Machine Learning

Concepts, approaches and examples

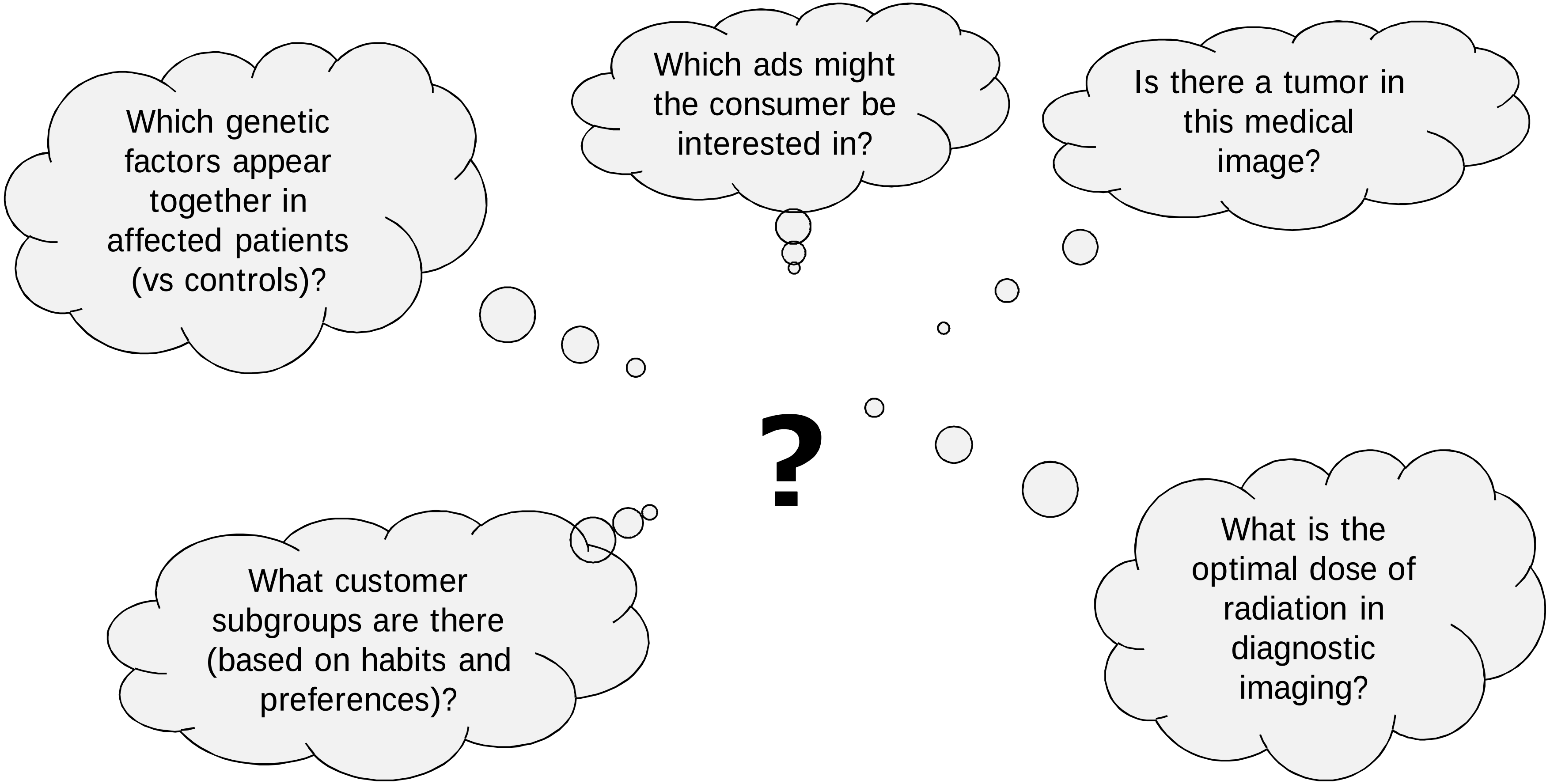
Learning goals

1. Understand the key concepts in machine learning.
2. Get an idea of key machine learning techniques and their fields of application.
3. Understand the phases of a machine learning project.

Contents

1. Approaches to machine learning
2. A walkthrough of a machine-learning task (Iris case)
   - Business objectives
   - Data understanding
   - Data preparation
   - Modelling
   - Evaluation
   - Deployment
3. Other application areas and approaches

Questions for machine learning



Which genetic factors appear together in affected patients (vs controls)?

Which ads might the consumer be interested in?

Is there a tumor in this medical image?

?

What customer subgroups are there (based on habits and preferences)?

What is the optimal dose of radiation in diagnostic imaging?

Machine learning

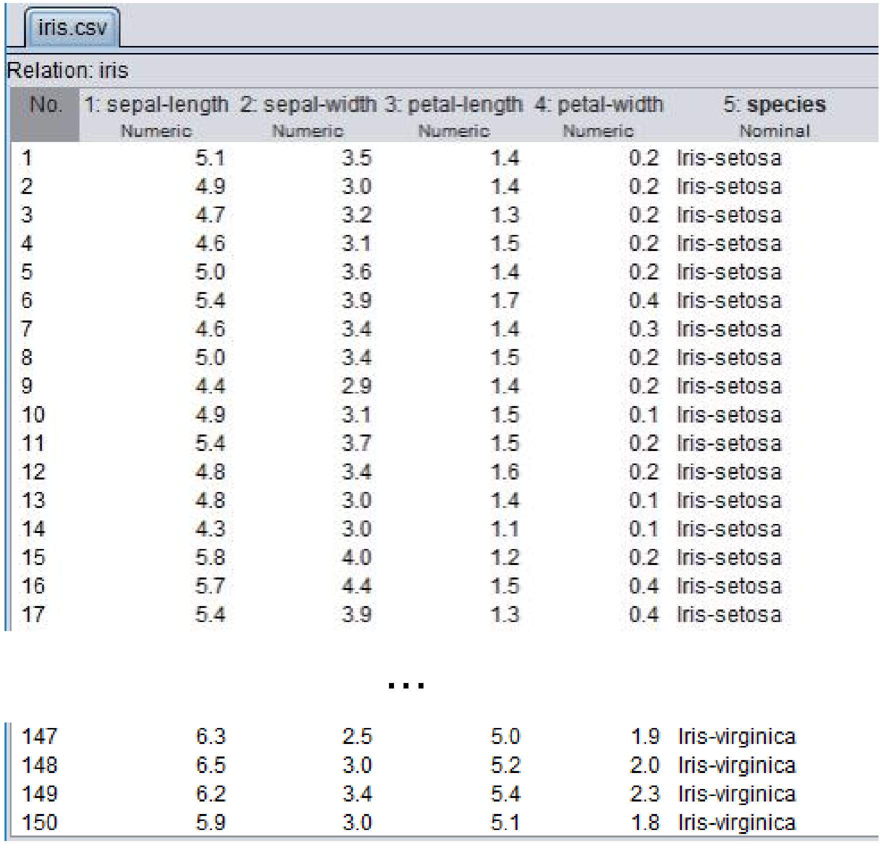
- **Machine learning** comprises of a set of techniques that give a computer the ability to learn without being explicitly programmed.
  - A means to achieve artificial intelligence.
- Particularly powerful in contexts where the set of ’rules’ can be defined easily.
  - Samuel, A. L., Some studies in Machine Learning using the Game of Checkers. IBM Journal, July 1959.
  - AlphaZero chess AI, see e.g. the article in The Guardian (December 7, 2017).

Three types of machine learning

1. **Supervised learning**
   - A training set (i.e. a data set with correct answers) is provided.
   - Example: recognize hand-written numbers from images.
     1. First give the algorithm a set of hand-written numbers with their human-classified counterparts.
     2. Build a model (a classifier) based on the training data.
     3. Then, use the model to predict the correct classification for new images.
2. **Unsupervised learning**
   - No training set is provided.
   - Example: find phrases of words that frequently appear together, as well as other linguistic structures.
3. **Reinforcement learning**
   - The software agents explore their environment and evolve towards the optimal solution.

Case: Iris

- The **Iris data set**[1] is one of the best known machine learning data sets.
  - The ’Hello World’ of ML: simple and understandable.
- Let’s carry out an example ML project based on the Iris data.



Relation: iris

No.	1: sepal-length	2: sepal-width	3: petal-length	4: petal-width	5: species
	Numeric	Numeric	Numeric	Numeric	Nominal
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
			…		
147	6.3	2.5	5.0	1.9	Iris-virginica
148	6.5	3.0	5.2	2.0	Iris-virginica
149	6.2	3.4	5.4	2.3	Iris-virginica
150	5.9	3.0	5.1	1.8	Iris-virginica

[1] Fisher, R.A. The use of multiple measurements in taxonomic problems" Annals of Eugenics, 7, Part II, 179-188 (1936)

Image: Miya.m., via Wikimedia Commons. Creative Commons Attribution-Share Alike 3.0 Unported licence.

How to start?

- Apply a **process model** for a data science task.
  - CRISP-DM
  - OSEMN
- CRISP-DM process model contains six phases.

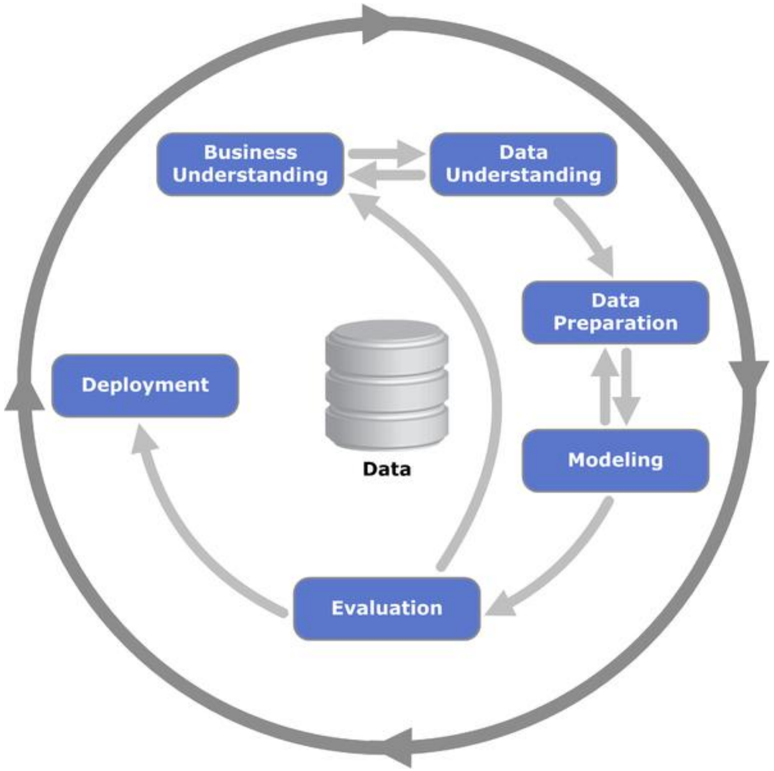


Image: CRISP DM process. Kenneth Jensen, via Wikimedia Commons, Creative Commons Attribution-Share Alike 3.0 Unported licence.

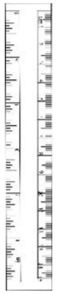
1/6

Phase 1: Business understanding

The indicator on the slides points to the corresponding CRISP-DM Phase

- Define the **business goals** for the ML project.
  - Business is applied in broad sense: can be money or welfare.
- What are we trying to achieve?
- In the Iris case, let’s state the goal as defining a method for the lab assistant to find the correct species based on the measurements.
- We recognize that this is a classification problem.

+



= ?

Image (I. setosa): Miya.m., via Wikimedia Commons. Creative Commons Attribution-Share Alike 3.0 Unported licence.

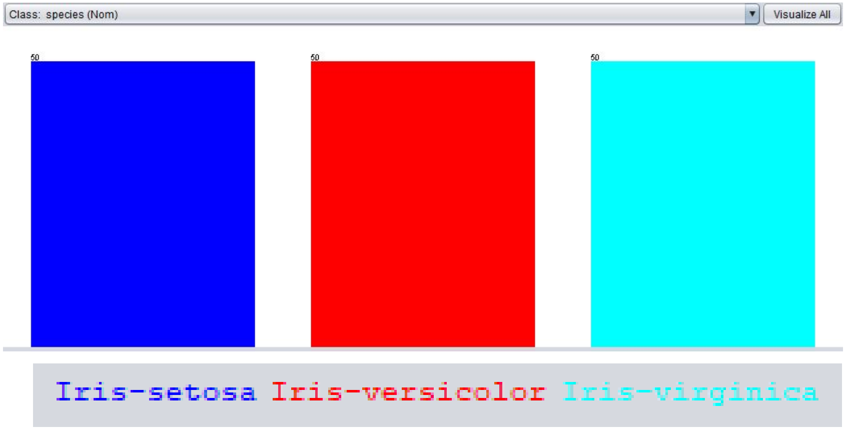
Phase 2: Data understanding

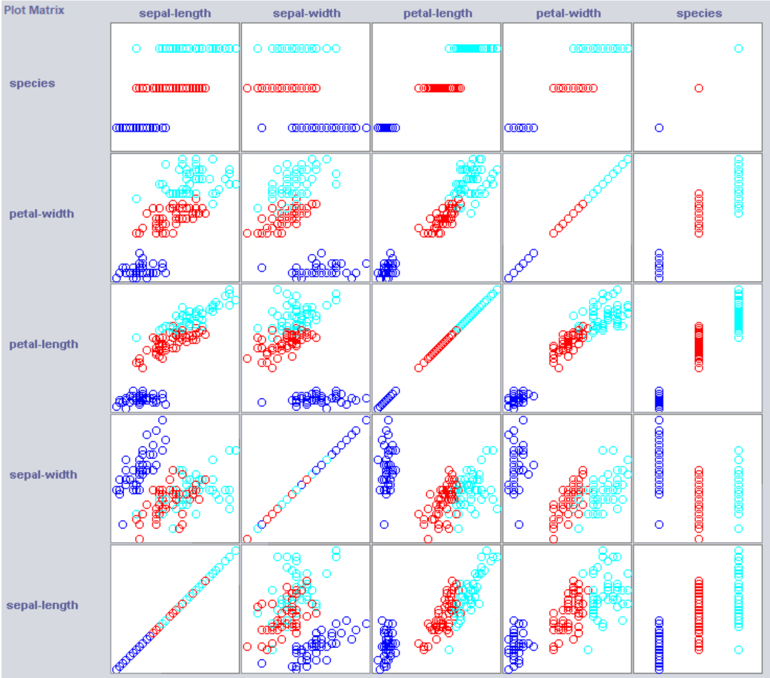
- We should **understand the data**:
  - How the data is collected?
  - How tha data is encoded?
  - What kind of distributions do the variables have?
  - Are there missing data points or outliers?

- GIGO principle: Garbage In, Garbage Out.

Phase 2: Data understanding

- Techniques such as **histograms** and pairwise **scatterplots** help get an overview of the data.
  - ML software and/or libraries provide this functionality.
- The graphs reveal that Iris setosa plants can be recognized by looking at petal length only (alternatively, petal width).
  - *Iris versicolor* and *Iris virginica* are not separable, though, by any single variable, or combination of values.





3/6

Phase 3: Data preparation

- The **data preparation** phase contains all the necessary modifications for the data to have it ready for analysis.
- This step may take up to 90% of the time.
  - The format may be wrong.
    - Typically, CSV format is the *lingua franca*.
    - Not feasible for very large/complex data such as complex images.
  - There can be more than one data set to combine.
  - Errors and missing data need manipulation.
  - The interpretation and encoding of variables needs to be verified.
- The iris data set contains no known errors or outliers.
- However, the correct interpretation of the variable types needs to be checked.

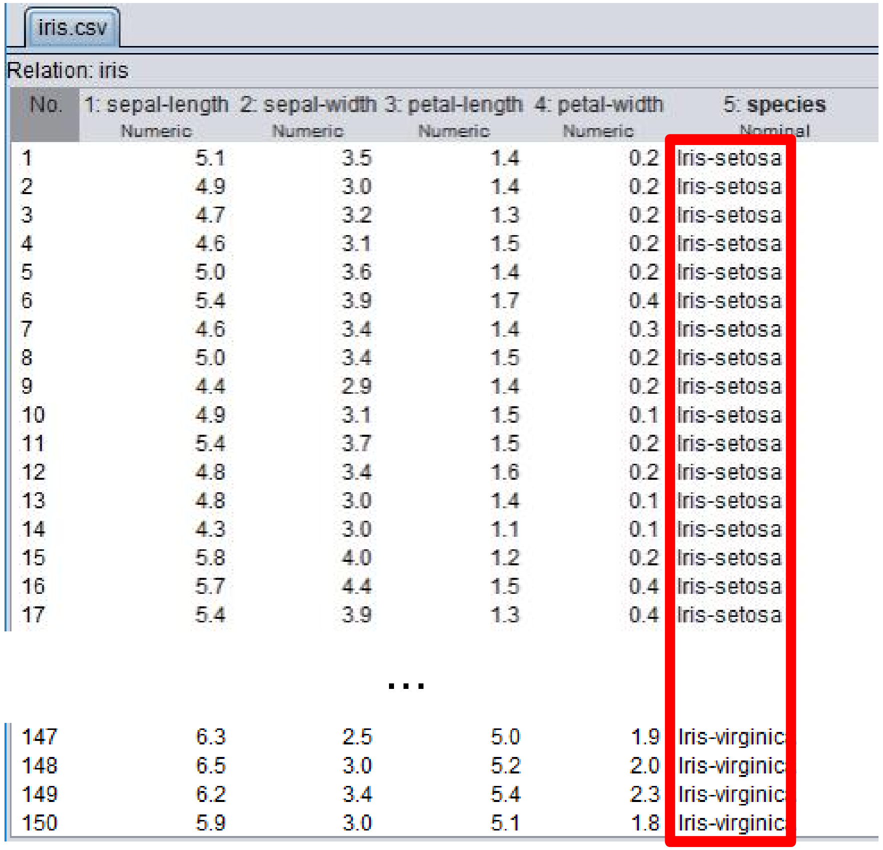
Phase 4: modelling

- **Modelling** refers to building the machine learning ”engine” that turns the data into knowledge.
- In the iris example, we have:
  - Four **explanatory variables**
  - One **response variable**
- Thus, we want to build a model that turns the values of the response variables into a predicted value of the (nominal) response variable.
  - This is the general goal of a classifier.
  - The response variable carries the business value.
  - The explanatory variable values are easy to obtain in comparison to the response variable.

4/6

Training data

- To build a classifier, we need **training data**.
- The training data must have the ”correct answers” for the response variable (i.e. species), accompanied with the values of the explanatory variables.
- This is exactly what we have!



iris.csv

Relation: iris

No.	1: sepal-length	2: sepal-width	3: petal-length	4: petal-width	5: species
	Numeric	Numeric	Numeric	Numeric	Nominal
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
		...			
147	6.3	2.5	5.0	1.9	Iris-virginic
148	6.5	3.0	5.2	2.0	Iris-virginic
149	6.2	3.4	5.4	2.3	Iris-virginic
150	5.9	3.0	5.1	1.8	Iris-virginic

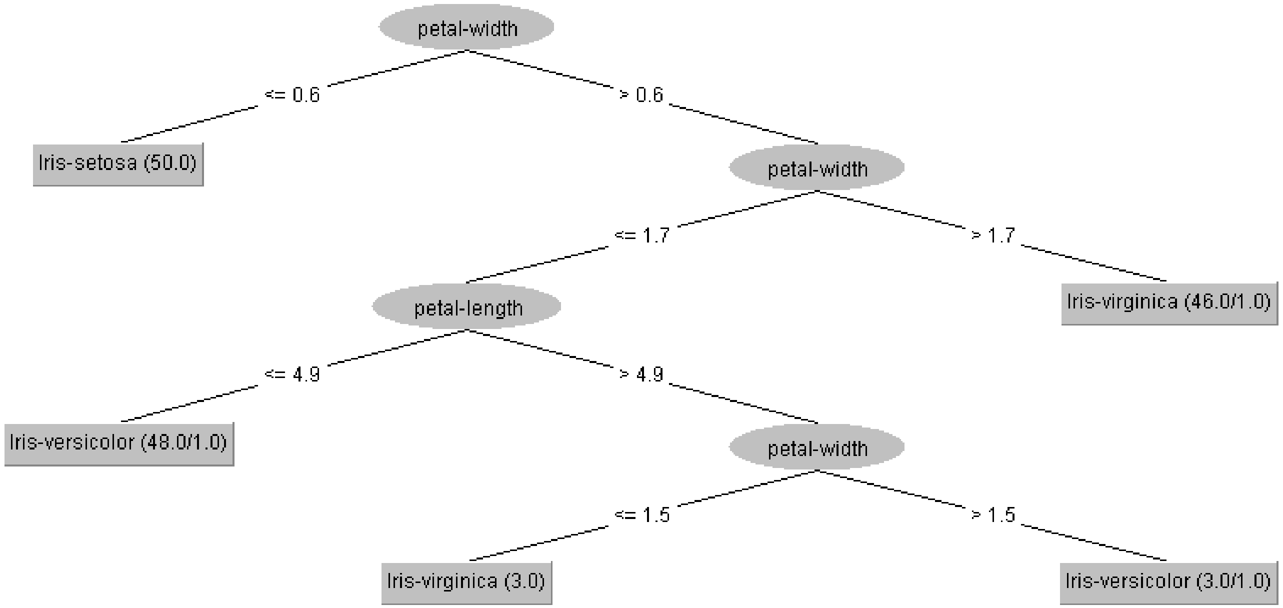
4/6

Selection of ML method

- How do we solve the classification problem?
- Possible techniques include:
  - kNN (k nearest neighbours)
  - Decision tree
  - Random forest
  - Neural network
- In this example, let’s build a **decision tree** classifier.
  - Simple and easy to understand
  - Transparent (the model can be decomposed, understood and explained).
- For now, we just obtain the tree.
  - Later, we will study how the tree is constructed.

Modelling for the Iris data

- The ML algorithm produces a model.
  - In this case, the model is a decision tree (with a confidence factor threshold of 0.25).

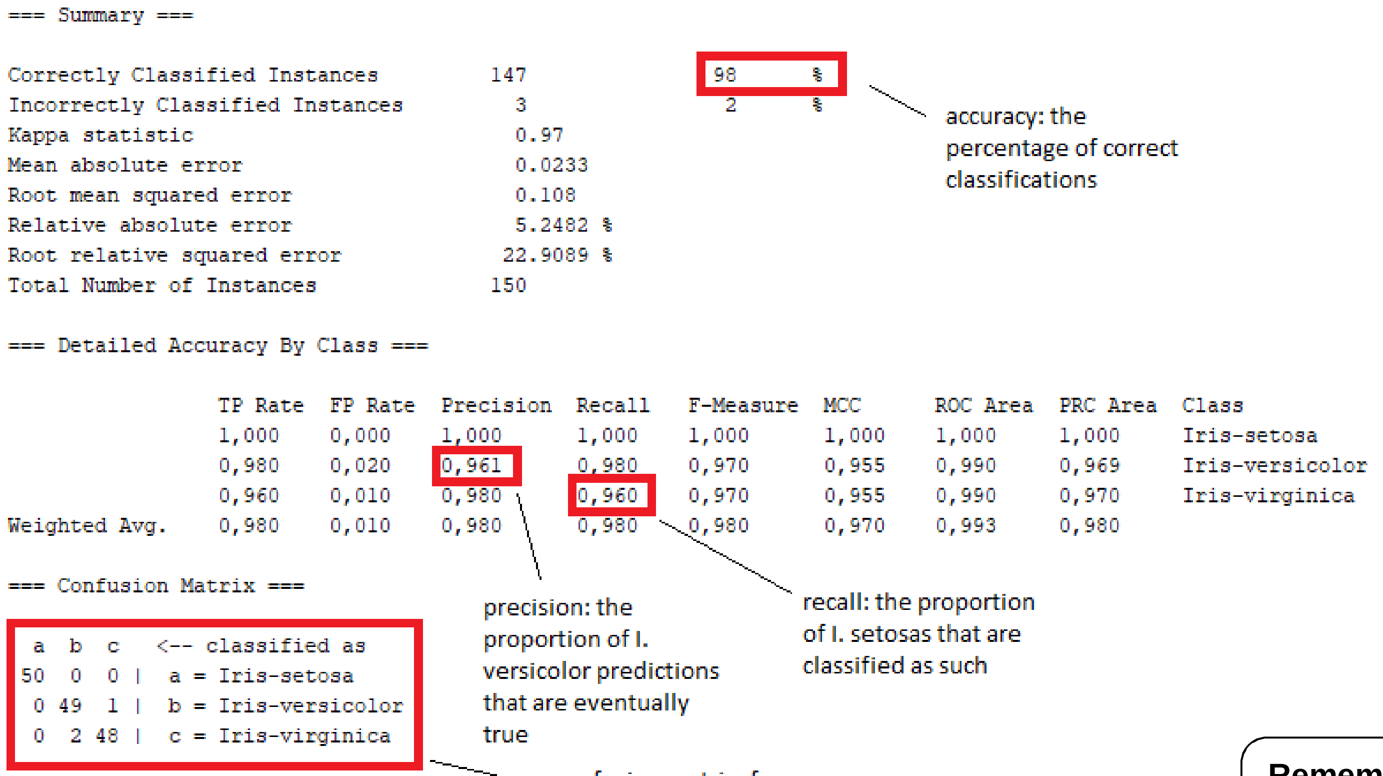


Phase 5: evaluation

- The key question is: Is this model good?
  - Can reliable predictions be made on it?
- Measures for a classification model's goodness include:
  - **Accuracy**
    - The proportion of correct predictions
  - **Precision**
    - The probability of a predicted class membership being correct.
  - **Recall**
    - The probability of being predicted to a given class, provided that an observation falls into that class in reality.
- To familiarize oneself with the measures, we evaluate the model goodness from the training set.
  - Such an approach is prone to model overfitting, which we well address later.

5/6

Evaluating the model



```
=== Summary ===

Correctly Classified Instances         147               98      %
Incorrectly Classified Instances         3                2      %
Kappa statistic                          0.97
Mean absolute error                      0.0233
Root mean squared error                  0.108
Relative absolute error                  5.2482 %
Root relative squared error             22.9089 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     Iris-setosa
                 0,980    0,020    0,961      0,980    0,970      0,955    0,990     0,969     Iris-versicolor
                 0,960    0,010    0,980      0,960    0,970      0,955    0,990     0,970     Iris-virginica
Weighted Avg.    0,980    0,010    0,980      0,980    0,980      0,970    0,993     0,980

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

accuracy: the percentage of correct classifications

precision: the proportion of I. versicolor predictions that are eventually true

recall: the proportion of I. setosas that are classified as such

confusion matrix, for the breakdown of classification performance

Remember:
Danger of model overfitting.

5/6

Evaluating the model

- In the previous example, we evaluated the model goodness from the training set.
- This is dangerous, as the constructed model is always prone to **model overfitting**.
  - That is, the model may grasp the peculiarities and random properties of the training set.
  - If this model is evaluated using the same set, we get an overestimate of the model’s performance.
- The model overfitting tends to be more severe when
  1. The number of observations in the training set is small
  2. The number of variables is high (”curse of dimensionality”)

Model overfitting

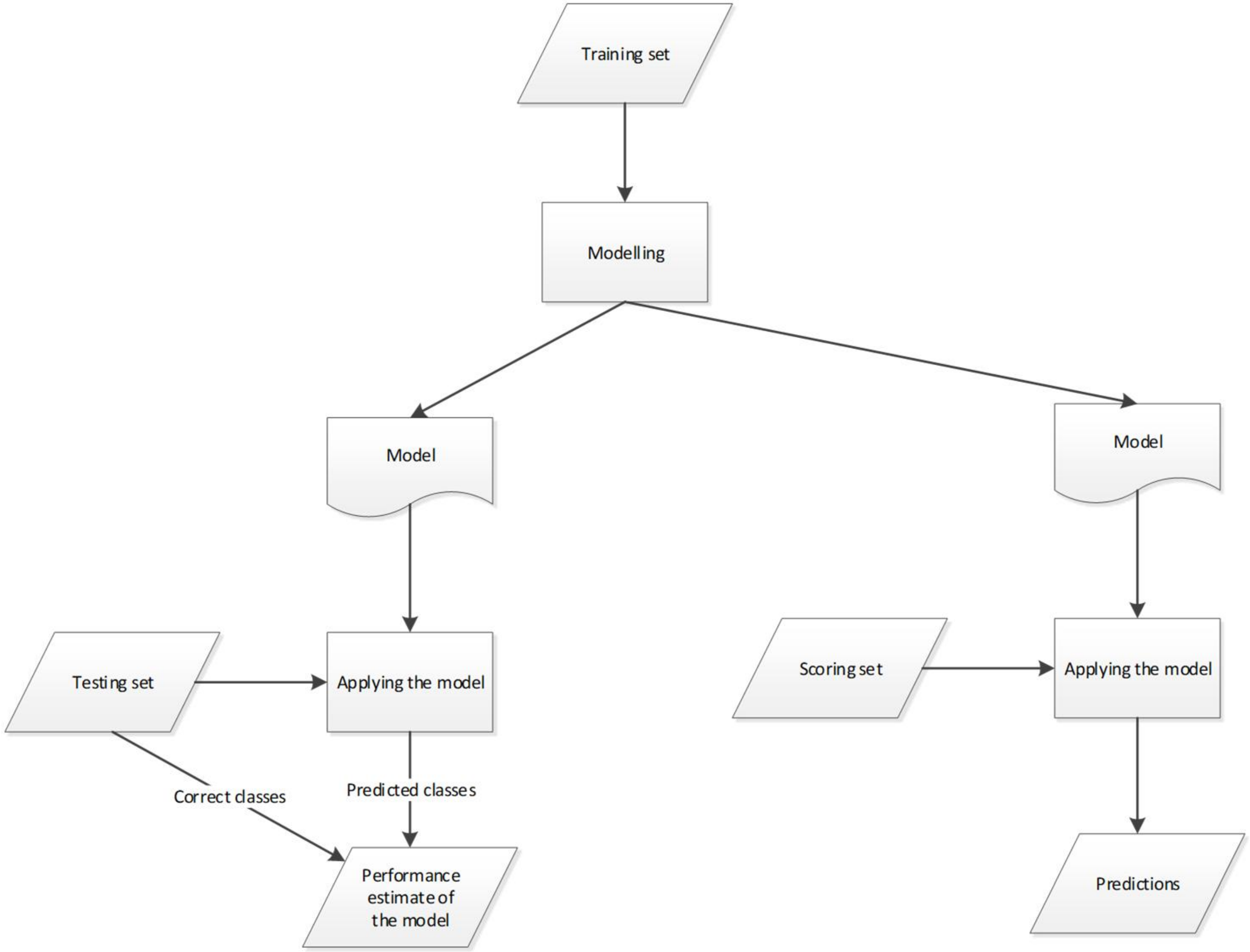
- The ML software produces the model (e.g. a decision tree) based on the training set.
- When the data set is small, the model can be based on rules that are not applicable in the general population, and, consequently, for any data set drawn from that. This is called **model overfitting**.
- Example: the goal is to construct a decision tree that classifies people into left and right handed persons based on a very large number of variables descibing their external characteristics.
  - Let's assume that there are 20 people in the training set, 3 of whom are left-handed.
  - It is certain to find a set of characteristics that correctly specifies these 3 persons. For example, it may turn out that all of them have either a hearing aid or shoe size 41, whereas none of the right-handed happens to satisfy this criterion.
  - The resulting decision tree matches the training set perfectly. 100% accuracy!
  - However, as the tree is applied to a new set of individuals, it turns out to be useless.
  - We tried to use a seriously overfit model.
- **Validation** reveals model overfitting.
  - It should always be done.
  - Only a validated model can be reliad on for decision making.
- Note: model overfitting does not imply that the training set would be inherently different than the general population.
  - The problem exists even if the training set is a proper, random sample of the general population.
  - It is a consequence of the limited sample size.

5/6

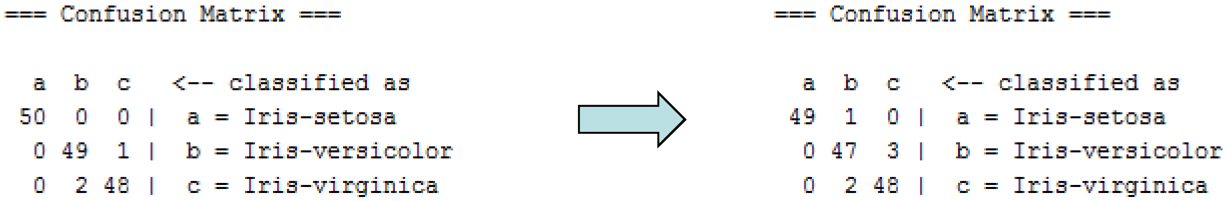
Validation

- The purpose of **validation** is to evaluate the goodness of a model by data that has not been used in the construction of the model.
- Idea: use two data sets:
  - training set (classes known, used in model construction only)
  - testing set (classes known, used in validation only)
    - a.k.a test set, validation set
- The validation methods differ from each other on how the training and validation sets are formed.
  - If the data size is limited, same date needs to be ”recycled” in the training and testing sets.

Analysis pipeline (with validation) 5/6



Iris case: validation



```
=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

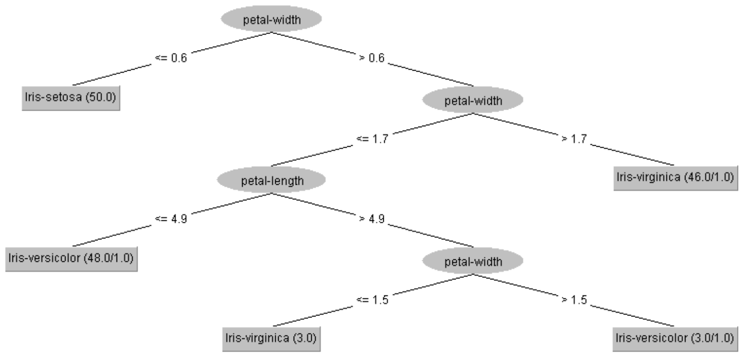
- The confusion matrices for a decision tree obtained from the Iris data set:
  - The accuracy calculated from the training set (on the left) is 98%.
    - This corresponds to no validation.
  - The accuracy estimate obtained by a new data set (on the right), is 96%.
- The estimate of the accuracy should be based on validation (right).
  - The limited data size may still make this point estimate for the accuracy unreliable.
  - However, it is corrected for model overfitting.
- Based on the validated data, we may conclude that the decision tree classifier with the chosen parameters passes evaluation.
  - It is able to predict the Iris species accurately.

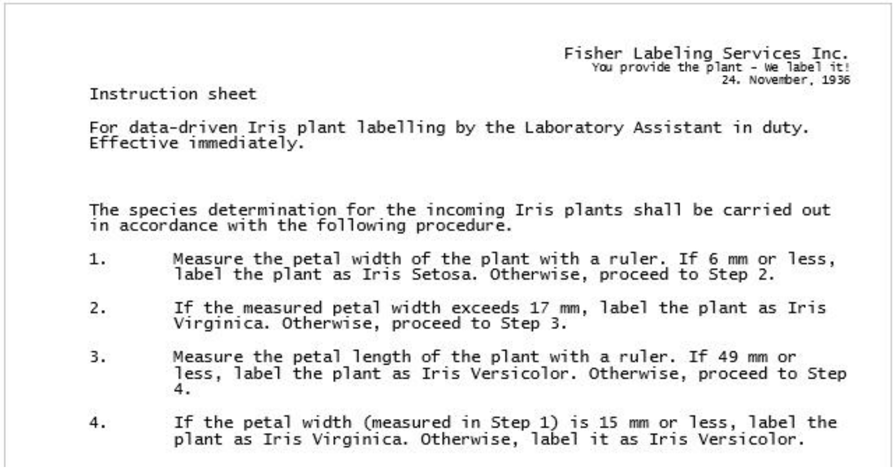
Phase 6: Deployment

- In the final phase, the outcome of the ML study is applied in real life.
- The outcome of this phase can be a **recommendation** for:
  - Applying the generated model in business.
  - Rejecting the idea
  - Collecting new data

Deployment in the Iris case

- The validated model could be taken into use in **daily business**.
- This concludes the case.




Fisher Labeling Services Inc.
You provide the plant - We label it!
24. November, 1936

Instruction sheet

For data-driven Iris plant labelling by the Laboratory Assistant in duty. Effective immediately.

The species determination for the incoming Iris plants shall be carried out in accordance with the following procedure.

1. Measure the petal width of the plant with a ruler. If 6 mm or less, label the plant as Iris Setosa. Otherwise, proceed to Step 2.
2. If the measured petal width exceeds 17 mm, label the plant as Iris Virginica. Otherwise, proceed to Step 3.
3. Measure the petal length of the plant with a ruler. If 49 mm or less, label the plant as Iris Versicolor. Otherwise, proceed to Step 4.
4. If the petal width (measured in Step 1) is 15 mm or less, label the plant as Iris Virginica. Otherwise, label it as Iris Versicolor.

Real classification problems

- The Iris case was an extremely simple example for learning purposes.
- Yet, a large number of real-life problems are **classification problems**:
  - Automating decision-making
    - ”Can a line of credit be established for this client?”
  - Medical diagnostics
    - ”Should we suspect condition X for this patient, based on his/her blood test values?”
  - Image recognition
    - ”Is there a malign tumor in this X-ray image?”
    - ”Which object appear in this photograph?”
      - Example of multi-class classification

Neural networks as classifiers

- Simple classification problems are often solved efficiently with basic ML methods (e.g. decision trees).
- Complex problems (such as image recognition) often demand **artificial neural networks** (ANN).
- A lot of similarities to the Iris example:
  - Overall problem definition (predicting class membership)
  - Measures for goodness of classification (accuracy etc.)
  - Validation techniques (cross-validation etc.)
- The modelling phase requires more complex machinery.
  - Possibly affects the data preparation phase as well.

A neural network

- A **neural network** is a model that contains:
  - Input nodes (for input variables).
  - Output nodes (for possible class memberships).
  - One or more hidden layers of nodes.
  - Edges that connect the nodes.
- Edges have weights that update when an ANN is trained.
- Each hidden node turns the input values into an output value. Two components:
  - An adder that sums the input
  - An activation function that maps the weighted sum into an output value.

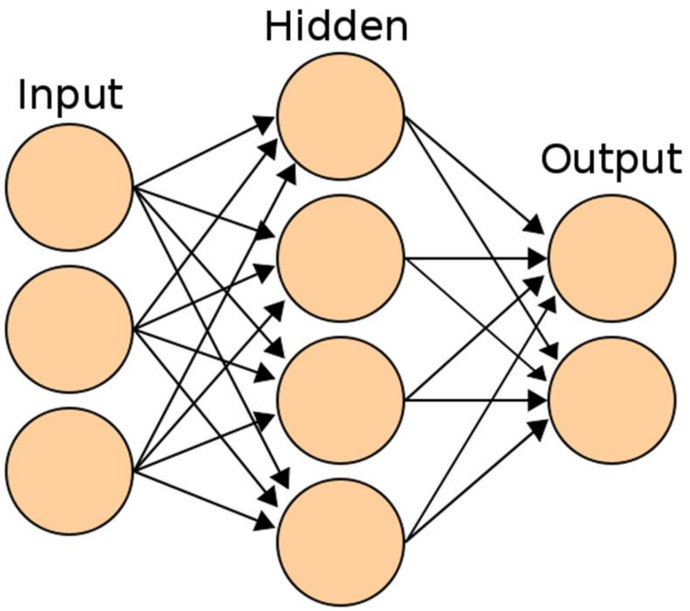


Image: JokerXtreme, via Wikimedia Commons, Creative Commons Attribution-ShareAlike licence.

Deep learning

- **Deep learning** refers to tackling complex problems with **convolutional neural networks** (CNNs).
- Commonly used in image classification.
- Computationally heavy.
  - GPUs provide additional power.
- A CSS is an ANN with a stack of specialized layers.
  - The specialized layers reduce the number of inputs in a carefully designed way.
  - For instance:
    - A convolutional layer combines the pixel information of the nearby pixels into a single value using a prespecified kernel (matrix).
    - Pooling layers further diminish dimensionality.

Other questions in ML

- In the previous case, we considered classifiers.
  - Also the use of neural networks was shown for classification.
- There are other important groups of ML problems:
  - Predicting **numerical values**
    - Linear regression modes, neural networks
  - Finding **hidden structure** in the data
    - Clustering algorithms (k-means, hierarchical clustering, ...)
  - Making **recommendations**
    - Association methods, neighbourhood search (kNN), ...

Take-home message

1. Machine learning provides a means to **learn from the data**.
2. The methods are divided into **supervised**, **unsupervised** and **reinforcment learning** methods.
3. A machine learning project benefits from using a **process model**.
4. Remember to **validate** your results.
5. The key ML tasks are **classification**, **clustering**, **numerical prediction** and **association/recommendation**.