Data Mining 2024
Introduction

Ad Feelders

Universiteit Utrecht

The Course

- Literature: Lecture Notes, Book Chapters, Articles, Slides (the slides appear in the schedule on the course web site).
- Course Form:
  - Lectures (Tuesday, Thursday)
  - Lab session (Thursday after the lecture).
- Grading: two practical assignments (50%), a digital exam in Remindo (50%), and 4 homework exercise sets (5% bonus).
- Web Site: https://ics-websites.science.uu.nl/docs/vakken/mdm/
- MS Teams: mainly for finding team mates, and questions about practical assignments. Videos of lectures from previous years will be uploaded.

Personnel

Lecturer: Ad Feelders

Teaching Assistants:

- Panagiotis Andrikopoulos
- Ziv Hochman
- Stylianos Psara

Practical Assignments

Two practical assignments: one assignment with emphasis on programming and one with emphasis on data analysis.

1. Write your own classification tree and random forest algorithm in Python or R, and apply the algorithm to a bug prediction problem (30%).
2. Text Mining: predict whether hotel reviews are genuine or fake (20%).

Assignments should be completed by teams of 3 students.

We will not teach you how to program in Python or R. If you don't know either of these languages yet, you will have to invest some time.
Of course we will try to help you if you have questions about them.

Homework exercise sets

There are four homework exercise sets:

1. Classification trees, bagging and random forests.
2. Undirected graphical models (Markov random fields).
3. Frequent pattern mining.
4. Directed graphical models (Bayesian networks).

We will use Remindo for handing in the exercise sets.

What is Data Mining?

Selected definitions:

- (Knowledge discovery in databases) is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al.)
- Analysis of *secondary* data (Hand)
- The induction of understandable models and patterns from databases (Siebes)
- The *data-dependent* process of selecting a statistical model (Leamer, 1978 (!))

What is Data Mining?

Data Mining as a subdiscipline of computer science:

is concerned with the development and analysis of algorithms for the (efficient) extraction of patterns and models from (large, heterogeneous, ...) data bases.

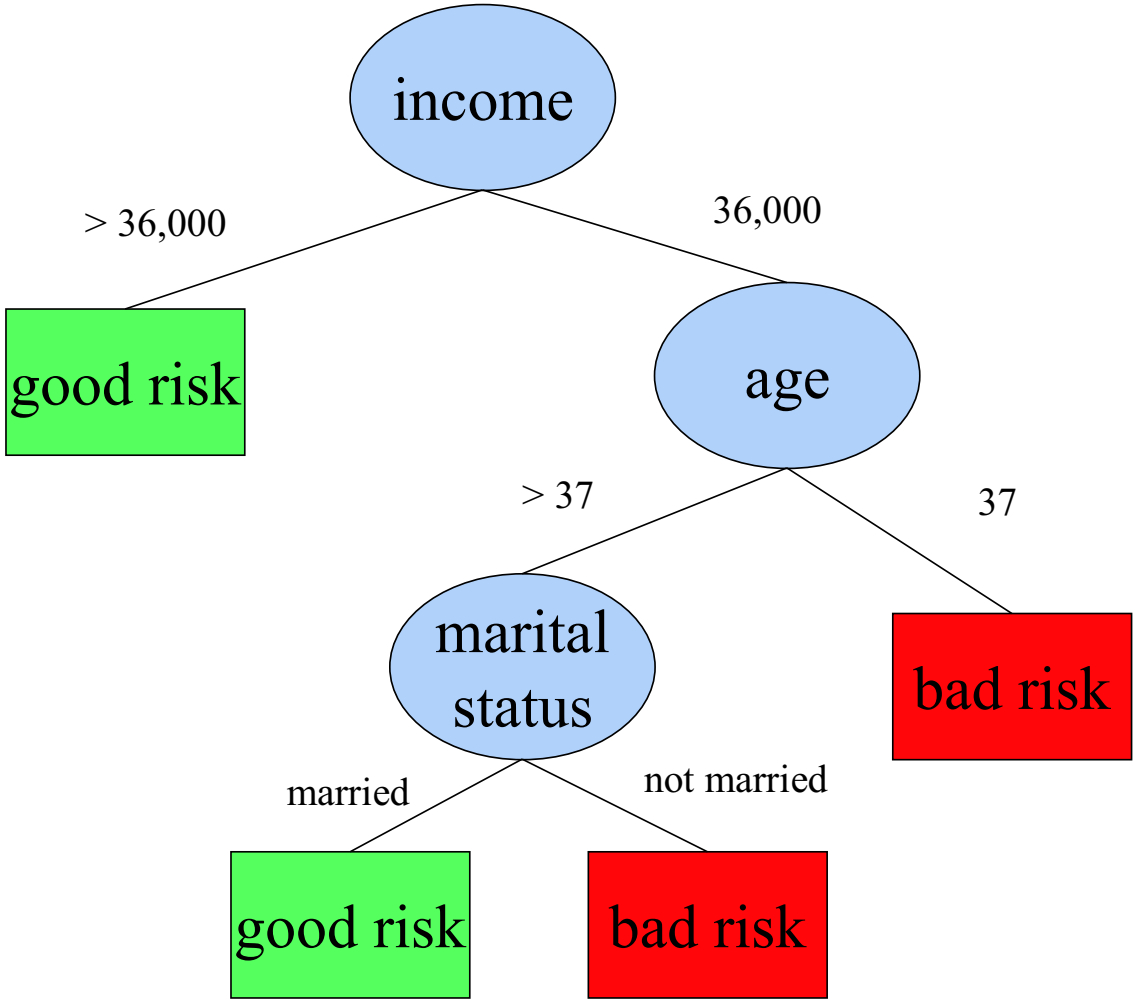
Models

A model is an abstraction of a part of reality (the application domain).

In our case, models describe relationships among:

- attributes (variables, features),
- tuples (records, cases),
- or both.

Example Model: Classification Tree



Patterns

Patterns are **local models**, that is, models that describe only part of the database.
For example, association rules:

$$Diapers \rightarrow Beer,\ support = 20\%,\ confidence = 85\%$$

Although patterns are clearly different from models, we will use *model* as the generic term.

Diapers $\rightarrow$ Beer

Reasons to Model

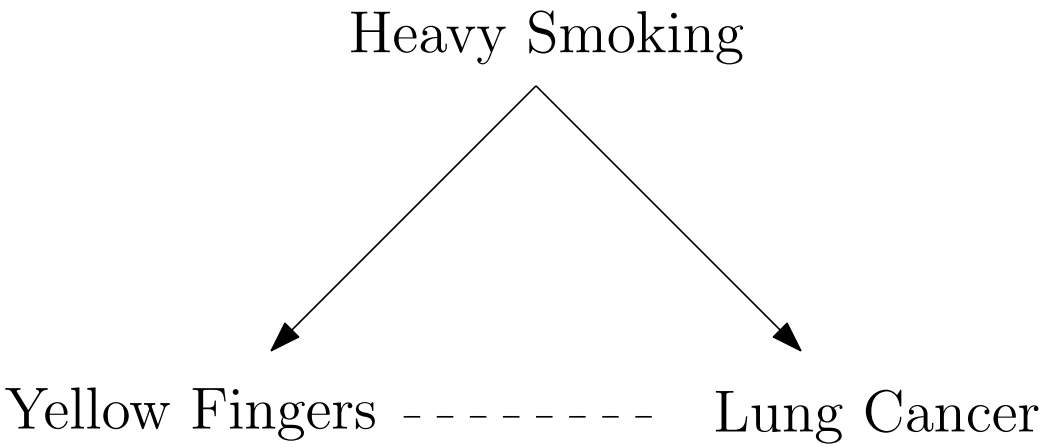
A model

- can help to gain *insight* into the application domain
- can be used to make *predictions*
- can be used for *manipulating/controlling* a system (causality!)

A model that predicts well does not always provide understanding.

Correlation $\neq$ Causation
Can causal relations be found from data alone?

Causality and Correlation



Washing your hands doesn't help to prevent lung cancer.

Induction vs Deduction

Deductive reasoning is *truth-preserving*:

1. All horses are mammals
2. All mammals have lungs
3. Therefore, all horses have lungs

Inductive reasoning *adds information*:

1. All horses observed so far have lungs
2. Therefore, all horses have lungs

Induction (Statistical)

1. 4% of the products we tested are defective
2. Therefore, 4% of all products (tested or otherwise) are defective

Inductive vs Deductive: Acceptance Testing Example

100,000 products; d is proportion of defective products

sample 1000; $\hat{d}$ is proportion of defective products in the sample

Suppose 10 of the sampled products turn out to be defective
(1% of the sample; $\hat{d} = 0.01$)

Deductive: $d \in [0.0001, 0.9901]$

Inductive: $d \in [0.004, 0.016]$ with 95% confidence.

95% confidence interval:

$$\hat{d} \pm se(\hat{d}) \times z_{0.975} = 0.01 \pm \underbrace{\sqrt{\frac{0.01 \times 0.99}{1000}} \times 1.96}_{\approx 0.006}$$

Experimental data

The experimental method:

- Formulate a hypothesis of interest.

  For example: "This fertilizer increases crop yield"
- Design an experiment that will yield data to test this hypothesis.

  For example: apply different levels of fertilizer to different plots of land and compare crop yield of the different plots.
- Accept or reject hypothesis depending on the outcome.

Experimental vs Observational Data

Experimental Scientist:

- Assign level of fertilizer randomly to plot of land.
- Control for other factors that might influence yield: quality of soil, amount of sunlight,...
- Compare mean yield of fertilized and unfertilized plots.

Data Miner:

- Notices that yield is somewhat higher under trees where birds roost.
- Conclusion: bird droppings increase yield;
- ... or do moderate amounts of shade increase yield?

Observational Data

- In observational data, many variables may move together in systematic ways.
- In this case, there is no guarantee that the data will be "rich in information", nor that it will be possible to isolate the relationship or parameter of interest.
- Prediction quality may still be good!

Example: linear regression

$$\widehat{\text{mpg}} = a + b \times \text{cyl} + c \times \text{eng} + d \times \text{hp} + e \times \text{wgt}$$

Estimate a, b, c, d, e from data. Choose values so that sum of squared errors

$$\sum_{i=1}^{n} (\text{mpg}_i - \widehat{\text{mpg}}_i)^2$$

is minimized.

$$\frac{\partial \widehat{\text{mpg}}}{\partial \text{eng}} = c$$

Expected change in mpg when (all else equal) engine displacement increases by one unit.

Engine displacement is defined as the total volume of air/fuel mixture an engine can draw in during one complete engine cycle.

The Data

```
> cars.dat[1:10,]
   mpg cyl eng  hp  wgt
1   18   8 307 130 3504 "chevrolet chevelle malibu"
2   15   8 350 165 3693 "buick skylark 320"
3   18   8 318 150 3436 "plymouth satellite"
4   16   8 304 150 3433 "amc rebel sst"
5   17   8 302 140 3449 "ford torino"
6   15   8 429 198 4341 "ford galaxie 500"
7   14   8 454 220 4354 "chevrolet impala"
8   14   8 440 215 4312 "plymouth fury iii"
9   14   8 455 225 4425 "pontiac catalina"
10  15   8 390 190 3850 "amc ambassador dpl"
...
```

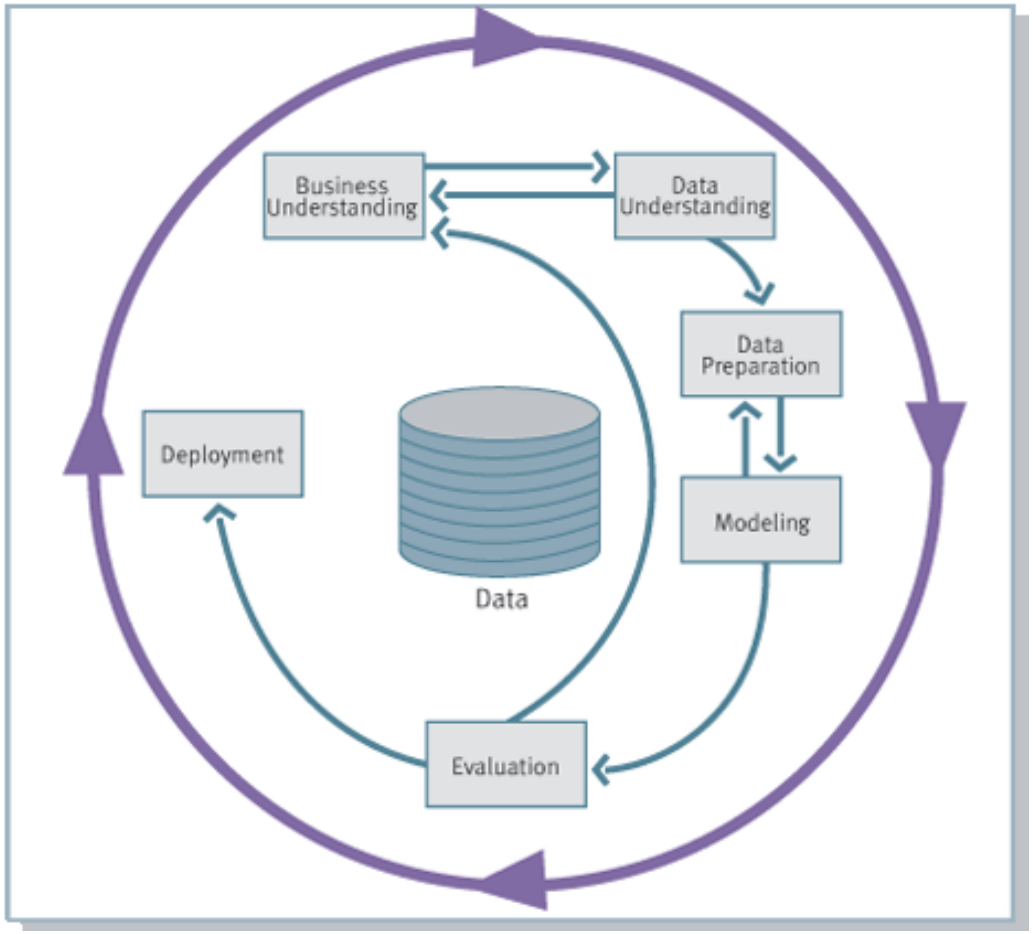
Fitted Model

```
Coefficients:
             Estimate Pr(>|t|)
(Intercept) 45.7567705  < 2e-16 ***
cyl         -0.3932854 0.337513
eng          0.0001389 0.987709
hp          -0.0428125 0.000963 ***
wgt         -0.0052772 1.08e-12 ***
---

Multiple R-Squared: 0.7077

> cor(cars.dat)
            mpg        cyl        eng         hp        wgt
mpg  1.0000000 -0.7776175 -0.8051269 -0.7784268 -0.8322442
cyl -0.7776175  1.0000000  0.9508233  0.8429834  0.8975273
eng -0.8051269  0.9508233  1.0000000  0.8972570  0.9329944
hp  -0.7784268  0.8429834  0.8972570  1.0000000  0.8645377
wgt -0.8322442  0.8975273  0.9329944  0.8645377  1.0000000
```

KDD Process: CRISP-DM



This course is mainly concerned with the modeling phase.

Data Cleaning

Cleaning data is a complete topic in itself, we mention two problems:

1. data editing: what to do when records contain *impossible* combinations of values?
2. incomplete data: what to do with missing values?

Data Editing: Example

We have the following edits (impossible combinations):

$E_1 = \{Driver's\ Licence{=}yes,\ Age < 18\}$
$E_2 = \{Married{=}yes,\ Age < 18\}$

Make the record:

Driver's Licence=yes, Married=yes, Age=15

consistent by changing attribute values.

What change(s) would you make? (Wooclap)

Of course it's better to *prevent* such inconsistencies in the data!

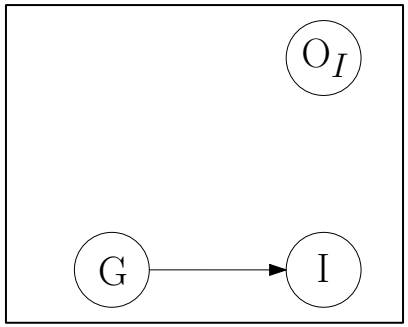
What to do with missing values?

- One can remove a tuple if one or more attribute values are missing.
  Danger: how representative is the remaining sample?
  Also, you may have to ignore a large part of the data!
- One can remove attributes for which values are missing.
  Danger: this attribute may be important.
- You do *imputation*: you fill in a value.
  Note: but not just any value!

Missing Data Mechanisms: Not Data Dependent (NDD)

Suppose we have data on gender and income.
Gender (G) is fully observed, income (I) is sometimes missing.
O_I indicates whether income is observed ($O_I = 1$) or not ($O_I = 0$).

For example, missingness is determined by the roll of a die.



- There will be no bias if we remove tuples with missing income.
- If we do imputation, what values should we fill in? (Wooclap)

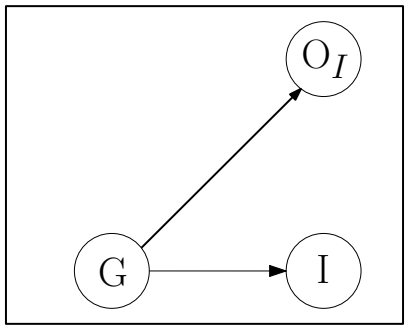
Missing Data Mechanisms: NDD

We could perform imputation as follows:

- If person is male, pick a random male with income observed and fill in his value.
- If person is female, pick a random female with income observed and fill in her value.

Missing Data Mechanisms: Seen Data Dependent (SDD)

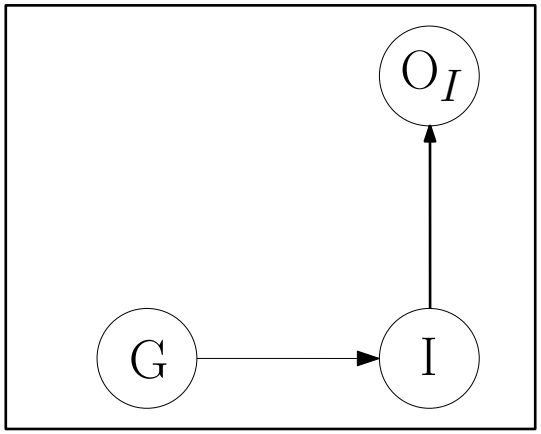
For example: men are less likely to report their income than women.



- This time there *will* be bias if we remove tuples with missing income. What bias? (Wooclap)
- Imputation: same as before, still works.

Missing Data Mechanisms: Unseen Data Dependent (UDD)

For example: people with high income are less likely to report their income.



- We can't "fix" this unless we have knowledge about the missing data mechanism.

Missing Data

- NDD is a necessary condition for the validity of complete case analysis.
- SDD provides a minimal condition on which valid statistical analysis can be performed without modeling the missing data mechanism.
- Unfortunately, we cannot infer *from the observed data alone* whether the missing data mechanism is SDD or UDD.
- We might have knowledge about the nature of the missing data mechanism however ...
- Practice: if you don't know, assume SDD and hope for the best.

Construct Features

Quite often, the raw data is not in the proper format for analysis, for example:

- You have data on income and fixed expenses and you think disposable income is important.
- You have to analyze text data, for example hotel reviews. You could represent the text as a *bag-of-words*.
- Relational data bases: 1:1 relationships between tables are easy, but what to do with 1:n relationships?

Bag of Words

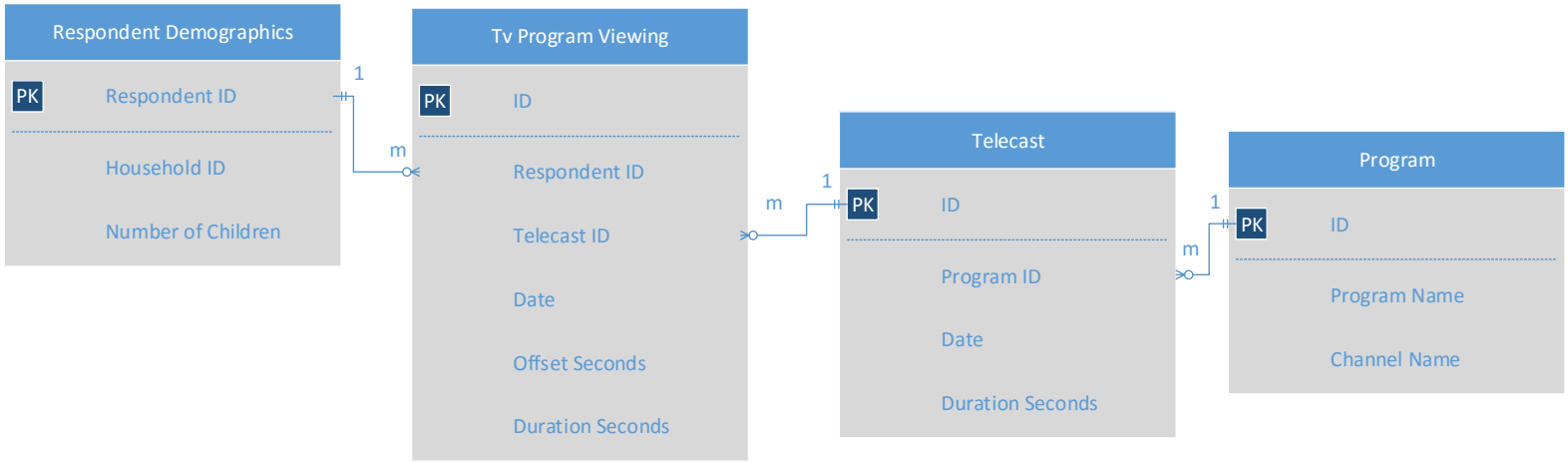
```
Doc 1: a view to a kill
Doc 2: license to kill
```

Bag of words representation:

	a	view	to	kill	license
Doc 1	2	1	1	1	0
Doc 2	0	0	1	1	1

One-to-Many Relationships: TV Viewing

Predict household composition from TV viewing behavior.



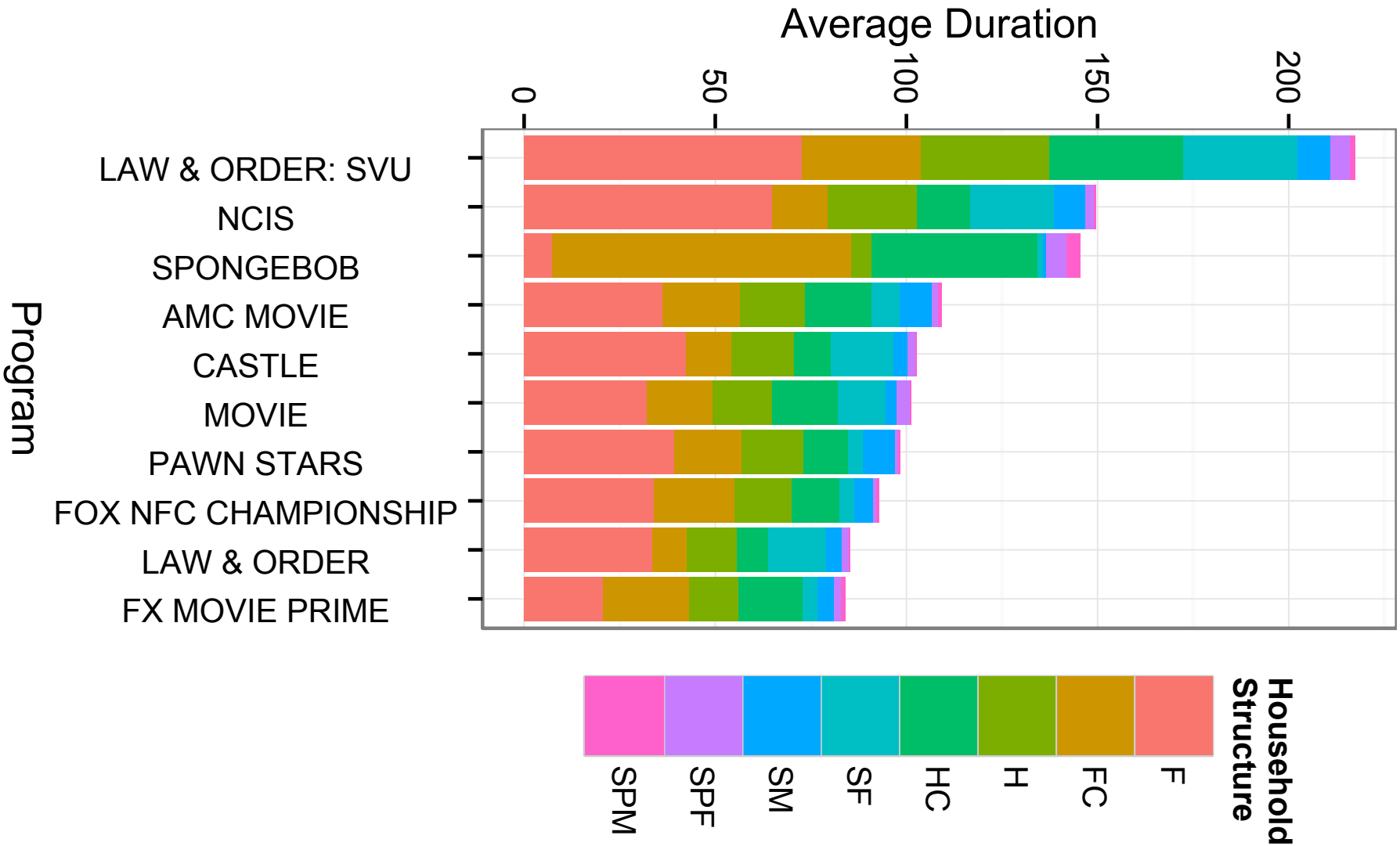
Aggregating the data to household level

Viewing behaviour has to be aggregated, for example:

- Weekly viewing frequency of different programs.
- Weekly viewing duration of different programs.
- Weekly viewing frequency of different program *categories*.
- etc.

Potentially results in a huge number of attributes.

Some descriptive statistics



Modeling: Data Mining Tasks

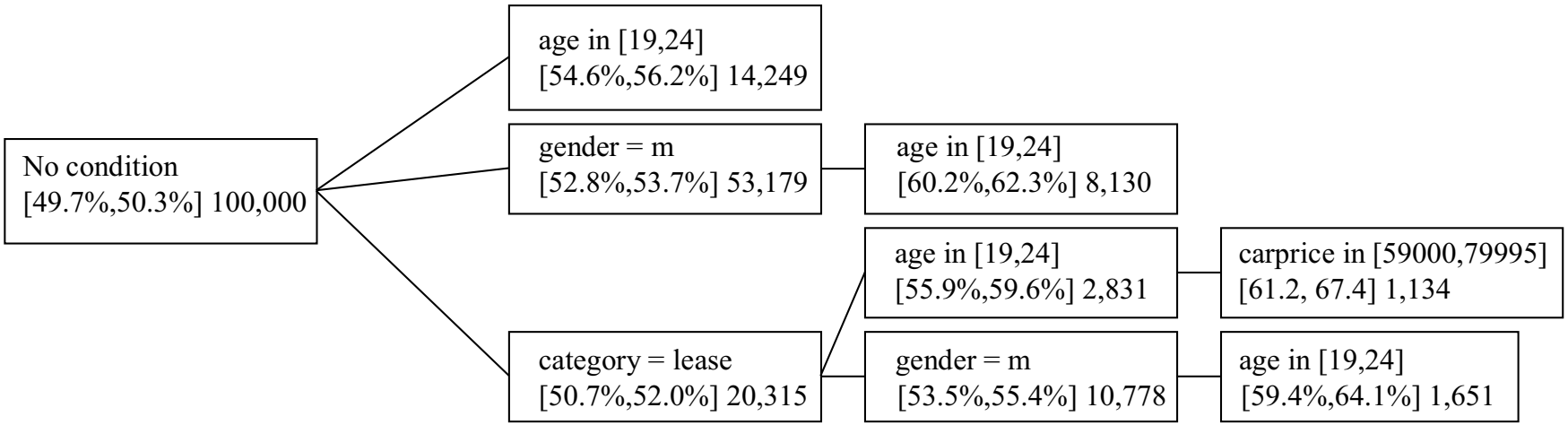
Common data mining tasks:

- Classification / Regression
- Dependency Modeling (Graphical Models; Bayesian Networks)
- Frequent Pattern Mining (Association Rules)
- Subgroup Discovery (Rule Induction; *Bump-hunting*)
- Clustering
- Ranking

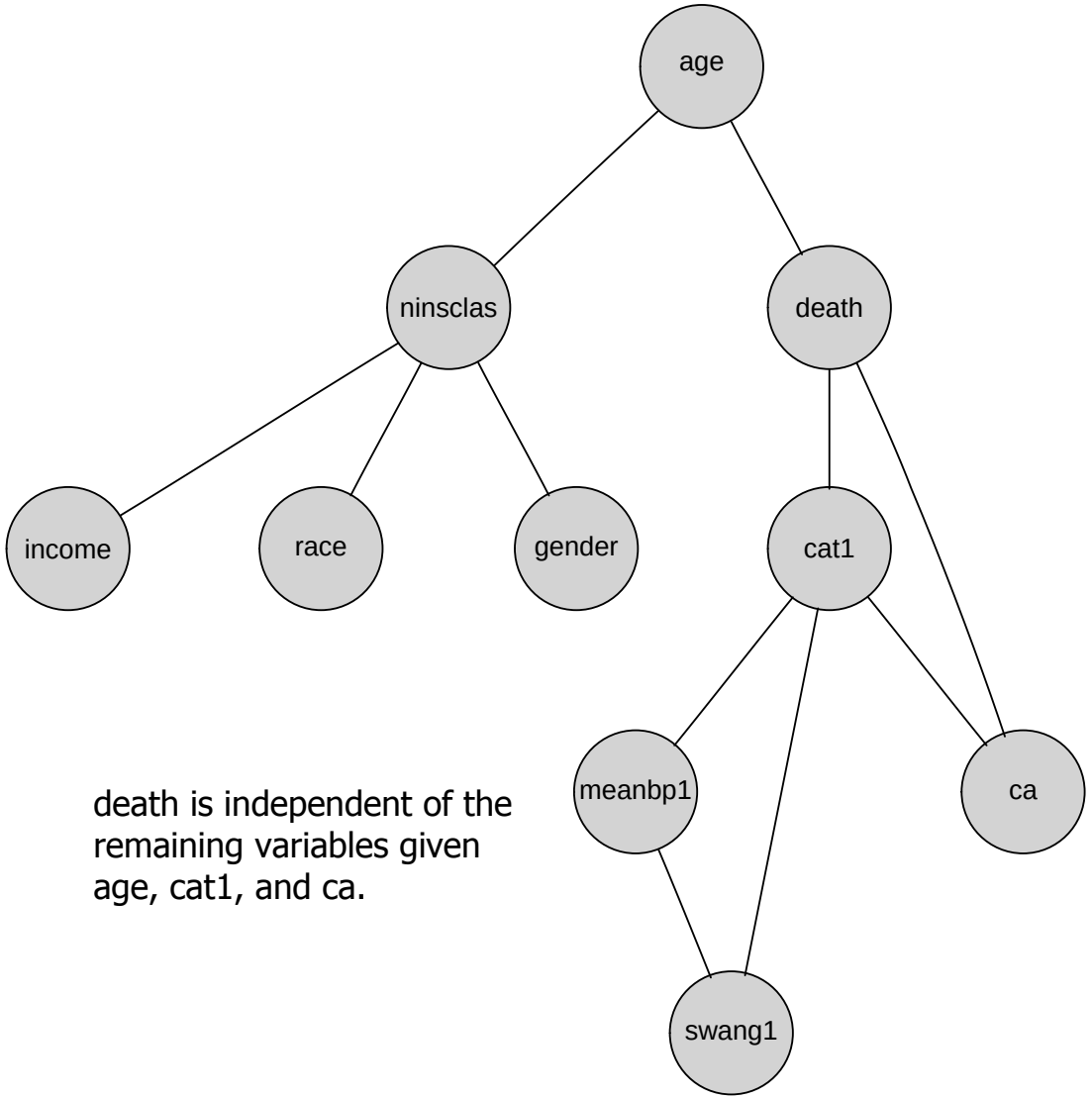
Subgroup Discovery

Find groups of objects (persons, households, transactions, ...) that score relatively high (low) on a particular *target* attribute.

Car insurance example (target: did person claim?):



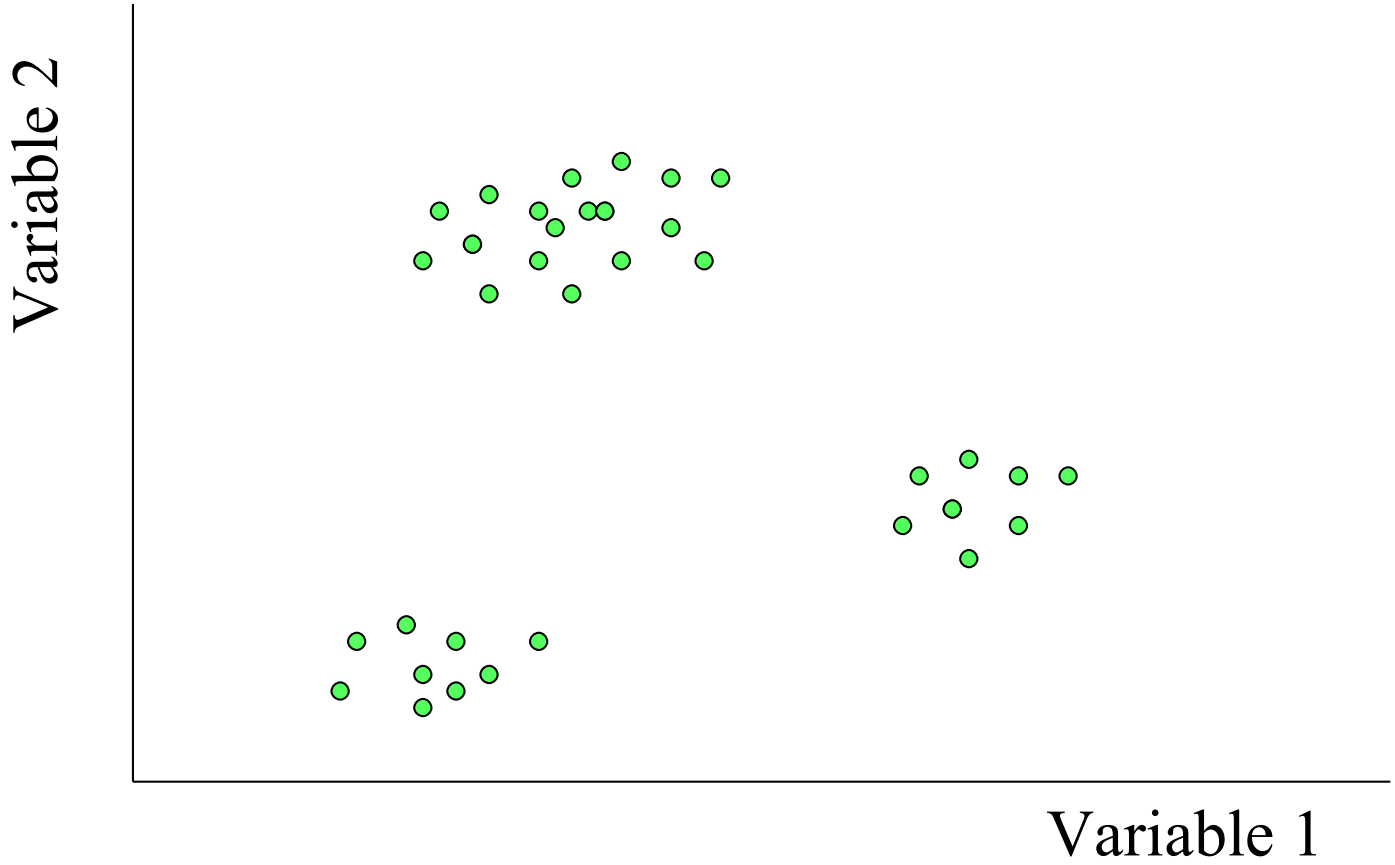
Dependency Modeling: Intensive Care Data



death is independent of the remaining variables given age, cat1, and ca.

Clustering

Put objects (persons, households, transactions, ...) into a number of groups in such a way that the objects within the same group are similar, but the groups are dissimilar.



Ranking

For example:

- Rank web pages with respect to their relevance to a query.
- Rank job applicants with respect to their suitability for the job.
- Rank loan applicants with respect to default risk.
- ...

Has similarities with regression and classification, but in ranking we are often only interested in the *order* of objects.

Components of Data Mining algorithms

Data Mining Algorithms can often be regarded as consisting of the following components:

1. A representation language: what models are we looking for?
2. A quality function: when do we consider a model to be good?
3. A search algorithm: how de we go about finding good models?

Representation Languages

Representation languages define the set of all possible models, for example:

- linear models: $y = b_0 + b_1 x_1 + \cdots + b_m x_m$
- association rules: $X \rightarrow Y$
- subgroups: $X_1 \in V_1 \wedge \cdots \wedge X_m \in V_m$
- classification trees
- Bayesian networks (DAGs)

Quality Functions

The quality score of a model often contains two elements:

- How well does the model fit the data?
- How complex is the model?

For example (regression)

$$\text{score} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + 2 \times \#\text{ parameters}$$

If independent test data is used, the quality score usually only considers the fit on the test data.

Overfitting on the training data

Slogan:

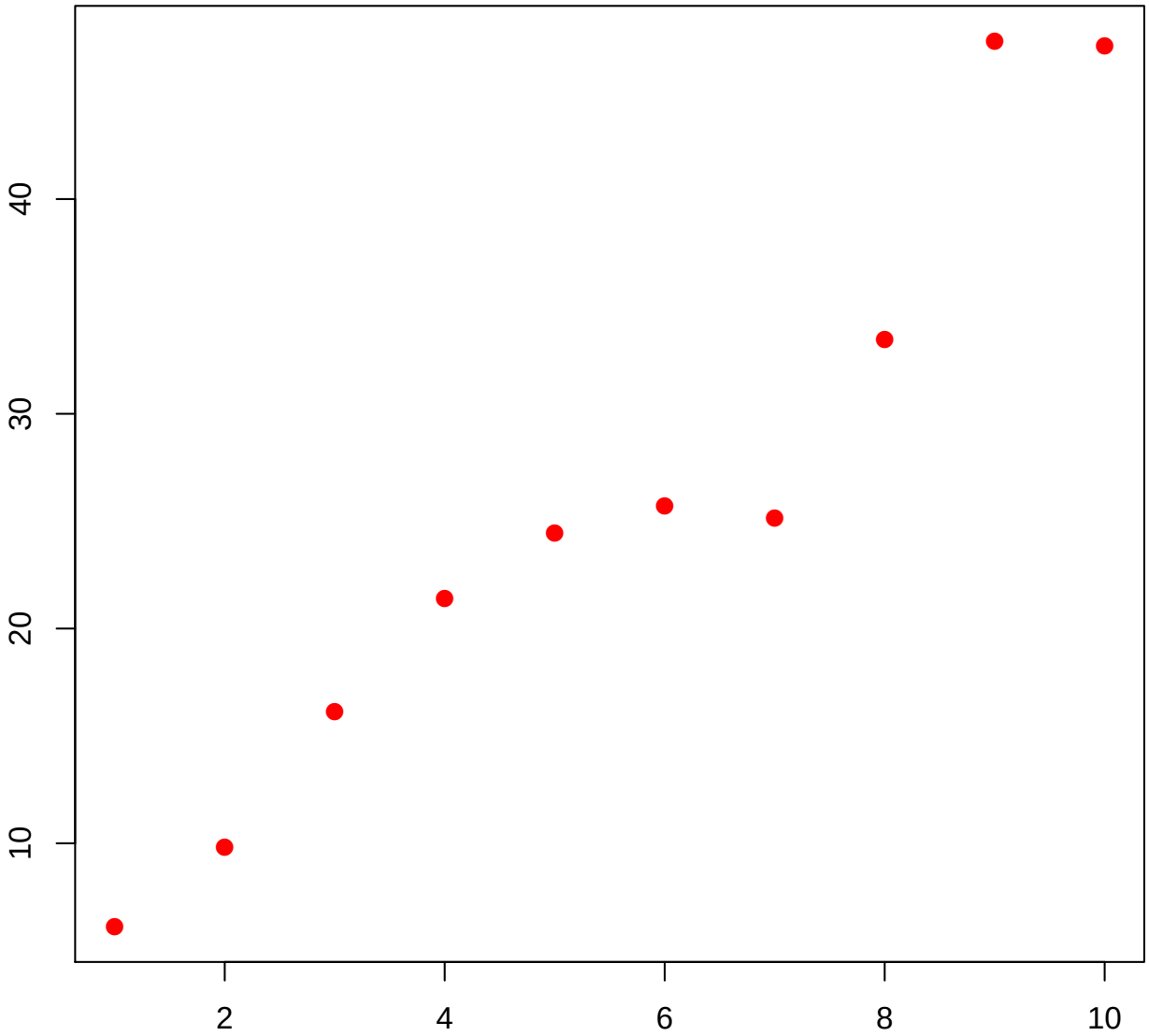
$$\text{DATA} = \text{STRUCTURE} + \text{NOISE}$$

We want to capture the structure, not the noise!

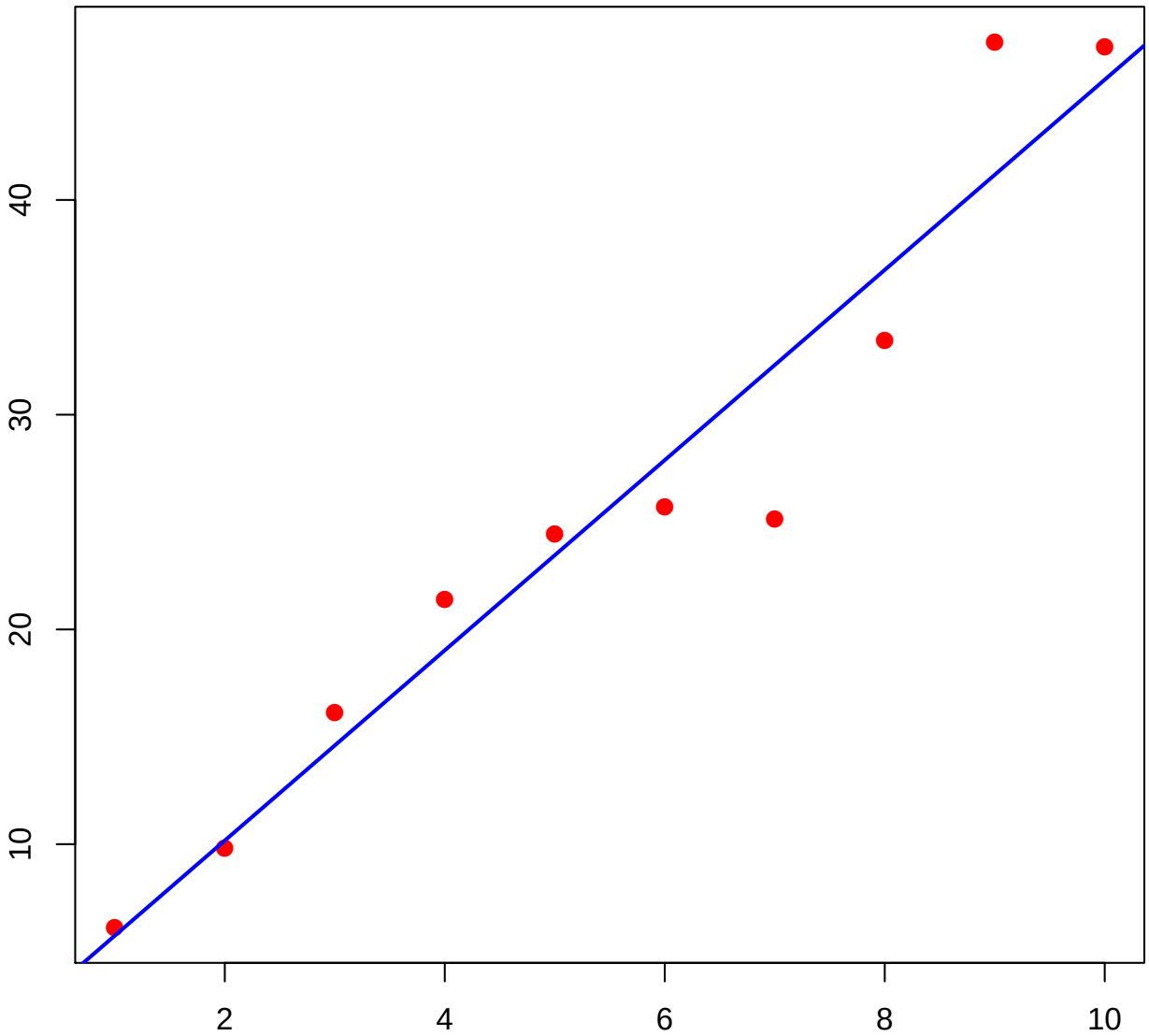
Regression example:

$$y_i = a + bx_i + \varepsilon_i \qquad\qquad i = 1, \ldots, n$$

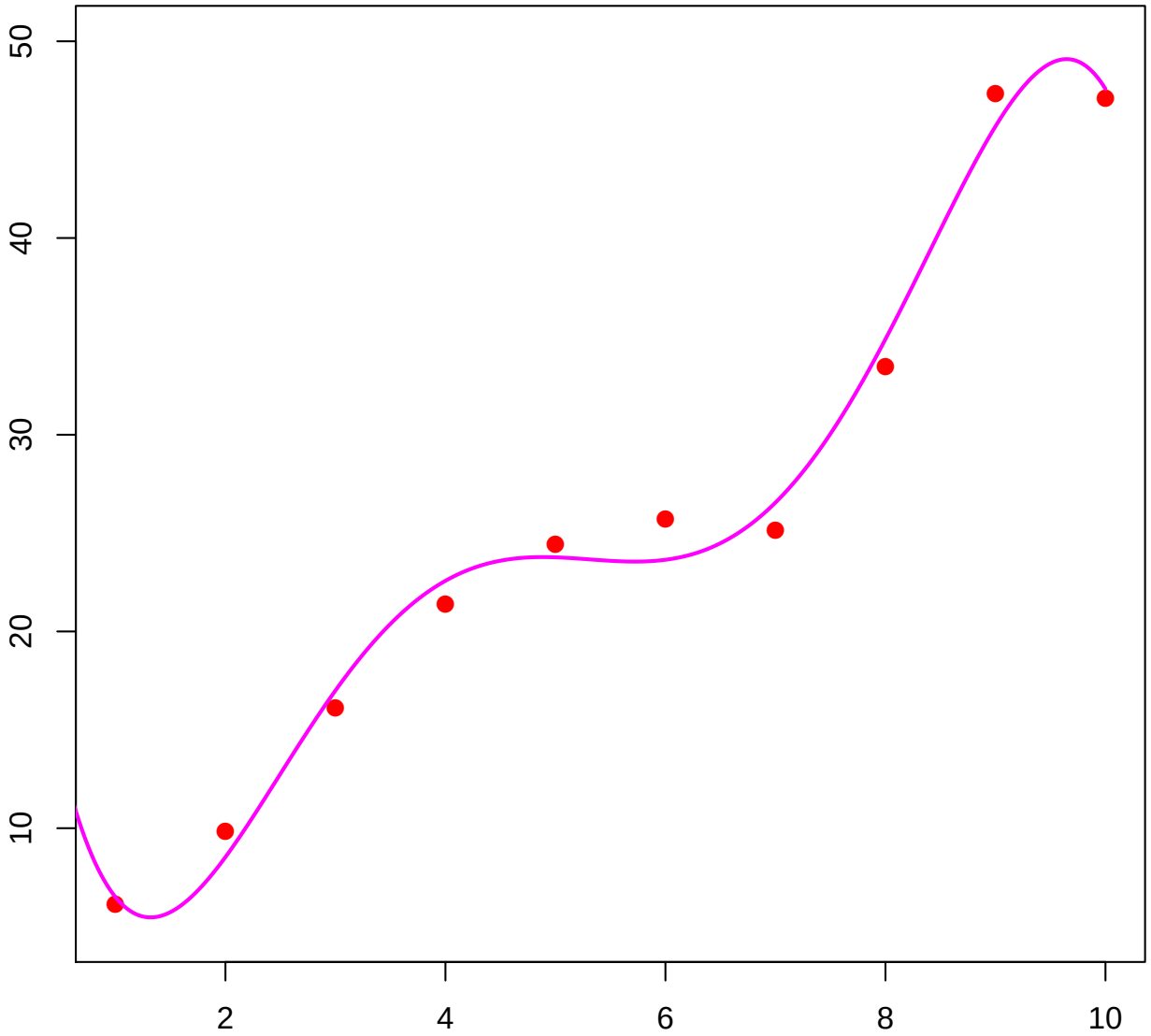
The training data: $y_i = a + bx_i + \varepsilon_i$



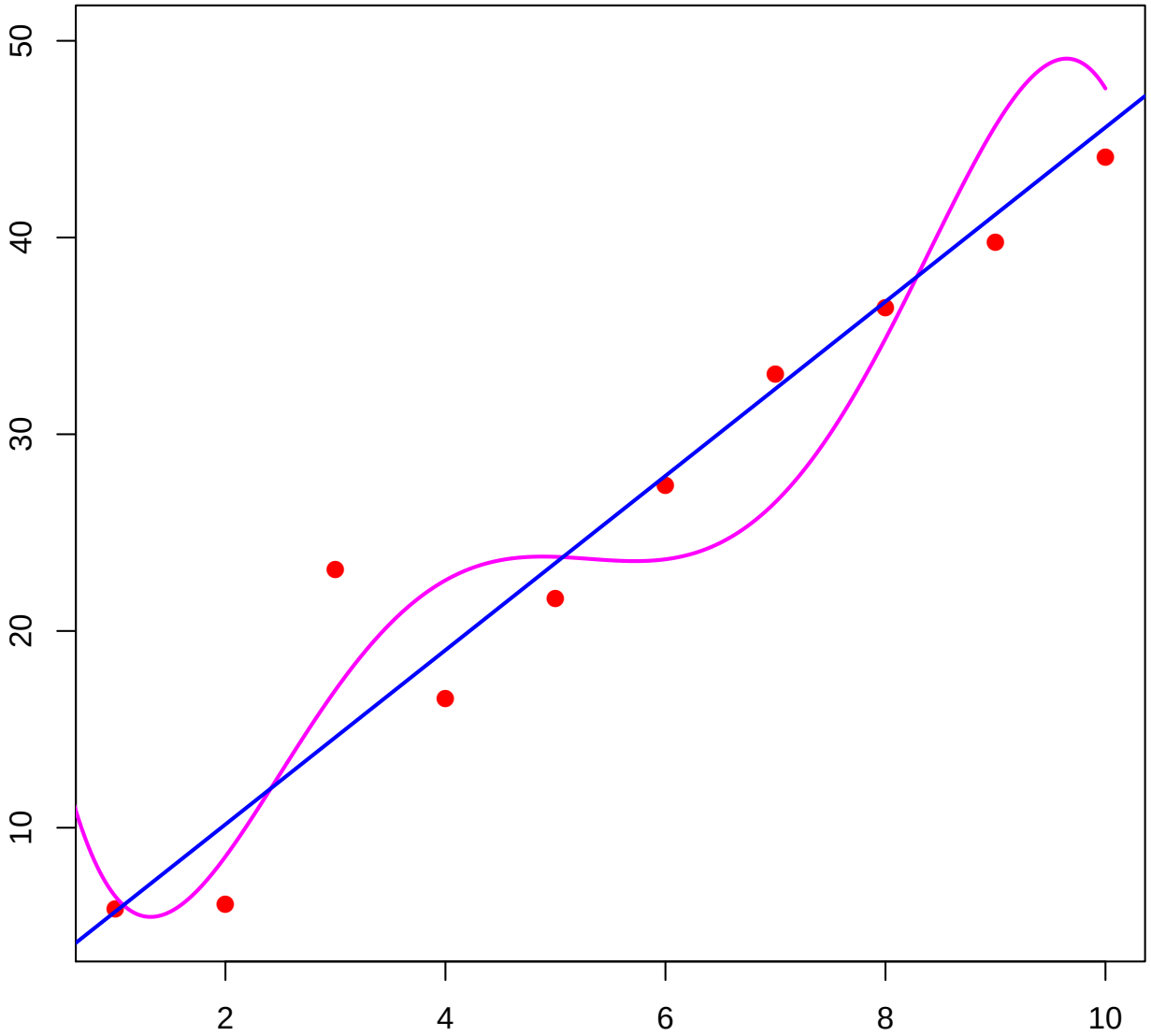
Fitting a linear model to the training data



A degree 5 polynomial fits the training data better!



Overfitting: linear model generalizes better to new data



Search for the good models

- Sometimes we can check all possible models, because there are rules with which to prune large parts of the search space; for example, the "a priori principle" in frequent pattern mining.
- Usually we have to employ *heuristics*
  - A general search strategy, such as a hill-climber or a genetic (evolutionary) algorithm.
  - Search operators that implement the search strategy on the representation language. Such as, a neighbour operator for hill climbing and cross-over and mutation operators for genetic search.

Search: example

In linear regression, we want to predict a numeric variable y from a set of predictors $x_1, \ldots, x_m$. We might include any subset of predictors, so the search space contains 2^m models. E.g., if $m = 30$, we have $2^{30} = 1,073,741,824$, i.e. about one billion models in the search space.

It is common to use a hill-climbing approach called stepwise search:

1. Start with some initial model, e.g. $y = a$, and compute its quality.
2. Neighbours: add or remove a predictor.
3. If all neighbours have lower quality, then stop and return the current model; otherwise move to the neighbour with highest quality and return to 2.

Classical Text Book Approach (Theory Driven)

- Specify hypothesis (model) of interest. The model is determined up to a fixed number of unknown parameter values.
- Collect relevant data.
- Estimate the unknown parameters from the data.
- Perform test, typically whether a certain parameter is zero, using the same data!

It is allowed to use the same data for fitting the model and testing the model, because we did not use the data to determine the model specification.

Data Mining (Data Driven)

A simple analysis scenario could look like this:

- Formulate question of interest.
- Select potentially relevant data.
- Divide the data into a training and test set.
- Use the training set to fit (many) different models.
- Use the test set to compare how well these models generalize.
- Select the model with the best generalization performance.

In this scenario, we cannot use the training data both to fit models and to test models!

Vacancies in Education Advisory Committee

The master Education Advisory Committee (EAC) of computer science has 3 vacancies:

1. Student member Data Science (DASC)
2. Student member Computing Science (COSC)
3. Student member Game- and Media Technology (GMTE)

The EAC gives solicited and unsolicited advice about individual courses, the curriculum, Education and Exam Regulations (EER), etc.

Typically 5 meetings per year to discuss course evaluations (i.e. Caracal) after each period, and the Education and Exam Regulations (EER).

Financial compensation $\pm$ €600.

Interested? Send e-mail with motivation to `a.j.feelders@uu.nl`.