Introduction

What is data mining

- Data mining aims to discover **patterns** and **rules** from large amount of data.
- Helps with the **prediction** of future *events* and **classification** of past ones.
- Also known as **KDD**: *Knowledge Discovery in Database*.

Data mining Process

```
   Raw Data ← ──────────────────────────────────────── Knowledge
       |                                                    ↑
       ↓                                                    |
Data Warehouse → Target Data → Transformed Data → Patterns and Rules
```

Origin of data

- The data usually comes from measurements of the real world
- Using data from the past, you can predict the future

Problems with data

- **Errors —** data might not be not accurate, or missing/
- **Quantity —** data might no be enough to find accurate patterns.
- **Quality —** data might not contain the information you need.
- **Cost —** data might be too expensive to obtain.

Data mining techniques

- K-nearest neighbour
- K-means clustering
- Decision trees
- Neural networks
- Market basket analysis
- Logistic regression
- ARIMA — *Autoregressive Integrated Moving Average*

Glossary

- **Data —** *Data* are measurements of *values* associated with *variables*.
- **Data Model —** A representation of some aspects of the *data*, usually performing a *task*.
- **Inference —** Querying a *model* to obtain a prediction from some input *data*.

- **Learning —** *Learning* is the process of using *data* to build a *model* capable of performing a *task*.
- **Nominal —** Nominal *Variables* describe a quality, such as gender or colour. They are also known as *categorical labels*.
- **Numeric —** *Variables* can be numeric, accepting numbers as *values*.
- **Task —** What the end system is supposed to do.
- **Variable —** A property that can be measured and varies.
- **Value —** A *variable* can be assigned to a certain set of *values*.

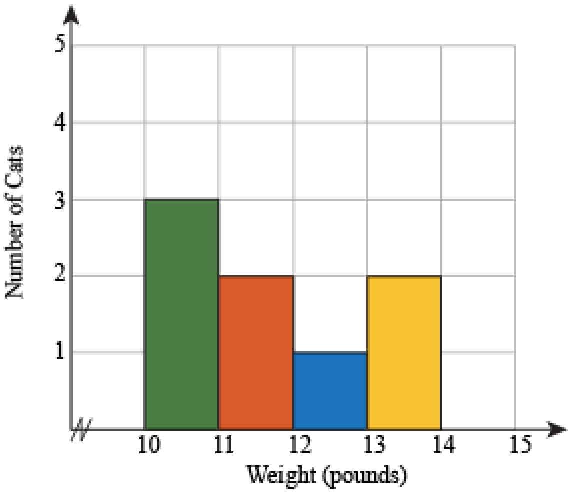
Data Preparation

To mine specific knowledge out of some raw data, you need appropriate data.

Distribution

A **frequency distribution** is a count of how often each variables contains each value in a data set. For discrete values, like nominal ones, it is simply a count of each category. For continuous numbers, you need to split in range, and see how often values in those range appears.

The easiest way to **plot** distribution is using an histogram.



What to look at in a distribution

Outliers

Outliers are a small number of values that are much bigger or much smaller than the average. They are usually errors, or a sign that the recorded data is not complete.

Minority values

Values that appear very rarely in the distribution.

Data Balancing

Data balance is altering the underlying dataset, like removing or duplicating data, with the intent of helping training process to produce more accurate results.

Data Quantity

How much data do you need to represent the dynamics of the system modelled?

Noise and Variability

When plotting two variables into a scatter diagram, they might produce a straight line. In that case there is a **linear correlation** between the two variables.

They could also lie in a cloud along a straight line. In this case, the spread around the line is called the **correlation**, and this could be caused by:

- Imperfections in measurements
- Variability caused by other variables
- Randomness

Creating a line that minimise the errors between all the samples, is basically creating a model for the dataset. The mean distance of the model from each sample is the **error** of the model, usually referred as *mean square error* or **MSE**.

Creating model for a relationship that is not linear is much more difficult, and requires more data.

Learning is the process or minimising the **MSE**, either using linear regression, or iterative search.

Much more data is required to learn a non-linear relationship.

Summary

Data quality and quantity rely on:

- The **shape** of the data's distribution.
- The **number** of variables in the data.
- The **degree of linearity** in the relationship to be captured.
- The amount of **noise** and **unaccounted for variability** in the data.

Classification

Classification is assigning an object to a certain class, based on its **similarity** to previous examples.

The classification usually comes with a degree of certainty. It might either be the probability of that object belonging to that class, or how closely it resembles it.

The classification can either have a:

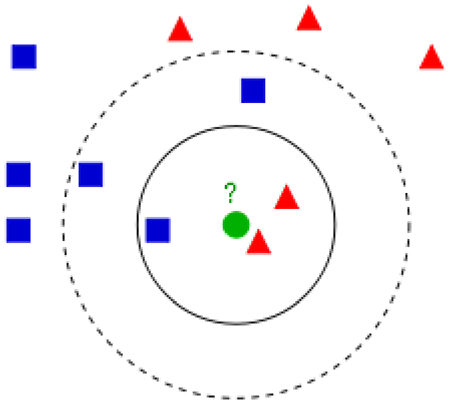
- **Non-paremetric model** (*k-nearest neighbours*)
- **Mathematical model** (*neural netoworks*)
- **Rule-based model** (*decision trees*)

A classification that indicate the propensity to an act, is called **predictive**. A classification that indicates the similarity to an object is called **definitive**.

Non-parametric models

K-Nearest Neighbour

Places all the entries in an n-dimensional space, counts all the example in a defined radius, and the ones that appear the most are picked as the winning classification



Rule-based models

Rule-based models attemps to classify an object based on rules on their attributes.

Decision trees

A decision tree automatically discover a rule from the data, and produces a set of branching decisions that end in a classification. It works best on nominal attributes.

The model tries to optimise the tree by arranging the decision in the most effective way. The ID3 algorithm splits on the variables that give the greatest **information gain**. The information gain of a certain event is defined as:

$$I(e) = -\log_2(P(e))$$

Entropy

The **weighted average information** accross al lthe possible values of a variable is called **Entropy**. It is calculated as the sum of the probability of each possible event times its information gain:

```
H(x) = Σ P(xᵢ) I(xᵢ) = -Σ P(xᵢ) log₂(P(xᵢ))
```

Conditional entropy is written as `H(outcome | known)`, and measures the uncertainty about the outcome, given what is known.

If `H(outcome)` and `H(outcome | input)` are known, it is possible to calculate how much `input` tells about `outcome` as:

```
H(outcome) - H(outcome | input)
```

And is defined as **information gain**.

Data Visualisation

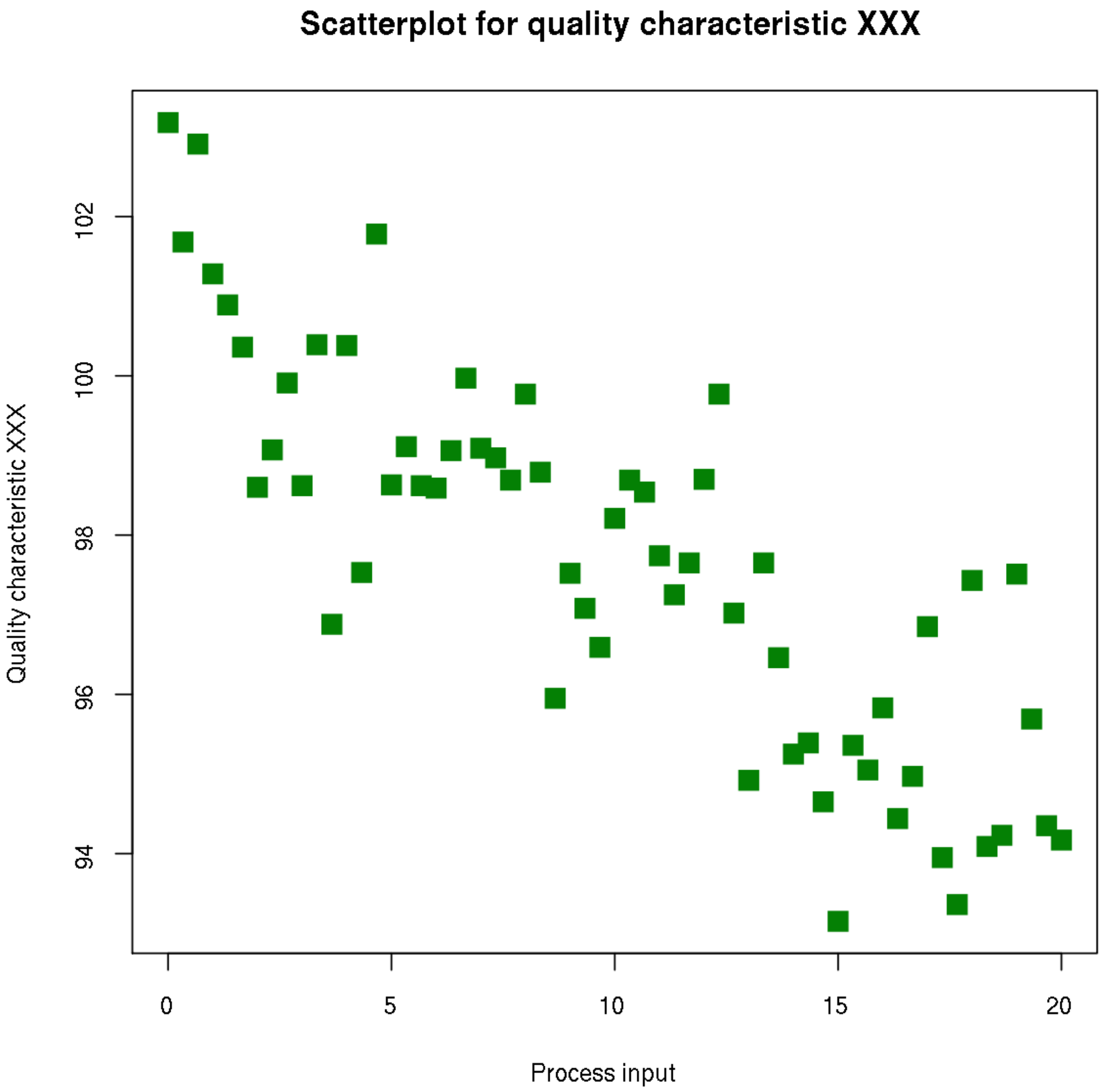
Human eyes are particularly apt at spotting patterns trends and clusters. This only works effectively up to three dimensions.

Data Visualisation is used:

- Before a data mining project to understand the problem.
- To guide the data mining project choosing a technique.
- To fine-tune the data mining techniques
- To show the results of the data mining.

Scatter Plots

Scatter plots are two-dimensional charts that places two variables as dots in an x-y diagrams.



Can be quickly used to understand the **correlation** between two variables and spot **clusters**.

To reduce probles with **overlapping** of variables, the values can be added with a random delta. This process is known as **jitter**. Also size of the dots, or different colours can be used instead of jittering.

Projection happens when trying to plot a graph with more than two dimensions. There always is some loss of information, but there are ways of reducing it.

Prediction

Prediction is the act of guessng the identity of one thing, purely based on the description of another. Not necessarily predicting future events.

Prediction differs from **classification** because it is usually able of predicting continuous numerical values, whereas classifications are able to *predict* discrete nominal values.

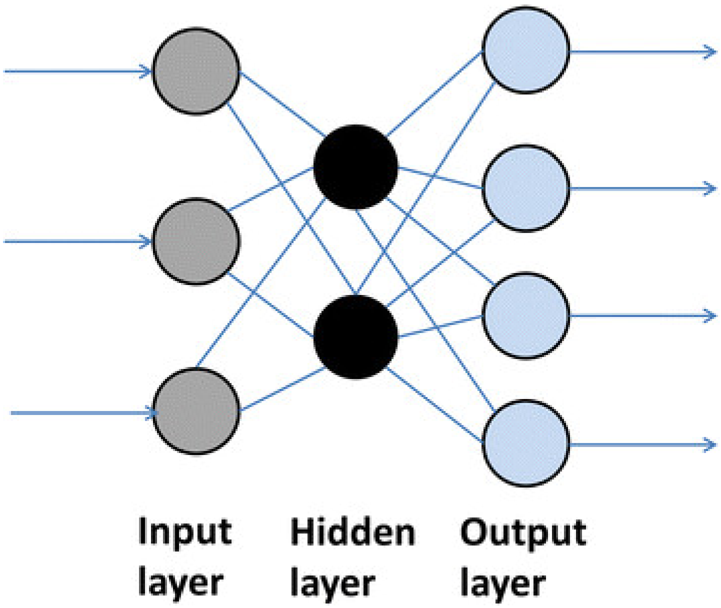
Most prediction techniques are based on mathematical models:

- Linear regression
- Power series
- Neural networks
- Radial Basis Function

Mathematical models learn the relationship between all predictors at once and the predicted outcome.

Neural Networks

A certain type of neural network called **multi-layer perceptron** (MLP) can learn a function between inputs, by building a network of smaller simple functions joined together by weighted connections.



A MLP has:

- Input layer
- Hidden layer(s)
- Output layer

During the **training** process the neural network uses the data to modify the weights of the connections between the nodes, until it is able to accurately predict the data.

Neural networks use **backpropagation** to balance the weights to reduce the output error.

Each neuron is composed of two units:

1. The weighted sum of the inputs.
2. The output of an activation functions.

Advantages

- Very powerful predictions.

- Works well with non-linear relationship.
- Works well with both numerical and nominal values.
- Able to generalise data that has not been provided before.

Disadvantages

- Difficult to explain the reasoning behind a prediction
- No rules to look at
- Can be very ineffective if trained wrongly

Overfitting

A data mining predictor can capture structures in the data to the point that irrelevant relationships in the training model are used in the creation of the model, while not generally true for a bigger data set. This phenomenon is known as **overfitting**. This usually happens when a dataset is not big enough.

Data Mining Project

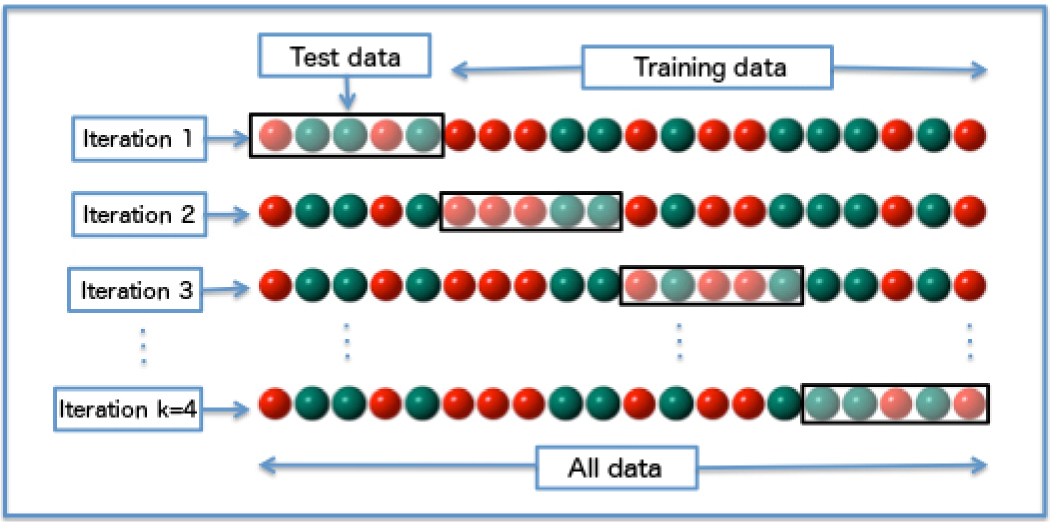
CRISP DM standard stands for CRoss Industry Standard Process for Data Mining.

Data Quantity

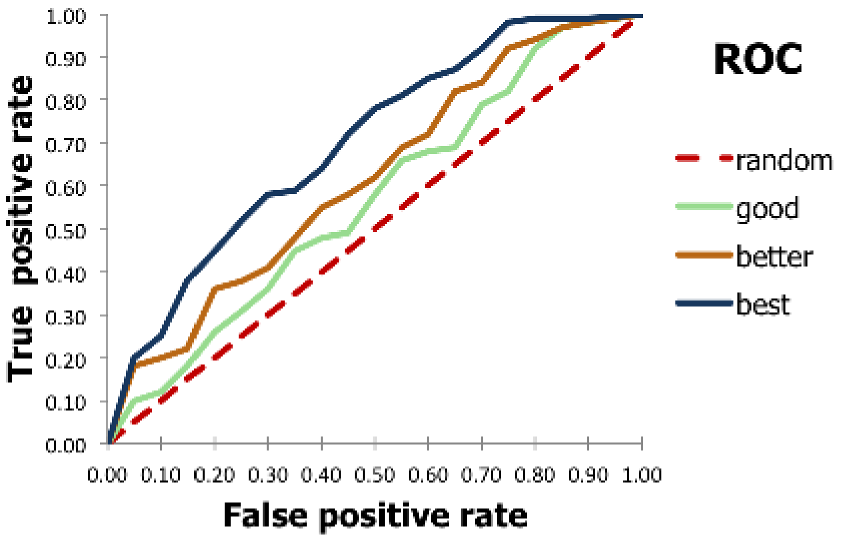
- Chose the variables to be used for the model
- Look at the distribution of the chosen values
- Look at the level of noise in the data
- Decide whether or not there are sufficient examples in the data
- Treat unbalanced data

Data Quality

Cross-validation splits the data into n subsets, then trains the model using n-1 subsets for training, and 1 for testing. This is repeated for each subset. This helps reducing overfitting.



ROC Curves are a tool for displaying the sorting efficiency of the model. The Y axis of the curve represents the sensitivity and the X axis the specificity. It shows how an increase of sensitivity affects the specificity of the model. The closer the ROC curve is to the top left border the more accurate the model is. The worst possible curve is one that lies along the 45° diagonal.



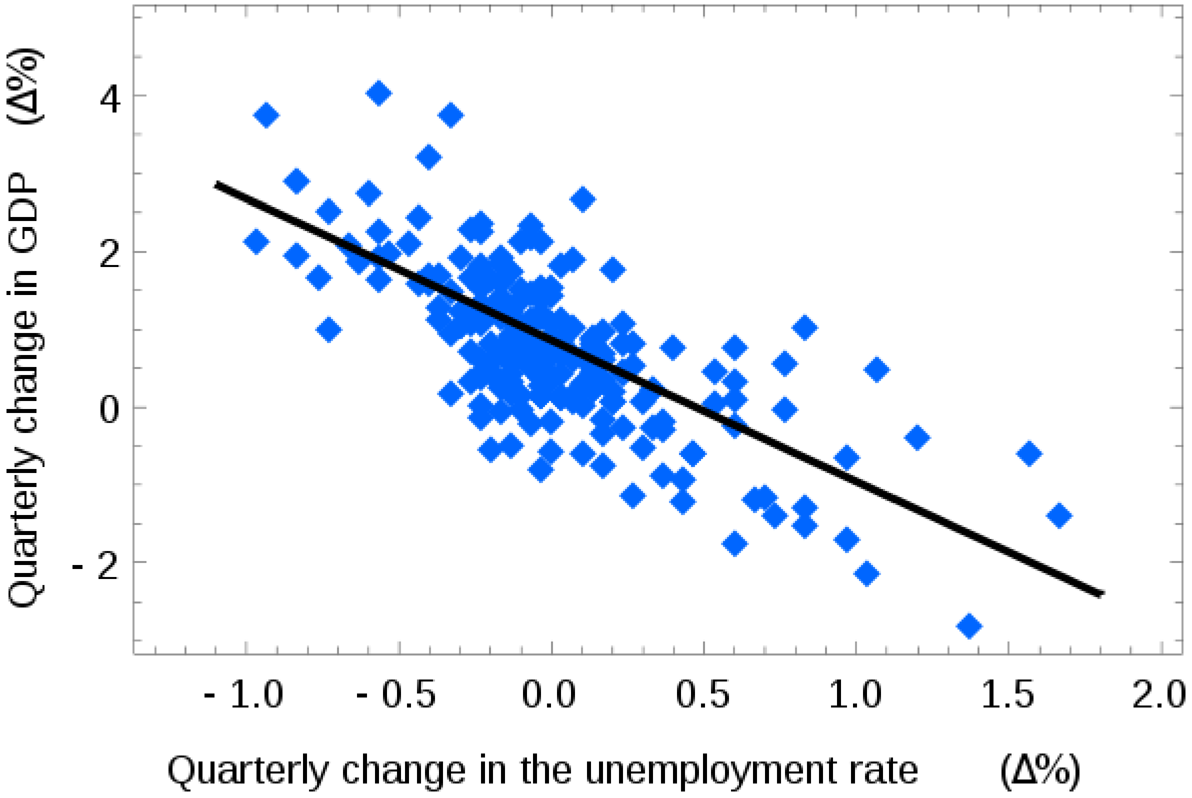
Lift is a measure of effectiveness of a predictive model, calculated as the ratio between the results obtained with and without the predictive model. **Lift Charts** show in the y-axis the lift, and on the x-axis the percentage percentage of the dataset.

Regression

Regression analysis is a statistical process for estimating the relationships among variables.

Simple Linear Regression

Simple Linear Regression uses the **Ordinary Least Squares** method. OLS is capable of estimating unknown parameters in a linear regression, by minimising the sum of the squares of the differences between the dependent variable and those predicted by the linear function.



Logistic Regression

Logistic regression allows to use one or more nominal values in a regression model.

It does that with the use of **log odds**.

The odds of an event with probability P(c) are:

```
O(c) = P(c) / (1 - P(c))
```

The log odds are are a function known as **logit**:

```
L(c) = ln(O(c)) = ln(P(c) / (1 - P(c)))
```

After some magic, the probability of an event `c`, given `x` are:

$$P(c \mid x) = 1 / (1 + e^{-(ax + b)})$$

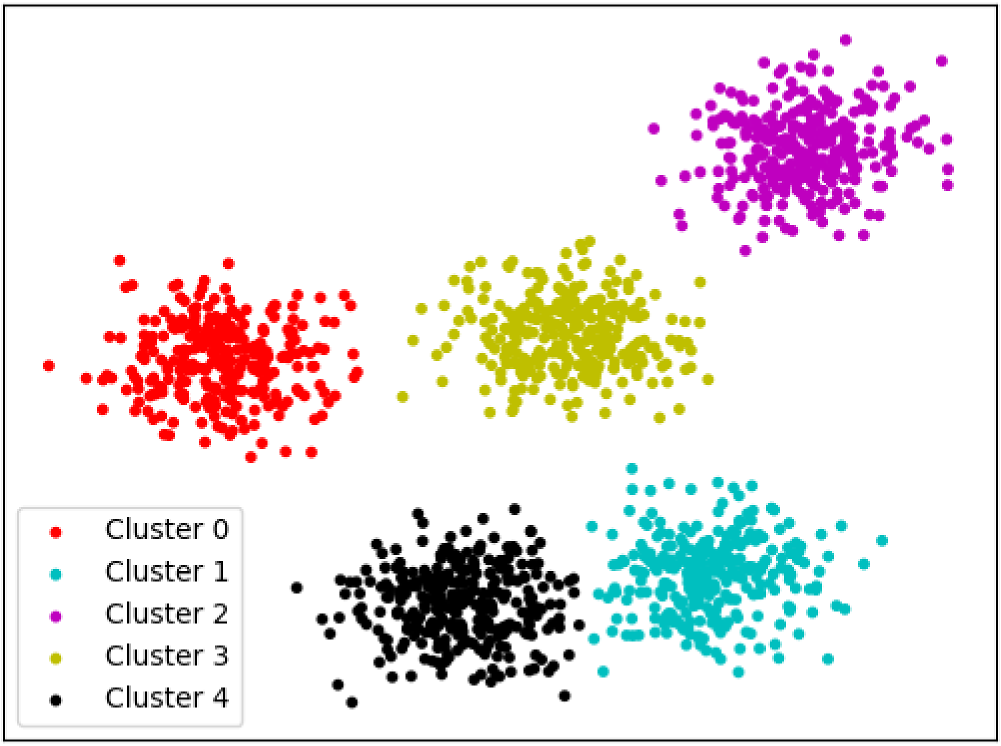
The **likelihood** is the reverse of a conditional probability:

```
L(x | y) = P(y | x)
```

The whole thing was out of scope, why did I even bother?

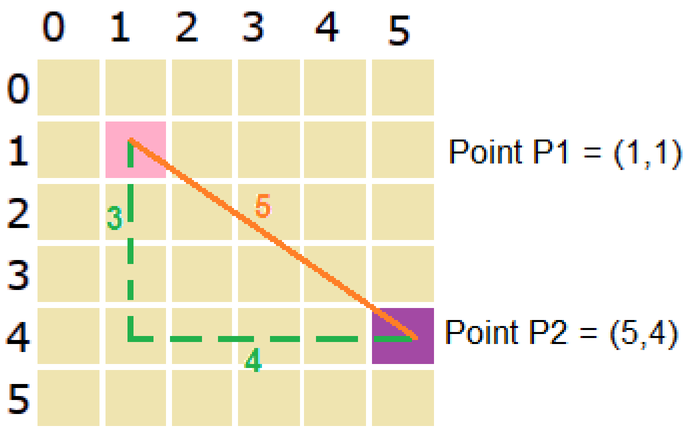
Clustering & Association Rules

Clustering or **Cluster Analysis** is the task of grouping set of similar objects in a cluster, based on some similarity algorithm.



There are various ways of measuring similarity:

- Numerical Data
  - Euclidean Distance
  - Manhattan Distance
- Categorical Data
  - Hamming Distance
  - Jacard Coefficient
- Combined Data
  - Weighted normalized distance



Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2} = 5$

Manhattan distance = $|5-1| + |4-1| = 7$

Mean Clustering

$\bar{x}$ is the mean of a value x in a dataset with size s, and is obtained as:

$$\bar{x} = (\Sigma x) / s$$

Clustering algorithms are able to find the mean of subsets that are called clusters, each with its mean.

The goal of the K-Means clustering algorithm is to minimise the sum of square of distance from all data points to their mean.

Algorithm:

- Pick K different points from the data and assume they are the centers
- Repeat until stabilisation
  - Assign each point to the closest cluster center
  - Calculate the center of each cluster and move the centroids

The disadvantages of the k-means clustering are:

- Assumes the clusters to be round
- K needs to be given in advance
- It assumes that all distances are equally important

Hierarchical Clustering Algorithm

Algorithm:

- Start with a cluster for each data point
- Repeat until reached number of clusters (could be one)
    - Joins the two clusters that are most similar
    - Calculate new center

Market Basket Analysis

Market Basket Analysis is a data mining technique that aims to discover co-occurence relationships between activities performed by a specific group of individuals. It is used, for instance, in retail to understand the purchase behaviour of customers. Example: `IF CUSTOMER BOUGHT A THEN HE WILL BUY B`. The process of finding the relationship is called **association rule discovery**.

- The **itemset** is the collection of one or more items.
- The **support count** (σ) is the frequency of occurence of an itemset.
- The **support** is the fraction of the transactions that contain an itemset.
- The **frequent itemset** an itemset whose support is greater than a treshold.

Association rules are expressed as X → Y, where X and Y are itemsets. Transaction rules are not symmetric.

The **apriori** algorithm is capable of finding association rules from data. Steps:

- Generate itemsets with a minimum support
- Generate rules with a minimum confidence

Time Series Forecasting

Time Series

A **Time Series** is a sequence of values or events where the next event is determined by events that precede it.

A time series reflects the **process** being measured. This process has certain **components** that affect its behaviour.

The **level** of a time series is the average of the values of the series at each point in time. If the average remains the same, the series is said to be **stationary**.

A time series is said to have a **trend** if its value increases or decreases continuously over time (non-stationary).

A **season** is any period of time that repeats through the data.

Cycles are smooth undulations of a process (often physical) and with a sine-like behaviour.

Time Series Forecasting Techniques

Different techniques are designed to work with a specific component of a time series. More than one technique might be necessary to effectively predict the time series.

Simple techniques predict the next step based on:

- The previous step
- The average of the last few
- The weighted average of the last few

Many processes with a fixedl level tend to go back to that certain level.

ARMA models can predict how quickly the process moves back to its level after a **shock**.

ARMA stands for Auto-Regressive Moving Average.

$$X_t = c + e_t + \Sigma f_i \cdot X_{t-i} + t_i \cdot e_{t-i}$$

Let's just write formulae, pretending we know what they do.

ARIMA stands for Auto-Regressive Integrated Moving Average, and is an extension of the ARMA model that takes **trends** in account.

Seasonal factors might either be:

- Additive
- Multiplicative

Seasonality must be identified and modelled, and then removed from the time series to look for other components.

Auto-Correlation Auto-correlation is a method of finding the correlation between each value and the n-values before it.

Fourier Transform is a mathematical method for decomposing a signal into a set of sine waves.

Recurrent Neural Networks are good at finding cyclical components in a time series.

Unless the system is completely closed, there always will be outside forces at play. There forces can be measurable, thus added to the the model, or they will just appear as noise in the data, decreasing the **confidence score**. Any part of the of the series that cannot be predicted by the model is called **residual**.

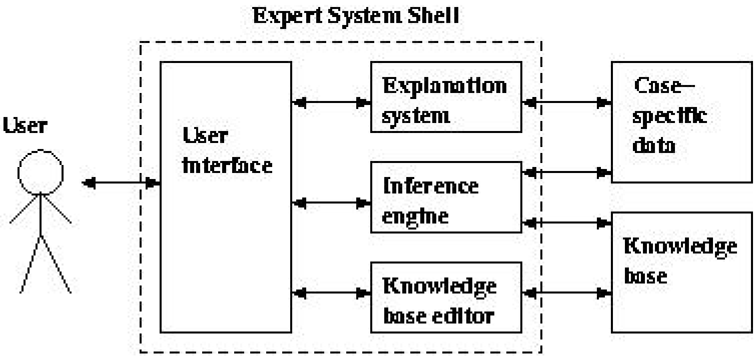
Reasoning Systems

Rule-based Systems (RBS)

Expert systems aim to capture the **knowledge** of a human expert in a doman and embody it within a **software** system.

Rule-based systems have the following:

- **Knowledge base —** Rules embodying expert knowledge about the problem domain.
- **Database —** Contains a set of known facts about the problem domain.
- **Inference engine —** carries out the reasoning process, using rules and facts.
- **Explaination Facilities —** Provides information to users about the reasoning steps that are being followed.
- **User interface —** communication between the user and the system.



Expert System Shells are programs and or/framework that can be customised to create a RBS. They are known as **shells** or **rule engines**.

The shell usually provides the *inference engine*, *explaination facilities*, and *infrastructure* for populating the knowledge base and the rules. Sometimes, they also provide interfaces for both users and developers.

The **inference process** used in a RBS is **deductive inference**, meaning that rules of logic are used to create new knowledge combining previous knowledge and rules.

Deductive inference

There are two main approaches to deductive inference:

- **Forward Chaining**
- **Backward Chaining**

Forward Chaining

Forward chaining uses rules like:

```
        If A and B Then C
```

To increase the number of facts in knowledge base.

The problem with this approach is controlling it. Different rules might be valid at different times, so you need to **indentify** (matching) them, and **decide** which one to use (conflict resolution). This makes it useful for RBS with **no specific goal**.

Backward Chaining

Backward chaining uses rules like:

```
C If A and B
```

The system tries to justify all the subrules that satisfies the **goal** until an answer is found. This might lead to many dead-ends before a solution.

Certainty factors in RBS

To improve the working of an RBS, a way of representing the degree of uncertainty related to both facts and rules.

There are many reasons why uncertainty is introduced:

- Information is incomplemte
- Information is not reliable
- Language use imprecise
- There is conflicting information
- Information is approximate

Certainty factors, also known as *confidence factors*, are a measure which represents a degree of confidence that some condition is *true*.

Certainty factors have to be combined when two rules are combined together.

```
CF(A and B) = min(CF(A), CF(B))
CF(A or  B) = max(CF(A), CF(B))
```

When applying a rule such as:

```
If P Then Q @ n
```

The uncertainty factor of Q is obtained by combining the CF of the rule and the CF of the fact

```
CF(Q) = CF(P) · n
```

Altought Confidence Factors resembles **probability**, they are not the same. There is no rigiorious mathematical foundation for CF. They are often altered with the design, and there is no strict way of evaluating the certainty of a rule and/or fact. More often than not though, they are extremely effective in practical applications.

Older RBS used Bayesian probability to handle uncertainty.

This model assumes that:

- Hypotesis are mutually exclusive and exhaustive
- Piece of evidence are conditially independent

This model is usually inaccurate, because the assumptions above are mostly never true, but the model has evolved into **belief networks** that are giving very promising results.

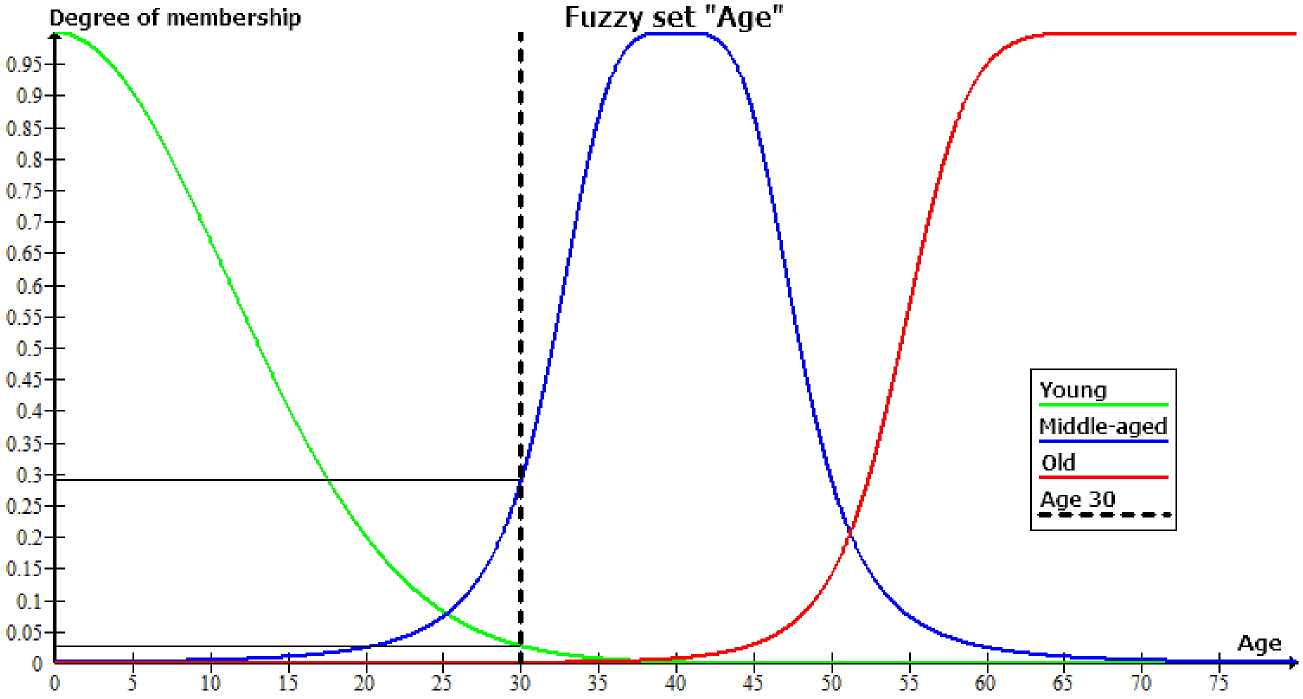
Fuzzy Logic

Fuzzy logic is an alternative method to boolean logic for determining the value of a property. Instead of being either true or false, a percentage of how much the current condition fits a definition is used.

For instance, the temperature could be either be represented by a boolean `cold, not cold` (*boolean*) or a set of values could be defined `20% Cold` (*fuzzy*), specifying a degree of how much the value fits in that category (*fuzzy measure*).

Fuzzy set theory and fuzzy logic provide a precise and mathematical basis for reasoning about uncertainty.

Fuzzy sets can be effectively shown in graphs, allowing to calculate the corresponding degree given the measurement. The actual measurement is definited as opposite of fuzzy *crisp*.



Fuzzy Sets can be used in RBS to define how much something fits a certain quality, instead of using an arbitrary change point with a specific temperature.

This is extremely successful in automatic control systems, like washing machines, air conditioner, etc.

Defuzzification

Defuzzification is the process of producing a quantifiable result in Crisp logic, given fuzzy sets and corresponding membership degrees. It is the process that maps a fuzzy set to a crisp set. It is typically needed in fuzzy control systems.

A common and useful defuzzification technique is center of gravity.

Hedges

Qualifying words could be provided using a mathematical definition.

Given a function `f` taking values `T`, the following could be defined:

```
        very_f(T) = f(T)²
        not_f(T) = 1 - f(T)
        slightly_f = f(T)¹/²
```

Case-based Reasoning

Another important Reasoning System is Case-Based reasoning. Differently from rule-based systems, and fuzzy logic, **does not use inference** (rules or logical deduction). CBR relies on the **analogy** between a problem a previous one, looking for the most similar problem that has been solved in the past, and applying the same solution.

In a CBR knowledge is stored as a **record of cases**.

Procedure

The steps towards creating a CBR are:

- Identifying attributes
- Identify cases
- Compare a new case, with the record of cases

cbr-architecture

To solve a new problem, the system does the following:

- **Retrieve** the most similar case(s) from the library
- **Reuse** the case(s) atempt to solve the problem

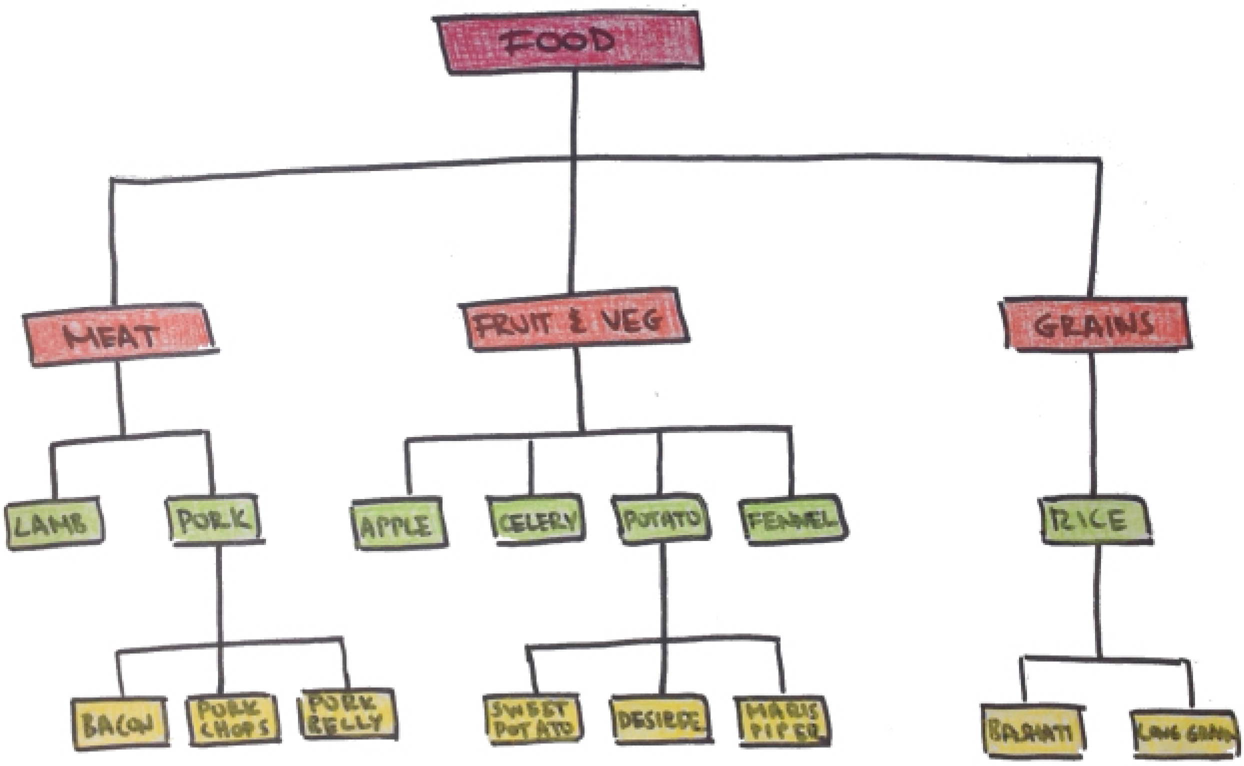
- **Revise** the proposed solution if necessary
- **Retain** the new solution as part of a new case

Similarity

CBRs often use the **nearest neighbour algorithm** to find the most similar case. This algorithm uses the distance of two points in a n-dimensional space.

The nearest neighbour approach is only effective with **numerical values**, for **nominal values** another way of specifying how much something is similar to something else must be specified. **Tables**, **Taxonomies**, **Ordered Lists**, etc are possible solutions.

Taxonomy-based comparison is suitable for comparing nominal values that can be classified into groups.



Adaptation

After having found the best match, the solution might need **adaptation**, since the case won't probably be the exact same. The system might either require a human expert to intervene, or use some other tool, like a neural network to adapt the results to the current case.

The newly created case, should not be directly added to the case-base, or it will become progressively **degraded**. They should first be analysed and accepted.

Advantages and Disadvantages

Advantages

- Knowledge based on previous data
- Causal relationships don't need indentification
- Less formal logic required
- Adding new knowledge is simple
- No need for general rules
- Most people reason by analogy

Disadvantages

- Indentify the most signifcant attributes
- All the attributes values must be found
- Decide an algorithm to measure similarity
- Make an effective adaptation strategy

When to use CBR

- There are records of previously solved cases
- Historical cases are used to solve new cases
- Human experts tend talk about examples, rather than rules
- The problem domain is not well-defined or well-understood
- Experience is as important as theoretical knowledge

Bayesian Belief Networks

Probability

The probability is calculated as: `positive outcomes ÷ total outcomes`.

Probabilities can be combined:

```
P(a OR b) = P(a) + P(b) // Disjoint and undependent

P(a OR b) = P(a) + P(b) - P(a ∩ b)

P(a AND b) = P(a) · P(b) // Disjoint and undependent

P(a AND b) = P(a | b) · (b)

not P(a) = 1 - P(a)
```

The **conditional probability** is the probability of `a` knowing that `b` has happened, and is written as:

```
P(a | b)
```

If `P(a | b) = P(a)` then `a` is completely unrelated to `b`.

Bayes' Theorem

In general `P(a | b) ≠ P(b | a)`, but `P(a | b) * P(b) = P(b | a) * P(a)` (because of the product rule), thus:

```
P(a | b) = P(b | a) · P(a) ÷ P(b)
```

This is known as the **Bayes' Theorem**, and allows to determine the probabiliy of *a* given *b*, from the probability of *b* given *a*.

Bayesian Classifier

Bayesian Classfiers are statistical tools that can predict the probability that a particular sample is a member of a particular class. One of the simplest Bayesian Classifier is known as the **Naïve Bayesian Classfier**, based on an *independence* assumption.

Bayesian Networks

Bayesian Belief Networks (BNNs) are a way of modeling probabilities based on data or knowledge to estimate probabilities of new events.