Lecture 6

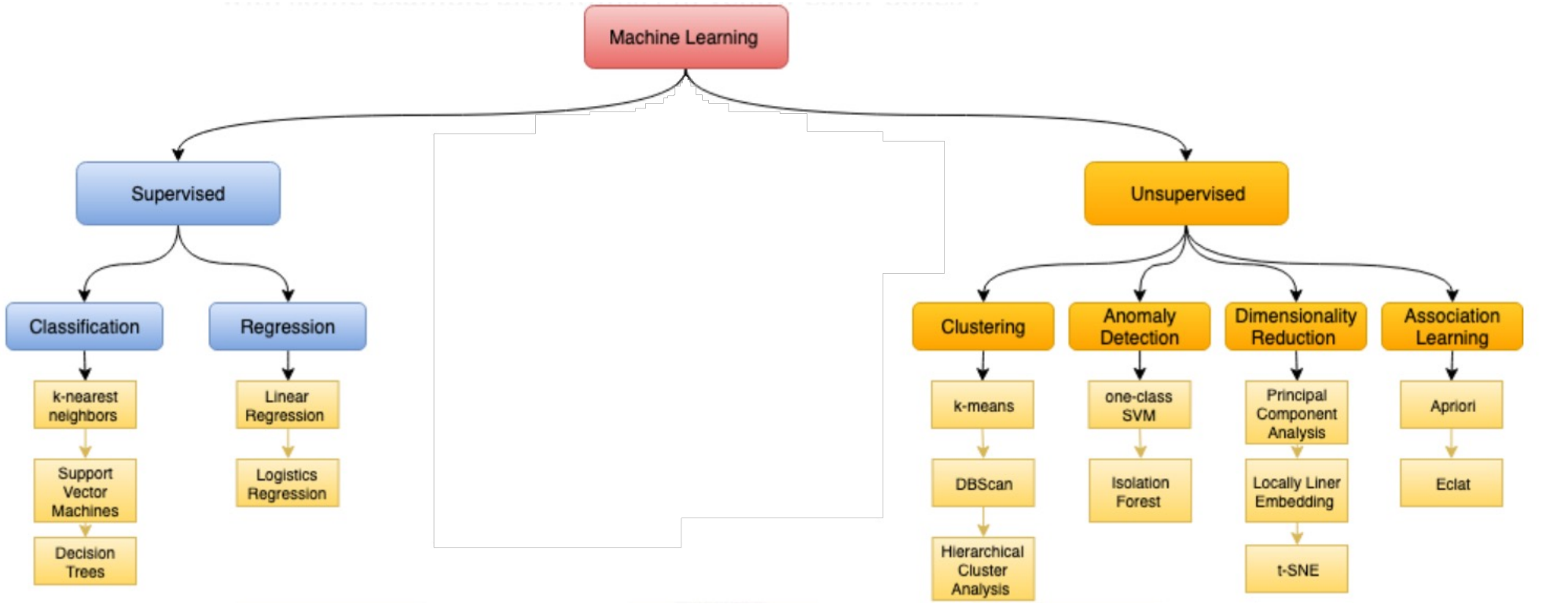
Classification I

Ioanna Miliou, PhD

Senior Lecturer, Stockholm University

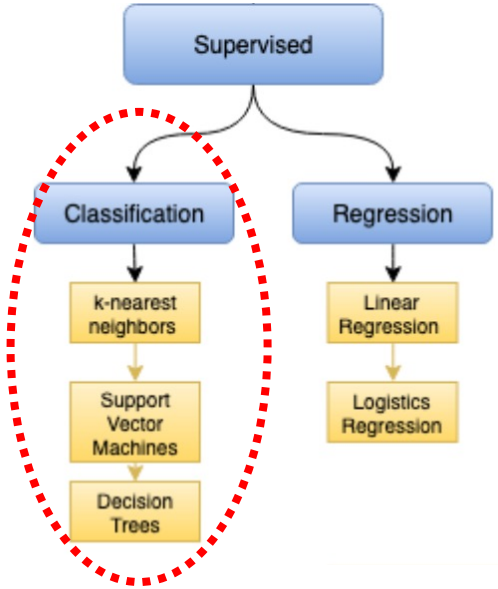
Stockholms universitet

Types of Machine learning in DAMI



Stockholms universitet

Supervised learning



Experience: objects that have been assigned **class labels**

Performance: typically concerns the ability to **classify** new (**previously unseen**) objects

Stockholms universitet

Classification example

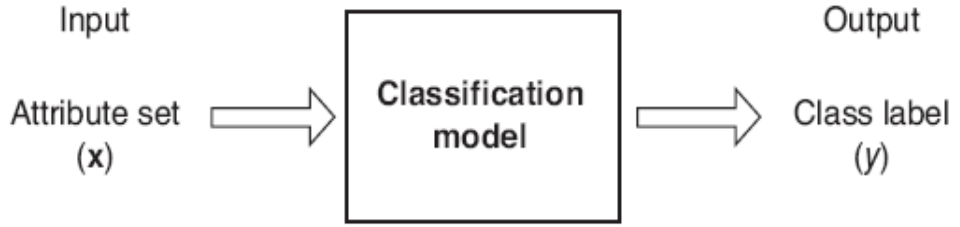
- Borrowers who may **default** on their loans
- Different **attributes/features**
- Learn how to identify **future** borrowers who may also default on their debt

	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

What is Classification?

Classification is the task of ***learning a target function*** f that maps attribute set x to one of the predefined class labels y

	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



The target function f is known as a ***classification model***

Stockholms universitet

Mean Squared Error

- Suppose we have learned a function **f** using some training dataset

- Independent test set $\{x_1, x_2, \ldots, x_n\}$
- Classes of the test set $\{y_1, y_2, \ldots, y_n\}$
- The mean squared error (MSE):

predicted value

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

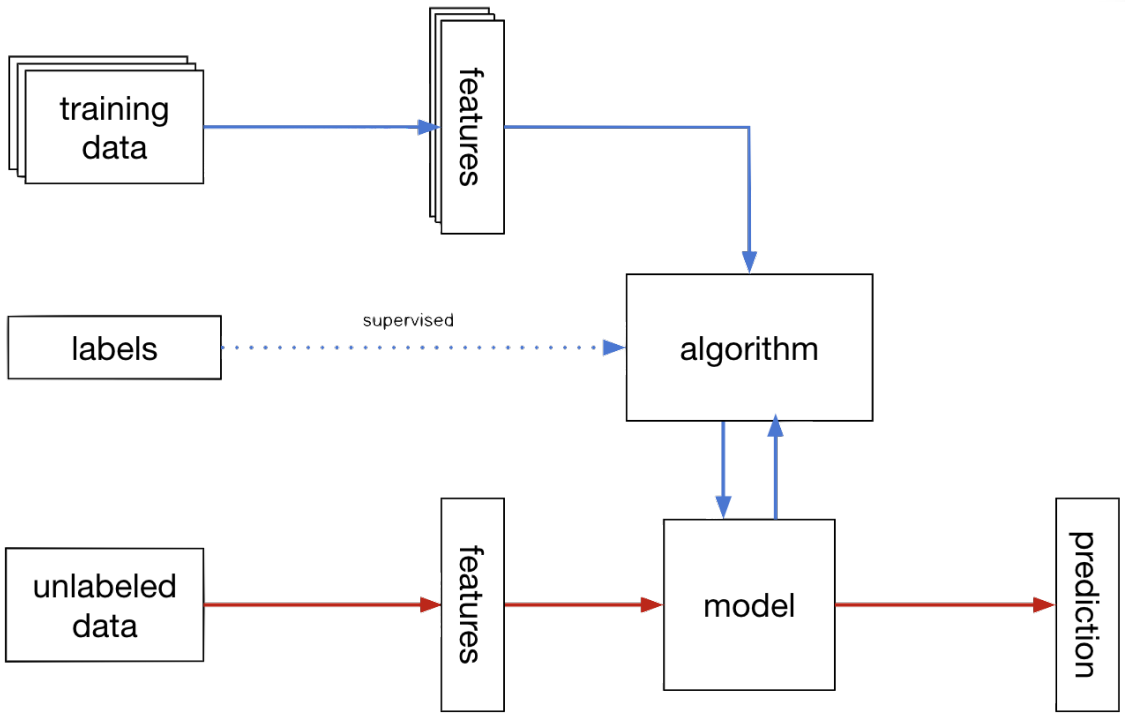
Stockholms universitet

Typical applications

- credit approval
- target marketing
- medical diagnosis
- treatment effectiveness analysis
- identity fraud detection

Stockholms universitet

General approach to classification



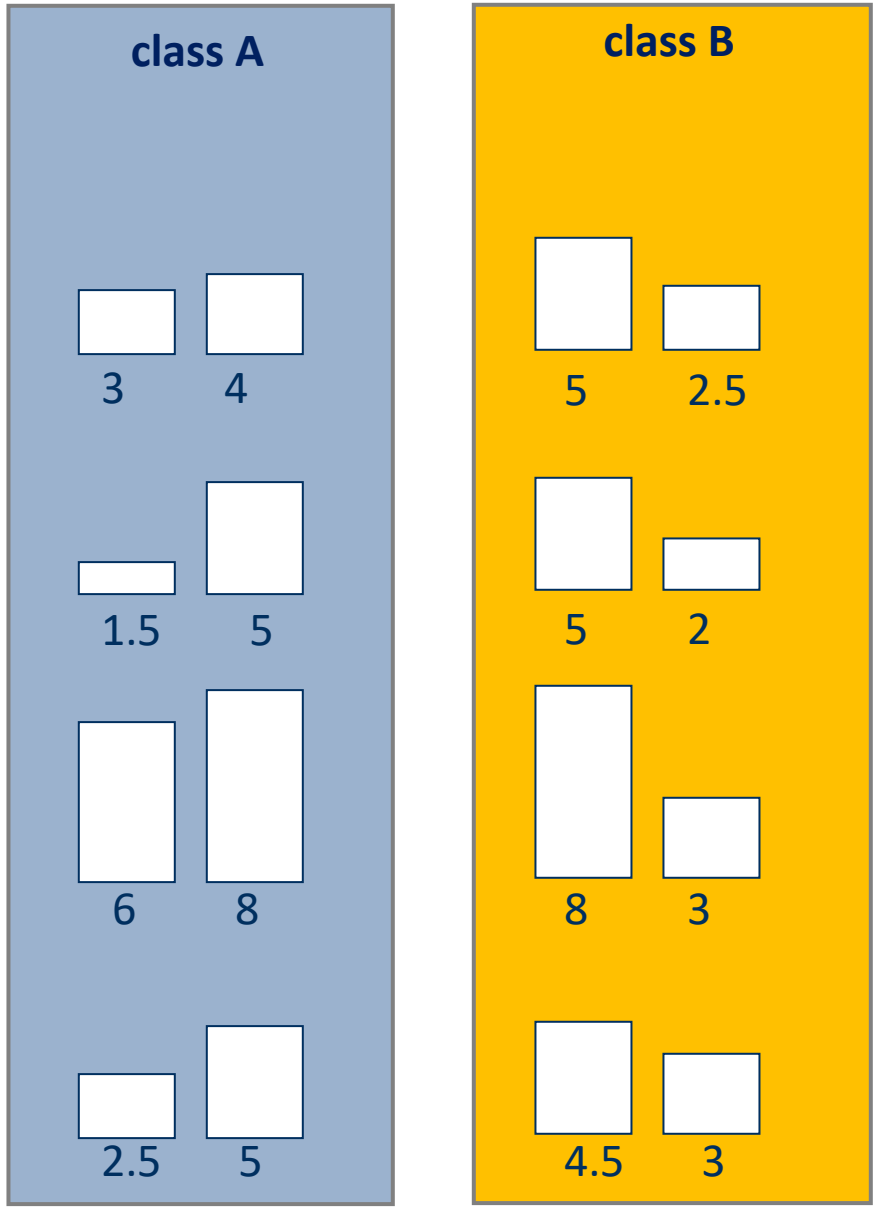
The training set consists of records with known class labels

The training set is used to build a classification model

The classification model is applied to the test set that consists of records with unknown class labels

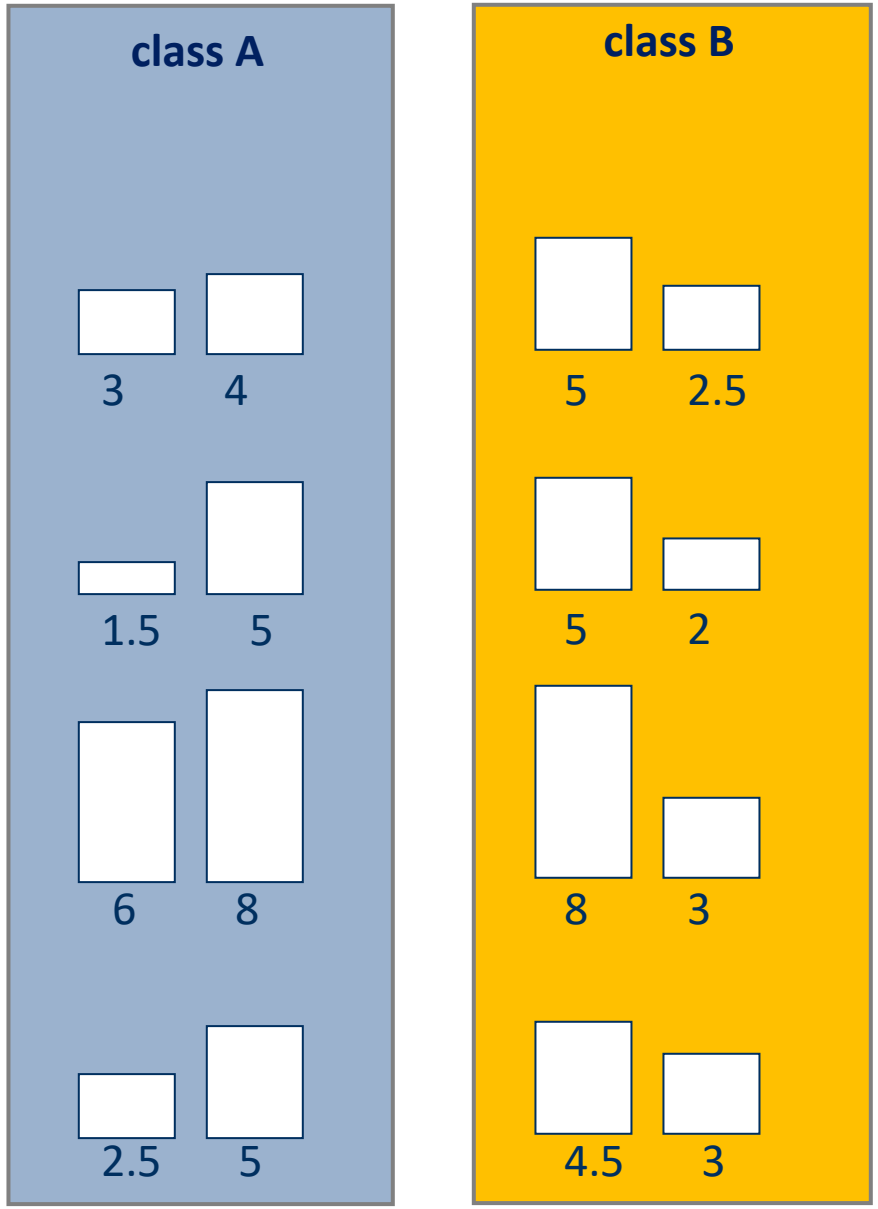
Stockholms universitet

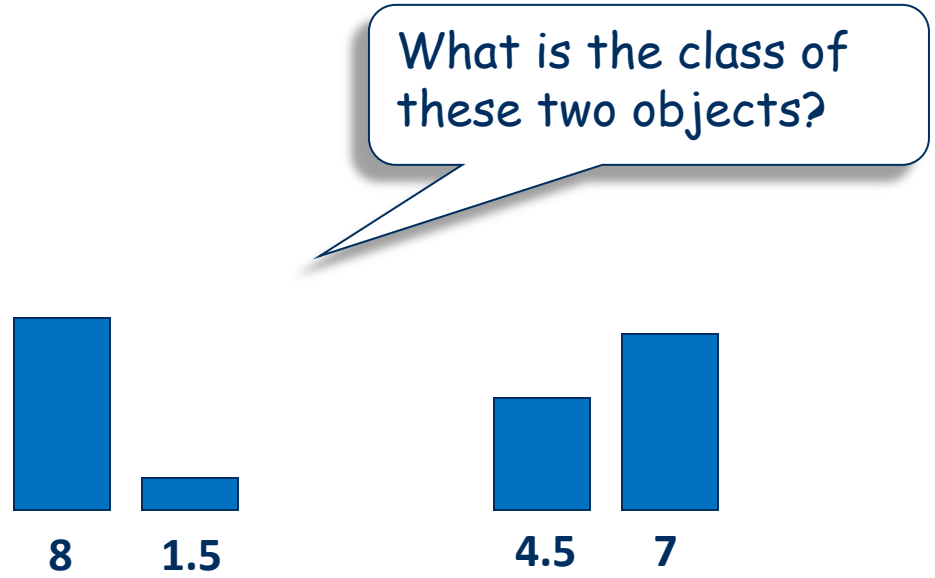
Problem 1



Stockholms universitet

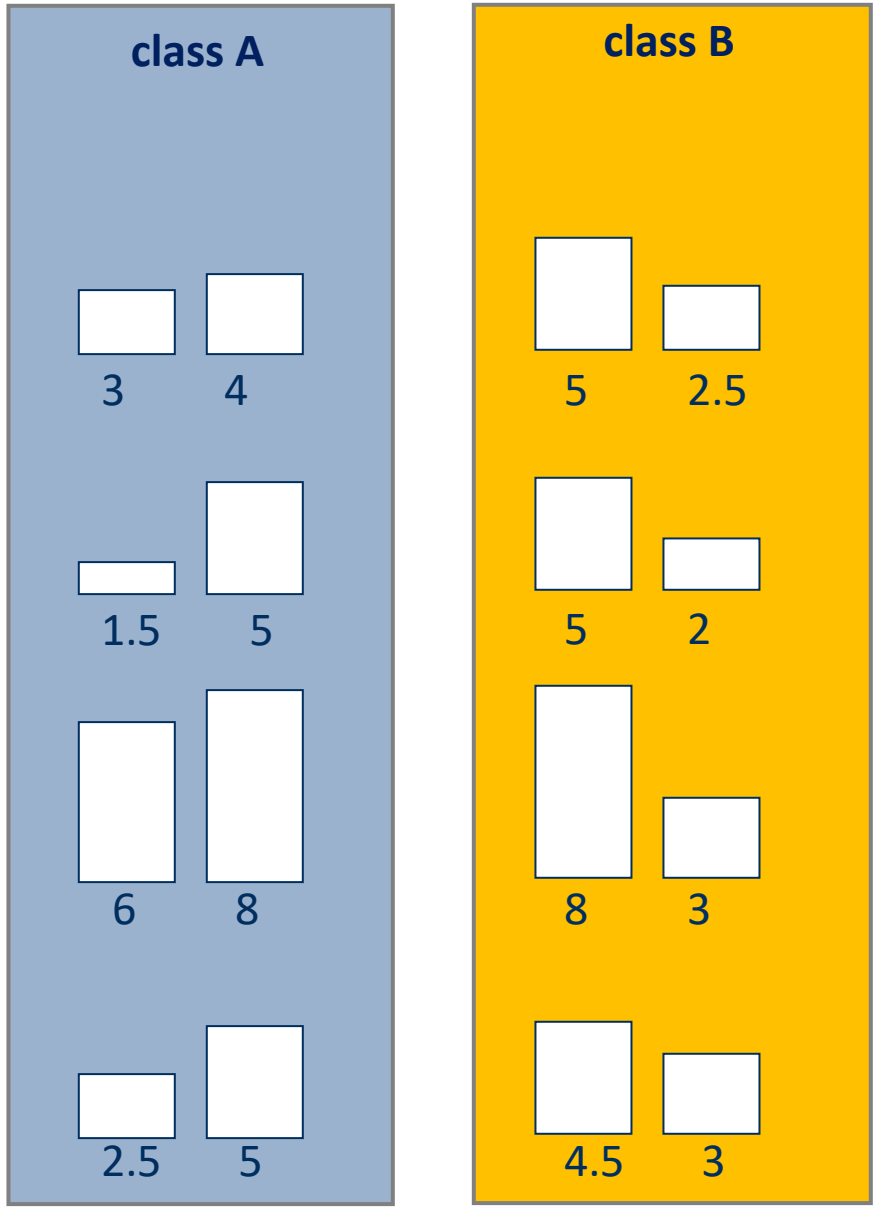
Problem 1



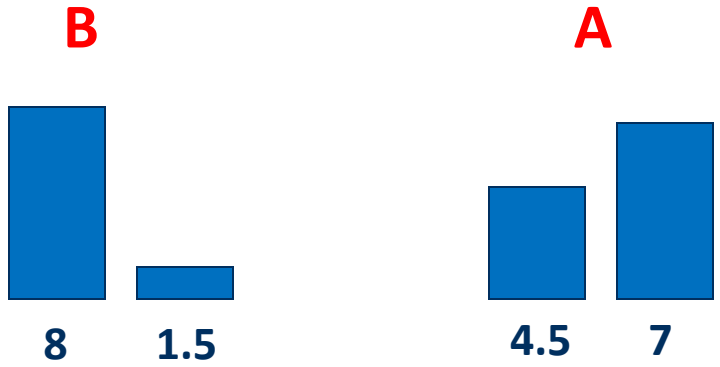


Stockholms universitet

Problem 1



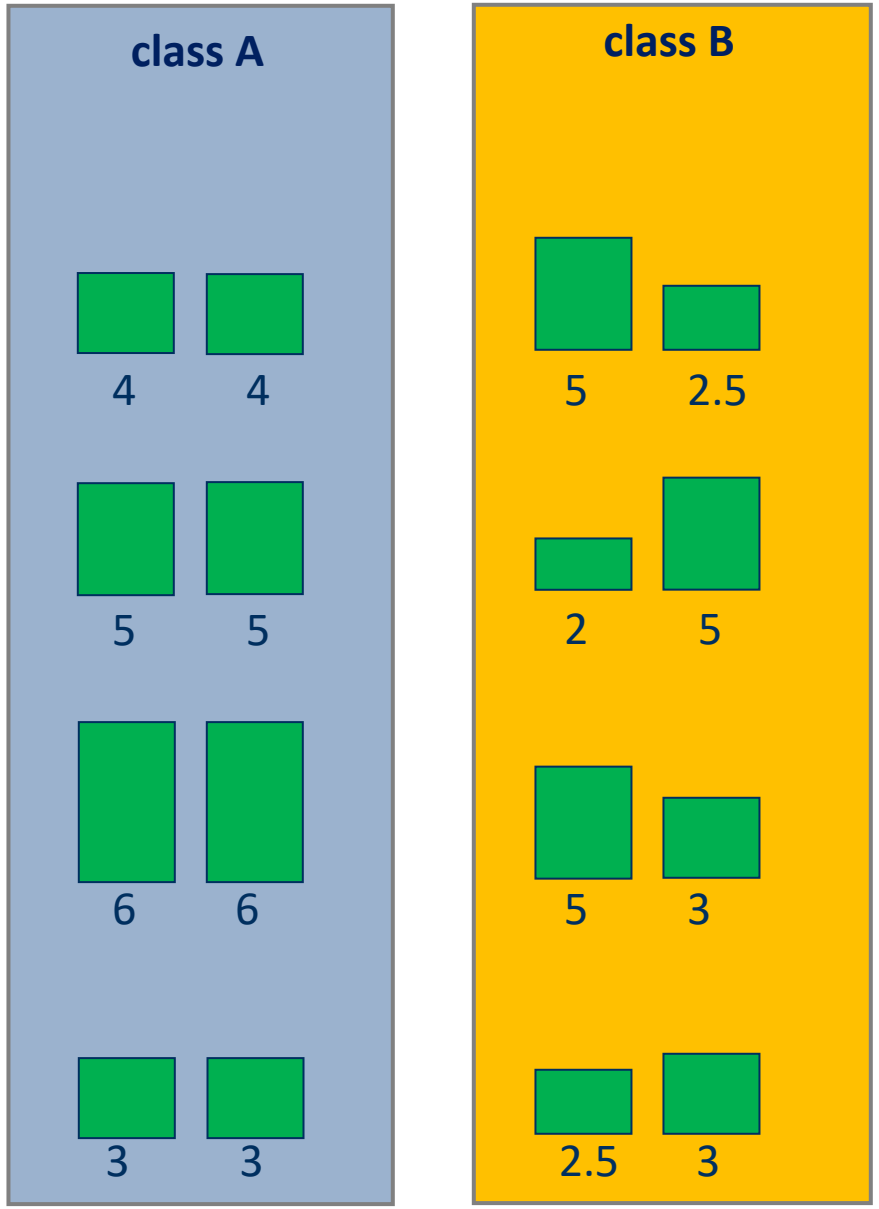
The left one is a B, and the right one is an A!



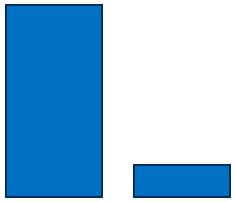
Here is the rule:
If the left bar is smaller than the right bar, it is an A.
Otherwise, it is a B.

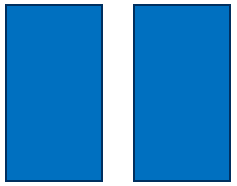
Stockholms universitet

Problem 2



How about these two?

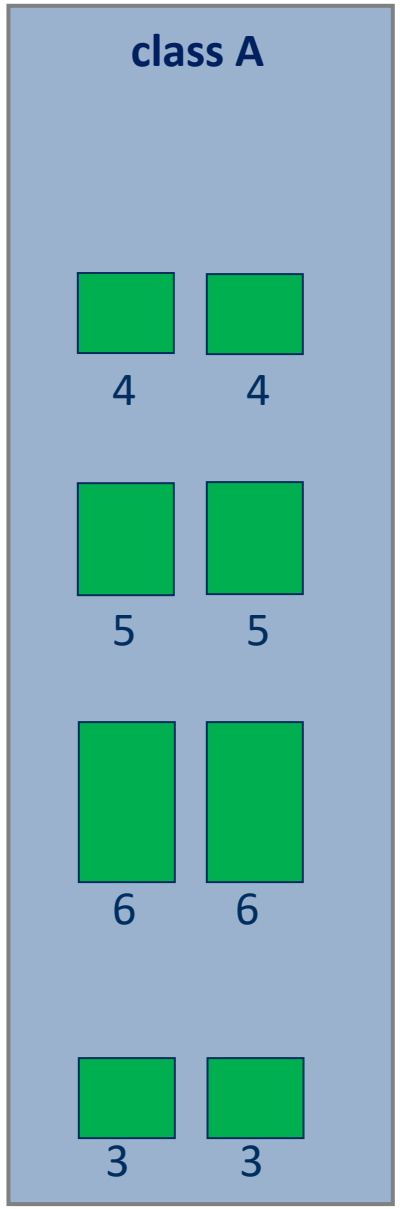


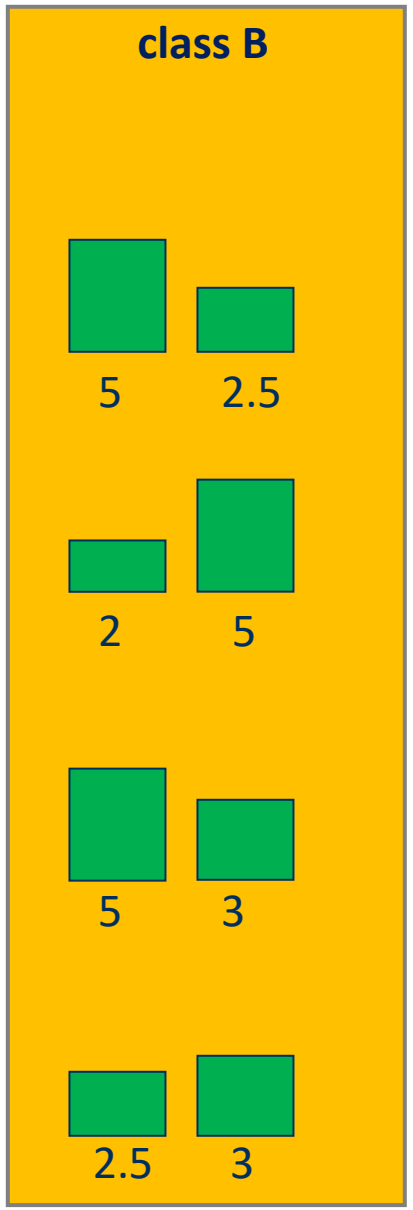


Even I know this one!

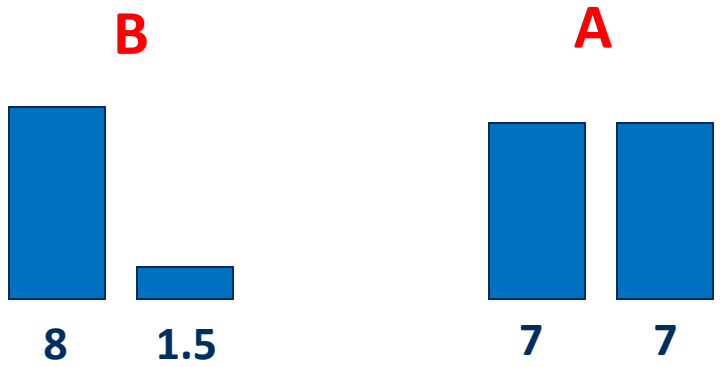
Stockholms universitet

Problem 2





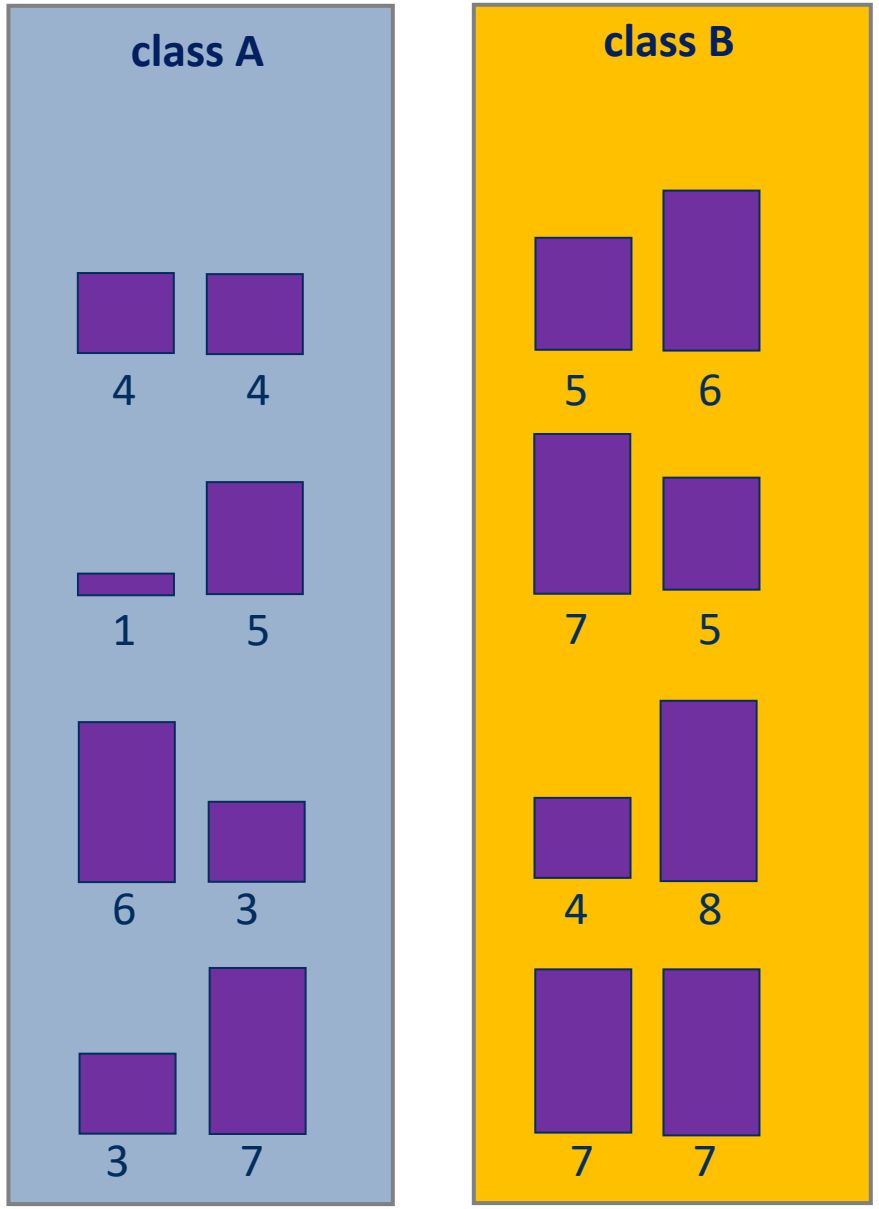
The left one is a B, and the right one is an A!

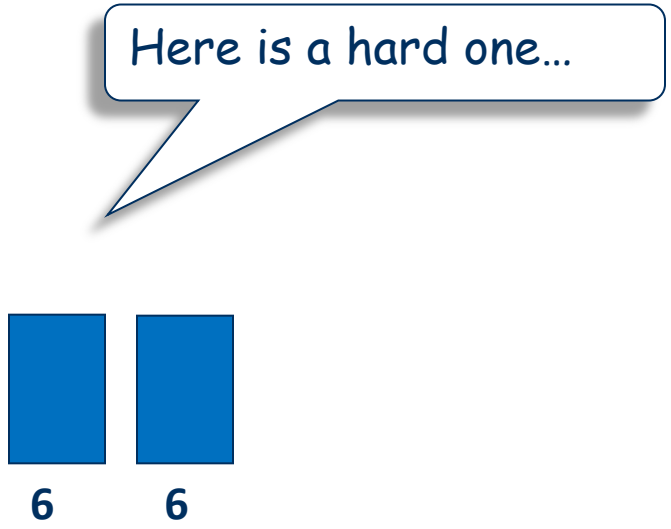


Here is the rule:
If both bars are equal, it is an A.
Otherwise it is a B.

Stockholms universitet

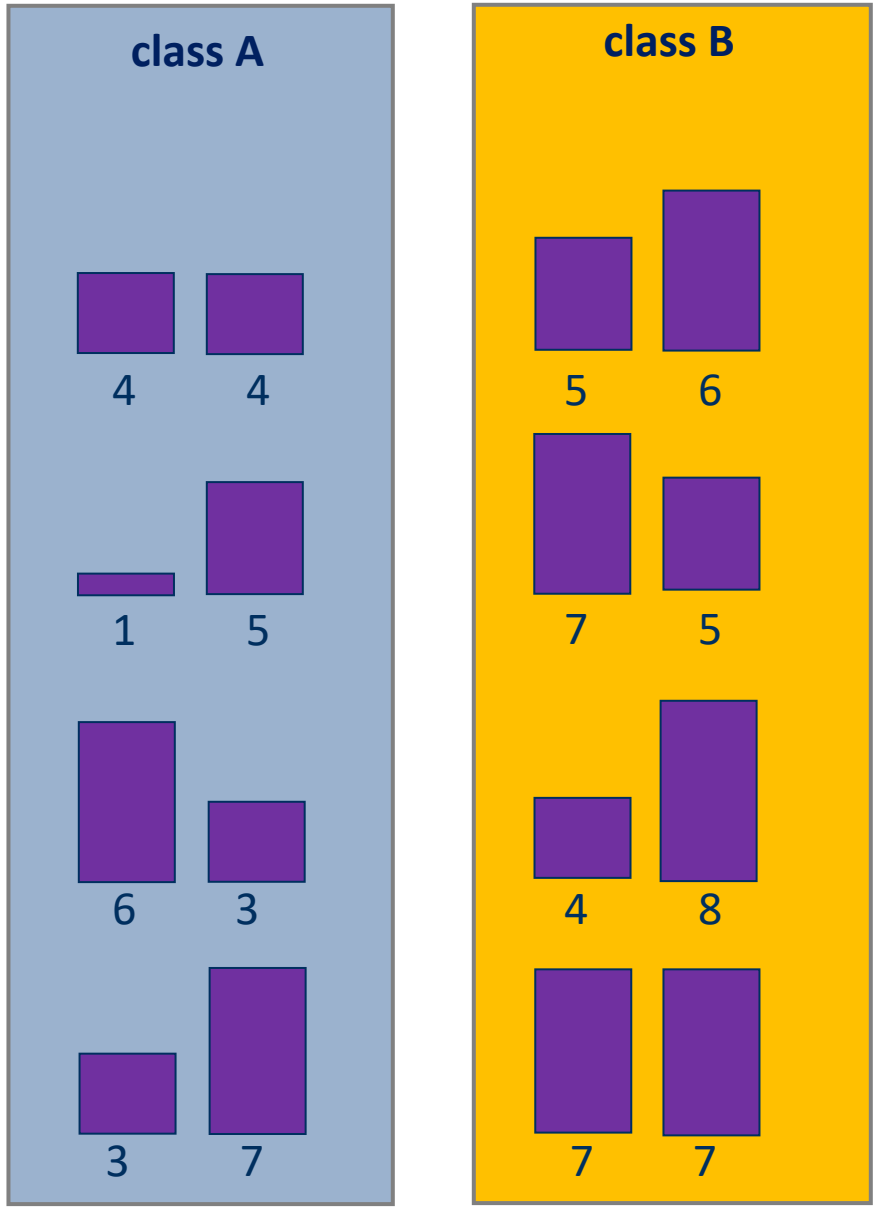
Problem 3





Stockholms universitet

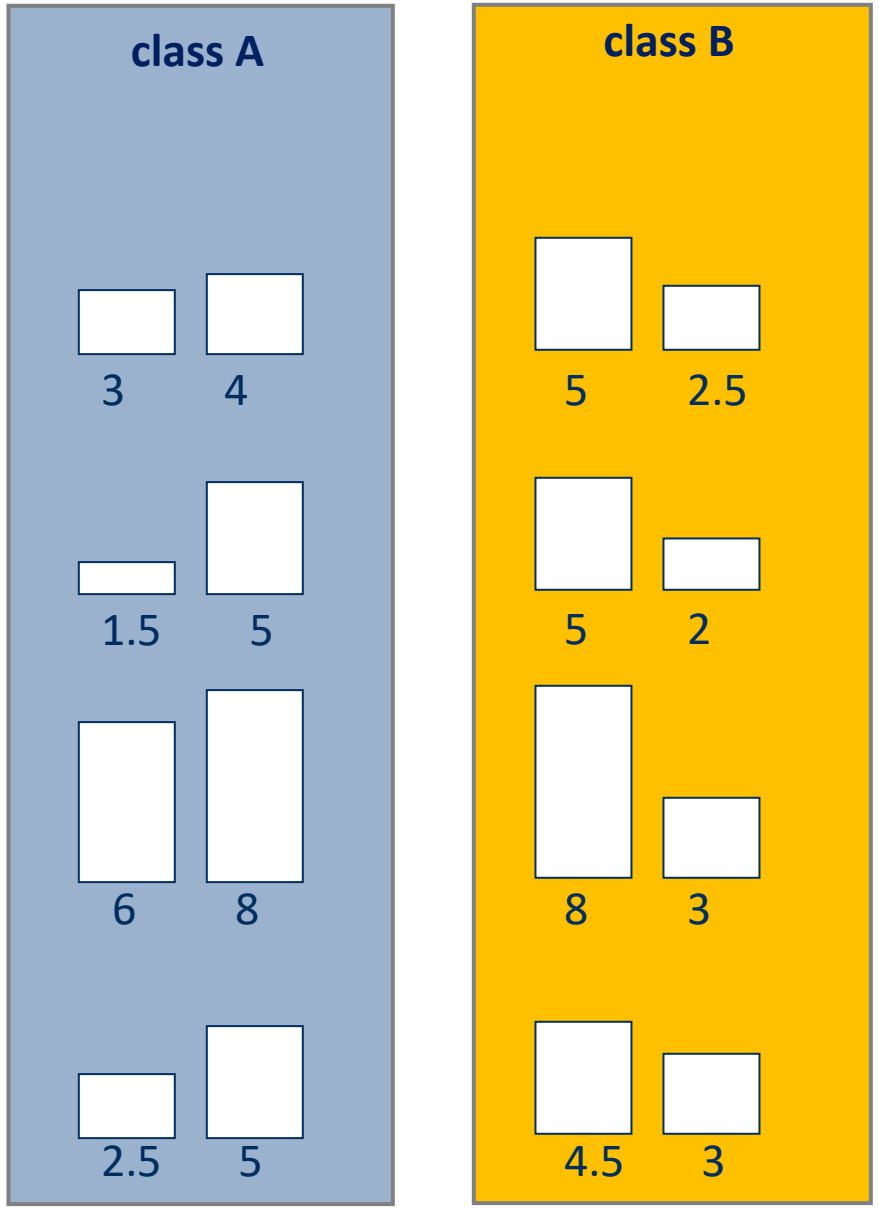
Problem 3

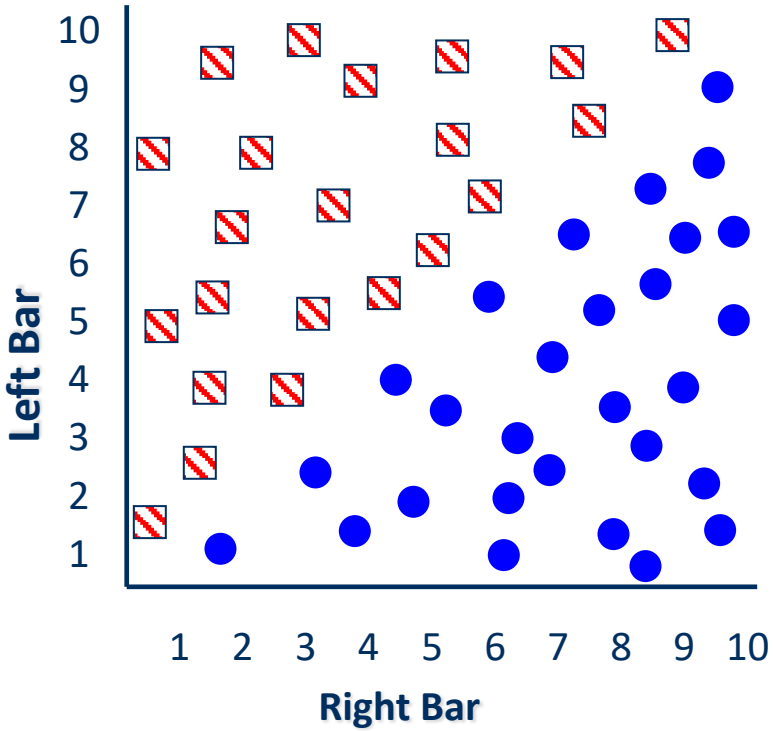


Here is the rule:
If the square of the sum of the two bars is less than or equal to 100, it is an A.
Otherwise, it is a B.

Stockholms universitet

Problem 1

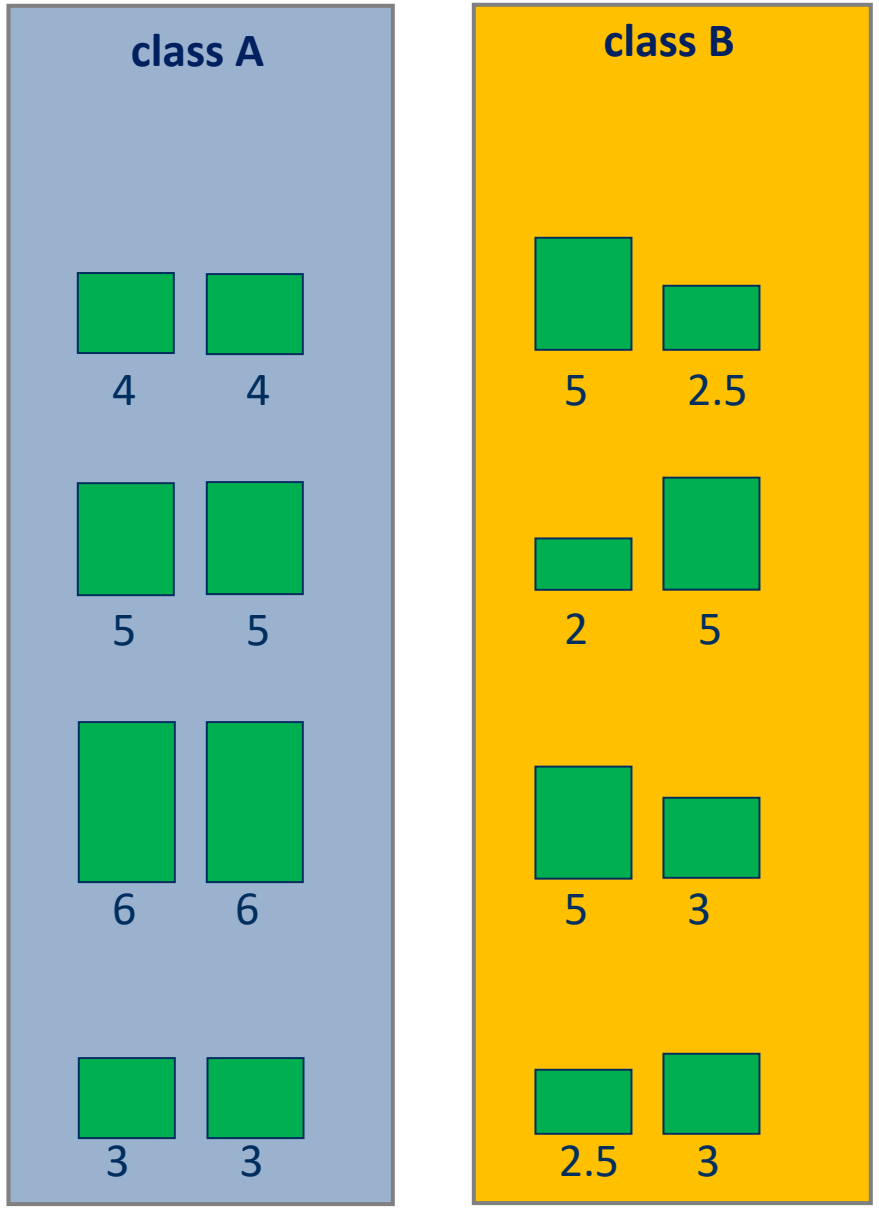


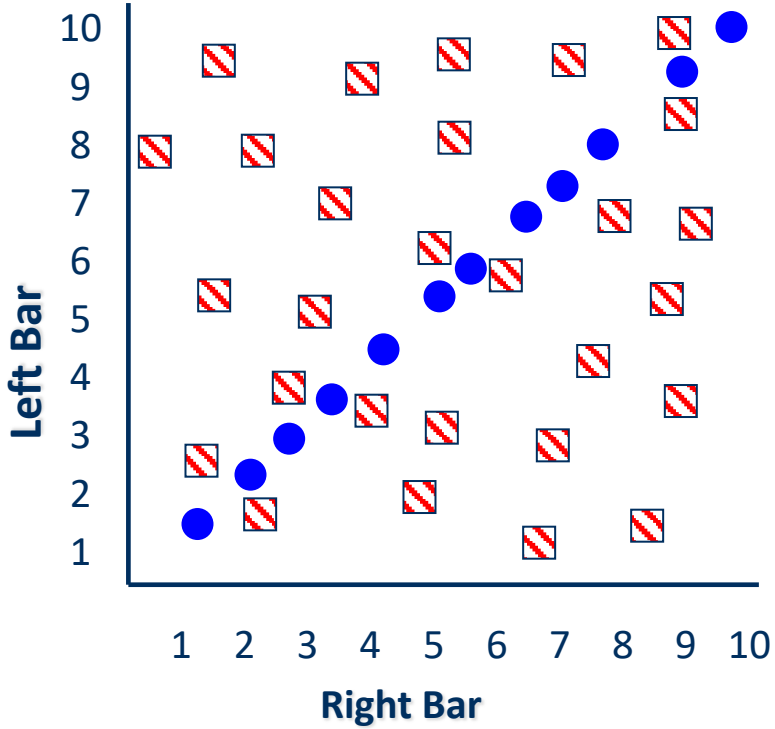


Here is the rule:
If the left bar is smaller than the right bar, it is an A.
Otherwise, it is a B.

Stockholms universitet

Problem 2

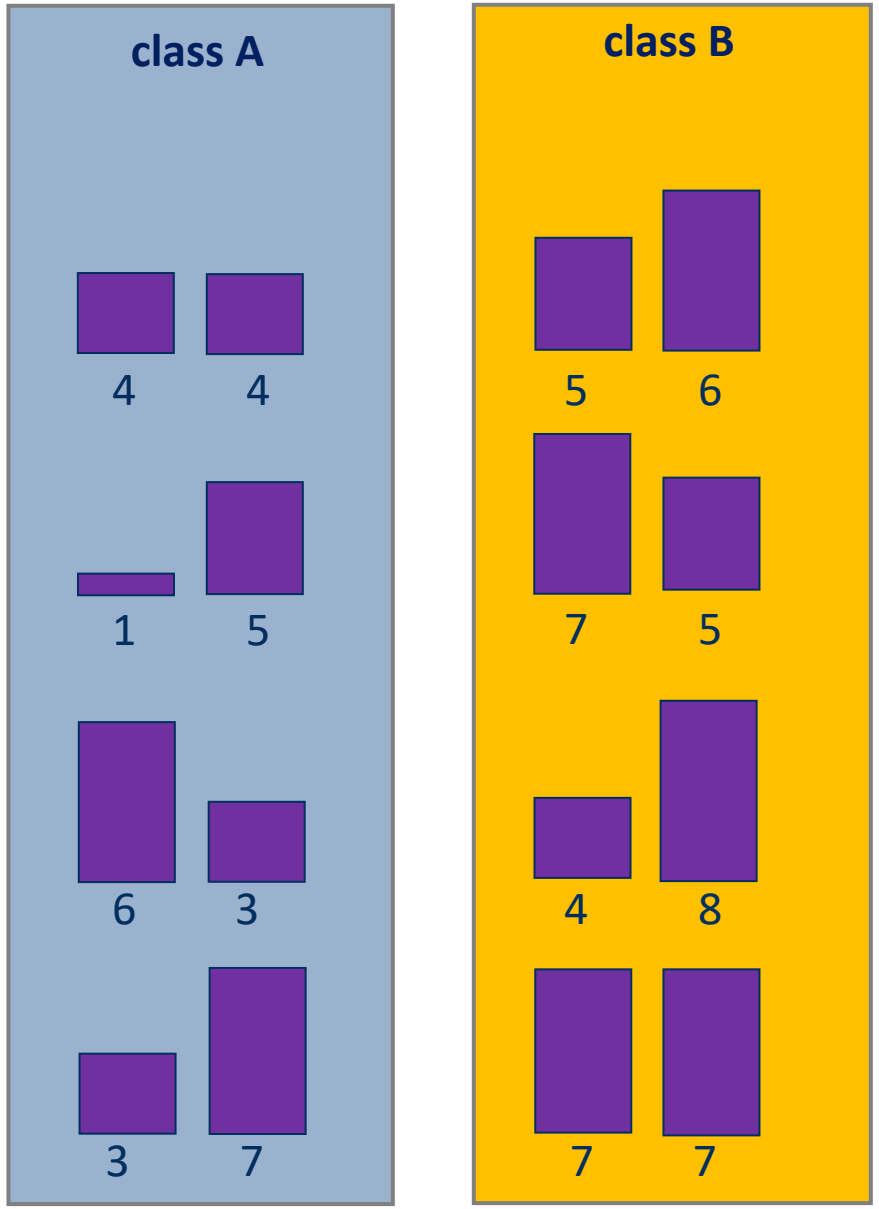


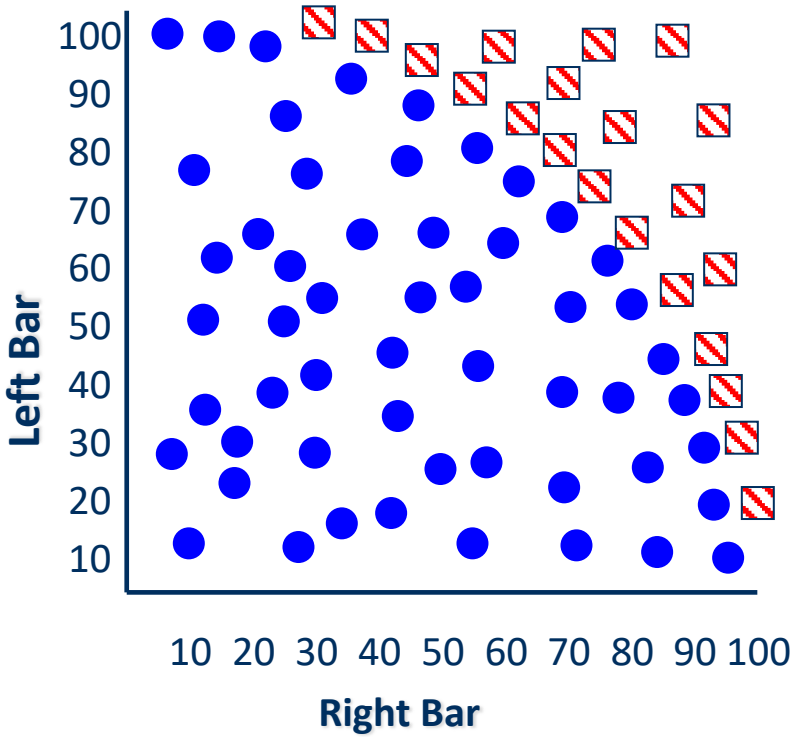


Here is the rule:
If both bars are equal, it is an A.
Otherwise it is a B.

Stockholms universitet

Problem 3





Here is the rule:
If the square of the sum of the two bars is less than or equal to 100, it is an A.
Otherwise, it is a B.

Stockholms universitet

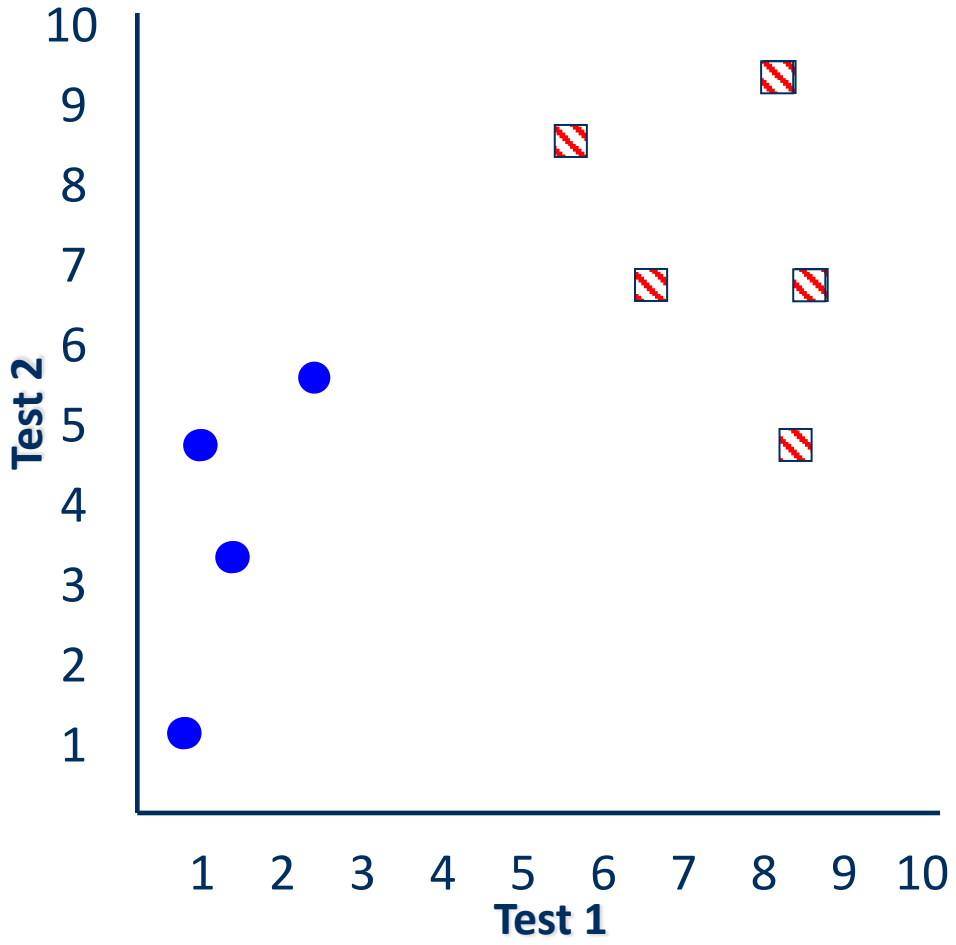
An example: tumor classification

Patients with a suspicious tumor take two tests before they are assigned to a class

- Two features:
  - Test 1
  - Test 2
- Two tumor classes
  - **Benign**
  - **Malicious**

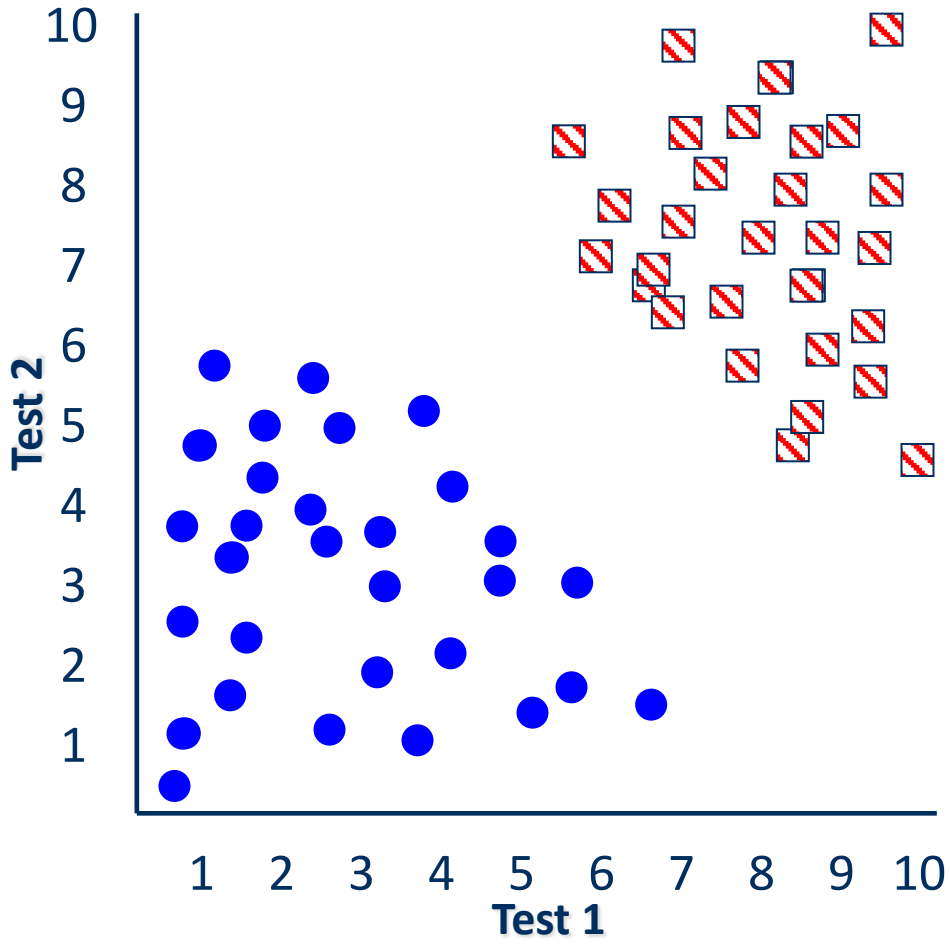
Patient ID	Test 1	Test 2	Tumor Class
1	2.7	5.5	**Benign**
2	8.0	9.1	**Malicious**
3	0.9	4.7	**Benign**
4	1.1	3.1	**Benign**
5	5.4	8.5	**Malicious**
6	6.4	1.3	**Benign**
7	6.1	6.6	**Malicious**
8	0.5	1.0	**Benign**
9	8.3	6.6	**Malicious**
10	8.1	4.7	**Malicious**

An example: tumor classification



Patient ID	Test 1	Test 2	Tumor Class
1	2.7	5.5	Benign
2	8.0	9.1	Malicious
3	0.9	4.7	Benign
4	1.1	3.1	Benign
5	5.4	8.5	Malicious
6	6.4	1.3	Benign
7	6.1	6.6	Malicious
8	0.5	1.0	Benign
9	8.3	6.6	Malicious
10	8.1	4.7	Malicious

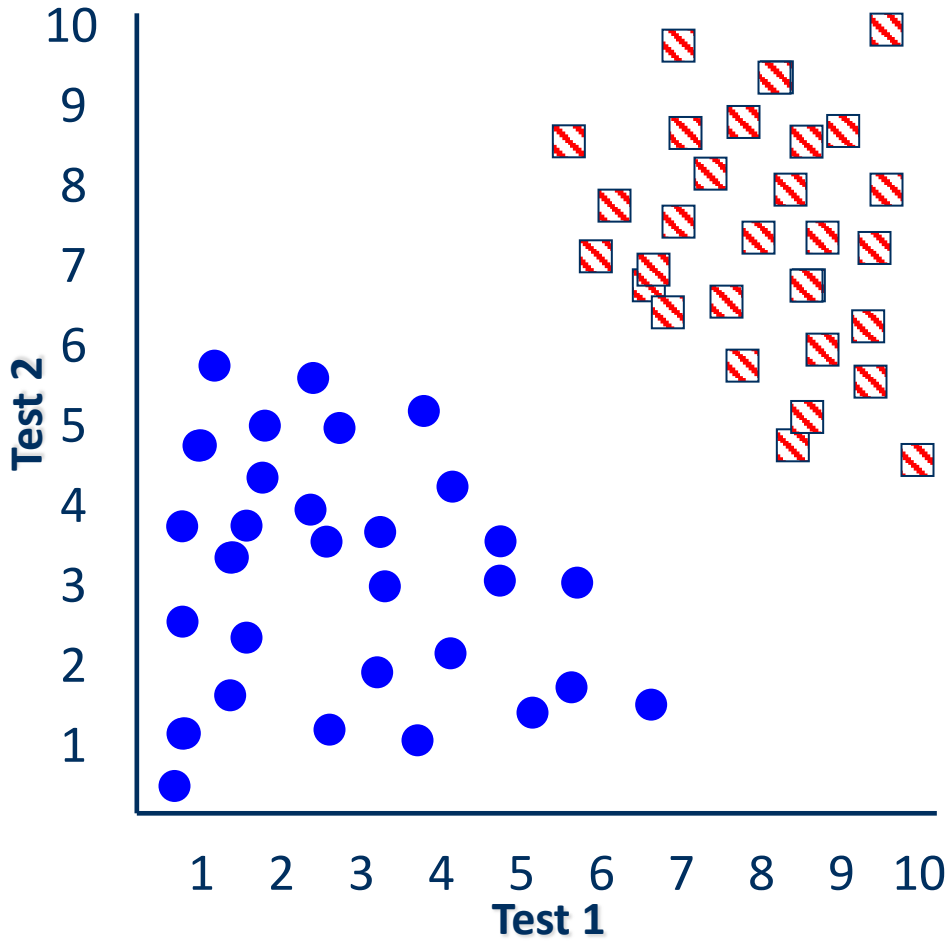
An example: tumor classification



Patient ID	Test 1	Test 2	Tumor Class
1	2.7	5.5	Benign
2	8.0	9.1	Malicious
3	0.9	4.7	Benign
4	1.1	3.1	Benign
5	5.4	8.5	Malicious
6	6.4	1.3	Benign
7	6.1	6.6	Malicious
8	0.5	1.0	Benign
9	8.3	6.6	Malicious
10	8.1	4.7	Malicious

Let us add more examples

An example: tumour classification

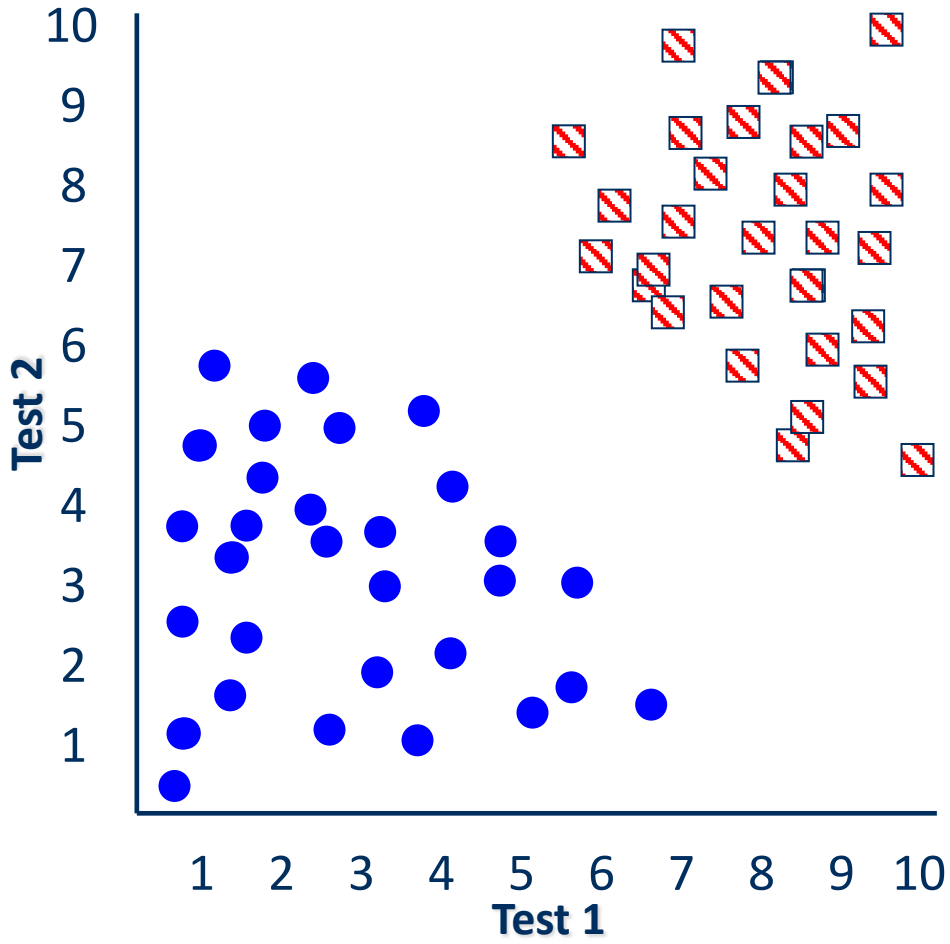


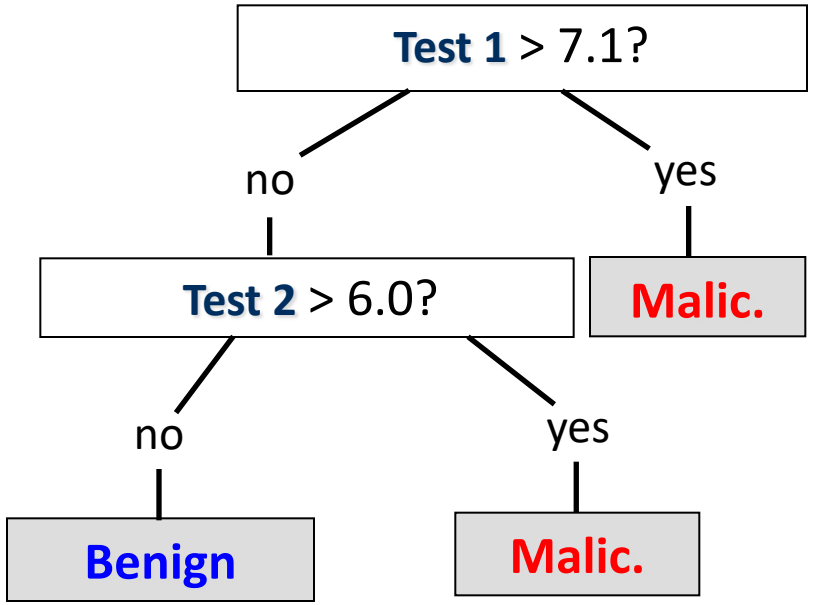
Goal:

Define a **model** that can distinguish **Benign** from **Malicious**

Stockholms universitet

An example: tumour classification

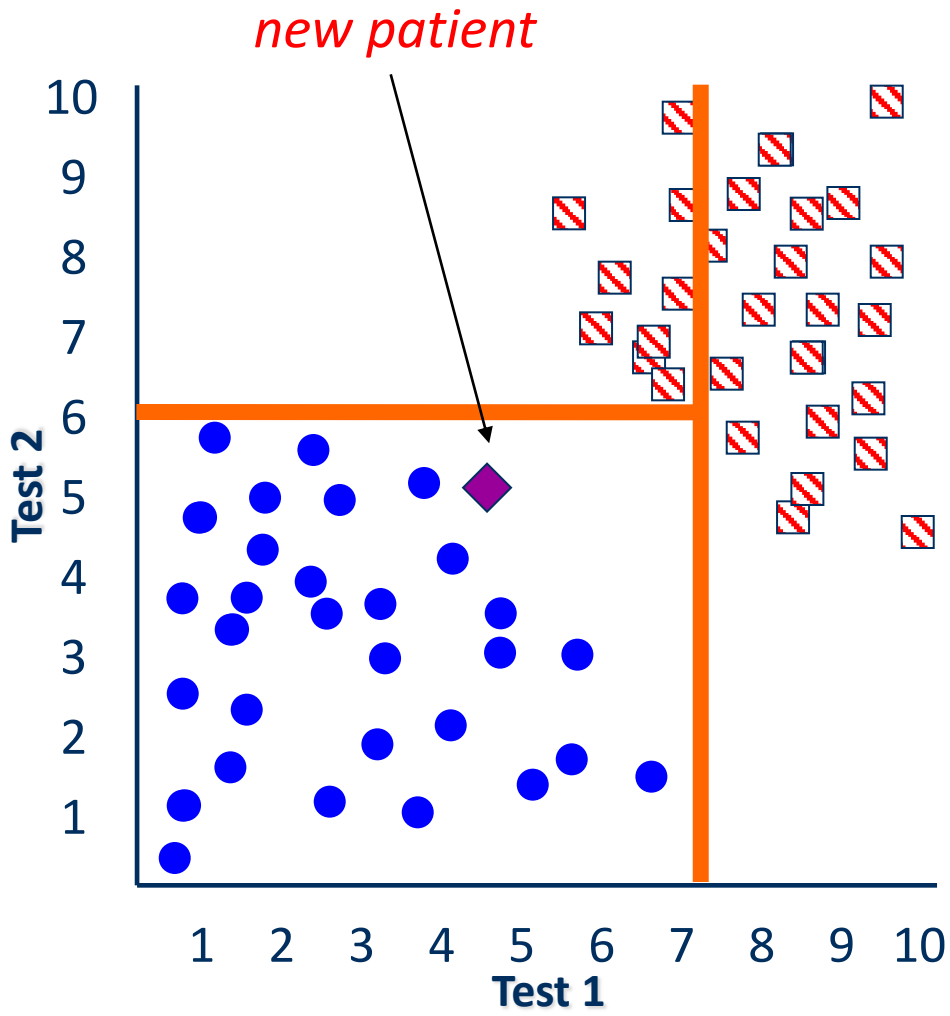


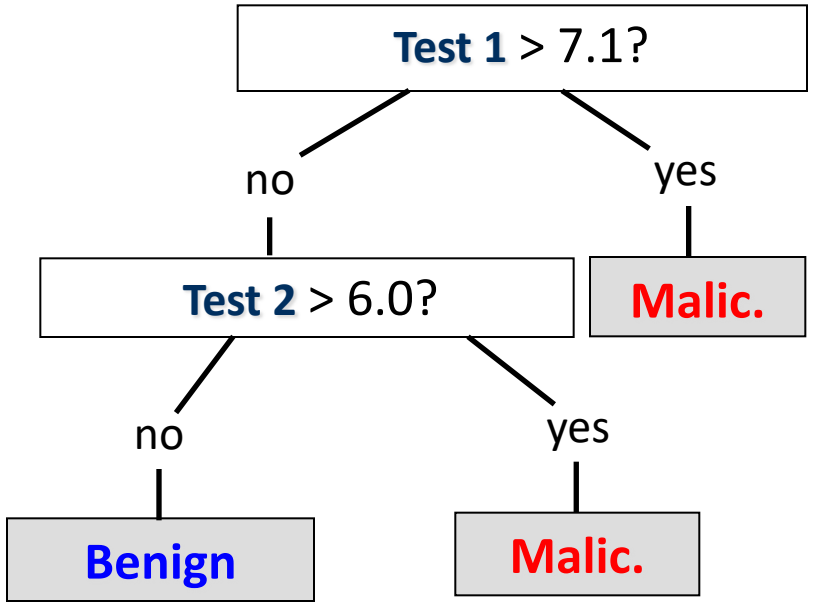


A decision tree

Stockholms universitet

The Decision Tree Classifier



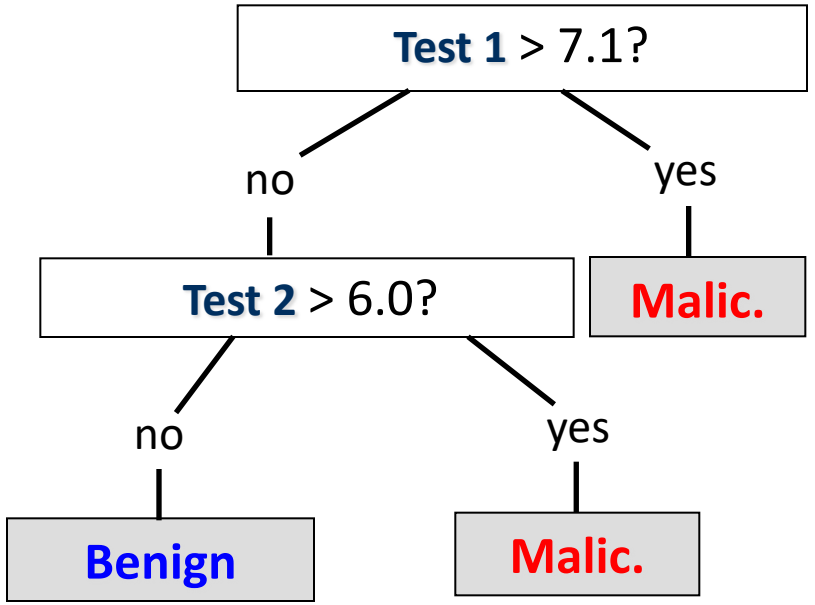


A decision tree

Stockholms universitet

Decision Tree Induction

- 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 or class distribution



Classification by Decision Tree Induction

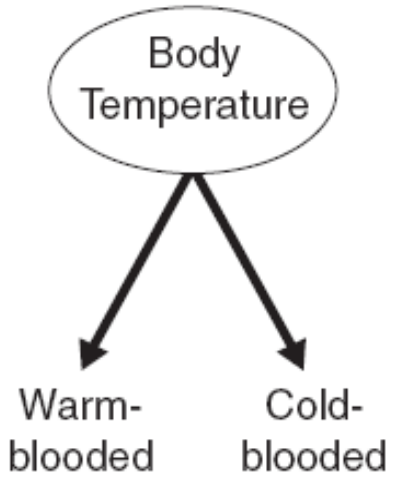
- The decision tree generation consists of **two phases**:
  - Tree construction
    - At the start, all the training examples are at the **root**
    - Partition examples **recursively** based on selected attributes
    - All examples that **satisfy the condition** on the selected attribute move to that branch
  - Tree pruning
    - Identify and **remove** branches that reflect noise or outliers
- Classifying an **unknown sample**:
    - **Test** the attribute values of the sample against the decision tree

Design Issues

- How should the training records be split?
- How should the splitting procedure stop?

Splitting methods

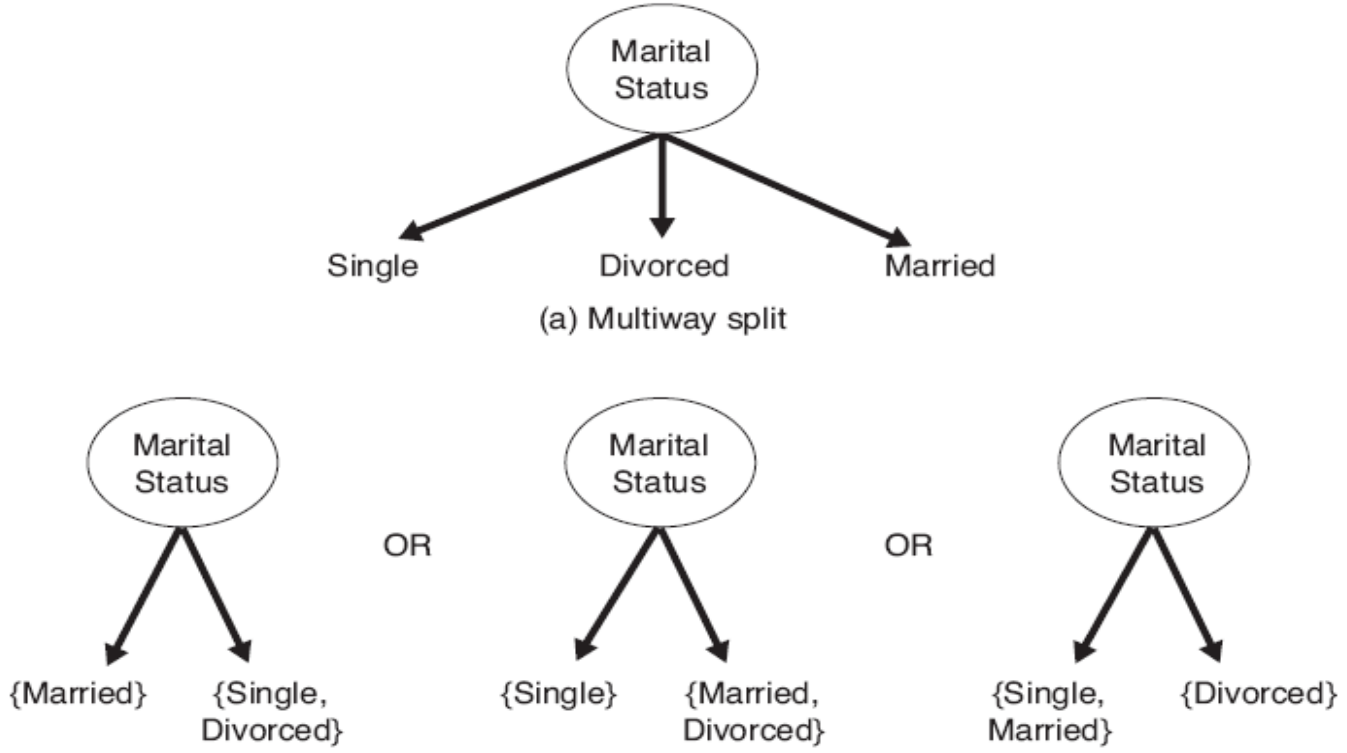
- Binary attributes



Stockholms universitet

Splitting methods

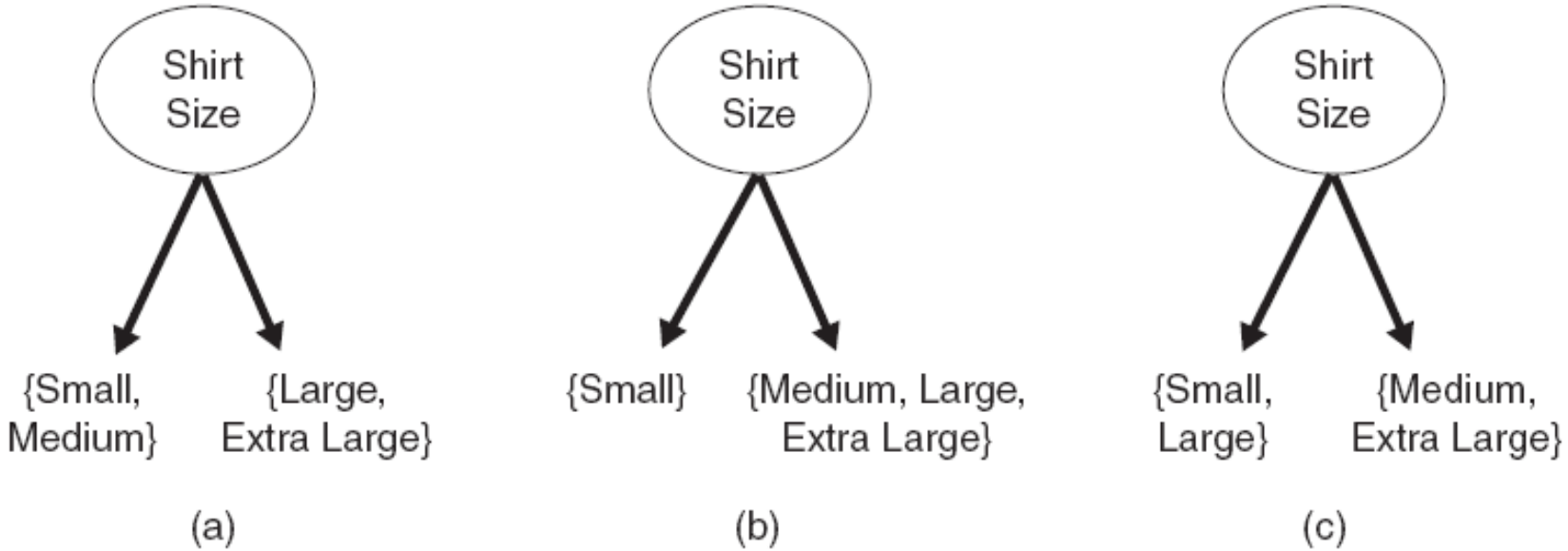
- Nominal or categorical attributes



Stockholms universitet

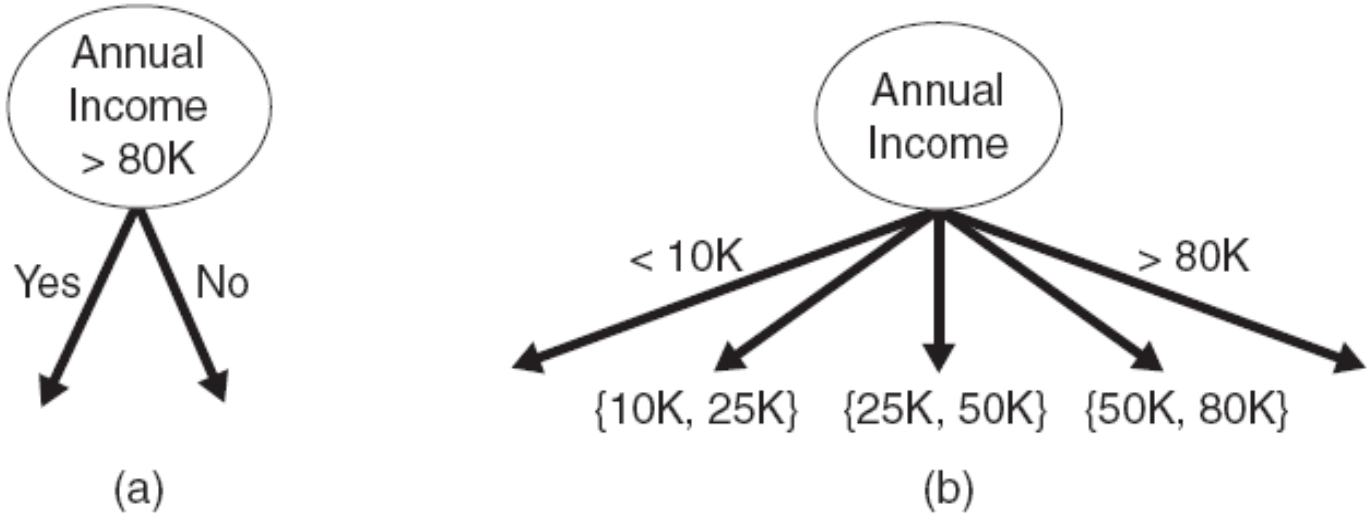
Splitting methods

- Ordinal attributes



Splitting methods

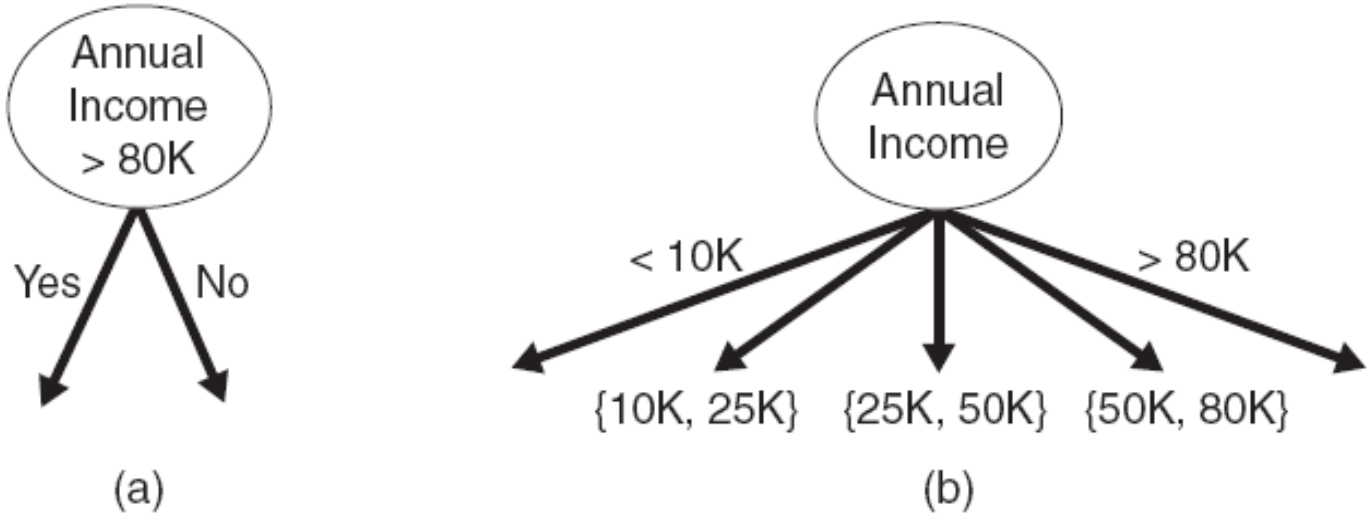
- Continuous attributes



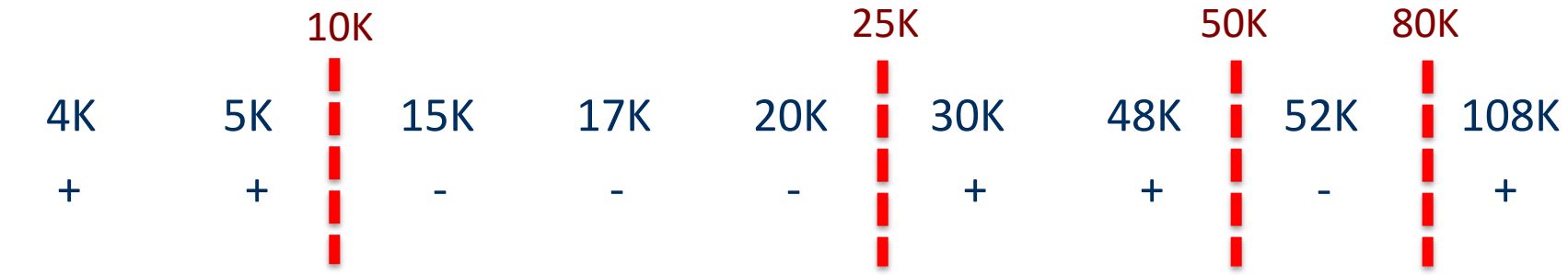
Stockholms universitet

Splitting methods

- Continuous attributes

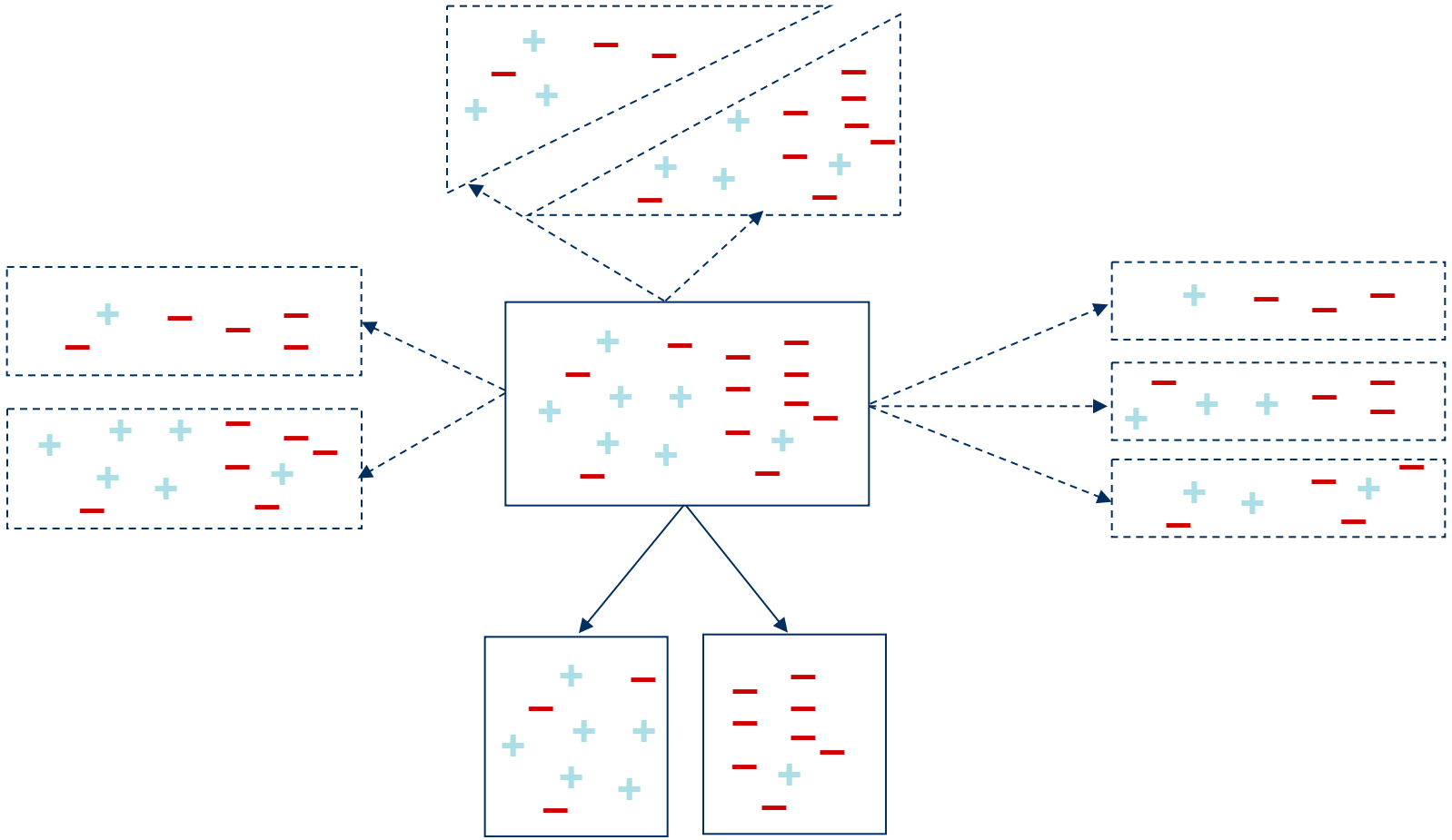


- Sort the attribute values in ascending order
- Try splitting at each position where the class label changes:



Stockholms universitet

Choice of attribute



An *impurity* measure should be defined: it should assess how "pure" are the splits

Impurity

- A split is pure if, after the split, for all branches, all the examples choosing a branch, belong to the same class
- We want many items in pure sets
- Many different purity measures:
  - Information Gain (IG)
  - Gini Index

Impurity

- A split is pure if, after the split, for all branches, all the examples choosing a branch, belong to the same class
- We want many items in pure sets
- Many different purity measures:
  - **Information Gain (IG)**
    - It measures the reduction in entropy or surprise by splitting a dataset according to a given value of an attribute
  - Gini Index

Impurity

- A split is pure if, after the split, for all branches, all the examples choosing a branch, belong to the same class
- We want many items in pure sets
- Many different purity measures:
  - Information Gain (IG)
  - **Gini Index**
    - It measures the probability that a data example will be mislabeled when random labelling is done

Entropy

- Entropy quantifies how much information there is in a random attribute, or more specifically, its probability distribution
- The expected information (entropy) needed to classify an example in our training dataset ***D*** is defined as:

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

where ***m*** is the number of classes, and p_i is the probability that an example in ***D*** belongs to class label C_i:

$$p_i = \frac{|C_{i,D}|}{|D|}$$

where $C_{i,D}$ are the examples in ***D*** that belong to class C_i

Entropy

- Information theory: **entropy** is the minimum number of bits needed to encode the class label of an example!

- So, if we have two classes, we can use two bits:
  - Class A: 1
  - Class B: 0

- So, if all examples are of class A, what is the **entropy**?

Entropy

- So, if all examples are of class A, what is the **entropy**?
  - Probability of class A: $p_1 = 1$
  - Probability of class B: $p_2 = 0$

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

$$Info(D) = -1log_2(1) - 0log_2(0) = 0$$

Makes sense! If all classes are of the same label, there is no need to encode anything!

Entropy

- What if half of the examples are of class A and half are of class B?
  - Probability of class A: $p_1 = 0.5$
  - Probability of class B: $p_2 = 0.5$

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

$$Info(D) = -0.5log_2(0.5) - -0.5log_2(0.5) = 0.5 + 0.5 = 1$$

Makes sense! This is the most not-pure setting! We need 1 bit per class (0 or 1).

Max Entropy

- For *m* classes, the max entropy is $\log_2(m)$
  - Two classes: 1
  - Four Classes: 2
  - Eight Classes: 3
  - 16 classes: 4

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

Information Gain

- Suppose that attribute A is chosen for a split
- ***D*** is split into ***v*** partitions based on A: $D_1, D_2, \ldots, D_v$
- Information needed to classify ***D*** :

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} Info(D_j)$$

- Information gained by branching on attribute A:

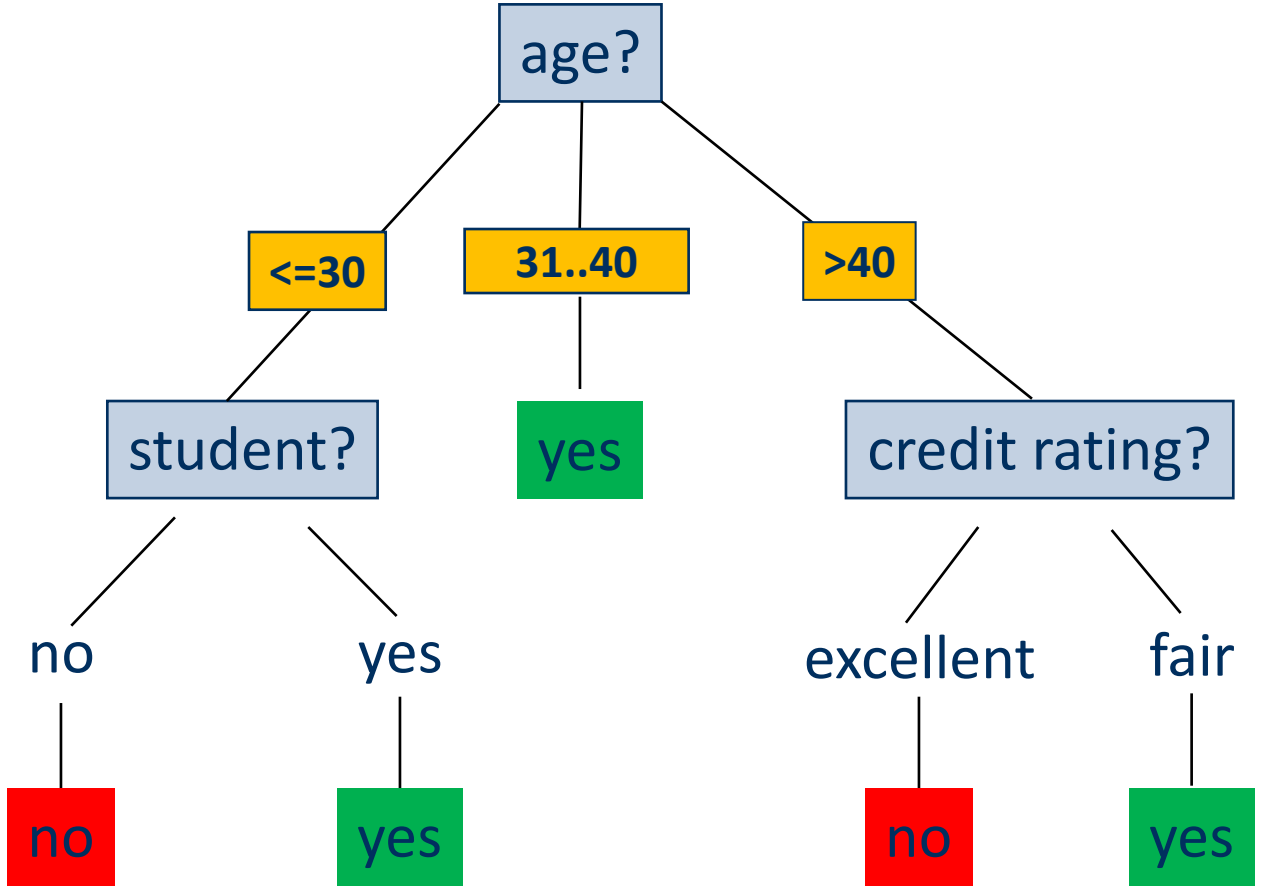
$$Gain(A) = Info(D) - Info_A(D)$$

Example

class

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

Output: A Decision Tree for “buys_computer”



Stockholms universitet

Attribute Selection Measure: **Information Gain**

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

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

age	pos	neg
<=30	2	3
31…40	4	0
>40	3	2

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

$$Info(D) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

Attribute Selection Measure: **Information Gain**

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} Info(D_j)$$

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

Let’s select first as attribute the age:

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

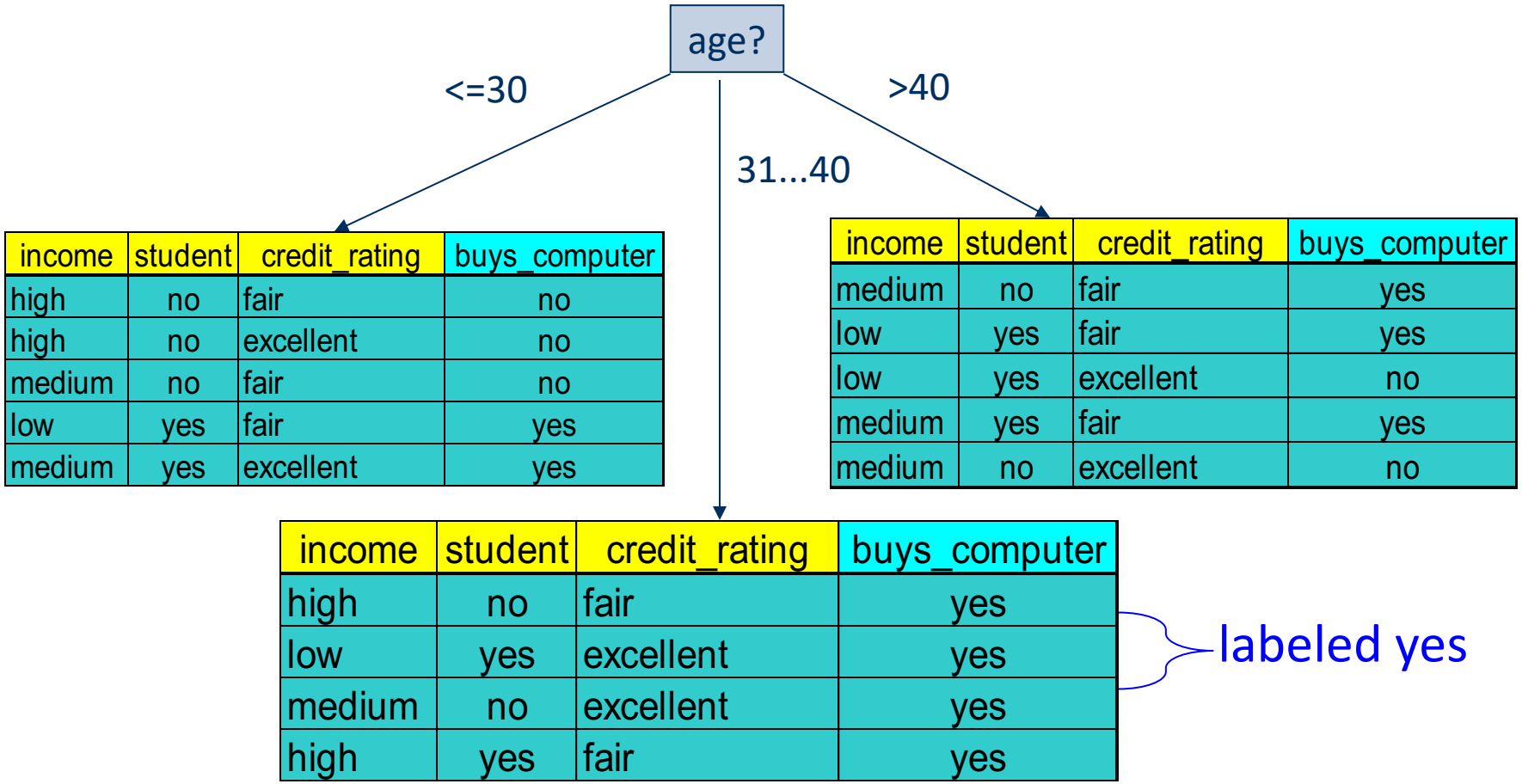
Similarly for the other attributes:

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

Splitting the samples using *age*



Stockholms universitet

Gini index

- If a data set ***D*** contains examples from ***m*** classes, gini index, *gini*(*D*) is defined as:

$$gini(D) = 1 - \sum_{j=1}^{m} p_j^2$$

where p_j is the relative frequency of class *j* in ***D***

- Minimum Gini index?
  - when the node is ***pure***: all examples are of the **same class**
  - **$Gini_{min}$** = $1 - (1^2) = 0$
- Maximum Gini index?
  - probability of the two classes are the same
  - **$Gini_{max}$** = $1 - (0.5^2 + 0.5^2) = 0.5$

Gini index

- If a data set ***D*** is split on A into two subsets D_1 and D_2, the Gini index is defined as follows:

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

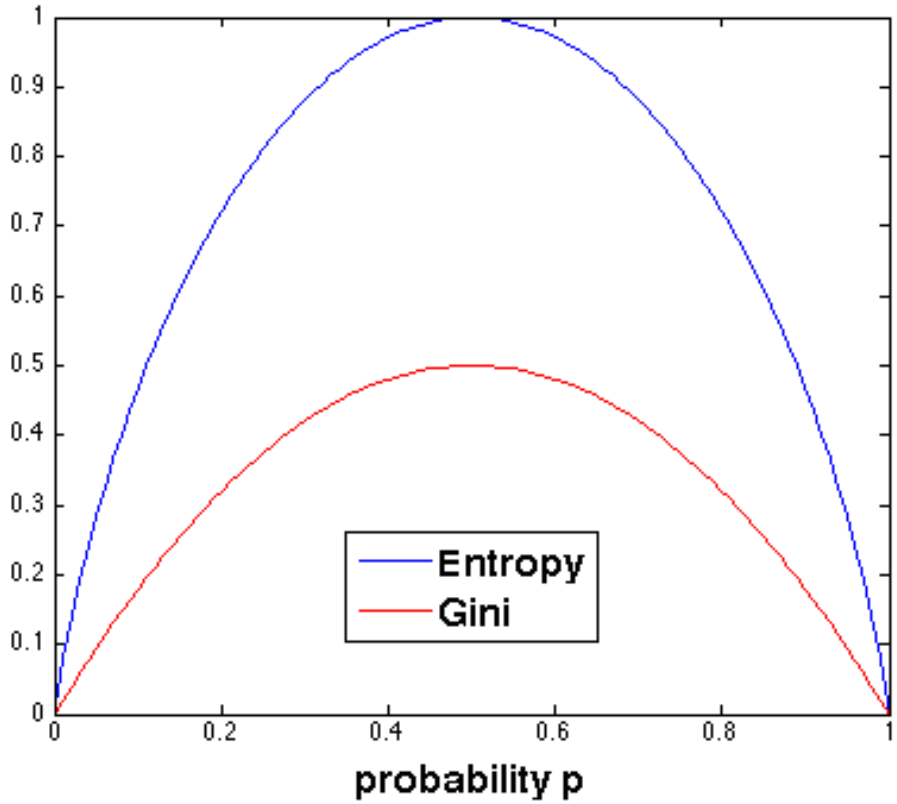
- The attribute that provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node

Entropy vs Gini

- Consider the 2-class problem:
  - Class A: probability p
  - Class B: probability 1 – p

$$Info(D) = -p\log_2(p) - (1-p)\log_2(1-p)$$

$$gini(D) = 1 - p^2 - (1-p)^2$$



Stockholms universitet

Comparing attribute selection measures

- The two measures, in general, return good results but
  - Both favor multi-valued attributes
  - Gini may have difficulties when # of classes is large
  - Gini tends to favor test sets that result in large and equal-sized partitions and high purity in both partitions
  - Gini is computationally faster that entropy (as it does not have to compute logarithms)

Is minimizing impurity enough?

- **Information gain** favors attributes with a large number of values
- What happens if you choose the ID attribute as the split attribute?
  - All nodes are pure! Each node contains only one example!
- A test condition with a large number of outcomes may not be desirable
  - # of records in each partition is too small to make predictions

Gain ratio

- A modification that reduces its bias on high-branch attributes
- Gain ratio should be
  - Large when the number of branches is small
  - Small when the number of branches is large

Gain ratio

- Takes the number and size of branches into account when choosing an attribute
- Corrects the gain by using splitinfo:
  - taking the *information* of a split into account
  - i.e., how much info do we need to tell which branch an example belongs to

Gain ratio = Gain(A) /splitinfo

Algorithm for Decision Tree Induction
[ID3 – Quinlan 1986 algorithm]

- Basic algorithm (a greedy algorithm)
  - The tree is constructed in a top-down recursive divide-and-conquer manner
  - At the start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Samples are partitioned recursively based on selected attributes
  - Test (split) attributes are selected based on a heuristic or *impurity measure* (e.g., information gain)

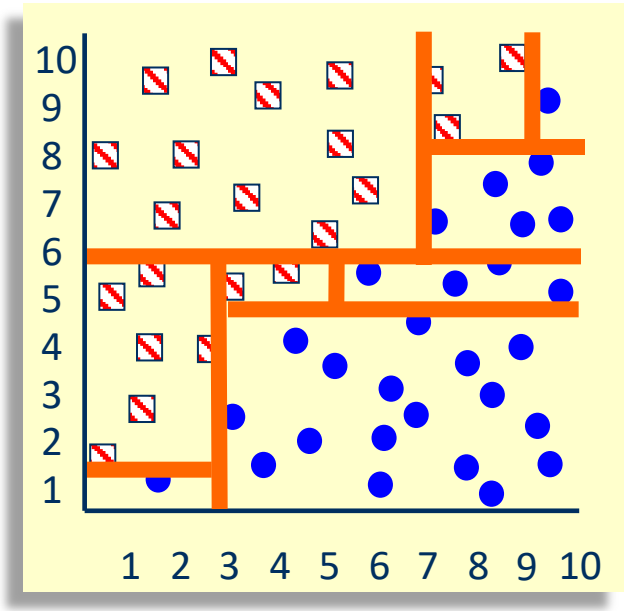
Algorithm for Decision Tree Induction
[ID3 – Quinlan 1986 algorithm]

- Conditions for stopping the partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning; majority voting is employed for classifying the leaf
  - There are no samples left

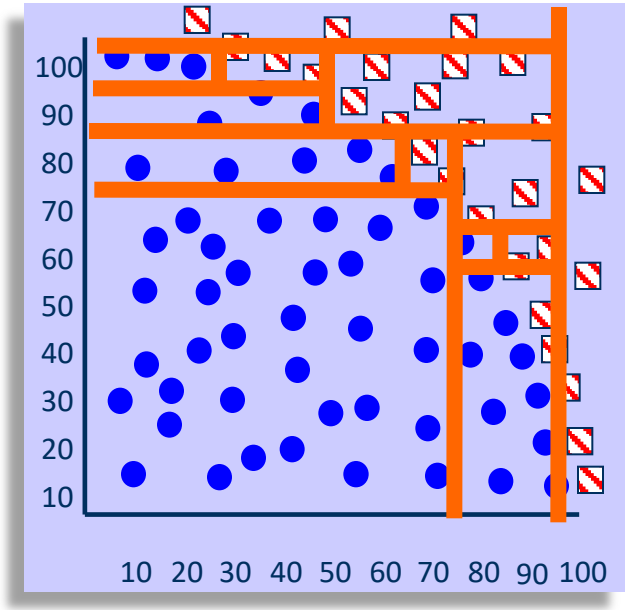
Revisiting our motivating examples

Which of the thre problems can be solved by a Decision Tree?

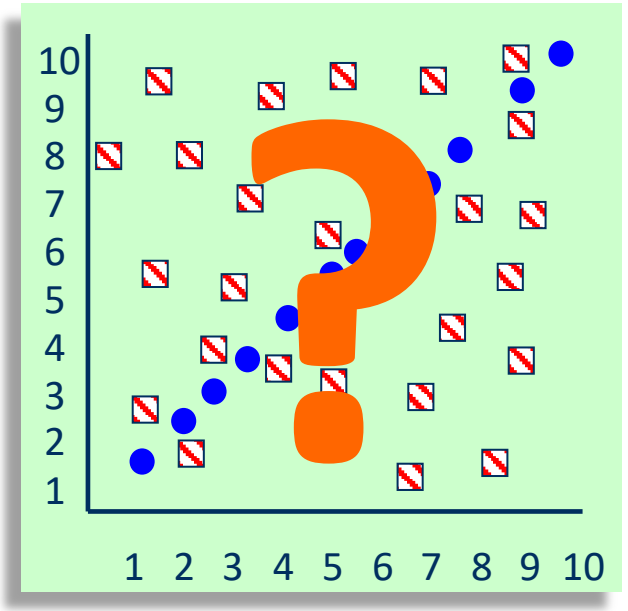
Problem 1



Problem 3



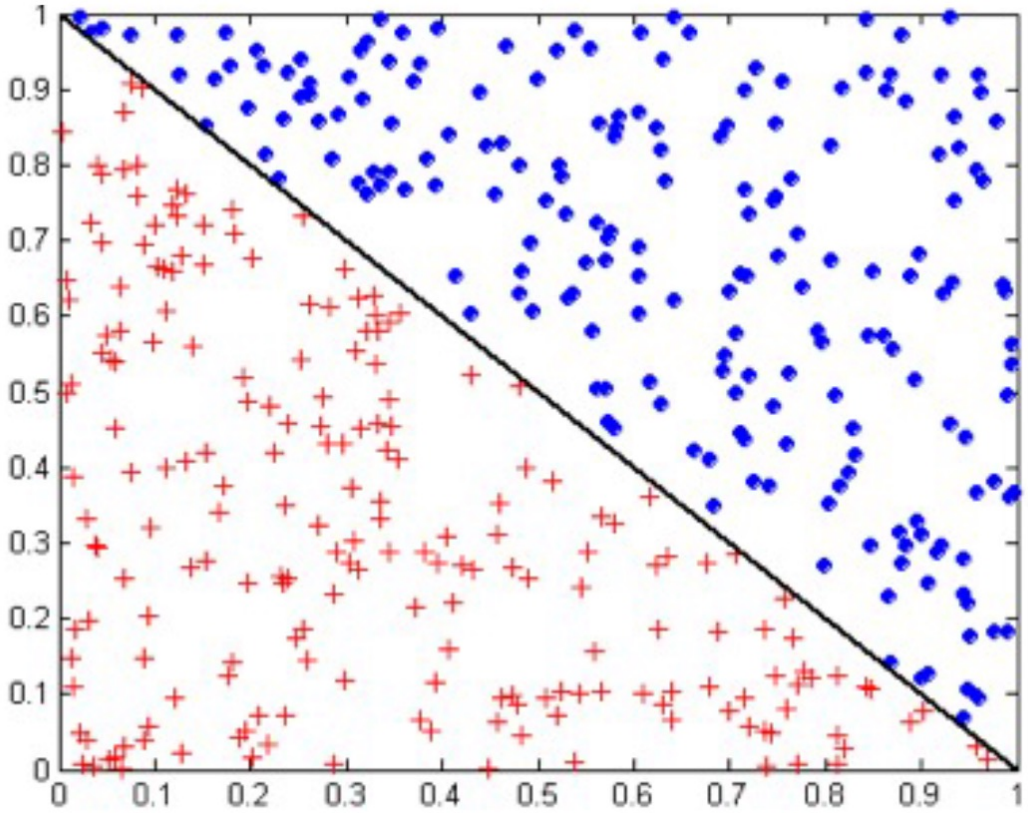
Problem 2



The Decision Tree has a hard time with correlated attributes

Stockholms universitet

Oblique Decision Trees

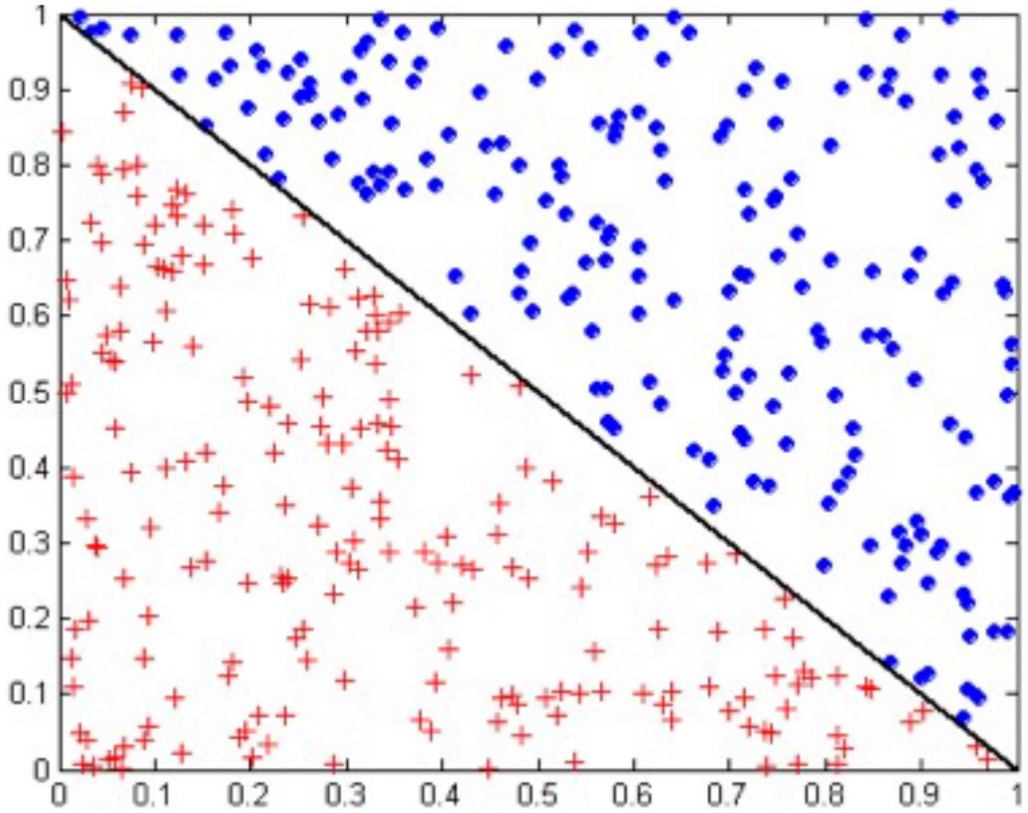


?

Not all datasets can be partitioned optimally using test conditions using single attributes!

Stockholms universitet

Oblique Decision Trees

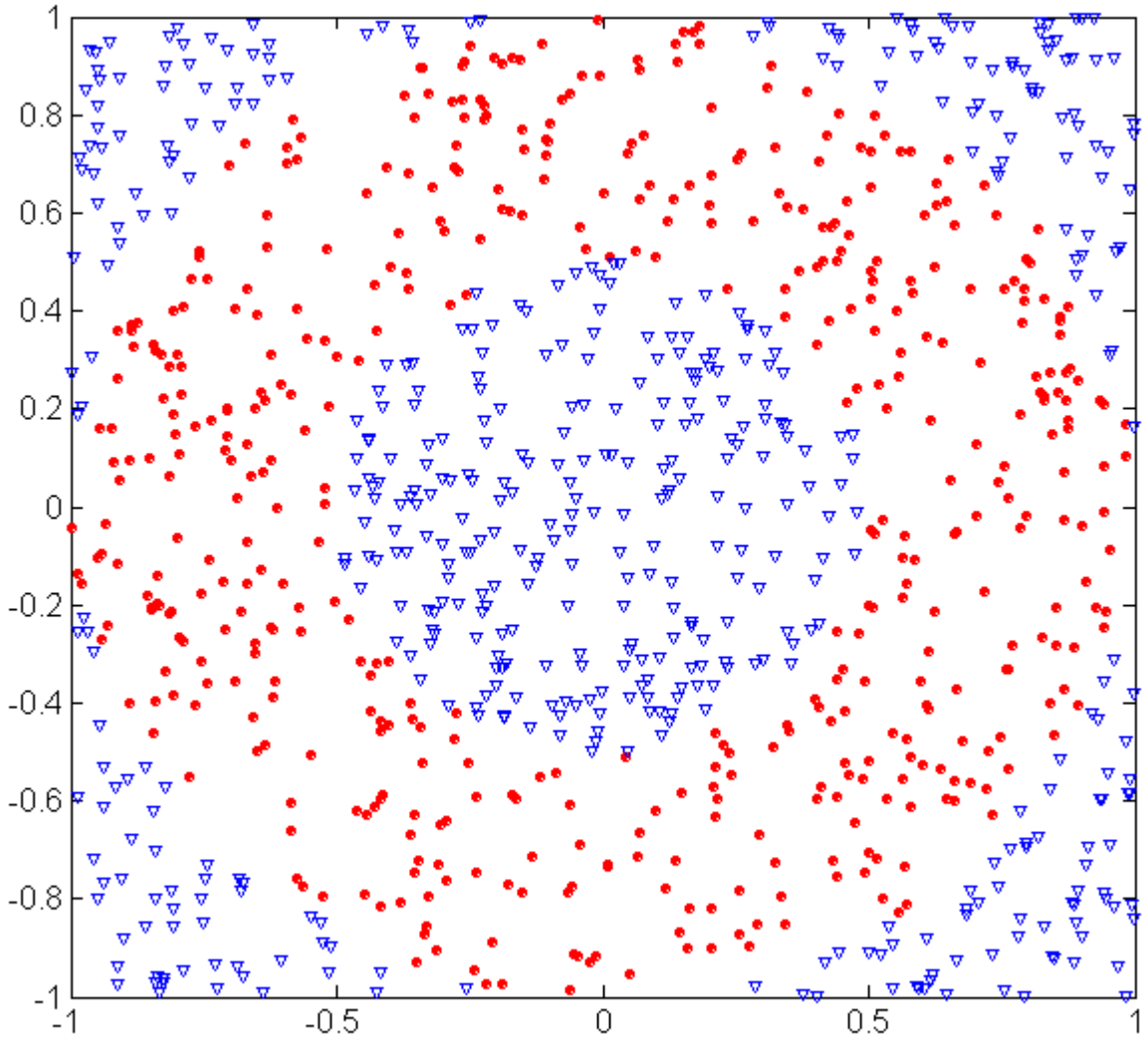


Test on multiple attributes

If **x+y < 1**
then
red class

Not all datasets can be partitioned optimally using test conditions using single attributes!

Oblique Decision Trees



500 circular and 500 triangular data points.

Circular points:

$0.5 \leq \text{sqrt}(x_1^2+x_2^2) \leq 1$

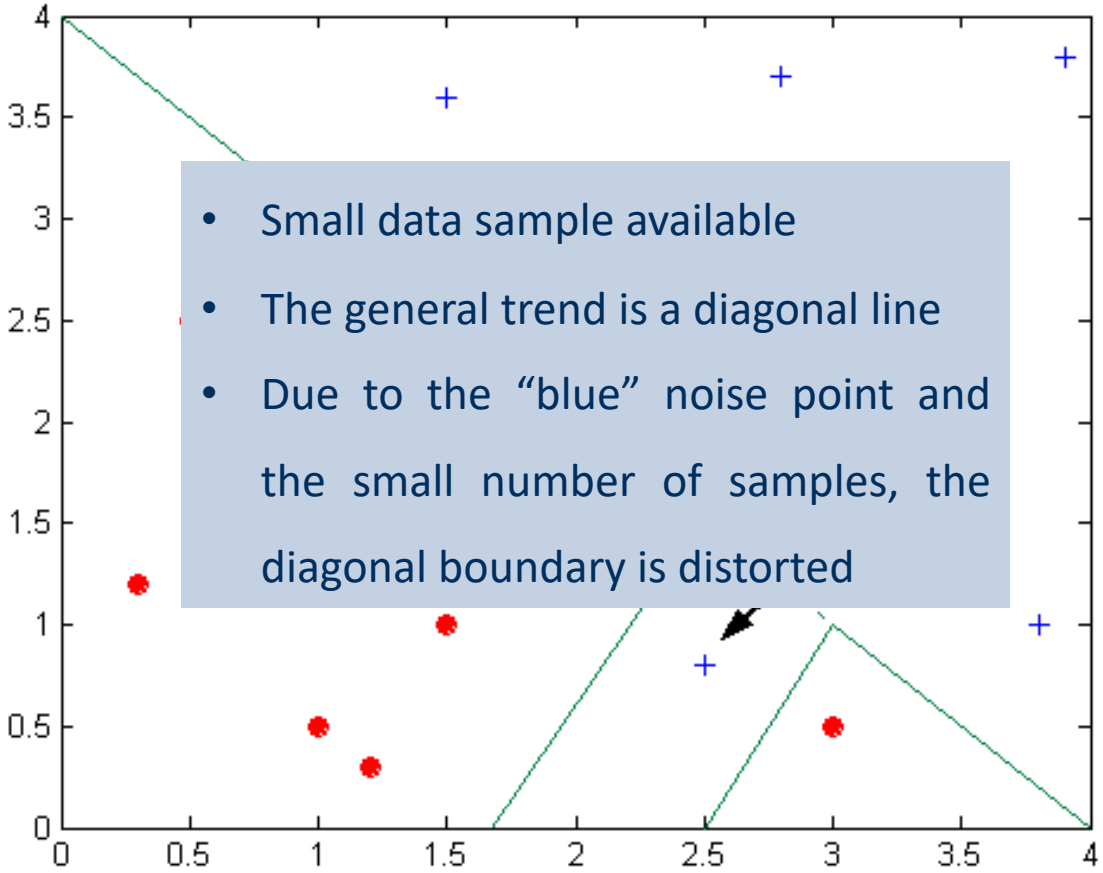
Triangular points:

$\text{sqrt}(x_1^2+x_2^2) > 1$ or

$\text{sqrt}(x_1^2+x_2^2) < 0.5$

Stockholms universitet

Overfitting due to noise



- Small data sample available
- The general trend is a diagonal line
- Due to the "blue" noise point and the small number of samples, the diagonal boundary is distorted

Decision boundary is distorted by noise point

Overfitting

- An induced tree may overfit the training data
  - too many branches, some may reflect anomalies due to noise or outliers
  - poor accuracy for unseen samples
- Two approaches to avoid overfitting

Overfitting and Tree Pruning
[C4.5 – Quinlan 1993 algorithm]

- Prepruning: Halt tree construction early
  - do not split a node if this would result in the goodness measure falling below a threshold
  - **difficult to choose an appropriate threshold**
- Postpruning: Remove branches from a "fully grown" tree
  - get a sequence of progressively pruned trees
  - use validation data to decide which one is the best

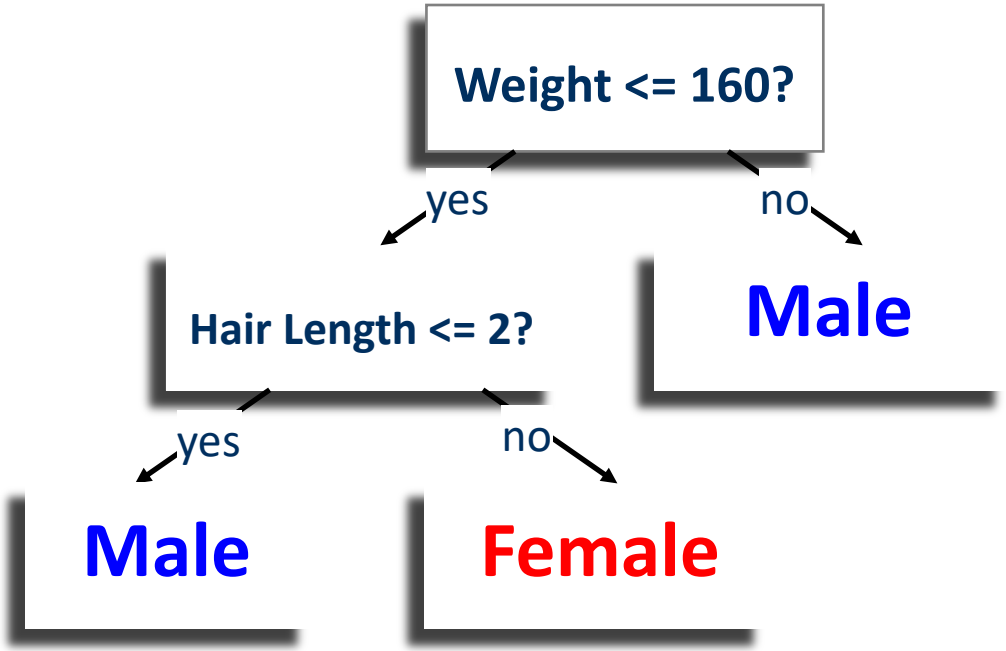
Pros and Cons of decision trees

- **Pros**
  - **+ Reasonable training time**
  - **+ Fast application**
  - **+ Easy to interpret**
  - **+ Easy to implement**
  - **+ Can handle many features**

- **Cons**
  - **– Cannot handle complicated relationships between features**
  - **– Simple decision boundaries**
  - **– Problems with lots of missing data**
  - **– NP-complete**

Extracting Rules

It is trivial to convert Decision Trees to rules…



Three rules to classify Males/Females

- **If** *Weight* **greater than** 160, classify as **Male**
- **If** *Hair Length* **less than or equal** to 2, classify as **Male**
- **If** *Weight* **less than** 160 **and** *Hair Length* **greater than** 2, classify as **Female**

Well-known decision tree learning implementations

- ID3: Quinlan JR (1986) Induction of decision trees. Machine Learning 1:81–106
- C4.5: Quinlan JR (1993): Programs for machine learning. Morgan Kaufmann (J48: Implementation of C4.5 in WEKA)
- CART: Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and Regression Trees. Wadsworth

	Splitting Criteria	Attribute type	Missing values	Pruning Strategy	Outlier Detection
ID3	Information Gain	Handles only Categorical value	Do not handle missing values.	No pruning is done	Susceptible to outliers
C4.5	Gain Ratio	Handles both Categorical & Numeric value	Handle missing values.	Error Based pruning is used	Susceptible to outliers

Today…

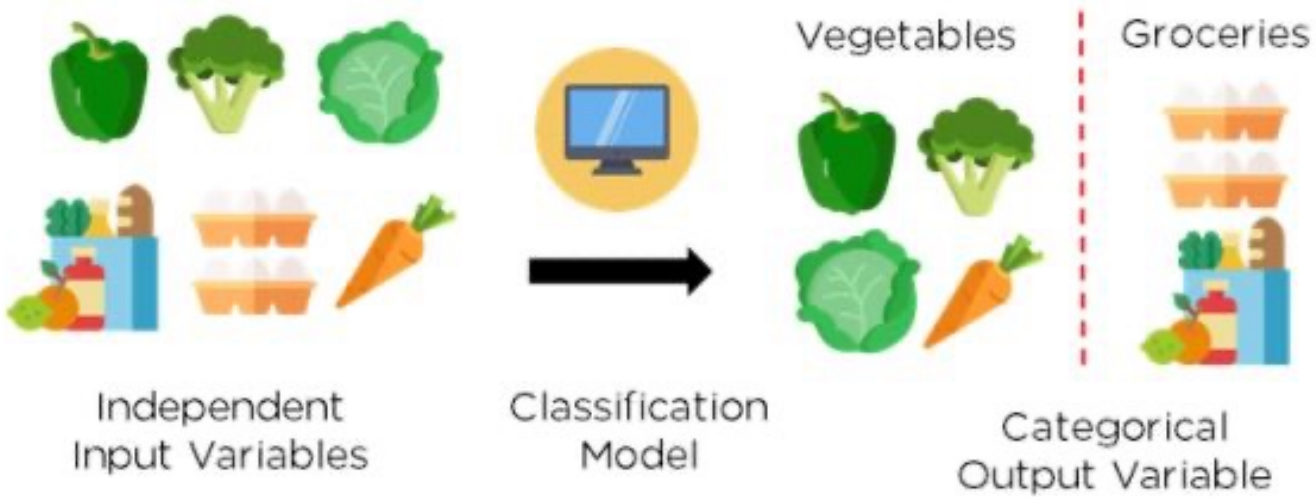
What is a **Decision Tree**?

Which are the **Impurity Measures**?

What is the **Information Gain?**

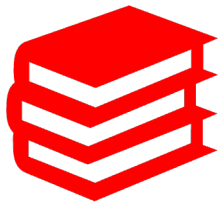
What is the difference with **Gini Index**?

How do we avoid **Overfitting**?



Stockholms universitet

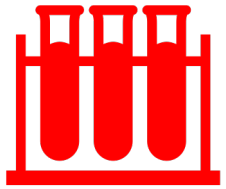
TODOs



Reading:

Main course book:

Chapter 3.2, Chapter 9

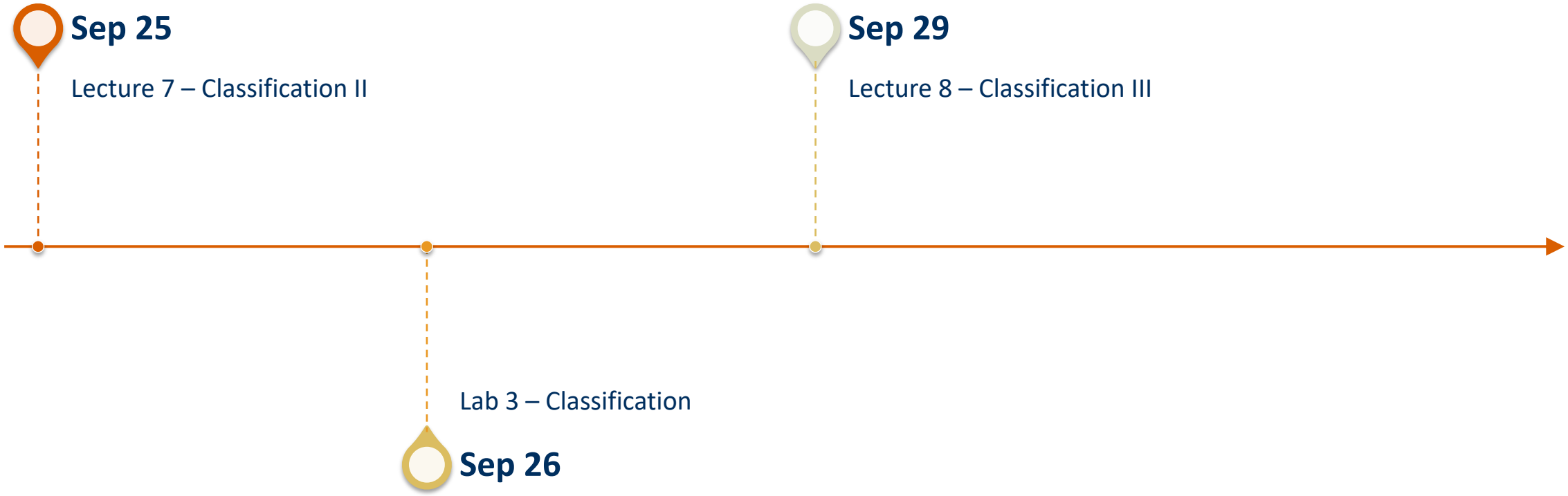


Lab 2

Recommended to complete the lab before the end of the week

Quiz 2

Coming up next



Stockholms universitet