Introduction to machine learning

Magnus Nielsen, SODAS, UCPH

Agenda

- SODAS & me
- Overall course agenda
- Expectations
- Motivation
- Causal inference & prediction issues
- Machine learning in the social sciences
- Python & Anaconda
- Environments
- Exercises

SODAS & me

SODAS

Copenhagen Center for Social Data Science

- What is SODAS?
  - Computational social science & digital methods (?)
- Interdisciplinary
  - Within social sciences
  - With the natural sciences
- Qualitative & quantitative

Me

- Ph.d. fellow at SODAS
- Social data scientist / computational social scientist (economist)

- A love of data
  - Nation-scale social networks
  - UCPH & DST
  - Prediction with sequence data

- Teaching
  - Courses at the Economics Department and Social Data Science

Andreas Bjerre-Nielsen

- Associate professor at the department of Economics and SODAS, UCPH

Overall course agenda

Session #	Topic
1	Introduktion til kurset og ML
2	Indførsel til Python
3	Model- og hyperparameterselektion
4	Supervised ML
5	Unsupervised ML
6	Fortolkning af modeller
7	Algorithmic audits
8	Kausalitet – Træbaserede modeller
9	Kausalitet - double machine learning

The practical stuff

Slides and exercises (+ possible reading list) will be available through the course website

The course will be mostly hands on, and reading is not necessary!

Some of you might find Python Machine Learning useful if you want a book that introduces machine learning in Python

- All the information can be found online

Expectations

Your expectations

What do you expect from this course?

- Specific subjects to cover?
- Specific skills to learn?

- How much time do you want to spend preparing?
  - Reading or Python?

My expectations

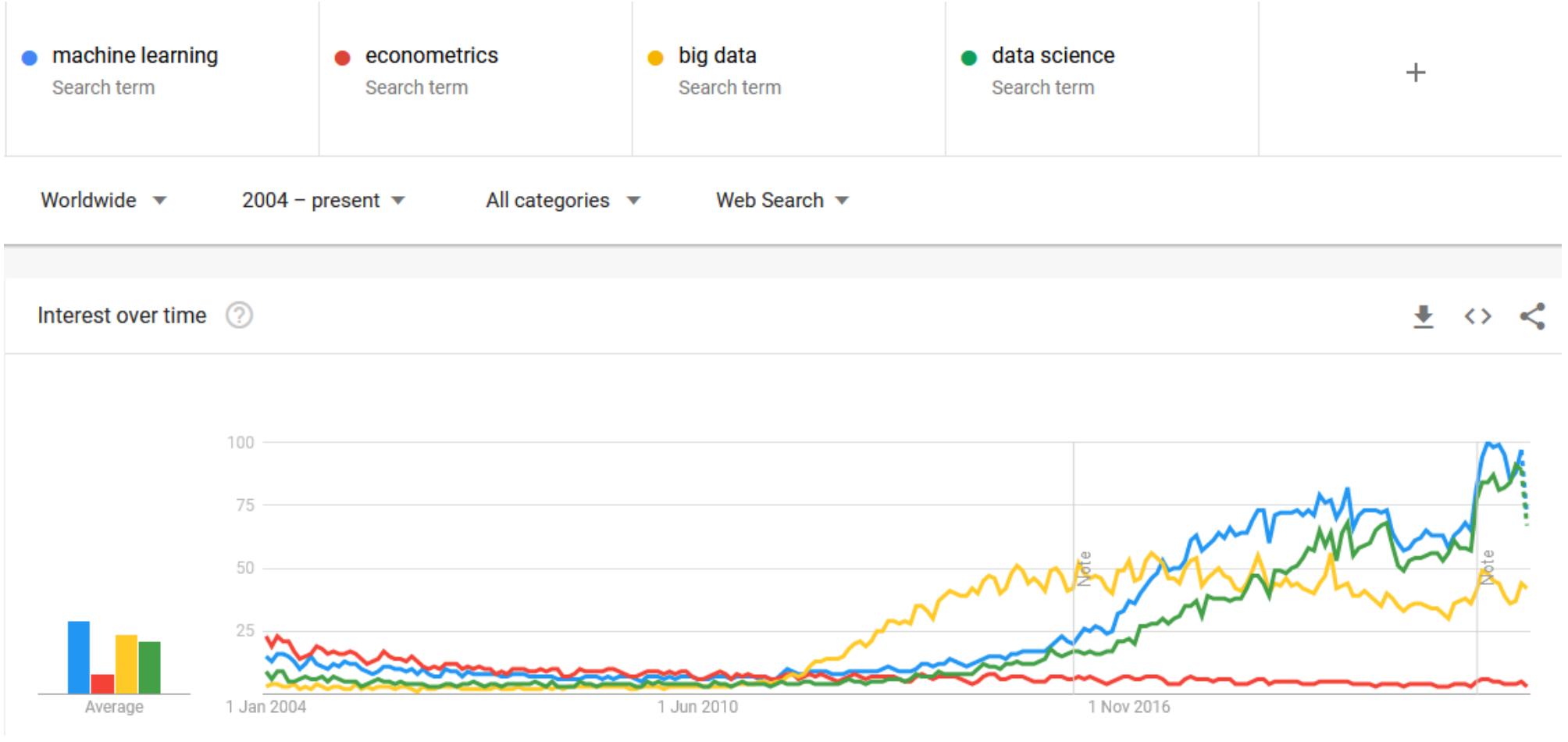
- A diverse group
  - Different parts will be difficult to different people
  - Talk to eachother, the internet and me
- You will get hands on experience with implementation
  - Become comfortable with Python
  - Be able to implement and evaluate machine learning algorithms
- Impossible to become experts in 9 sessions
  - But we’ll try nevertheless!

Motivation

Twofold

1. Machine learning
2. Python

Machine learning is popular

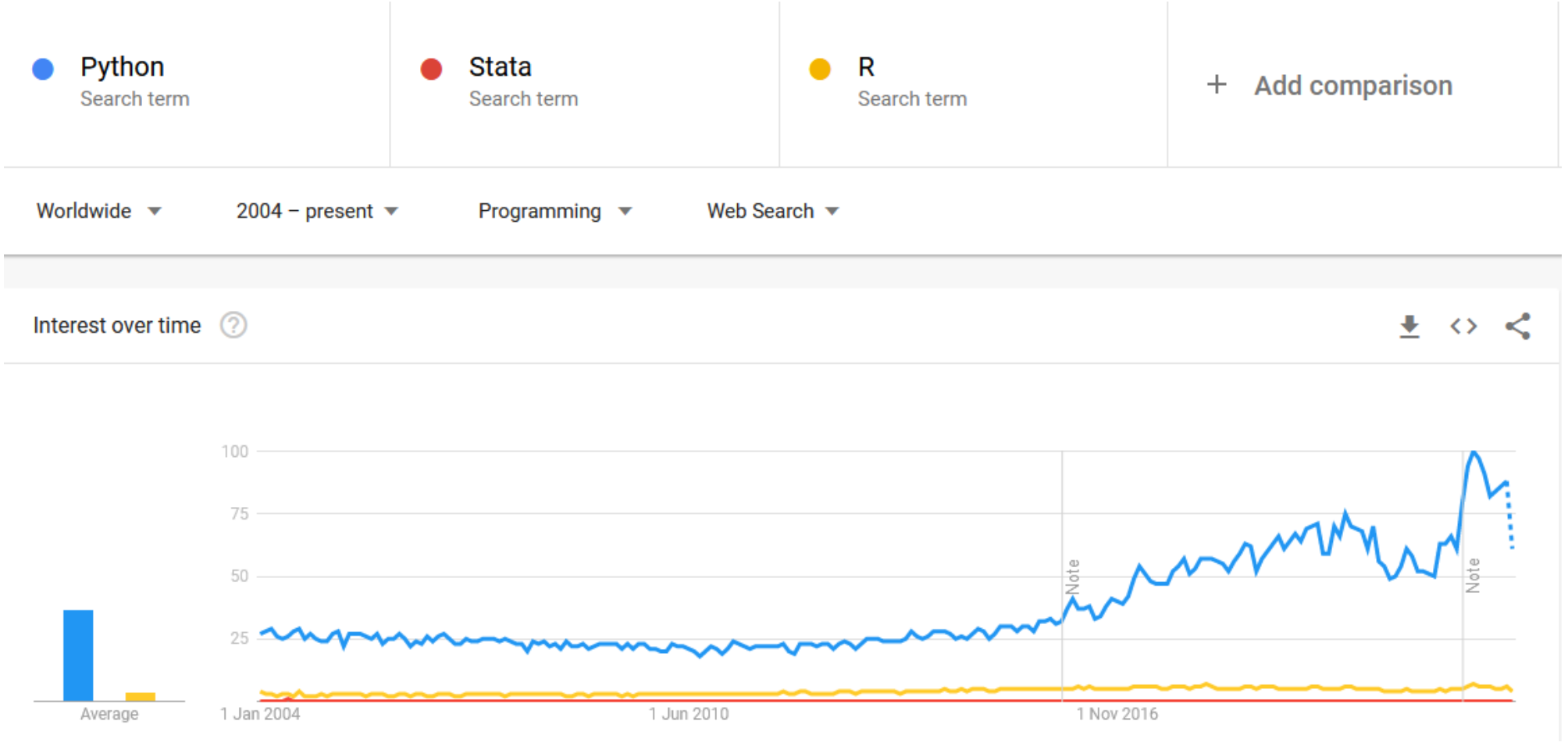


Source: Google Trends

But why is it popular?

- More and more data available
  - Important to efficiently use this data
  - Enables use of previously proof-of-concept models
- Models are becoming more capable and flexible
- Insights from machine learning can be used for other things
  - When is it no longer machine learning?

Python is also popular



Source: Google Trends

Popularity is once again not be-all end end-all, however…

- Python is a general purpose language
  - Can do a lot of different things!
- Python (and R) is open source
- Bigger communities have more support and active development
- Consistent interface through sklearn

R (and Stata) are probably better in some settings (e.g. statistics)

Question

Who here has experience with:

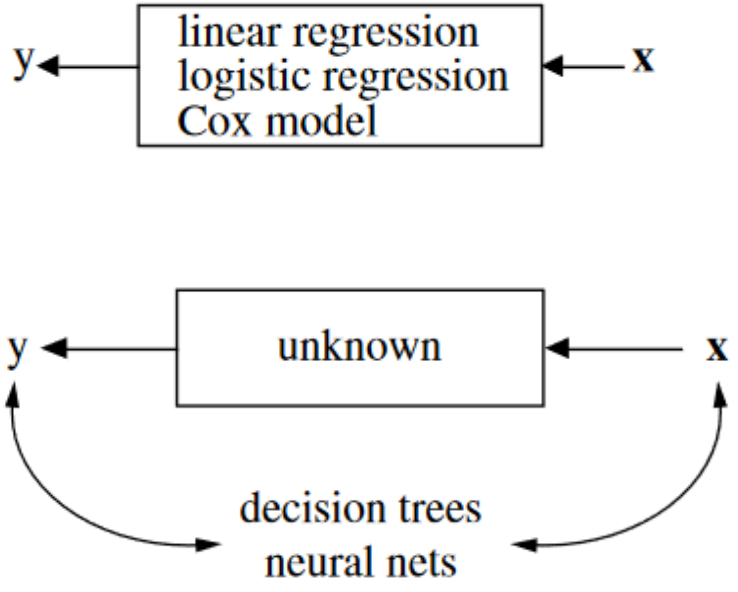
- Stata
- R
- Python

Causal inference & prediction issues

Two different cultures

Breiman (2001) postulates that there exists two cultures (within statistics):

- The data modeling culture
  - Assumes a specific data generating process
  - Model validation: goodness-of-fit tests and residual examination
- The algorithmic culture
  - Assumes a black box
  - Model validation: predictive accuracy



Source: Breiman, 2001

Question

In which culture would you place yourself?

Two different problems

The first culture still dominates economics and social sciences (Athey & Imbens, 2019; Verhagen, 2022)

Breiman (2001) writes that “our goal as a field is to use data to solve problems;” (referencing statistics)

Should we as social scientists limit ourselves to just one culture?

NO!

Question

1. Have you worked on projects where prediction was the main goal?

2. Have you worked on projects which could be rephrased as a prediction problem?

3. Is forecasting and prediction the same? Why?

Causal inference

Causal inference centers around treatment effects

Average treatment effect

$$\tau = E[Y_i(1) - Y_i(0)]$$

Subgroup treatment effect

$$\tau_g = E[Y_i(1) - Y_i(0)|G_i = g]$$

Conditional treatment effect

$$\tau(x) = E[Y_i(1) - Y_i(0)|X_i = x]$$

Requires large sample properties, such as unbiasedness, consistency, normality and efficiency

Causality in *most* 'machine learning papers' often center around directed acyclic graphs and structural causal models, and do not utilize the potential outcomes approach

Prediction

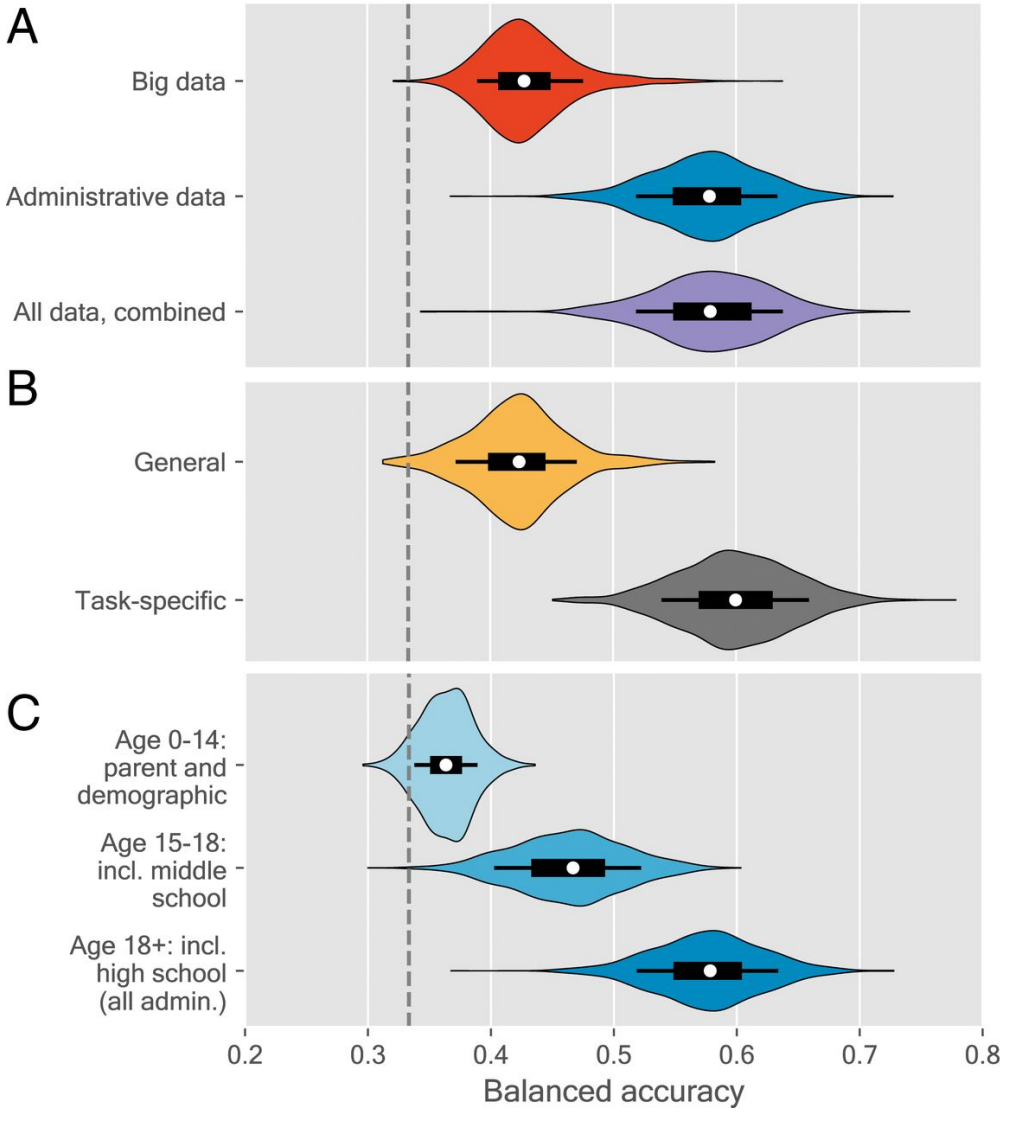
Prediction centers around minimizing loss functions, $L(\hat{y}, y)$

Classification: Accuracy

$$L(\hat{y}, y) = I[\hat{y} \neq y]$$

Regression: Mean squared error

$$L(\hat{y}, y) = (\hat{y} - y)^2$$



Source: Bjerre-Nielsen et al., 2021

Machine learning in the social sciences

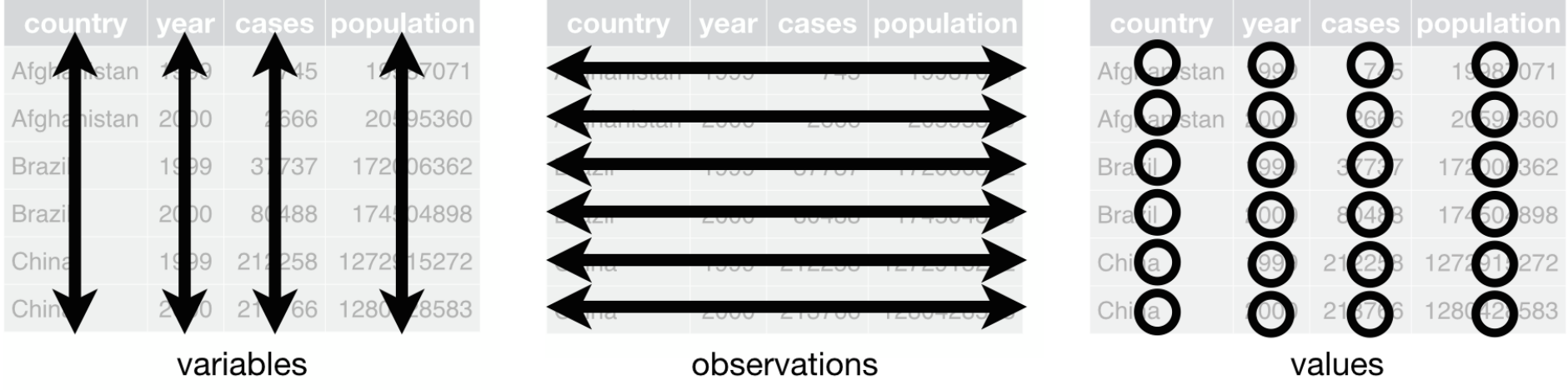
Three paradigms exist

- Supervised learning
  - Models designed to infer a relationship between input and **labeled** data
  - We define the **target** as labels in data that we wish to model
- Unsupervised learning
  - Find patterns and relationships from **unlabeled** data
  - This may involve clustering (e.g. group objects that share certain degree of similarity), dimensionality reduction and more
- Reinforcement learning
  - Models to infer optimal behavior in some (potentially) mathematically unknown environment
  - Needs no labeling, and suboptimal behavior is corrected through experience (penalty/reward)
  - *Not covered*

Supervised learning

Has a given target and uses structured datasets

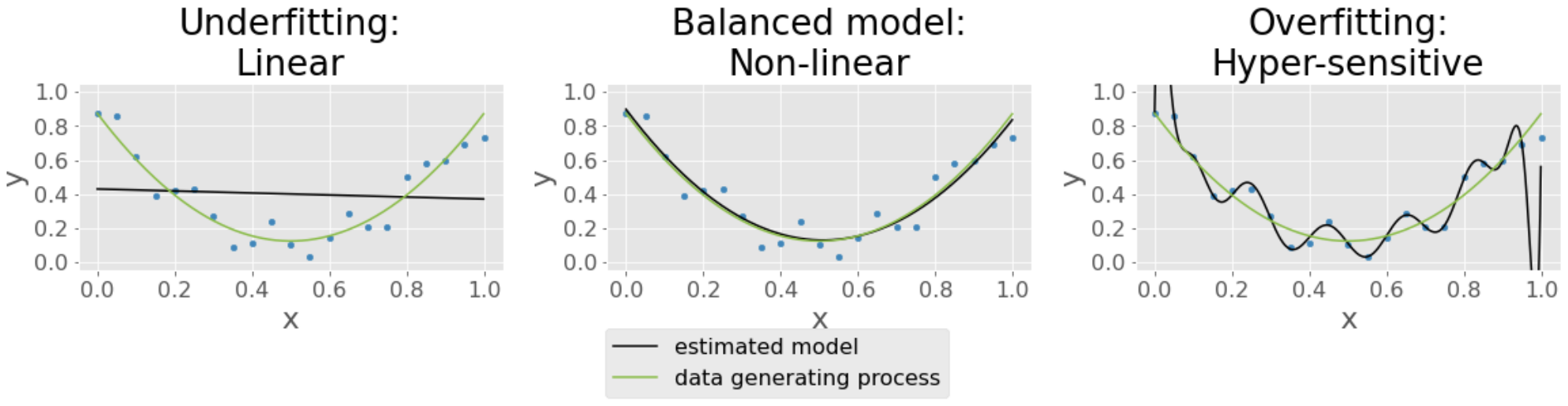
- Every column is a variable
- Every row is an observation
- Every cell is a single value



So… like OLS?

Supervised learning

We control the bias-variance trade-off!

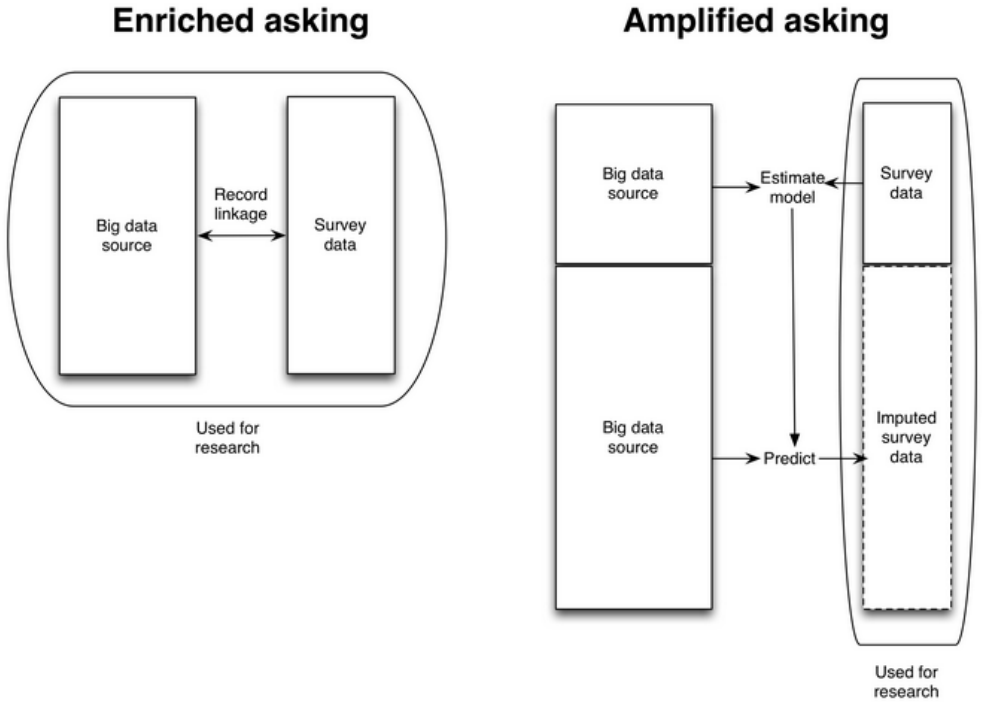


This is done using model- and hyperparameterselection

Supervised learning

When is this useful?

- Prediction policy problems (Kleinberg et al., 2015)
- Inferring data to enhance datasets (Salganik, 2019)



Source: Salganik, 2019

Unsupervised learning

No given target and structure not necessary

Different models ‘create’ their own target and structure

Utilize data sources such as text and images in novel ways

Generate *new* data, such as text or images

- Can require huge amounts of compute and advanced programming knowledge
  - More easily accesible models exist
    - Pretrained models packaged in an easy-to-access API

HUGGING FACE

Unsupervised learning

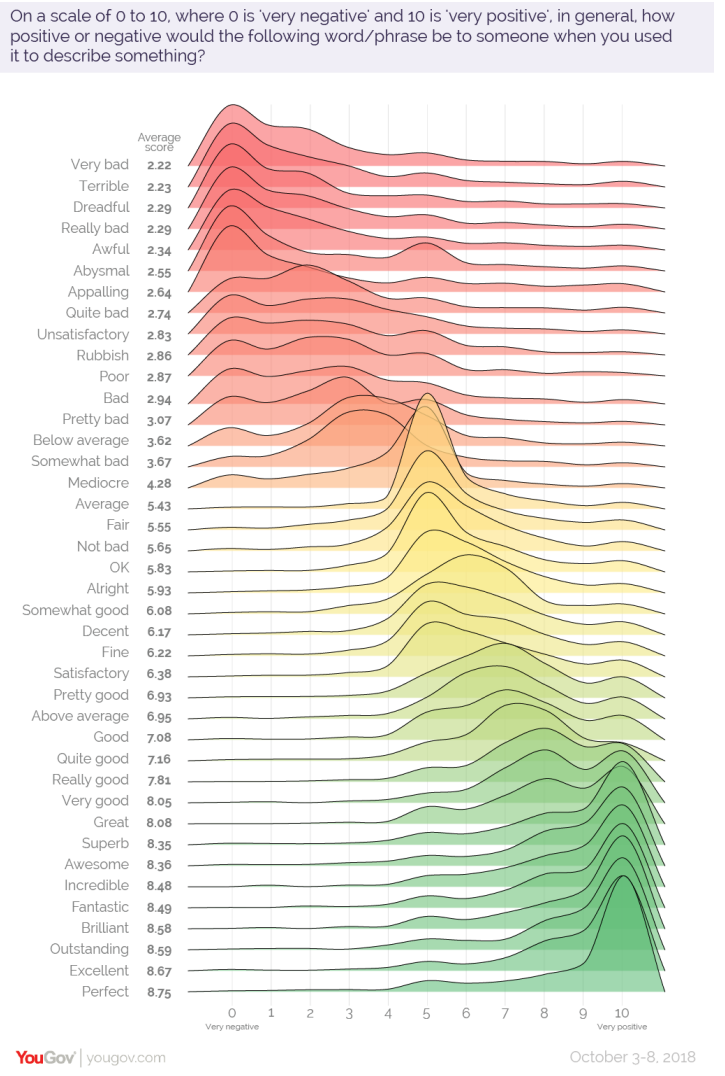
We are able to reduce the dimensionality of high-dimensional inputs, enabling new uses (old methods)

- Wellbeing surveys in primary school
- Clustering
  - Split articles into topics (Athey et al., 2021)

Transforming text into useful variables for further analysis

- Topic models (Blei, 2012)
- Sentiment analysis

Aid in interpretation of text itself (Nelson, 2020)



Source: YouGov, 2018

Python & Anaconda

What's what?

Python

- The programming language itself

conda

- A package and environment manager

Anaconda

- A distribution which includes Python and conda (+ many other things)

PyCharm

- An integrated development environment (IDE)

Question

Who has a working Python installation (can print ‘hello world’)?

The statement `print('hello world')` should return ‘hello world’.

Scripts and notebooks

Scripts (.py) and notebooks (.ipynb) are not the same

Scripts are plain-text files

- It’s just Python and comments
- Can only be run as a whole

Notebooks are interactive computational environments

- Supports code blocks, Markdown and embedded plots & videos.

Feel free to use whichever you prefer

PyCharm (natively in Professional, with a plug-in in Community Edition) support a thing in between, where .py scripts can be executed in blocks

- This makes life quite a bit easier, and we will install it later

Environments

Why?

You work in projects and both Python and installed packages have many versions

- `ssc install package_name`
- `install.packages(package_name)`
- `conda install package_name` or `pip install package_name`

For replicability (and to avoid breaking your own code), it is important to keep these fixed!

What is an environment

An environment contains information regarding

- Which version of Python is used
- What packages are used
  - and what versions of these packages

As such, the only variant thing is now system-wide settings, such as your other software and hardware. With high probability not important for you, but if you ever need to deal with this, look into Docker

With PyCharm

Somewhat automated

In essence the same workflow as without PyCharm, but each environment is associated with a project[1]

- PyCharm automatically remembers your environment, so less work for you!

PyCharm has a introduction to environments with conda

Anaconda has a introduction to environments with PyCharm

1. To share an environment, open the environment using the Anaconda Prompt, export the environment to YAML and create a new project from a folder with this file present in it (change the environment name in the YAML file [top and bottom]). See next section for an introduction

Without PyCharm

(to be skipped, for the curious)

Creating an environment

In the Anaconda Prompt:

Creating an environment

```
conda create -n my_env
```

Creating an environment with a specific Python version (e.g. 3.10)

```
conda create -n my_env python=3.10
```

Activating an environment

To activate an environment

```
conda activate my_env
```

That's it!

To deactivate an active environment

```
conda deactivate
```

If you forget your environment's name, `conda info --env` lists all environments

Workflow

1. When starting a new project, create an environment
2. When working on a project, activate the environment before launching your IDE or Python

Sharing an environment

Someone else needs to work on the project, what to do?

While the environment is active, export the environment to a YAML file

```
conda env export > filename.yml
```

Create an environment from the YAML file

```
conda env create -f filename.yml
```

Exercises

Ex. 1: Make sure you can `print('hello world')` in PyCharm

- PyCharm has a 'get started' guide
- Anaconda has an introduction to PyCharm

Ex. 2: Install the PyCharm Cell Mode plugin

- I found this guide helpful
- Assert that you can run only part of your code

Ex. 3: Create two projects with different environments

- Install a package (e.g. the flask package) in one of the projects
- Import it (e.g. `import flask`) and succeed
- Switch projects (and thus environment)
- Import it (e.g. `import flask`) and fail

Ex. 4: (Optional) An important part of coding is version control. When changing workflow and program, the cost of implementing it is at the lowest

- If you know version control and Git:
  - set Git up with PyCharm. Note that PyCharm has a tutorial
- If you do not know what version control is:
  - What is version control?
  - Think of a time where version control could have helped you. What happened?
  - What is Git?
  - What's the difference between Git and GitHub?
  - Consider whether you should use version control.
    - If you should, set Git up with PyCharm. Note that PyCharm has a tutorial

References

Athey, S., & Imbens, G. W. (2019). Machine learning methods that economists should know about. Annual Review of Economics, 11, 685-725.

Athey, S., Mobius, M., & Pal, J. (2021). The impact of aggregators on internet news consumption (No. w28746). National Bureau of Economic Research.

Bjerre-Nielsen, A., Kassarnig, V., Lassen, D. D., & Lehmann, S. (2021). Task-specific information outperforms surveillance-style big data in predictive analytics. Proceedings of the National Academy of Sciences, 118(14), e2020258118.

Blei, D. M. (2012). Probabilistic topic models. Communications of the ACM, 55(4), 77-84.

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical science, 16(3), 199-231.

Kleinberg, J., Ludwig, J., Mullainathan, S., & Obermeyer, Z. (2015). Prediction policy problems. American Economic Review, 105(5), 491-95.

Mullainathan, S., & Spiess, J. (2017). Machine learning: an applied econometric approach. Journal of Economic Perspectives, 31(2), 87-106.

Nelson, L. K. (2020). Computational grounded theory: A methodological framework. Sociological Methods & Research, 49(1), 3-42.

Salganik, M. J. (2019). Bit by bit: Social research in the digital age. Princeton University Press.

Verhagen, M. D. (2022). A pragmatist's guide to using prediction in the social sciences. Socius, 8, 23780231221081702.