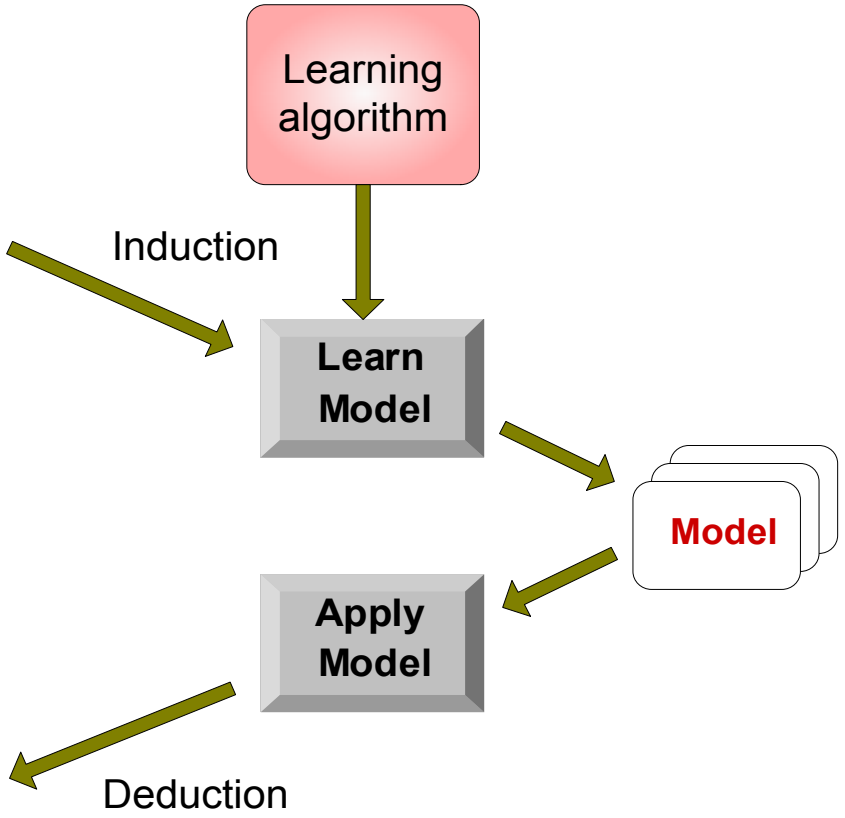
Illustrating Classification Task

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

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



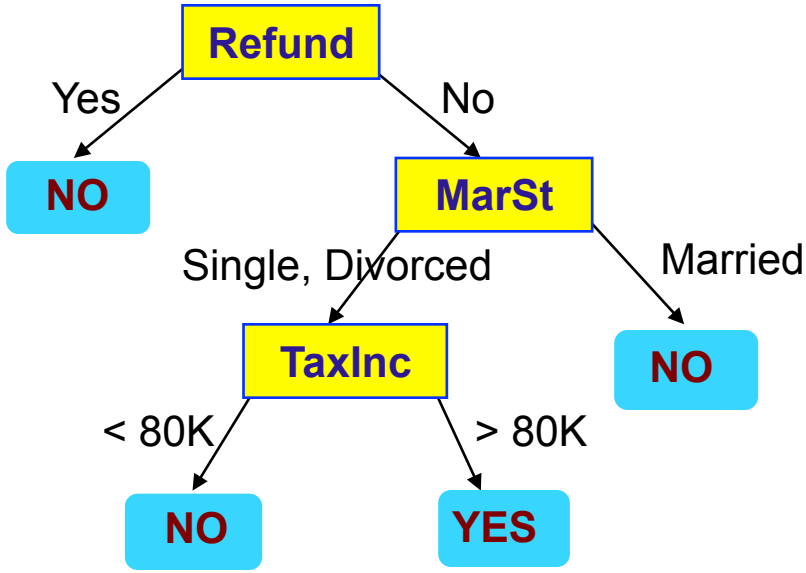
Classification Techniques

- There are a number of different classification algorithms to build a model for classification
  - Decision Tree based Methods
  - Rule-based Methods
  - Memory based reasoning, instance-based learning
  - Neural Networks
  - Genetic Algorithms
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines
- In this introduction, we illustrate classification tasks using Decision Tree methods
- Features can have numeric values (continuous) or a finite set of values (categorical/nominal), including boolean true/false

Example of a Decision Tree

boolean 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

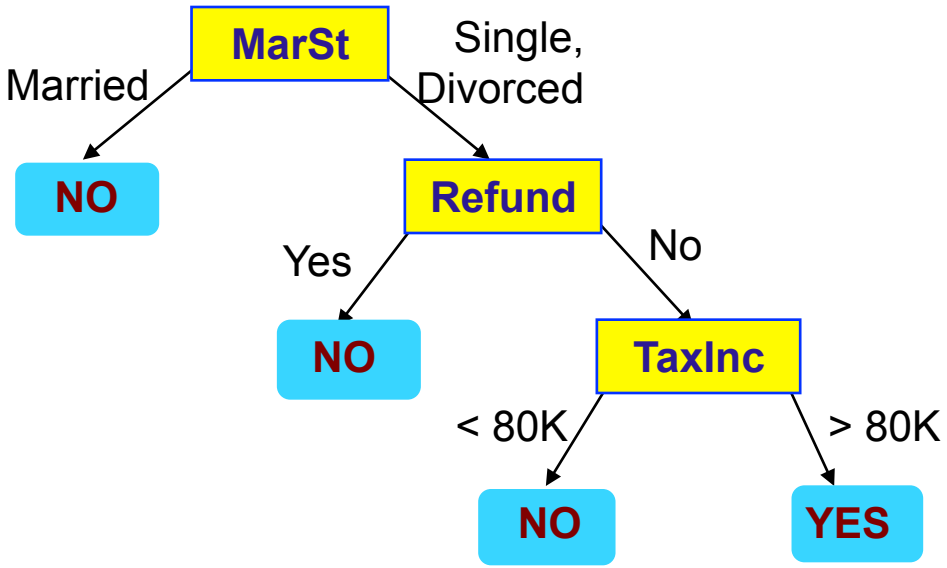
Model: Decision Tree

Example task: Given the marital status, refund status, and taxable income of a person, label them as to whether they will cheat on their income tax.

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



There could be more than one tree that fits the same data!

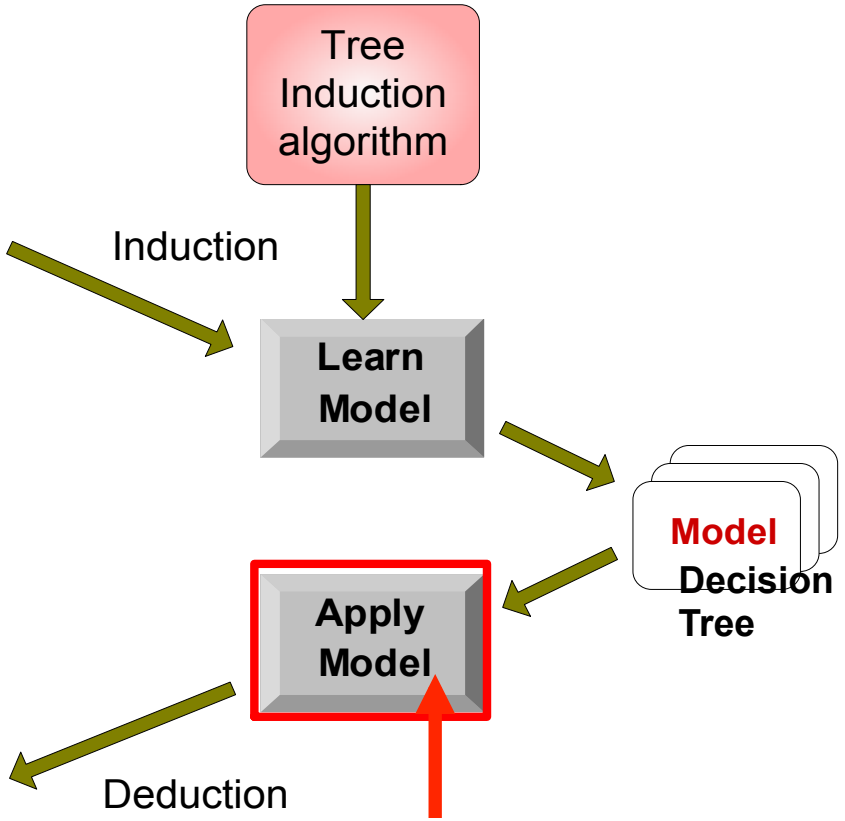
Decision Tree Classification Task

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

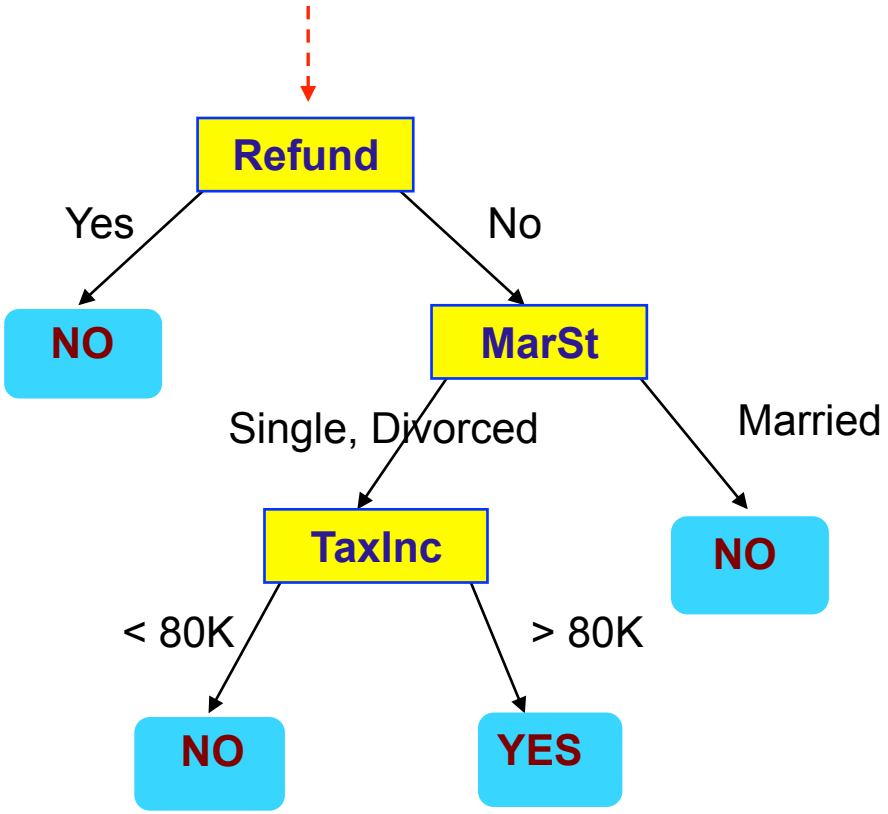
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Apply Model to Test Data

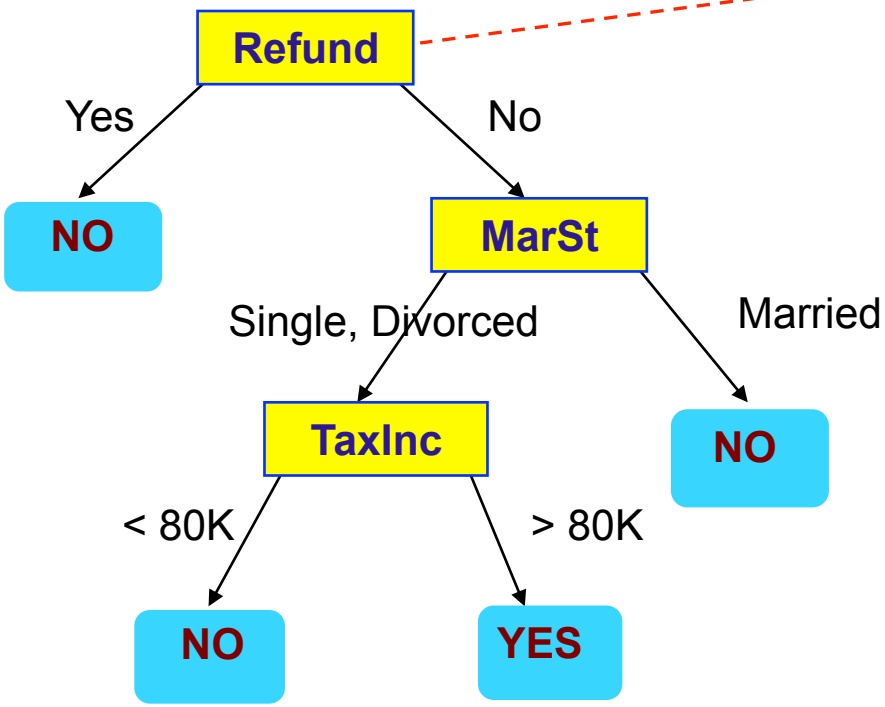
Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

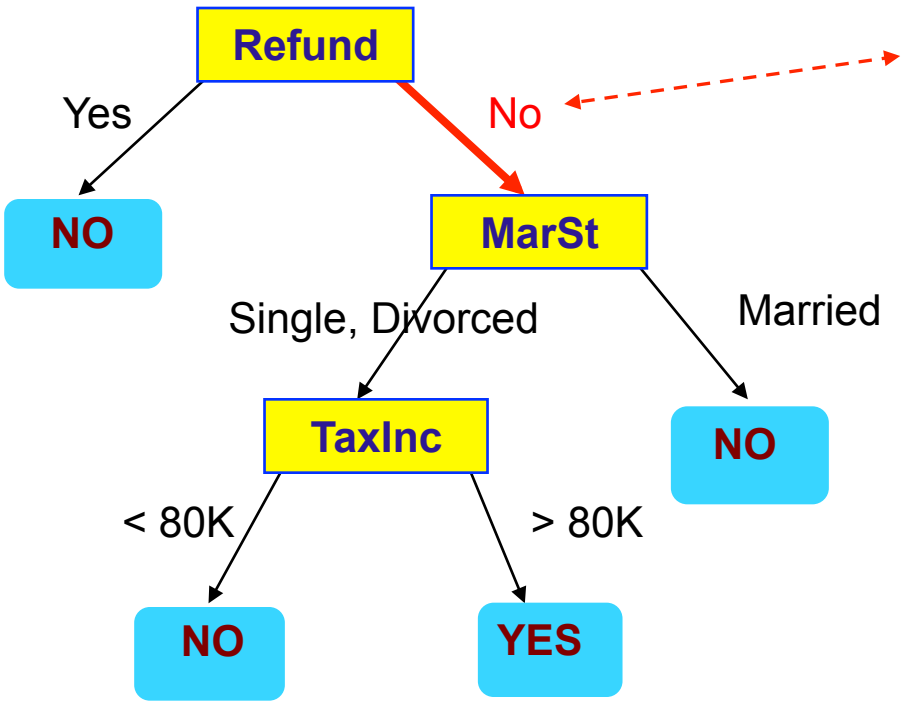
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

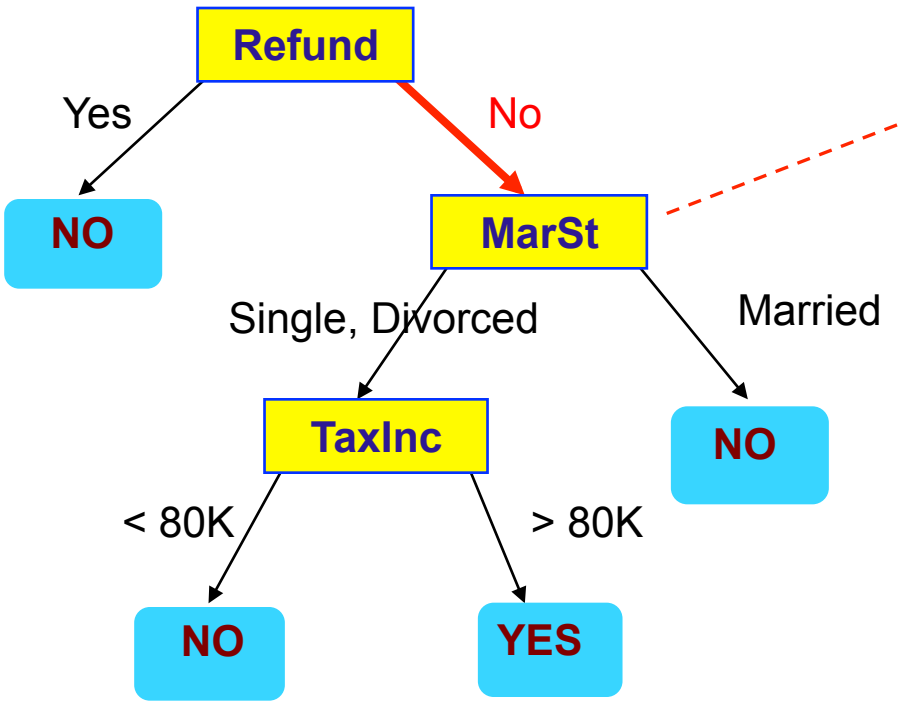
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

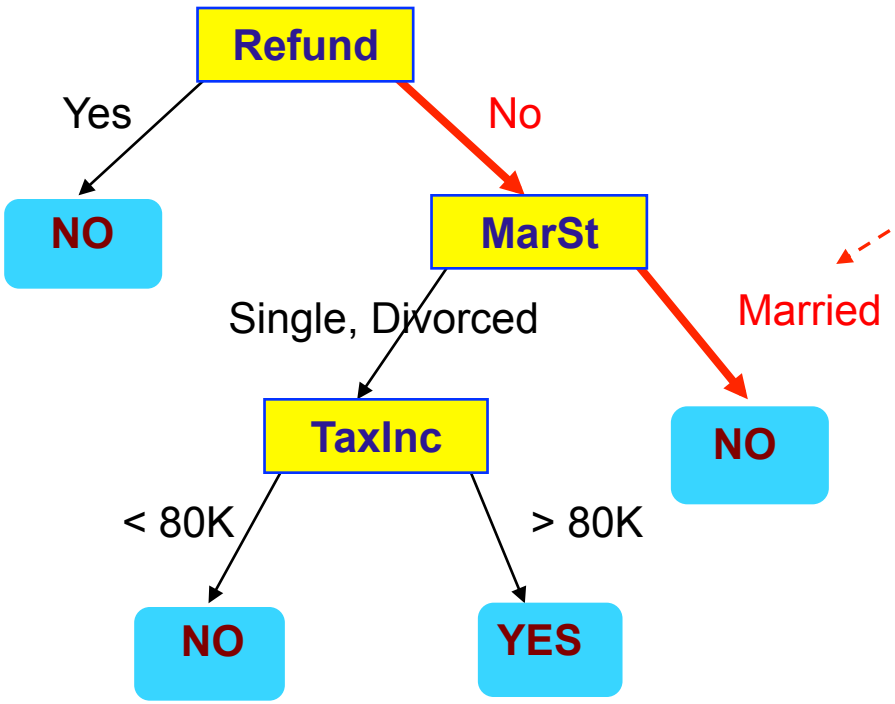
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



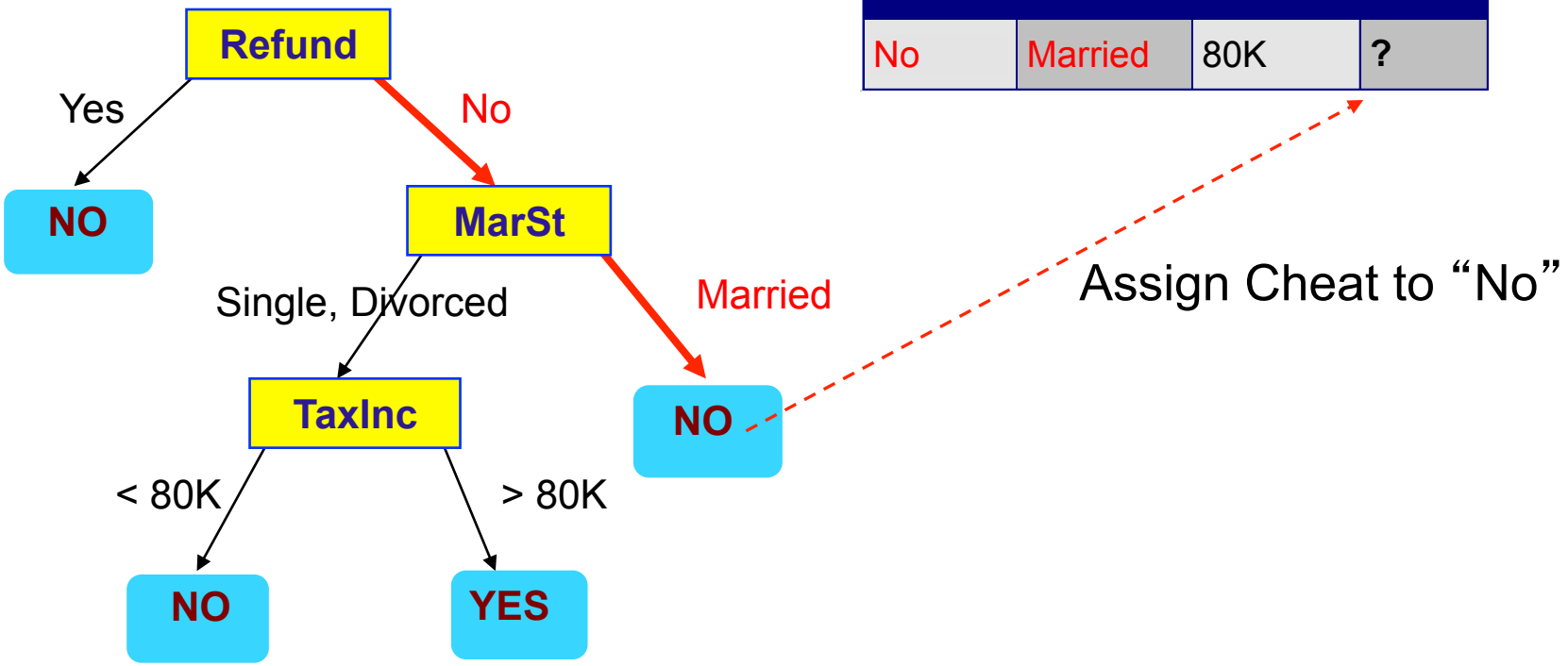
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



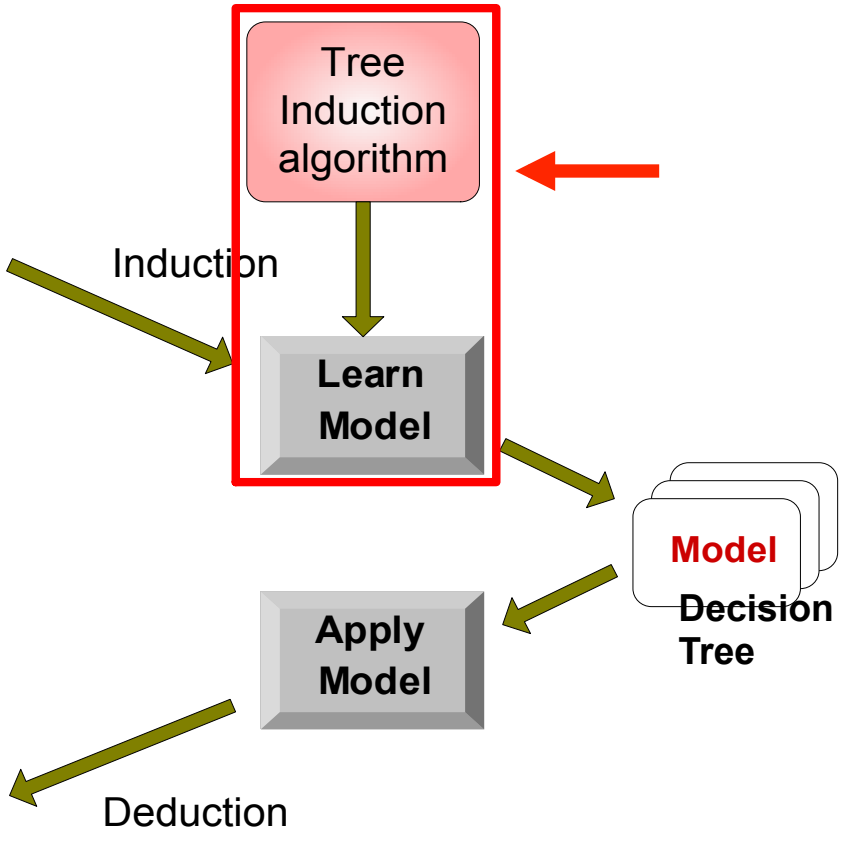
Decision Tree Classification Task

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



Evaluating Classification Methods

- Accuracy
  - classifier accuracy: predicting class label
  - predictor accuracy: guessing value of predicted attribute
- Speed
  - time to construct model (training time)
  - time to use the model (classification/prediction time)
- Robustness: handling noise and missing values
- Scalability: efficiency in disk-resident databases
- Interpretability
  - understanding and insight provided by the model
- Other measures, e.g. goodness of rules, such as decision tree size or compactness of classification model

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 for a binary classifier (two labels):

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

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Classifier Accuracy Measures

- Another widely-used metric: Accuracy of a classifier M is the percentage of test set that are correctly classified by the model M

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

	Yes - C_1	No - C_2
Yes - C_1	a: True positive	b: False negative
No - C_2	c: False positive	d: True negative

classes	buy_computer = yes	buy_computer = no	total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
total	7366	2634	10000

Other Classifier Measures

- Alternative accuracy measures (e.g., for cancer diagnosis or information retrieval)

sensitivity = t-pos/pos /* true positive recognition rate */

specificity = t-neg/neg /* true negative recognition rate */

precision = t-pos/(t-pos + f-pos)

recall = t-pos/(t-pos + f-neg)

accuracy = sensitivity * pos/(pos + neg) + specificity * neg/(pos + neg)

Multi-Class Classification

- Most classification algorithms solve binary classification tasks, while many tasks are naturally multi-class, i.e. there are more than 2 labels
- Multi-Class problems are solved by training a number of binary classifiers and combining them to get a multi-class result
- Confusion matrix is extended to the multi-class case
- Accuracy definition is naturally extended to the multi-class case
- Precision and recall are defined for the binary classifiers trained for each label

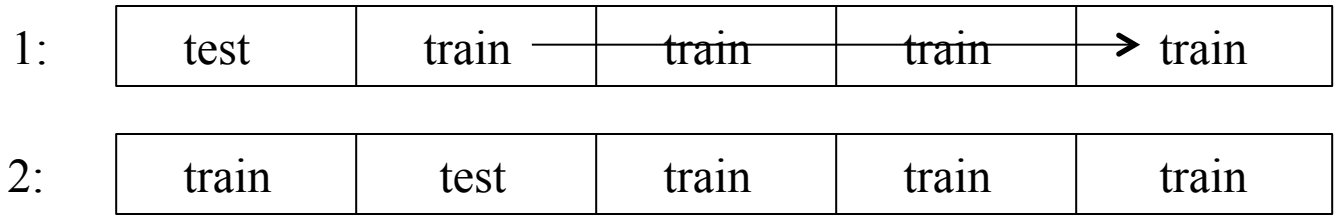
Issues with imbalanced classes

- Consider a 2-class problem with labels Yes and No
  - Number of No examples = 990
  - Number of Yes examples = 10

- If model predicts everything to be No, accuracy is 990/1000 = 99 %
  - Accuracy is misleading because model does not detect any Yes example
  - Precision and recall will be better measures if you are training a classifier to find rare examples.

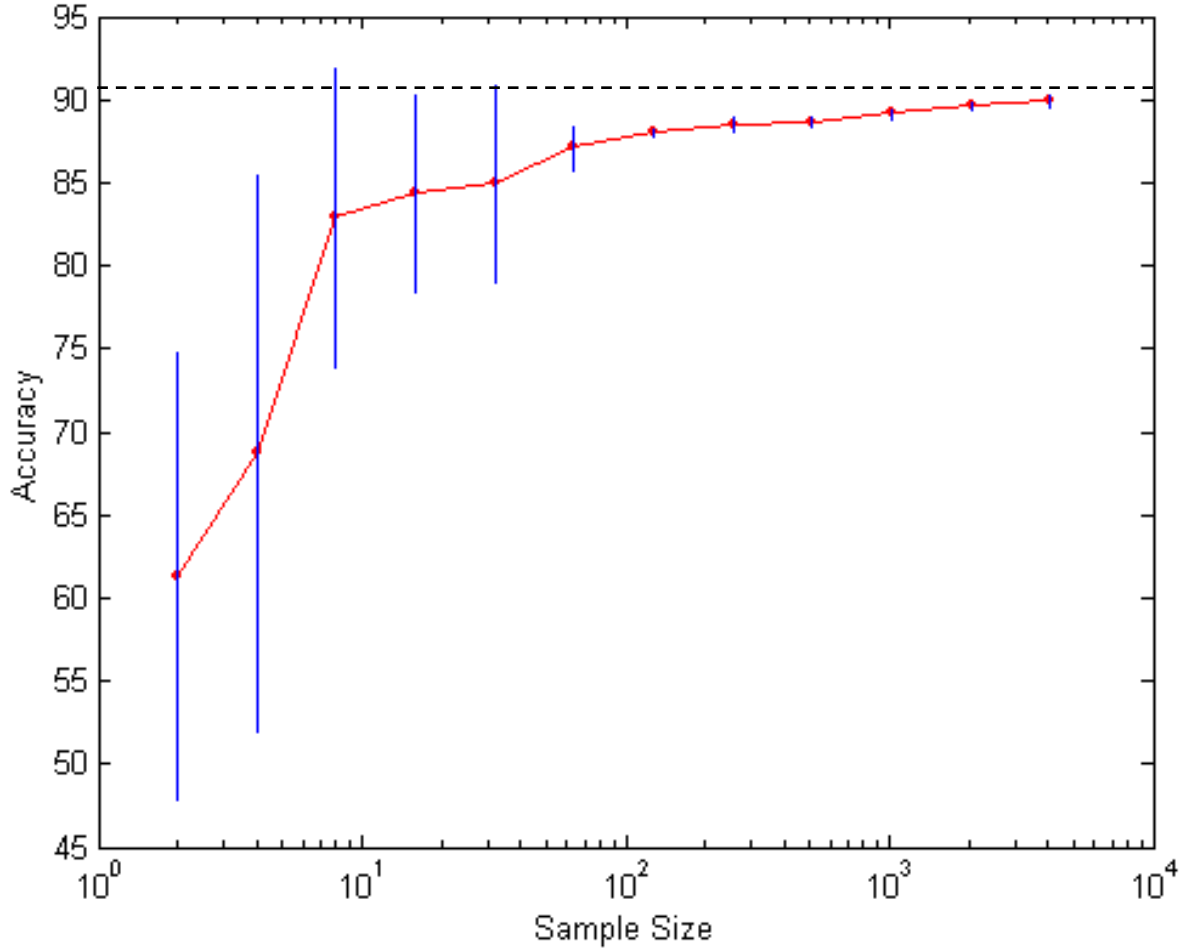
Evaluating the Accuracy of a Classifier

- Holdout method
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
- Cross-validation (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set



. . .

Evaluating the Model - Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve

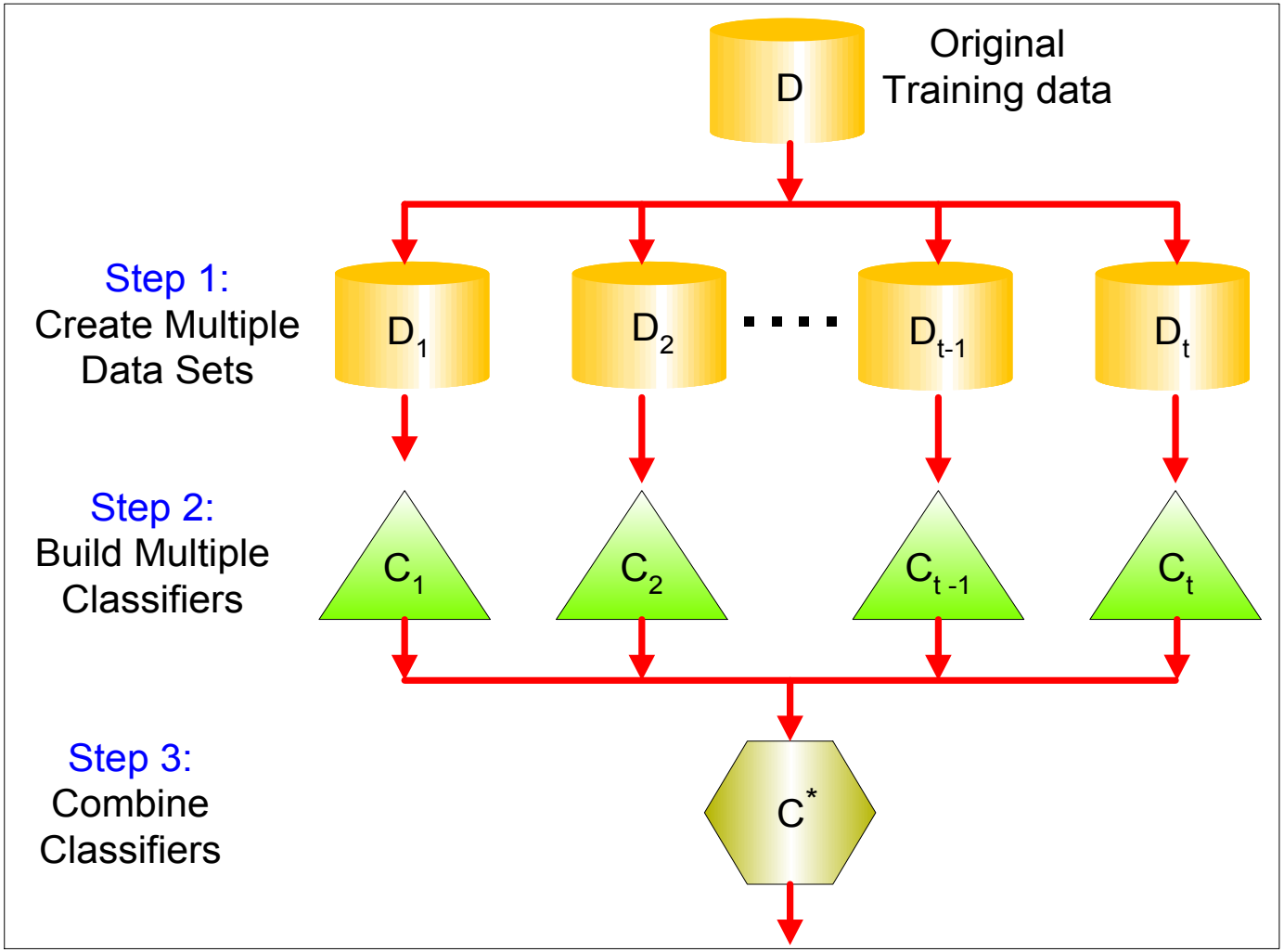
Classifier Performance: Feature Selection

- Too long a training or testing is a performance issue for classification of problems with large numbers of attributes or nominal attributes with large numbers of values
- Feature selection techniques aim to reduce the number of features by finding a smaller or minimal set that can accurately classify the problem
  - reduce the training and prediction time by eliminating noisy or redundant features
- Two main types of techniques
  - Filtering methods apply a statistical or other information measure to the attribute values without running any training and testing
  - Wrapper methods try different combinations of attributes, run cross-validation evaluations and compare the results

Classifier Performance: 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
- Examples of ensemble methods
  - Bagging
  - Boosting
  - Heterogeneous classifiers trained on different feature subsets
    - sometimes called mixture of experts

General Idea



Examples of Classification Problems

- Some NLP problems are widely investigated as supervised classification problems, and use a variety of problem instances
  - Text categorization: assigning topic labels to documents
  - Word Sense Disambiguation: assigning a sense to a word, as it occurs in a document
  - Semantic Role Labeling: assigning semantic roles to phrases in a sentence
- From the NLTK book, chapter 6:
  - Classify first names according to gender
  - Document classification (text categorization)
  - Part-Of-Speech tagging
  - Sentence Segmentation
  - Identifying Dialog Act types
  - Recognizing Textual Entailment

Text Categorization

- Represent each document by the words/tokens/terms it contains
  - Sometimes called unigrams, sometimes bag-of-words
- Identify terms from the document text
  - Remove symbols with little meaning
  - Remove words with little meaning – the stop words
  - Stem the meaningful words
    - Remove endings to get root of the word
      - From *enchanted, enchants, enchantment, enchanting*, get the root word *enchant*
  - Group together words into phrases (optional)
    - Proper names or other words that are likely to have a different meaning as a phrase than the individual words
  - After grouping, may also want to lowercase the terms

Document Features

- Use a feature vector to represent all the words in a document – one position for each word in the collection, representing the weights (often frequency) of words
  - *"Water, water everywhere, and not a drop to drink!"*

water everywhere not drop drink

(2, 1, 1, 1, 1, 0, …) (shown with frequency weights)

- Another document with the word drink:
  - *"drink ..."*

water everywhere not drop drink

(0, 0, 0, 0, 1, 0, …)

- Feature vectors may have thousands of words and are often restricted by a threshold frequency of 5 or more

- Weka demonstration to observe feature vectors
  - Compare
    - Traditional data mining problem
    - Text mining problem