Advanced Data Analytics — Lecture 1

Introduction to the course, logistics, crash-course to the computational infrastructure, and Python

Simon Scheidegger
Department of Economics, University of Lausanne, Switzerland

September 15th, 2025 | 10:15 - 12:00: Internef 126 | 16:30 - 18:00: Anthropole 3185

What is this course about?

What is this course about?

- Study some of the most commonly used machine learning paradigms and algorithms.
- Sufficient amount of detail on their mechanisms: explain why they work, not only how to use them.
- Plenty of coding.

Course Roadmap I

1. Introduction to Machine Learning.
2. A crash course on programming in Python.
3. Supervised Learning — the general idea, Linear Regression (with multiple variables), Gradient Descent, Polynomial Regression, Tuning Model Complexity, Introduction to Pandas.
4. Linear Classification, Discriminant Functions, Generative Models, Discriminant Models, Logistic Regression, How to evaluate Classifiers.
5. k-Nearest-Neighbours, Naive Bayes, Decision Trees, Combining Models (Boosting, Bagging).
6. Deep learning basics, Multi-layer perception, Feed-forward networks, Network training - SGD, Error back-propagation, some notes on overfitting, Introduction to Tensorflow (applied to supervised machine learning problems).

Course Roadmap II

7. Natural language processing, Large language models (LLMs)
8. Gaussian Process regression, Kernels with noise, Option-pricing examples, Bayesian active learning, GP classification.
9. The curse of Dimensionality, Active Subspaces and Gaussian Process Regression, Principal Component Analysis.
10. k-Means, Gaussian Mixture Models, Expectation Maximization, Hierarchical Clustering, Density-based Clustering.
11. Introduction to Reinforcement learning, RL: Optimal portfolio choice application, Putting things together: an optimal growth model, solved with value function iteration, Gaussian Process regression, Active subspaces, and Gaussian Mixture Models.

“Non scholae sed vitae”: Admin and Logistics

- **Meeting**: Mondays, 10:15-12:00; 16:30 – 18:00
- **Exercises**: Weekly, Mondays 17:15 – 18:00
- **Lecturer:** Simon Scheidegger (simon.scheidegger@unil.ch)
- **TA team**: Maria Pia Lombardo (mariapia.lombardo@unil.ch), Anna Smirnova (anna.smirnova@unil.ch)
- **Nuvolos Cloud Support**: support@nuvolos.cloud
- **Course Website**: Lecture notes are on *Nuvolos.cloud*
- To enroll in this class, please click on this enrollment key: https://app.nuvolos.cloud/enroll/class/sBsa1T_Mm5Y, and follow the steps.
- Course website: https://ap-unil-2025.github.io/ada-course-materials/.
  - For questions regarding the previous class, use the website: https://ap-unil-2025.github.io/ada-course-materials/
  - Office hours for details/exercises: ask the TAs directly.

Your lecturer: Simon Scheidegger

- Associate Prof. in Economics.
- Ph.D. in theoretical nuclear Astrophysics.
- Senior Visiting Fellow, Grantham Institute, London School of Economics.
- Visiting Fellow, BIS.
- Research in Computational finance and economics, Deep learning, climate change economics, and machine learning applied to economics and finance, high-performance computing, macro-finance.
- Associate Editor, The Journal of Financial Econometrics.
- Find some of my research here in case you are interested.

The TA team

- Maria Pia Lombardo: Ph.D. student in Economics.
- Anna Smirnova: Ph.D. student in Economics.

Enhance your productivity with GitHub

- Open a free GitHub account!
- github.com

Prerequisites: what do you need to know?

You should know how to do **math** and how to program:

- Multivariate calculus
- Probability / statistics
- Algorithms / Big O notation
- Linear Algebra
- Optimization
- Programming:
  - You will implement and apply the algorithms and apply them to data sets.
  - Assignments will be pen, paper, and mainly in Python.

I will review those things, but I will not teach them.

Background

- Calculus:

$$E = mc^2 \Rightarrow \frac{\partial E}{\partial c} = 2mc$$

- Linear algebra:

$$\boldsymbol{A}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i; \quad \frac{\partial}{\partial \boldsymbol{x}} \left(\boldsymbol{x}^T \boldsymbol{a} \right) = \boldsymbol{a}$$

- Probability:

$$p(X) = \sum_Y p(X, Y); p(x) = \int p(x, y) dy; \mathbb{E}_x[f] = \int p(x) f(x) dx$$

- It will be possible to refresh, but if you've never seen these before this course will be very difficult.

Grading

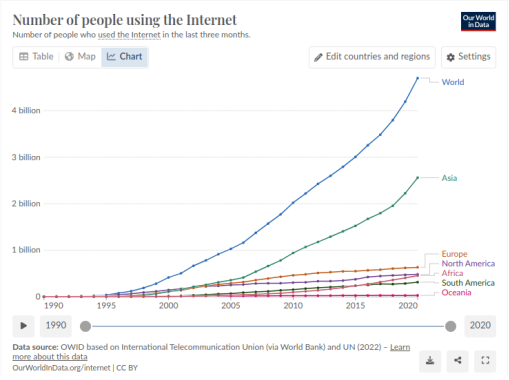
- The final grade of the class will be based on a **Capstone Project**.
- All students INDIVIDUALLY need to submit a **Capstone Project** at the end of the semester.
- The grade will be based on the recorded presentation, the write-up, the code base, and the originality/complexity of the project.
- See `capstone_project/capstone_project.pdf` on Nuvolos for more details.

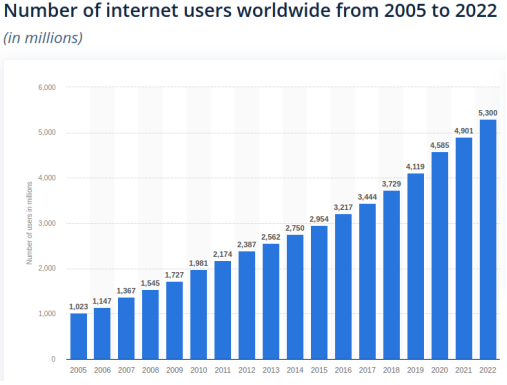
"We are drowning in information and starving for knowledge."

— Rutherford D. Roger

Big Data and its availability

https://ourworldindata.org/internet





Big Data and its availability

http://www.live-counter.com/how-big-is-the-internet

Size of the internet as we speak: TBD Petabytes

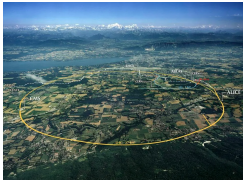
- 1 Gigabyte $\sim$ 1000 MB
- 1 Terabyte $\sim$ 1000 GB
- 1 Petabyte $\sim$ 1000 TB
- 1 Exabyte $\sim$ 1000 PB
- 1 Zettabyte $\sim$ 1000 EB

1 Gigabyte: If an author writes a book of about 190 pages, more specifically, of 383,561 characters (with spaces and punctuation included) every week for 50 years — this would be a billion letters or bytes.
1 Exabyte: 212 million DVDs weighing 3,404 tons.
1 Zettabyte: 1,000,000,000,000,000,000,000 bytes or characters.
This, printed on graph paper (with one letter in each mm^2 square) would be a paper measuring a billion km. The entire surface of the Earth (510 million km^2) would be covered by a layer of paper almost twice.

Other sources of Big Data

- **Scientific experiments**
  - CERN (e.g., LHC) generates $\sim$ **25 petabytes** per year (2012).
  - LIGO generates $\sim$ **1 Petabyte** per year
- **Numerical computations**
- ...

https://www.olcf.ornl.gov/summit/



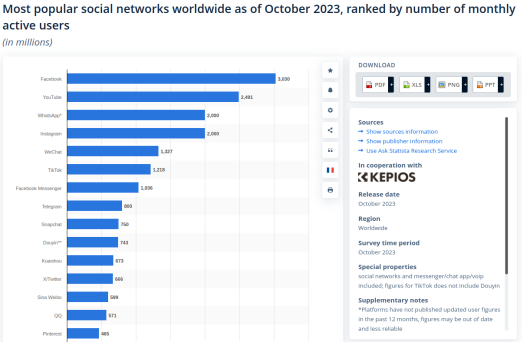
https://home.cern/

https://www.ligo.caltech.edu/

The need for Data Analytics

- Widespread use of personal computers and wireless communication leads to "big data".
- We are both producers and consumers of data.
- Data is often not random, it has structure, e.g., customer behavior.
- We need "big theory" to extract that structure from data for
  - Understanding the data-generating process.
  - Making predictions for the future.

$\Longrightarrow$ **We need Data Analytics**



The purpose of Data Analytics

Xia, B. S., & Gong, P. (2014). Review of business intelligence through data analysis. *Benchmarking: An International Journal*, 21(2), 300-311

Data analysis is a process of

- inspecting data
- cleansing data
- transforming data
- modeling data

with the goal of

1. discovering useful information.
2. informing conclusions.
3. supporting decision-making.

Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, while being used in different business, science, and social science domains.
In today's business, data analysis is playing a role in

- making decisions more scientific.
- helping the business achieve effective operation.

Data Mining

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems.

- **Retail:** Market basket analysis, Customer relationship management (CRM).
- **Finance:** Credit scoring, fraud detection, trading.
- **Manufacturing:** Control, robotics, troubleshooting.
- **Medicine:** Medical diagnosis.
- **Telecommunications:** Spam filters, intrusion detection.
- **Bioinformatics:** Motifs, alignment.
- **Web mining:** Search engines

Why Machine Learning?

- **Machine learning** aims at gaining insights from data and making predictions based on it.
- Build a model that is a good and useful approximation to the data.
- Machine learning methods have been investigated for more than 60 years, but became mainstream only recently due to more data being available and advances in computing power ("*Moore's Law*").
- There is no need to "learn" to calculate, for example, the payroll.
- Learning is used when:
  - Human expertise does not exist (navigating on Mars).
  - Humans are unable to explain their expertise (speech recognition).
  - Solution changes in time (routing on a computer network).
  - Solution needs to be adapted to particular cases (user biometrics).
- You're relying on machine learning every day, maybe without being aware of it! You certainly use a Smart Phone?

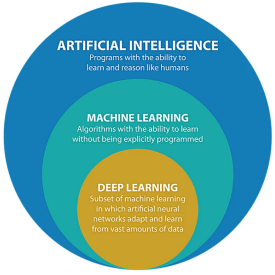
Set some terminology straight

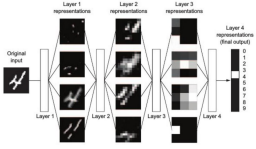
Artificial intelligence (AI)

Can computers be made to "think"? — a question whose ramifications we're still exploring today. A concise definition of the field would be as follows: The effort to automate intellectual tasks normally performed by humans.

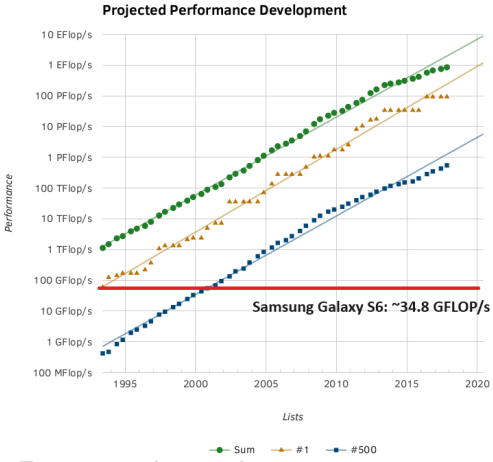
Machine Learning (ML)

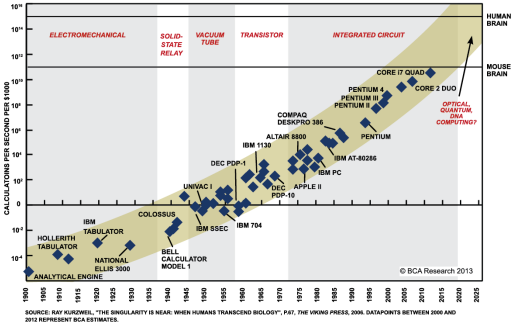
Deep Learning as a particular example of an ML technique





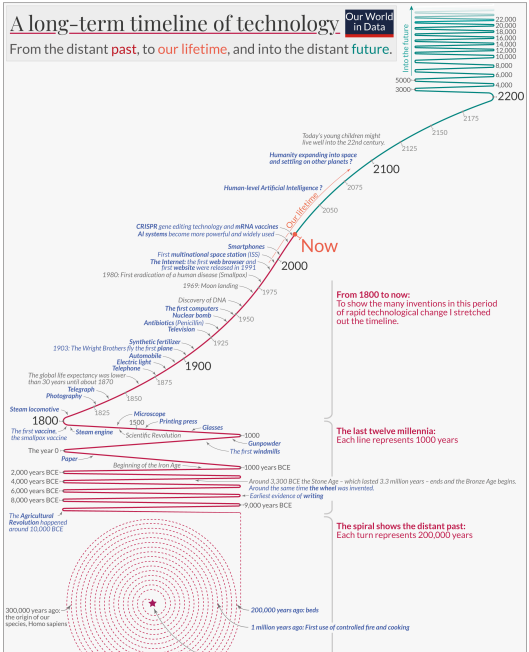
Moore's Law





Exponential growth in computing power allows us to move away from stylized models toward "realistically-sized" problems.

Speed of Scientific Discovery

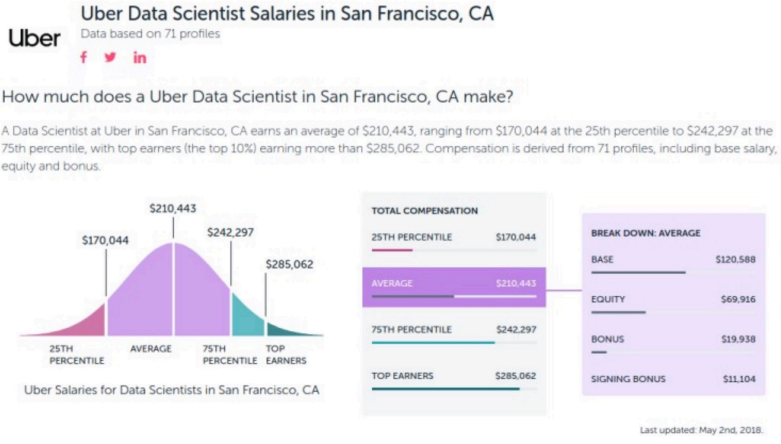


ML – the "coolest thing" in science

"*A breakthrough in machine learning would be worth ten Microsofts*"

— Bill Gates

Dozens of Billions/year USD$ globally spend on AI & ML.



Structured Data

- We often deal with structured data.
- Example: Data about monthly rent of real estate in a city.

Area	EstateType	DistanceToCenter	EnergyClass	MonthlyRent
58.20	Apartment	1.1	A	450
122.20	House	5.6	A	620
28.00	Apartment	0.5	B	320
8.00	Storage	6.3	B	150
75.78	House	2.2	C	375
66.00	Office	1.6	A	525
10.00	Storage	9.2	D	120

Unstructured and Semi-structured Data

- Many interesting datasets are unstructured, typically natural language texts written by humans, or semi-structured, interleaving structured and unstructured data.
- Example: Newspaper articles with assigned categories.

Category	Content
Sports	Bayern Munich was defeated by Real Madrid in the Champions League quarter finals...
Politics	Theresa May called for a snap general election on Monday...

Feature Types (I)

When working with structured data, we distinguish different types of features, depending on which operations can be applied.

Area	EstateType	DistanceToCenter	EnergyClass	MonthlyRent
58.20	Apartment	1.1	A	450
122.20	House	5.6	A	620
28.00	Apartment	0.5	B	320
8.00	Storage	6.3	B	150
75.78	House	2.2	C	375
66.00	Office	1.6	A	525
10.00	Storage	9.2	D	120

Dataset consists of five features.

Feature Types (II)

- **Nominal** features can be compared (==, !=) and counted (e.g., gender of a person, color of a car).
- **Ordinal** features can, in addition, be compared (<, >) (e.g., customer satisfaction level, energy class of car).
- **Numerical** features allow in addition for arithmetic operations (+, -, *, /), so that we can compute the difference between values, compute their mean, compute their variance, etc. (e.g., the fuel consumption of a car, income of a household).

Feature Types (III)

Area	EstateType	DistanceToCenter	EnergyClass	MonthlyRent
58.20	Apartment	1.1	A	450
122.20	House	5.6	A	620
28.00	Apartment	0.5	B	320
8.00	Storage	6.3	B	150
75.78	House	2.2	C	375
66.00	Office	1.6	A	525
10.00	Storage	9.2	D	120

Area: Num; EstateType: Nom; DistanceToCenter: Num; EnergyClass: Ord; MonthlyRent:Num.

ML Applications

- **Association**.
- **Supervised Learning**.
  Assume that training data is available from which they can learn to predict a target feature based on other features (e.g., monthly rent based on area).
  - **Classification**.
  - **Regression**.
- **Unsupervised Learning**
  Take a given dataset and aim at gaining insights by identifying patterns, e.g., by grouping similar data points.
- **Reinforcement Learning**.

Learning Associations

- Basket analysis:
- $P(Y \mid X)$ probability that somebody who buys X also buys Y, where X and Y are products /services.
- Example: $P(\text{salt} \mid \text{pepper}) = 0.7$

Supervised Regression

- Regression aims at predicting a numerical target feature based on one or multiple other (numerical) features.
- Example: Price of a used car.
  - x : car attributes
  - y : price
  - $y = h(x|\theta)$
  - $h()$: model
  - θ : parameters

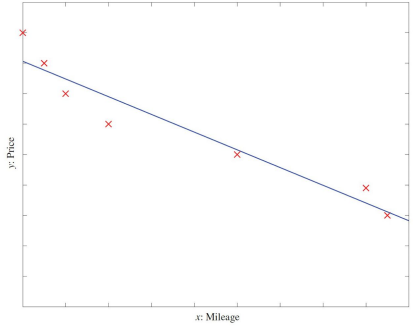


Fig. from Alpaydin (2014)

Supervised Classification

Example 1: Spam Classification

- Decide which emails are Spam and which are not.
- Goal: Use emails seen so far to produce a good prediction rule for **future** data.

Example 2: Credit Scoring

- Differentiating between low-risk and high-risk customers from their income and savings.
- Discriminant: `IF income >` θ_2 `AND savings >` θ_2 `THEN low-risk ELSE high-risk`

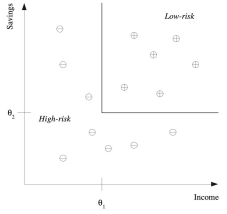


Fig. from Alpaydin (2014)

Classification: More applications

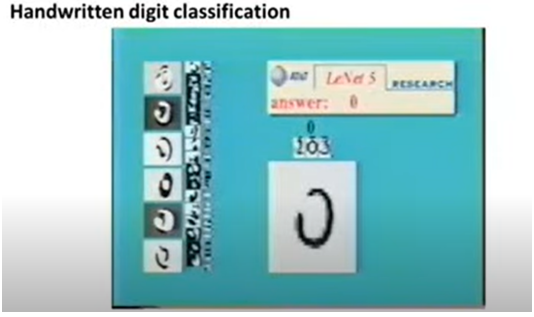
- **Face recognition**: Pose, lighting, occlusion (glasses, beard), make-up, hairstyle.
- **Character recognition**: Different handwriting styles.
- **Speech recognition**: Temporal dependency.
- **Medical diagnosis**: From symptoms to illnesses.
- **Biometrics**: Recognition/authentication using physical and/or behavioral characteristics such as face, iris, signature, etc.
- Outlier/novelty detection.

Random sampling of MNIST

Handwritten Digit Classification (LeNet)

Movie from the early 90's. We have come a long way since then...

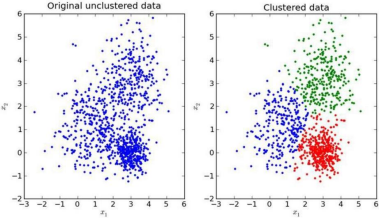
Handwritten Digit Classification – by Yann Lecun



Unsupervised Learning

Learning “what normally happens”.

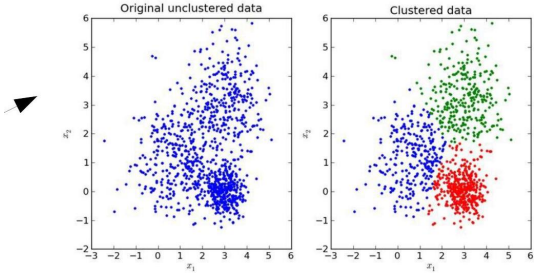
- No output.
- Clustering: Grouping similar instances.
- Example applications:
  - Customer segmentation.
  - Image compression: Color quantization.
  - Bioinformatics: Learning motifs.

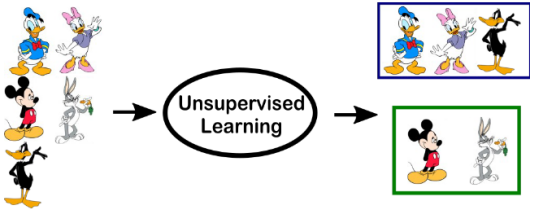


Unsupervised Learning

Learning “what normally happens”.

- No output.
- Clustering: Grouping similar instances.
- Example applications:
  - Customer segmentation.
  - Image compression: Color quantization.
  - Bioinformatics: Learning motifs.





Reinforcement Learning

- Learning a policy: A sequence of outputs.
- No supervised output but delayed reward.
- Credit assignment problem.
- Game playing.
- Robot in a maze.
- Multiple agents, partial observability, ...

→ *DeepMind's Qlearning*.

Self-Driving Cars

- Carnegie Mellon University — 1990's: Self Driving Cars S1E2: ALVINN.
- Google — 2017: Waymo.
- Classification.
- Regression.
- Reinforcement learning.
- Prediction.

Coloring Old Movies

AI movie restoration — Scarlett O'Hara HD

Two-Legged Robots

$\rightarrow$ *BostonDynamics' AtlasRobotCanDoParkour.*

$\rightarrow$ *Handyman.*

Thanks to machine-learning algorithms,
the robot apocalypse was short-lived.

Chopin with AI

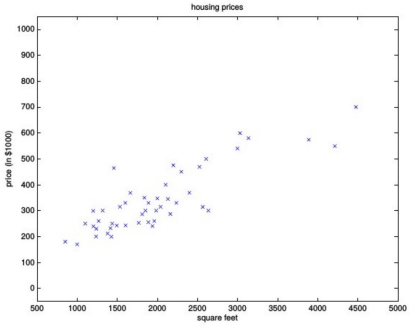
https://openai.com/research/musenet

Building an ML Algorithm

- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Inference from a sample.
- Role of computer science: Efficient algorithms to
  - Solve the optimization problem.
  - Representing and evaluating the model for inference.

Building an ML Algorithm (II)

Living area (feet ^{2})	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

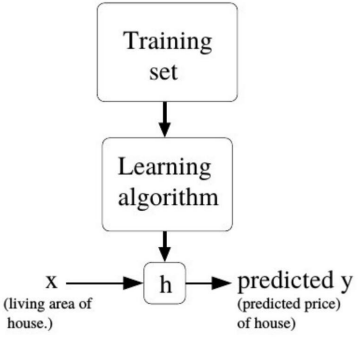


Given data like this, how can we learn to predict the prices of other houses as a function of the size of their living areas?

Building an ML Algorithm (III)

- $x(i)$: **"input" variables** (living area in this example), also called **input features**.
- $y(i)$: **"output" / target variable** that we are trying to predict (price).
- **Training example**: a pair $(x(i), y(i))$.
- **Training set**: a list of m training examples $(x(i), y(i)); i = 1, ..., m$.

To perform supervised learning, we must decide how we're going to represent **functions/hypotheses** h in a computer.



Building an ML Algorithm (IV)

- **Model/Hypothesis:**

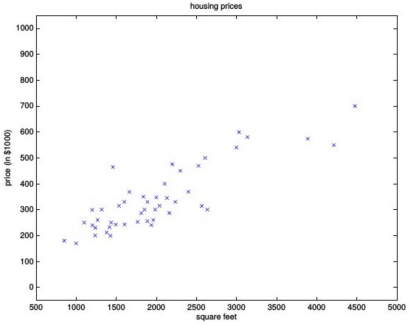
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

θ_i's: parameters

- **Cost Function:**

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$\implies$ Minimize $J(\theta)$ in order to obtain the coefficients θ.



Building an ML Algorithm (V)

In general, Machine Learning in 3 Steps:

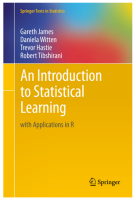
- Choose a model $h(x \mid \theta)$.
- Define a cost function $J(\theta \mid x)$.
- Optimization procedure to find $\theta*$ that minimizes $J(\theta)$.

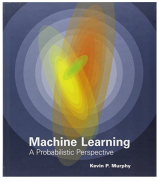
Computationally, we need data, linear algebra, statistics tools, and optimization routines.

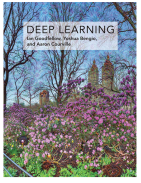
Some source material

I will not follow a textbook, but will point in my slides to the relevant literature.
Some useful textbooks:

- ***Machine Learning: a Probabilistic Perspective***
  K. Murphy, MIT Press, 2012. https://www.cs.ubc.ca/~murphyk/MLbook/index.html
- ***An Introduction to Statistical Learning***
  Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani; Springer, 8th editon, 2017.
  https://www-bcf.usc.edu/~gareth/ISL/
- ***Deep Learning***
  Ian Goodfellow and Yoshua Bengio and Aaron Courville; MIT Press 2016.
  http://www.deeplearningbook.org







Some source material (II)

- ***Pattern Recognition and Machine Learning***
  C. M Bishop; Springer 2006. (pdf freely available)
- ***Python Machine Learning***
  S. Raschka; PACKT Publishing 2017
- ***Introduction to Machine Learning***
  Ethem Alpaydin; MIT Press 2014.
- ***A Primer on Scientific Programming with Python***
  Hans Petter Langtangen; Springer 2016.
- ***Mathematics for Machine Learning***
  Deisenroth, A. Aldo Faisal, and Cheng Soon Ong; Cambridge University Press 2020.