University of Reading

CSMDM21 - Data Analytics and Mining

Classification

Module convenor
Dr. Carmen Lam
carmen.lam@reading.ac.uk
Department of Computer Science

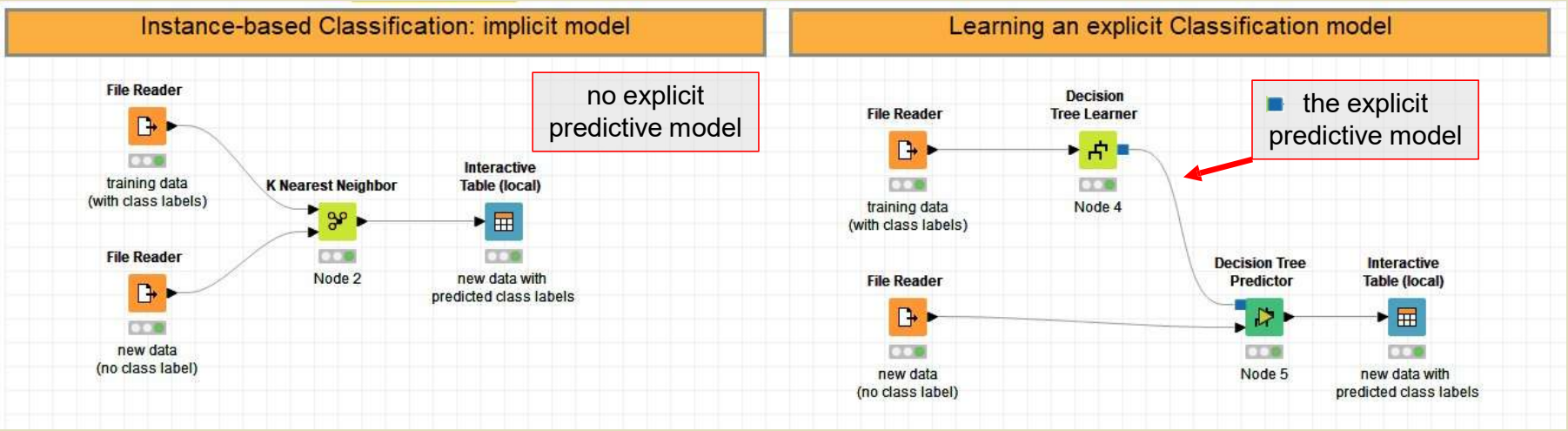
Lecture notes and videos created by
Prof. Giuseppe Di Fatta

Overview

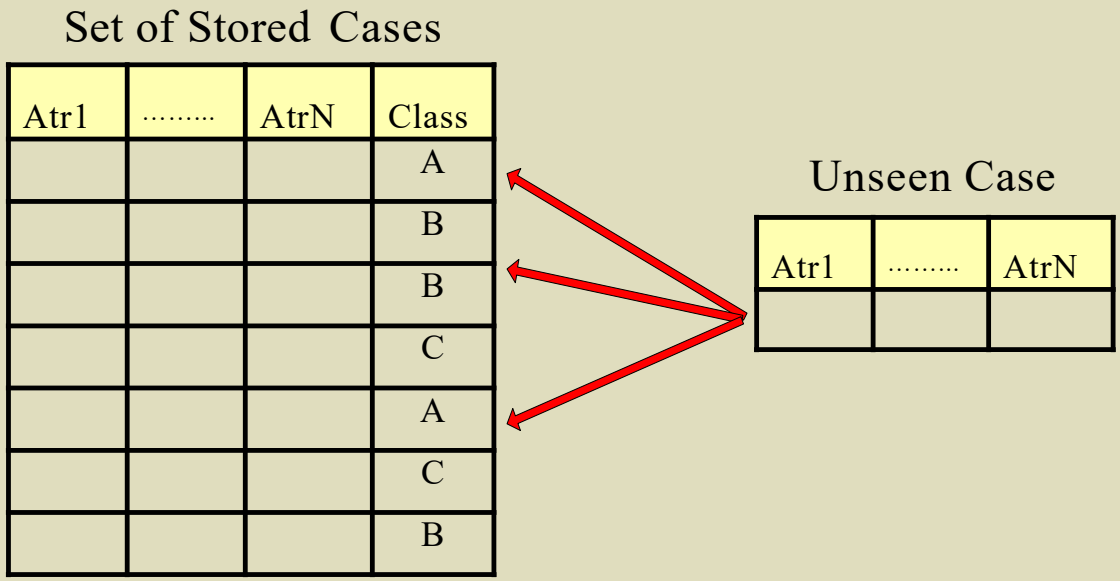
- Classification
  - Memory based reasoning
  - Decision Trees
  - Rule-based methods
  - Artificial Neural Networks
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines
  - Classification by Regression
    - Multi-response linear regression
    - Pairwise classification
    - Logistic regression
- Model evaluation
  - Accuracy and error rate
  - Cross-validation
  - ROC

The Classification Task

- Given a collection of records, where each record contains a set of *attributes* and
  - one of the attributes is the *class* (a nominal attribute).
- Task: find a *model* for the class attribute as a function of the values of the other attributes.
  - Build a predictive model from a training dataset, which have been previously classified, e.g., by an expert.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - New (future) data with no class labels are classified by the predictive model.
  - Typically a *test dataset* with class labels is used to determine the accuracy of the model (predicted vs actual label).



Instance-Based Classification



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

Instance-based Classification

- Simplest form of learning: ***rote learning***
  - Training instances are searched for instance that most closely resembles new instance
  - Instance-based learning is *lazy* learning: the instances themselves represent the knowledge
  - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly

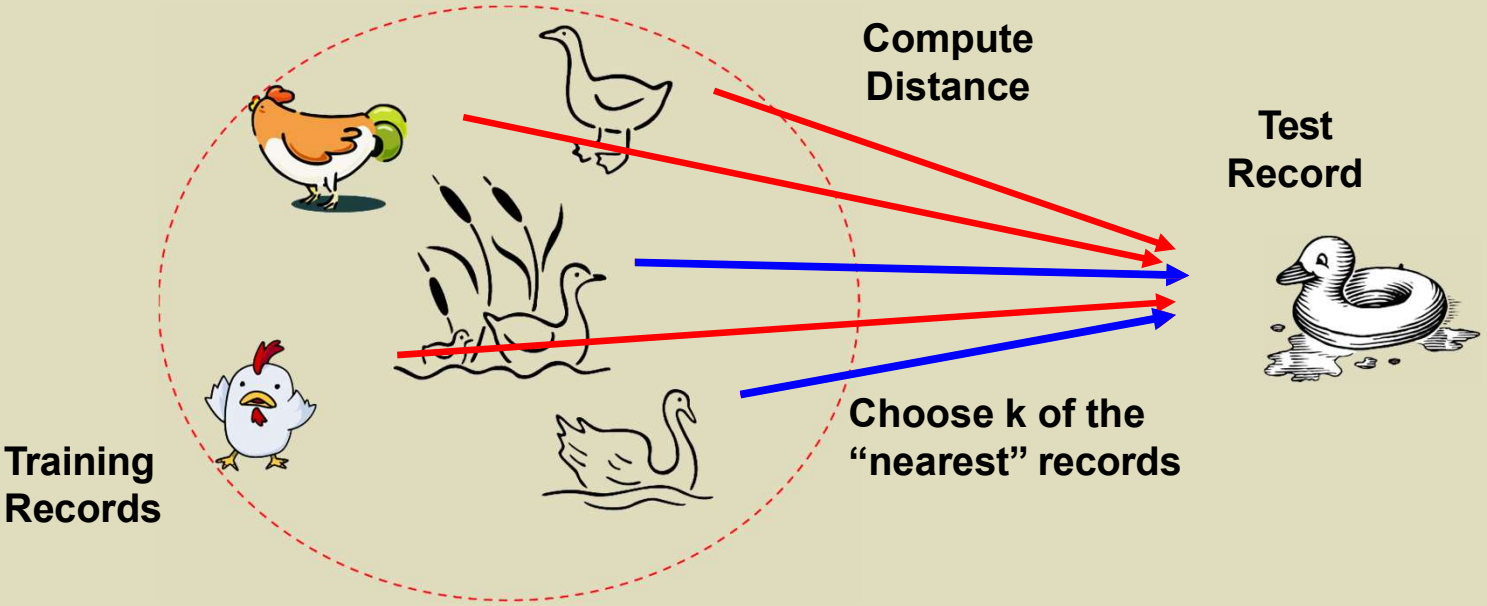
- Proximity measure (similarity/dissimilarity) defines what's "learned"

- **Nearest Neighbor**:
  - Uses "closest" points (nearest neighbors) for performing classification
    - *nearest-neighbor (1-NN)*
    - *k-nearest-neighbor (k-NN)*

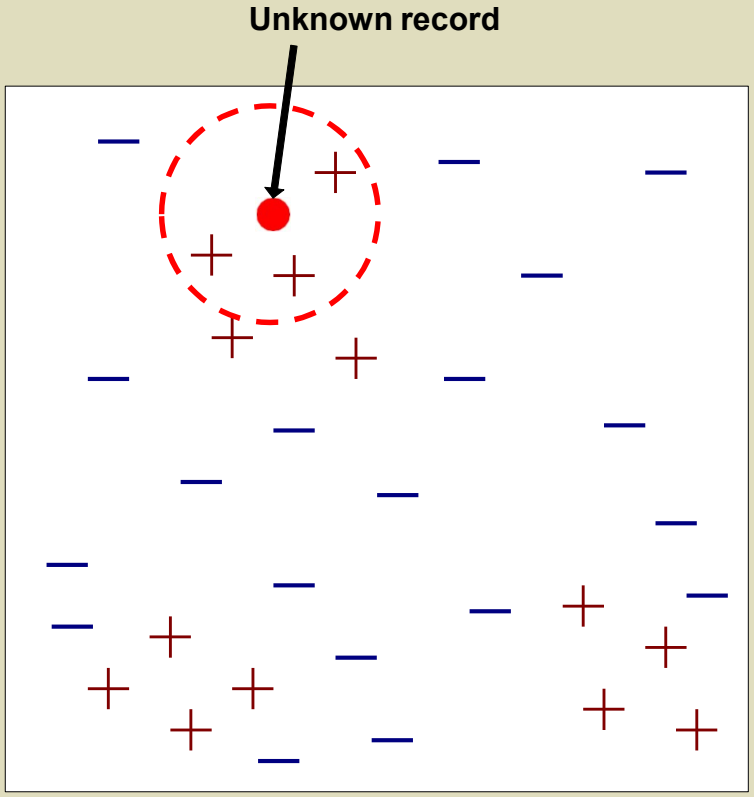
Nearest-Neighbor Classifiers

- **Nearest-Neighbor**:
  - Uses “closest” points (nearest neighbors) for performing classification

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it’s probably a duck



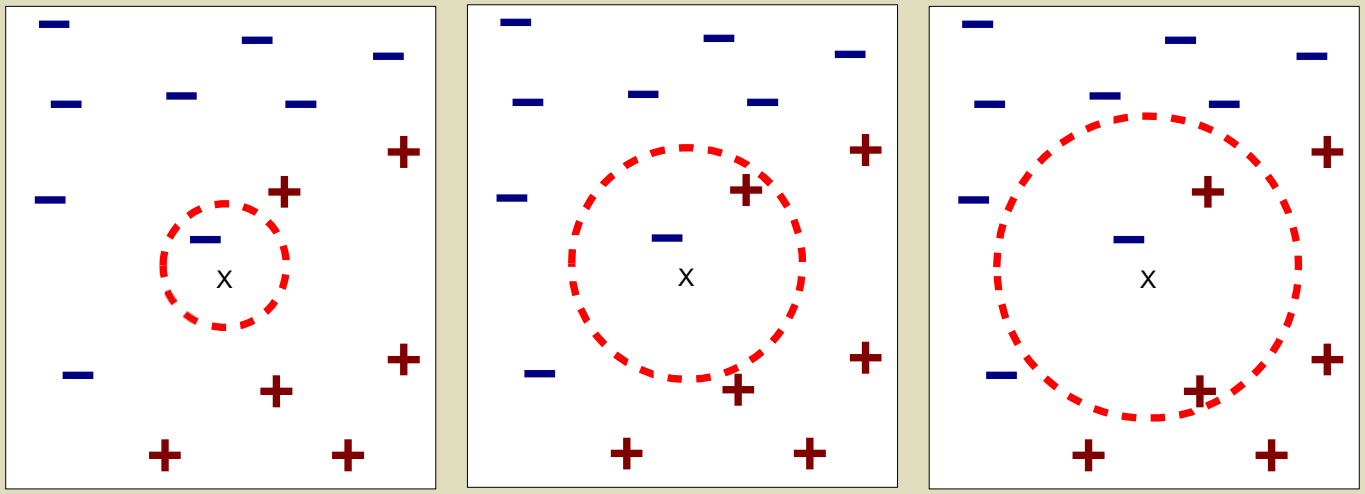
Nearest-Neighbor Classifiers



- 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:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

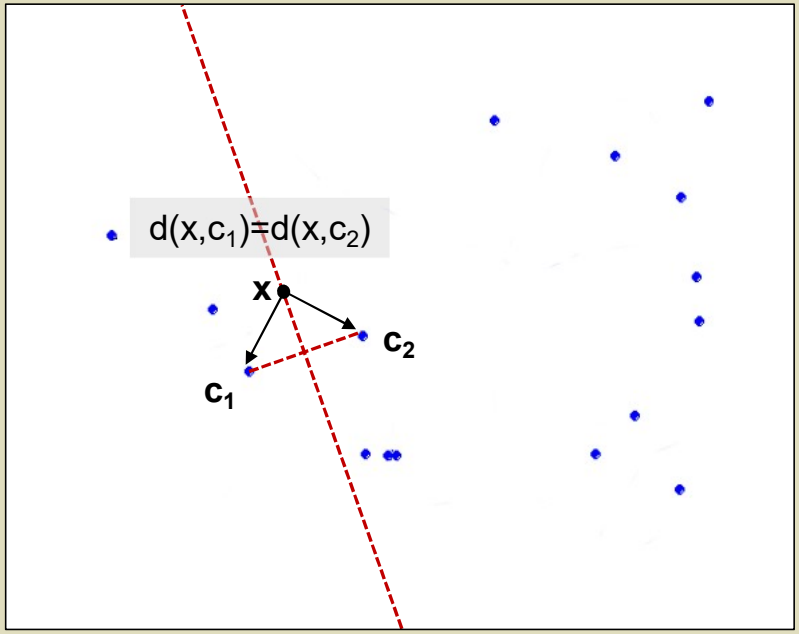
Nearest Neighbor

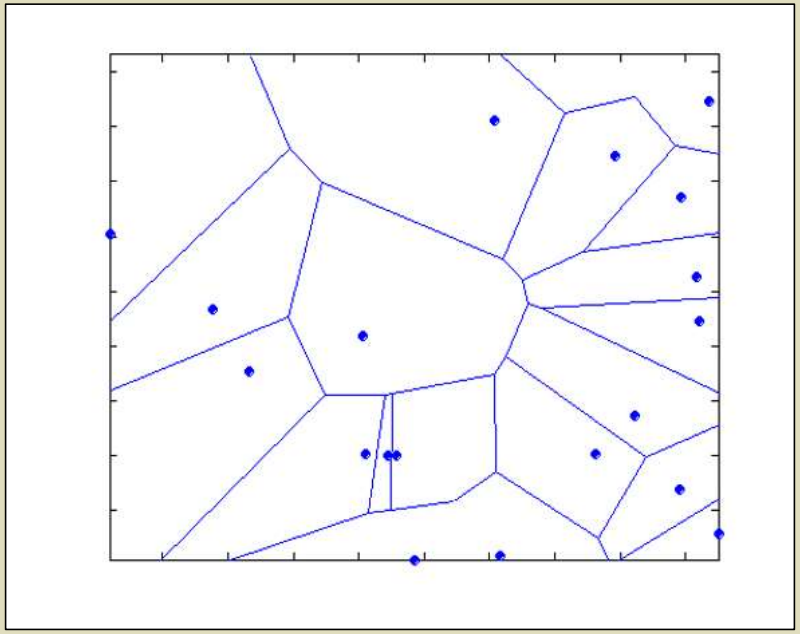


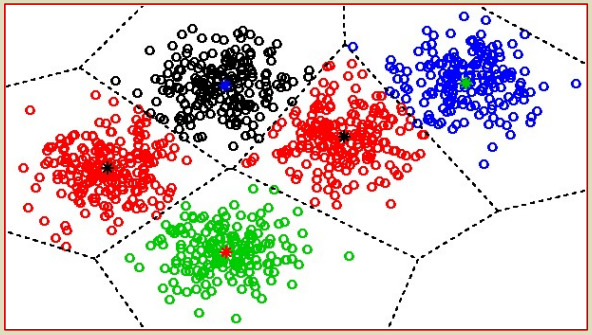
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

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

1-NN: Voronoi Diagram







- What is the relation with k-means clustering?
  - In k-means the input data are clustered around k centroids. The k centroids form a 1-NN Voronoi tessellation.

Nearest-Neighbor Classification

- Problem with Euclidean measure:
  - High dimensional data
    - curse of dimensionality
  - Can produce counter-intuitive results

$b_{11}\, b_{10}\, b_9\, b_8\, b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0$

b_i: 1 or 0 ❼ feature [i] is/is not present

1 1 1 1 1 1 1 1 1 1 1 0

0 1 1 1 1 1 1 1 1 1 1 1

d_E = 1.4142

vs

1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1

d_E = 1.4142

Solutions:

- Normalize the vectors to unit length
- Use different metric (e.g. Tanimoto distance)

Nearest-Neighbor Classification

- k-NN classifiers are lazy learners
  - It does not build models explicitly, unlike eager learners such as decision tree induction, rule-based systems, ANN, etc.
  - Classifying unknown records is relatively expensive for large datasets.
    - For a dataset D (|D|=n) in a d-dimensional space, the naïve implementation has complexity O(dn).

- Often very accurate and slow
  - simple version of 1-NN scans entire training data to derive a prediction
  - optimised versions make use of space partitioning trees (e.g., KD-trees, R-trees) for logarithmic average time complexity O(log(n))
- Assumes all attributes are equally important
  - Remedy: attribute selection or weights
- Possible remedies against noisy instances:
  - Take a majority vote over the *k* nearest neighbors
  - Removing noisy instances from dataset
- Statisticians have used *k*-NN since early 1950s
  - If $n \to \infty$ and $k/n \to 0$, classification error approaches minimum

When to Consider NN Classifiers

- Instances map to points in R^d
- Less than 20 attributes per instance
- Lots of training data

Advantages:

- Training is very fast
- Learn complex target functions
- Do not loose information

Disadvantages:

- Slow at query time
- Easily fooled by irrelevant attributes

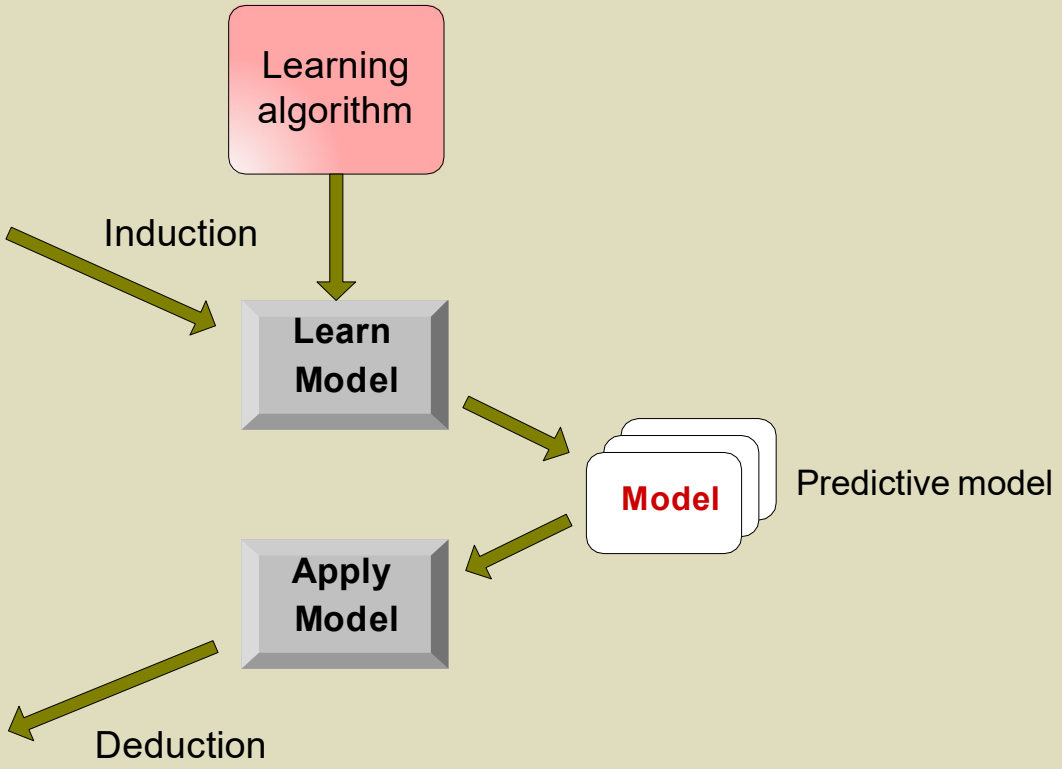
Classification: Learning Predictive Models

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	?

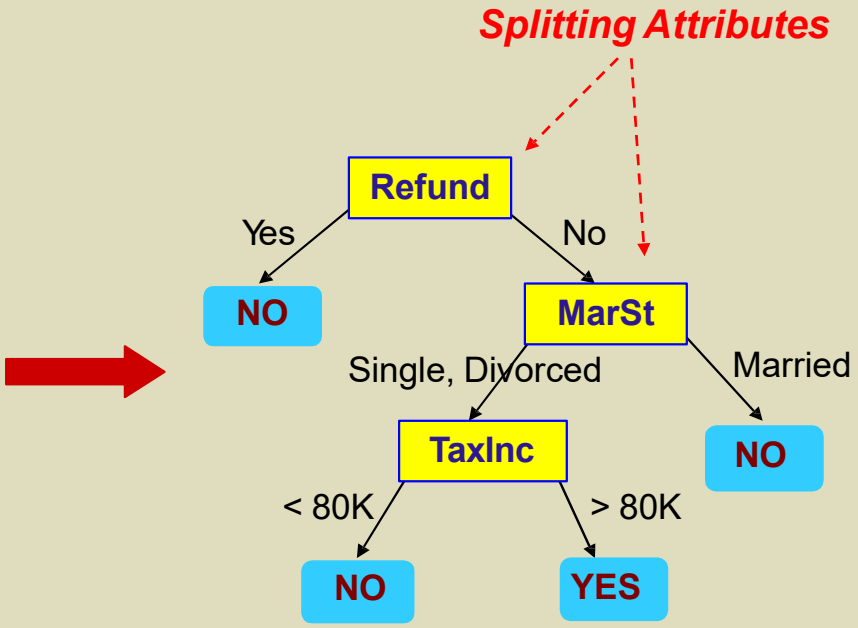
Test Set



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

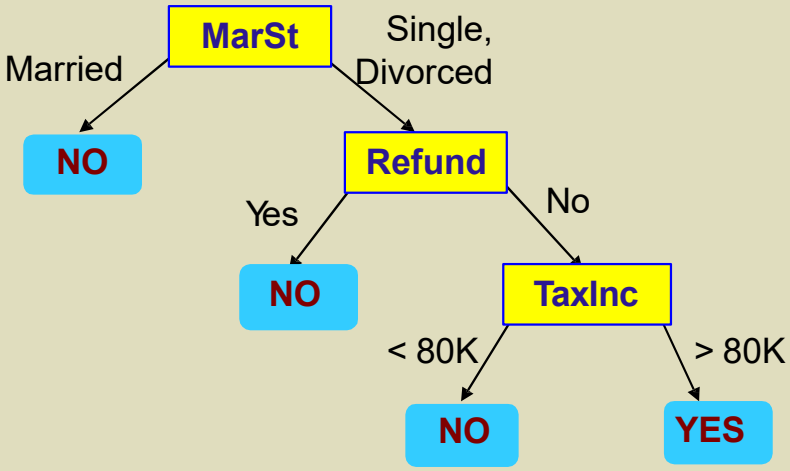
Training Data



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!

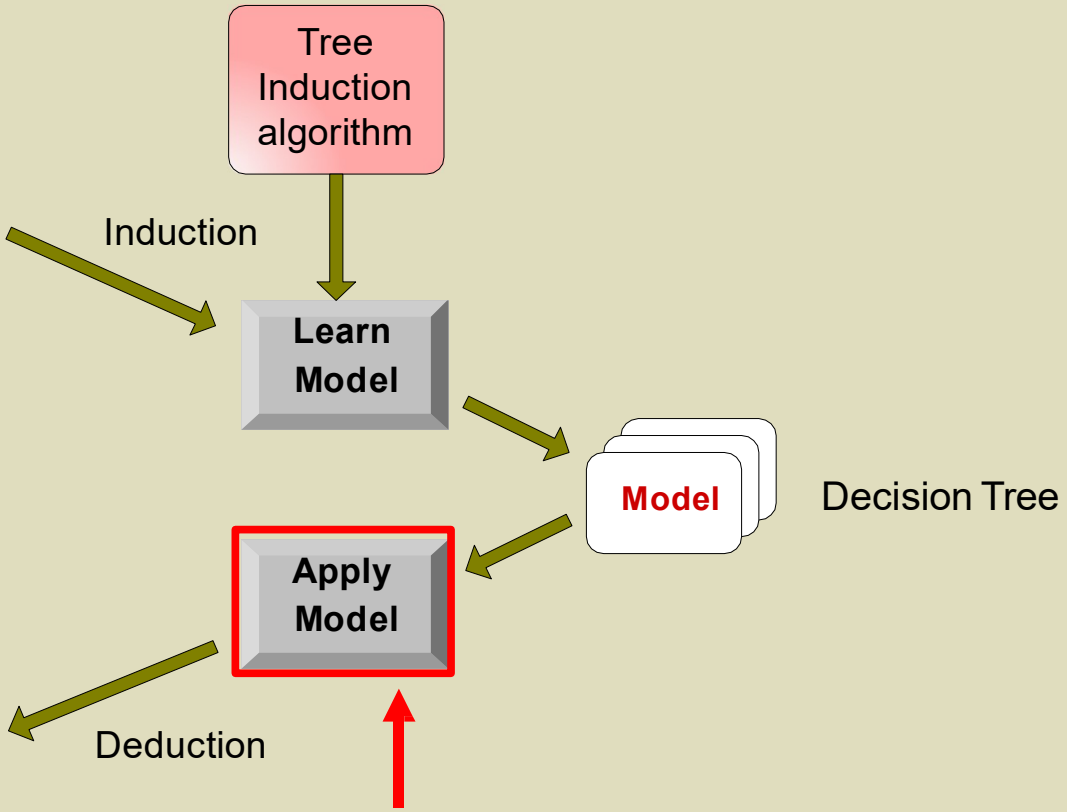
Decision Tree Classification Task

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

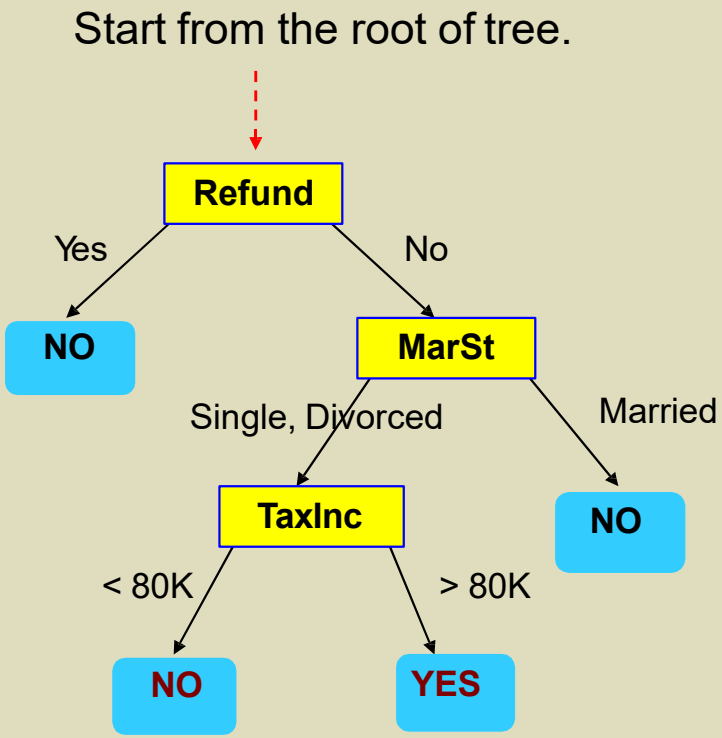
Training Set

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



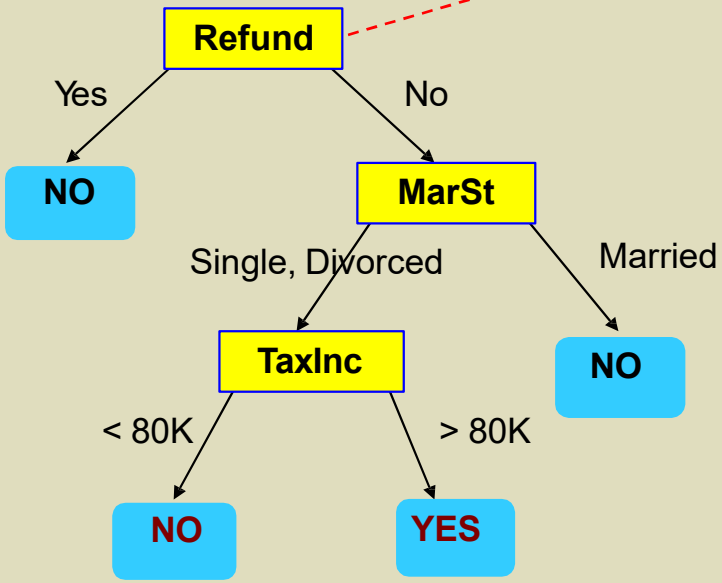
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

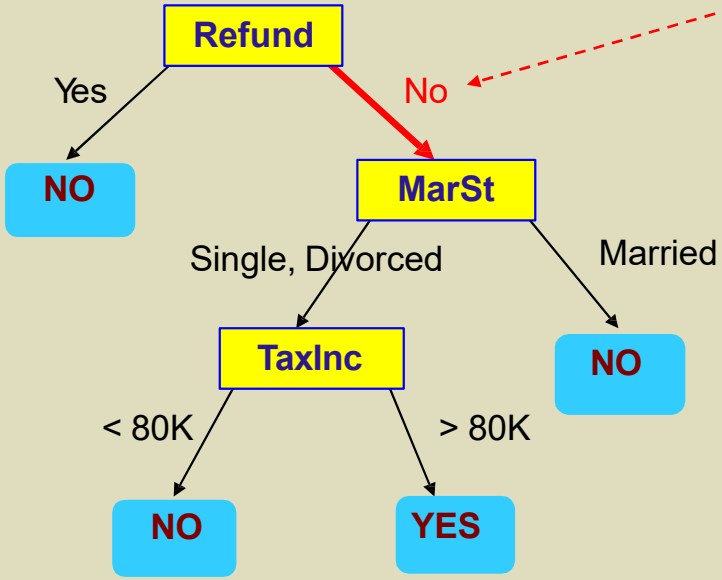
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

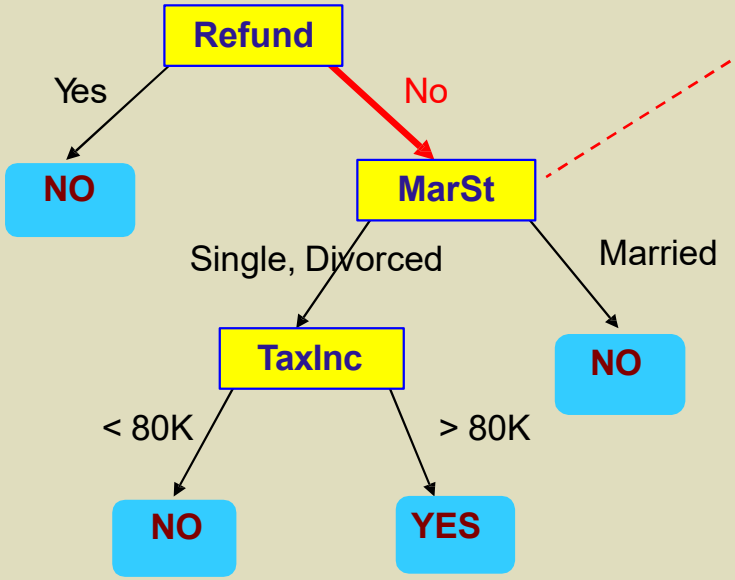
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

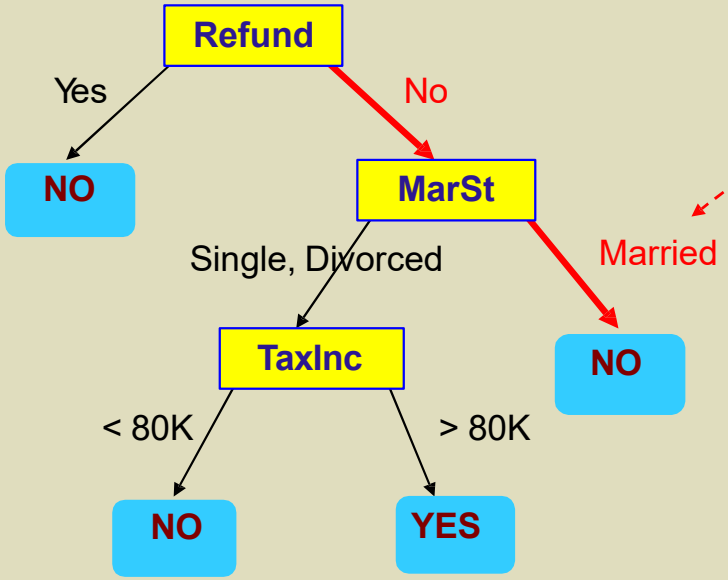
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

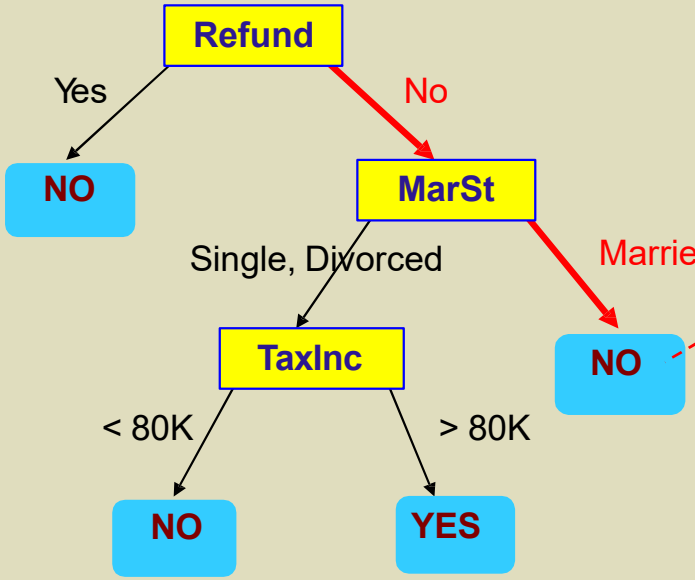
Apply Model to Test Data



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”

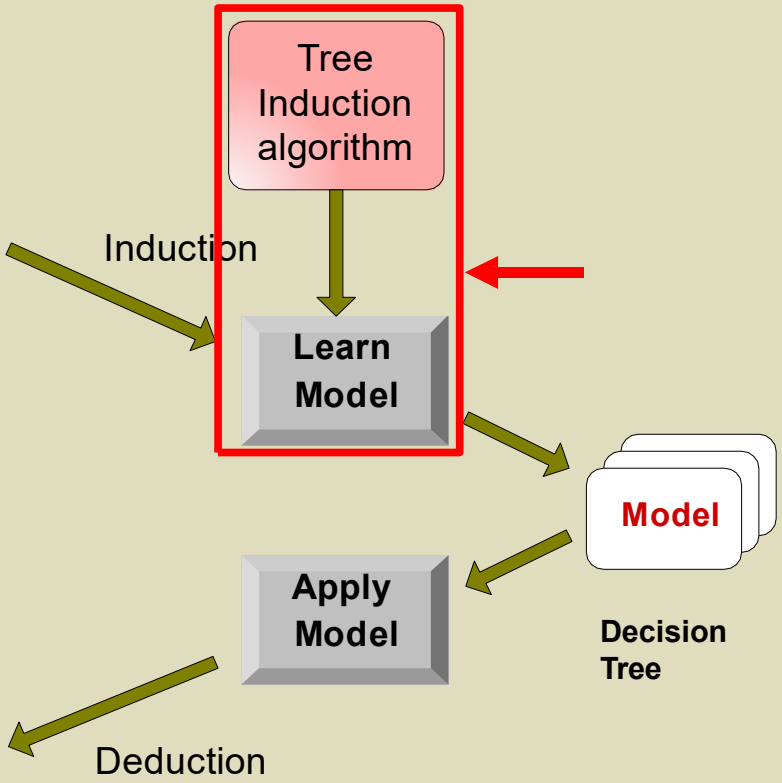
Decision Tree Learning Task

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



Decision Tree Induction Algorithms

- Many algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART: Classification And Regression Tree
  - ID3, C4.5, C5.0
  - SLIQ
  - SPRINT
  - ...

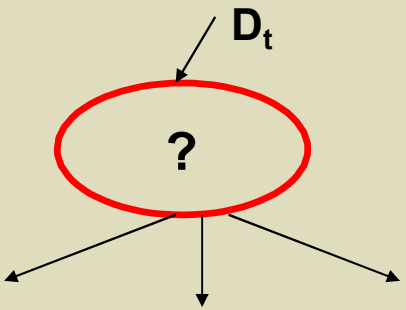
General Structure of Hunt's Algorithm

Let D_t be the set of training records that reach a node t

General recursive procedure:

- If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
- If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
- If D_t contains records that belong to more than one class:
  - Use an attribute test to split the data into smaller subsets (child nodes).
  - Recursively apply the procedure to each subset.

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



Tree Induction

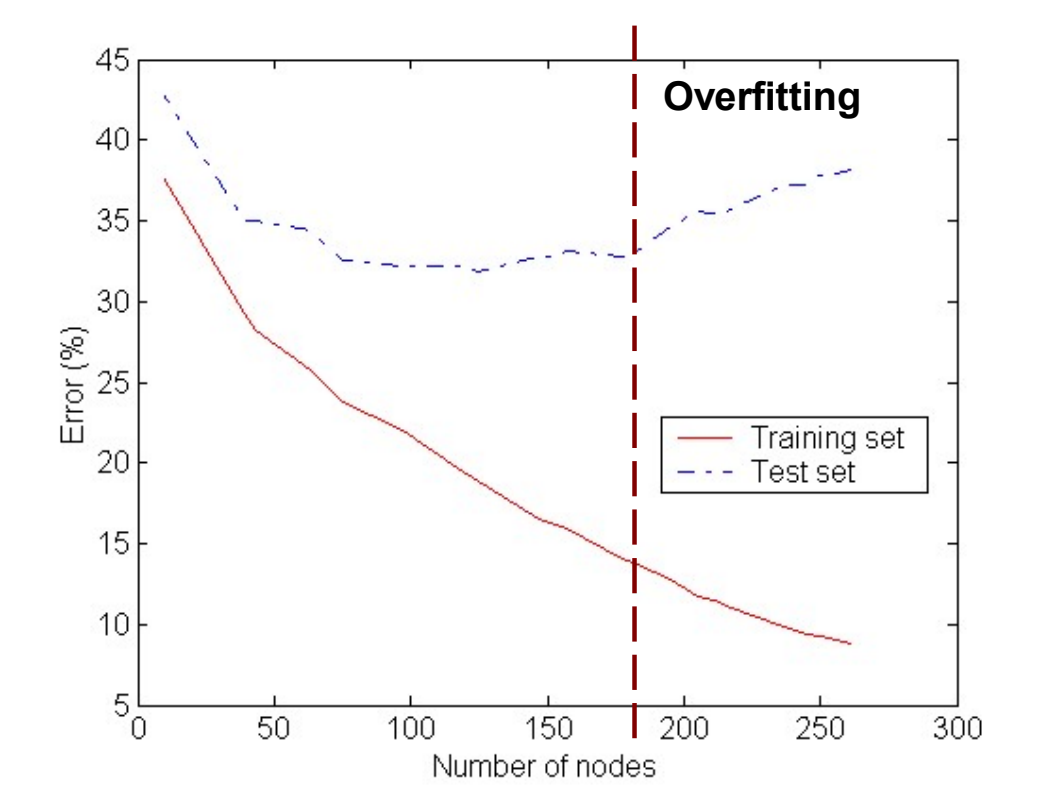
- Greedy strategy
  - Split the records based on an attribute test that optimizes a certain criterion.
    - Gini Index (CART)
    - Entropy and Information Gain (ID3, C4.5)
    - Misclassification error

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

Practical Issues of Classification

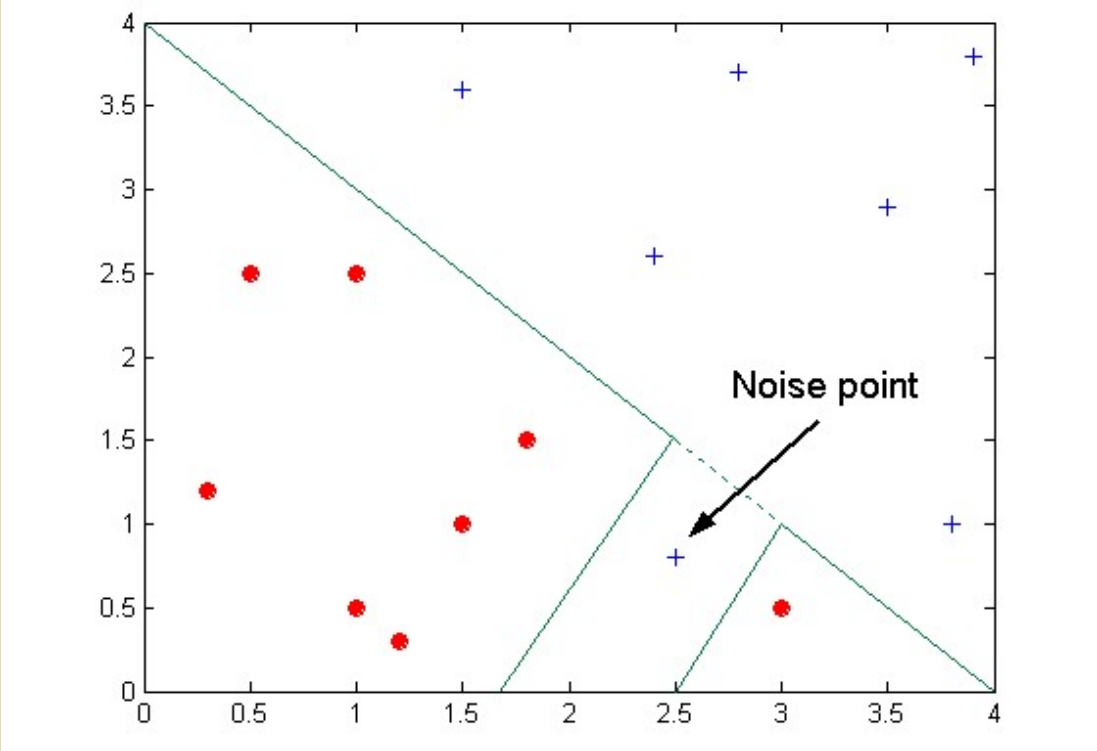
- Underfitting and Overfitting
- Missing Values
- Costs of Classification

Underfitting and Overfitting



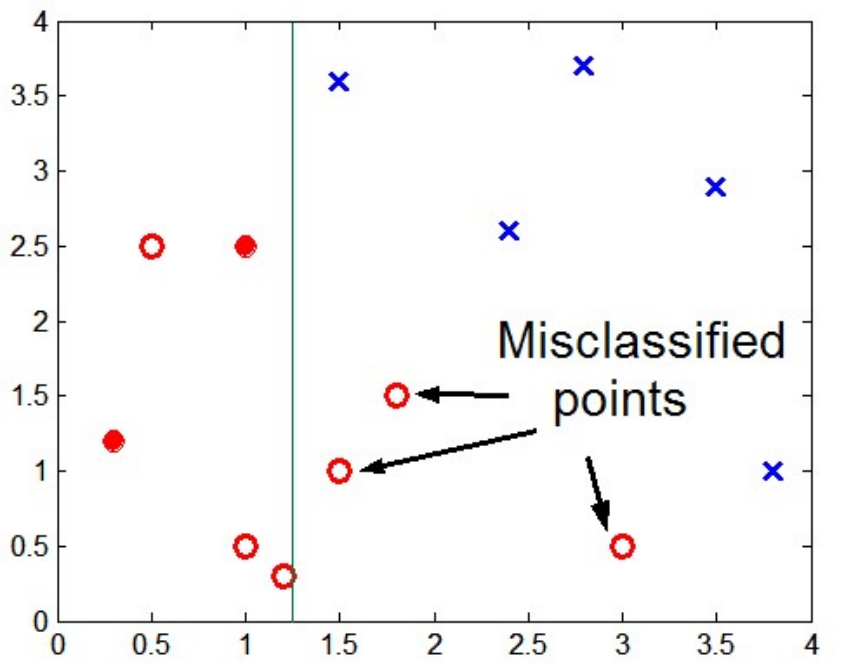
Underfitting: when model is too simple, both training and test errors are large

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

Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
  - Pre-pruning
  - Post-pruning
- Training error (error on training data): also referred to as 'resubstitution' error
  - Training error does not provide a good estimate of how well the tree will perform on previously unseen records
- Need ways for estimating error
  - Estimating generalization error with penalty term to compensate
  - Testing error (error on test data): model evaluation (e.g., cross-validation)

Next:

- Model Evaluation