Data & Things

(Spring 26)

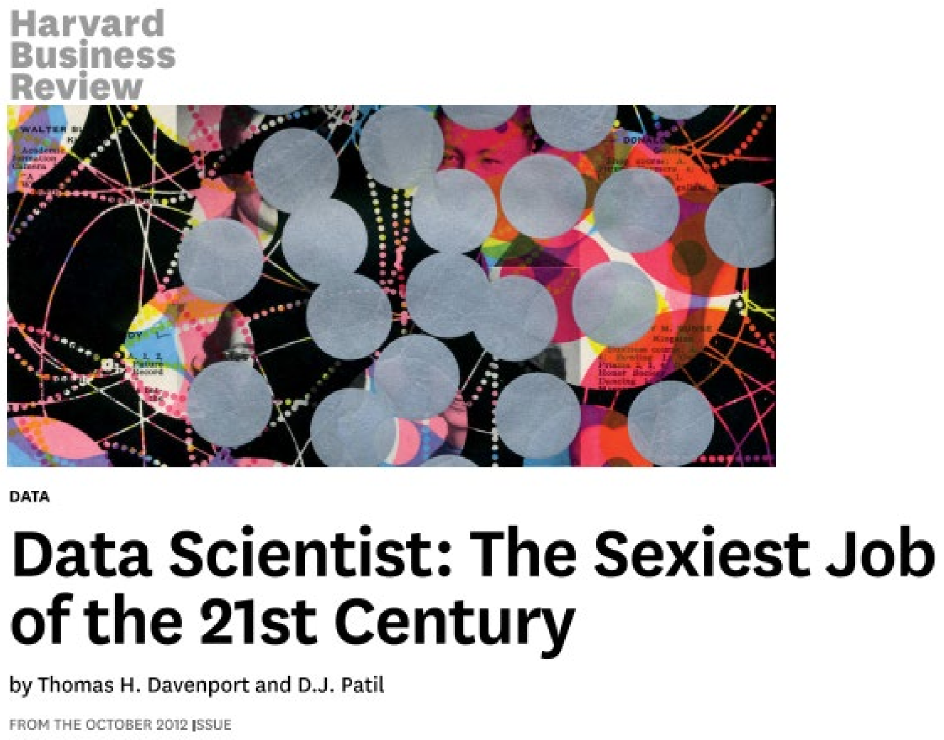
Monday February 2

Lecture 1: Introduction to the course, data science, and Python for data science

Jens Ulrik Hansen

Roskilde Universitet

Data & Things – spring 26


Harvard Business Review

DATA

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

The Economist
The world's most valuable resource
Data and the new rules of competition
Google

https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data

Roskilde Universitet

Data & Things – spring 26

Dall-E, 2025

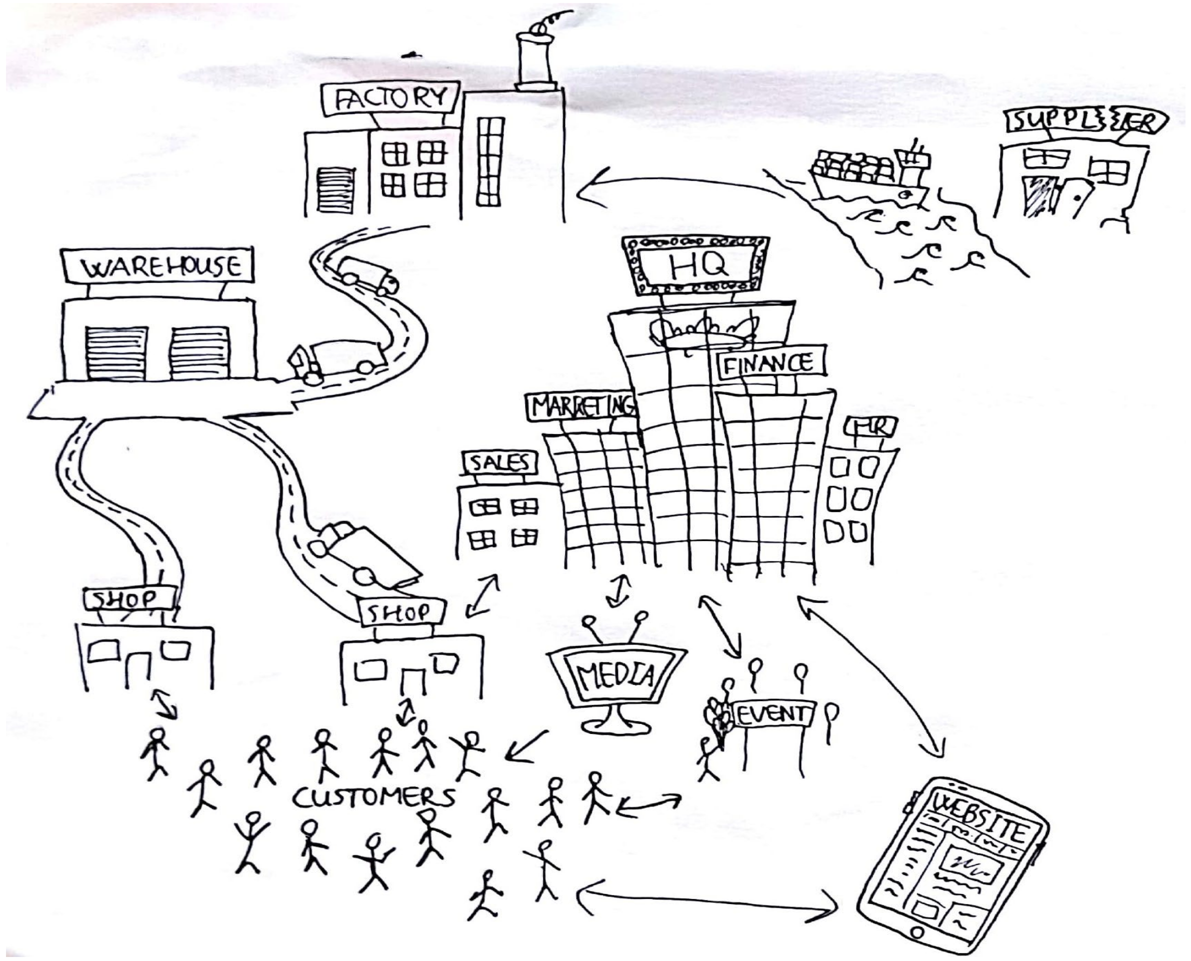
Examples of data centric applications

- Find out which website design make most visitors subscribe to a service (A/B testing)
- Predicting when a factory machine will break down or when a product will be returned (Predictive maintenances)
- Predicting when a customer will cancel a subscription or when an employee will quit (Churn analysis, People Analytics)
- Segmenting customers into groups to target individually (Clustering)
- Predicting if an image of a skin lesion shows sign of skin cancer (Image recognition)
- Learning the seasonality in a company’s sales data (time series analysis)
- Generate new text from an input text (Generative AI)

Roskilde Universitet

Data & Things – spring 26

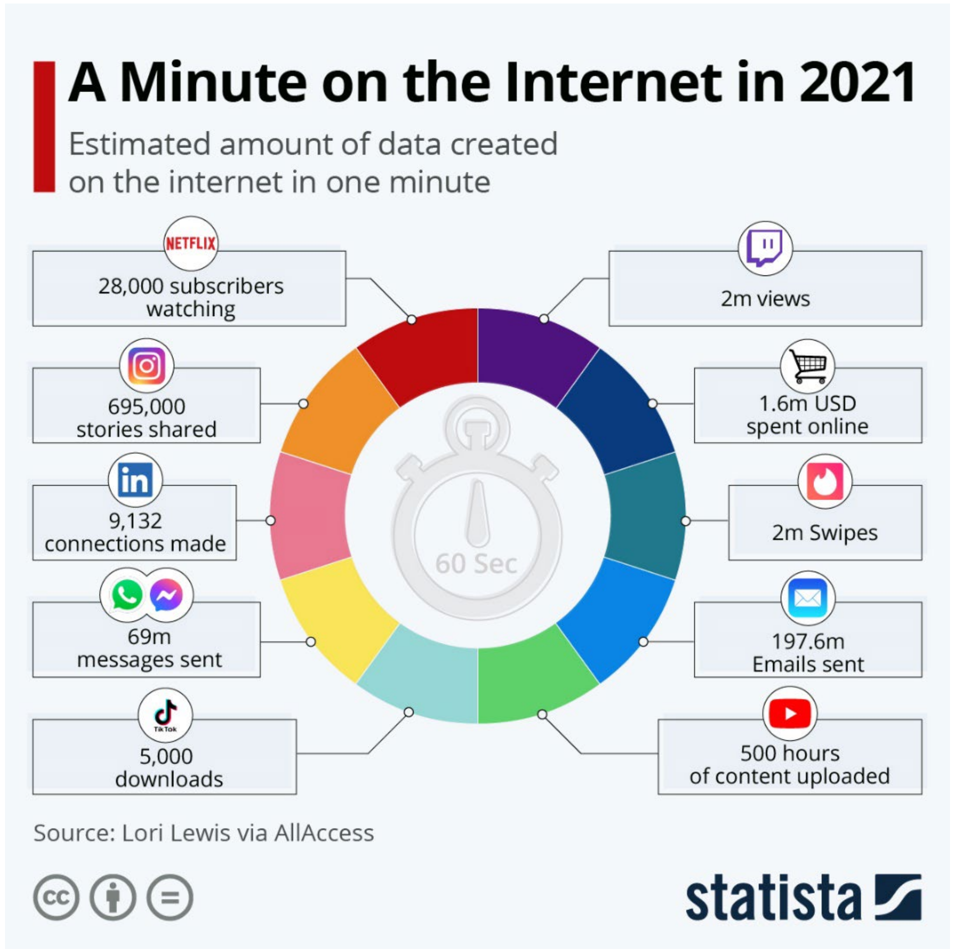
- Every aspect of a business can be datafied….
- It helps optimize business processes, increase business understanding, develop better products, and reaching the right costumers



Roskilde Universitet

Data & Things – spring 26

- Why now? Where does all the data come from?
  - Scientific instruments
  - The internet (and social media)
  - Smartphones
  - Internet of Things (IoT)



Roskilde Universitet

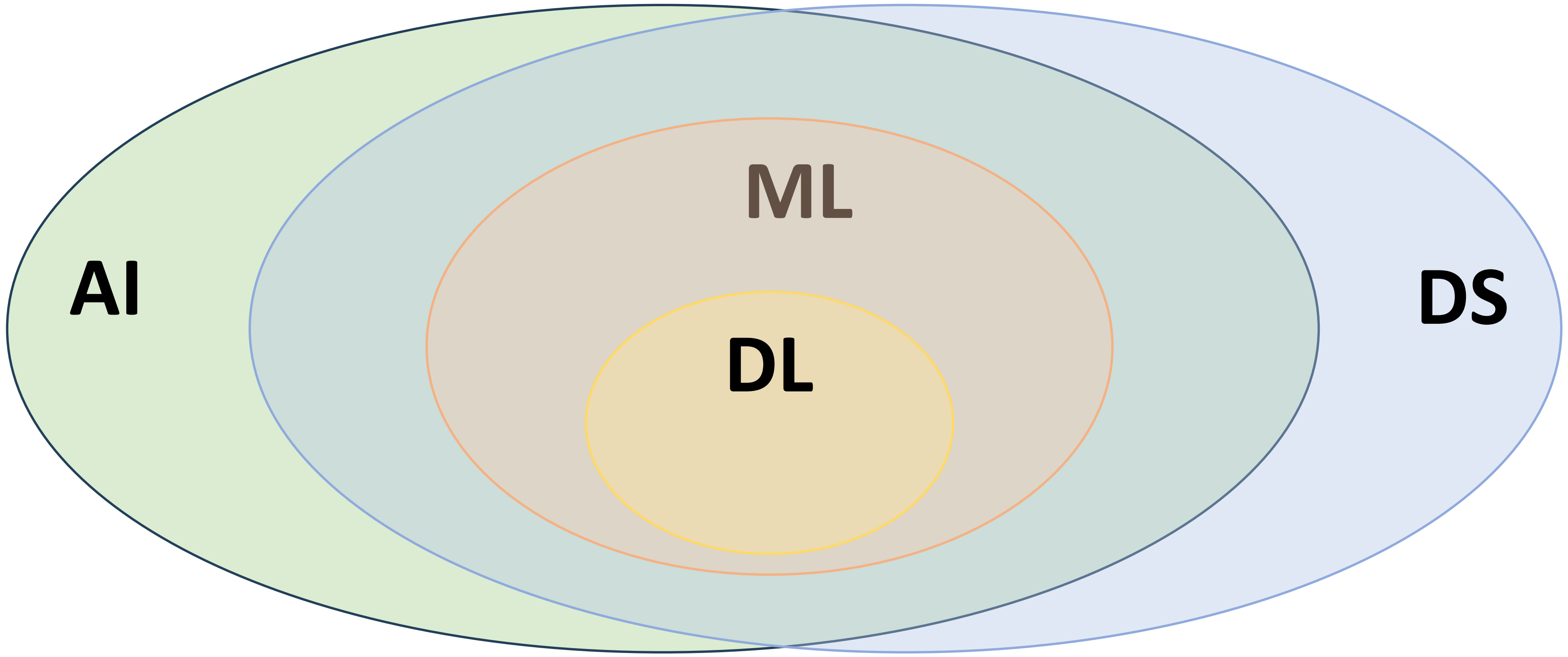
Data & Things – spring 26

- **Getting under the hood**
  - How do we collect, store and manage the necessary data?
  - How do transform data to be useful?
  - How do we analyze data to create insights?
  - How do we build predictive (machine learning) models?
  - How do we build and manage data centric application?
  - What are the ethical and societal consequence of the data centric applications?

- All of these questions are the focus of this course, with a strong focus on analyzing data to create insights and building predictive (machine learning) models

Data & Things – spring 26

- AI: Artificial Intelligence
- DS: Data Science
- ML: Machine Learning
- DL: Deep Learning



Roskilde Universitet

Outline of this lecture

- Introduction to the course
- Course practicalities and the exam
- Introduction to Data Science (and Data Engineering)
- Brief introduction to Python (and Jupyter Notebook)
- Essential data science packages in Python for data science: (NumPy) and Pandas
- Examples on real data
- Exercises

Roskilde Universitet

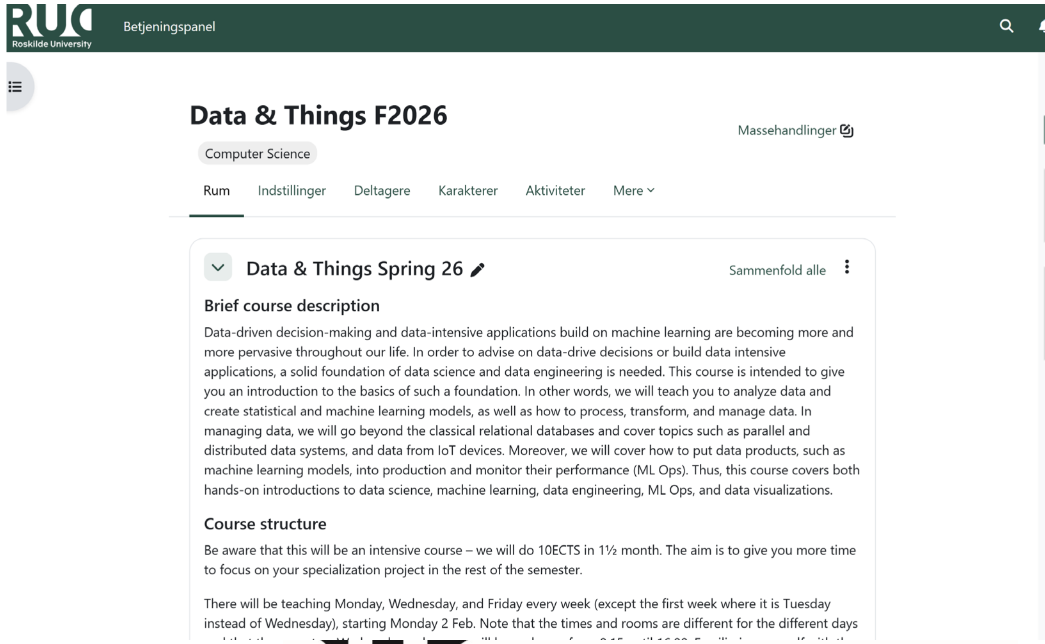
Course practicalities

- **We expect you to attend all classes and show up on time!**
  - The course is planed under the expectation that you show up for all classes
  - It is your own responsibility if you choose not to show up for the classes!
- **Note the complex and condensed schedule – see study.ruc.dk**
  - Teaching will be on Monday, Wednesday, and Friday (except tomorrow)
  - Note the times differs for the different days and the different weeks!
  - Note also to room differs from time to time, so always check study.ruc.dk well ahead of time!!!
- **Expected workload**
  - A full-time study and no other study activities simultaneously with Data & Things

• Total hours (from study.ruc.dk):	270 hours
• Classes (18 * 4 hours + Q&A session):	76 hours
• Preparation for classes (approx. 7 hours pr class):	140 hours
• Exam and preparation before the course:	54 hours

Course practicalities

- **Course website**
  - Course material etc. all on the course moodle page: https://moodle.ruc.dk/course/view.php?id=25120
  - Most of the overall topics are there already
  - Readings for the classes will be posted continuously (most of it is there already)
    - Some of the later topics will be updated as we go along – so keep an eye out for updates!
  - Updated room information on: https://study.ruc.dk/class/view/38286



Data & Things F2026

Data & Things Spring 26

Brief course description

Data-driven decision-making and data-intensive applications build on machine learning are becoming more and more pervasive throughout our life. In order to advise on data-drive decisions or build data intensive applications, a solid foundation of data science and data engineering is needed. This course is intended to give you an introduction to the basics of such a foundation. In other words, we will teach you to analyze data and create statistical and machine learning models, as well as how to process, transform, and manage data. In managing data, we will go beyond the classical relational databases and cover topics such as parallel and distributed data systems, and data from IoT devices. Moreover, we will cover how to put data products, such as machine learning models, into production and monitor their performance (ML Ops). Thus, this course covers both hands-on introductions to data science, machine learning, data engineering, ML Ops, and data visualizations.

Course structure

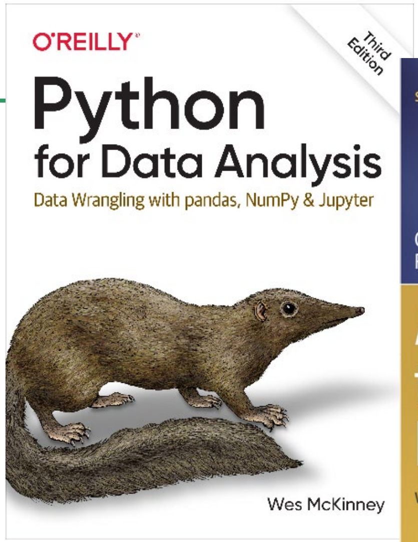
Be aware that this will be an intensive course – we will do 10ECTS in 1½ month. The aim is to give you more time to focus on your specialization project in the rest of the semester.

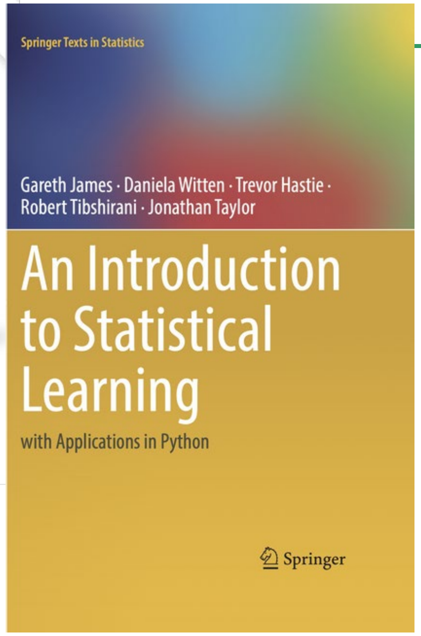
There will be teaching Monday, Wednesday, and Friday every week (except the first week where it is Tuesday instead of Wednesday), starting Monday 2 Feb. Note that the times and rooms are different for the different days

Roskilde Universitet

Course practicalities

- **Course material**
  - Different books, papers and websites…
    - We have tried to find materials that are freely available online or available online through the RUC library.
  - Books we will use:
    - *Python for Data Analysis,* 3rd edition (2022) by Wes McKinney (O'Reilly): https://wesmckinney.com/book/
    - *An Introduction to Statistical Learning – with Applications in Python* by James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J., (2023), Springer (freely downloadable from: https://www.statlearning.com/)
    - *Deep Learning with Python*, 3rd edition (2025), Manning Publishing by François Chollet and Matthew Watson: https://deeplearningwithpython.io/chapters/
      - We will only use the first 4 chapters, approx. 120 pages. Used by the deep learning course as well
    - … Part of other books, research papers, package documentation, blogs, etc.




Springer Texts in Statistics
Gareth James · Daniela Witten · Trevor Hastie · Robert Tibshirani · Jonathan Taylor
An Introduction to Statistical Learning
with Applications in Python
Springer

DEEP LEARNING with Python
THIRD EDITION
François Chollet
Matthew Watson
MANNING
READ ONLINE
BUY THE BOOK
Perfect for anyone interested in learning by doing from one of the industry greats.
— Anthony Goldbloom, Founder of Kaggle

Roskilde Universitet

Course practicalities

- **<u>Prerequisite</u>**
  - We do not assume any knowledge of Python or Data Science, however…
    - We assume basic math skills (corresponding to level B of high school)
    - We assume that you, as a graduate student in computer science, can install Python and Jupyter Notebooks on your own and familiarize yourself with the very basics of Python – we will teach you how to use Python to do Data Science
    - Clearly if you have already taken the elective course on Data Science and Visualization on the bachelor level, have done semester projects about data science, or taken general NatBach courses in Python and mathematics, you are likely to find the course easier – that is how it is with the RUC structure!
      - We do not assume any of these prerequisite!

Course practicalities

- **The lecturers**
  - Jens Ulrik Hansen
    - Associate Professor of Computer Science, RUC
    - jensuh@ruc.dk
  - Shabab Akhter
    - People & Product Manager - Data & AI CoE, ROCKWOOL Group
    - shabab@ruc.dk
  - Frederik Møller Henriksen
    - Post.Doc at the Department of Communication and Arts, RUC
    - frmohe@ruc.dk

Roskilde Universitet

Course practicalities

- **The classes**

Lect.	Content	Lect.	Date
1	Intro. to course, data science, and python	Jens	Feb 2
2	Data transformation and exploratory data analysis	Jens	Feb 3
3	Statistics	Jens	Feb 6
4	Data Engineering	Shabab	Feb 9
5	Regression	Jens	Feb 11
6	Time series	Shabab	Feb 11
7	Classification I	Jens	Feb 13
8	Classification II	Jens	Feb 16
9	IoT and sensor data	Shabab	Feb 18
10	Clustering	Jens	Feb 20

Lect.	Content	Lect.	Date
11	Machine Learning Operations (MLOps)	Shabab	Feb 23
12	Introduction UCloud	Jakub/ Jens	Feb 25
13	Social Network Analysis and large data	Frederik	Feb 25
14	Neural networks and deep learning I	Jens	Feb 27
15	Neural networks and deep learning II	Jens	Mar 2
16	Generative AI	Shabab	Mar 4
17	Explainability	Shabab	Mar 6
18	Ethical reflections on data science, end of course	Jens	Mar 9
	Exan Q&A	Jens & Shabab	Mar 11

Roskilde Universitet

Course practicalities

- **<u>My classes</u>**
  - I will assume you have looked at the mandatory reading – it is up to you how careful
  - I will briefly review the basic parts and put it in larger contexts, potentially going beyond the reading focusing on central concepts, methods, and theories
  - I will show examples (code) on (real) data
  - You will work with (real) data in exercises for the last 1-2 hours

Roskilde Universitet

Course practicalities

- **The exam**
  - Exam Dates:
    - **Hand-in of written product: Sunday March 15 at 10.00am on eksamen.ruc.dk**
    - **Oral exam: Wednesday March 18, Thursday March 29, and Friday March 20**
  - Format: 20 min individual oral exam based on a written product

Course practicalities

- **<u>The oral exam</u>**
  - Dates: March 18-20
  - There will 20 minutes for each student, including time for assessment and feedback
  - At the beginning of the exam, the student draws a random number. Each number will correspond to an exam topic (14 topics – see exam document on moodle).
  - The student then present on the topic (3-5 min) including potential selected exercises handed in as part of the written product. (There is not enough time to go through all your code!)
  - This is followed by questions about the presentation and the exam question from the examiners (5-10 min).
  - At the end, the examiners might relate their questions to some of other exam topics.
  - For each of the exam topics, the student is expected to know the central concepts, methods, theories, and problems discussed in class and be able to explain and exemplify them. Moreover, for those exam topics where there are selected exercises handed in as part of the written product, the student is expected to be able to explain how they solved the exercises and be able to explain their entire code.

Course practicalities

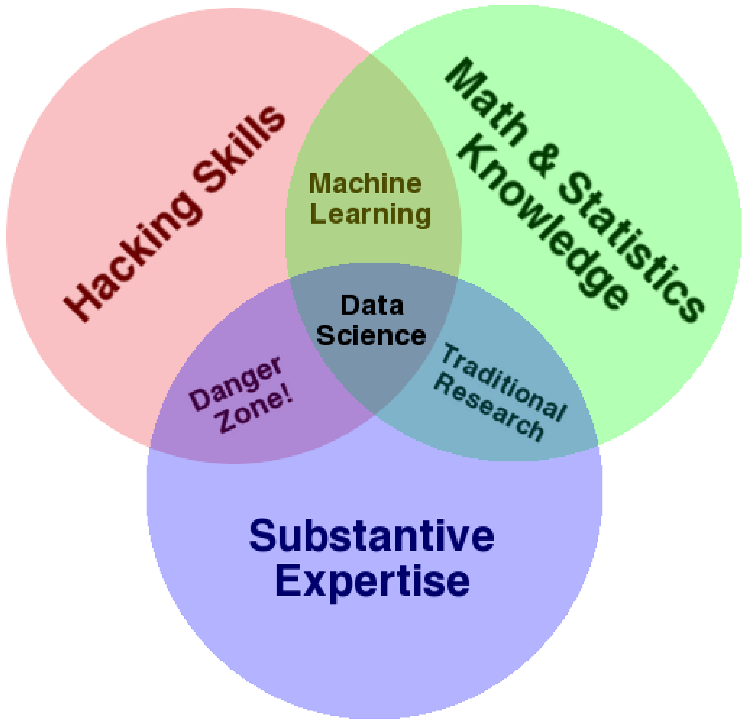
- **The written product**
  - Hand-in date: Sunday, March 15 at 10.00am
  - The written product will consist of answers to selected exercises - these will be selected from those that have already been done in class. Only some of the exam topics will have such selected exercises (most of them will!).
  - The list of selected exercises will be made public on the last day of class (Monday March 9).
  - The handed in answers to the selected exercises should be in the format of a single Jupyter Notebook or a zip file containing a notebook for each of the selected exercises.
  - The students can do the selected exercises in self-made groups and either hand in as a group on Eksamen.ruc.dk, or hand in the notebook(s) individually.

Outline of this lecture

- Introduction to the course
- Course practicalities and the exam
- Introduction to Data Science (and Data Engineering)
- Brief introduction to Python (and Jupyter Notebook)
- Essential data science packages in Python for data science: (NumPy) and Pandas
- Examples on real data
- Exercises

Introduction to Data Science

- Wikipedia
  - *"Data science is an interdisciplinary academic field [1] that uses statistics, scientific computing, scientific methods, processes, algorithms and systems to extract or extrapolate knowledge and insights from noisy, structured and unstructured data.[2] Data science also integrates domain knowledge from the underlying application domain (e.g., natural sciences, information technology, medicine).[3]",* https://en.wikipedia.org/wiki/Data_science, retrieved 2023-01-31
  - "The data science Venn diagram", http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram, retrieved 2023-01-31
  - "A data scientist is someone who knows more statistics than a usual programmer and someone who knows more programming than a usual statistician"
- Me: Using data and scientific methods to solve problems and create product



Roskilde Universitet

Introduction to Data Science

Dall-E, 2025

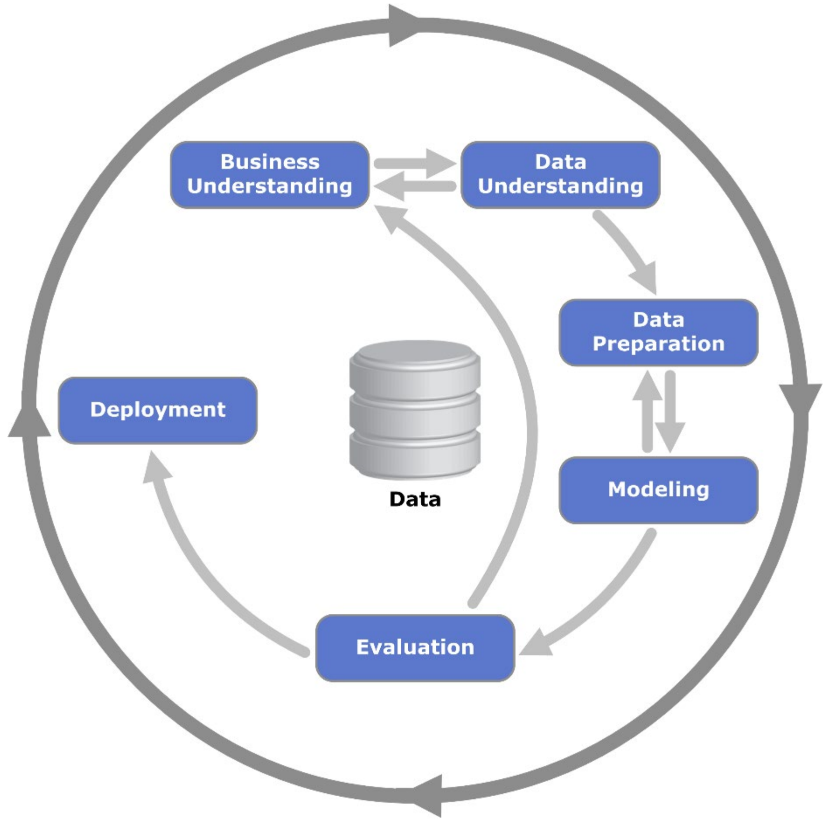
Recall the examples…

- Find out which website design make most visitors subscribe to a service (A/B testing)
- Predicting when a factory machine will break down or when a product will be returned (Predictive maintenances)
- Predicting when a customer will cancel a subscription or when an employee will quit (Churn analysis, People Analytics)
- Segmenting customers into groups to target individually (Clustering)
- Predicting if an image of a skin lesion shows sign of skin cancer (Image recognition)
- Learning the seasonality in a company's sales data (time series analysis)
- Generate new text from an input text (Generative AI)

Roskilde Universitet

Introduction to Data Science

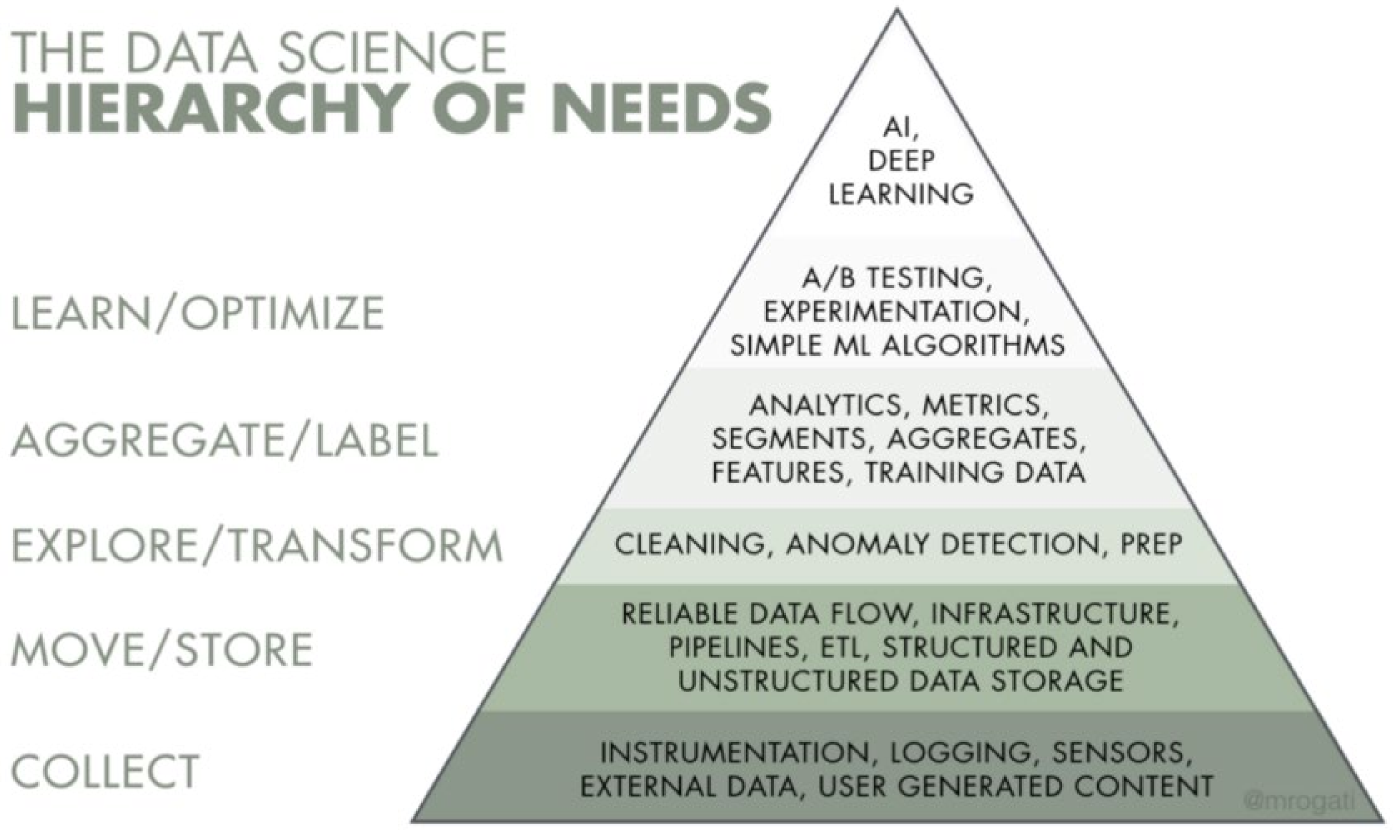
- **The Data Science process**
  - CRISP-DM: Cross-industry standard process for data mining
    - https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining
    - From the late 90'ties, but still used today
  - Data Science folklore: Data preparation/Data transformation/Data Wrangling/Data tidying take up 70% of the time in a data science project



Roskilde Universitet

Introduction to Data Science

- **<u>The pre-requisite for Data Science</u>**
  - Data
  - Data infrastructure
  - … in other words ***"Data Engineering"***
  - *Data literacy* – understanding of relevant data and the problem

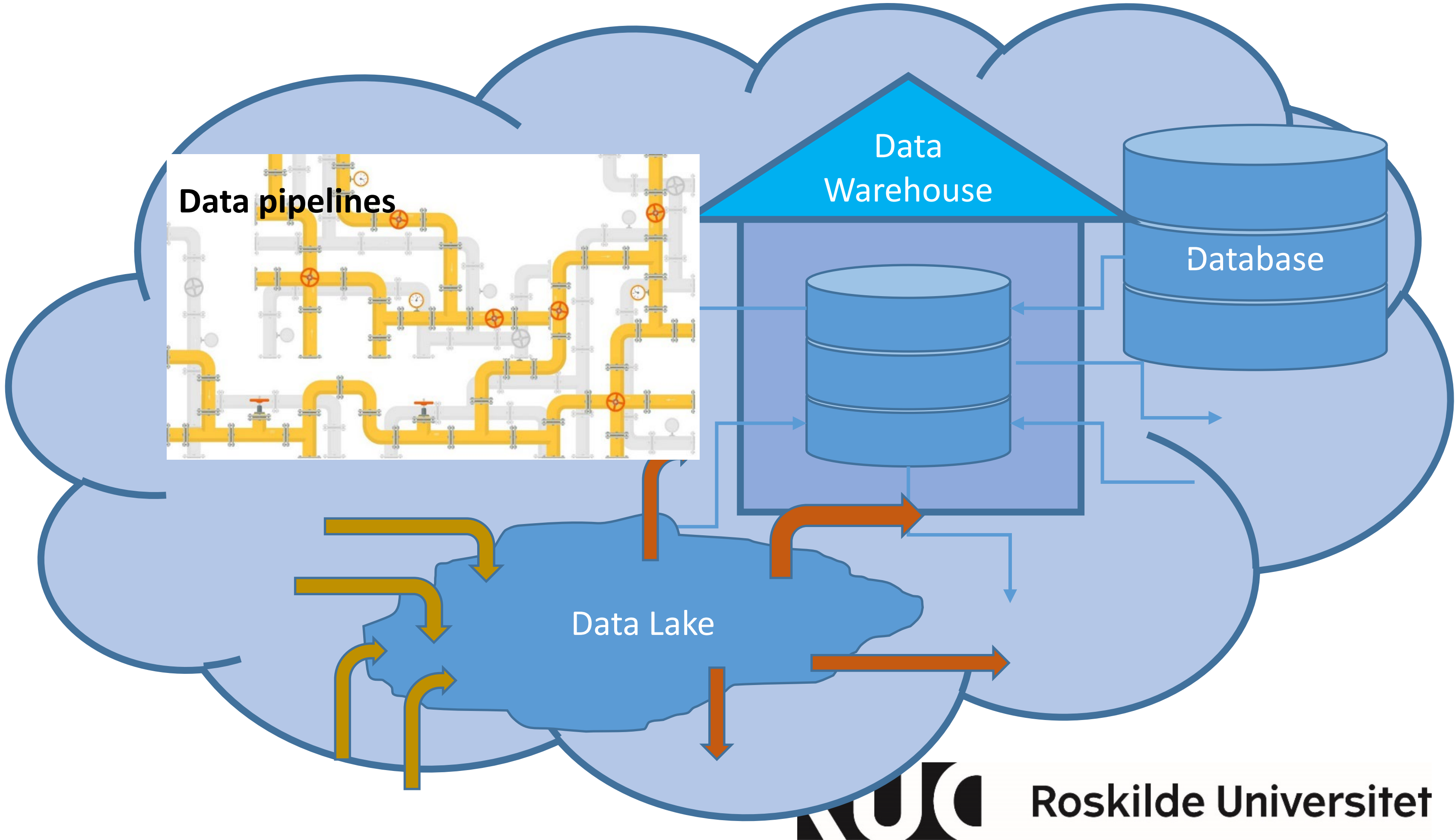

THE DATA SCIENCE
HIERARCHY OF NEEDS
LEARN/OPTIMIZE
AGGREGATE/LABEL
EXPLORE/TRANSFORM
MOVE/STORE
COLLECT
AI, DEEP LEARNING
A/B TESTING, EXPERIMENTATION, SIMPLE ML ALGORITHMS
ANALYTICS, METRICS, SEGMENTS, AGGREGATES, FEATURES, TRAINING DATA
CLEANING, ANOMALY DETECTION, PREP
RELIABLE DATA FLOW, INFRASTRUCTURE, PIPELINES, ETL, STRUCTURED AND UNSTRUCTURED DATA STORAGE
INSTRUMENTATION, LOGGING, SENSORS, EXTERNAL DATA, USER GENERATED CONTENT
@mrogati

Source: Monica Rogati's fantastic Medium post "The AI Hierarchy of Needs"

Introduction to Data Science

- **<u>The metaphors of data infrastructure</u>**
  - Database
  - Data Warehouse
  - Data Lake
  - Data pipelines
  - The could/IaaS

  - ***Data engineering: developing and maintaining this data infrastructure***



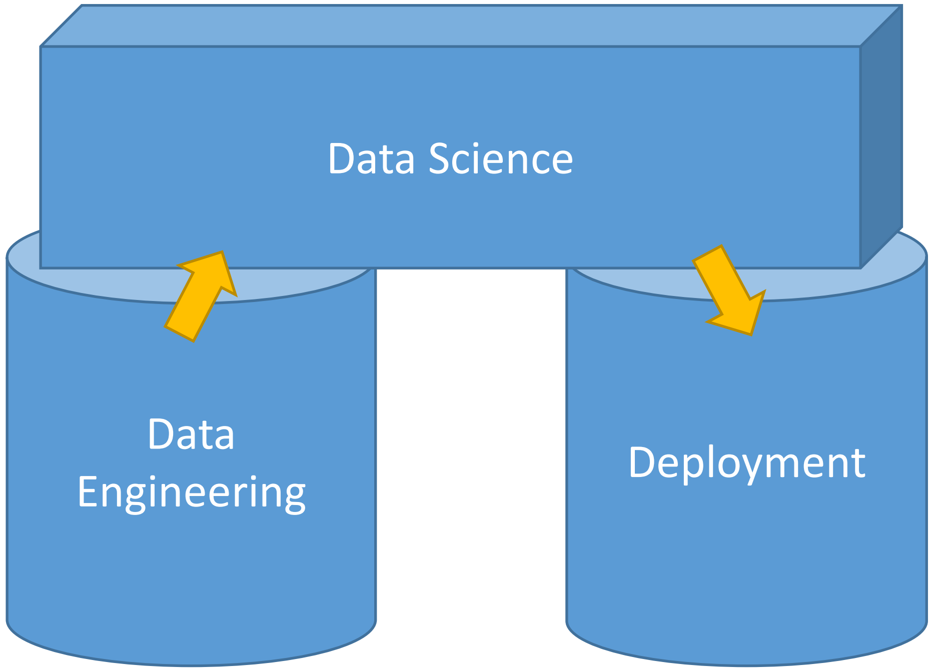
Roskilde Universitet

Introduction to Data Science

- **<u>The output of a data science projects:</u>**
  - A Visualization
  - New data
  - A report
  - Slides
  - A dashboard or interactive web-application
  - An API for a predictive model
  - An application involving AI/ML elements
- **<u>Type of delivery</u>**
  - A one-off production
  - A scheduled job
  - A live service/application

Introduction to Data Science

- Data engineering, Data Science, and Deployment
  - The boundaries are not hard and fixed
    - Data pre-processing can be done both by data scientists or data engineers
    - Data scientists can be involved in creating data science products
  - The deployment part is often referred to as “MLObs” and are gaining more and more focus



Introduction to Data Science

- **How doing data science differs from software development**
  - Similarities
    - They both deal with data and models of the world
    - They both use programming languages
    - If a data science product (like a predictive machine learning model) is embedded in a software application, a lot of software development and CI/CD is needed.

Introduction to Data Science

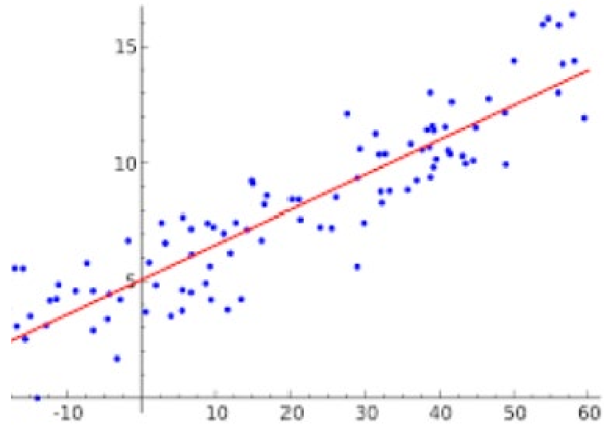
- **Software development**
  - In software development applications, a data model is specified as a model of the world where all relationships as specified in advance. The application then uses this to store, update and retrieve information (about costumers for instance).
  - (Normalized) databases are ideal for this type of dealing with data
  - The data aspect of the application can be **tested** on made up test data

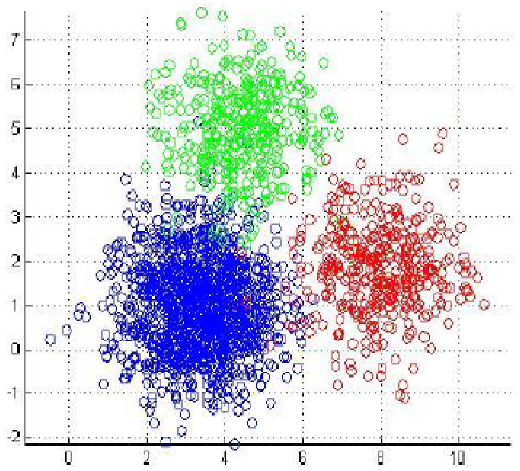
- **Data science**
  - We have no model of the world in advance, and instead of specifying one, we want to learn it from large amount of historical data about the world.
  - We are not doing a lot of read, write, update, so normalized databases are not necessarily and advantage – we might as well have data in an CSV file, as we might just do one process of it to learn.
  - We cannot test how good a machine learning model is on made up test data – we need real test data from the world. Moreover, we need mathematical and statistical theory about machine learning models to be able to decide how good a model we have made.

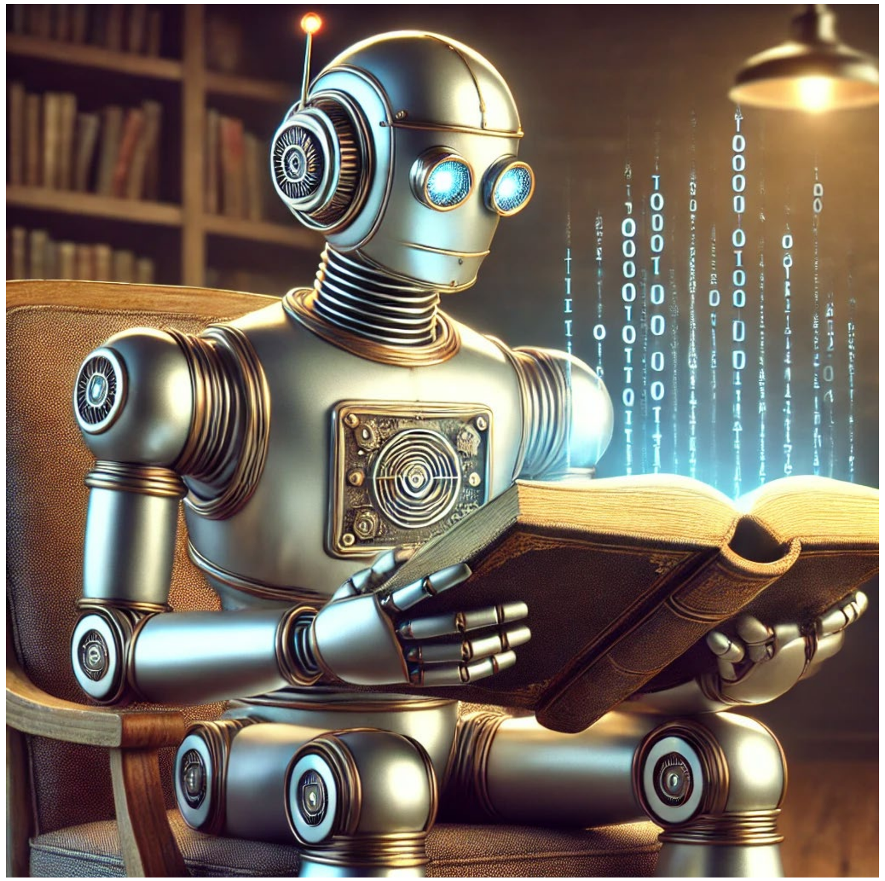
There are important differences in how we model the world, deal with data, and test or models.

Introduction to Data Science

- **Introduction to machine learning**
  - Machine learning is about developing algorithms or systems that can learn from data
  - Machine learning is about detecting/ learning patterns in historical data useful for making predictions about new cases



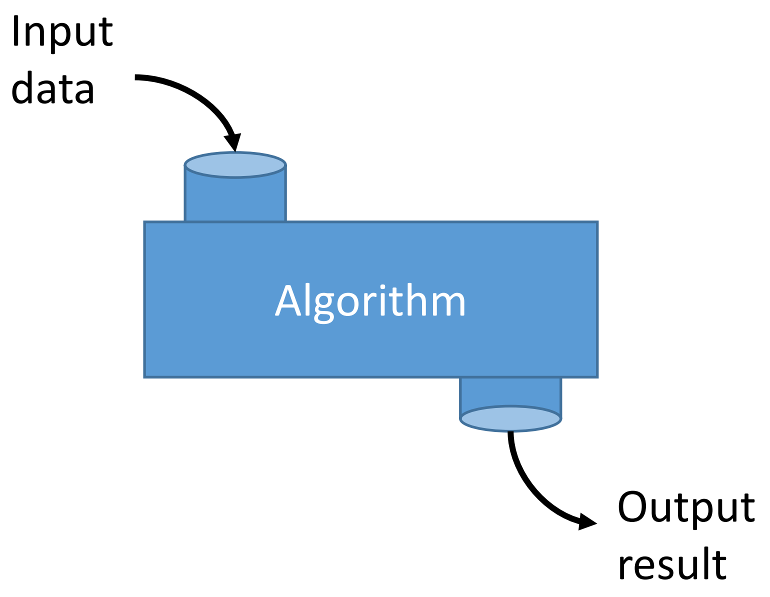




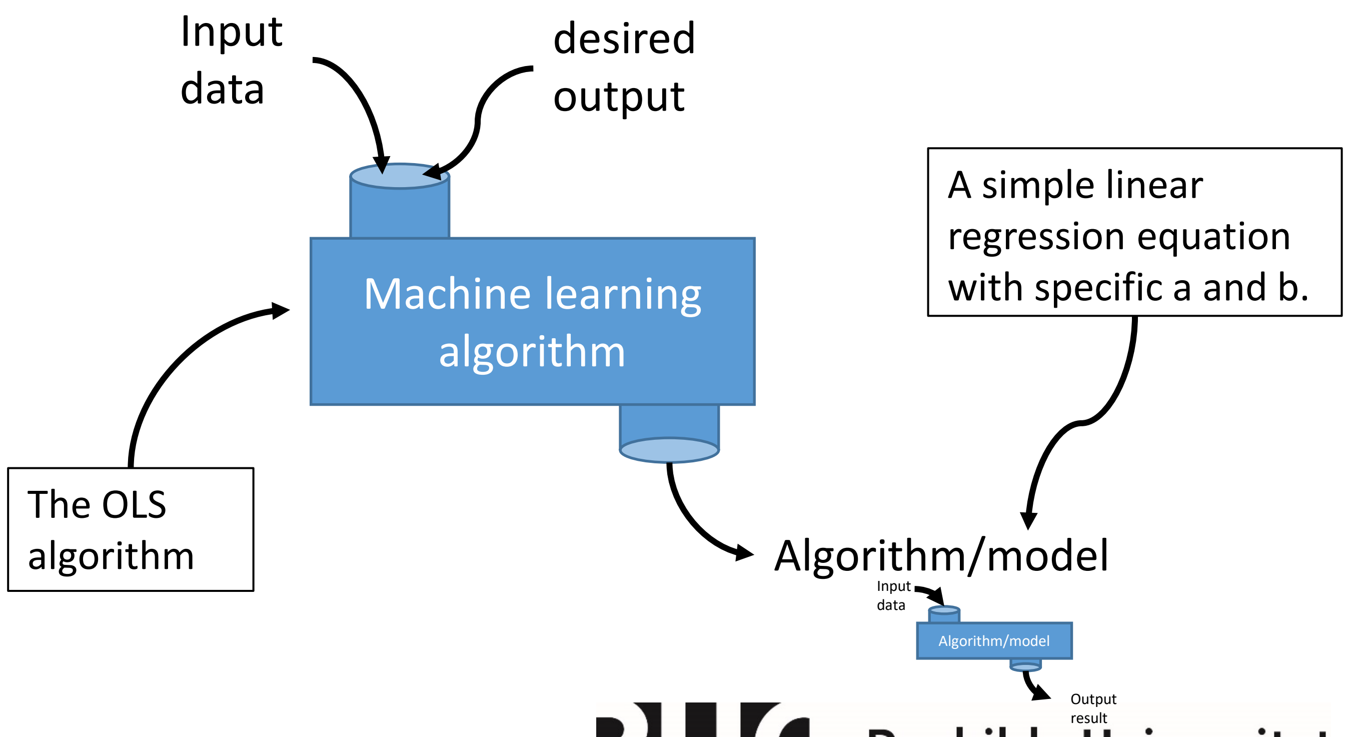
Dall-E, 2025

Introduction to Data Science

- **Traditional programming (and symbolic AI)**
  - Some input is transformed using rules and specification to generate some output



- **Machine learning (supervised)**
  - Given input and desired output, we want to automatically learn an algorithm that generate that output from the input.



Introduction to Data Science

Dall-E, 2025

Examples of data centric applications / Examples of Machine Learning

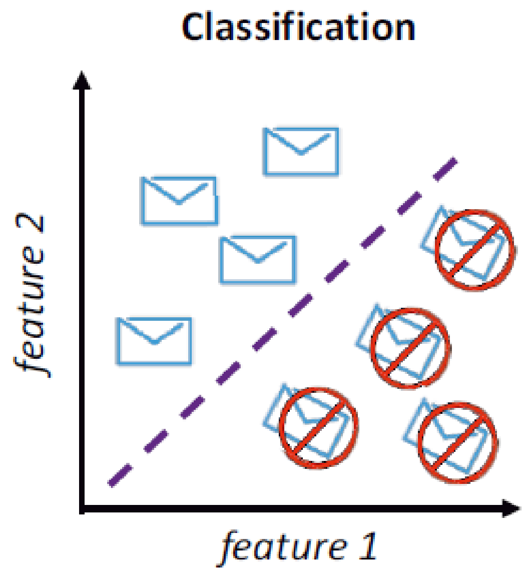
- Find out which website design make most visitors subscribe to a service (A/B testing)
- Predicting when a factory machine will break down or when a product will be returned (Predictive maintenances)
- Predicting when a customer will cancel a subscription or when an employee will quit (Churn analysis, People Analytics)
- Segmenting customers into groups to target individually (Clustering)
- Predicting if an image of a skin lesion shows sign of skin cancer (Image recognition)
- Learning the seasonality in a company’s sales data (time series analysis)
- Recommending new products to buy, new series to watch, or posts to engage with (Recommender systems)

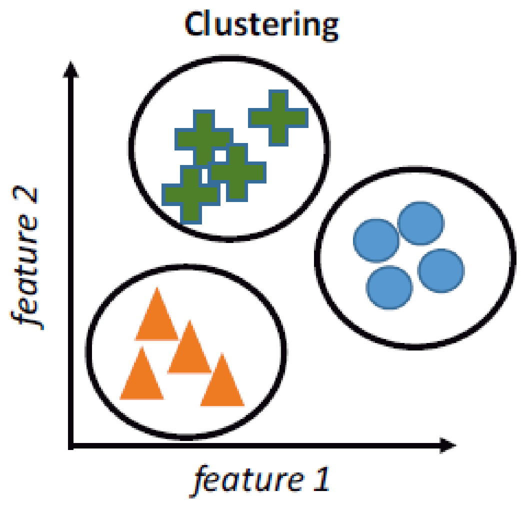
Roskilde Universitet

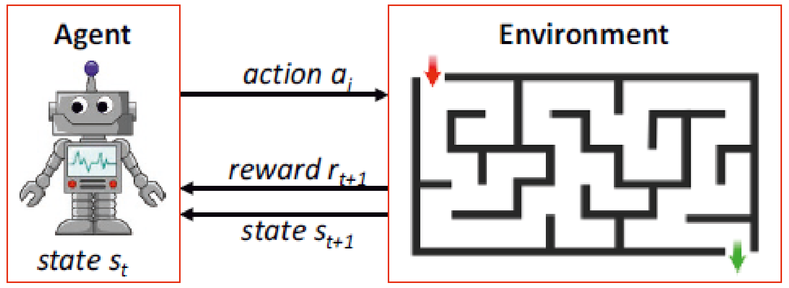
Introduction to Data Science

- **Three classic types of machine learning**

  1) Supervised learning *2) Unsupervised learning* *3) Reinforcement learning*







** Mixing supervised and unsupervised learning or combining them is also becoming more common…*

Roskilde Universitet

Introduction to Data Science

- **Unsupervised learning** find patterns in *unlabeled data*. It works on input data directly.
  - E.g., clustering, customer segmentation, outlier detection for website access patterns, etc.
- **Supervised learning** generalizes from *Labeled data* to facilitate future predictions of label based on input data.
  - **Classification**: Predict a discrete value from a *pre-defined* set of class labels
    - E.g., given a loan applicant, predict if she/he is a *good* or *bad* client. (*Approval* or *rejection*)
    - More examples: Churn prediction, skin cancer detection, fraud detection in banking, spam filtering, etc.
  - **Regression**: Predict a continuous value from a continuous range
    - E.g., predict the price of a stock or the price of a house

Introduction to Data Science

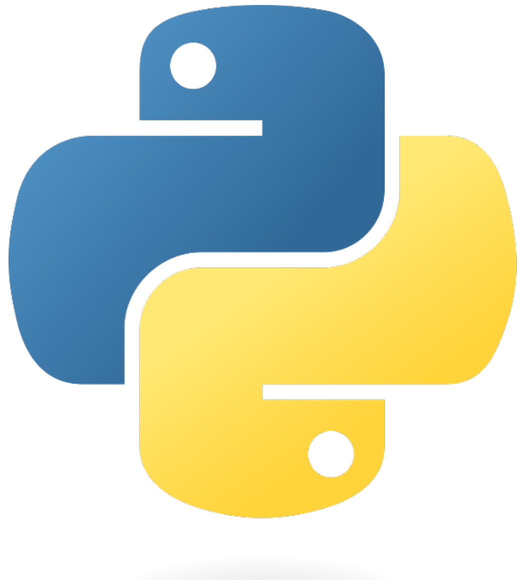
- **Data for supervised learning**
  - Uses supervised or label data, data of the form (input, output).
  - The *input data* is usually denoted by *X* and referred to as the *independent variable(s)*, *feature(s)*, or *predictor variable(s)*.
  - The output data is usually denoted by *Y* and referred to as the *dependent variable*, *response variable*, *target variable*, or *labels*.
  - Examples:
    - Features; Data about a house such as size, number of rooms, lot size, distance to school, neighborhood, etc. Response variable: House price
    - Features: Data about Titanic passengers such as age, passenger class, embarked, gender, etc. Response variable: Survived or not
    - Features: Data images of handwritten digits from 0 to 9, Response variable: a specification of what number the image depict

Roskilde Universitet

Outline of this lecture

- Introduction to the course
- Course practicalities and the exam
- Introduction to Data Science (and Data Engineering)
- Brief introduction to Python (and Jupyter Notebook)
- Essential data science packages in Python for data science: (NumPy) and Pandas
- Examples on real data
- Exercises

Brief introduction to Python



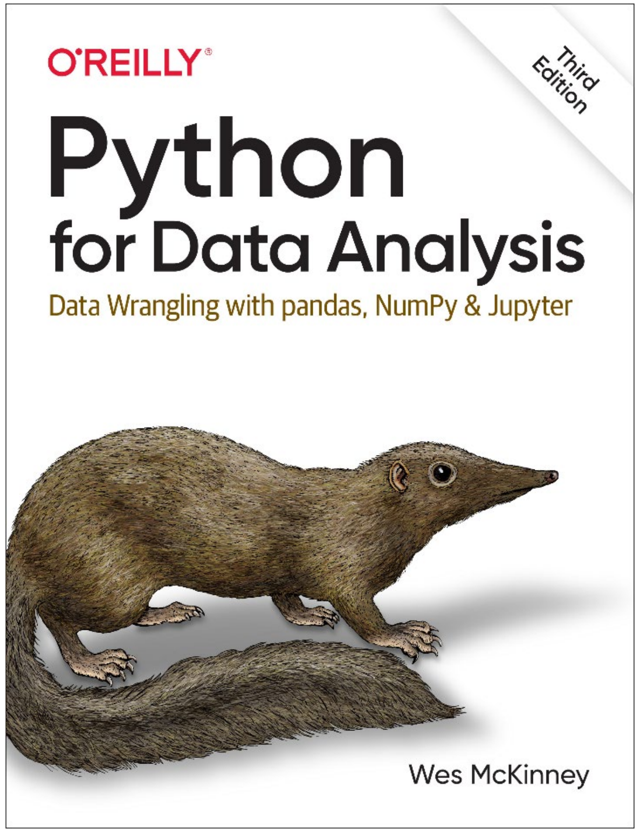
- We will use Python together with Jupyter Notebook
- We recommend installing Python through the miniconda distribution (https://docs.conda.io/en/latest/)
  - See Section 1.4 of Python for Data Analysis: https://wesmckinney.com/book/preliminaries#installation_and_setup
  - It will give you Python 3.14 (I think) and Jupyter Notebook
  - Alternatively, you can install the full Anaconda distribution of Python – it will give you more tools and packages and an Anaconda application
  - Alternative, you can use python notebooks in Google Colab or Visual Studio Code
- Python
  - Fairly old (early 1990ties), but very popular programming language
  - One of the two major programming languages for Data Science – the other one being R
  - General-purpose programming language with both object-oriented and functional programming.

Roskilde Universitet

Brief introduction to Python

- Jupyter Notebook
  - Named after Julia, Python and R.
  - Formerly IPython Notebook
  - Web-based interactive environment for creating and running notebook documents
  - A notebook is a type of document that consists of code and text cells
  - Great for learning as you can read and try out code at the same time
  - Great for communicating and sharing data science project as it works like a report that also contains all the code
  - Jupyter Notebooks can be shared and run without any problem
  - (JupyterLab is a newer and more extensive application)

Brief introduction to Python



- Starting up Jupyter notebook with Python
  - https://wesmckinney.com/book/preliminaries#installation_and_setup
  - Start the miniconda prompt (*"Anaconda Prompt (miniconda3)"*)
  - Create an environment (*"conda create -y -n pydata-book python=3.10"*)
  - Activate the environment (*"conda activate pydata-book"*)
  - Potentially install new packages *("conda install -y pandas jupyter matplotlib scikit-learn")*
  - Starting Jupyter Notebook (*"jupyter notebook"*)
  - Starting JupyterLab ("jupyter lab")
  - Go through example notebooks (from moodle) if you are not familiar with Python and Jupyter notebooks
    - A Crash Course in Jupyter Notebook.ipynb
    - A Crash Course in Python.ipynb

Outline of this lecture

- Introduction to the course
- Course practicalities and the exam
- Introduction to Data Science (and Data Engineering)
- Brief introduction to Python (and Jupyter Notebook)
- Essential data science packages in Python for data science: (NumPy) and Pandas
- Examples on real data
- Exercises

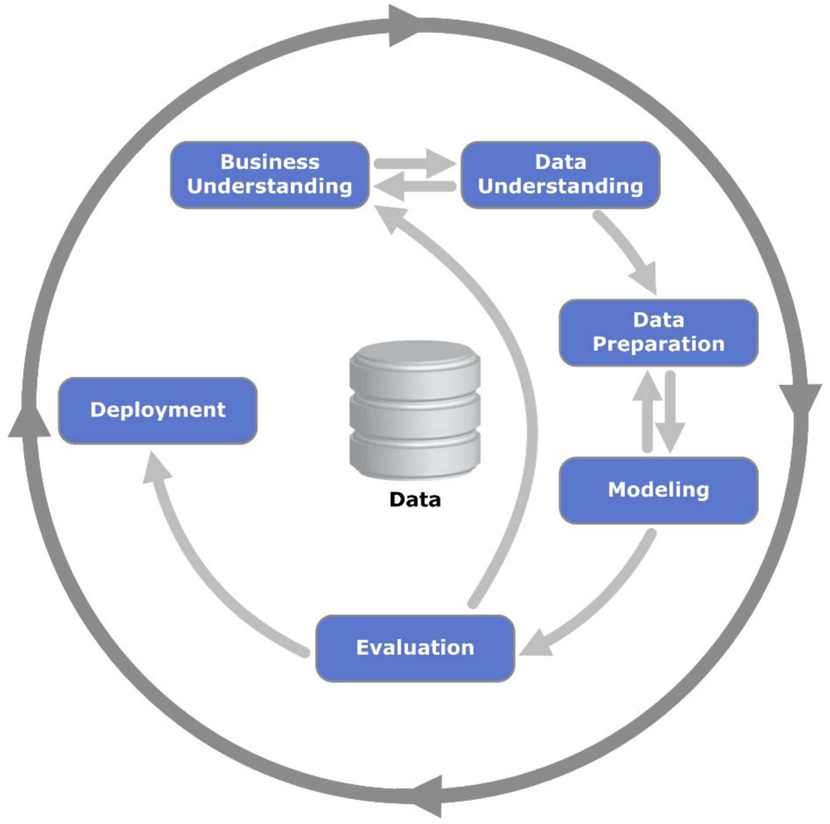
Roskilde Universitet

Essential data science packages in Python for data science: (NumPy) and Pandas

- Some of the most essential data science packages in Python
  - **NumPy** (https://numpy.org/)
    - A library for optimized numerical computations and advanced tensor data structures
  - **pandas** (https://pandas.pydata.org/)
    - Introduces the DataFrame data structure along with tools for reading data into DataFrames, as well as tools for manipulating, transforming and summarizing DataFrames.
  - **Matplotlib** (https://matplotlib.org/)
    - A classic library for making data visualizations in Python
  - **Seaborn** (https://seaborn.pydata.org/)
    - Another plotting library that builds on top of matplotlib to provide nicer visualizations and integration with pandas.
  - **plotnine** (https://plotnine.org/)
    - Another plotting library inspired by the powerful and popular ggplot2 of R
  - **Scipy** (https://scipy.org/)
    - Extent NumPy with more functionality to make scientific computations including basic statistics
  - **Statsmodels** (https://www.statsmodels.org/stable/index.html)
    - A library for doing statistical modeling in Python
  - **scikit-learn** (https://scikit-learn.org/stable/)
    - The classic library in Python to do machine learning
  - **…**

Essential data science packages in Python for data science: (NumPy) and Pandas

- Back to the data science process…
  - *"Data Science folklore: Data preparation/Data transformation/Data Wrangling/Data tidying take up 70% of the time in a data science project"*
  - Being skilled in this part can really make a difference in the success and speed of data science projects!
  - It is hard to teach! (at least without being somewhat boring)
    - We will talk a bit about it today and next time, but you will only properly learn these skills by getting your hands dirty by working on real data (- thus, you will improve your skills throughout the course)



Essential data science packages in Python for data science: (NumPy) and Pandas

- NumPy
  - Has a basic data structure called ndarray that are highly useful multi dimensional arrays (also useful to represent the tensors used in Deep Learning)
  - Allow for efficient multiple computation without the need of for loops
  - See the notebook “NumPy introduction.ipynb”

Essential data science packages in Python for data science: (NumPy) and Pandas

- Pandas
  - Works with tabular heterogeneous data (contrary to NumPy that works with homogeneously data of type numerical).
  - Has basic data type pandas Series and pandas DataFrame
  - DataFrame is a central data structure in doing data science on structured data
  - Pandas also provide functions for reading data into DataFrames
  - See the notebook “pandas basic.ipynb”

Essential data science packages in Python for data science: (NumPy) and Pandas

- **Good steps for loading data into Python from a file**
  1) Figure out what format the file is
  2) Figure out the right data reading function to load that format
  3) Have a look at the file (in Excel, LibreOffice, a text editor, … etc.)
  4) Determine whether there are any specific arguments you need to provide to the data reading function (separator, skipping rows, … etc.)
  5) When you have loaded the data into Python, have a look at it (using head, tail, shape, etc.)
  6) Is the data as expected? If not, repeat step 3-6.

Outline of this lecture

- Introduction to the course
- Course practicalities and the exam
- Introduction to Data Science (and Data Engineering)
- Brief introduction to Python (and Jupyter Notebook)
- Essential data science packages in Python for data science: (NumPy) and Pandas
- Examples on real data
- Exercises

Examples on real data

- Let us look at some real examples
    - The notebook “first data examples.ipynb”

Outline of this lecture

- Introduction to the course
- Course practicalities and the exam
- Introduction to Data Science (and Data Engineering)
- Brief introduction to Python (and Jupyter Notebook)
- Essential data science packages in Python for data science: (NumPy) and Pandas
- Examples on real data
- Exercises

Exercises

- Make sure you have Python and Jupyter notebook installed!
- If you are unfamiliar with Python or Jupyter notebook, you can look at the notebooks “A Crash Course in Jupyter Notebook.ipynb” and “A Crash Course in Python.ipynb” on moodle.
- **Do the exercise at the end of the “first data examples.ipynb” notebook**
- (Load the IRIS dataset from scikit-learn – what is the data all about?)
- (Find an interesting dataset on UCI Machine Learning Repository and load it into a Python notebook and investigate it)