Machine Learning
Lecture 1: intro to ML

Young && Yandex

girafe ai

Radoslav Neychev

Outline

01 Introduction to Machine Learning, motivation

02 ML thesaurus and notation

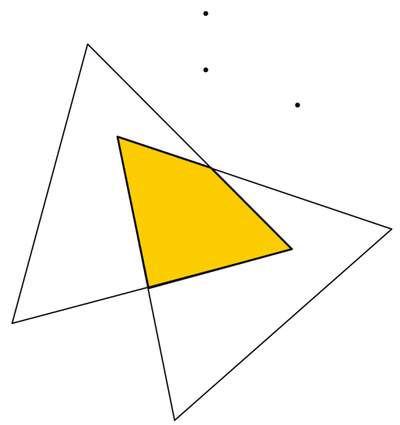
03 Maximum Likelihood Estimation

04 Machine Learning problems overview (selection):

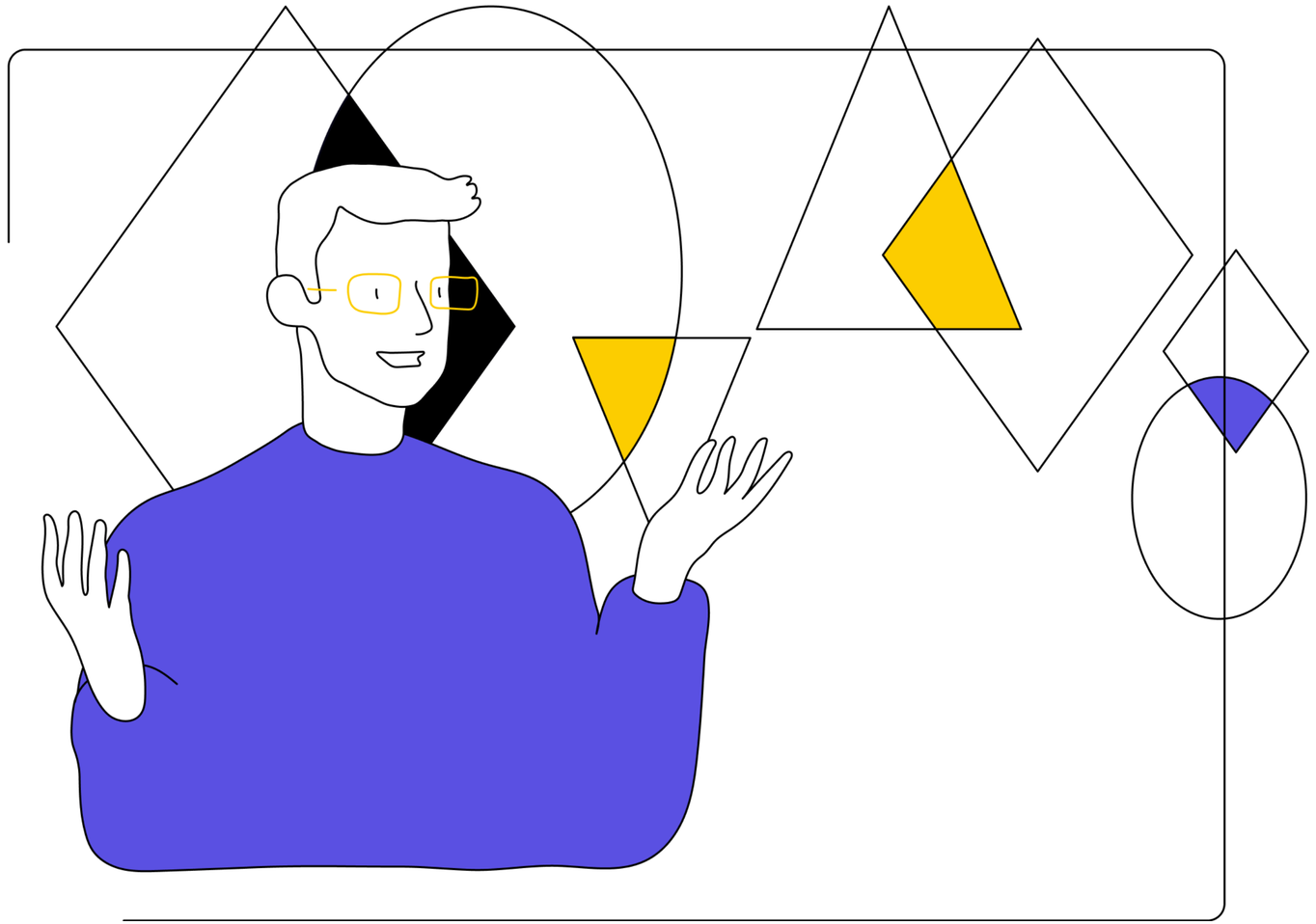
- Classification
- Regression
- Dimensionality reduction

05 Naïve Bayes classifier

06 k Nearest Neighbours (kNN)



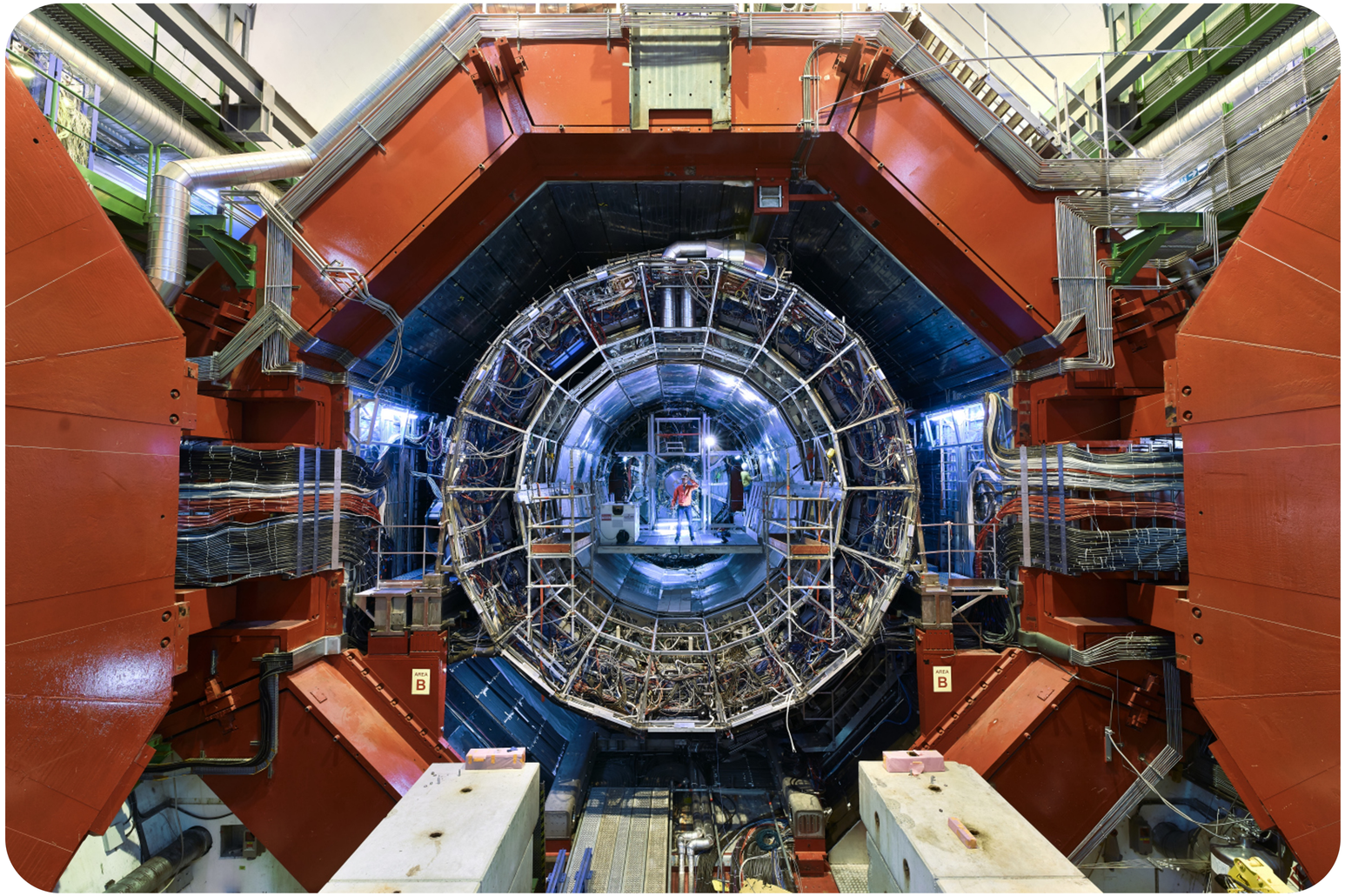
Motivation, historical overview, current state of ML

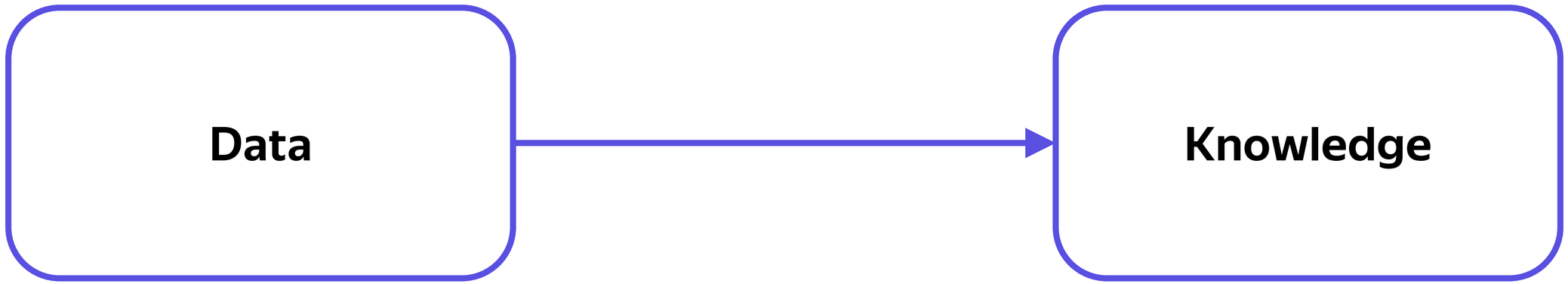


01

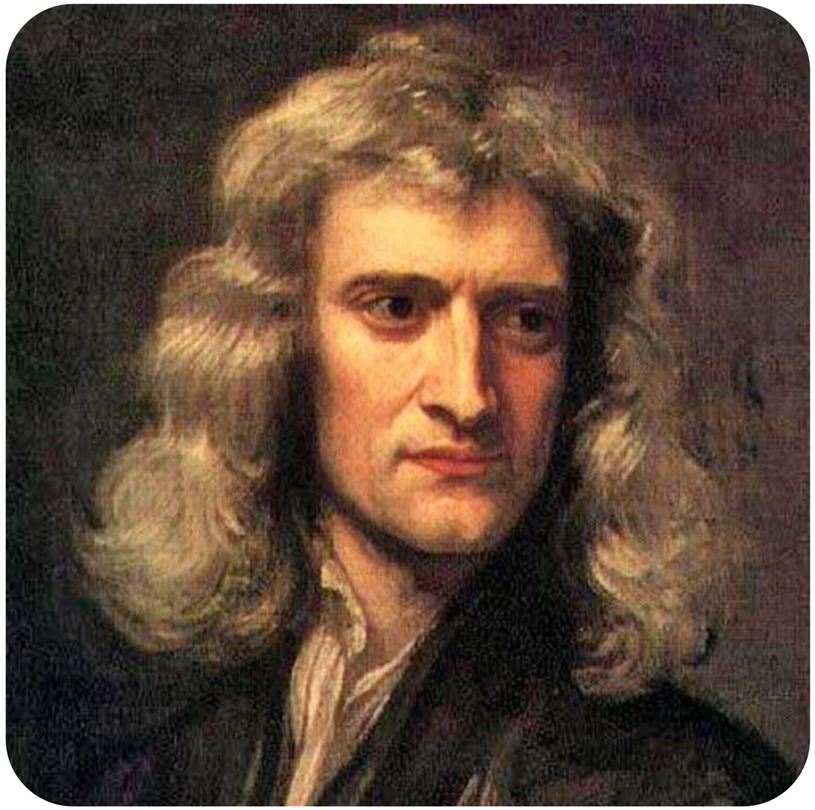
Machine Learning around us

Machine Learning around us




Data
Knowledge

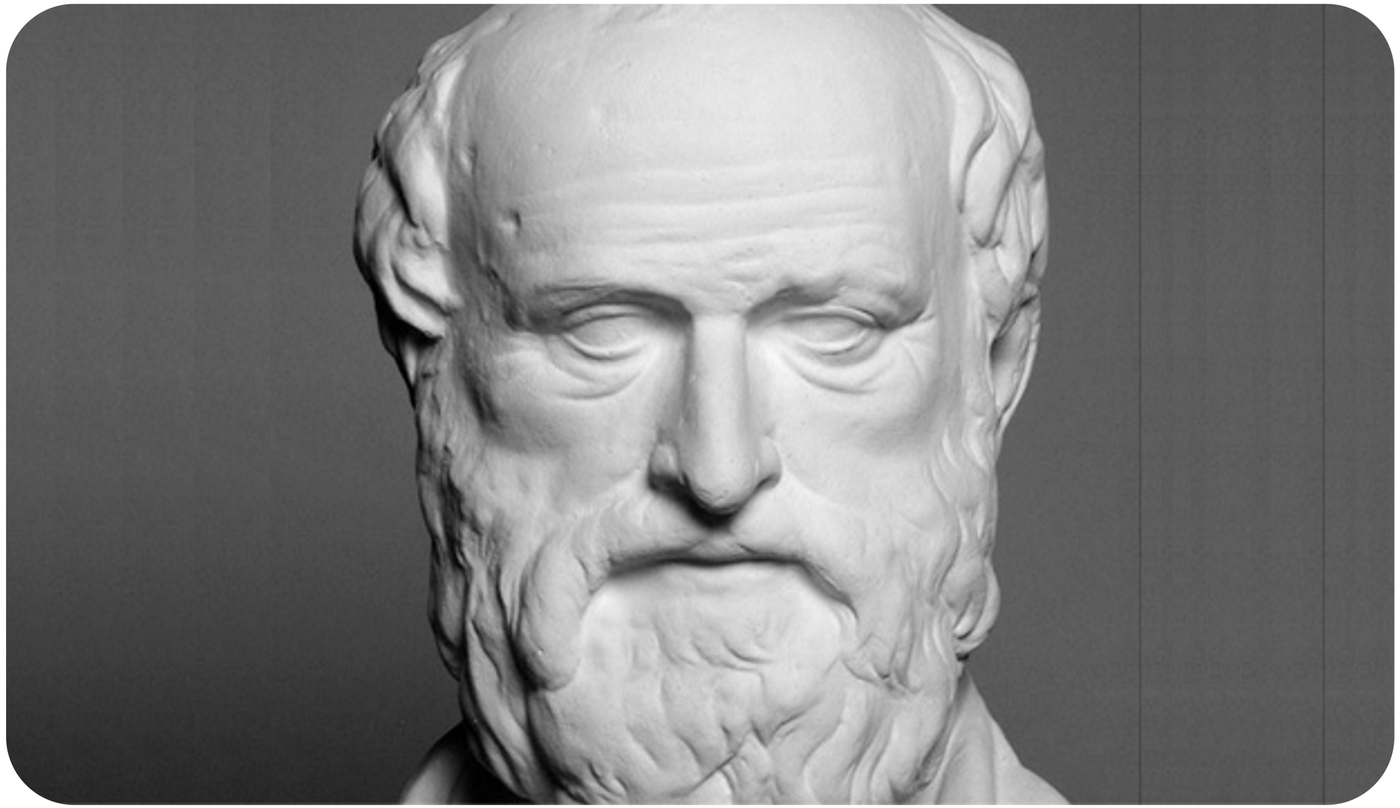
Long time ago



Isaac Newton

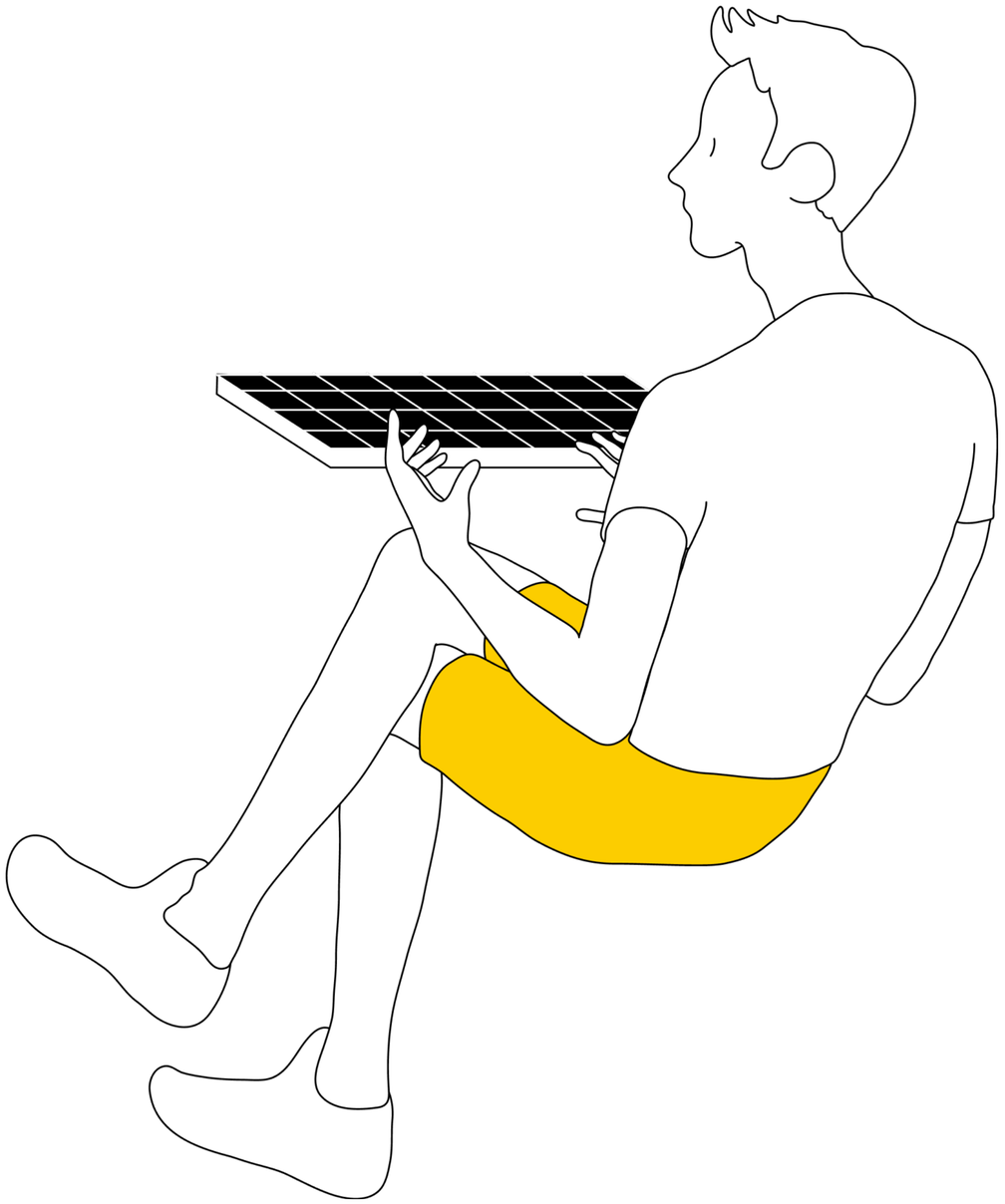
Johannes Kepler

Long before the ML



Eratosthenes

ML thesaurus



02

ML thesaurus

Denote the **dataset**.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

ML thesaurus

Observation (or datum, or data point) is one piece of information.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

In many cases the observations are supposed to be **i.i.d.**

- **independent**
- **identically distributed**

ML thesaurus

Feature (or predictor) represents some special property.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

In many cases the observations are supposed to be **i.i.d.**

- **independent**
- **identically distributed**

ML thesaurus

These all are features

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

ML thesaurus

These all are features

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

ML thesaurus

These all are features

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

ML thesaurus

These all are features

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

ML thesaurus

And even the name is a **feature**

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

ML thesaurus

The **design matrix** contains all the features and observations.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

Features can even be multidimensional, we will discuss it later in this course.

ML thesaurus

Target represents the information we are interested in.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

Target can be either a **number** (real, integer, etc.) – for **regression** problem

ML thesaurus

Target represents the information we are interested in.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

Or a **label** – for **classification** problem

ML thesaurus

Target represents the information we are interested in.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

Mark can be treated as a label too (due to finite number of labels: 1 to 5). We will discuss it later.

ML thesaurus

Further we will work with the numerical target (mark)

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)
John	22	5	4	Brown	English	5
Aahna	17	4	5	Brown	Hindi	4
Emily	25	5	5	Blue	Chinese	5
Michael	27	3	4	Green	French	5
Some student	23	3	3	NA	Esperanto	2

ML thesaurus

The **prediction** contains values we predicted using some **model**.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Predicted (mark)
John	22	5	4	Brown	English	5	4.5
Aahna	17	4	5	Brown	Hindi	4	4.5
Emily	25	5	5	Blue	Chinese	5	5
Michael	27	3	4	Green	French	5	3.5
Some student	23	3	3	NA	Esperanto	2	3

One could notice that prediction just averages of Statistics and Python marks.
So our **model** can be represented as follows:

$$\widehat{\text{mark}}_{\text{ML}} = \frac{1}{2}\text{mark}_{\text{Python}} + \frac{1}{2}\text{mark}_{\text{Statistics}}$$

ML thesaurus

The **prediction** contains values we predicted using some **model**.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Predicted (mark)
John	22	5	4	Brown	English	5	1
Aahna	17	4	5	Brown	Hindi	4	5
Emily	25	5	5	Blue	Chinese	5	2
Michael	27	3	4	Green	French	5	4
Some student	23	3	3	NA	Esperanto	2	3

Different models can provide different predictions:

$$\widehat{\text{mark}}_{\text{ML}} = \text{random}(\text{integer from } [1; 5])$$

ML thesaurus

The **prediction** contains values we predicted using some **model**.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Predicted (mark)
John	22	5	4	Brown	English	5	1
Aahna	17	4	5	Brown	Hindi	4	5
Emily	25	5	5	Blue	Chinese	5	2
Michael	27	3	4	Green	French	5	4
Some student	23	3	3	NA	Esperanto	2	3

Different models can provide different predictions.
Usually some **hypothesis** lies beneath the model choice.

ML thesaurus

Loss function measures the error rate of our model.

Mean Squared Error (where **y** is vector of targets):

$$\mathrm{MSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{n}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Square deviation	Target (mark)	Predicted (mark)
16	5	1
1	4	5
9	5	2
1	5	4
1	2	3

ML thesaurus

Loss function measures the error rate of our model.

Mean Squared Error (where **y** is vector of targets):

$$\mathrm{MAE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{n}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_1 = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Absolute deviation	Target (mark)	Predicted (mark)
4	5	1
1	4	5
3	5	2
1	5	4
1	2	3

ML thesaurus

To learn something, our **model** needs some degrees of freedom:

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Predicted (mark)
John	22	5	4	Brown	English	5	4.5
Aahna	17	4	5	Brown	Hindi	4	4.5
Emily	25	5	5	Blue	Chinese	5	5
Michael	27	3	4	Green	French	5	3.5
Some student	23	3	3	NA	Esperanto	2	3

$$\widehat{\text{mark}}_{\text{ML}} = \frac{1}{2}\text{mark}_{\text{Python}} + \frac{1}{2}\text{mark}_{\text{Statistics}}$$

ML thesaurus

To learn something, our **model** needs some degrees of freedom:

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Predicted (mark)
John	22	5	4	Brown	English	5	4.447
Aahna	17	4	5	Brown	Hindi	4	4.734
Emily	25	5	5	Blue	Chinese	5	5.101
Michael	27	3	4	Green	French	5	3.714
Some student	23	3	3	NA	Esperanto	2	3.060

$$\widehat{\text{mark}}_{\text{ML}} = w_1 \cdot \text{mark}_{\text{Statistics}} + w_2 \cdot \text{mark}_{\text{Python}}$$

ML thesaurus

To learn something, our **model** needs some degrees of freedom:

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Predicted (mark)
John	22	5	4	Brown	English	5	1
Aahna	17	4	5	Brown	Hindi	4	5
Emily	25	5	5	Blue	Chinese	5	2
Michael	27	3	4	Green	French	5	4
Some student	23	3	3	NA	Esperanto	2	3

$$\widehat{\text{mark}}_{\text{ML}} = \text{random}(\text{integer from } [1; 5])$$

ML thesaurus

Last term we should learn for now is **hyperparameter**.

Hyperparameter should be fixed before our model starts to work with the data.

We will discuss it later with kNN as an example.

ML thesaurus

Recap:

Dataset

Observation (datum)

Feature

Design matrix

Target

Prediction

Model

Loss function

Parameter

Hyperparameter

Machine Learning problems overview

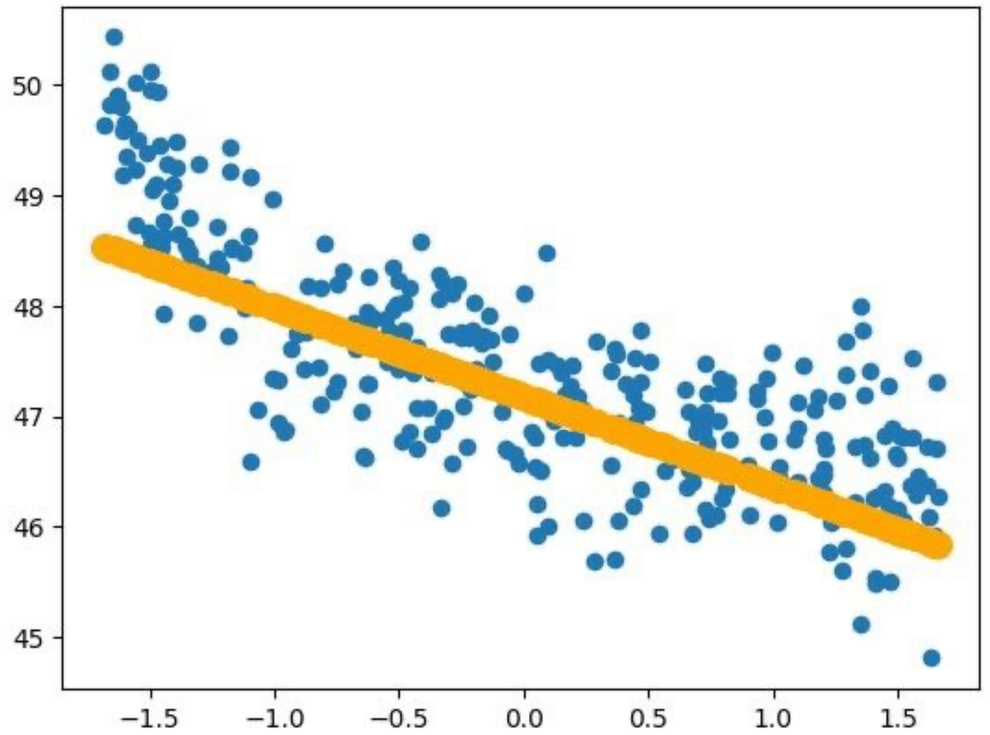
03

Supervised learning problem statement

Let’s denote:

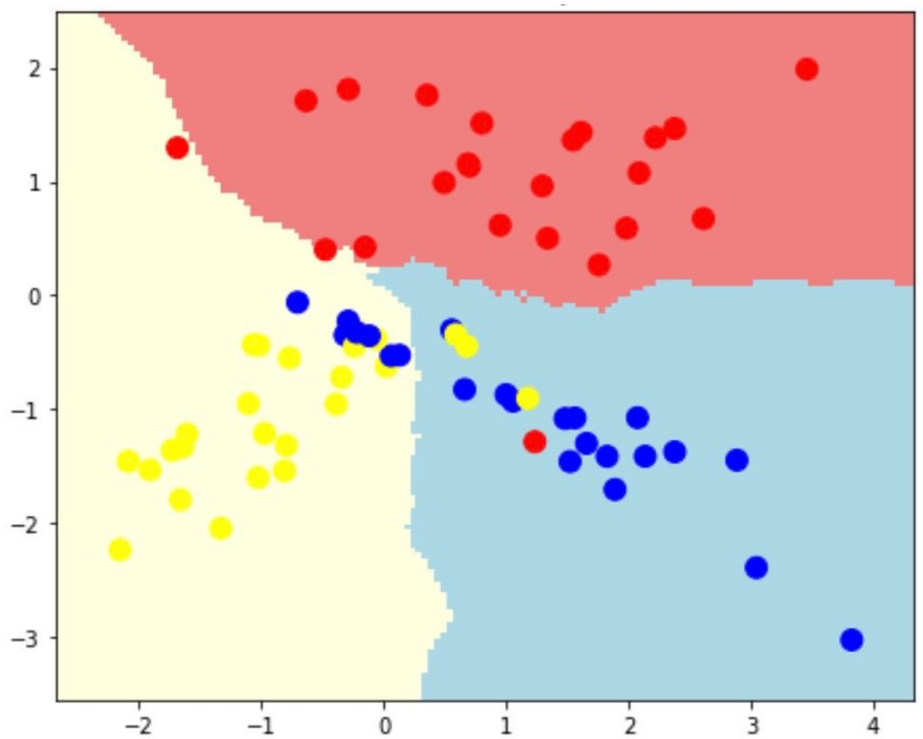
- Training set $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$,where
  - $\boldsymbol{x}^{(i)} \in \mathbb{R}^p, y^{(i)} \in \mathbb{R}$ for regression
  - $\boldsymbol{x}^{(i)} \in \mathbb{R}^p, y^{(i)} \in \{C_1, \ldots, C_K\}$ for classification
- Model $f(\boldsymbol{x})$ predicts some value for every object
- Loss function $L(\boldsymbol{x}, y, f)$ that should be minimized

- **Regression problem**



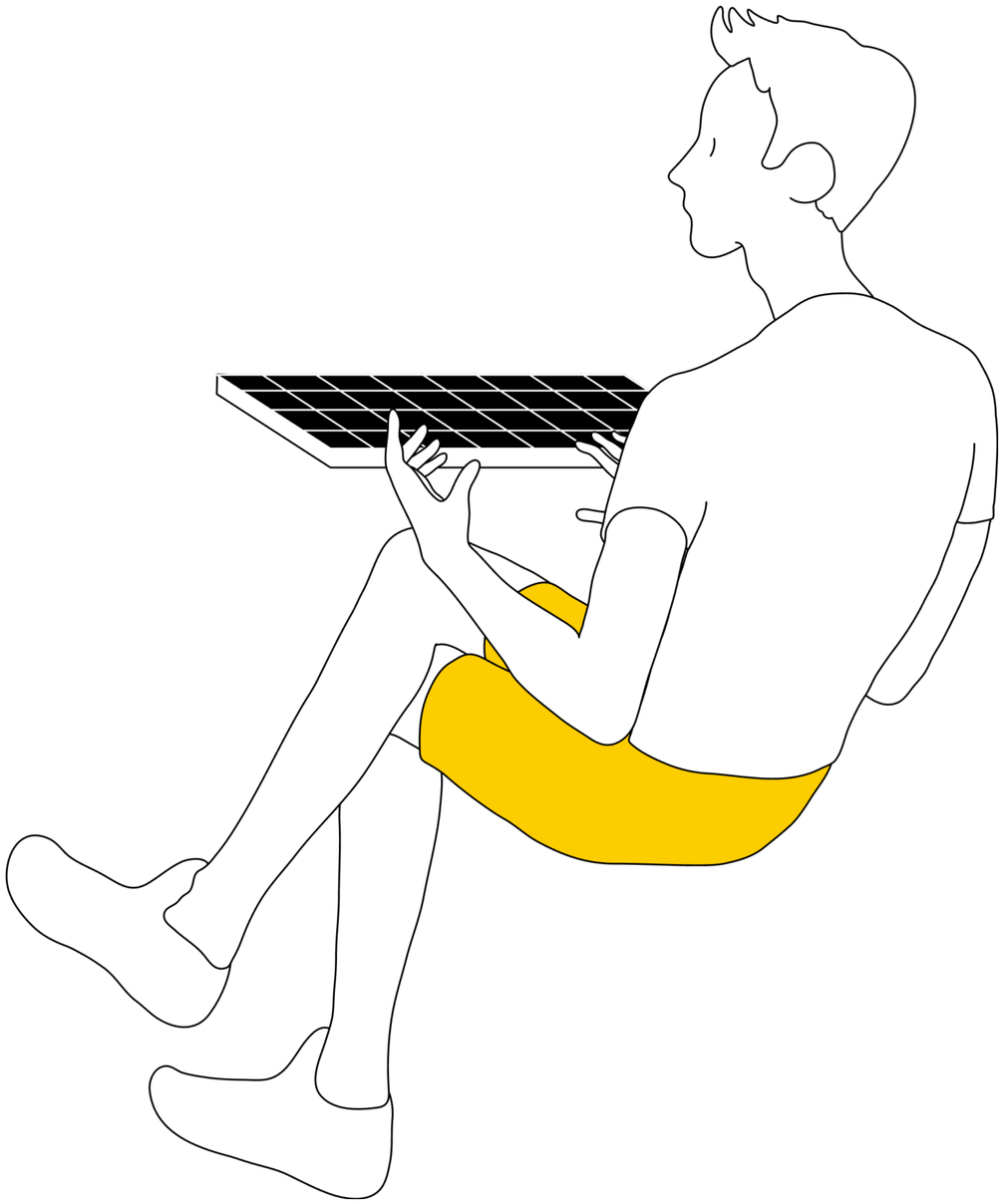
$$y = \boldsymbol{w}^\top \boldsymbol{x} + b = \sum_{i=1}^{p} w_i x_i + b$$

- Regression problem
- **Classification problem**



$$y = \mathrm{sign}\big(\sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)\big)$$

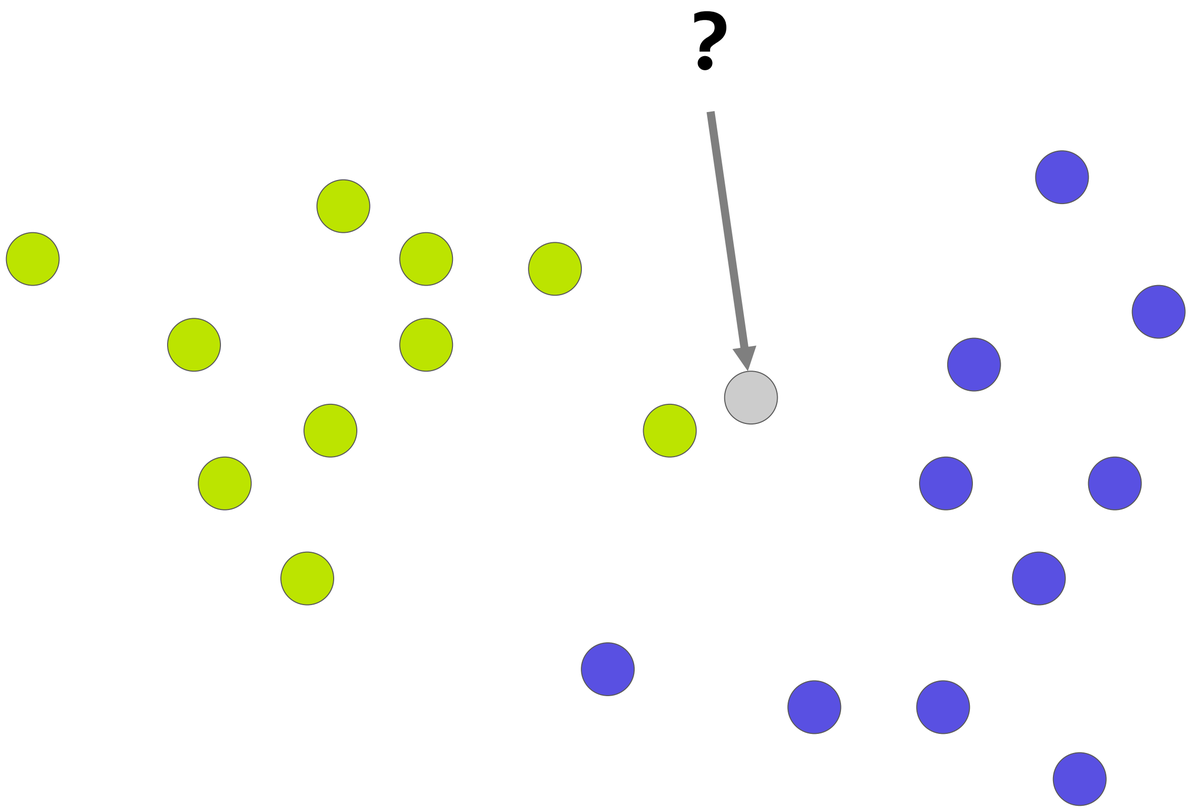
kNN – k Nearest Neighbours



04

kNN – k Nearest Neighbours

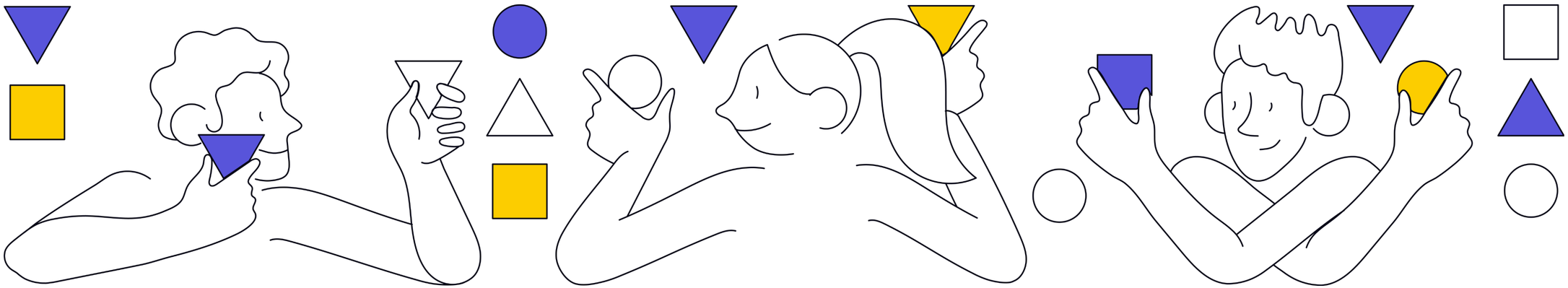
kNN – k Nearest Neighbours



k Nearest Neighbors

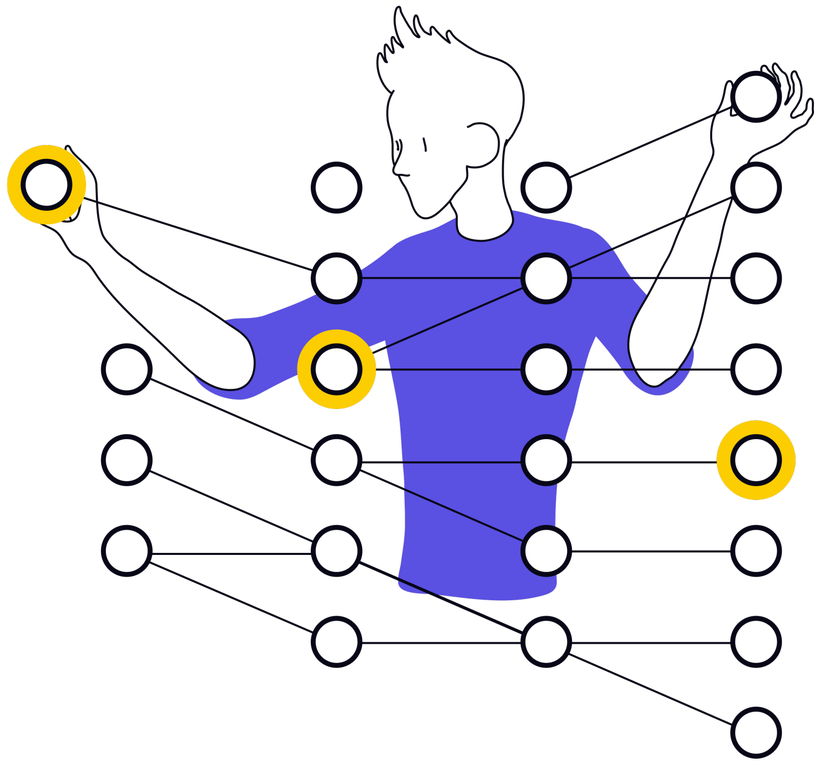
Given a new observation:

1. Calculate the distance to each of the samples in the dataset.
2. Select samples from the dataset with the minimal distance to them.
3. The label of the new observation will be the most frequent label among those nearest neighbors.

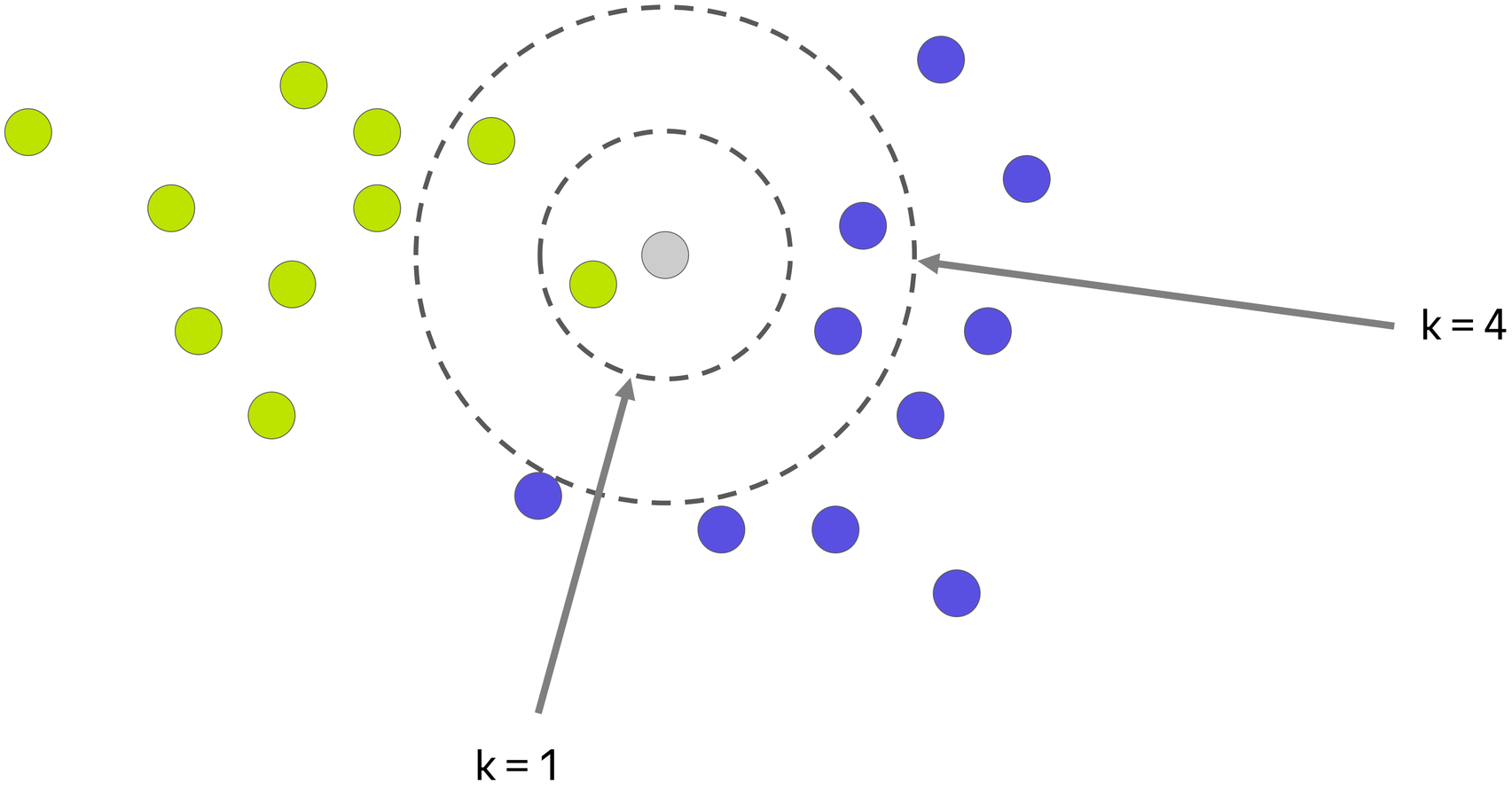


How to make it better?

- The number of neighbors k (it is a **hyperparameter**)

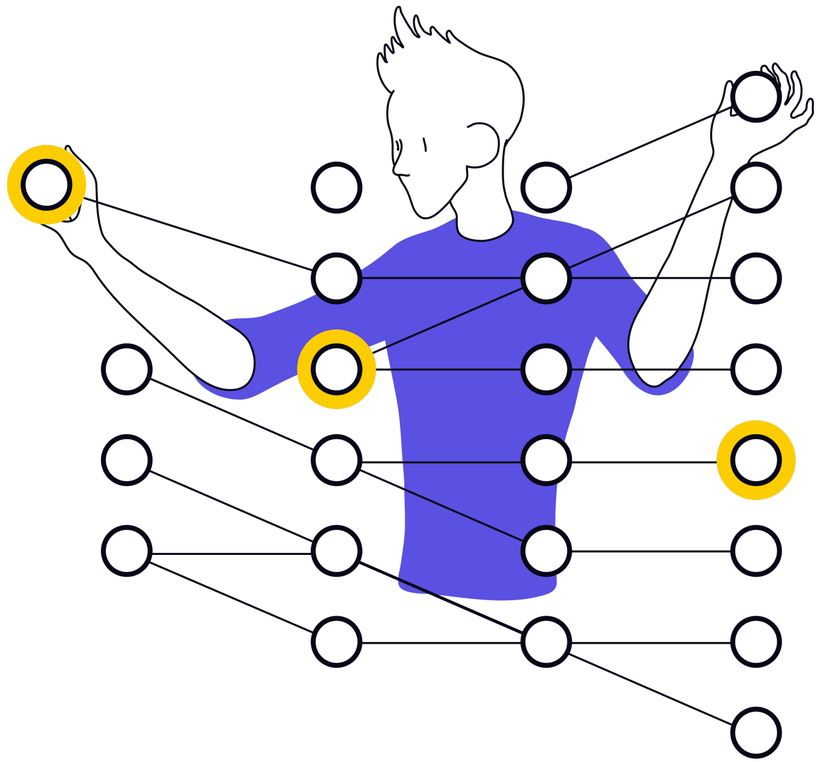


kNN - k Nearest Neighbours

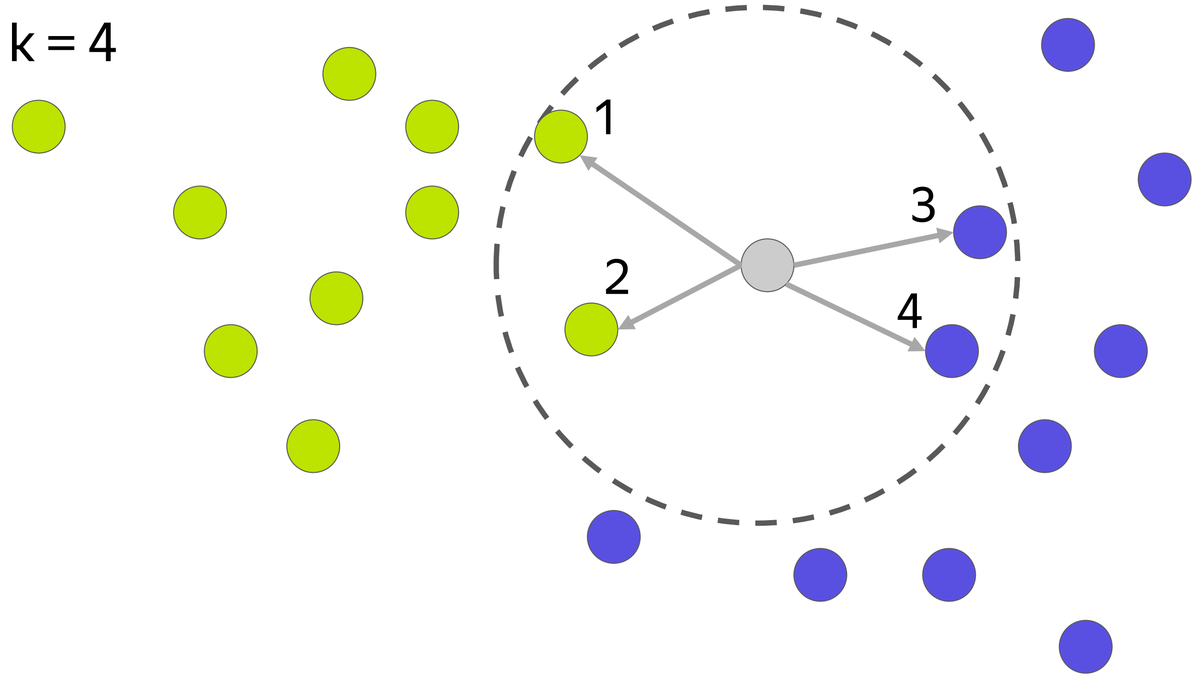


How to make it better?

- The number of neighbors k (it is a **hyperparameter**)
- The distance measure between samples
  a) Hamming
  b) Euclidean
  c) cosine
  d) Minkowski distances
  e) etc.
- Weighted neighbours

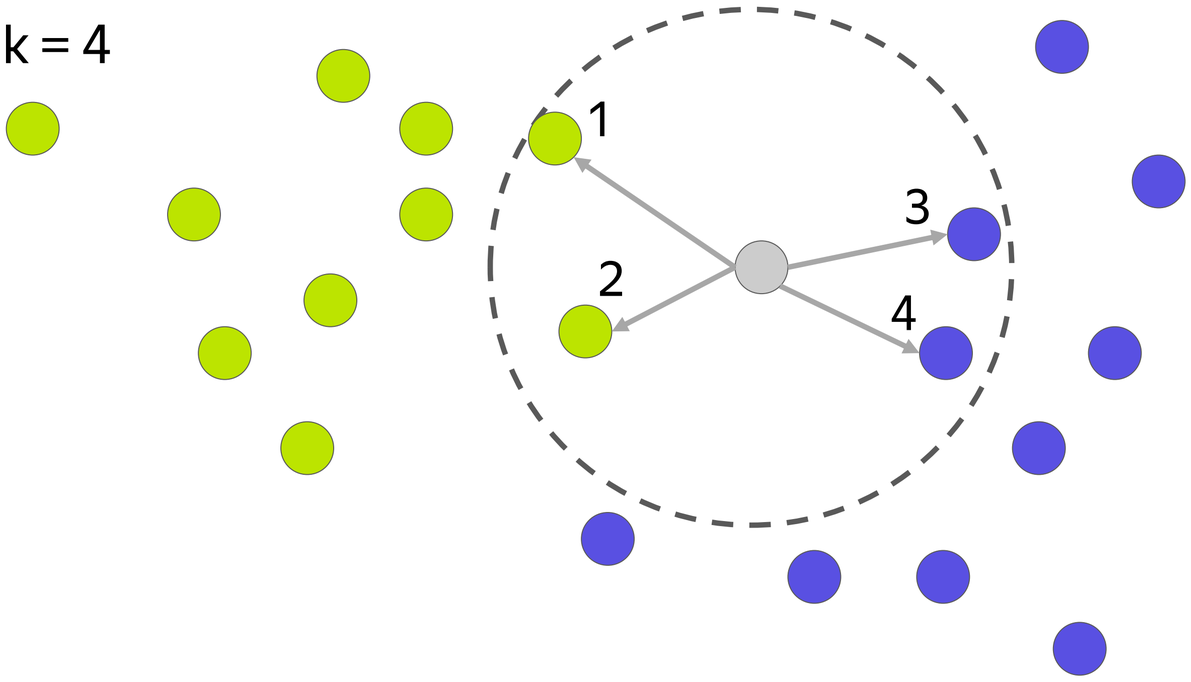


Weighted kNN



Weighted kNN

k = 4

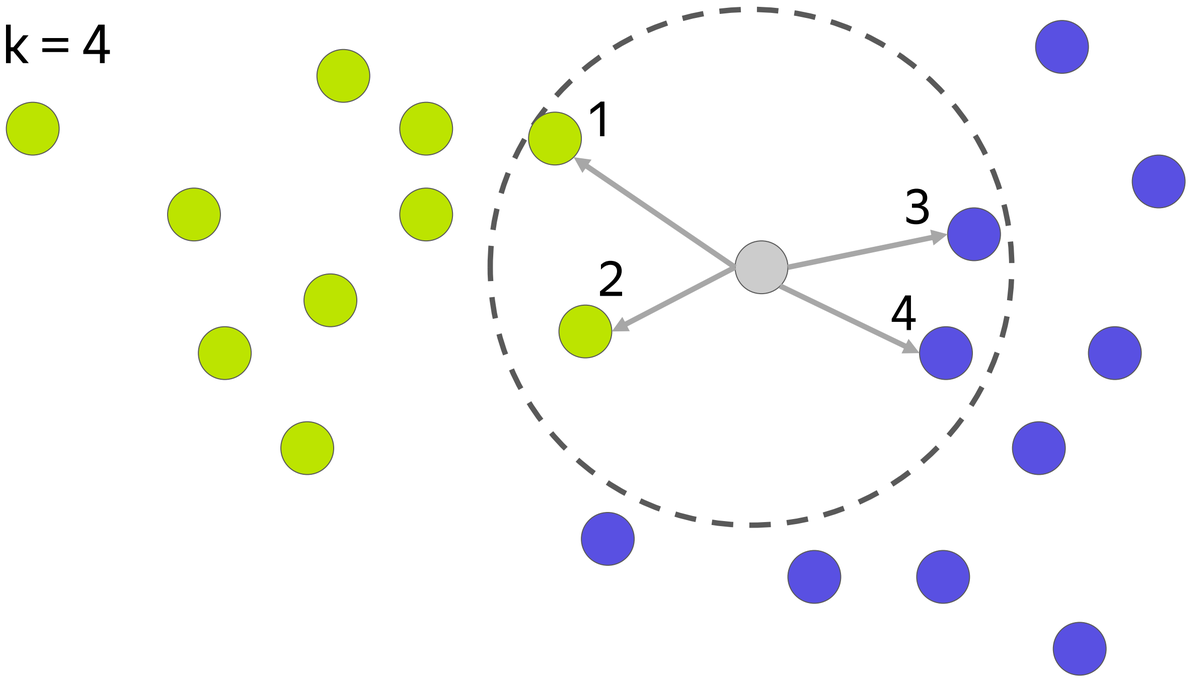


- Weights can be adjusted according to the neighbors order,

$$w(\boldsymbol{x}^{(i)}) = w_i$$

Weighted kNN

k = 4



- Weights can be adjusted according to the neighbors order,

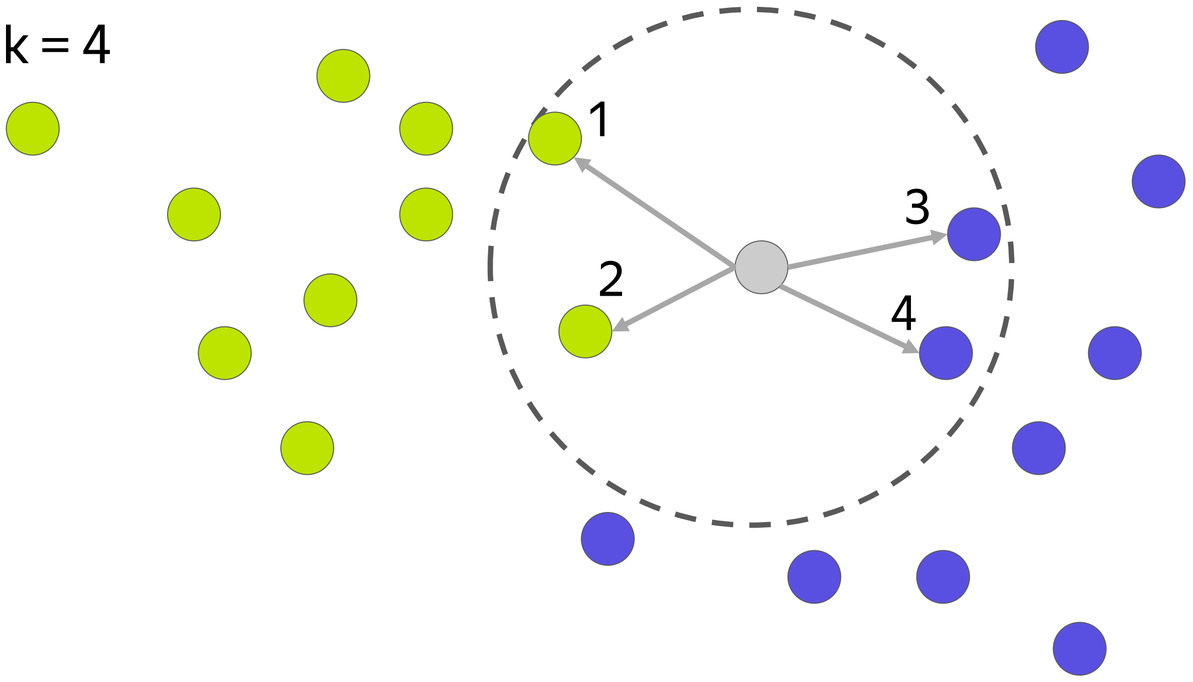
$$w(\boldsymbol{x}^{(i)}) = w_i$$

- or on the distance itself

$$w(\boldsymbol{x}^{(i)}) = w\big(d(\boldsymbol{x}^{(i)}, \boldsymbol{x})\big)$$

Weighted kNN

k = 4



- Weights can be adjusted according to the neighbors order,

$$w(\boldsymbol{x}^{(i)}) = w_i$$

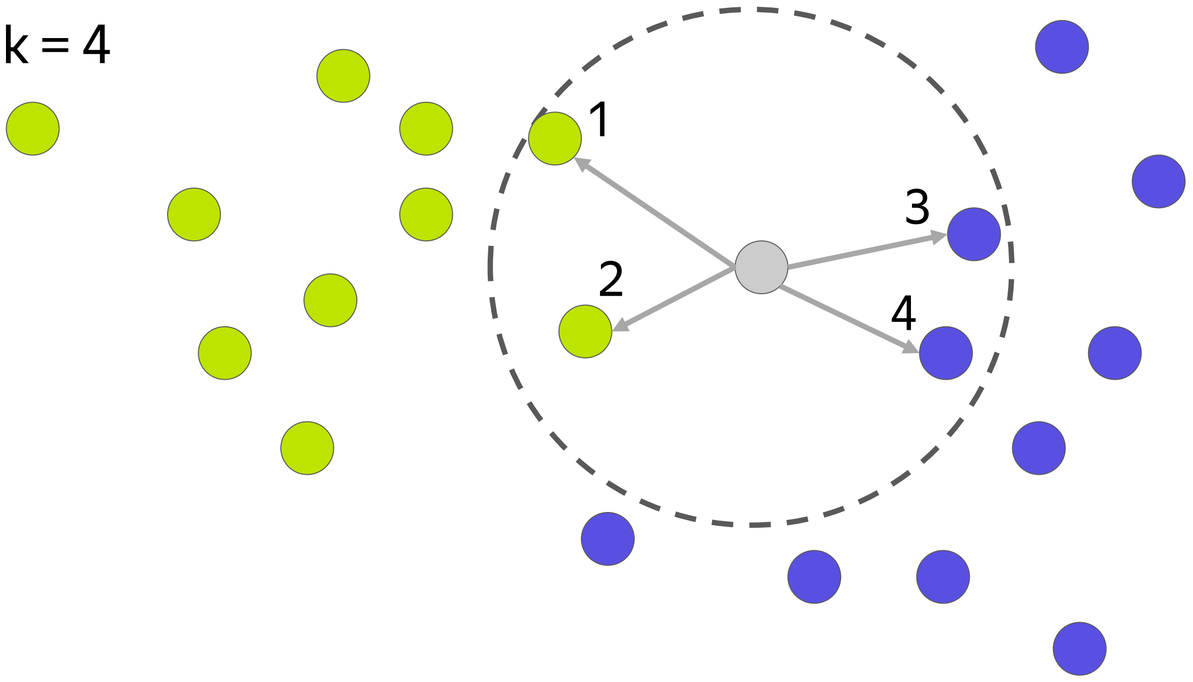
- or on the distance itself

$$w(\boldsymbol{x}^{(i)}) = w\big(d(\boldsymbol{x}^{(i)}, \boldsymbol{x})\big)$$

$$p_{\text{green}} = \frac{w(\boldsymbol{x}^{(1)}) + w(\boldsymbol{x}^{(2)})}{w(\boldsymbol{x}^{(1)}) + w(\boldsymbol{x}^{(2)} + w(\boldsymbol{x}^{(3)}) + w(\boldsymbol{x}^{(4)})}$$

Weighted kNN

k = 4



- Weights can be adjusted according to the neighbors order,

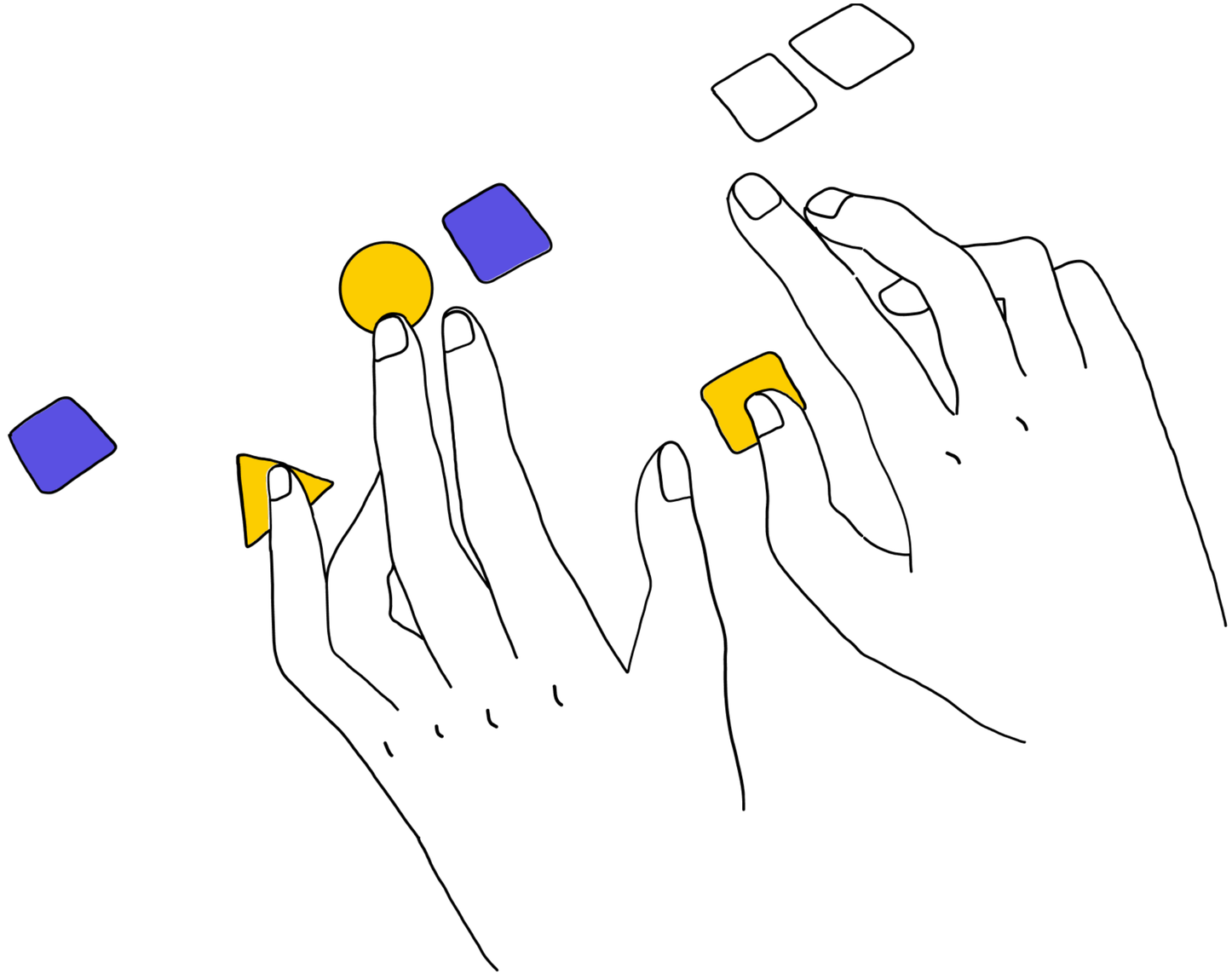
$$w(\boldsymbol{x}^{(i)}) = w_i$$

- or on the distance itself

$$w(\boldsymbol{x}^{(i)}) = w\big(d(\boldsymbol{x}^{(i)}, \boldsymbol{x})\big)$$

$$p_{\text{blue}} = \frac{w(\boldsymbol{x}^{(3)}) + w(\boldsymbol{x}^{(4)})}{w(\boldsymbol{x}^{(1)}) + w(\boldsymbol{x}^{(2)} + w(\boldsymbol{x}^{(3)}) + w(\boldsymbol{x}^{(4)})}$$

Maximum Likelihood Estimation



05

Likelihood

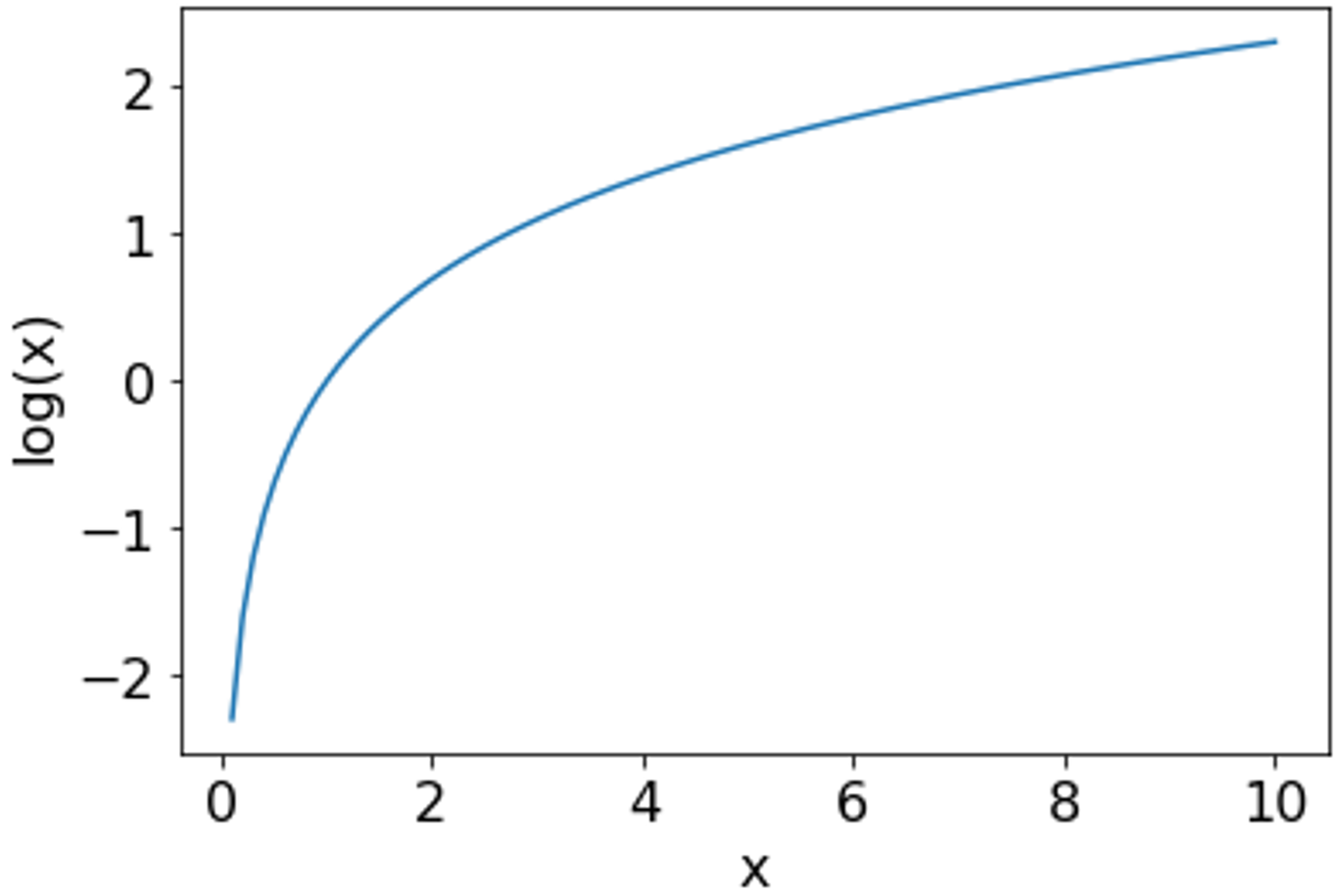
Denote dataset generated by distribution with parameter θ
Likelihood function:

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) = P(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{\theta}) = \prod_{i=1}^{n} P(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}|\boldsymbol{\theta})$$

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) \longrightarrow \max_{\boldsymbol{\theta}}$$

samples should be i.i.d.

Maximum Likelihood Estimation



Likelihood

Denote dataset generated by distribution with parameter θ
Likelihood function:

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) = P(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{\theta}) = \prod_{i=1}^{n} P(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}|\boldsymbol{\theta})$$

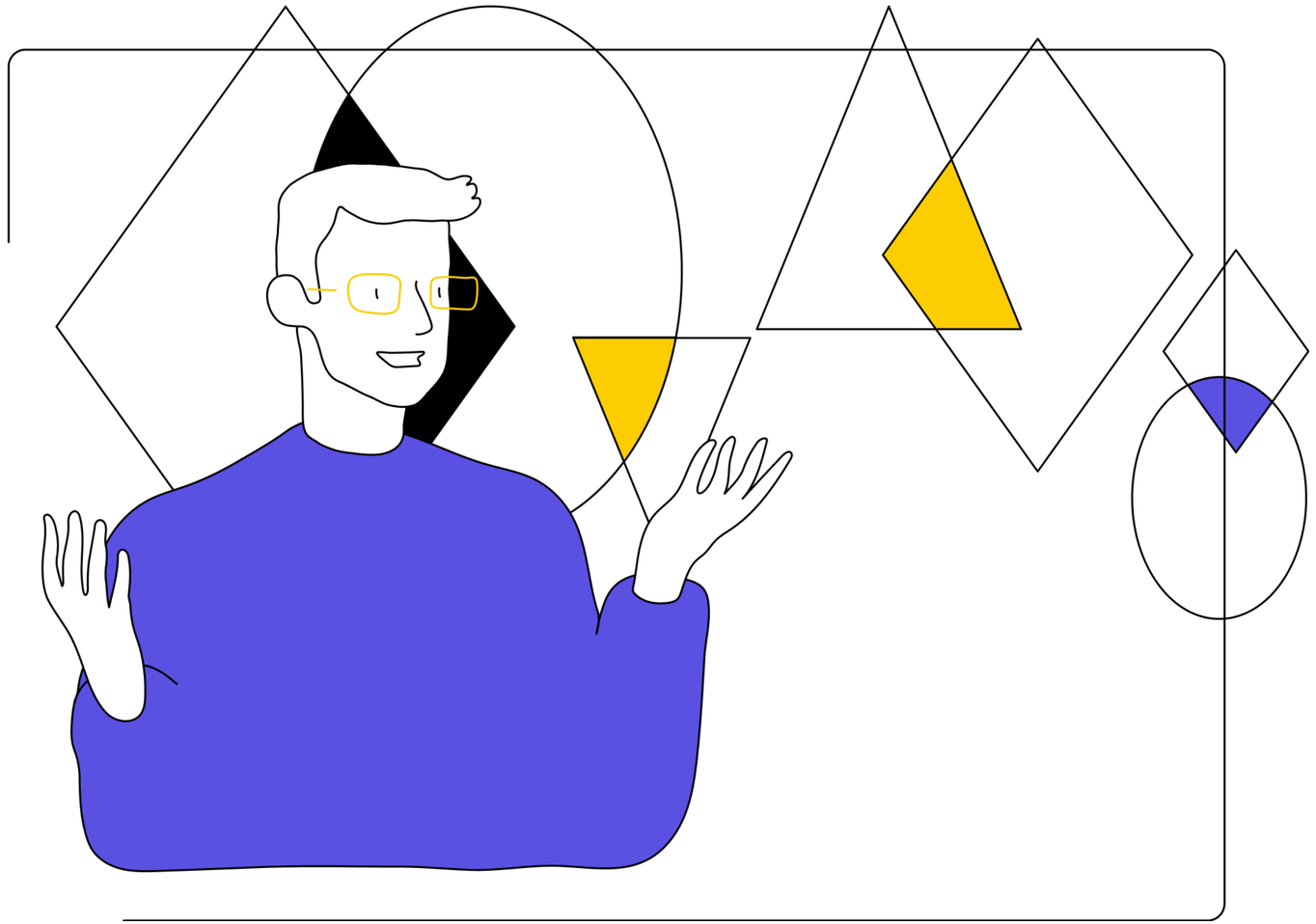
$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) \longrightarrow \max_{\boldsymbol{\theta}}$$

samples should be i.i.d.

equivalent to

$$\log \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n} \log P(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}|\boldsymbol{\theta}) \longrightarrow \max_{\boldsymbol{\theta}}$$

Naïve Bayes classifier



06

Naïve Bayes classifier

Let’s denote:

- Training set $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$, where
  - $\boldsymbol{x}^{(i)} \in \mathbb{R}^p, y^{(i)} \in \{C_1, \ldots, C_K\}$ for K-class classification

Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

or, in our case

$$P(y^{(i)} = C_k|\boldsymbol{x}^{(i)}) = \frac{P(\boldsymbol{x}^{(i)}|y^{(i)} = C_k)P(y^{(i)} = C_k)}{P(\boldsymbol{x}^{(i)})}$$

Naïve Bayes classifier

Let’s denote:

- Training set $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$, where
  - $\boldsymbol{x}^{(i)} \in \mathbb{R}^p, y^{(i)} \in \{C_1, \ldots, C_K\}$ for K-class classification

$$P(y^{(i)} = C_k | \boldsymbol{x}^{(i)}) = \frac{P(\boldsymbol{x}^{(i)} | y^{(i)} = C_k) P(y^{(i)} = C_k)}{P(\boldsymbol{x}^{(i)})}$$

Naïve assumption: features are **independent**

Naïve Bayes classifier

$$P(y^{(i)} = C_k | \boldsymbol{x}^{(i)}) = \frac{P(\boldsymbol{x}^{(i)} | y^{(i)} = C_k) P(y^{(i)} = C_k)}{P(\boldsymbol{x}^{(i)})}$$

Naïve assumption: features are **independent:**

$$P(\boldsymbol{x}^{(i)} | y^{(i)} = C_k) = \prod_{l=1}^{p} P(x_l^{(i)} | y^{(i)} = C_k)$$

Naïve Bayes classifier

$$P(y^{(i)} = C_k|\boldsymbol{x}^{(i)}) = \frac{P(\boldsymbol{x}^{(i)}|y^{(i)} = C_k)P(y^{(i)} = C_k)}{P(\boldsymbol{x}^{(i)})}$$

Optimal class label:

$$C^* = \arg\max_k P(y^{(i)} = C_k|\boldsymbol{x}^{(i)})$$

To find maximum we even do not need the denominator

But we need it to get probabilities

Outro

1

Remember the i.i.d. property

2

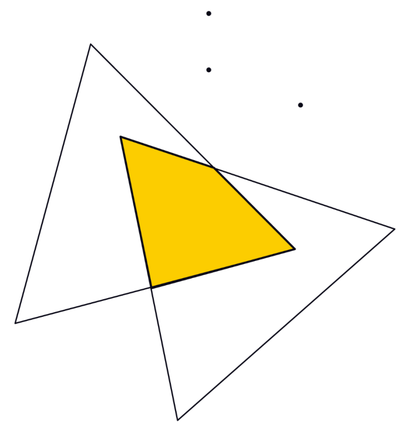
Usually the first dimension corresponds to the batch size, the second (and so on) to the features/time/...

3

Even the naïve assumptions may be suitable in some cases

4

Simple models provide great baselines



Revise

1

Introduction to Machine Learning, motivation

2

ML thesaurus and notation

3

Maximum Likelihood Estimation

4

Machine Learning problems overview (selection):

- Classification
- Regression
- Dimensionality reduction

5

Naïve Bayes classifier

6

k Nearest Neighbours (kNN)

Q&A
Thanks for attention!

Radoslav Neychev