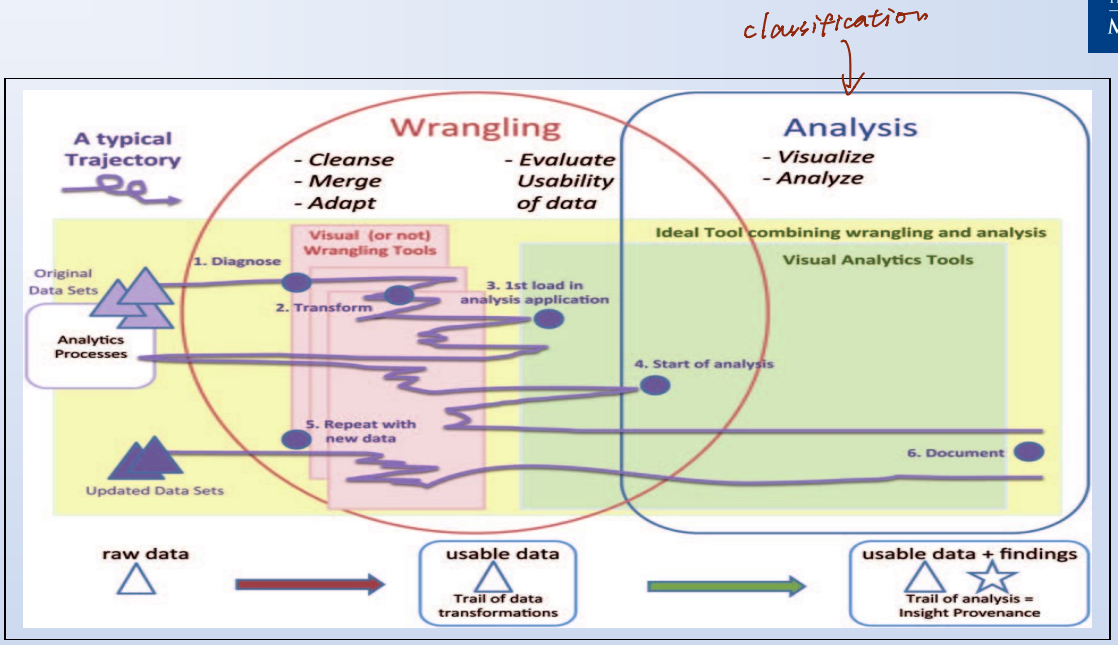
COMP20008 Elements of Data Processing

Classification

Semester 1, 2020

Contact: uwe.aickelin@unimelb.edu.au

Where are we now?



Classification

- Introduction to classification
- Comparing Classification and Regression
- Decision tree classification
  - Will make connections to entropy
- k nearest neighbour classification

Classification vs Regression

Classification

- Classification
  - Making predictions about the future, based on historical data

- Predictive modelling/classification
  - The sexy part of data science ?!
  - A foundation for machine intelligence, AI, machines replacing humans and taking our jobs ….

Classification – Example 1

Predicting disease from microarray data

label

Training

	Gene 1	Gene 2	Gene 3	…	Gene n	Develop cancer <1 year
Person 1	2.3	1.1	0.3	…	2.1	1
Person 2	3.2	0.2	1.2	…	1.1	1
Person 3	1.9	3.8	2.7	…	0.2	1
…	…	…	…	…	…	..
Person m	2.8	3.1	2.5	…	3.4	0

Testing

	Gene 1	Gene 2	Gene 3	…	Gene n	Develop cancer <1 year
Person X	2.1	0.9	0.6	…	1.9	?

Classification – Example 2

Animal classification

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Test data

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

reptile ?

https://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

Classification- Example 3

Banking: classifying borrowers

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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 set for predicting borrowers who will default on loan payments.

Test data

Tid	Home Owner	Marital status	Annual Income	Defaulted Borrower
11	No	Single	55K	?

No ?

Classification – Example 4

Detecting tax cheats

Tid	Refund	Marital Status	Taxable Income	Cheat
				class
	categorical	categorical	continuous	
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

Test data

Tid	Refund	Marital Status	Taxable Income	Tax Cheat
11	Yes	Married	125K	?

No.

Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one *class label*.
- Find a predictive *model* for each class label as a function of the values of all attributes, i.e. $y = f(x_1, x_2, \ldots, x_n)$
  - y: *discrete value*, target variable
  - $x_1, \ldots x_n$: attributes, predictors
  - f: is the predictive model (a tree, a rule, a mathematical formula)
- Goal: previously unseen records should be assigned a class as accurately as possible
- A *test set* is used to determine the accuracy of the model, i.e. the full data set is divided into training and test sets, with the training set used to build the model and the test set used to validate it

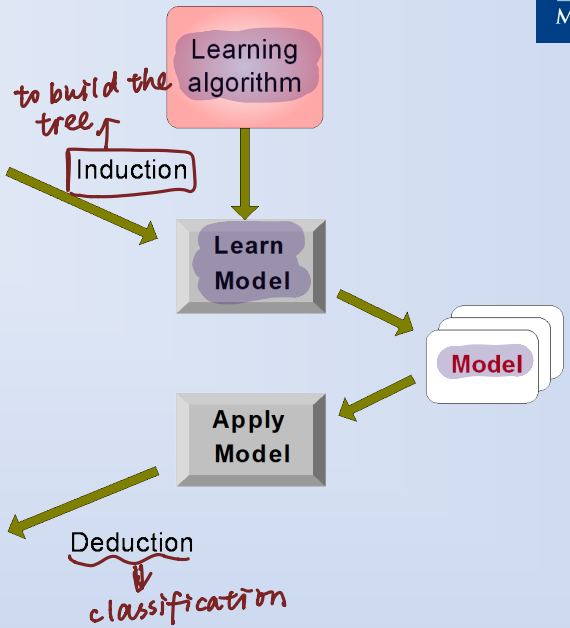
Classification framework

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

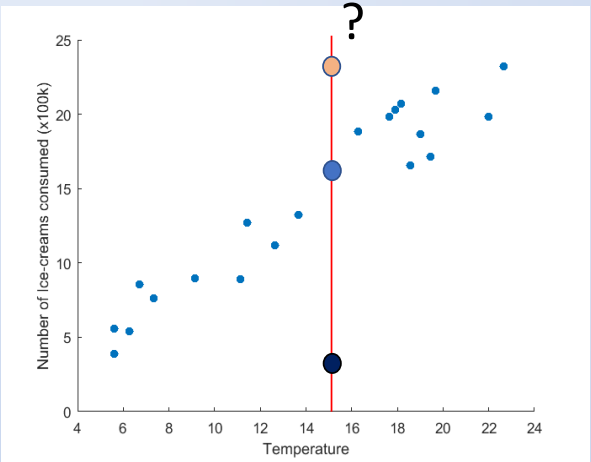


Regression: Recall Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes* and one *target variable*
- Learn predictive model from data, i.e. $y = f(x_1, x_2, \ldots, x_n)$
- y: target variable is **continuous** (key difference to classification)
- $x_1, \ldots x_n$: attributes, predictors

Regression example 1

Predicting ice-creams consumption from temperature: y = f(x)



Regression example 2

Predicting the activity level of a target gene

	Gene 1	Gene 2	Gene 3	…	Gene n	Gene n+1
Person 1	2.3	1.1	0.3	…	2.1	3.2
Person 2	3.2	0.2	1.2	…	1.1	1.1
Person 3	1.9	3.8	2.7	…	0.2	0.2
…	…	…	…	…	…	…
Person m	2.8	3.1	2.5	…	3.4	0.9

	Gene 1	Gene 2	Gene 3	…	Gene n	Gene n+1
Person m+1	2.1	0.9	0.6	…	1.9	?

Quiz

- Is it classification or regression?
  - Prediction of rain or no rain tomorrow based on today's rain data classification
  - Prediction of the amount of rain tomorrow based on today's rain data regression
  - Prediction of exam mark based on study hours regression
  - Prediction of exam pass or fail based on study hours classification
  - Prediction of salary based on degree results regression
  - Prediction of complications of surgery based on pre-op medical data classification

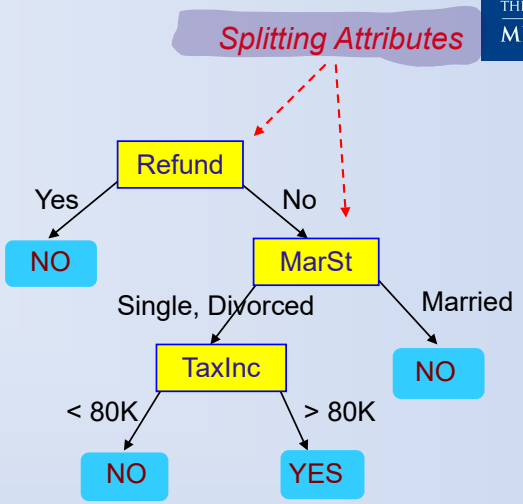
if the output is percentage, could do this as regression model

Decision Trees

Decision Tree example: Prediction of tax cheats

	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

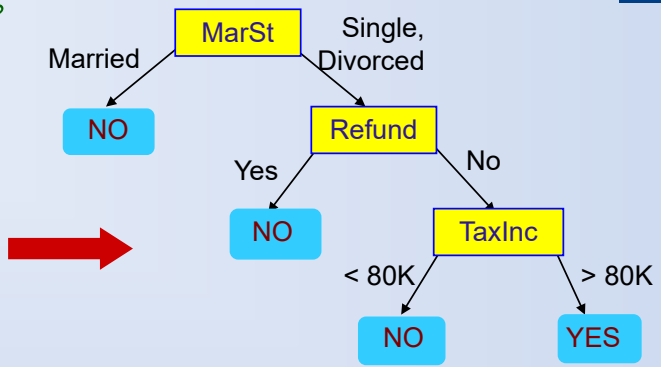


Model: Decision Tree

Decision Tree example: Prediction of tax cheats 2

	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

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

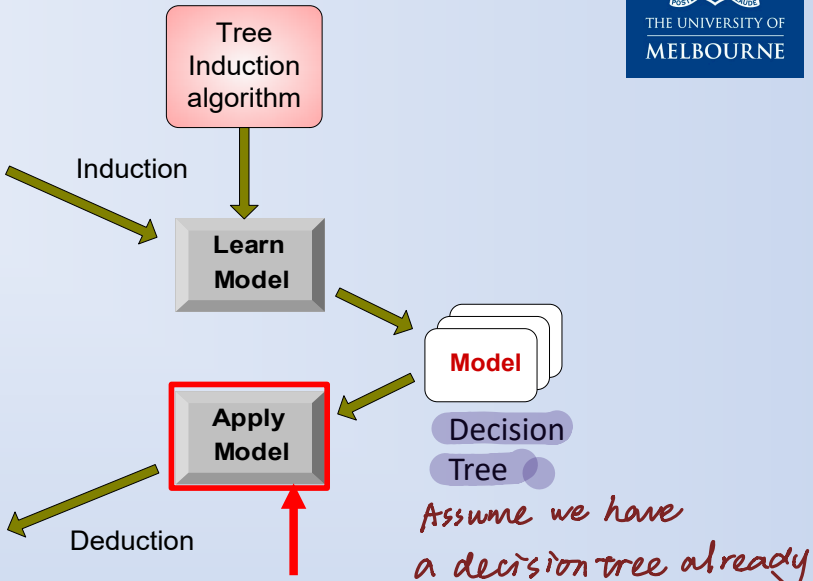
Decision Tree Classification Task

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

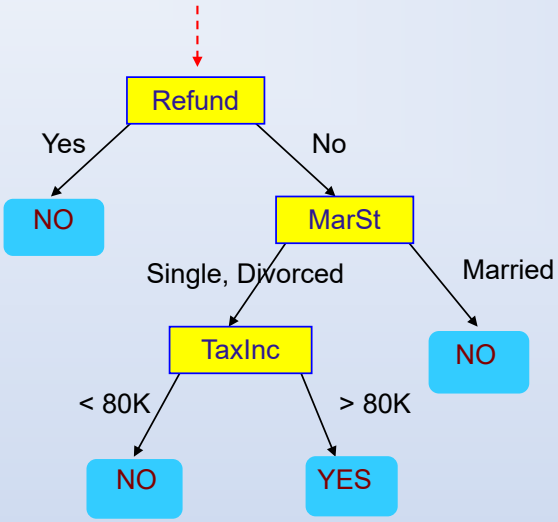
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Apply DT Model to Test Data (1)

Start from the root of the tree



Test Data

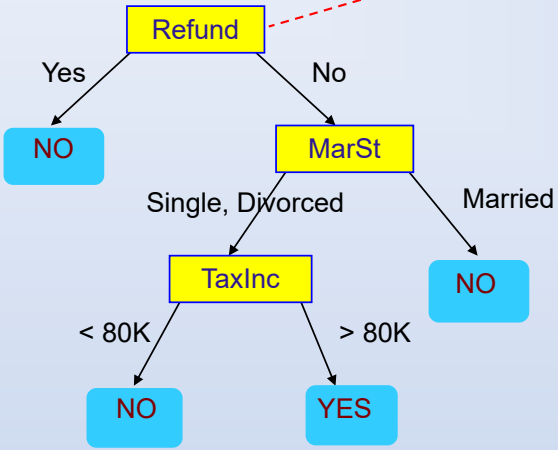
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

No

Apply DT Model to Test Data (2)

Test Data

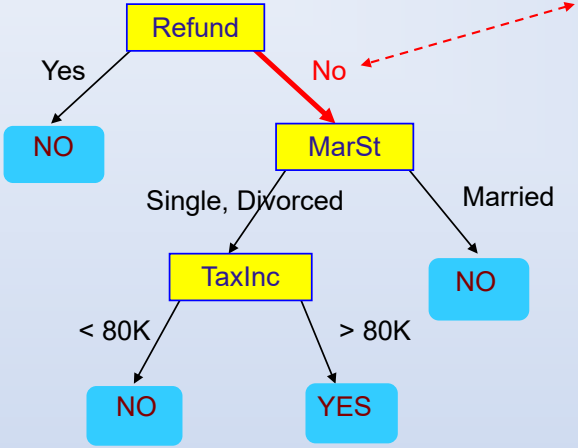
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply DT Model to Test Data (3)

Test Data

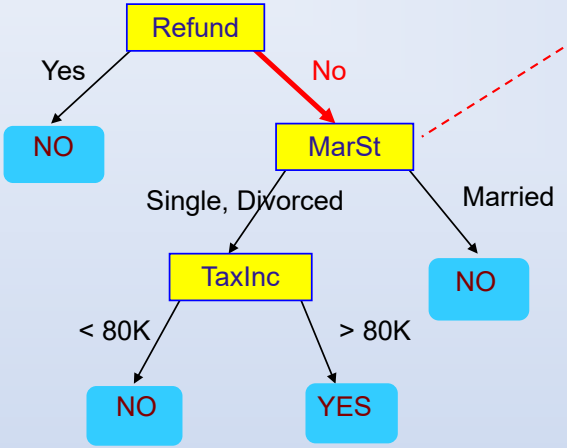
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply DT Model to Test Data (4)

Test Data

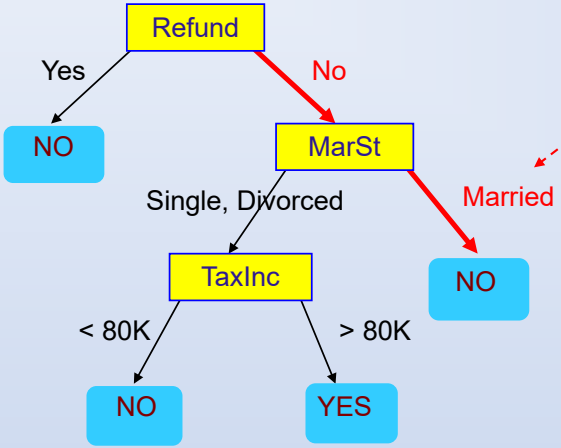
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply DT Model to Test Data (5)

Test Data

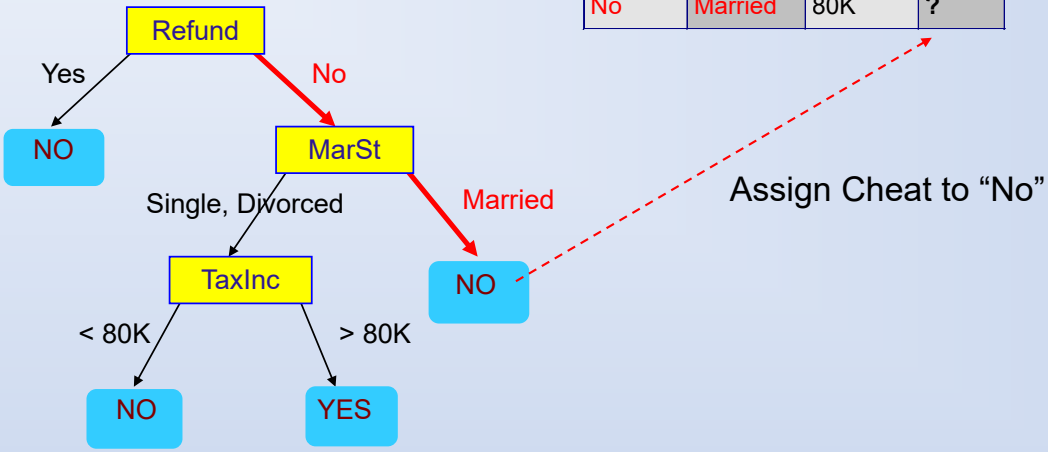
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply DT Model to Test Data (6)

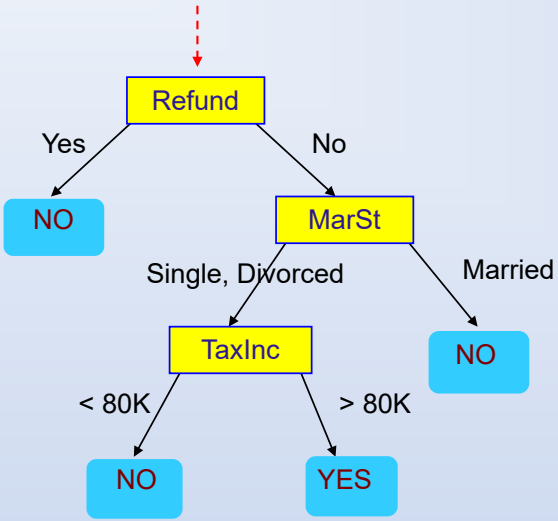
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply DT Model to Test Data (7)

Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Single	100K	? Yes

Decision Trees

Decision tree

- A flow-chart-like tree structure
- Internal node denotes a test on an attribute
- Branch represents an outcome of the test
- Leaf nodes represent class labels

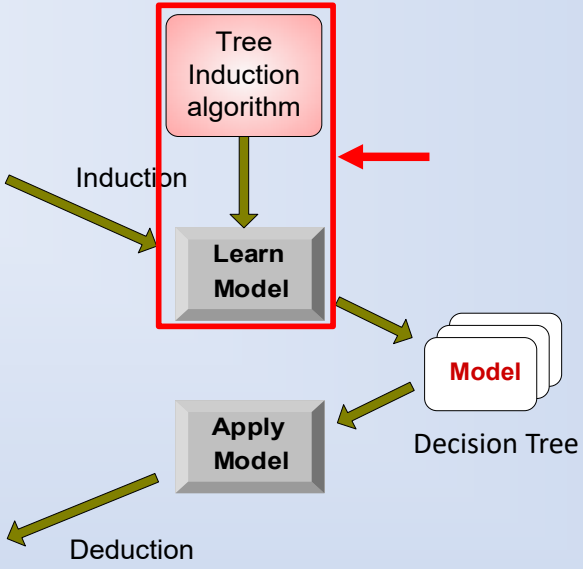
Decision Tree Algorithm

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 Algorithm (2)

- Many Algorithms:
  - We will look at a representative one (Hunt's algorithm)

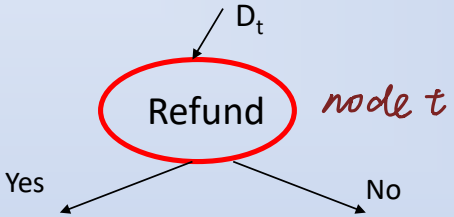
- How would you code an algorithm that automatically constructs a decision tree for the tax cheat data?

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

General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach node t
- General Procedure:
  - If D_t contains only records that belong to 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
  - 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

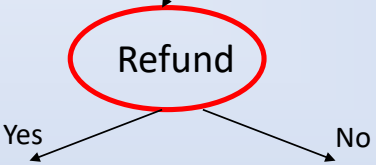


Hunt's Algorithm Example

If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. 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

D_t

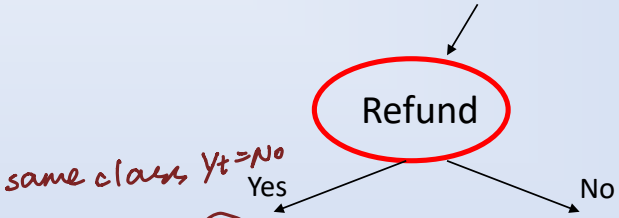


Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
4	Yes	Married	120K	No
7	Yes	Divorced	220K	No

Tid	Refund	Marital Status	Taxable Income	Cheat
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

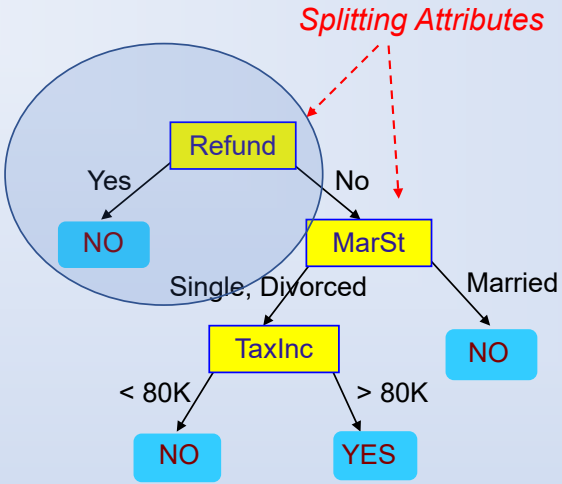
Stopping condition: leaf node

- If D_t contains records that all belong to 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 as the default class y_d



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
4	Yes	Married	120K	No
7	Yes	Divorced	220K	No

Decision Tree splitting example



Tree induction (creation)

Tree Induction

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?

- Determine when to stop splitting
  - When a node only has instances of a single class

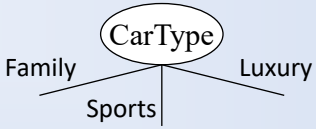
How to Specify Test Conditions?

- Depends on attribute types
  - Nominal → also referred as categorical attributes and there is no order (rank, position) among values of nominal attributes.
  - Ordinal → variable have two or more categories (like nominal variable) only the categories can also be ordered or ranked.
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

1. Nominal:
   - The values of a nominal attribute are just different names, i.e. nominal attributes provide only enough information to distinguish one object from another(=,≠)
   - Examples: zip codes, employees ID numbers.
2. Ordinal:
   - The values of an ordinal attribute provide enough information to order objects(<, >)
   - Examples: Hardness of minerals, street numbers

Splitting Nominal Attributes

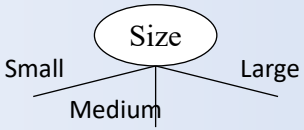
- Multi-way split: Use as many partitions as distinct values



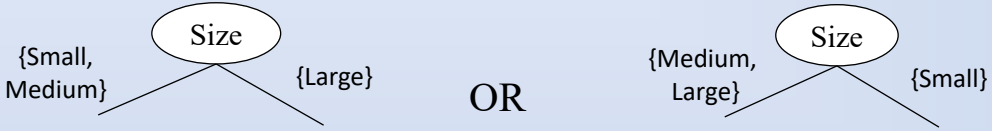
- Binary split: Divide values into two subsets, find an optimal partitioning

Splitting Ordinal Attributes

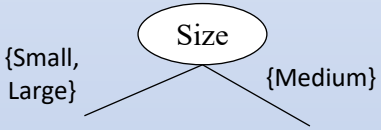
- Multi-way split: Use as many partitions as distinct values



- Binary split: Divides values into two subsets, find optimal partitioning



- What about this split?



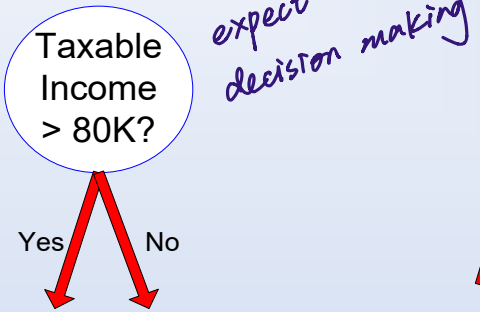
for some medical problem, this is exactly the way we want to split
eg. how many blood cells

Splitting Continuous Attributes

- Different ways of handling
  - Discretisation to form an ordinal categorical attribute
    - Static – discretise once at the beginning → we know the maximum range and minimum range
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles) or clustering → we don't know beforehand what's gonna happens → depends on the arriving
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and find the best cut
    - can be more computing intensive

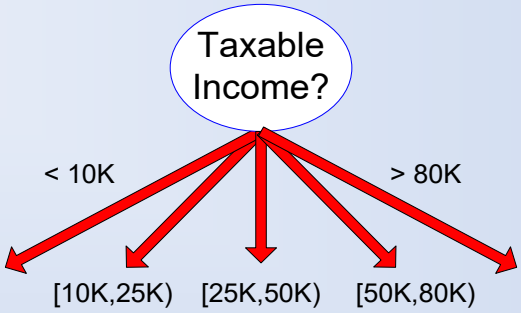
give a finite number of combinations

Splitting Continuous Attributes



expect based decision making

(i) Binary split



(ii) Multi-way split

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

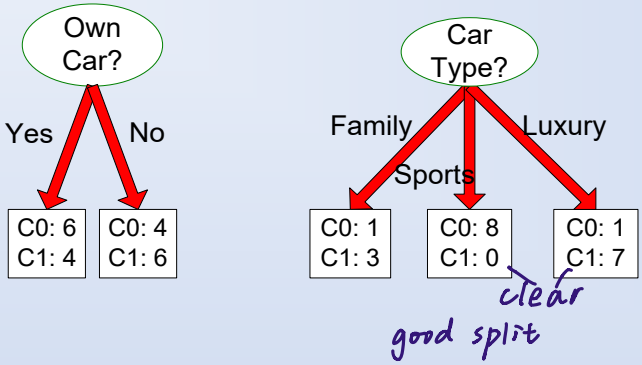
Tree Induction

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?

- Determine when to stop splitting
  - When a node only has instances of a single class

How to determine the Best Split

Before Splitting: **10 records of class 0,**
10 records of class 1



Which test condition is better?

How to determine the best split

All items after the split belongs to one class → good split

- Greedy approach: Nodes with homogeneous class distribution are better
- Need a measure of node impurity:

The more pure the split, the better!

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity

good!

Measures of Node Impurity

- Entropy
  - We have seen entropy in the feature correlation section, where it was used to measure the amount of uncertainty in an outcome
  - *Entropy can also be viewed as an impurity measure*
    - The set {A,B,C,A,A,A,A,A} has low entropy: low uncertainty and **low impurity**
    - The set {A,B,C,D,B,E,A,F} has high entropy: high uncertainty and **high impurity**

Node Impurity Criteria based on entropy

Entropy (H) at a given node t:

conditional Entropy formula ⇒ $$H(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t)

Measures homogeneity of a node: maximised for uniform distribution ?

- Maximum ($\log n_c$) when records are equally distributed amongst all classes (n_c is number of classes)
- Minimum (0.0) when all records belong to one class

Examples of computing Entropy

$$H(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 $\log_2$ 0 – 1 $\log_2$ 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Question: What is the entropy?

$$H(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	**13**
C2	**20**

P(C1) = $\frac{13}{33}$ P(C2) = $\frac{20}{33}$

Entropy = $-\left(\frac{13}{33}\log_2\frac{13}{33} + \frac{20}{33}\log\frac{20}{33}\right) = 0.967294$?

typo

Question: What is the entropy?

$$H(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	**13**
C2	**20**

P(C1) = $\frac{13}{33}$ P(C2) = $\frac{20}{33}$

Entropy = - (13/33)log(13/33))+ (20/33)log(20/33) = 1.07

How good is a Split? ⇒ how much more sorted is my data after spliting

Compare the impurity (entropy) of parent node (before splitting) with the impurity (entropy) of the children nodes (after splitting)

$$Gain = H(Parent) - H(Parent|Child)$$

$$= H(Parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} H(v_j)$$

- $H(v_j)$: impurity measure of node v_j
- j: children node index
- $N(v_j)$: number of data points in child node v_j
- N: number of data points in parent node

The larger the gain, the better

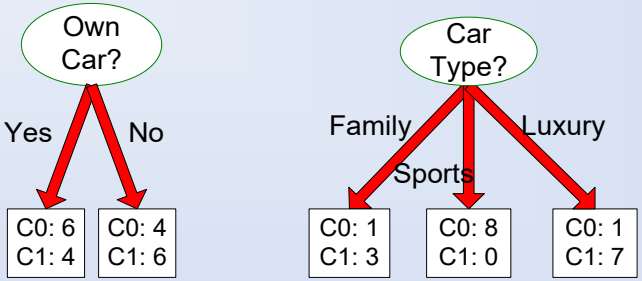
How good is a Split?

$$Gain = H(Parent) - H(Parent|Child)$$

$$= H(Parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} H(v_j)$$

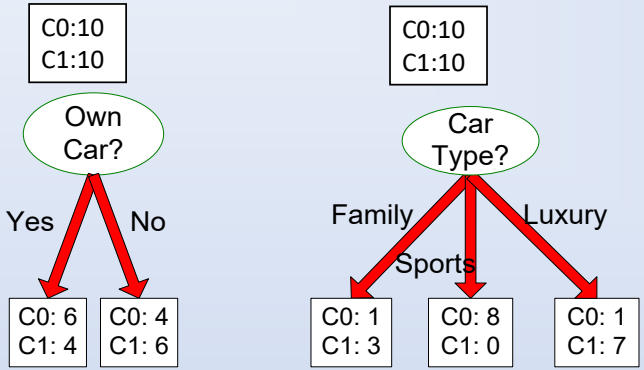
- Note: the information gain is equivalent to the mutual information between the class feature and the feature being split on
- Thus splitting using the information gain is to choose the feature with highest information shared with the class variable

How good is a Split?



How to determine the Best Split?

Before Splitting: **10 records of class 0,**
10 records of class 1

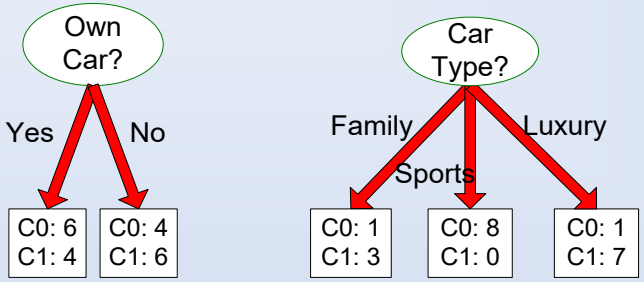


Which test condition is better?

- Compute the gain of both splits
- Choose the one with the larger gain

How to determine the Best Split?

Before Splitting: 10 records of class 0,
10 records of class 1



Own Car: Information gain=0.029

Car type: Information gain=0.62

We should choose Car type as the better split

Creating a decision tree

- Calculate information gain [Left Child],[Right Child] for all:
  - Refund [Yes], [No]
  - Marital status [Single],[Married],[Divorced]
  - Taxable income
    - [60,60], (60,220]
    - [60,70], (70,220]
    - [60,75],(75,220]
    - [60,85],(85,220]
    - [60,90],(90,220]
    - [60,95],(95,220]
    - [60,100],(100,220]
    - [60,120],(120,220]
    - [60,125],(125,220]
- Choose feature + split with the highest information gain and use this as the root node and its split
- Do recursively, terminating when a node consists of only Cheat=No or Cheat=Yes.

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

Question 2c) from 2016 exam

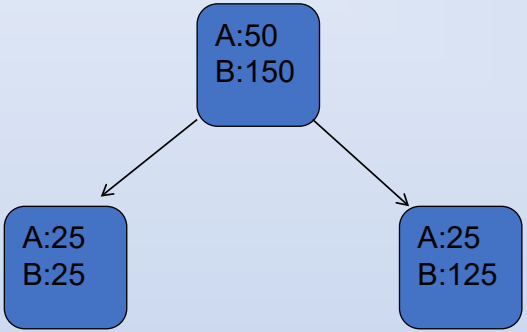
Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility.

Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility.

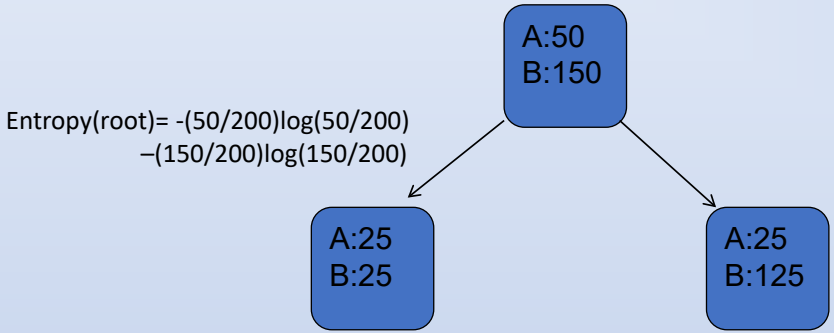
Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility.



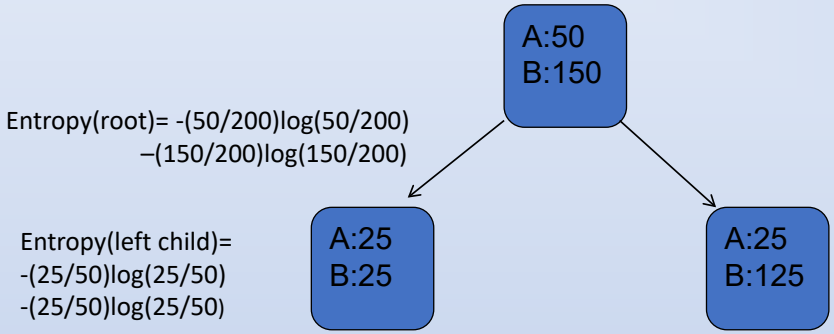
Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility.



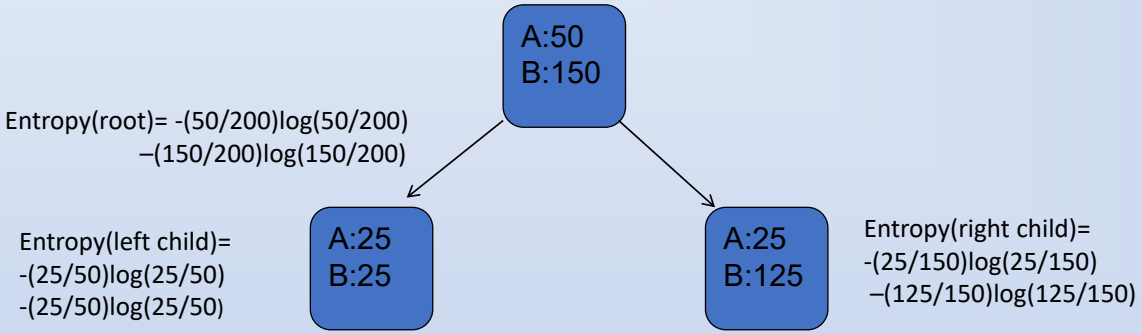
Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility.



Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility.



Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility.

Split utility= Information Gain

=Entropy(root) – Entropy(root|split)

=Entropy(root) – [(50/200)*Entropy(left child) +(150/200)*Entropy(right child)]

MI trees: advantages and disadvantages

- Advantages
  - Easy to interpret
  - Relatively efficient to construct
  - Fast for deciding about a test instance

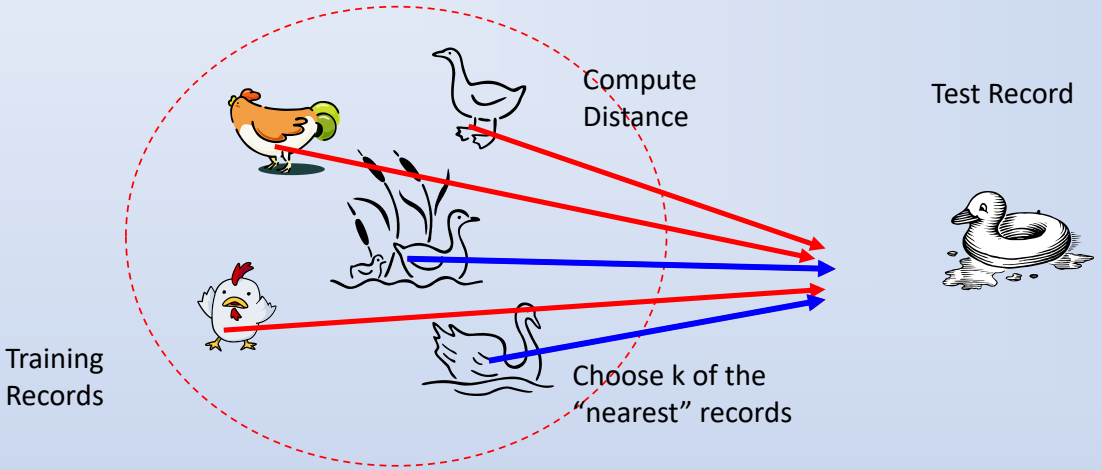
Assume a dataset with many many features, if you follow the basic approach, you will have a giant tree

- Disadvantages
  - A simple greedy construction strategy, producing a set of If ..then rules. Sometimes this is too simple for data with complex structure:
    - Everything should be as simple as possible, but not simpler
  - For complex datasets, the tree might grow very big and difficult to understand

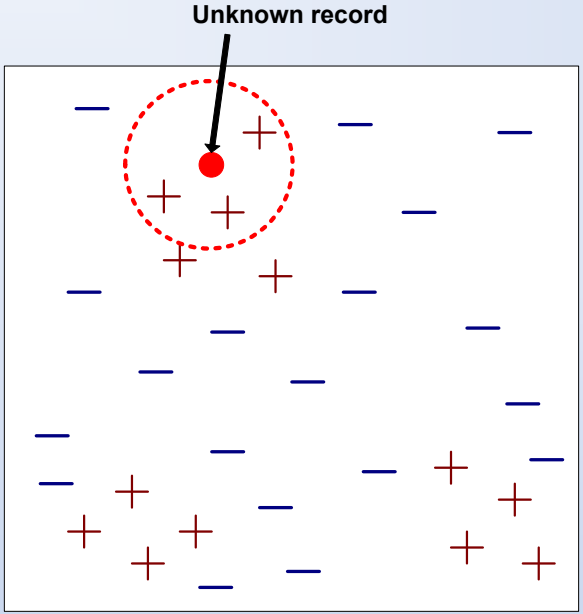
K-nearest neighbour classifier

Nearest Neighbour Classifier

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



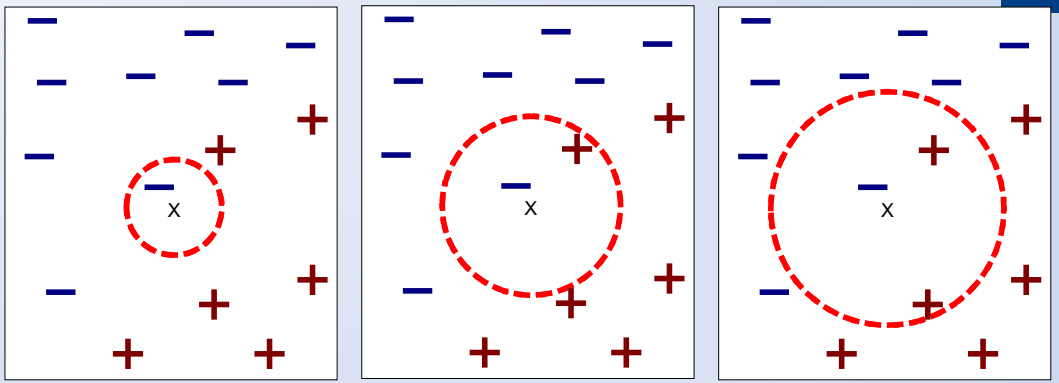
Nearest-Neighbour Classifier



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbours 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 a majority vote)

Definition of Nearest Neighbour

sensitive to k



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

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

Distance measure

- Compute distance between two points p=(p_1,p_2,…), q=(q_1,q_2,…)
  - Euclidean distance

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

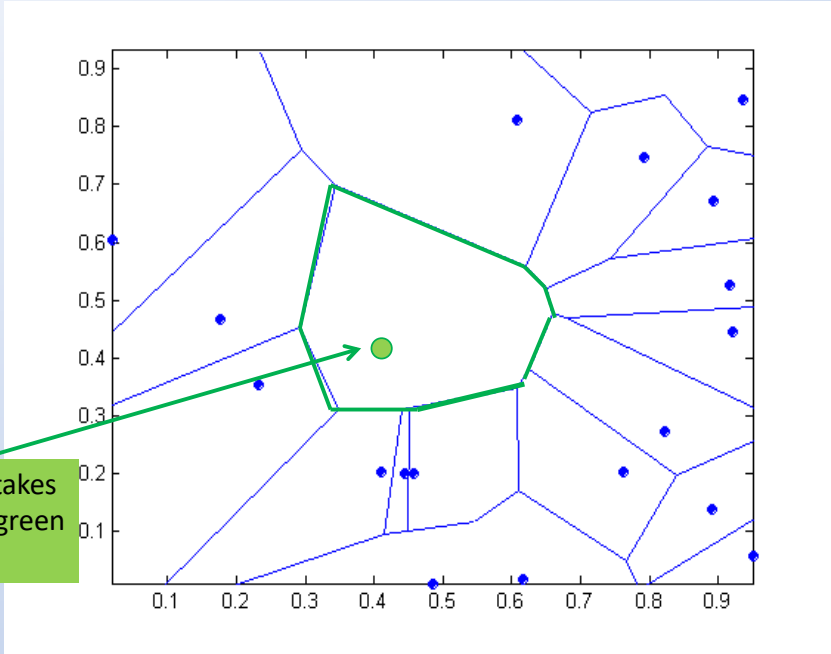
  - Can also use Pearson coefficient (similarity measure)

- Determine the class from nearest neighbour list
  - take the majority vote of class labels among the k-nearest neighbours
  - weigh the vote according to distance, e.g. w = $\frac{1}{d^2}$

↓ closer → higher weight

1 nearest-neighbour

Voronoi Diagram defines the classification boundary



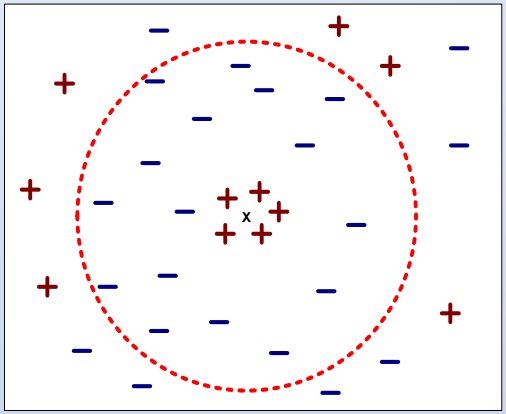
The green area takes the class of the green point

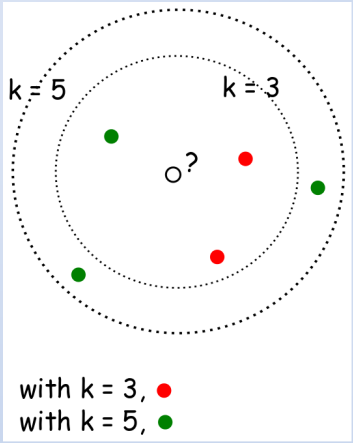
K-Nearest Neighbour classifier

Choosing a value for k:

- If k is too small, its sensitive to noise points
- If k is too large, the neighbourhood may include points from other classes

unreliable





http://2.bp.blogspot.com/-EK4tA2525EM/U-c6Q4jJuwI/AAAAAAAADjw/hdqRXuunpnQ/s1600/knn.png

Points to remember

- ✓ Understand what is the difference between classification and regression and why it is useful to build models for these tasks
- Understand how a decision tree may be used to make predictions about the class of a test instance
- Understand the key steps in building a decision tree:
  - how to split the instances
  - how to specify the attribute test condition
  - how to determine the best split
  - how to decide when to stop splitting

Points to remember

- Understand the operation and rationale of the k nearest neighbour algorithm for classification
- Understand the advantages and disadvantages of using the k nearest neighbour or decision tree for classification

Acknowledgements

Materials are partially adopted from …

- Previous COMP2008 slides including material produced by James Bailey, Pauline Lin, Chris Ewin, Uwe Aickelin and others
- https://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf
- CS059 - Data Mining - Slides
- http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap4_basic_classification.ppt
- http://people.duke.edu/~kh269/teaching/b351/20evaluatinglearning.pptx