Classification with machine learning: part A

Dr Maria Anastasiadi
(m.anastasiadi@cranfield.ac.uk)

14th January 2025

www.cranfield.ac.uk

Intended learning outcomes (ILOs)

- Be able to describe the underline concepts of the following ML algorithms:
  - K-Nearest Neighbours (k-NN);
  - Decision Trees ;
  - Classification Rules;
  - Give examples of applications.

Lazy Learning:
k-Nearest Neighbours (kNN)

- **K-NN algorithm** classifies unknown samples by assigning them the class of similar labelled samples.
- K-NN perform well when the items of similar class type tend to be fairly homogeneous.
- K-NN DON’T perform well when the data is noisy.
- Suitable for both classification and regression tasks.

k-NN algorithm

How it works:

a) An existing set of example data with known labels is assembled (training set).

b) The similarity between each sample is calculated based on distance functions.

c) When given a new piece of data without a label, the computer compares it to every piece of existing data.

d) Then it takes the **k** most similar pieces of data (the nearest neighbours) and looks at their labels.

e) Finally, a **majority vote** is obtained from the k most similar pieces of data, and the majority is the new class.

k-NN algorithm

Similarity and Distance

- The k-NN algorithm uses distance functions to calculate similarity between samples, e.g. Euclidean and Manhattan distance.
- **Similarity** is quantified as the distance between two samples in the N-dimensional space spanned by the N measured variables.
- The shorter the distance between two objects, the more similar they are considered to be.

Why is the k-NN algorithm lazy?

- No model is produced. The abstraction and generalisation processes are skipped altogether.
- The lazy learner is not really learning anything and the training phase is not really training anything.
- Nevertheless k-NN classifiers can be quite powerful and they allow the learner to find natural patterns rather than trying to fit the data into a preconceived and potentially biased functional form.

Lazy Learning: k-Nearest Neighbours (kNN)

Pros:

- Simple and effective (easy to understand);
- High accuracy;
- No assumptions about data required;
- Fast to train;
- Works easily on multi-class problems.

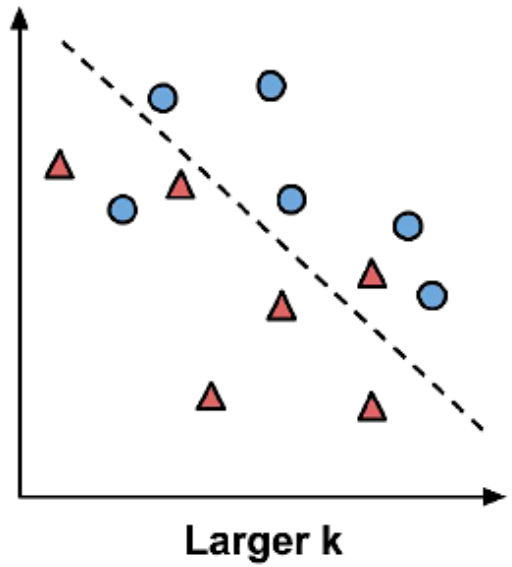
Cons:

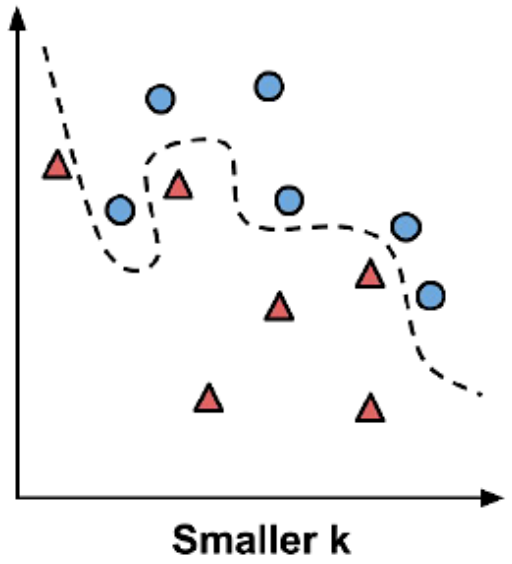
- Computationally expensive/ Memory intensive;
- Slow classification phase;
- Requires selection of an appropriate k;
- Does not work well on skewed target variables.

k-NN algorithm - considerations

When selecting k values the following apply:

- Large *k* reduces the impact or variance caused by noisy data, but can bias the learner so that it runs the risk of ignoring small, but important patterns.
- Smaller *k* values allow more complex decision boundaries that more carefully fit the training data.





Caution!
Too small *k* values allow the noisy data or outliers to influence the classification.

Preparing data for use with k-NN

Features are typically transformed to a standard range prior to applying the k-NN algorithm as the distance formula is highly dependent on how features are measured.

Range scaling 0-1

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Auto-scaling (z-score standardization)

$$X_{new} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{StdDev}(X)}$$

The same scaling method used on the k-NN training dataset must also be applied to the examples the algorithm will later classify.

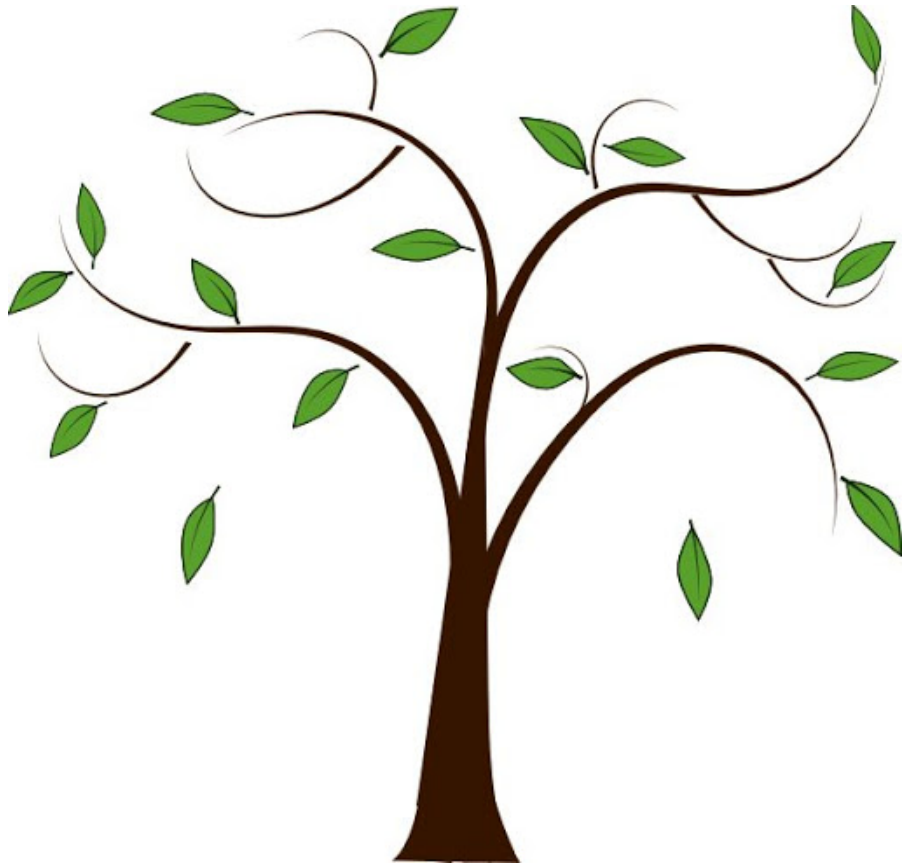
kNN: applications

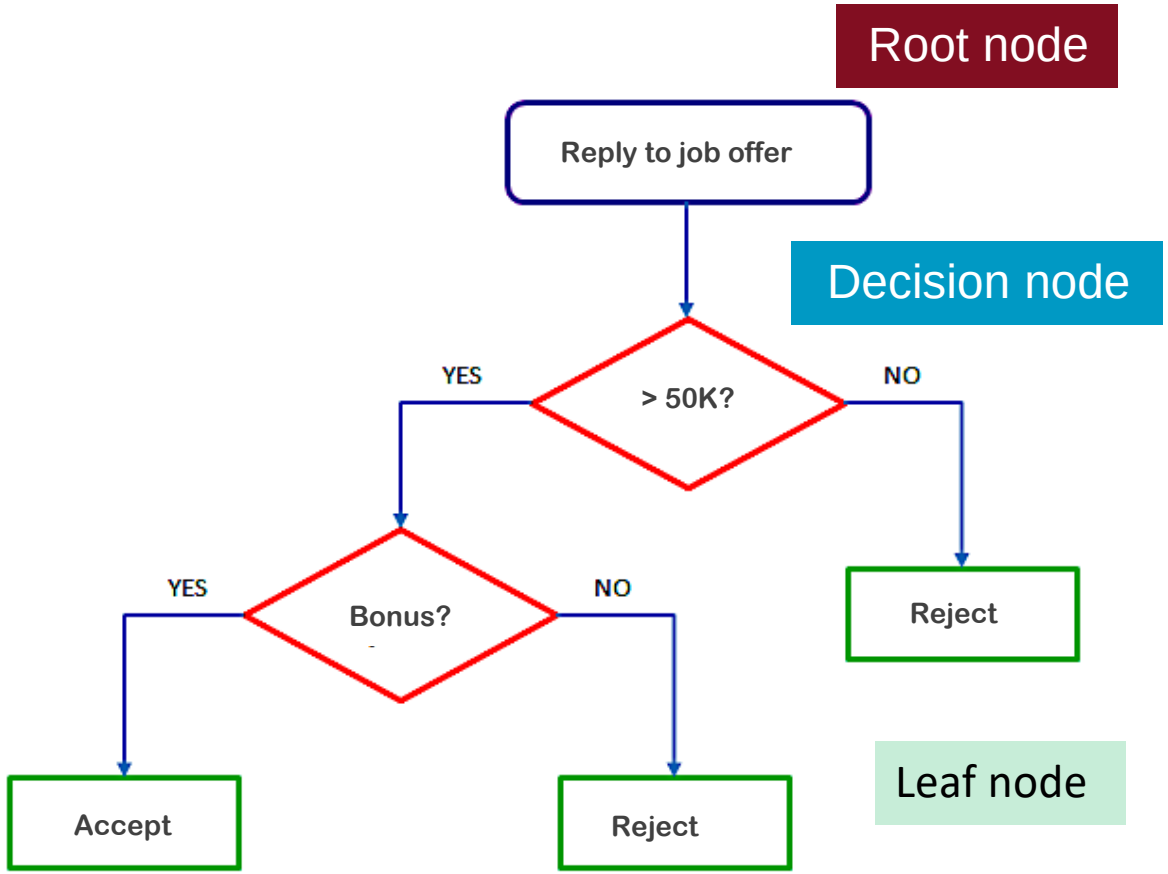
Examples of k-NN applications include:

- Pattern identification in genetic data e.g. for disease detection.
- Optical character recognition e.g. handwriting.
- Image analysis e.g. facial recognition, medical imaging.

Decision Trees

Decision tree learners are powerful classifiers, which utilise a **tree structure** to model the relationships among the features and the potential outcomes.





Decision Trees

Examples of decision trees applications:

Banking

- Credit scoring models in which the criteria that causes an applicant to be rejected need to be clearly documented and free from bias.

Healthcare Management

- Prediction of problems in cognitive development, language and motor development in preschool children based on their medical records (Chang 2007).
- Survival rate prediction of breast cancer patients with 93.6% accuracy (Delen et al., 2005).

Engineering

- Energy consumption prediction.
- Fault diagnosis.

Decision Trees

How it works

1. **Collect the data**: instrumental analysis, online repositories, in house databases.
2. **Explore and prepare the data**: create random training and test data sets.
3. **Train a model on the data**: use a decision tree algorithm to construct a tree data structure.
4. **Test**: calculate the error rate with the learned tree.
5. **Improve model performance**: adaptive boosting (AdaBoost), a process in which many decision trees are built and the trees vote on the best class for each sample.

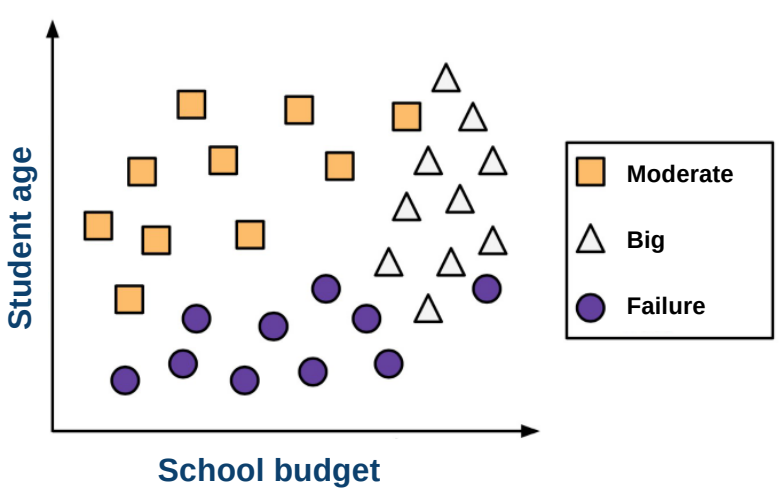
How are decision trees constructed?

Divide and conquer

Decision trees are built using a heuristic called **recursive partitioning.**

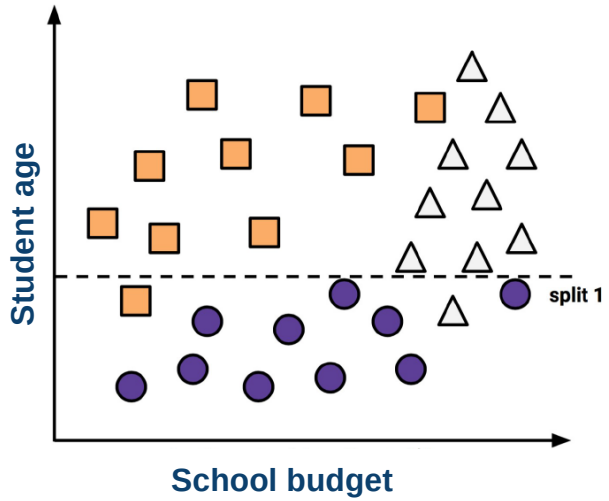
Decision Trees

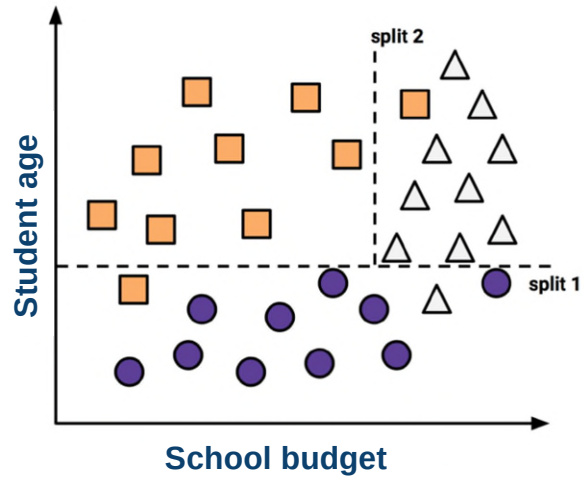
Divide and conquer in practice



Success rate of government programme to control norovirus outbreaks in schools

axis-parallel splits





- 80% of the examples in each group belong to the correct class.
- Further splitting may result in overfitting.

Decision trees in R

Available with **rpart, tree, maptree, C50, caret** packages in R

Strengths

- It can handle both numeric and nominal data as well as missing values.
- Excludes unimportant features.
- Relatively easy to interpret results.
- Works well with both small and large datasets.

Weaknesses

- Decision tree models are often biased towards splits on features with large numbers of levels.
- It is easy to overfit or underfit the model.
- Small changes in the training dataset can have great impact to decision logic.
- Large trees can be difficult to interpret.

Decision trees algorithm

How it works

The decision trees algorithm splits a dataset in a way that makes unorganised data more organised using concepts from *information theory*.

To quantify the degree of disorder or randomness within a dataset we use the concept of **Entropy (H)**.

$$Entropy(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

Where:

S = a segment of the dataset

c = the number of classes

p_i = the proportion of values falling into class level i.

The decision tree tries to find splits that reduce entropy, ultimately increasing homogeneity within the groups.

Decision trees algorithm

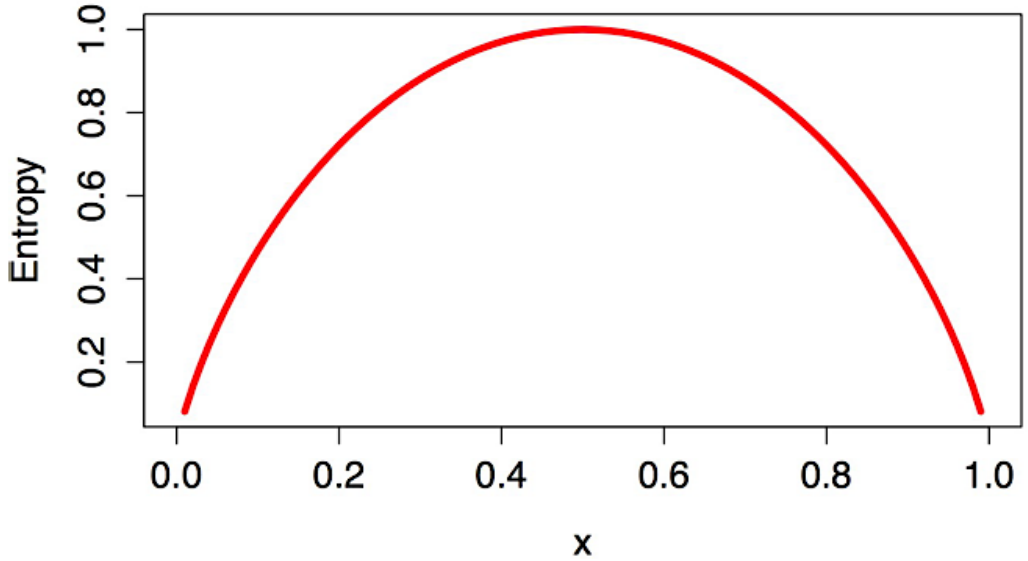
Example:

Entropy calculation for a binary class problem.

For 2 classes, entropy ranges between 0 to 1.

H=1, when the samples are equally split between the two classes.

Entropy curve for a binary class



x = the proportion of samples in one class

In general:

- Min entropy indicates that the samples are completely homogenous.
- Max entropy indicates that the data are as diverse as possible.
- For *n* classes, entropy ranges from 0 to $log_2(n)$.

Decision trees algorithm

Now that we know how to calculate the level of disorder in our dataset we need to start splitting it.

But how do we select which feature to split upon?

By calculating the change in homogeneity that would result from a split on each possible feature. This measure is known as **information gain.**

$$InfoGain(F) = Entropy(S1) - Entropy(S2)$$

Where: **F** = the feature to split upon, **Entropy(S1**) = the entropy in the segment before the split, **Entropy(S2)** = the entropy after the split.

The higher the information gain, the better a feature is at creating homogeneous groups after a split on this feature.

Splitting criterion- Information Gain

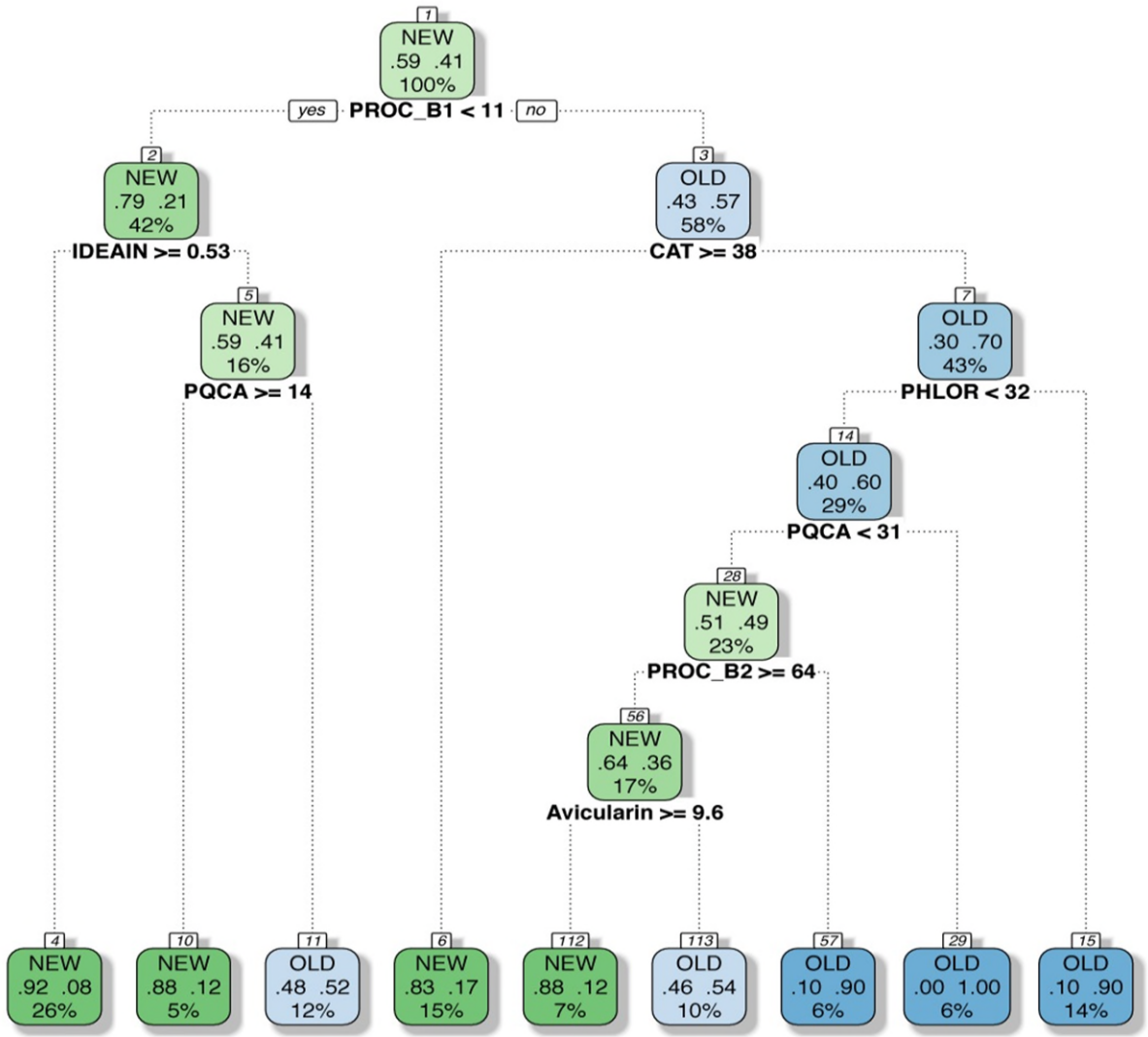
The total entropy after the split, is the sum of the entropy of each of the n partitions (P_i) weighed by the proportion of samples (w_i), in each partition i.

$$\boldsymbol{Entropy(S2) = \sum_{i=1}^{n} w_i Entropy(P_i)}$$

The aim is to reduce the total entropy as much as possible without overfitting the model.

Splitting numeric features

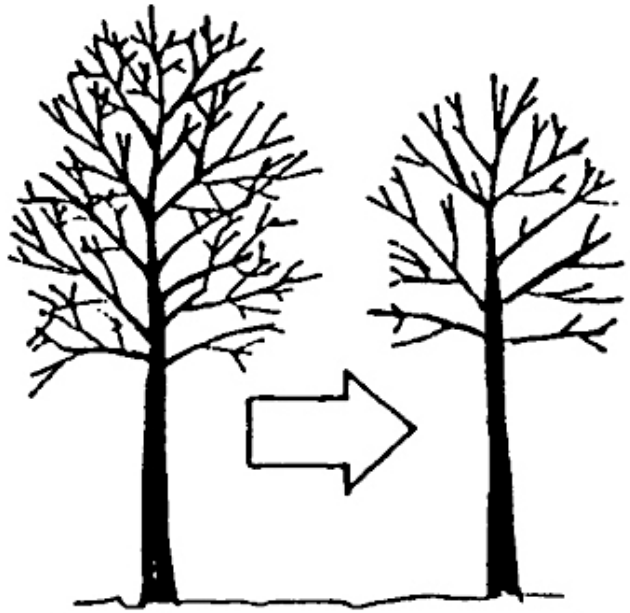
- Decision trees use information gain for splitting on numeric features as well.
- A common practice is to reduce the numeric feature into a **two-level categorical feature** that allows information gain to be calculated as usual.
- The numeric cut point yielding the **largest information gain** is chosen for the split.



Decision Tree for Phenolic Model for Age Prediction – (minsplit = 50)

Anastasiadi et al., 2017.

Pruning the decision tree



- An overly large tree is prone to overfitting.
- **Pruning** a decision tree involves reducing its size such that it generalises better to unseen data.
- **Approach 1: Pre-pruning** involves stopping the tree from growing once it reaches a certain number of decisions. Potentially risky as it may miss important patterns.
- **Approach 2: Post-pruning** involves growing a tree that is intentionally too large and pruning leaf nodes to reduce the size of the tree to a more appropriate level. Often more effective strategy.

Decision Trees

Pros:

- Computationally cheap to use;
- Easy for humans to understand learned results;
- Missing values are usually not a problem
- Decision trees can be used both with Numeric values and Nominal values.

Cons:

- Prone to overfitting.

Classification Rules

- **Rule-based** models can be used as an alternative to constructing classification tree models (decision trees).
- **Rule learners are generally applied to problems where the features are primarily or entirely categorical.**
- Classification rules use logical if-else statements to assign a class to unknown samples. For example: "if the hard drive is making a clicking sound, then it is about to fail.“
- The results of a rule learner can be more simple, direct, and easier to understand than a decision tree built on the same data.

Classification Rules

Main Principle

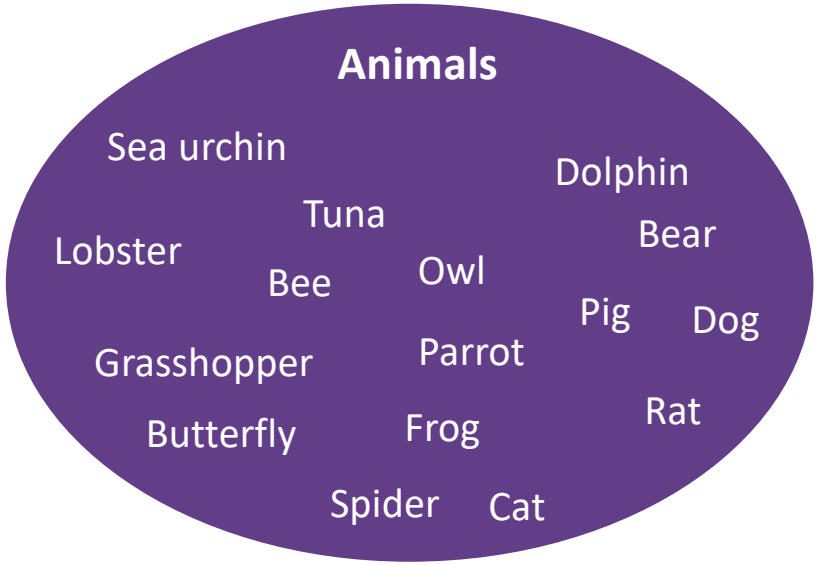
Separate and conquer

How it works

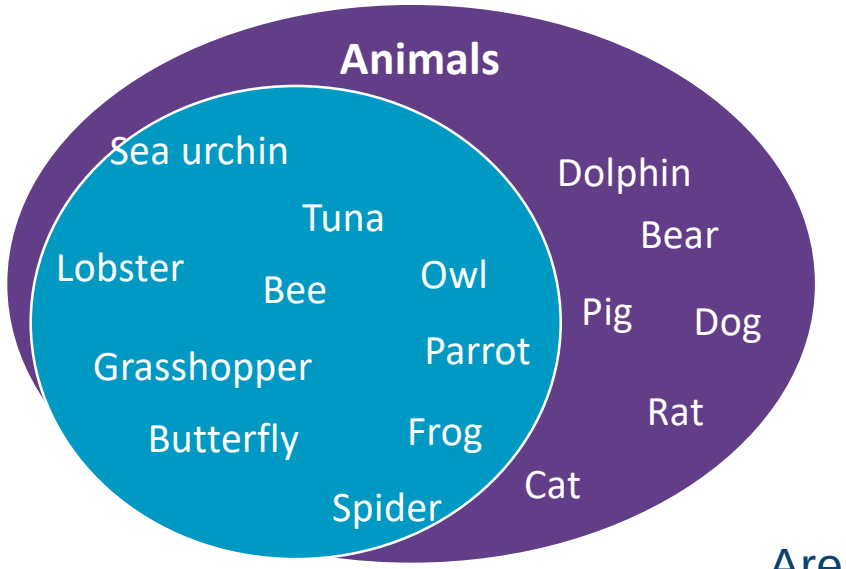
The process involves identifying a rule that covers a subset of samples in the training data, and separating this partition from the remaining data. As the rules are added, additional subsets of the data are separated until the entire dataset has been covered and no more samples remain.

Classification Rules - Example

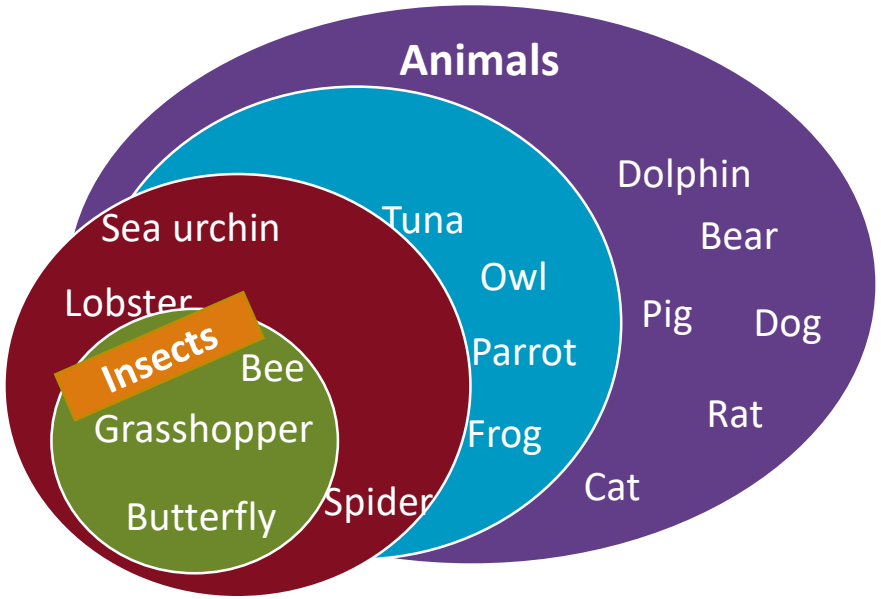
Which of these animals are insects?



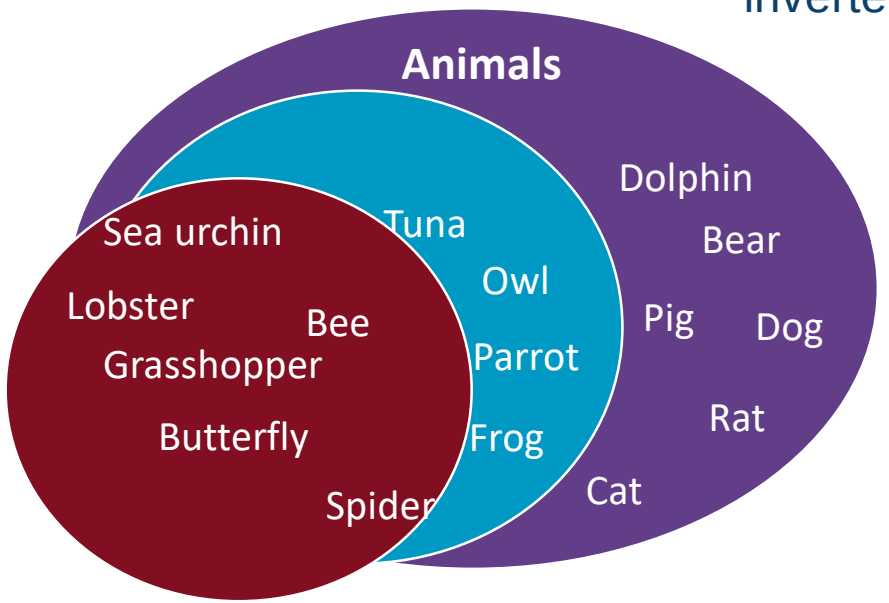
Do they hatch from an egg?



Are they invertebrates?



Are they hexapods?



Summary

- Lazy Classifiers (k-NN);
- Decision Trees:
  - Divide and conquer;
  - Pruning the decision tree;
  - Cost matrix.
- Classification Rules;
  - Separate and conquer

www.cranfield.ac.uk

T: +44 (0)1234 750111

@cranfielduni

@cranfielduni

/cranfielduni