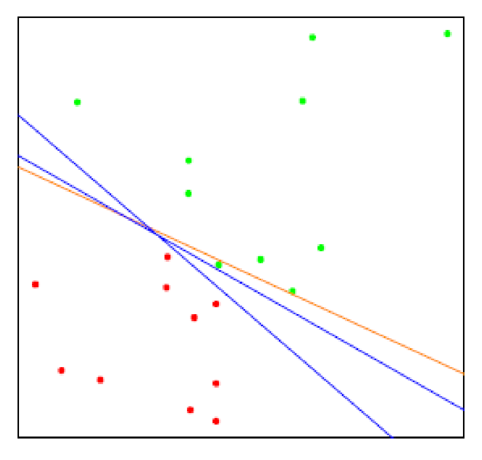


Parallel & Scalable Data Analysis

Introduction to Machine Learning Algorithms

Dr. – Ing. Morris Riedel

Adjunct Associated Professor

School of Engineering and Natural Sciences, University of Iceland

Research Group Leader, Juelich Supercomputing Centre, Germany

LECTURE 1

Machine Learning Fundamentals

November 23th, 2017

Ghent, Belgium

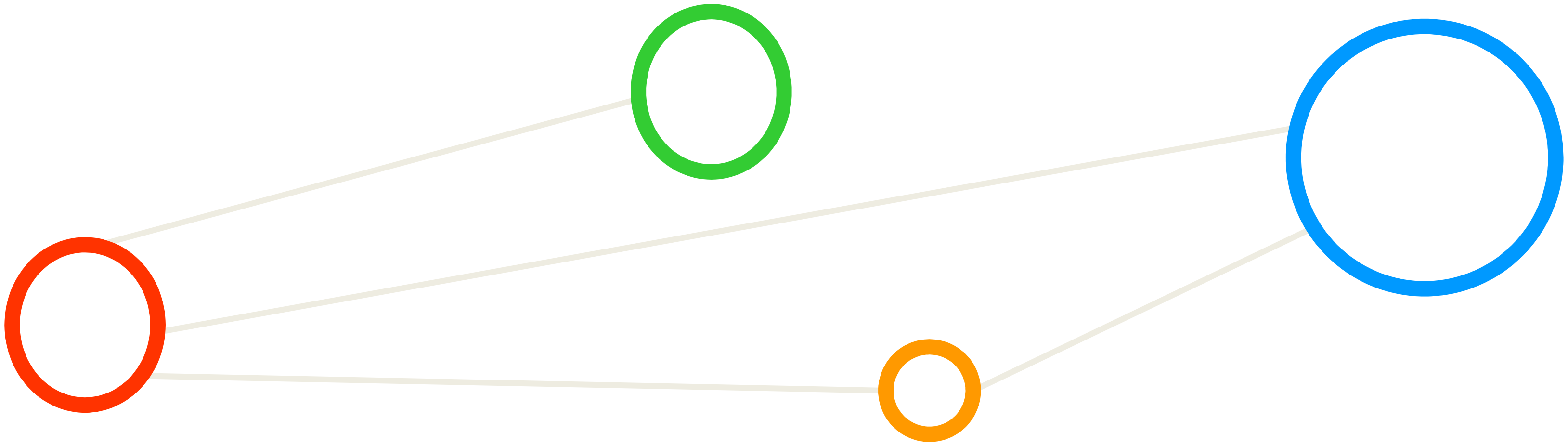
UNIVERSITY OF ICELAND

SCHOOL OF ENGINEERING AND NATURAL SCIENCES

FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH FORSCHUNGSZENTRUM

Outline



Outline of the Course

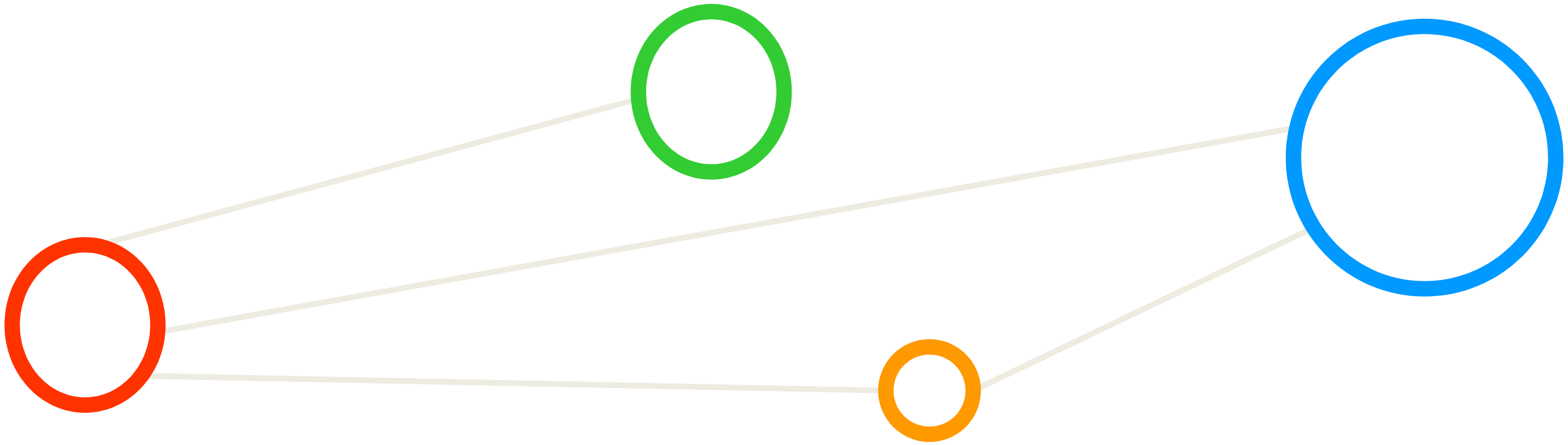
1. Machine Learning Fundamentals
2. Unsupervised Clustering and Applications
3. Supervised Classification and Applications
4. Classification Challenges and Solutions
5. Regularization and Support Vector Machines
6. Validation and Parallelization Benefits

Outline

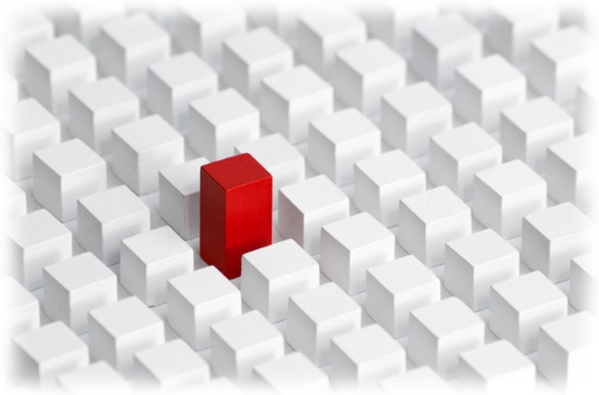
- Machine Learning Basics
  - Motivation
  - Methods Overview
  - Simple Application Example
  - Perceptron Learning Model
  - Decision Boundary & Linear Seperability

- Learning from Data
  - Systematic Process to Support Learning
  - Predictive and Descriptive Tasks
  - Different Learning Approaches
  - Terminologies
  - Model Evaluation with Testing

Machine Learning Basics



Motivation



- Rapid advances in data collection and storage technologies in the last decade
  - Extracting useful information is a challenge considering ever increasing massive datasets
  - Traditional data analysis techniques cannot be used in growing cases (e.g. memory limits)

- **Machine learning / Data Mining is a technology that blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data**
- **Machine Learning / Data Mining is the process of automatically discovering useful information in large data repositories ideally following a systematic process**

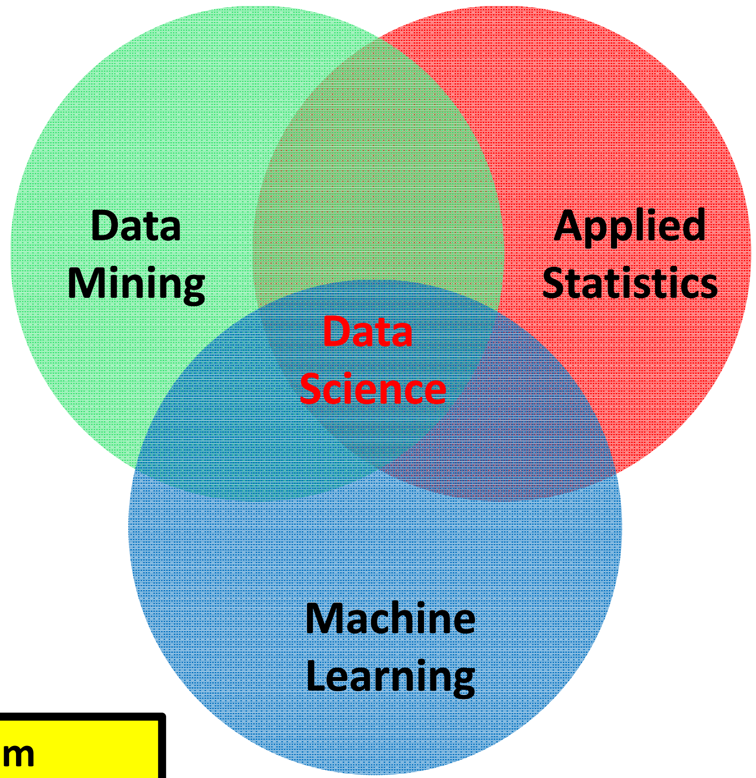
modified from [1] Introduction to Data Mining

- Machine Learning & Statistical Data Mining
  - Traditional statistical approaches are still very useful to consider
  - E.g. in order to reduce large quantities of data to most expressive datasets

Machine Learning Prerequisites

1. Some pattern exists
2. No exact mathematical formula
3. Data exists

- Idea 'Learning from Data' shared with a wide variety of other disciplines
  - E.g. signal processing, data mining, etc.
- Challenge: Data is often complex

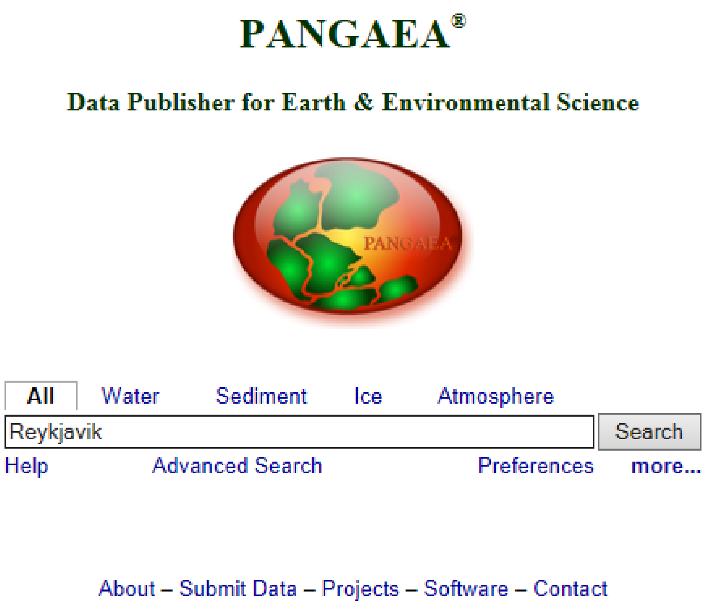


- **Machine learning is a very broad subject and goes from very abstract theory to extreme practice ('rules of thumb')**

Examples of Real Data Collections

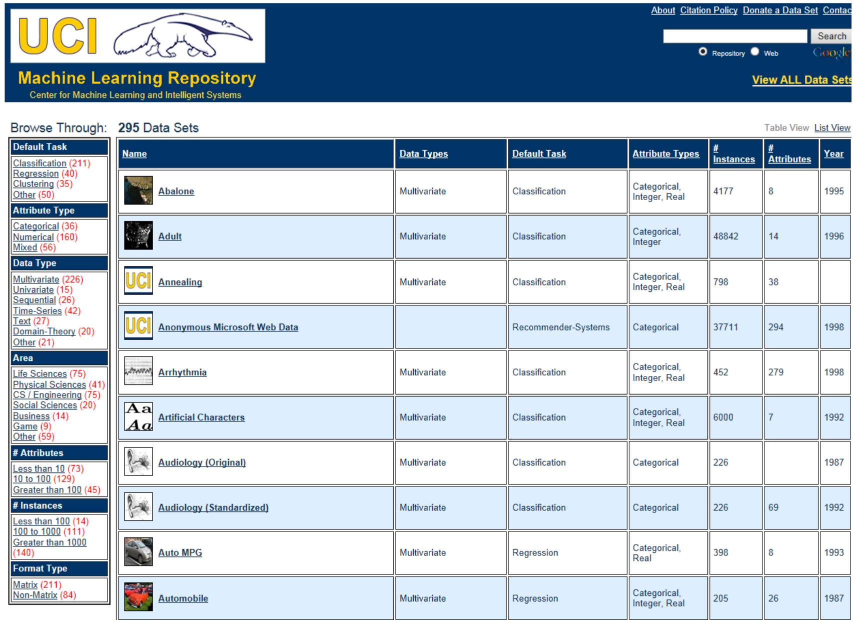
- Data collection of the earth and environmental science domain
  - Different from the known 'UCI machine learning repository examples'

(real science datasets examples)



[2] PANGAEA data collection

(examples for learning & comparisons)



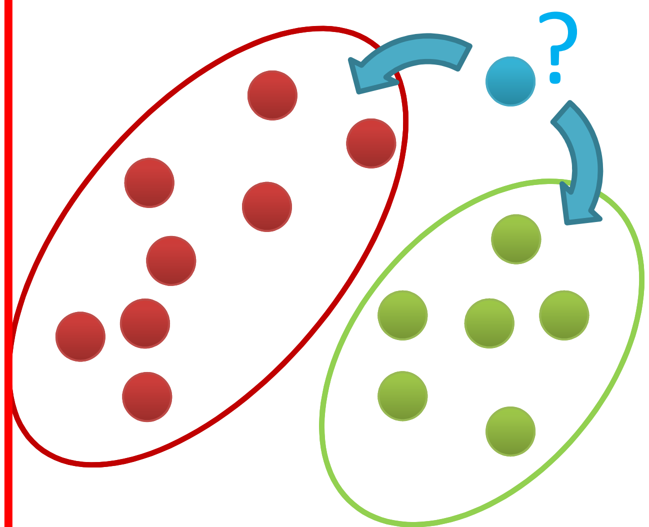
Name	Data Types	Default Task	Attribute Types	# Instances	# Attributes	Year
Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8	1995
Adult	Multivariate	Classification	Categorical, Integer	48842	14	1996
Annealing	Multivariate	Classification	Categorical, Integer, Real	798	38	
Anonymous Microsoft Web Data		Recommender-Systems	Categorical	37711	294	1998
Arrhythmia	Multivariate	Classification	Categorical, Integer, Real	452	279	1998
Artificial Characters	Multivariate	Classification	Categorical, Integer, Real	6000	7	1992
Audiology (Original)	Multivariate	Classification	Categorical	226		1987
Audiology (Standardized)	Multivariate	Classification	Categorical	226	69	1992
Auto MPG	Multivariate	Regression	Categorical, Real	398	8	1993
Automobile	Multivariate	Regression	Categorical, Integer, Real	205	26	1987

[3] UCI Machine Learning Repository

Methods Overview

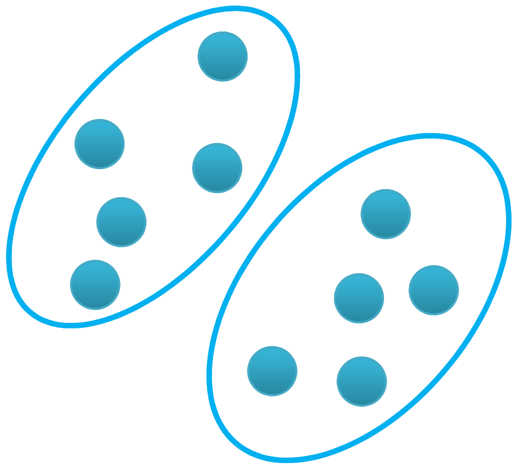
- **Machine learning methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction**

Classification



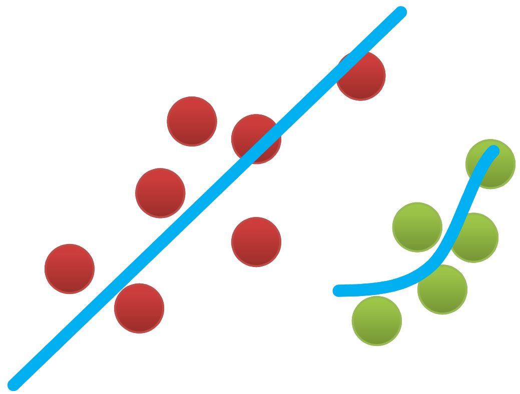
- Groups of data exist
- New data classified to existing groups

Clustering



- No groups of data exist
- Create groups from data close to each other

Regression



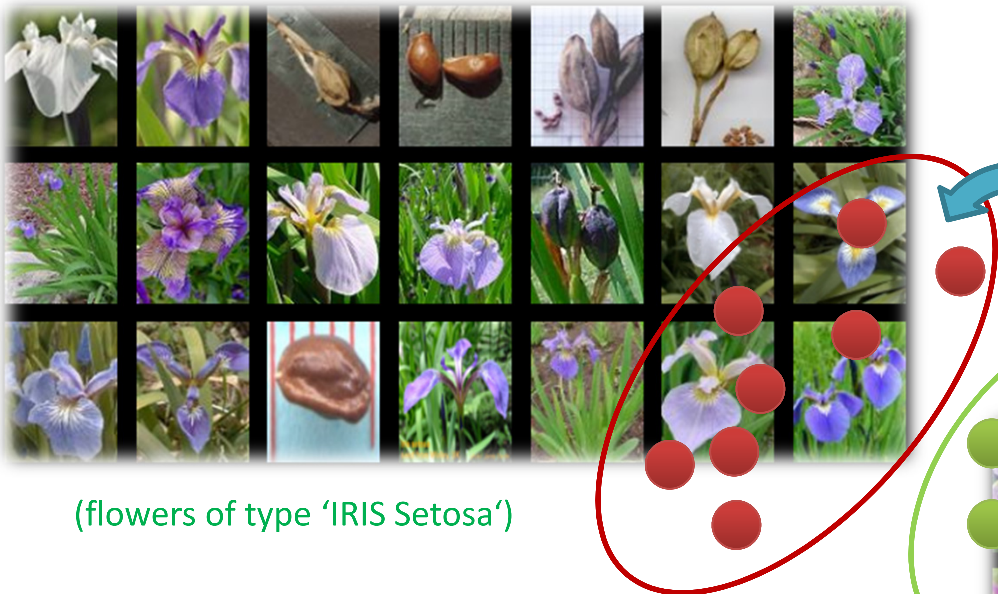
- Identify a line with a certain slope describing the data

➢ **The concrete focus of this course is classification using one specific technique out of many others**

Simple Application Example: Classification of a Flower

(1) Problem Understanding Phase

(what type of flower is this?)



(flowers of type 'IRIS Setosa')

(flowers of type 'IRIS Virginica')

- Groups of data exist
- New data classified to existing groups

[4] Image sources: Species Iris Group of North America Database, www.signa.org

The Learning Problem in the Example

(flowers of type 'IRIS Setosa') (flowers of type 'IRIS Virginica')

[4] Image sources: Species Iris Group of North America Database, www.signa.org

Learning problem: A prediction task

- Determine whether a new Iris flower sample is a "Setosa" or "Virginica"
- Binary (two class) classification problem
- What attributes about the data help?

(what type of flower is this?)

Feasibility of Machine Learning in this Example

1. Some pattern exists:
   - Believe in a 'pattern with 'petal length' & 'petal width' somehow influence the type
2. No exact mathematical formula
   - To the best of our knowledge there is no precise formula for this problem
3. Data exists
   - Data collection from UCI Dataset „Iris"
   - 150 labelled samples (aka 'data points')
   - Balanced: 50 samples / class

[5] Image source: Wikipedia, Sepal

(four data attributes for each sample in the dataset)

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm

(one class label for each sample in the dataset)

- class: Iris Setosa, or Iris Versicolour, or Iris Virginica

(2) Data Understanding Phase

[6] UCI Machine Learning Repository Iris Dataset

Exercises

Understanding the Data – Check Metadata

- First: Check metadata if available (metadata is not always available in practice)
  - Example: Downloaded `iris.names` includes metadata about data

```
1. Title: Iris Plants Database
     Updated Sept 21 by C.Blake - Added discrepency information

2. Sources:
     (a) Creator: R.A. Fisher
     (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
     (c) Date: July, 1988

                         ...

5. Number of Instances: 150 (50 in each of three classes)

6. Number of Attributes: 4 numeric, predictive attributes and the
class

7. Attribute Information:
   1. sepal length in cm
   2. sepal width in cm
   3. petal length in cm
   4. petal width in cm
   5. class:
      -- Iris Setosa
      -- Iris Versicolour
      -- Iris Virginica
```

(Subject, title, or context)

(author, source, or creator)

(number of samples, instances)

(attribute information)

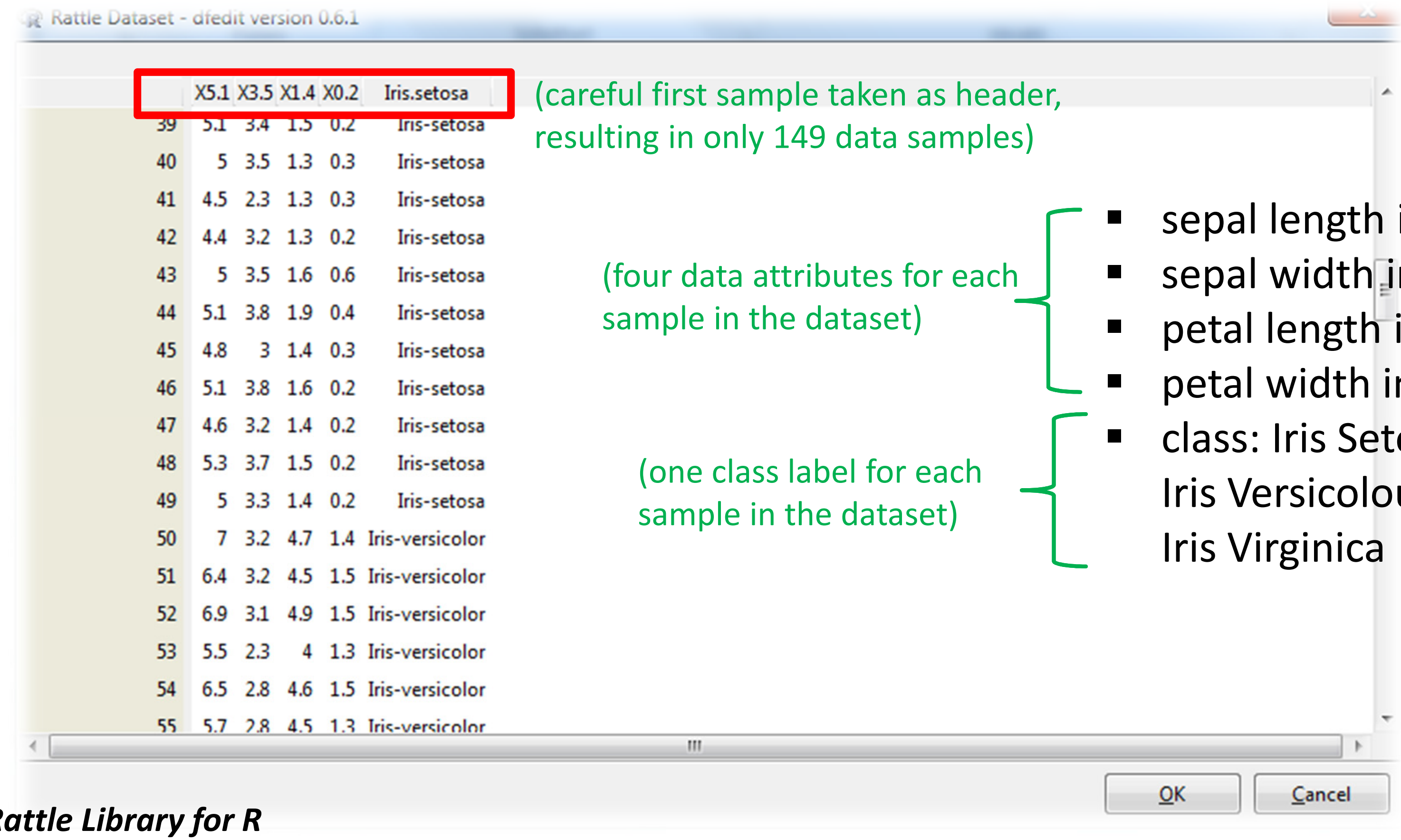
(detailed attribute information)

(detailed attribute information)

[6] UCI Machine Learning Repository Iris Dataset

Understanding the Data – Check Table View

- Second: Check table view of the dataset with some samples
  - E.g. Using a GUI like ‘Rattle‘ (library of R), or Excel in Windows, etc.
  - E.g. Check the first row if there is header information or if is a sample


Rattle Dataset - dfedit version 0.6.1

	X5.1	X3.5	X1.4	X0.2	Iris.setosa
39	5.1	3.4	1.5	0.2	Iris-setosa
40	5	3.5	1.3	0.3	Iris-setosa
41	4.5	2.3	1.3	0.3	Iris-setosa
42	4.4	3.2	1.3	0.2	Iris-setosa
43	5	3.5	1.6	0.6	Iris-setosa
44	5.1	3.8	1.9	0.4	Iris-setosa
45	4.8	3	1.4	0.3	Iris-setosa
46	5.1	3.8	1.6	0.2	Iris-setosa
47	4.6	3.2	1.4	0.2	Iris-setosa
48	5.3	3.7	1.5	0.2	Iris-setosa
49	5	3.3	1.4	0.2	Iris-setosa
50	7	3.2	4.7	1.4	Iris-versicolor
51	6.4	3.2	4.5	1.5	Iris-versicolor
52	6.9	3.1	4.9	1.5	Iris-versicolor
53	5.5	2.3	4	1.3	Iris-versicolor
54	6.5	2.8	4.6	1.5	Iris-versicolor
55	5.7	2.8	4.5	1.3	Iris-versicolor

OK Cancel

(careful first sample taken as header, resulting in only 149 data samples)

(four data attributes for each sample in the dataset)

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm

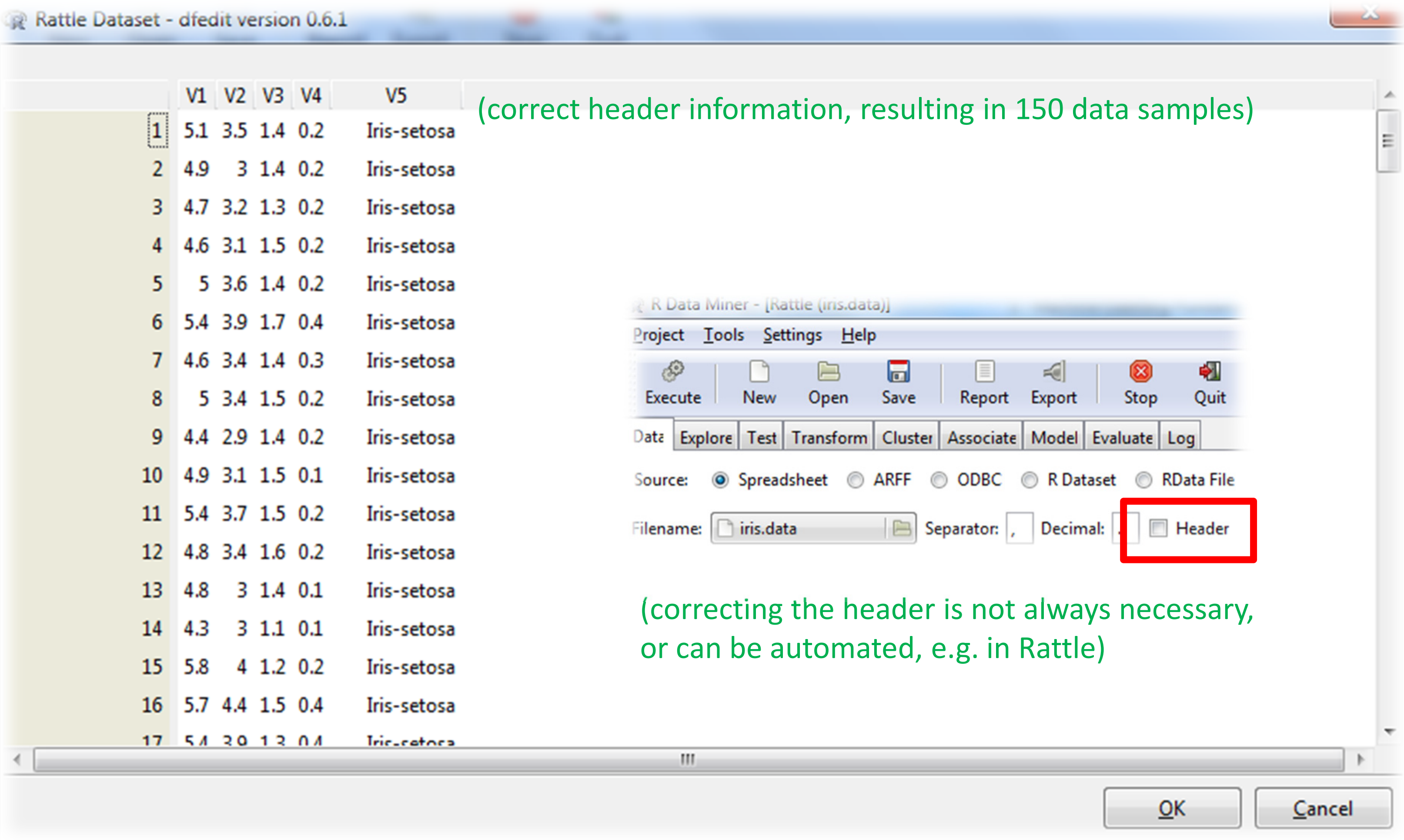
(one class label for each sample in the dataset)

- class: Iris Setosa, or Iris Versicolour, or Iris Virginica

[7] Rattle Library for R

Preparing the Data – Corrected Header

(3) Data Preparation Phase



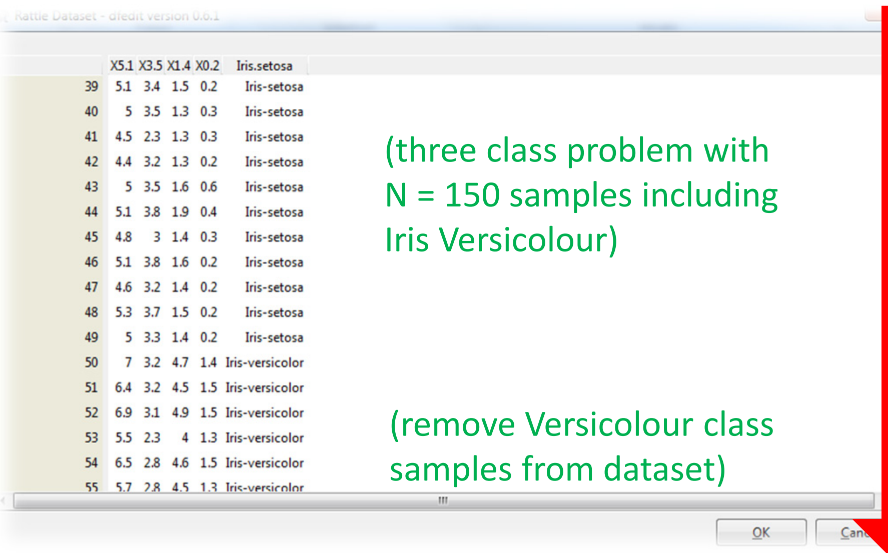
	V1	V2	V3	V4	V5
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
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	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa

(correct header information, resulting in 150 data samples)

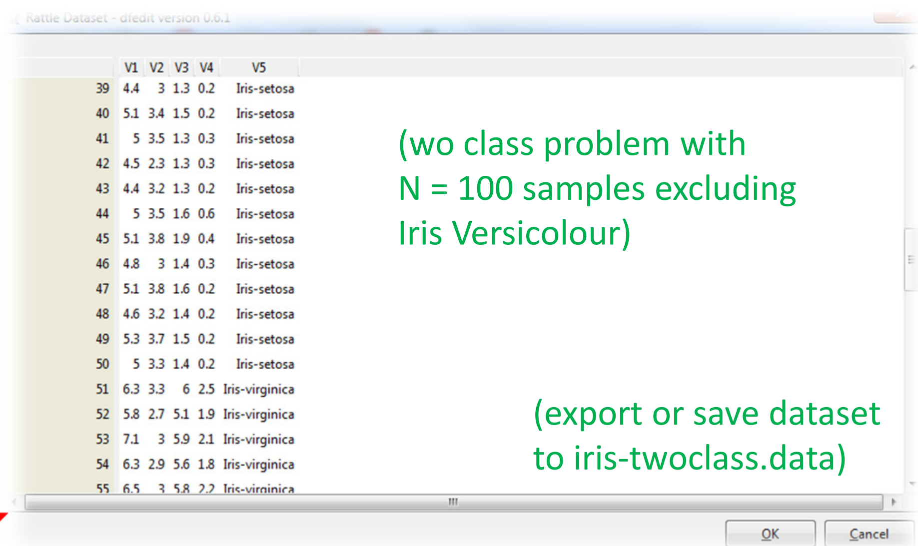
(correcting the header is not always necessary, or can be automated, e.g. in Rattle)

Preparing the Data – Remove Third Class Samples

- Data preparation means to prepare our data for our problem
  - In practice the whole dataset is rarely needed to solve one problem
  - E.g. apply several sampling strategies (but be aware of class balance)
- Recall: Our learning problem
  - Determine whether a new Iris flower sample is a “Setosa” or “Virginica”
  - Binary (two class) classification problem : ‘Setosa‘ or ‘Virginica‘



	X5.1	X3.5	X1.4	X0.2	Iris.setosa
39	5.1	3.4	1.5	0.2	Iris-setosa
40	5	3.5	1.3	0.3	Iris-setosa
41	4.5	2.3	1.3	0.3	Iris-setosa
42	4.4	3.2	1.3	0.2	Iris-setosa
43	5	3.5	1.6	0.6	Iris-setosa
44	5.1	3.8	1.9	0.4	Iris-setosa
45	4.8	3	1.4	0.3	Iris-setosa
46	5.1	3.8	1.6	0.2	Iris-setosa
47	4.6	3.2	1.4	0.2	Iris-setosa
48	5.3	3.7	1.5	0.2	Iris-setosa
49	5	3.3	1.4	0.2	Iris-setosa
50	7	3.2	4.7	1.4	Iris-versicolor
51	6.4	3.2	4.5	1.5	Iris-versicolor
52	6.9	3.1	4.9	1.5	Iris-versicolor
53	5.5	2.3	4	1.3	Iris-versicolor
54	6.5	2.8	4.6	1.5	Iris-versicolor
55	5.7	2.8	4.5	1.3	Iris-versicolor



	V1	V2	V3	V4	V5
39	4.4	3	1.3	0.2	Iris-setosa
40	5.1	3.4	1.5	0.2	Iris-setosa
41	5	3.5	1.3	0.3	Iris-setosa
42	4.5	2.3	1.3	0.3	Iris-setosa
43	4.4	3.2	1.3	0.2	Iris-setosa
44	5	3.5	1.6	0.6	Iris-setosa
45	5.1	3.8	1.9	0.4	Iris-setosa
46	4.8	3	1.4	0.3	Iris-setosa
47	5.1	3.8	1.6	0.2	Iris-setosa
48	4.6	3.2	1.4	0.2	Iris-setosa
49	5.3	3.7	1.5	0.2	Iris-setosa
50	5	3.3	1.4	0.2	Iris-setosa
51	6.3	3.3	6	2.5	Iris-virginica
52	5.8	2.7	5.1	1.9	Iris-virginica
53	7.1	3	5.9	2.1	Iris-virginica
54	6.3	2.9	5.6	1.8	Iris-virginica
55	6.5	3	5.8	2.2	Iris-virginica

Preparing the Data – Feature Selection Process

- Data preparation means to prepare our data for our problem
  - In practice the whole dataset is rarely needed to solve one problem
  - E.g. perform feature selection (aka remove not needed attributes)
- Recall: Our believed pattern in the data
  - A ‘pattern with ‘petal length‘ & ‘petal width‘ somehow influence the type

Rattle Dataset - dfedit version 0.6.1

	V1	V2	V3	V4	V5
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	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	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	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	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa

- ~~sepal length in cm~~
- ~~sepal width in cm~~
- petal length in cm
- petal width in cm
- class: Iris Setosa, or ~~Iris Versicolour~~, or Iris Virginica

(N = 100 samples with 4 attributes and 1 class label)

Rattle Dataset - dfedit version 0.6.1

	V3	V4	V5
1	1.4	0.2	Iris-setosa
2	1.4	0.2	Iris-setosa
3	1.3	0.2	Iris-setosa
4	1.5	0.2	Iris-setosa
5	1.4	0.2	Iris-setosa
6	1.7	0.4	Iris-setosa
7	1.4	0.3	Iris-setosa
8	1.5	0.2	Iris-setosa
9	1.4	0.2	Iris-setosa
10	1.5	0.1	Iris-setosa
11	1.5	0.2	Iris-setosa
12	1.6	0.2	Iris-setosa
13	1.4	0.1	Iris-setosa
14	1.1	0.1	Iris-setosa
15	1.2	0.2	Iris-setosa
16	1.5	0.4	Iris-setosa

OK Cancel

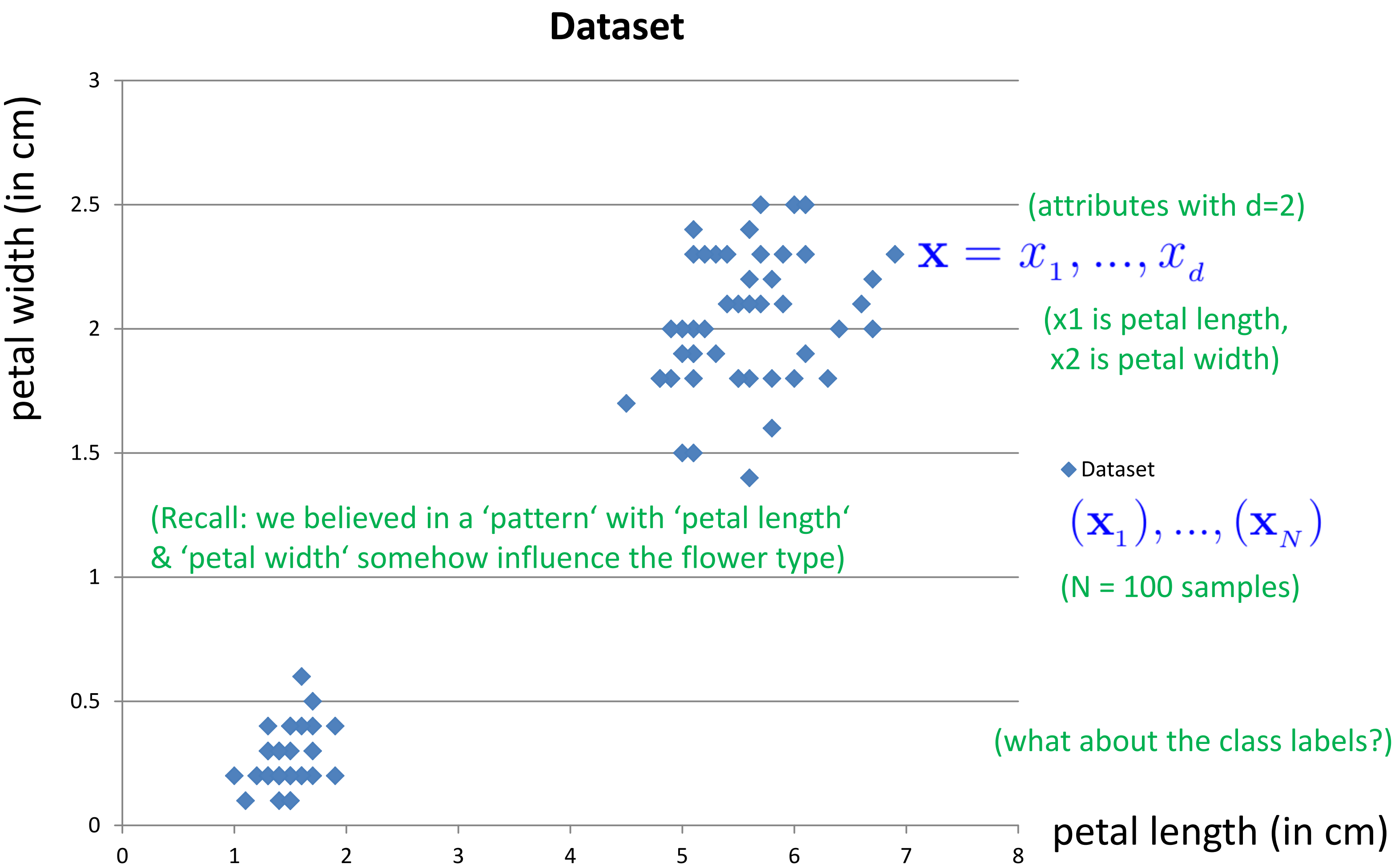
- petal length in cm
- petal width in cm
- class: Iris Setosa, or Iris Versicolour, or Iris Virginica

(export or save dataset to iris-twoclass-twoattr.data)

(N = 100 samples with 2 attributes and 1 class label)

Exercises

Check Preparation Phase: Plotting the Data



(attributes with d=2)

$\mathbf{x} = x_1, \ldots, x_d$

(x1 is petal length, x2 is petal width)

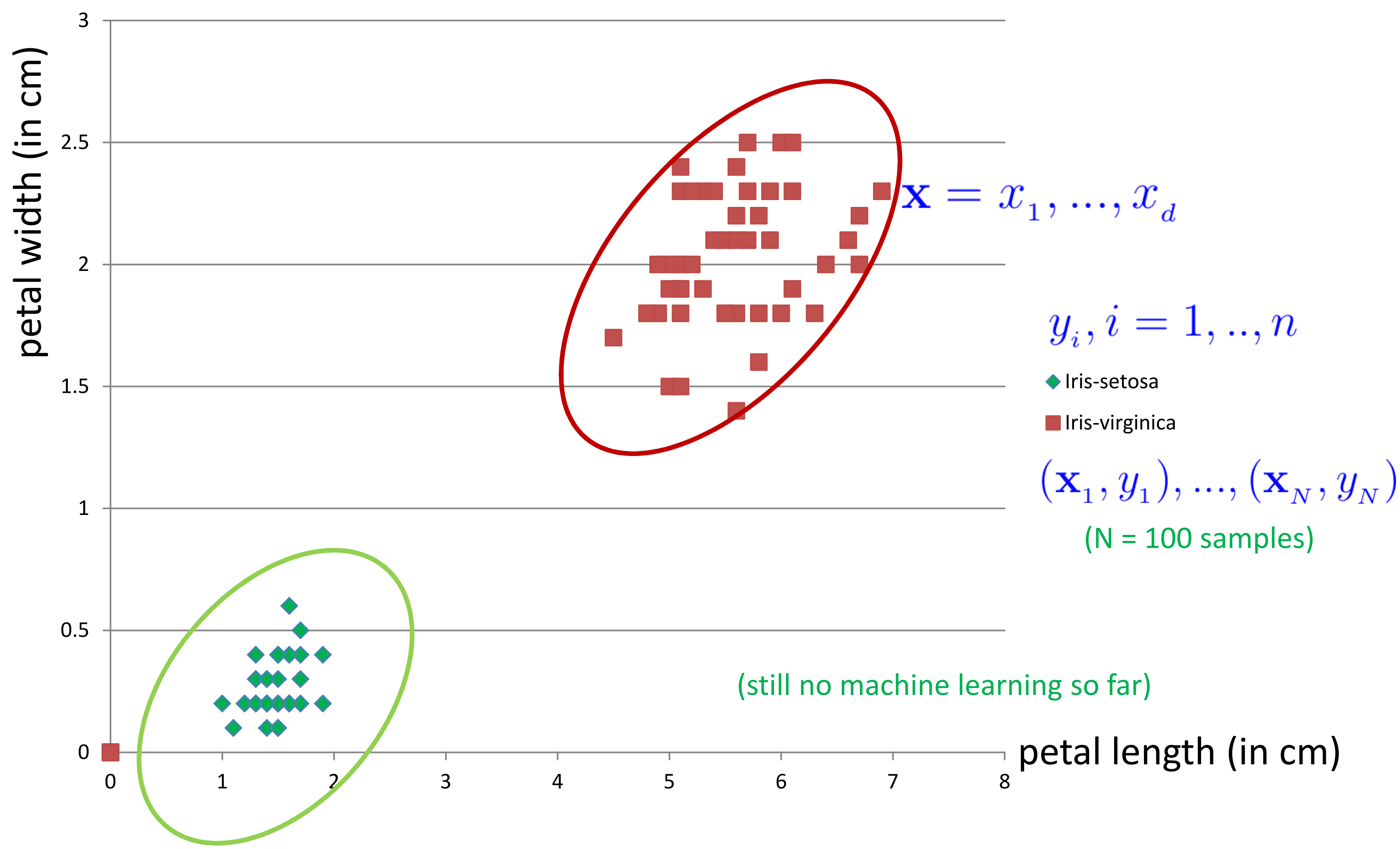
(Recall: we believed in a ‘pattern‘ with ‘petal length‘ & ‘petal width‘ somehow influence the flower type)

$(\mathbf{x}_1), \ldots, (\mathbf{x}_N)$

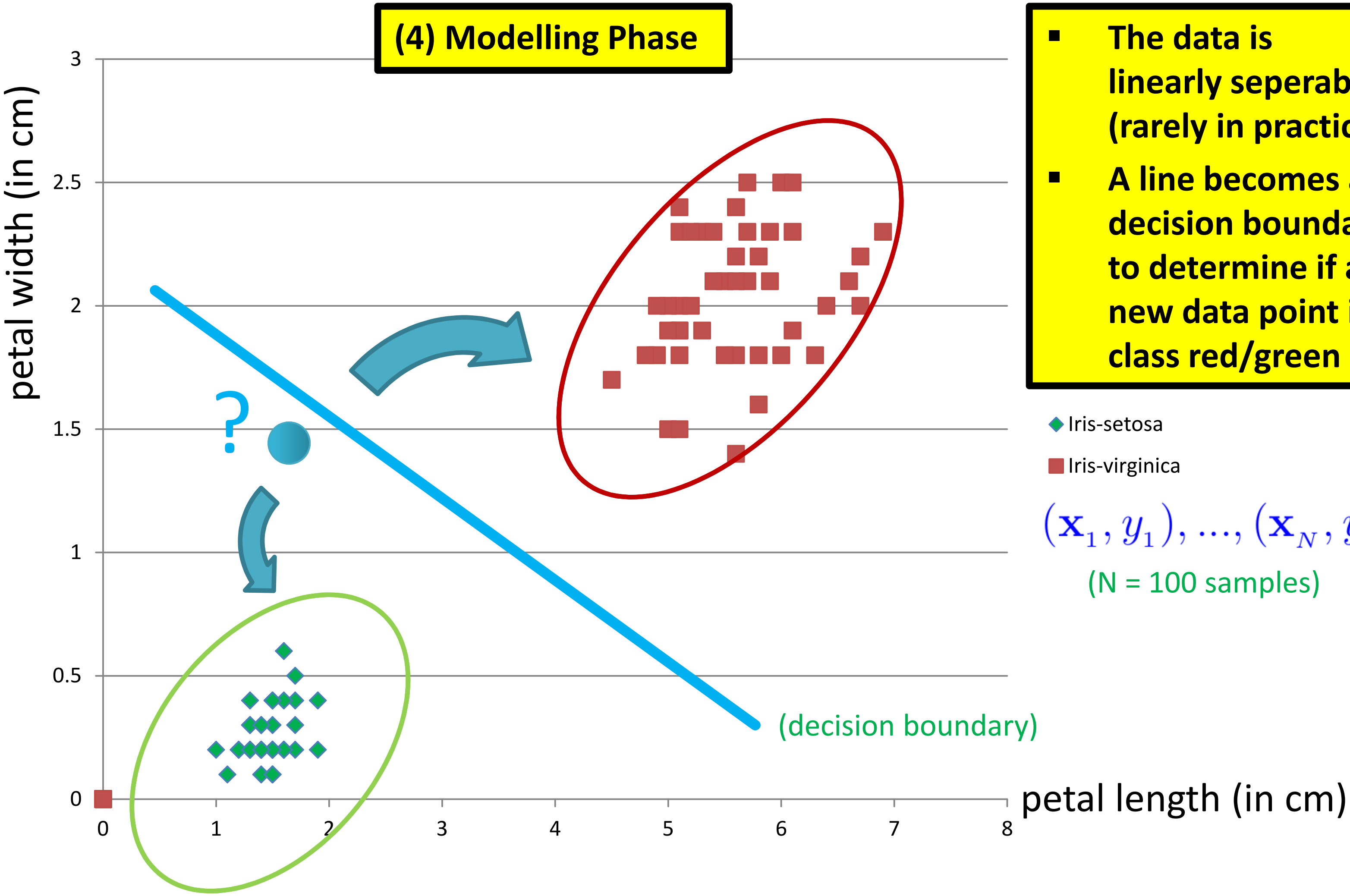
(N = 100 samples)

(what about the class labels?)

Check Preparation Phase: Class Labels



Linearly Seperable Data & Linear Decision Boundary



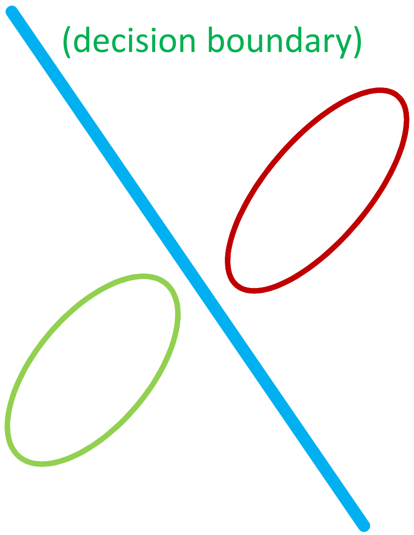
- The data is linearly seperable (rarely in practice)
- A line becomes a decision boundary to determine if a new data point is class red/green

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(N = 100 samples)

Separating Line & Mathematical Notation

- Data exploration results
  - A line can be crafted between the classes since linearly seperable data
  - All the data points representing Iris-setosa will be below the line
  - All the data points representing Iris-virginica will be above the line
- More formal mathematical notation
  - Input: $\mathbf{x} = x_1, \ldots, x_d$ (attributes of flowers)
  - Output: class +1 (Iris-virginica) or class -1 (Iris-setosa)



Iris-virginica if $\sum_{i=1}^{d} w_i x_i > threshold$

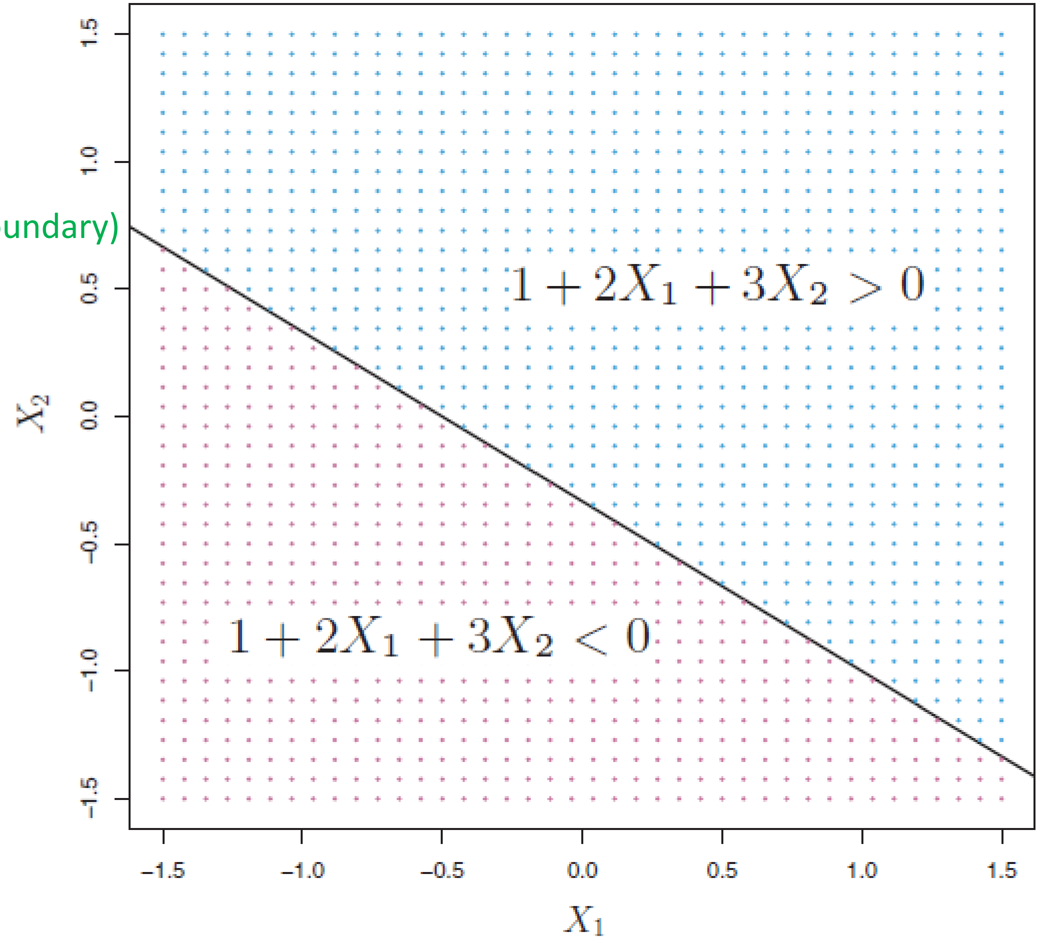
Iris-setosa if $\sum_{i=1}^{d} w_i x_i < threshold$

(w_i and threshold are still unknown to us)

$$sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right)$$ (compact notation)

Separating Line & ‘Decision Space‘ Example

(decision boundary)



(equation of a line)

$$1 + 2X_1 + 3X_2 = 0$$

(all points X_i on this line have to satisfy this equation)

modified from [13] An Introduction to Statistical Learning

A Simple Linear Learning Model – The Perceptron

- Human analogy in learning *[8] F. Rosenblatt, 1957*
  - Human brain consists of nerve cells called neurons
  - Human brain learns by changing the strength of neuron connections (w_i) upon repeated stimulation by the same impulse (aka a 'training phase')
  - Training a perceptron model means adapting the weights w_i
  - Done until they fit input-output relationships of the given 'training data'

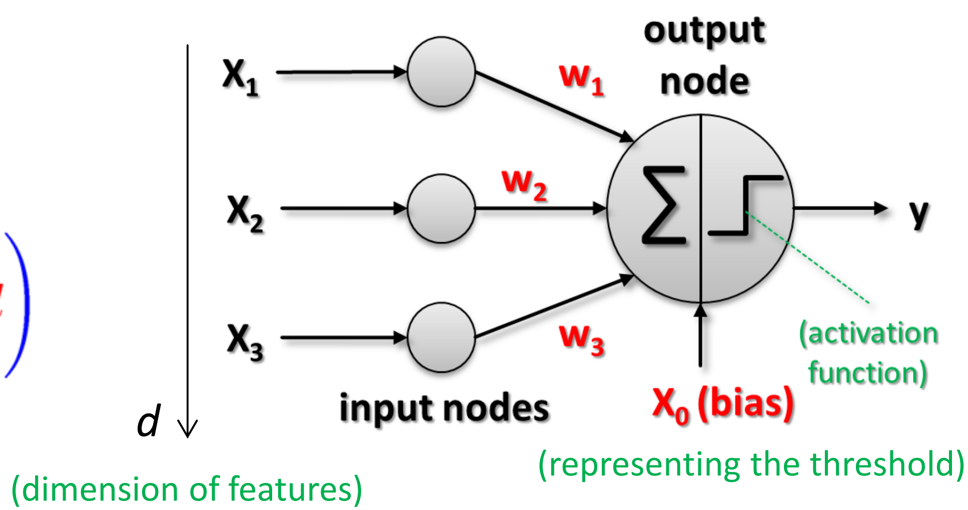
$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$

(training data)

$$sign\left(\underbrace{\left(\sum_{i=1}^{d} w_i x_i\right) - threshold}_{\text{(the signal)}}\right)$$

sign: (activation function, +1 or -1)

threshold: (modelled as bias term)



Perceptron – Example of a Boolean Function

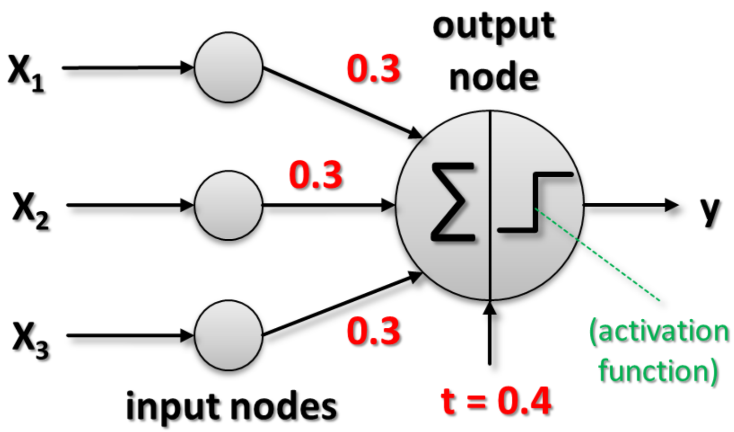
	x1	x2	x3	y
1	1	0	0	-1
2	1	0	1	1
3	1	1	0	1
4	1	1	1	1
5	0	0	1	-1
6	0	1	0	-1
7	0	1	1	1
8	0	0	0	-1

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(training data)

(training phase)

$$\boxed{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right)$$



(trained perceptron model)

- Output node interpretation
  - More than just the weighted sum of the inputs – threshold (aka bias)
  - Activation function sign (weighted sum): takes sign of the resulting sum

$$y = 1, \textbf{if } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0$$

(e.g. consider sample #3, sum is positive (0.2) → +1)

$$y = -1, \textbf{if } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0$$

(e.g. consider sample #6, sum is negative (-0.1) → -1)

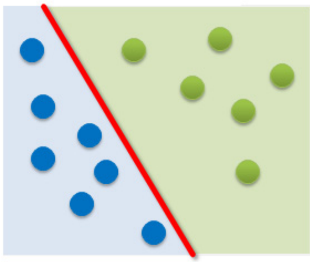
Summary Perceptron & Hypothesis Set h(x)

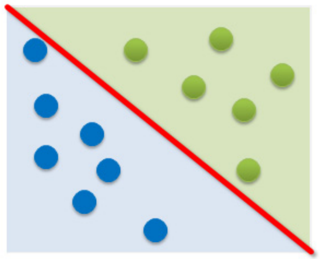
- When: Solving a linear classification problem *[8] F. Rosenblatt, 1957*
  - Goal: learn a simple value (+1/-1) above/below a certain threshold
  - Class label renamed: Iris-setosa = -1 and Iris-virginica = +1
- Input: $\mathbf{x} = x_1, \ldots, x_d$ (attributes in one dataset)
- Linear formula (take attributes and give them different weights – think of 'impact of the attribute')
  - All learned formulas are different hypothesis for the given problem

$$h(\mathbf{x}) = \boxed{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right); h \in \mathcal{H}$$

(parameters that define one hypothesis vs. another)

(each green space and blue space are regions of the same class label determined by sign function)





(red parameters correspond to the redline in graphics)

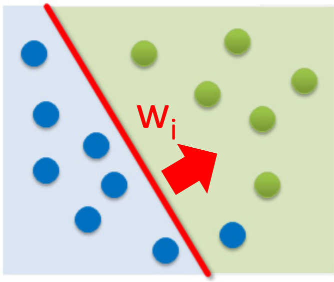
(but question remains: how do we actually learn w_i and threshold?)

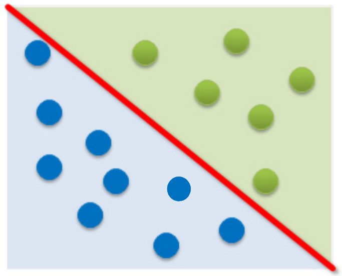
Perceptron Learning Algorithm – Understanding Vector W

- When: If we believe there is a linear pattern to be detected
  - Assumption: Linearly seperable data (lets the algorithm converge)
  - Decision boundary: perpendicular vector w_i fixes orientation of the line

$$\mathbf{w}^T\mathbf{x} = 0$$
$$\mathbf{w} \cdot \mathbf{x} = 0$$

(points on the decision boundary satisfy this equation)





$$h(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x})$$

(vector notation, using T = transpose)

$$\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_d)$$

$$\mathbf{w}_i^T = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \cdots \\ w_{id} \end{bmatrix}$$

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_d)$$

  - Possible via simplifications since we also need to learn the threshold:

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + w_0\right); w_0 = -threshold$$

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=0}^{d} w_i x_i\right)\right); x_0 = 1$$

$$h(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x})$$

(equivalent dotproduct notation)

[9] Rosenblatt, 1958

(all notations are equivalent and result is a scalar from which we derive the sign)

Understanding the Dot Product – Example & Interpretation

- ‘Dot product‘
  - Given two vectors
  - Multiplying corresponding components of the vector
  - Then adding the resulting products
  - Simple example: $(2,3) \cdot (4,1) = 2 * 4 + 3 * 1 = \boxed{11}$ (a scalar!)
  - Interesting: Dot product of two vectors is a scalar

$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i v_i$$

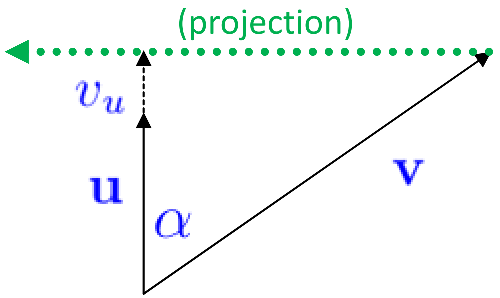
$$h(\mathbf{x}) = sign\left(\left(\sum_{i=0}^{d} w_i x_i\right)\right);\, x_0 = 1$$

$$h(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x})$$

(our example)

- ‘Projection capabilities of Dot product‘ (simplified)
  - Orthogonal projection of vector $\mathbf{v}$ in the direction of vector $\mathbf{u}$

$$\mathbf{u} \cdot \mathbf{v} = (\|v\| cos(\alpha))\|u\| = v_u \|u\|$$



- Normalize using length of vector

$$\frac{\mathbf{u}}{\|\mathbf{u}\|} \quad \|\mathbf{u}\| = length(\mathbf{u}) = L_2 norm = \sqrt{\mathbf{u} \cdot \mathbf{u}}$$

Perceptron Learning Algorithm – Learning Step

- Iterative Method using (labelled) training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(one point at a time is picked)

1. Pick one misclassified training point where:

$$sign(\mathbf{w}^T \mathbf{x}_n) \neq y_n$$

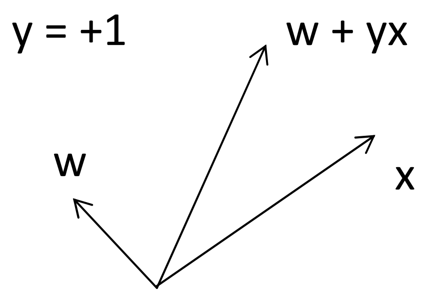
2. Update the weight vector:

$$\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n$$

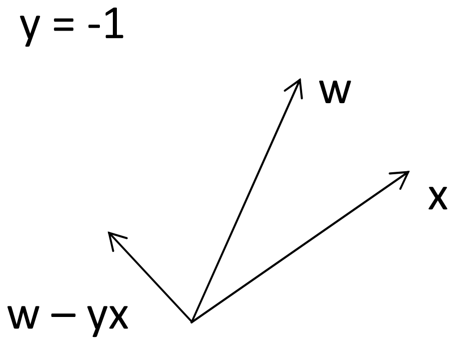
(y_n is either +1 or -1)

(a) adding a vector or
(b) subtracting a vector

(a)



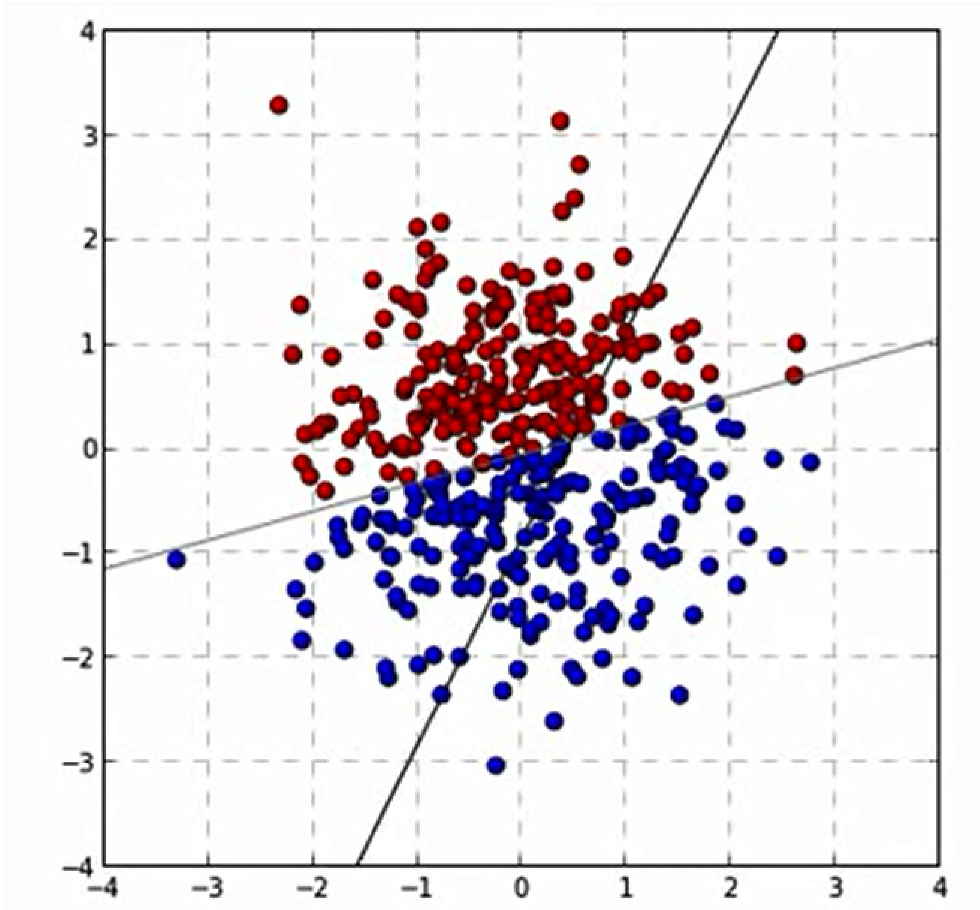
(b)



- Terminates when there are no misclassified points

(converges only with linearly seperable data)

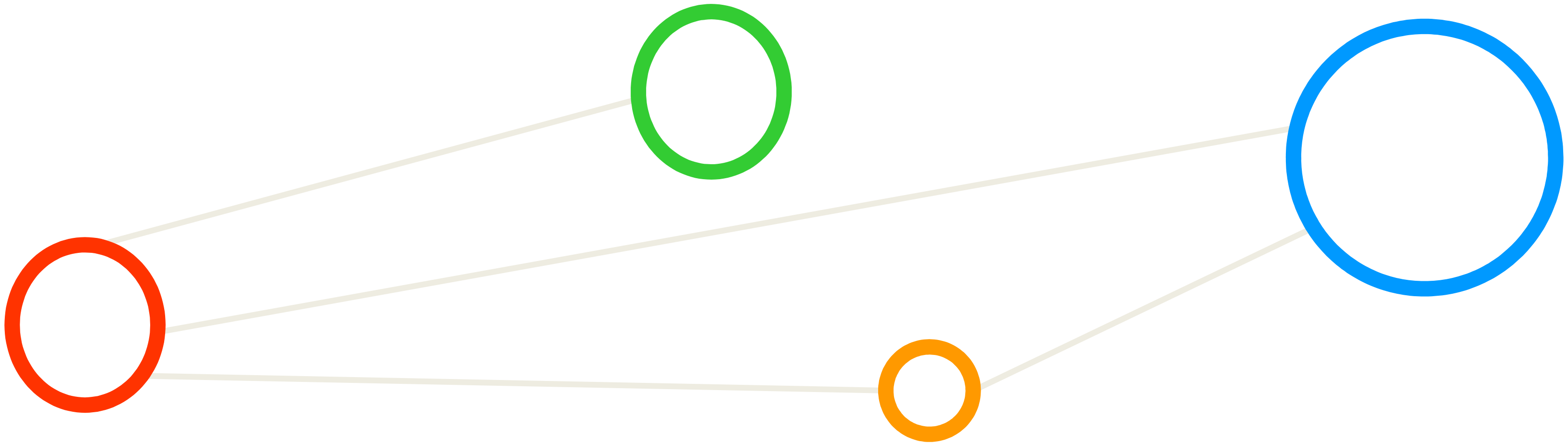
[Video] Perceptron Learning Algorithm



[10] PLA Video

Exercises

Learning from Data

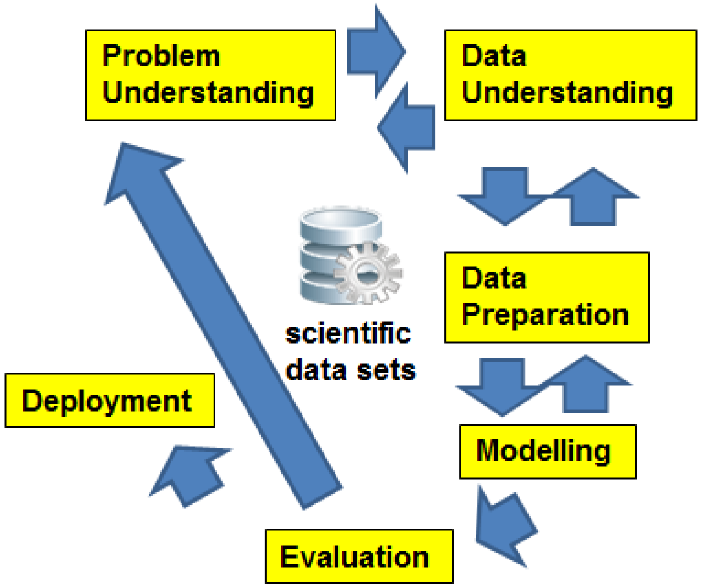


Systematic Process to Support Learning From Data

- Systematic data analysis guided by a ‘standard process‘
  - Cross-Industry Standard Process for Data Mining (CRISP-DM)

> - **A data mining project is guided by these six phases:**
>   **(1) Problem Understanding;**
>   **(2) Data Understanding;**
>   **(3) Data Preparation;**
>   **(4) Modeling;**
>   **(5) Evaluation;**
>   **(6) Deployment**

(learning takes place)



[11] C. Shearer, CRISP-DM model, Journal Data Warehousing, 5:13

- Lessons Learned from Practice
  - Go back and forth between the different six phases

> ➢ **A more detailed description of all six CRISP-DM phases is in the appendix of the slideset**

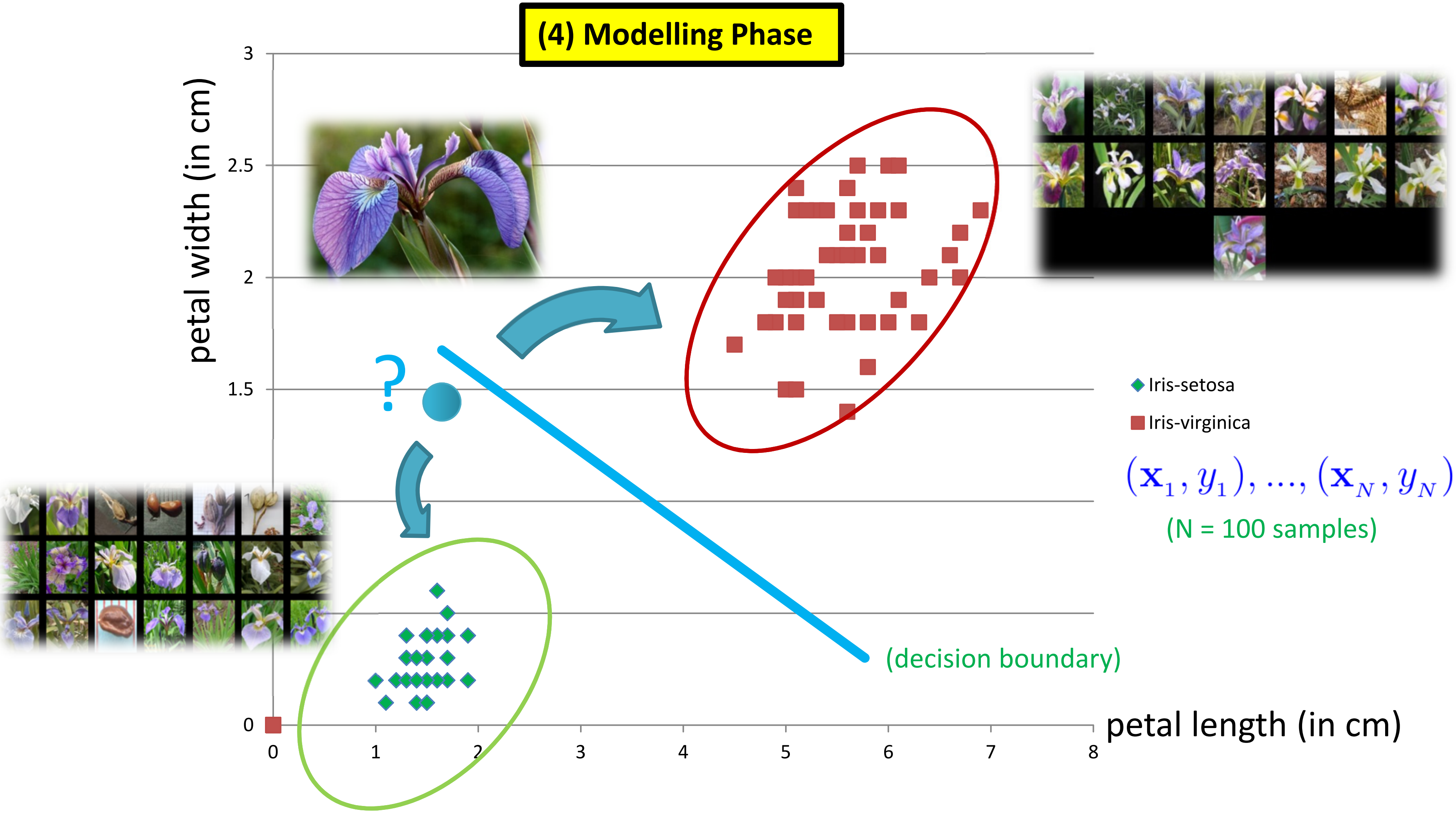
Machine Learning & Data Mining Tasks in Applications

- **Machine learning tasks can be divided into two major categories: Predictive and Descriptive Tasks**

[1] Introduction to Data Mining

- Predictive Tasks
  - Predicts the value of an attribute based on values of other attributes
  - Target/dependent variable: attribute to be predicted
  - Explanatory/independent variables: attributed used for making predictions
  - E.g. predicting the species of a flower based on characteristics of a flower
- Descriptive Tasks
  - Derive patterns that summarize the underlying relationships in the data
  - Patterns here can refer to correlations, trends, trajectories, anomalies
  - Often exploratory in nature and frequently require postprocessing
  - E.g. credit card fraud detection with unusual transactions for owners

Predicting Task: Obtain Class of a new Flower ‘Data Point‘



$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(N = 100 samples)

[4] Image sources: Species Iris Group of North America Database, www.signa.org

What means Learning?

- **The basic meaning of learning is ‘to use a set of observations to uncover an underlying process‘**
- **The three different learning approaches are supervised, unsupervised, and reinforcement learning**

- Supervised Learning
  - Majority of methods follow this approach in this course
  - Example: credit card approval based on previous customer applications
- Unsupervised Learning
  - Often applied before other learning → higher level data representation
  - Example: Coin recognition in vending machine based on weight and size
- Reinforcement Learning
  - Typical ‘human way‘ of learning
  - Example: Toddler tries to touch a hot cup of tea (again and again)

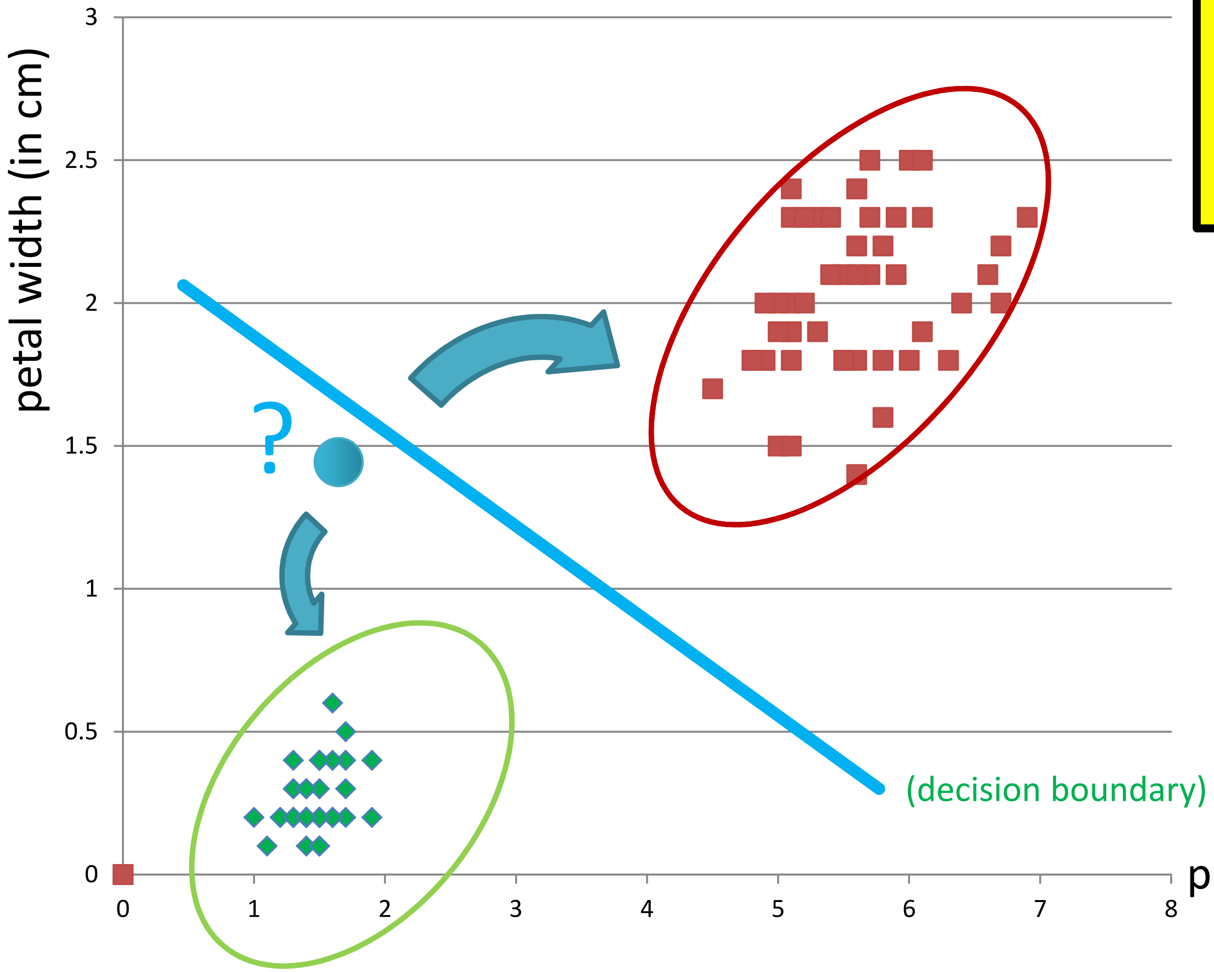
Learning Approaches – Supervised Learning

- Each observation of the predictor measurement(s) has an associated response measurement:
  - Input $\mathbf{x} = x_1, ..., x_d$
  - Output $y_i, i = 1, .., n$
  - Data $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$
- Goal: Fit a model that relates the response to the predictors
  - Prediction: Aims of accurately predicting the response for future observations
  - Inference: Aims to better understanding the relationship between the response and the predictors

- **Supervised learning approaches fits a model that related the response to the predictors**
- **Supervised learning approaches are used in classification algorithms such as SVMs**
- **Supervised learning works with data = [input, correct output]**

[13] An Introduction to Statistical Learning

Learning Approaches – Supervised Learning Example



- **The labels guide our learning process like a 'supervisor' is helping us**

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(N = 100 samples)

➢ **Lecture 2 provides details on the supervised learning approach using classification**

Learning Approaches – Unsupervised Learning

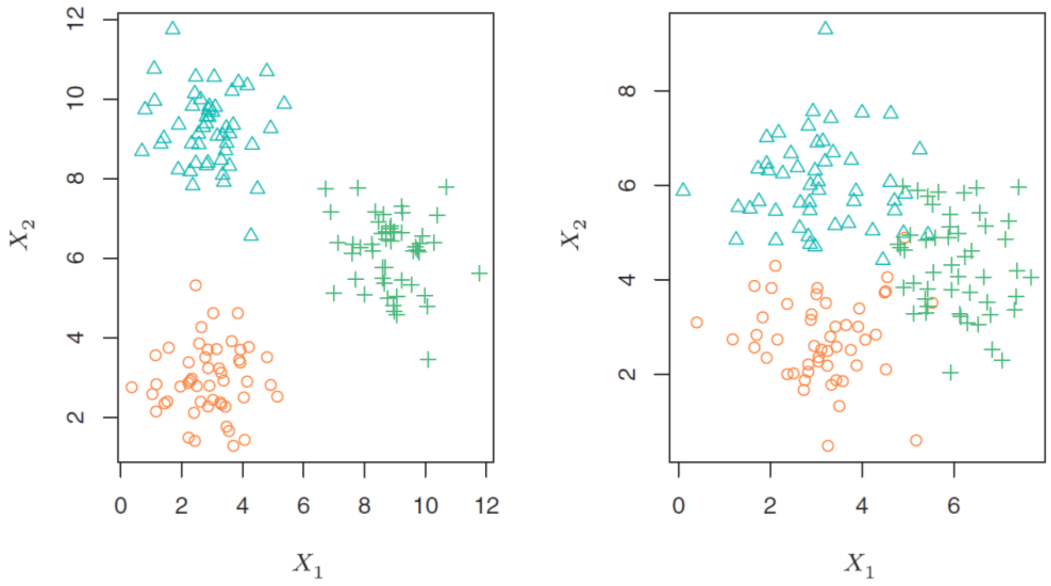
- Each observation of the predictor measurement(s) has no associated response measurement:
  - Input $\mathbf{x} = x_1, \ldots, x_d$
  - No output
  - Data $(\mathbf{x}_1), \ldots, (\mathbf{x}_N)$
- Goal: Seek to understand relationships between the observations
  - Clustering analysis: check whether the observations fall into distinct groups
- Challenges
  - No response/output that could supervise our data analysis
  - Clustering groups that overlap might be hardly recognized as distinct group

> - **Unsupervised learning approaches seek to understand relationships between the observations**
> - **Unsupervised learning approaches are used in clustering algorithms such as k-means, etc.**
> - **Unupervised learning works with data = [input, ---]**

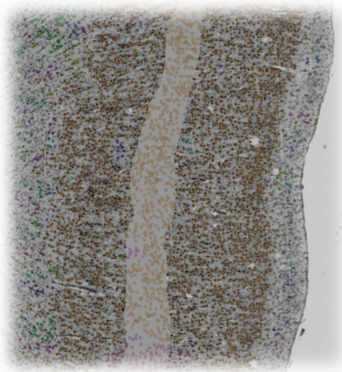
[13] An Introduction to Statistical Learning

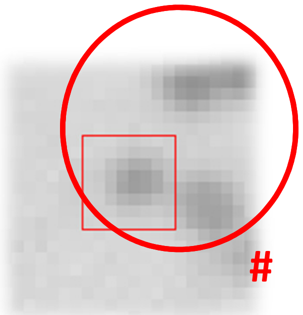
Learning Approaches – Unsupervised Learning Example

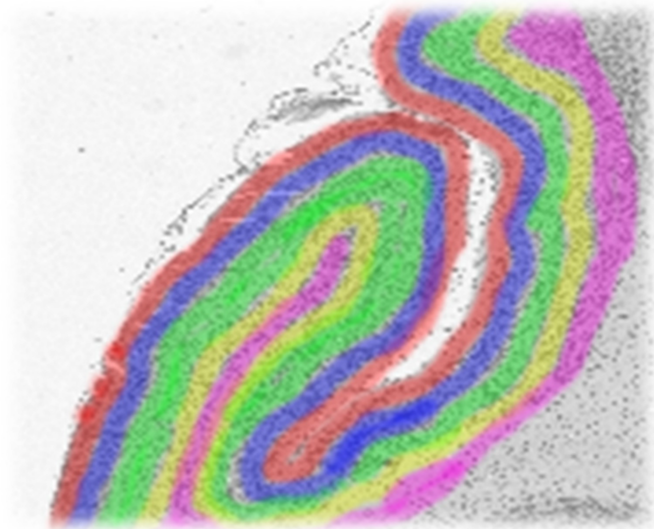
- Practice: The number of clusters can be ambiguities



[13] An Introduction to Statistical Learning







> **Lecture 2 offers more details about unsupervised learning using clustering algorithms in practice**

Learning Approaches – Reinforcement Learning

- Each observation of the predictor measurement(s) has some associated response measurement:
  - Input $\mathbf{x} = x_1, \ldots, x_d$
  - Some output & grade of the output
  - Data $(\mathbf{x}_1), \ldots, (\mathbf{x}_N)$
- Goal: Learn through iterations
  - Guided by output grade: check learning and compare with grade
- Challenge:
  - Iterations may require lots of CPU time (e.g. backgammon playing rounds)
- (Rarely tackled in this course, just for the sake of completion)

- **Reinforcement learning approaches learn through iterations using the grading output as guide**
- **Reinforcement learning approaches are used in playing game algorithms (e.g backgammon)**
- **Unupervised learning works with data = [input, some output, grade for this output]**

[13] An Introduction to Statistical Learning

Summary Terminologies & Different Dataset Elements

- Target Function $f : X \to Y$
  - Ideal function that 'explains' the data we want to learn
- Labelled Dataset (samples)
  - 'in-sample' data given to us: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
- Learning vs. Memorizing
  - The goal is to create a system that works well 'out of sample'
  - In other words we want to classify 'future data' (ouf of sample) correct
- Dataset Part One: Training set

(4) Modelling Phase

  - Used for training a machine learning algorithms
  - Result after using a training set: a trained system
- Dataset Part Two: Test set

(5) Evaluation Phase

  - Used for testing whether the trained system might work well
  - Result after using a test set: accuracy of the trained model

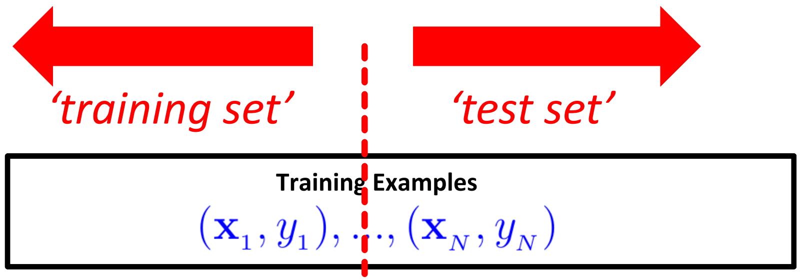
Model Evaluation – Training and Testing Phases

- Different Phases in Learning
  - Training phase is a hypothesis search
  - Testing phase checks if we are on right track (once the hypothesis clear)
- Work on 'training examples'
  - Create two disjoint datasets
  - One used for training only (aka training set)
  - Another used for testing only (aka test set)
  - Exact seperation is rule of thumb per use case (e.g. 10 % training, 90% test)
  - Practice: If you get a dataset take immediately test data away ('throw it into the corner and forget about it during modelling')
  - Reasoning: Once we learned from training data it has an 'optimistic bias'

(4) Modelling Phase

(5) Evaluation Phase

(e.g. student exam training on examples to get E_{in} ,down', then test via exam)



(historical records, groundtruth data, examples)

Model Evaluation – Testing Phase & Confusion Matrix

(5) Evaluation Phase

- Model is fixed
  - Model is just used with the testset
  - Parameter w_i are set and we have a linear decision function
- Evaluation of model performance
  - Counts of test records that are incorrectly predicted $sign(\mathbf{w}^T \mathbf{x}_n) \neq y_n$
  - Counts of test records that are correctly predicted $sign(\mathbf{w}^T \mathbf{x}_n) = y_n$
  - E.g. create confusion matrix for a two class problem

Counting per sample		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

(serves as a basis for further performance metrics usually used)

Model Evaluation – Testing Phase & Performance Metrics

Counting per sample		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

(5) Evaluation Phase

(100% accuracy in learning often points to problems using machine learning methos in practice)

- Accuracy (usually in %)

$$Accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions}$$

- Error rate

$$Error\ rate = \frac{number\ of\ wrong\ predictions}{total\ number\ of\ predictions}$$

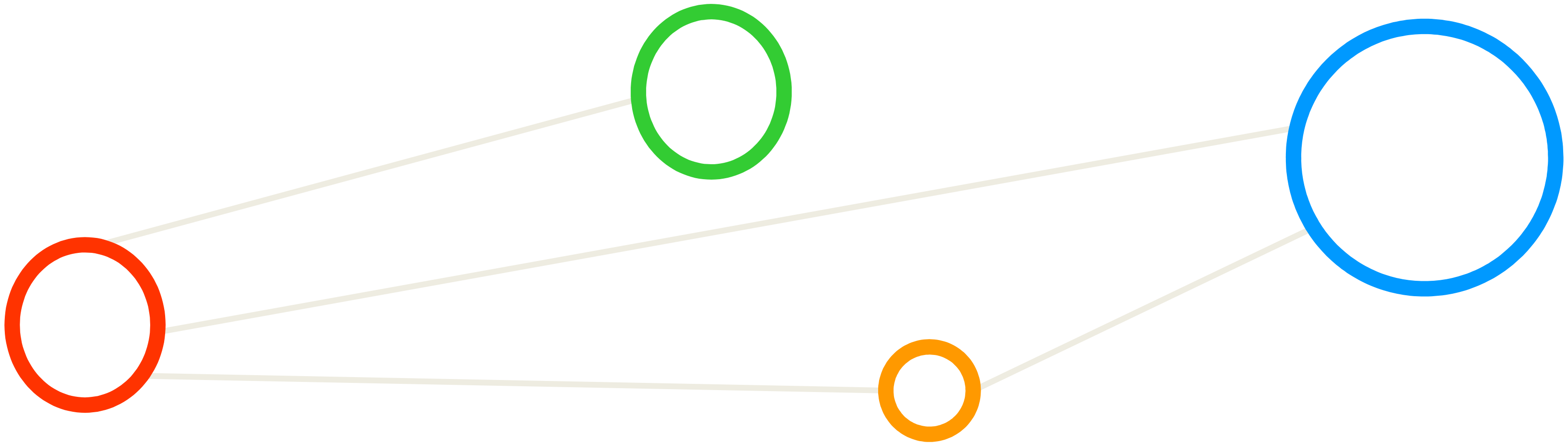
- If model evaluation is satisfactory: **(6) Deployment Phase**

Exercises

[Video] European Plate Observing System

[14] EPOS Data Community Services, YouTube

Lecture Bibliography



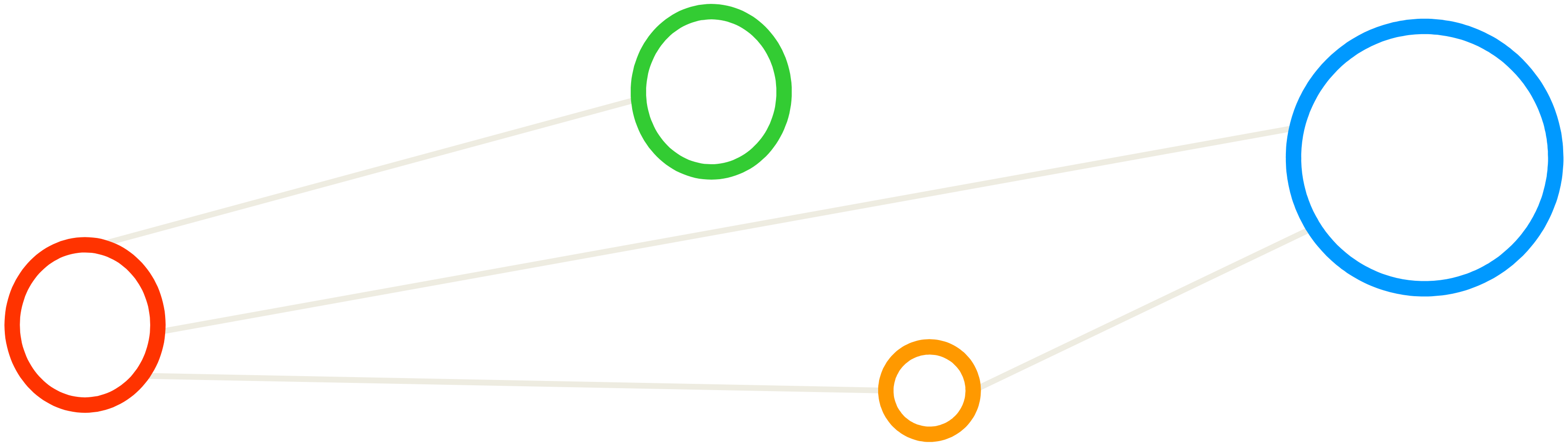
Lecture Bibliography (1)

- [1] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison Wesley, ISBN 0321321367, English, ~769 pages, 2005
- [2] PANGAEA Data Collection, Data Publisher for Earth & Environmental Science, Online: http://www.pangaea.de/
- [3] UCI Machine Learning Repository, Online: http://archive.ics.uci.edu/ml/datasets.html
- [4] Species Iris Group of North America Database, Online: http://www.signa.org
- [5] UCI Machine Learning Repository Iris Dataset, Online: https://archive.ics.uci.edu/ml/datasets/Iris
- [6] Wikipedia ‘Sepal‘, Online: https://en.wikipedia.org/wiki/Sepal
- [7] Rattle Library for R, Online: http://rattle.togaware.com/
- [8] F. Rosenblatt, ‘The Perceptron--a perceiving and recognizing automaton’, Report 85-460-1, Cornell Aeronautical Laboratory, 1957
- [9] Rosenblatt,’The Perceptron: A probabilistic model for information storage and orgainzation in the brain’, Psychological Review 65(6), pp. 386-408, 1958
- [10] PLA Algorithm, YouTube Video, Online:
- [11] C. Shearer, CRISP-DM model, Journal Data Warehousing, 5:13
- [12] Pete Chapman, ‘CRISP-DM User Guide’, 1999, Online: http://lyle.smu.edu/~mhd/8331f03/crisp.pdf

Lecture Bibliography (2)

- [13] An Introduction to Statistical Learning with Applications in R, Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [14] EPOS - European Plate Observing System -- Community Services, YouTube Video, Online: http://www.youtube.com/watch?v=zh-paxiQhKI

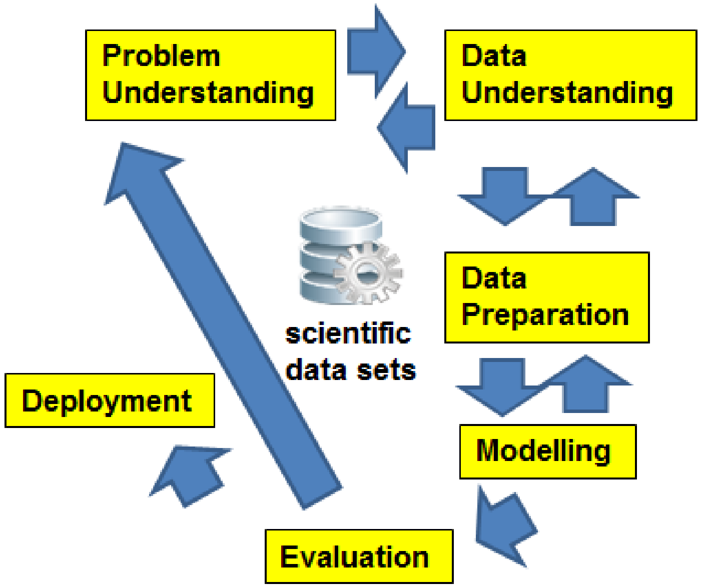
Appendix



Summary: Systematic Process

- Systematic data analysis guided by a ‘standard process‘
  - Cross-Industry Standard Process for Data Mining (CRISP-DM)

- **A data mining project is guided by these six phases: (1) Problem Understanding; (2) Data Understanding; (3) Data Preparation; (4) Modeling; (5) Evaluation; (6) Deployment**



[11] C. Shearer, CRISP-DM model, Journal Data Warehousing, 5:13

- Lessons Learned from Practice
  - Go back and forth between the different six phases

1 – Problem (Business) Understanding

- **The Business Understanding phase consists of four distinct tasks: (A) Determine Business Objectives; (B) Situation Assessment; (C) Determine Data Mining Goal; (D) Produce Project Plan**

[12] CRISP-DM User Guide

- Task A – Determine Business Objectives
  - Background, Business Objectives, Business Success Criteria
- Task B – Situation Assessment
  - Inventory of Resources, Requirements, Assumptions, and Contraints
  - Risks and Contingencies, Terminology, Costs & Benefits
- Task C – Determine Data Mining Goal
  - Data Mining Goals and Success Criteria
- Task D – Produce Project Plan
  - Project Plan
  - Initial Assessment of Tools & Techniques

2 – Data Understanding

- **The Data Understanding phase consists of four distinct tasks:**
  (A) Collect Initial Data; (B) Describe Data; (C) Explore Data; (D) Verify Data Quality

[12] CRISP-DM User Guide

- Task A – Collect Initial Data
  - Initial Data Collection Report
- Task B – Describe Data
  - Data Description Report
- Task C – Explore Data
  - Data Exploration Report
- Task D – Verify Data Quality
  - Data Quality Report

3 – Data Preparation

- **The Data Preparation phase consists of six distinct tasks: (A) Data Set; (B) Select Data; (C) Clean Data; (D) Construct Data; (E) Integrate Data; (F) Format Data**

[12] CRISP-DM User Guide

- Task A – Data Set
  - Data set description
- Task B – Select Data
  - Rationale for inclusion / exclusion
- Task C – Clean Data
  - Data cleaning report
- Task D – Construct Data
  - Derived attributes, generated records
- Task E – Integrate Data
  - Merged data
- Task F – Format Data
  - Reformatted data

4 – Modeling

- **The Data Preparation phase consists of four distinct tasks: (A) Select Modeling Technique; (B) Generate Test Design; (C) Build Model; (D) Assess Model;**

[12] CRISP-DM User Guide

- Task A – Select Modeling Technique
  - Modeling assumption, modeling technique
- Task B – Generate Test Design
  - Test design
- Task C – Build Model
  - Parameter settings, models, model description
- Task D – Assess Model
  - Model assessment, revised parameter settings

5 – Evaluation

- **The Data Preparation phase consists of three distinct tasks: (A) Evaluate Results; (B) Review Process; (C) Determine Next Steps**

[12] CRISP-DM User Guide

- Task A – Evaluate Results
  - Assessment of data mining results w.r.t. business success criteria
  - List approved models
- Task B – Review Process
  - Review of Process
- Task C – Determine Next Steps
  - List of possible actions, decision

6 – Deployment

- **The Data Preparation phase consists of three distinct tasks: (A) Plan Deployment; (B) Plan Monitoring and Maintenance; (C) Produce Final Report; (D) Review Project**

[12] CRISP-DM User Guide

- Task A – Plan Deployment
  - Establish a deployment plan
- Task B – Plan Monitoring and Maintenance
  - Create a monitoring and maintenance plan
- Task C – Product Final Report
  - Create final report and provide final presentation
- Task D – Review Project
  - Document experience, provide documentation


Data
Science
Big
Research
science
analysis
methods
big
simulations
processing
technologies
computing
computational
TB
European
Supercomputing
Scientific
HPC
Juelich
datasets
performance
Climate
modelling
structures
research
databases
Computational
Centre
climate
Computing
nodes
centers
storage
ENES
EUDAT
community
Service
data-intensive
management