MATHEMATICS
FOR MACHINE LEARNING

Nagoya University, Fall 2023

Lecture 1

Course information & Overview, Google colab & Python, Supervised learning I

https://www.henrikbachmann.com/mml2023.html

Who is teaching & helping?

Lecturer

Henrik Bachmann

www.henrikbachmann.com

henrik.bachmann@math.nagoya-u.ac.jp

Henrik Bachmann

Teaching Assistant

Risan

Risan

www.risan.io

NUEMI Tutors

keirrera Idle — Caleb

Moondust — Hevidu

raf — Muflih

Kita Ayase — Zhanfeng

Saraive7 — Ai

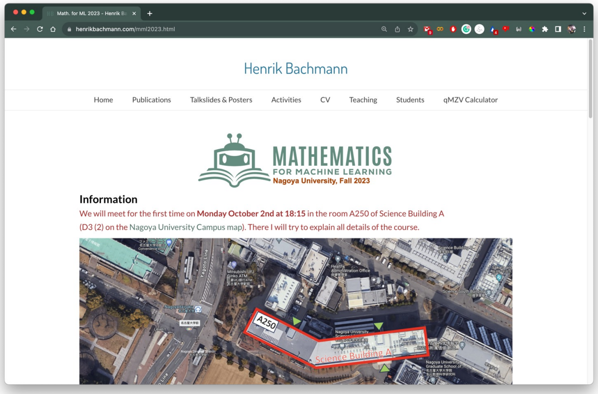
Pratham — Ersthardt Domnenberg

Rafi — Rafi

Raymond — Raymond

Alberto — albertothornton

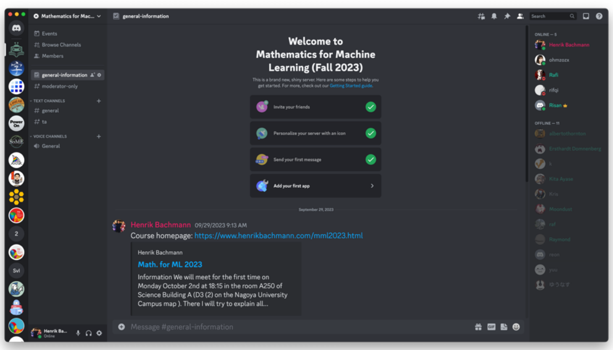
What where?



Homepage

https://www.henrikbachmann.com/mml2023.html

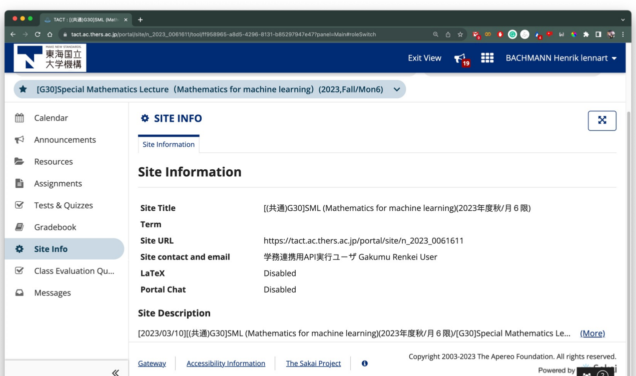
Contains schedule & course materials



Discord server

https://discord.gg/XG2qkFyGpD

Will be used for discussions and small announcements



TACT

https://tact.ac.thers.ac.jp/portal/site/n_2023_0061611

Will be used for Homework submission & important announcements

For who is this lecture?

- G30 Students of year >= 2.
  (For first year students: Consider taking this course in your second year)
- NUPACE Students.
- Japanese students who want to attend English lectures.

Basic knowledge in Linear Algebra and Calculus is helpful
(and sometimes necessary to understand everything in detail)

If you are **Japanese student**, please register at the NUEMI project for this lecture:

http://labguide.bio.nagoya-u.ac.jp/NUEMI/registration/

Free extra support & Study sessions

What is this course?

- This course is a "Special Mathematics Lecture" (Introduced by Prof. Richard)
- Can give 2 credit points depending on your year.
- For some students can count for GPA.
- You need to be registered in the NU Portal if you want these credits.
- Details: http://www.math.nagoya-u.ac.jp/~richard/Files/SML_official_rule.pdf

Goal of this lecture:

- Get an overview of some algorithms used in machine learning
- Understand the mathematics used in these algorithms
- Learn how to implement (simple version of) them in Python

Not goal of this lecture: Give detailed overview of machine learning libraries (tensorflow etc.) and learn how to use them

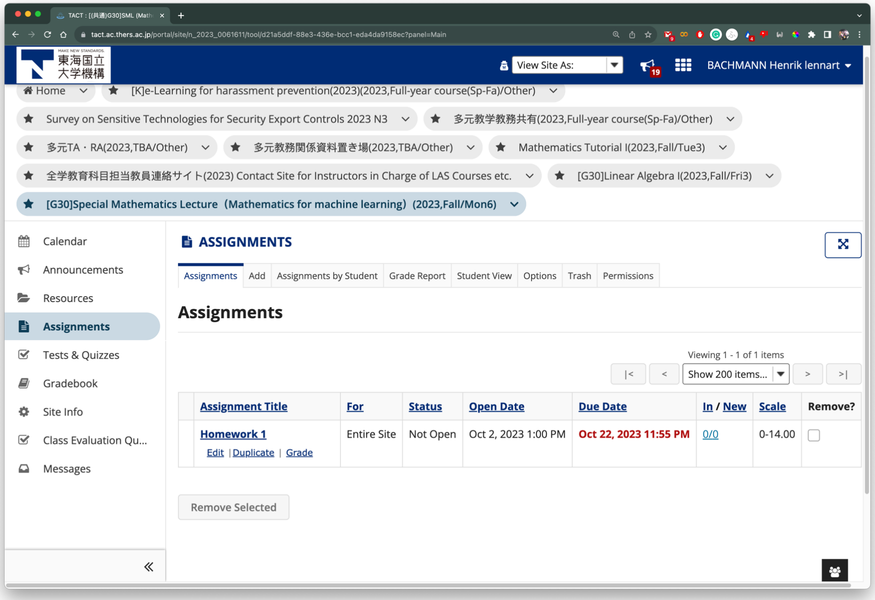
Grading

Grading will be based on

- Homework assignments (70%)
- Semester project (30%)

Homework assignments

- Programming exercises in Google Colab
- In total 5 Homework assignments
- 14 points each Homework
- Submission in TACT



Semester project

Objective: Choose a project topic related to Nagoya or Japan more broadly that can be addressed using machine learning algorithms. Your task is to develop a machine learning model to solve a specific problem or provide insights into an aspect of life, business, environment, culture, etc., in Nagoya/Japan.

Group Size: 1-3 members

Code: Preferably a Google Colab notebook. Exceptions are possible; please provide full documentation for any different technology or package used. If you plan not to submit a Google Colab, please contact us in advance.

Documentation: 5-10 slides as if you were going to present the project.
Your slides should cover (for example):

- Problem Statement
- Data Collection
- Data Exploration and Visualization
- Model Building and Evaluation
- Conclusion

If you have no clue what to do:
We will give more ideas/help
during the coming weeks

Tutorial

Time: Tuesday? Thursday? Friday? 6th period (vote now via menti.com)
Place: Same room (A250, Science Bld. A)

- Organized by the TAs.
- Talk again about the content of the lecture.
- Do the Assignments with the help of TAs & other students.

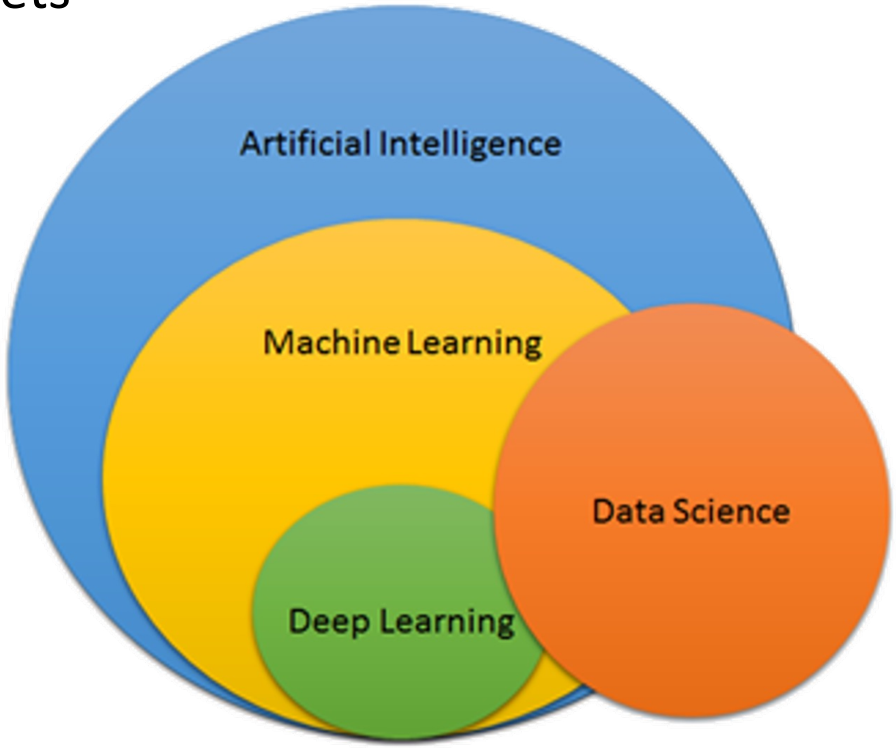
What is machine learning

Common definition:

Machine Learning is the study of computer algorithms that improve automatically through experience

Found its way in a lot of daily life applications, e.g.

- Self driving/flying cars/trains/planes/rockets
- Computer games
- Image & Video & Sound processing
- Speech recognition (Siri, Alexa,…)
- Translation (Google translate)
- Weather forecast
- Data analysis
- Email spam
- Robotics
- ChatGPT
- and much much more…



Not machine learning:
"Classical machines / programs"

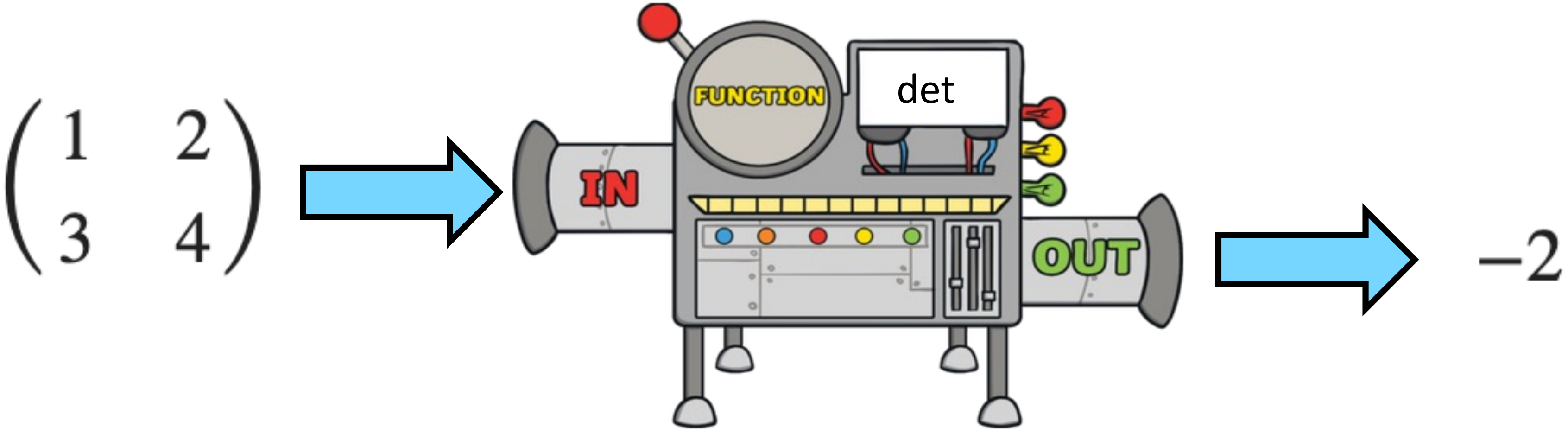
Programmer understands the problem and can implement an explicit algorithm, which turns an input into an output.

```
def determinant(A):
    total=0
    indices = list(range(len(A)))
    if len(A) == 2 and len(A[0]) == 2:
        val = A[0][0] * A[1][1] - A[1][0] * A[0][1]
        return val
    for fc in indices:
        As = A
        As = As[1:]
        height = len(As)
        for i in range(height):
            As[i] = As[i][0:fc] + As[i][fc+1:]
        sign = (-1) ** (fc % 2)
        sub_det = determinant(As)
        total += sign * A[0][fc] * sub_det
    return total
```

```
determinant([[1,2],[3,4]])
```

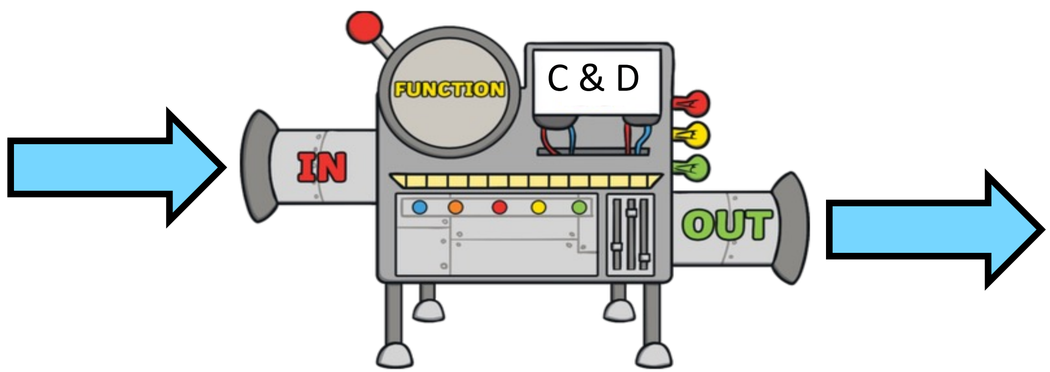
```
-2
```

Example: Determinant of a matrix



What do you see?

Want: Cat & Dog detection machine



"This is a dog!"

Cat & Dog detector

Smart person

Of course, I know how cats and dogs look like. But I do not know how to describe it with a formula or algorithm!

How do we know?

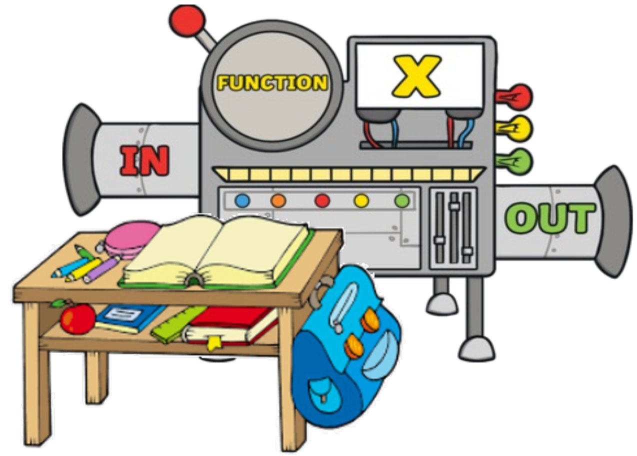
This is a dog because...

- I saw a lot of dogs & cats in my life.
- Whenever I saw a cat I was told by my parents „This is a cat!“. When I saw a dog they told me “This is a dog.”.
- I **learned** to distinguish between dogs and cats.
- But it is hard for me to explain the exact difference.

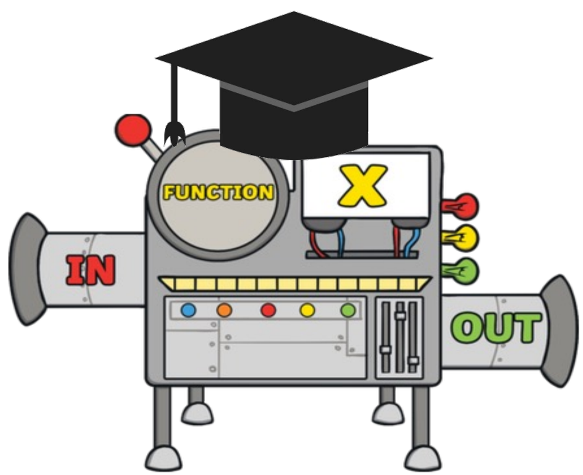
Machine learning

We can create machines/programs which we can teach

Training



After Training

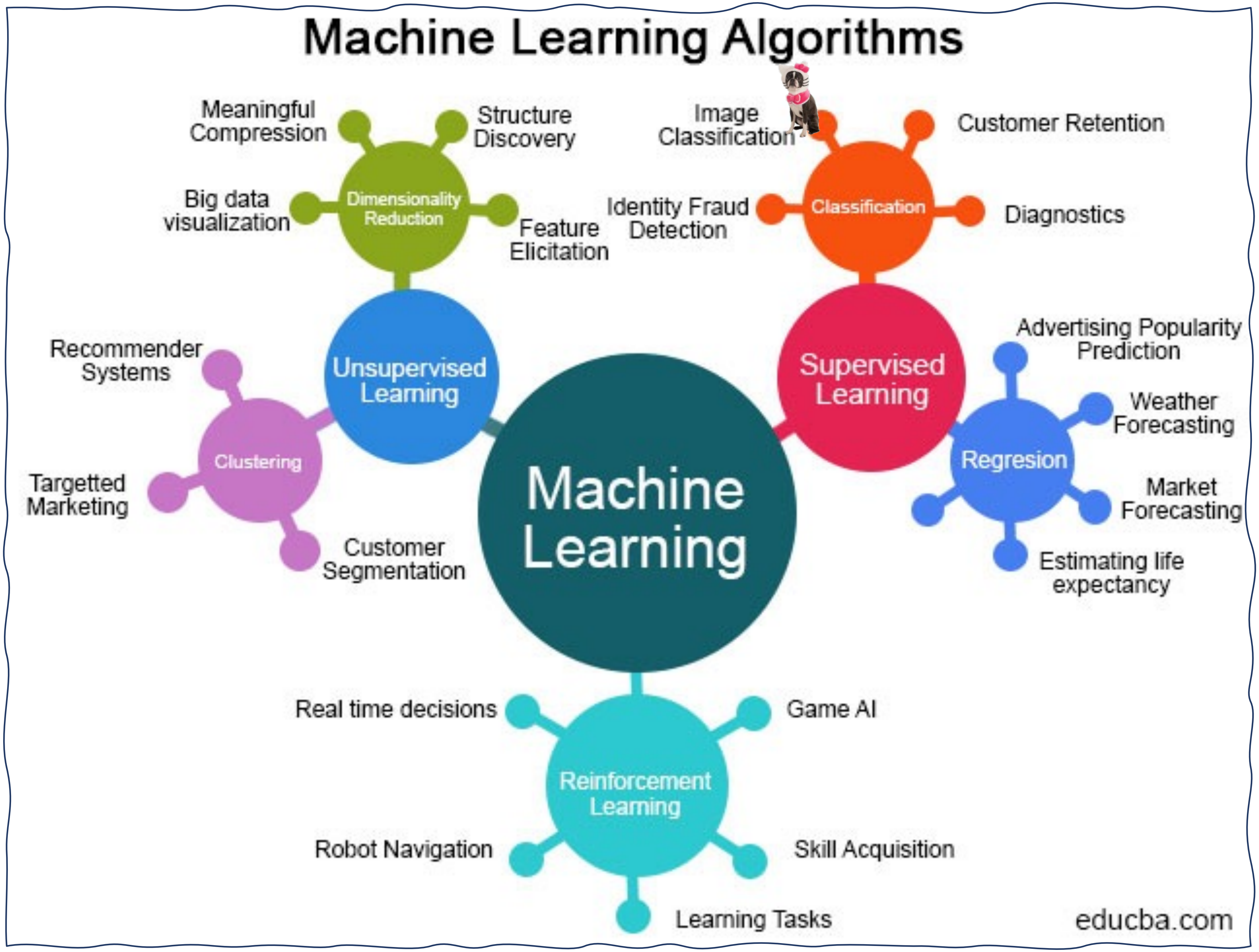


Cat & Dog detector

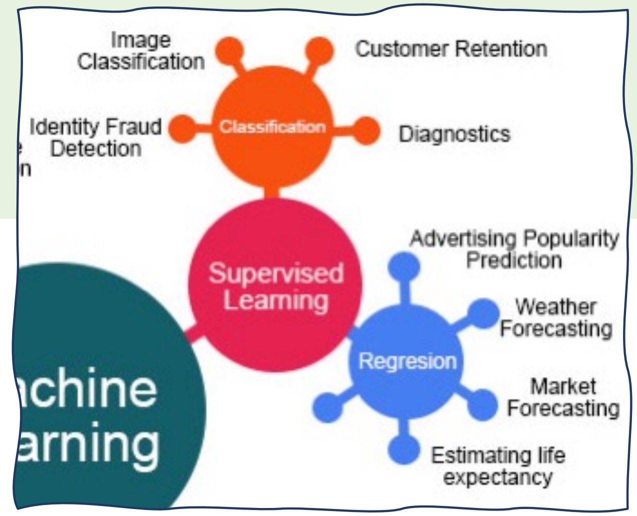
Dog!

… with 95% certainty!

Machine learning overview

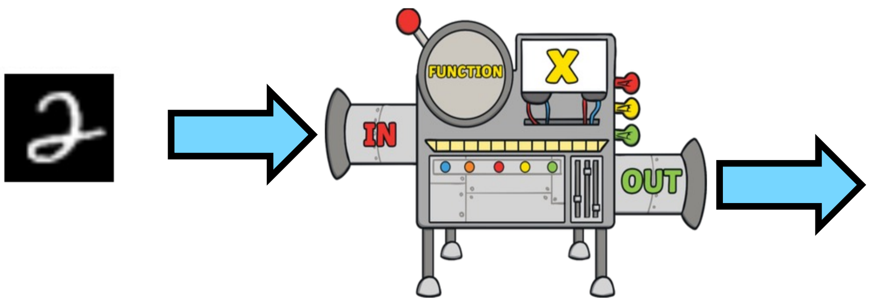


Supervised learning



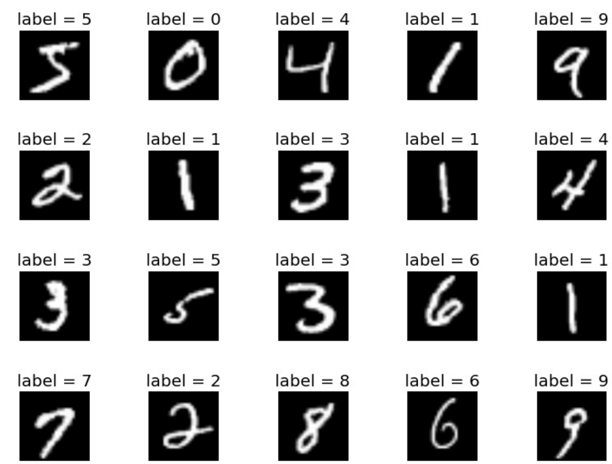
Classification (discrete output)

Output = category (e.g. “dog”, “cat” or “1”,”2”,…,”9”)



”2”

Handwriting recognition

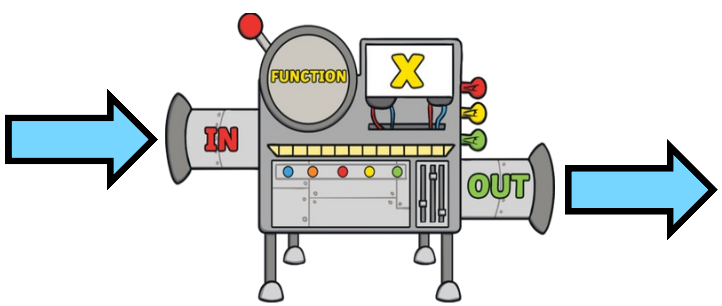


Learning by using some trainingsdata

Regression (continuous output)

Output = real number (like “price”, “weight”,…)

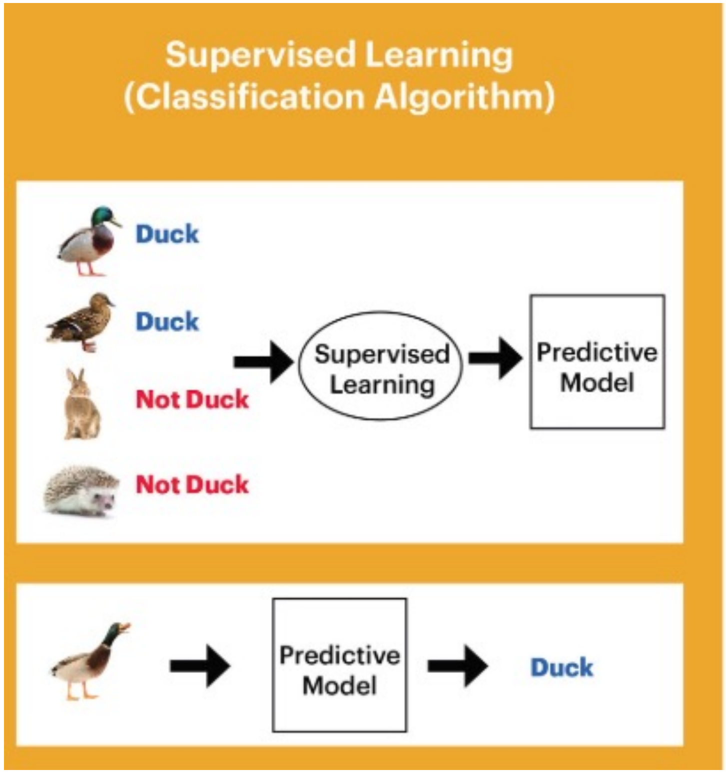
Size: $70m^2$
Rooms: 3

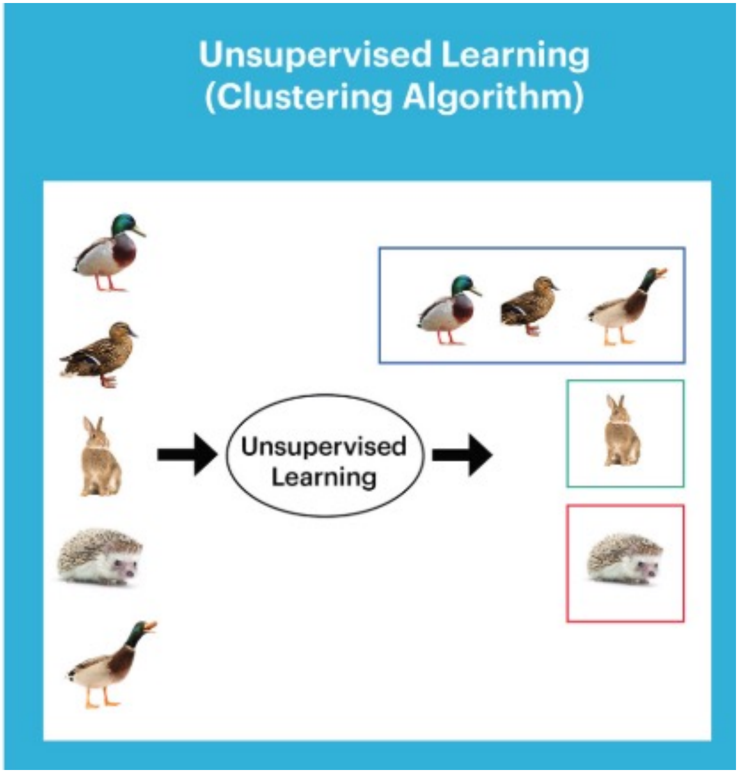


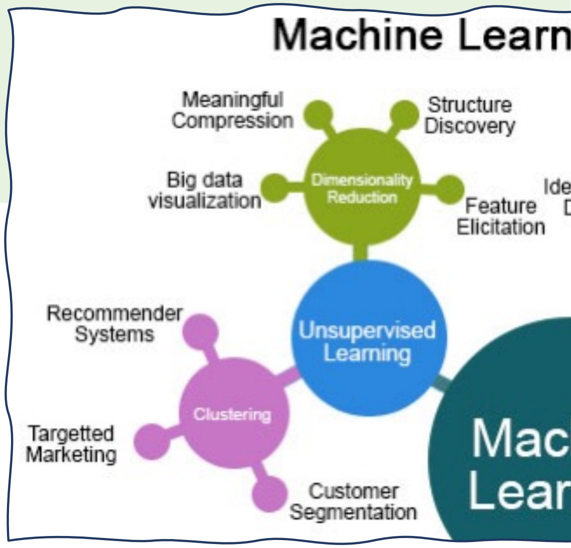
90.000 Yen

Apartment price prediction

Unsupervised learning

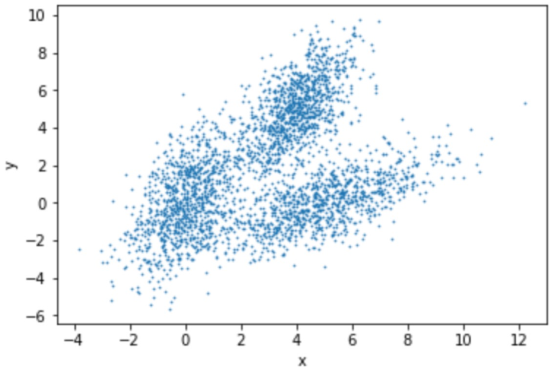




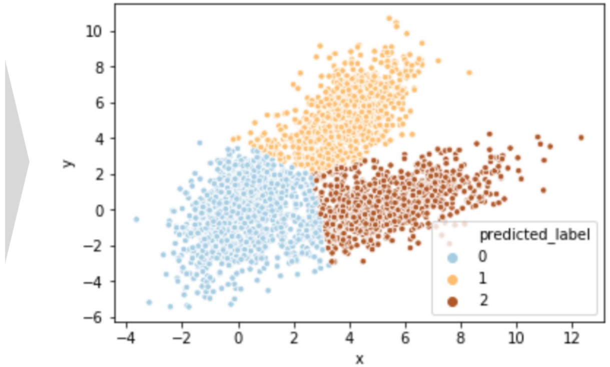


Clustering

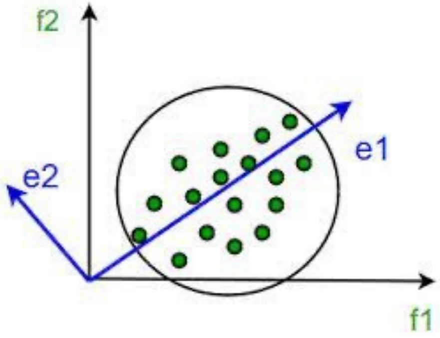
Before



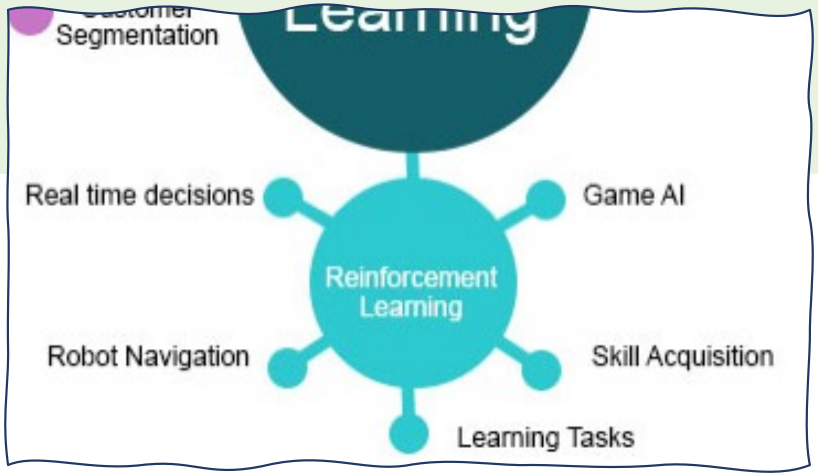
After

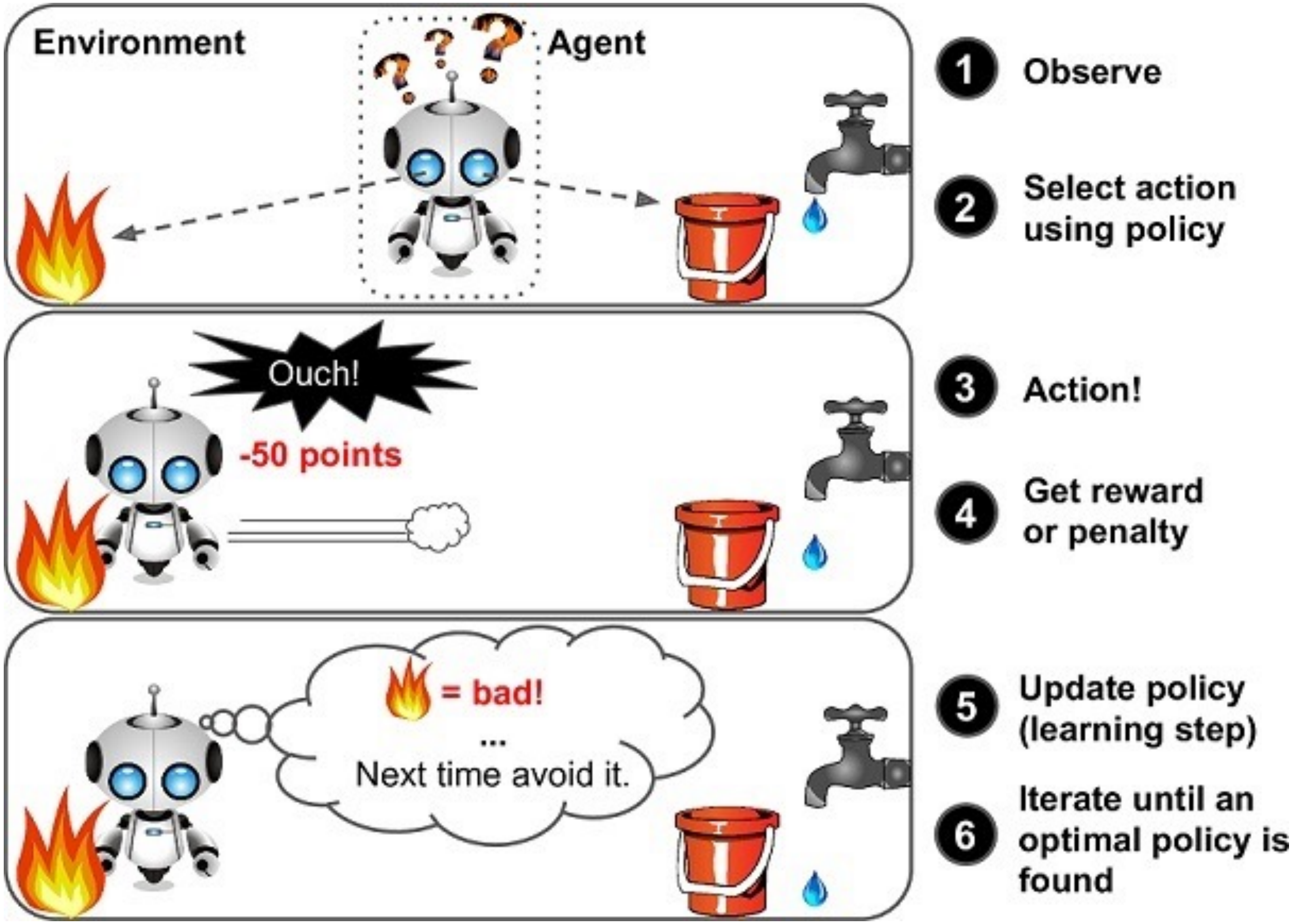


Dimension reduction



Reinforcement learning





Examples of AI in Chess, Go, Computer games, biology, math

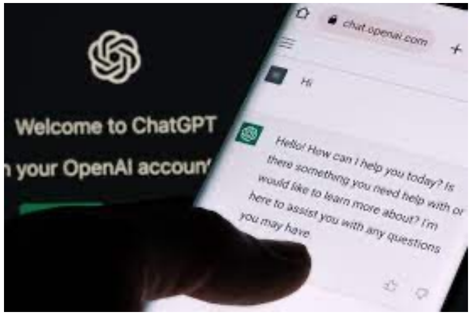
- 1996: Deep blue (not machine learning)
- 2016: Alpha Go
- 2017: Alpha Zero (plays chess, shōgi & go)
- 2019: AlphaStar (Starcraft 2)
- 2020: AlphaFold
- 2022: ChatGPT, AlphaTensor

Reinforcement learning

1996: Garry Kasparov (Chess world champion) loses against Deep blue.

2016: Lee Sedol loses against Alpha Go.

2019: AlphaStar beats Starcraft 2 pro gamers

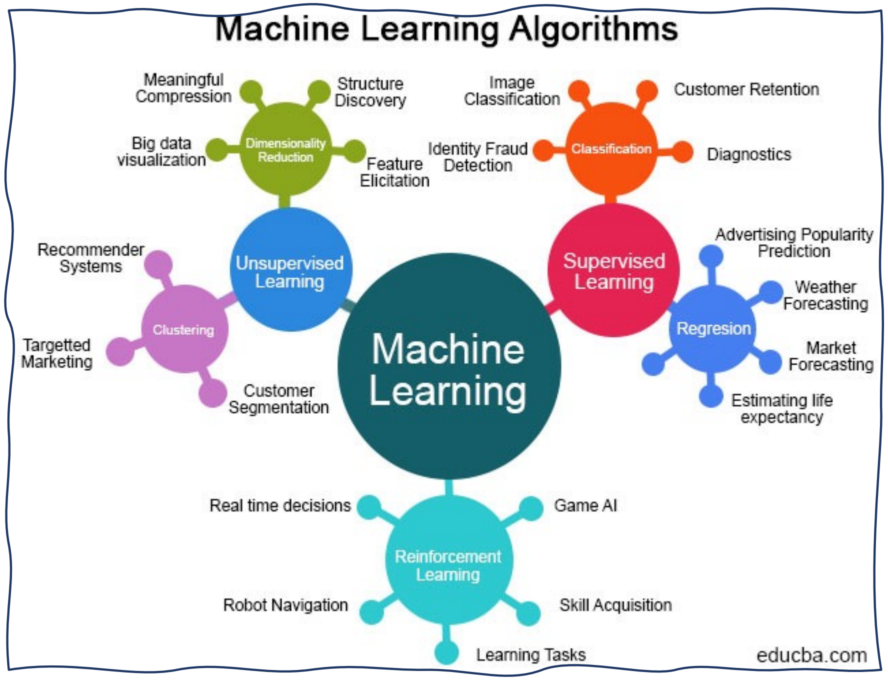


2022: ChatGPT… for lab reports and more

Over of the topics in this course

This year:

1. Introduction & Python
2. Supervised learning: Linear regression
3. Supervised learning: Neural networks (from scratch & advanced topics)
4. Unsupervised learning: Dimension reduction (e.g., PCA, Autoencoder)
5. Reinforcement learning: (deep) Q-learning
6. Language models (if time allows)



Fall 2022

1. Introduction to programming in Python
2. Supervised learning: Linear & logistic regression
3. Generative Learning algorithms: Naive Bayes
4. Reinforcement Learning: Q-Learning
5. Unsupervised learning: k-means clustering
6. Neural networks

Let us have a look at Google Colab and Homework 1

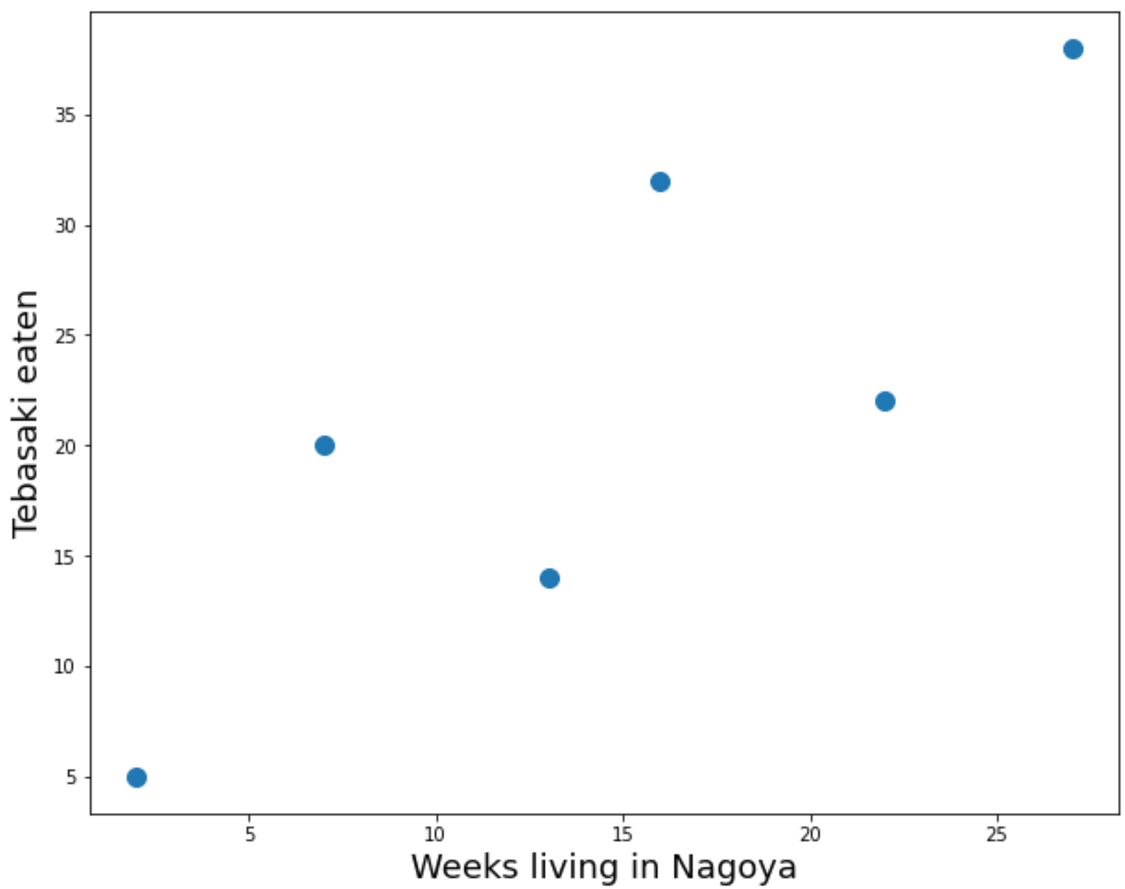
1 Supervised learning: Tebasaki example

Have: Some data of ”Weeks living in Nagoya” and “Tebasaki eaten”.

Want: A functions, which creates out of an an arbitrary input for “Weeks living in Nagoya” a prediction for “Tebasaki eaten”.

Weeks living in Nagoya	Tebasaki eaten
2	5
7	20
13	14
16	32
22	22
27	38

Yamachan & Furaibo – possible places to increase the number of ”Tebasaki eaten”



1 Supervised learning: Some notations

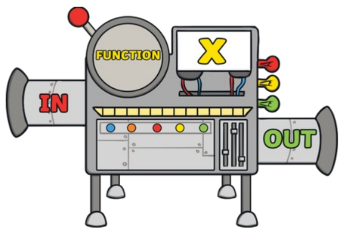
Training set

	feature	label (target)
	Weeks living in Nagoya	Tebasaki eaten
	2	5
	7	20
	13	14
Training example	16	32
	22	22
	27	38

Learning algorithm

(arbitrary number of) Weeks living in Nagoya

(feature)



hypothesis

Predicted number of eaten Tebasaki

(label)

1 Supervised learning: Notations

Weeks living in Nagoya	Tebasaki eaten
2	5
7	20
13	14
16	32
22	22
27	38

- Input values (Feature space): $\mathcal{X}$
- Output value (Label space): $\mathcal{Y}$
- Trainings example: $(x, y) \in \mathcal{X} \times \mathcal{Y}$.
- Trainings set (with n training examples): $\mathcal{T} = \big((x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\big) \in (\mathcal{X} \times \mathcal{Y})^n$.
- hypothesis: A function $h : \mathcal{X} \to \mathcal{Y}$.
- Learning algorithm: An algorithm to create a hypothesis h out of a trainings set $\mathcal{T}$.

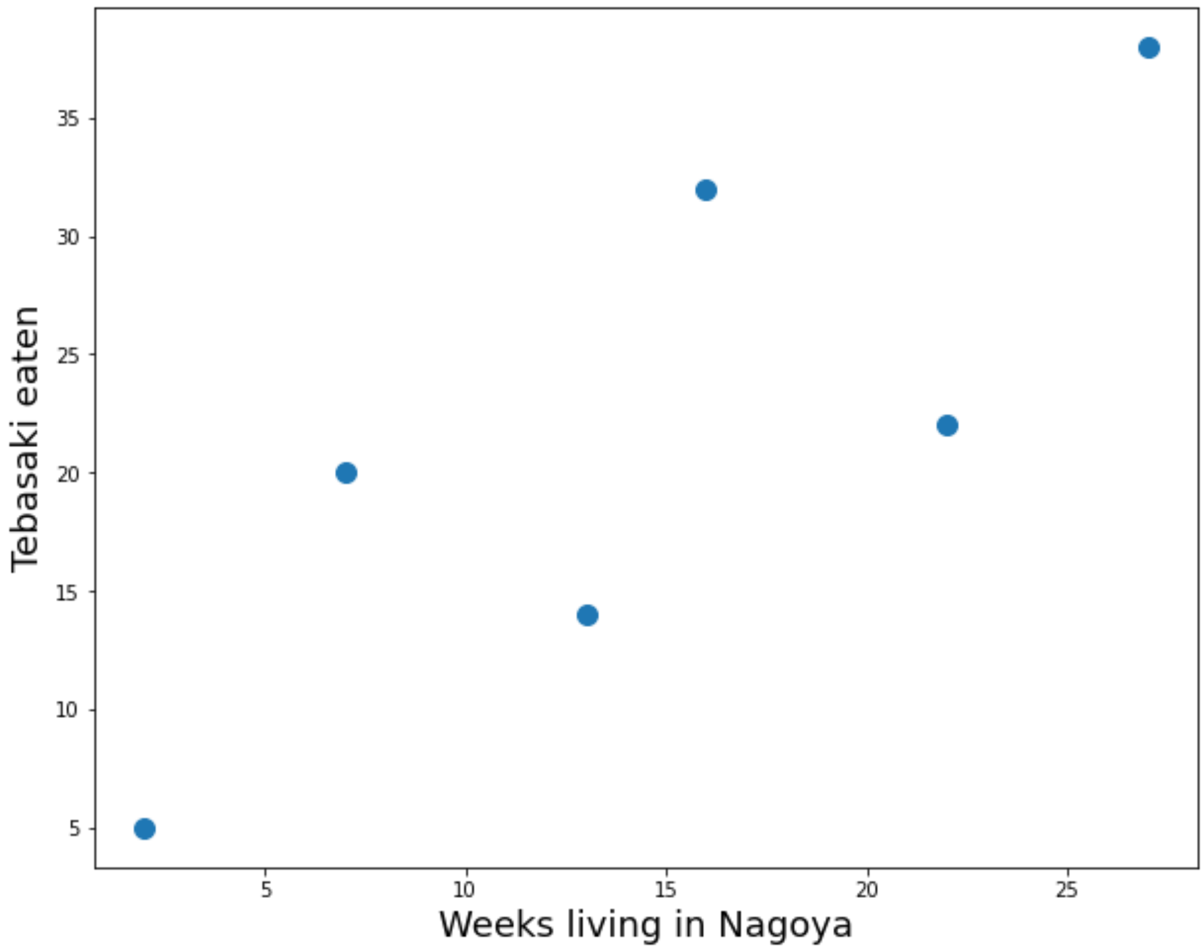
Tebasaki Example: $\mathcal{X} = \mathbb{R}$ $(\mathbb{N})$, $\mathcal{Y} =$

$n = 6$ $\mathcal{T} = ((2,5), (7,20), \ldots, (27,38)) \in (\mathbb{R} \times \mathbb{R})^6$

Tebasaki example:

Idea: $h_\theta(x) = \theta_0 + \theta_1 x$

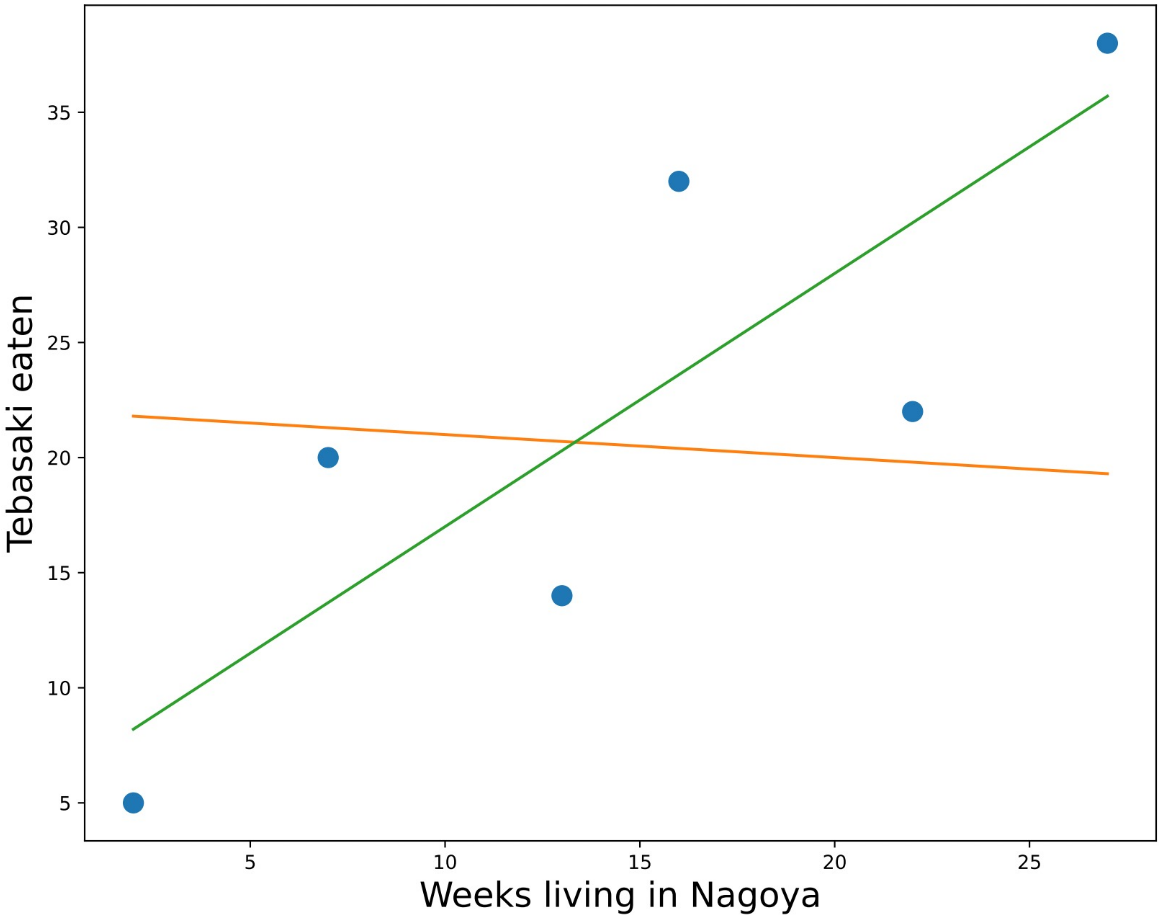
$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} \in \mathbb{R}^2$$



Weeks living in Nagoya	Tebasaki eaten
2	5
7	20
13	14
16	32
22	22
27	38

Tebasaki example:

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



The graphs of $h_{\theta'}$ and h_θ with $\theta' = \begin{pmatrix} 22 \\ -0.1 \end{pmatrix}$ and $\theta = \begin{pmatrix} 6 \\ 1.1 \end{pmatrix}$

1 Supervised learning – Linear Regression

Learning Algorithm: Linear Regression

Let $\mathcal{X} = \mathbb{R}^d$, i.e. we have d features, and $\mathcal{Y} = \mathbb{R}$. As an Ansatz for the hypothesis we set

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} \in \mathbb{R}^d$$

$$h_\theta(x) := \theta_0 + \theta_1 x_1 + \cdots + \theta_d x_d = \sum_{i=0}^{d} \theta_i x_i \,,$$

with **parameters (weights)** $\theta = (\theta_0, \theta_1, \ldots, \theta_d)^T \in \mathbb{R}^{d+1}$. In the second equation we set $x_0 := 1$.

Tebasaki example: $d = 1$, $x = (x_1)$

Goal: Determine the “best “parameters for a given trainings set.

1 Supervised learning – Linear Regression

Measure how good parameters are:

For a given training set $\mathcal{T} = \big((x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\big)$ we define the **cost function** by

$$J(\theta) = \frac{1}{2} \sum_{j=1}^{n} (h_\theta(x^{(j)}) - y^{(j)})^2 \,.$$

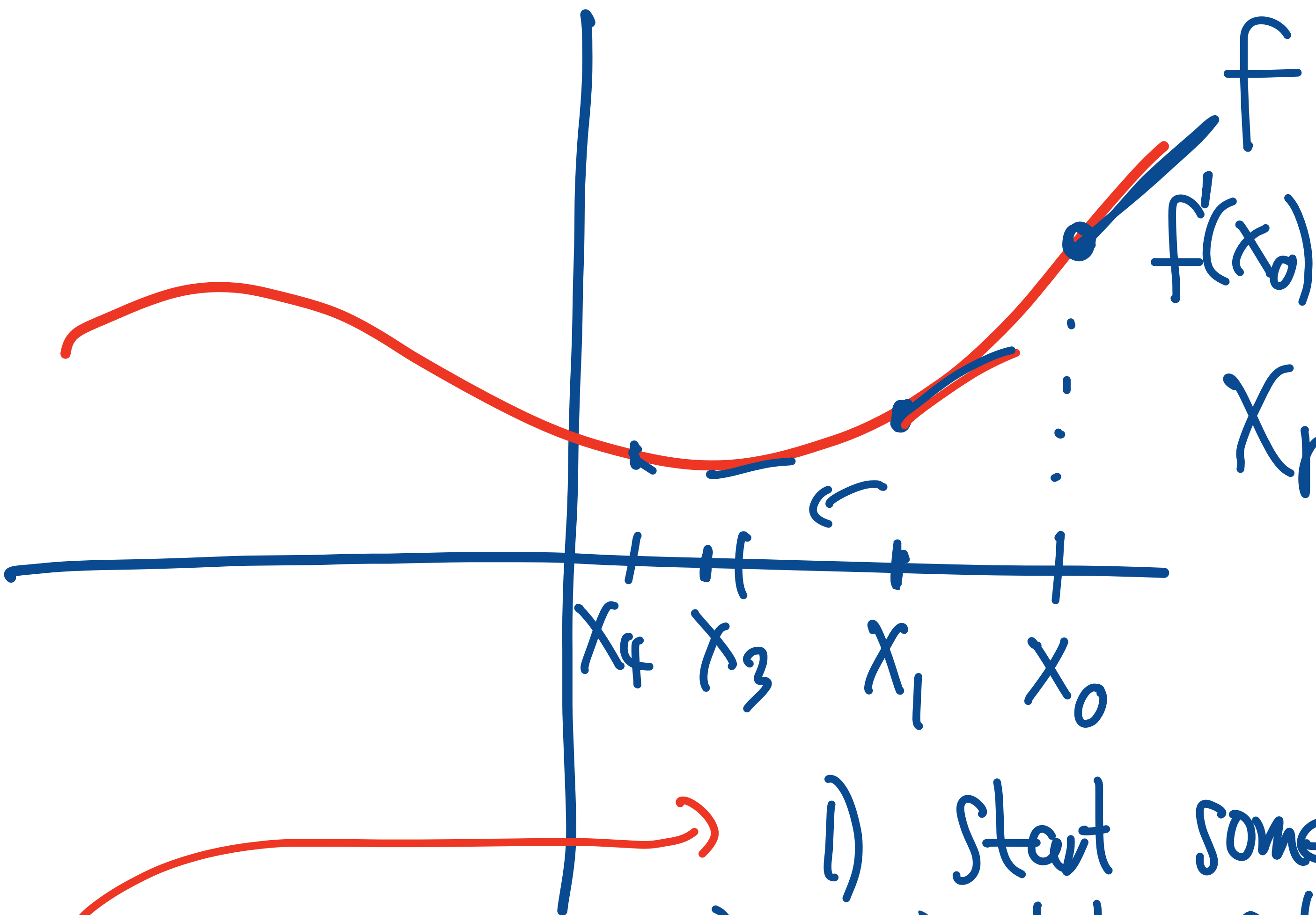
The cost function is a function $J : \mathbb{R}^{d+1} \to \mathbb{R}$, which we want to minimize.

Goal rephrased: Minimize the cost function for a given trainings set.

There are several different choices for cost functions. The above choice corresponds is the “least-squares cost function“.

1 Supervised learning – How to minimize a function?

$f: \mathbb{R} \to \mathbb{R}$



$x_{n+1} = x_n - f'(x_n)$

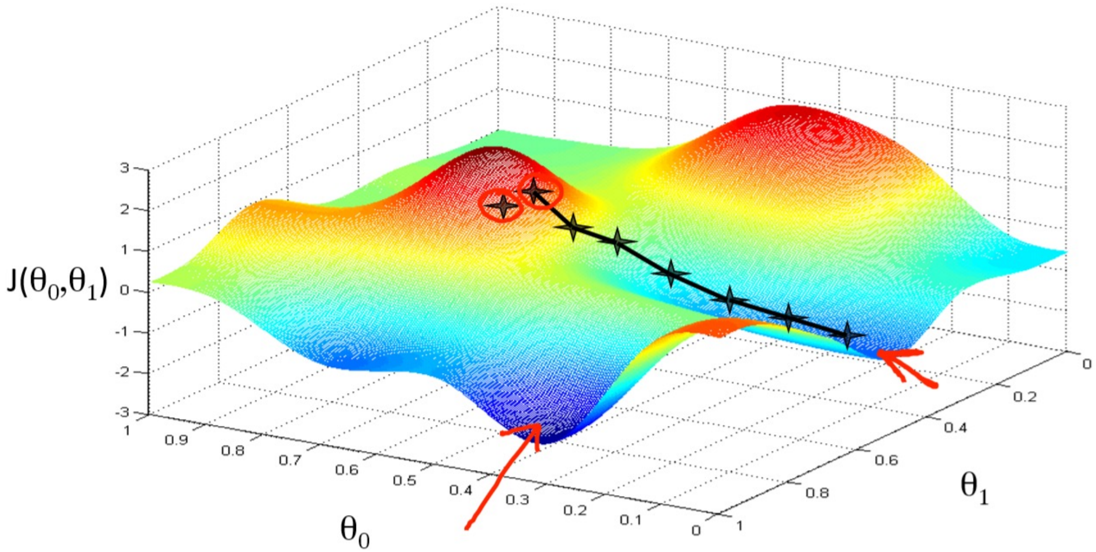
1) Start somewhere (x_0)

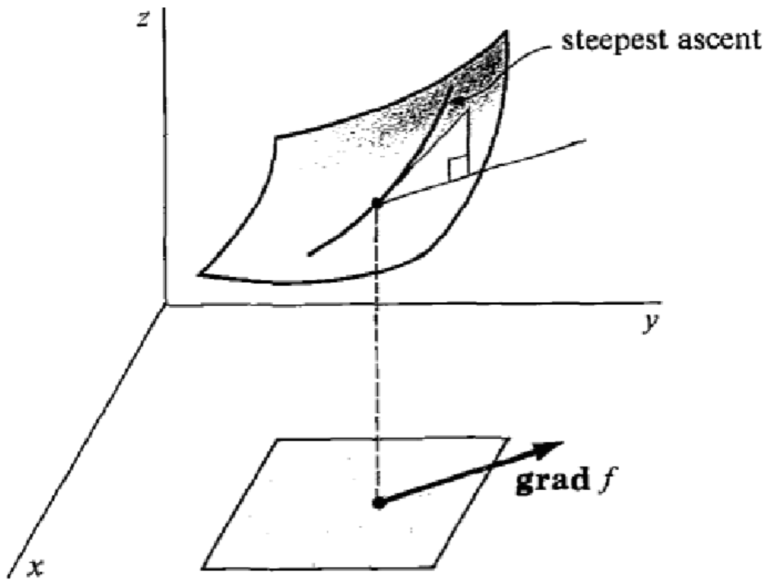
2) Calculate $f'(x_0)$

If $f'(x_0) > 0$ $\quad x_1 = x_0 - 0.1$

1 Supervised learning – Linear Regression

Gradient descent main idea:





Fact: The gradient shows in the direction of the steepest ascent