On practicing machine learning or “data science”

in a commercial consultative context in Finland right now

Janne Sinkkonen
Reaktor

My context here: Reaktor

Consulting: we develop for customers, not for ourselves.
Commercial: customers pay our salary.
Mostly in **Finland**, now a bit in US

~300 people in total, 8 PhD data scientists
Our group has been around for 2–3 years.

Reaktor does **software projects**, mostly around web.
Organic growth since year 2000
Flat (self-organization), informal culture

janne.sinkkonen@reaktor.fi
careers@reaktor.fi

What we do

Sell, in a benign sense
Concept work: needs —> processes and models

Handle data: collect, organize, recode, clean
Model
Implement
Visualize, report

But everything is in move, projects vary, and N is small.

Tools

R, **Python**, Java/Scala/Clojure, C/C++, (Javascript)

In R: **dplyr**, **ggplot2**, glm, lme4, mgcv, rstan

DB's, API's

Special tools: hca (for LDA), Vowpal Wabbit, …

Macs and servers, Amazon EWS, Heroku

Different specialisations

Coding statistician preferring small data

Hadoop/Spark Amazon-EWS ML guy

Salesman (PhD)

Doing concepts, modelling and training

Visualization and R expert

…

Hype

Has not always been, nor will it be.

You must now do “**big data**” and “**data science**”.

But (in Finland?) we have mostly small data!
Datasets fit in memory, at least for modelling.

Markets have expanded, and are a bit premature.

Data science?

Statistics, **ML**, algorithms

Infrastructure: accumulation, storage (DB), API's

Coding

Tools for iterative and interactive work (R etc.)

Defining the problems

Data

Data: stored bits. Very dangerous definition!
Hides the processes behind data.
What creates the data? What is done with the results?

The goal is not data analysis but

do causal inference
predict
find optimal actions

Define your goal and setup without using the word ‘data’.

Think in terms of a process, population, sample, dependencies, randomisation, experiments, utility, actionable outcome.

Old boring concepts

Is your **sample** representative (unbiased)?

Representative of what: **population**. What is it?

Causality? You need randomisation.

Randomization means intervening with the system.

Are you predictions **actionable**?

Predictions on actions need **experiments**.

Are **utilities** well defined?

Tools are preferred

Projects have budgets.
Even if not, people may expect you to deliver.
—> **Projects tend to use ready-made tools**.

In terms of modeling, this means **compromises**,
unless you have a very flexible tool.

Either use mgcv, lme4 or glmnet.
mgcv would have splines.
lme4 would have nice random effects (for nominal variables).
glmnet would have L1 for sparseness.

Probabilistic programming may help with this.

Almost science

Data science projects are more **unpredictable** than coding, almost like research.
—> perfect fit to agile mindset (minimal plans, iterations)
—> may be hard to sell

Organizations may not be ready for empirical.
Empirical means unpredictable and **dangerous**. :)
Results are not always used.
Agile may be hard except within small teams.

Examples from practice

Customer segmentation

Price optimization
Data has no (random) price variation.

Customer churn prediction: not actionable

Recommendation

Quick predictive models from passive data

Control the dynamics of a wave-power plant

In future

Probabilistic programming

Flexibility of modelling increases.

Stan, with VB, EP, stochastic approximation, etc.

Passive big data —> **massive experiments**

Julia?

Commodisation: just plug into an API etc.?

Web technologies: API's, Node.js, event-oriented

Two key points

Context is important.

Process and actions, not data

Practical problems break the standards setups.

—>Tools need to be flexible, and are becoming so.