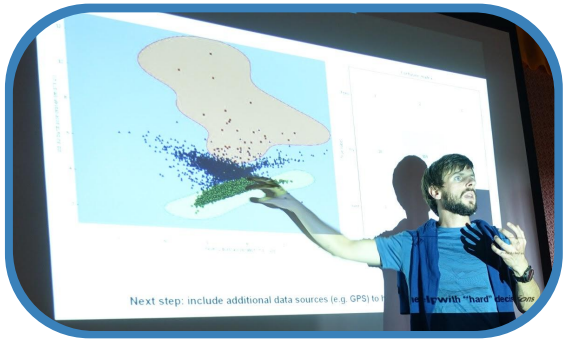


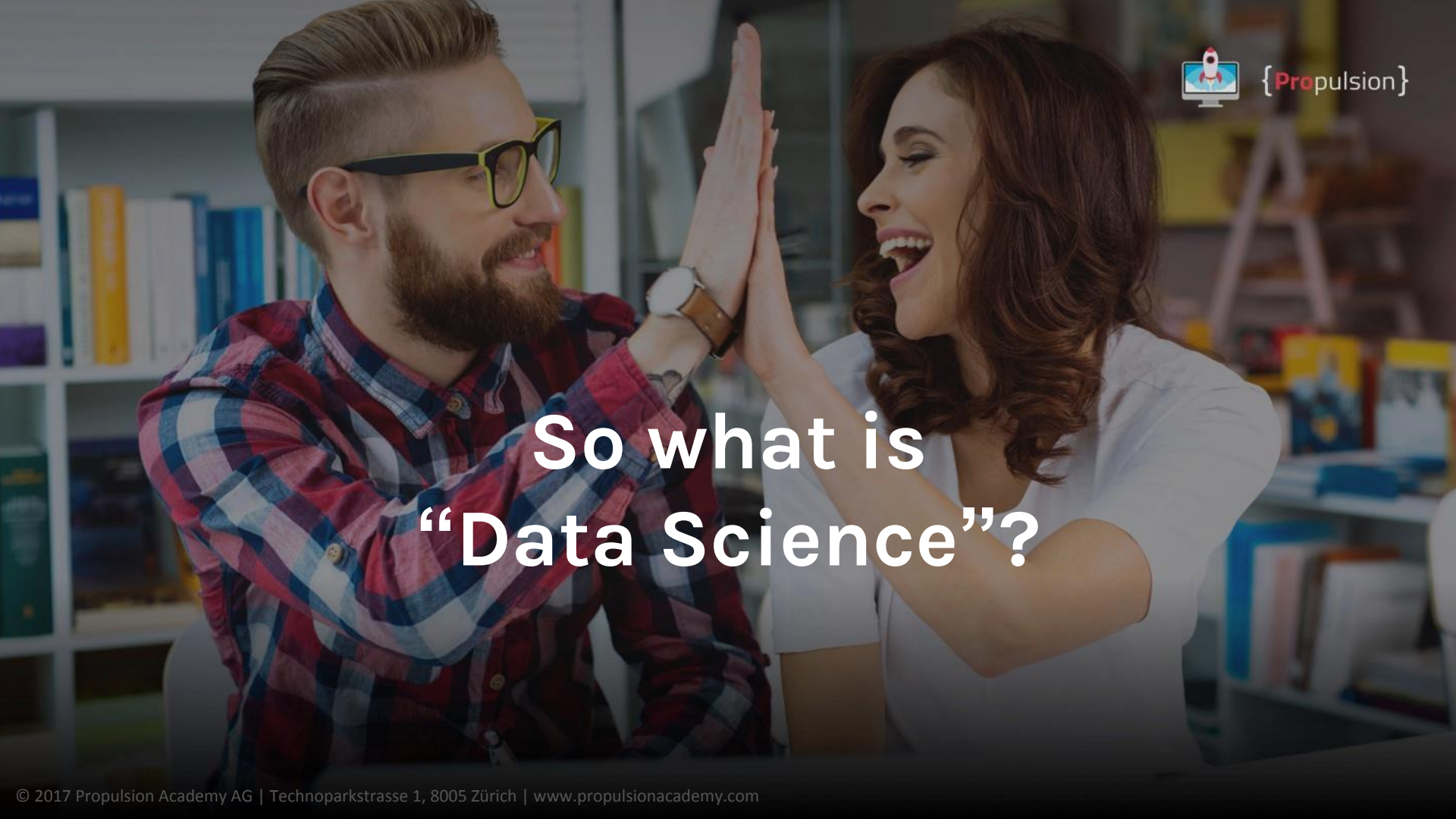
HELLO!

Nice to meet you. I'm Nicolas. Let's start with an introduction:

- What is your story?
- What do you enjoy doing in your job?
- What would you like to do more?
- Which tools/frameworks are you familiar with?
- What do you expect from the course?

Course plan for Week 1

- **Day 1**: intro to data science, discussions
  Hands-on: shell, version control
- **Day 2**: Python for data science
  Hands-on: programming exercises
- **Day 3**: NumPy & Pandas
  Hands-on: arrays, DataFrames, CSV files
- **Day 4**: Relational databases, web scraping
  Hands-on: SQL, Beautiful Soup
- **Day 5**: APIs, mini project
  Hands-on: JSON files, Twitter API, Google Maps API



The mind in the machine: Demis Hassabis on artificial intelligence

{Propulsion}

The problem is that these challenges are so complex that even the world's top scientists, clinicians and engineers can struggle to master all the intricacies necessary to make the breakthroughs required. It has been said that Leonardo da Vinci was perhaps the last person to have lived who understood the entire breadth of knowledge of their age. Since then we've had to specialise, and today it takes a lifetime to completely master even a single field such as astrophysics or quantum mechanics.

The systems we now seek to understand are underpinned by a vast amount of data, usually highly dynamic, non-linear and with emergent properties that make it incredibly hard to find the structure and connections to reveal the insights hidden therein. Kepler and Newton could write equations to describe the motion of planets and objects on Earth, but few of today's problems can be reduced down to a simple set of elegant and compact formulae.

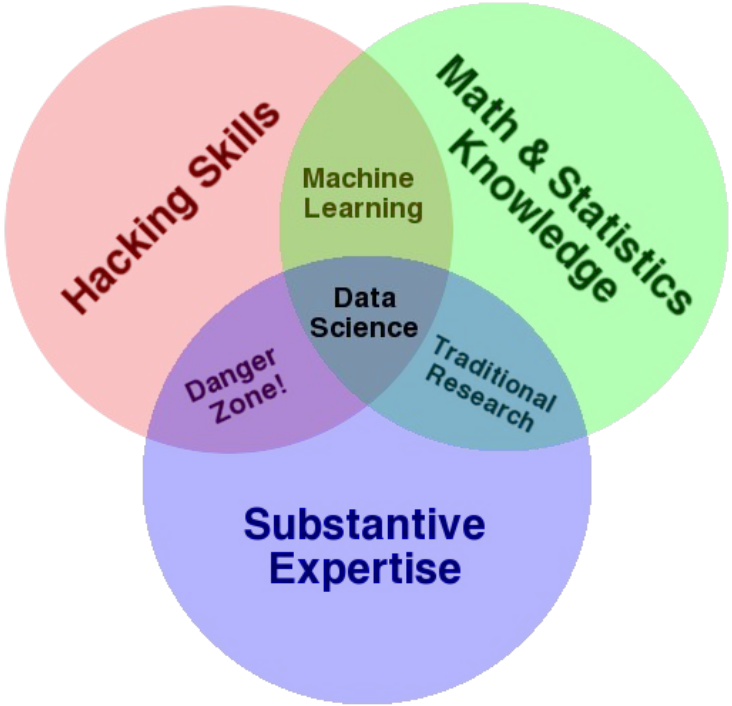
This is one of the greatest scientific challenges of our times. The founding fathers of the modern computer age — Alan Turing, John von Neumann, Claude Shannon — all understood the central importance of information theory, and today we have come to realise that almost everything can either be thought of or expressed in this paradigm. This is most evident in bioinformatics, where the genome is effectively a gigantic information coding schema. I believe that, one day, information will come to be viewed as being as fundamental as energy and matter.

Demis Hassabis

At its core, intelligence can be viewed as a process that converts unstructured information into useful and actionable knowledge. The scientific promise of artificial intelligence (AI), to which I have devoted my life's work, is that we may be able to synthesise, automate and optimise that process, using technology as a tool to help us acquire rapid new knowledge in fields that would remain intractable for humans unaided.

Getting the buzzwords right

- Data Science
- Machine Learning
- Artificial Intelligence
- Big Data

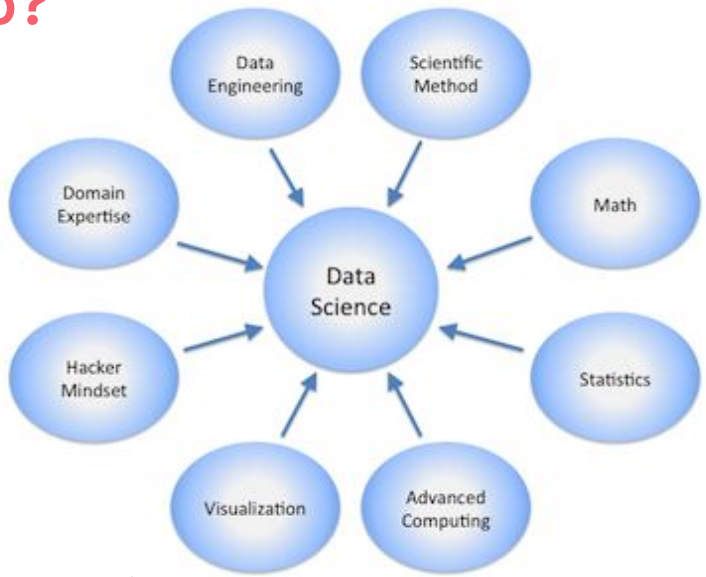


How do these all differ?

So what does a data scientist do?

“4 pillars”:

- Business domain
- Statistics and probability
- Computer science and programming
- Written and oral communication



Data scientist: person strong on *several* of these pillars who can leverage existing data sources and create new ones to extract meaningful information and actionable insights.

Examples of data science problems/deliverables

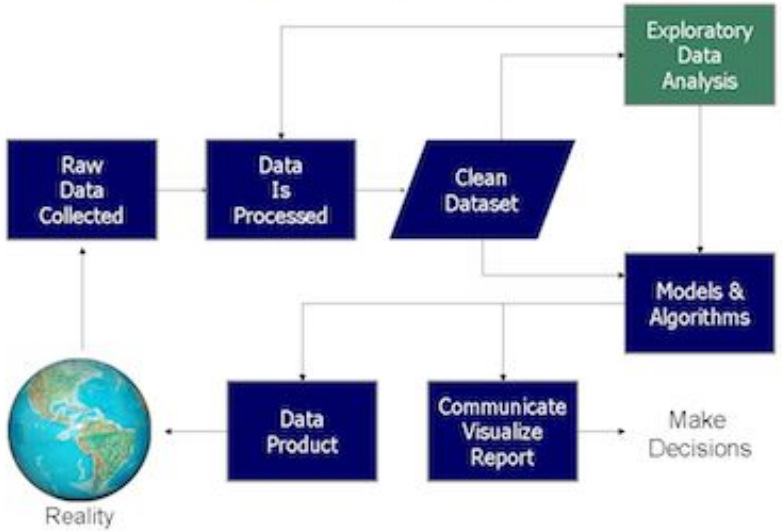
{Propulsion}

- Prediction (predict a value based on inputs)
- Classification (e.g., spam or not spam)
- Recommendations (e.g., Amazon and Netflix recommendations)
- Pattern detection and grouping (e.g., classification without known classes)
- Anomaly detection (e.g., fraud detection)
- Recognition (image, text, audio, video, facial, …)

- Actionable insights (via dashboards, reports, visualizations, …)
- Automated processes and decision-making (e.g., credit card approval)
- Scoring and ranking (e.g., FICO score)
- Segmentation (e.g., demographic-based marketing)
- Optimization (e.g., risk management)
- Forecasts (e.g., sales and revenue)

The data science process

- Data acquisition, collection, and storage
- Discovery and goal identification (ask the right questions)
- Access, ingest, and integrate data
- Processing and cleaning data (munging/wrangling)
- Initial data investigation and exploratory data analysis
- Choosing one or more potential models and algorithms
- Apply data science methods and techniques (e.g. ML)
- Measuring and improving results (validation and tuning)
- Delivering, communicating, and/or presenting final results
- Business decisions and/or changes are made based on the results
- Repeat the process to solve a new problem



The “science” in data science

Data science is the application of the scientific method to problems that can be solved through the analysis and modelling of data.

Workflow

- Ask a question and/or define a problem
- Collect and leverage data to come up with answers/solutions
- Test your solutions to see if the problem is solved
- Iterate as needed, and finalise the solution.

Data scientist vs data analyst

Data analysts do:

- Access and query (e.g., SQL) different data sources
- Process and clean data
- Summarise data
- Understand and use some statistics and mathematical techniques
- Prepare data visualisations and reports

Data analysts are not:

- Programmers
- Responsible for modelling (statistical/machine learning)
- Expected to generate the questions themselves
- Expected to come up with new solutions

Tools of data analysts

- Excel
- Tableau
- SAS
- SPSS
- Qlik

Data scientist vs data engineer

Data engineers are responsible for:

- Extract, Transform, and Load (ETL) pipelines on production sources
- Data architecture (data “lakes”)
- Data flow
- Data storage and warehousing
- Data infrastructure
- Scalability, reliability, availability, backups, …
- (sometimes) DevOps

Data engineers don’t (usually) do:

- Statistics
- Analytics and modelling
- Question-driven investigation

Data engineering tools/frameworks

- Hadoop, Spark
- SQL
- NoSQL
- Shell
- Docker
- …

Can you think of a complementary workflow in an organisation comprising data scientists, data engineers, and data analysts?

The data science toolbox

Languages

- Python
- R
- SQL
- Scala, Julia, Java, ...

Analysis/modelling libraries

- Jupyter
- NumPy
- Pandas
- Scikit-learn
- Tensorflow

Visualisation libraries

- Matplotlib
- ggplot2
- D3.js
- plot.ly

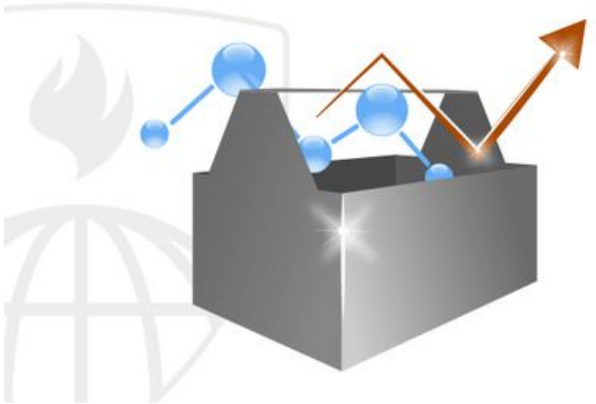
Big data frameworks

- Hadoop, Spark

Database systems

- Postgres, MySQL, SQLite
- MongoDB, Cassandra
- Neo4j, GraphBase

Version control tools, ...



Group work: “thinking like a data scientist”

Industry problems

- Online shop: “How should we recommend similar products to our users?”
- Media company: “How should we decide which shows to produce?”
- Real estate agency: “How should we advise a seller on the best price for their house?”
- Bank: “How should we detect fraudulent account activity?”

Questions to ask

- Which data sources will I leverage?
- Which variables will I extract?
- Which methods will I use?
- How will I test my results?
- What may I miss?
- How will I productionise my solution?
- How will it scale?

Time to go hands-on

{Propulsion}

Getting familiar with our environment

- Using a terminal/shell environment
- Installing Python, pip, Jupyter
- Setting up virtualenvs

Tutorials & exercises

- Bash
- git
- Python
- NumPy
- Pandas

Getting familar with UNIX and Bash

Interacting with a computer through the shell

- Logging securely to a remote workstation (ssh)
- Navigating the file system (cd, ls)
- Manipulating files (cp, mv)
- Understanding file permissions (chmod, chown)
- Calling a script

Bash (tiny.cc/propacad-ds-bash). Online terminal: tiny.cc/propacad-ds-cmd

- Shebang (#!/bin/sh)
- $PATH
- Writing simple ‘glue’ programs

Exercises

- SSH into test.rebex.net as demo/password
- Write script mixing two input strings (‘abc’ and ‘def’ -> ‘adbecf’)

Version control: the end of script_v3_final_WORKING.py

How do we keep track of:

- File versions
- Ownership of changes
- Software versions (branches)

Basic principles at tiny.cc/propacad-ds-vcs.
Let's dive into it at try.github.io (advanced: tiny.cc/propacad-ds-git).