girafe ai

# Machine Learning
# Lecture 1: intro to ML

**Radoslav Neychev**

Spring 2021

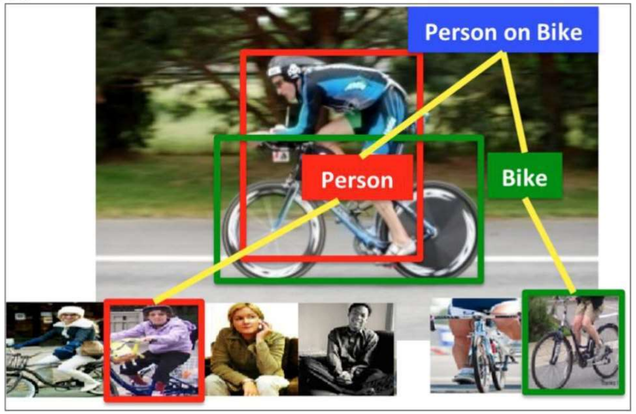
# Outline

1. Introduction to Machine Learning, motivation
2. ML thesaurus and notation
3. Maximum Likelihood Estimation
4. Machine Learning problems overview (selection):
   a. Classification
   b. Regression
   c. Dimensionality reduction
5. Naïve Bayes classifier
6. k Nearest Neighbours (kNN)

# Motivation, historical overview and current state of ML and AI

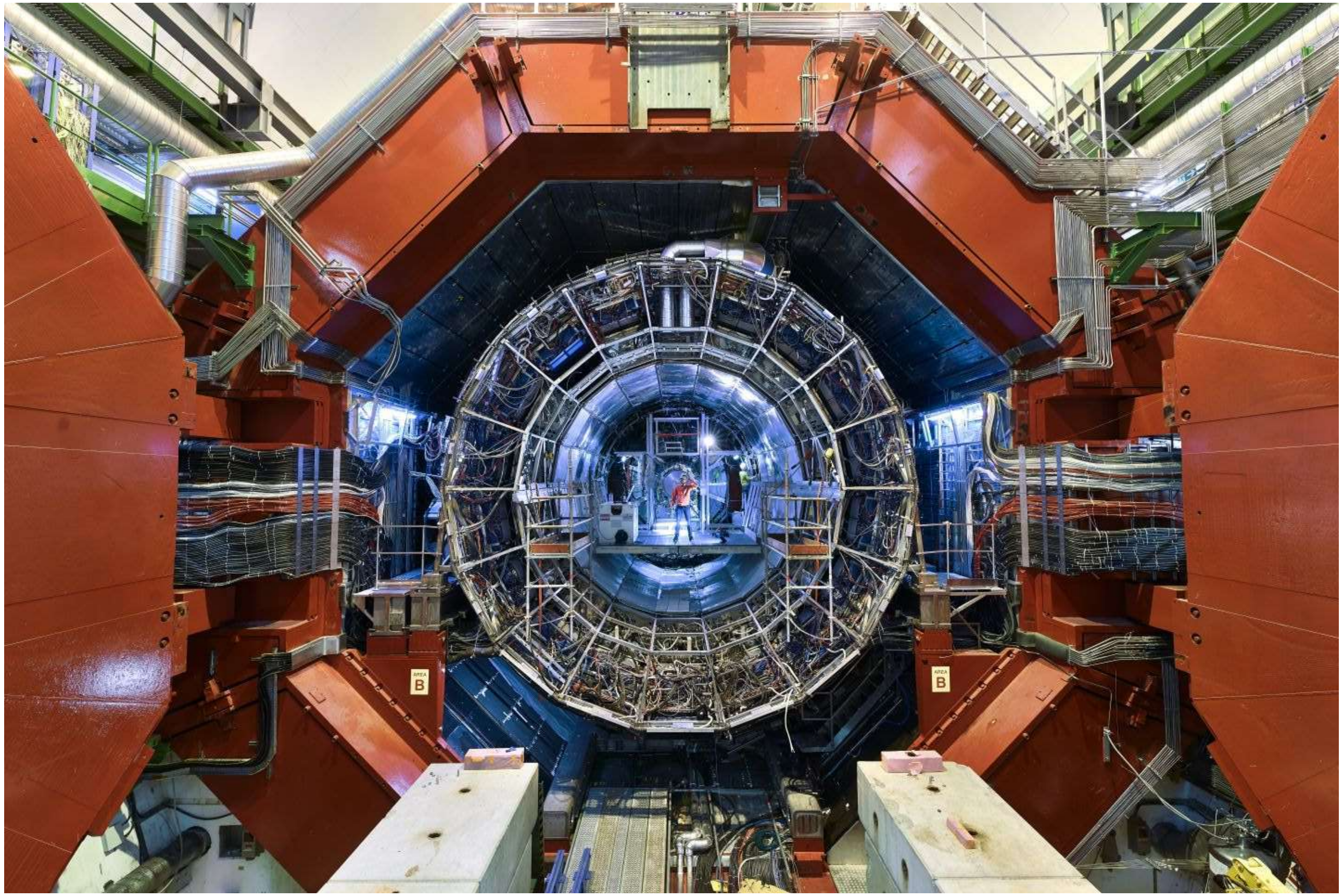
# Machine Learning applications

- Object detection
- Action classification
- Image captioning
- …

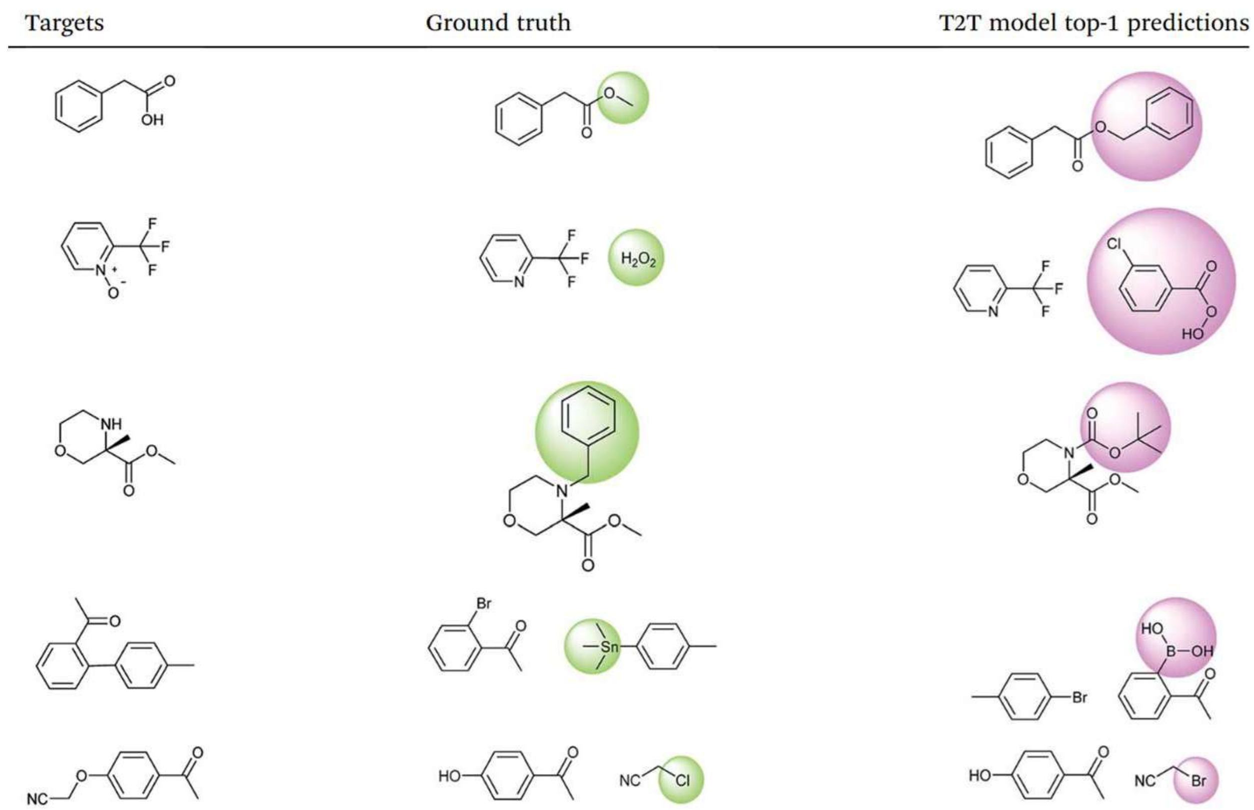



# Machine Learning applications

# Machine Learning applications

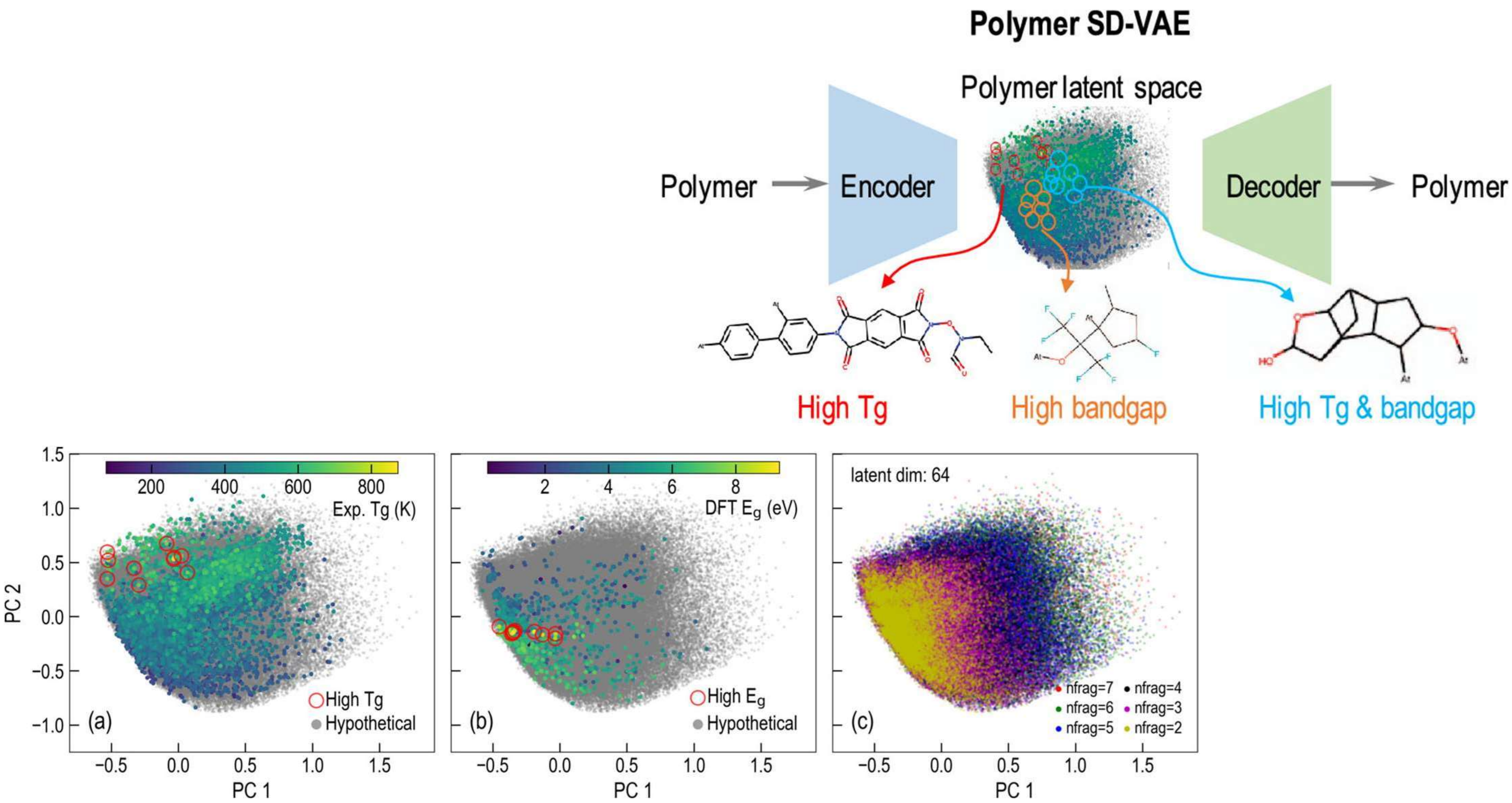


# Neural Machine Translation




DOI: 10.1039/C9RA08535A

# Variational AutoEncoders



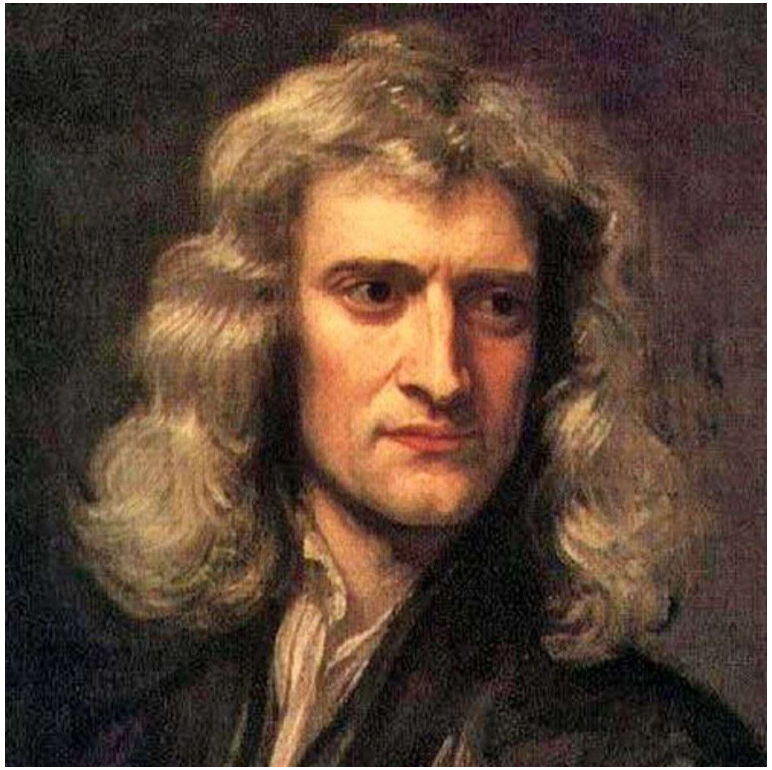
# Prompt engineering

| | |
|---|---|
| General Template | *You are an expert chemist. Given the molecular SMILES, your task is to predict the IUPAC name using your experienced chemical IUPAC name knowledge.* |
| Task-specific Template | *Please strictly follow the format, no other information can be provided.* |
| ICL | *Molecular SMILES: CN(C)c1ncc(-c2ccccc2F)c(C2CCCN(C(=O)CCN3CCOCC3)C2)n1*<br>*Molecular IUPAC name: 1-[3-[2-(dimethylamino)-5-(2-fluorophenyl)pyrimidin-4-yl]piperidin-1-yl]-3-morpholin-4-ylpropan-1-one*<br>*Molecular SMILES: O=C(Nc1cccnc1)c1cn(CC2CCCN(C(=O)C3CC3)C2)c2ccccc12*<br>*Molecular IUPAC name: 1-[[(3R)-1-(cyclopropanecarbonyl)piperidin-3-yl]methyl]-N-pyridin-3-ylindole-3-carboxamide*<br>*Molecular SMILES: CN(C(=O)C1CCCN1C(=O)C(N)Cc1ccc(O)cc1)C(Cc1ccccc1)C(=O)N1CCCC1C(=O)O*<br>*Molecular IUPAC name: 1-[2-[[1-[2-amino-3-(4-hydroxyphenyl)propanoyl]pyrrolidine-2-carbonyl]-methylamino]-3-phenylpropanoyl]pyrrolidine-2-carboxylic acid*<br>*Molecular SMILES: CC1c2ccsc2CCN1CC(=O)Nc1cc(S(=O)(=O)N2CCCCC2)ccc1Cl*<br>*Molecular IUPAC name: N-(2-chloro-5-piperidin-1-ylsulfonylphenyl)-2-[(4R)-4-methyl-6,7-dihydro-4H-thieno[3,2-c]pyridin-5-yl]acetamide*<br>*Molecular SMILES: Cc1cccc(NC(=O)CN2C(=O)c3cc(Cl)c(Cl)cc3C2=O)c1C(=O)N1CCCC1*<br>*Molecular IUPAC name: 2-(5,6-dichloro-1,3-dioxoisoindol-2-yl)-N-[3-methyl-2-(pyrrolidine-1-carbonyl)phenyl]acetamide* |
| Question | *Molecular SMILES: CC(C)(C)OC(=O)NC1CCN(C(=O)CN2CCOCC2C(=O)Nc2cc(Cl)cc3c2[nH]c2cnccc23)CC1*<br>*Molecular IUPAC name:* |
| Answer | 4-[3-(4-chloro-2H-indazol-3-yl)-1H-pyrrolo[2,3-b]pyridin-5-yl]morpholine-2-carbonyl]amino-N-(2-methylpropyl)benzamide |

Figure 4: An ICL prompt example for smiles2iupac prediction


Data
Knowledge

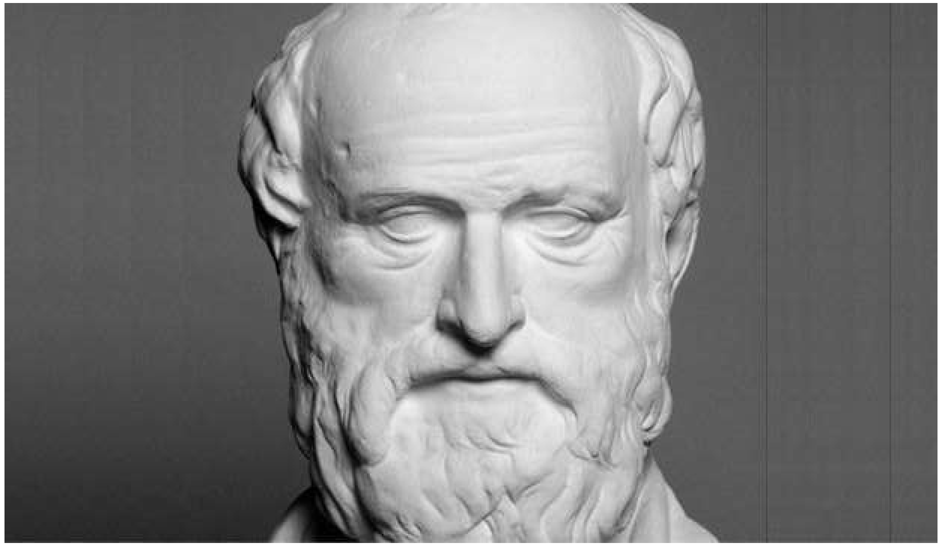
# Long before the ML



Isaac Newton

Johannes Kepler

# Long before the ML



Eratosthenes

# 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 |

***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 |

# 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 |

*(despite it might be not informative)*

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

***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.*

# 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:

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

# 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 |

*Different models can provide different predictions:*

$$\hat{\mathrm{mark}}_{ML} = \frac{1}{2}\mathrm{mark}_{Statistics} + \frac{1}{2}\mathrm{mark}_{Python}$$

# 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:*

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

# 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.

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

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

$$MSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 = \frac{1}{N}\sum_i (y_i - \hat{y}_i)^2$$

# ML thesaurus

***Loss function*** measures the error rate of our model.

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

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

$$MAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\|\mathbf{y} - \hat{\mathbf{y}}\|_1 = \frac{1}{N}\sum_i |y_i - \hat{y}_i|$$

# 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 |

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

# 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 | 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 |

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

# 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 |

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

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

# Maximum Likelihood Estimation

# Likelihood

Denote dataset generated by distribution with parameter $\theta$

***Likelihood*** function:

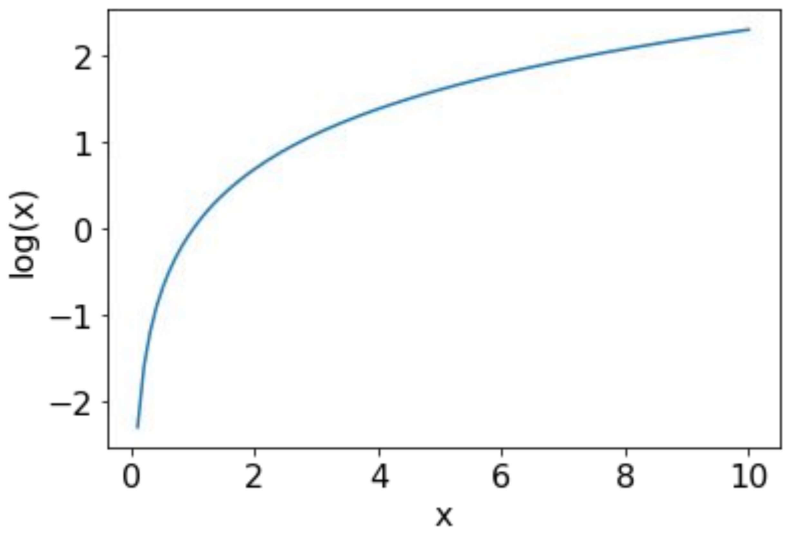
$$L(\theta|X,Y) = P(X,Y|\theta)$$

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

**samples should be i.i.d.**

$$L(\theta|X,Y) = P(X,Y|\theta) = \prod_i P(x_i, y_i|\theta)$$

# Maximum Likelihood Estimation



# Likelihood

Denote dataset generated by distribution with parameter $\theta$

***Likelihood*** function:

$$L(\theta|X,Y) = P(X,Y|\theta)$$

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

**samples should be i.i.d.**

$$L(\theta|X,Y) = P(X,Y|\theta) = \prod_i P(x_i, y_i|\theta)$$

equivalent to

$$\log L(\theta|X,Y) = \sum_i \log P(x_i, y_i|\theta) \longrightarrow \max_{\theta}$$

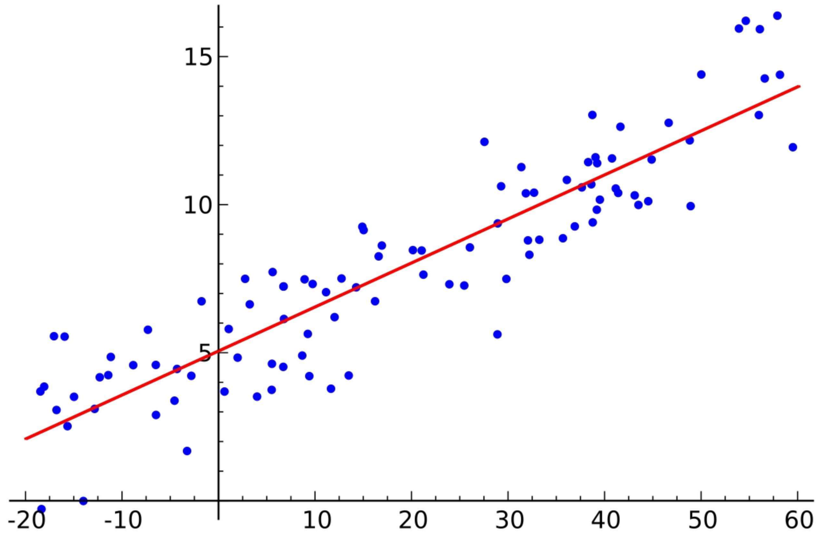
# Machine Learning problems overview

# Supervised learning problem statement

Let's denote:

- Training set $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ , where
  - $(\mathbf{x} \in \mathbb{R}^p,\ y \in \mathbb{R})$ for regression
  - $\mathbf{x}_i \in \mathbb{R}^p$ , $y_i \in \{+1, -1\}$ for binary classification
- Model $f(\mathbf{x})$ predicts some value for every object
- Loss function $Q(\mathbf{x}, y, f)$ that should be minimized

- Regression problem




Estimated (or predicted) Y value for observation i

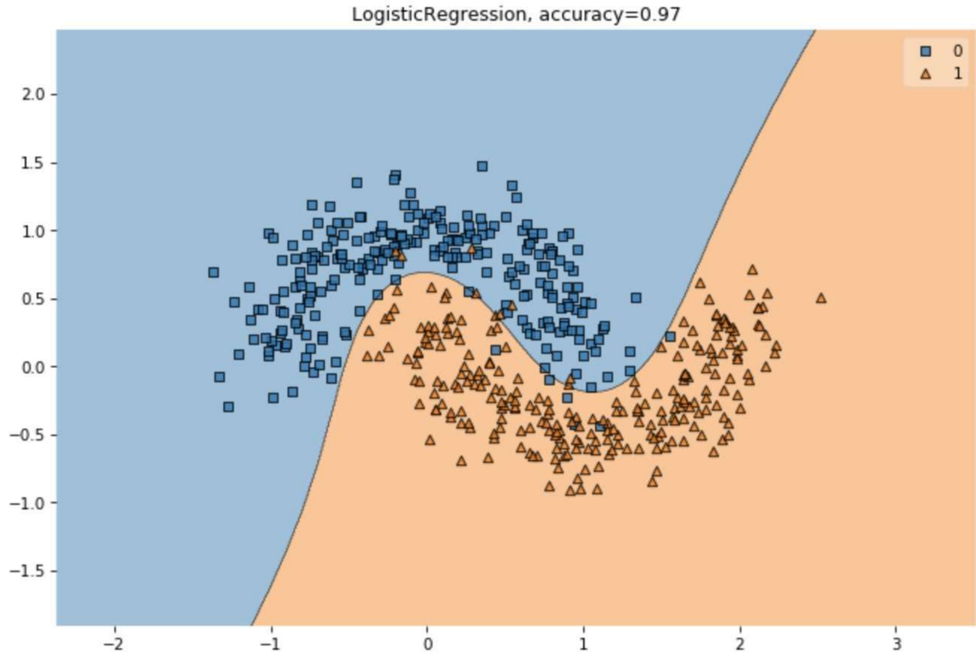
Estimate of the regression intercept

Estimate of the regression slope

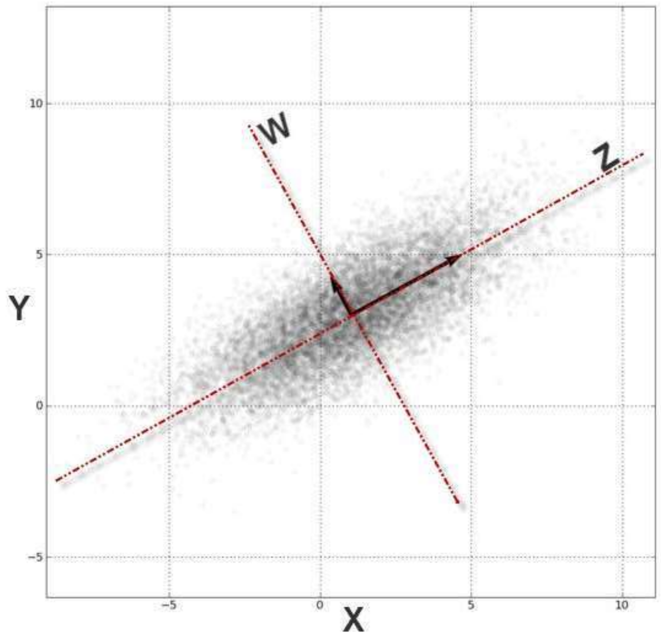
Value of X for observation i

$$\hat{Y}_i = b_0 + b_1 X_i$$

- Regression problem
- Classification problem



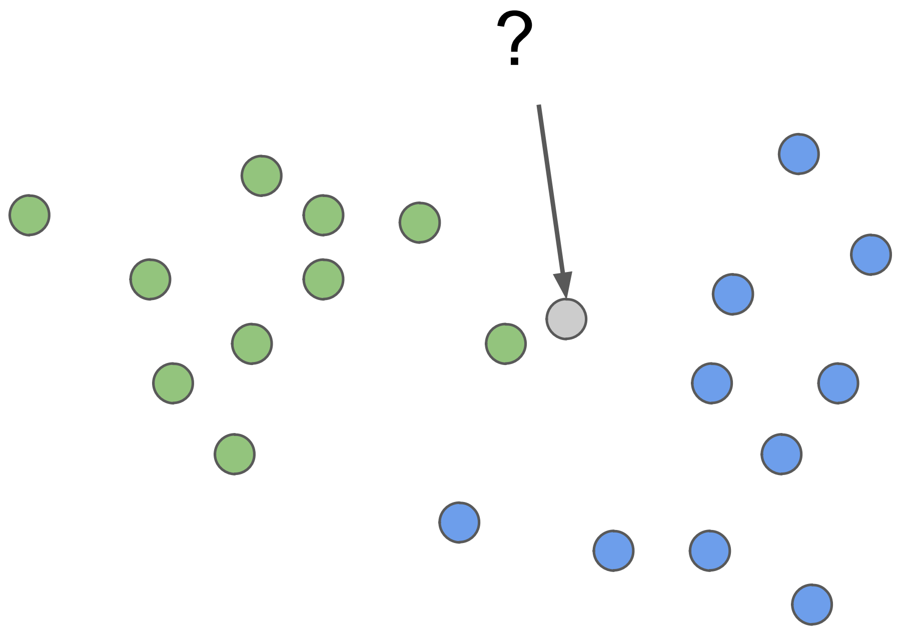
- Regression problem
- Classification problem
- Dimensionality reduction



# kNN – k Nearest Neighbors

# kNN - k Nearest Neighbours

# kNN - k Nearest Neighbours



# k Nearest Neighbors Method

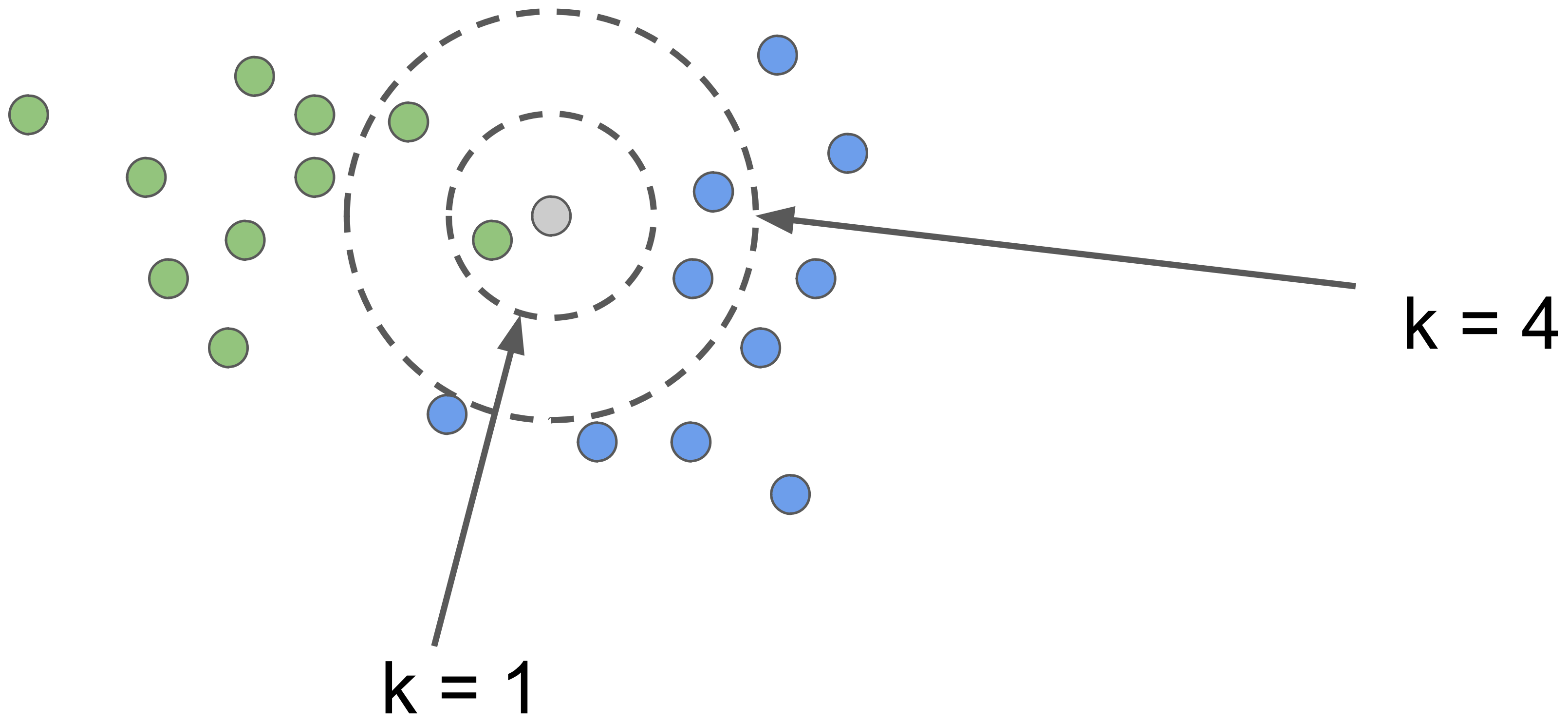
Given a *new observation*:

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

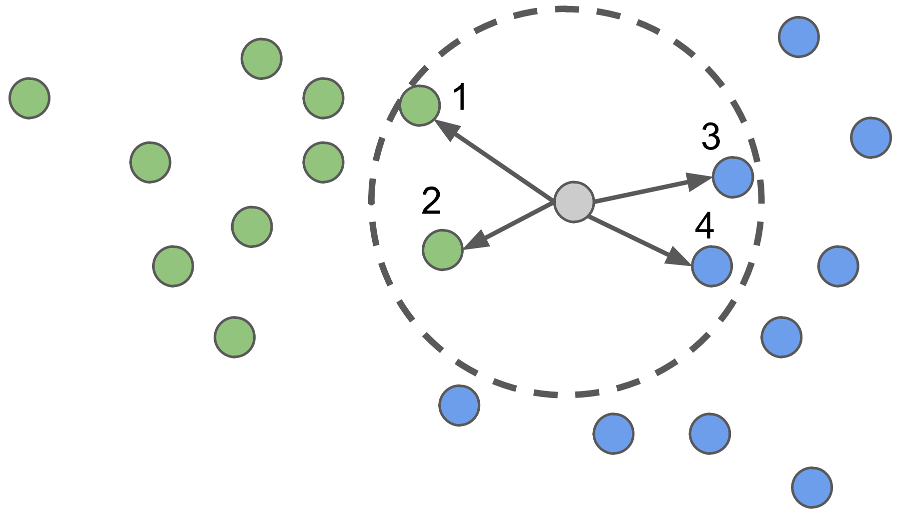


# How to make it better?

- 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

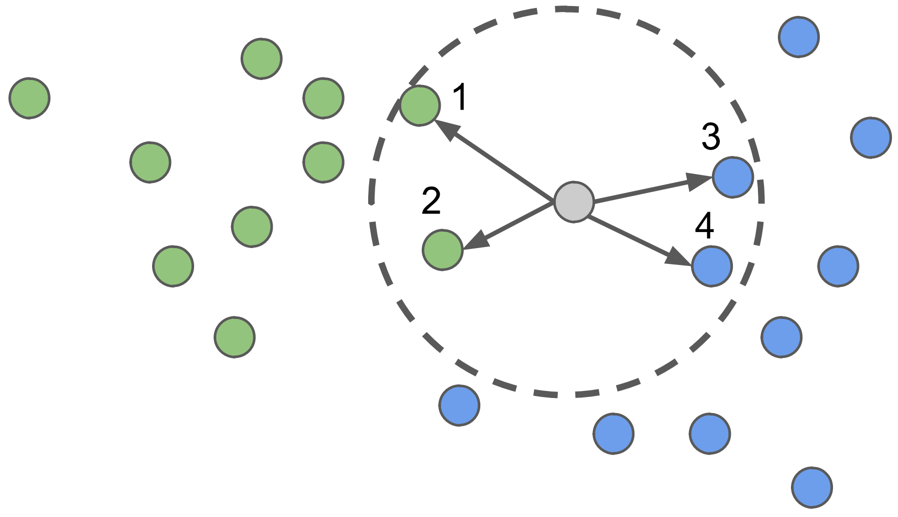
# Weighted kNN

k = 4



# Weighted kNN

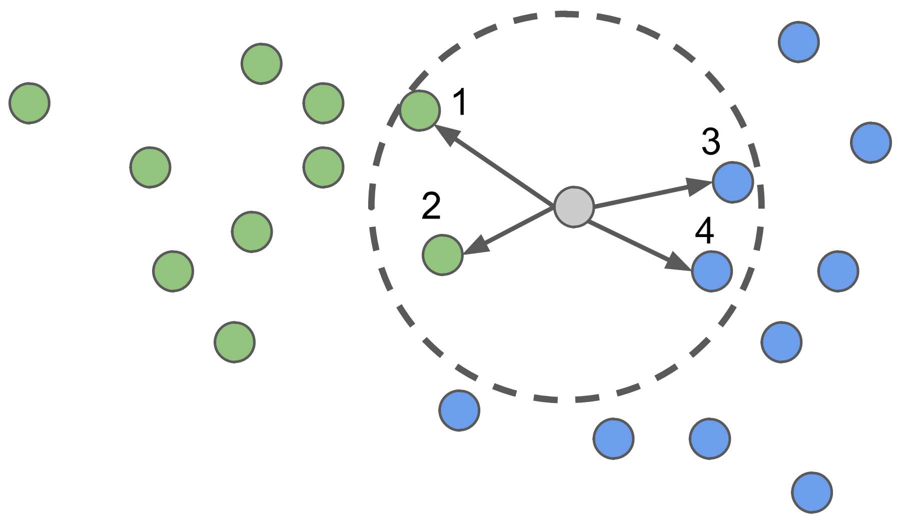
k = 4




- Weights can be adjusted according to the neighbors order, $w(\mathbf{x}_{(i)}) = w_i$

# Weighted kNN

k = 4




- Weights can be adjusted according to the neighbors order,

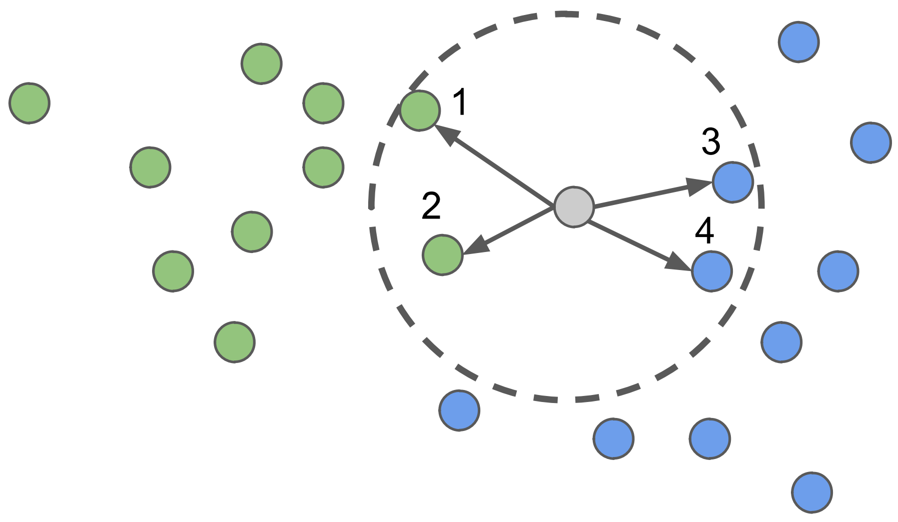
  $$w(\mathbf{x}_{(i)}) = w_i$$

- or on the distance itself

  $$w(\mathbf{x}_{(i)}) = w(d(\mathbf{x}, \mathbf{x}_{(i)}))$$

# Weighted kNN

k = 4



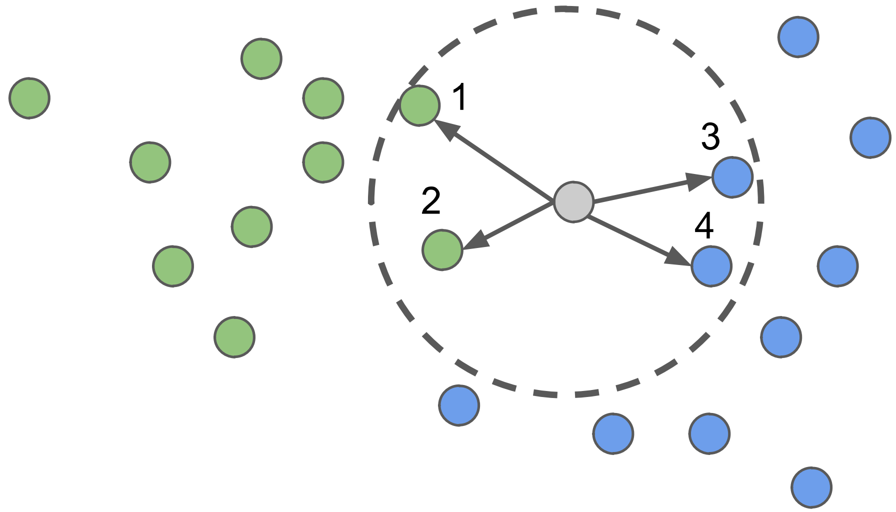

- Weights can be adjusted according to the neighbors order,
  $w(\mathbf{x}_{(i)}) = w_i$

- or on the distance itself
  $w(\mathbf{x}_{(i)}) = w(d(\mathbf{x}, \mathbf{x}_{(i)}))$

$$p_{\text{green}} = \frac{\boxed{w(\mathbf{x}_1) + w(\mathbf{x}_2)}}{w(\mathbf{x}_1) + w(\mathbf{x}_2) + w(\mathbf{x}_3) + w(\mathbf{x}_4)}$$

# Weighted kNN

k = 4




- Weights can be adjusted according to the neighbors order,

$$w(\mathbf{x}_{(i)}) = w_i$$

- or on the distance itself

$$w(\mathbf{x}_{(i)}) = w(d(\mathbf{x}, \mathbf{x}_{(i)}))$$

$$p_{\text{blue}} = \frac{\boxed{w(\mathbf{x}_3) + w(\mathbf{x}_4)}}{w(\mathbf{x}_1) + w(\mathbf{x}_2) + w(\mathbf{x}_3) + w(\mathbf{x}_4)}$$

# Outro

- Remember the **i.i.d.** property
- Usually the first dimension corresponds to the batch size, the second (and so on) to the features/time/…
- Even the naïve assumptions may be suitable in some cases
- Simple models provide great baselines

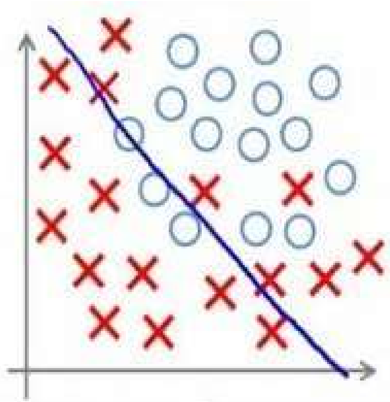
# Model validation and evaluation

# Supervised learning problem statement

Let's denote:

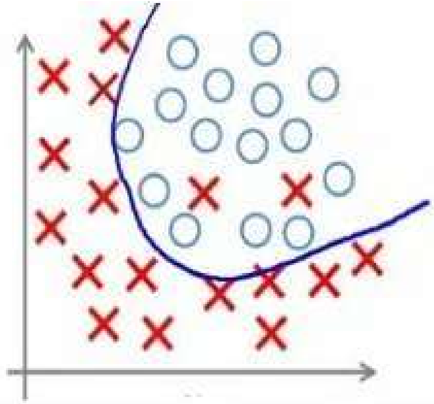
- Training set $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ , where
  - $(\mathbf{x} \in \mathbb{R}^p,\ y \in \mathbb{R}\ )$ for regression
  - $\mathbf{x}_i \in \mathbb{R}^p$ , $y_i \in \{+1, -1\}$ for binary classification
- Model $f(\mathbf{x})$ predicts some value for every object
- Loss function $Q(\mathbf{x}, y, f)$ that should be minimized

# Overfitting vs. underfitting

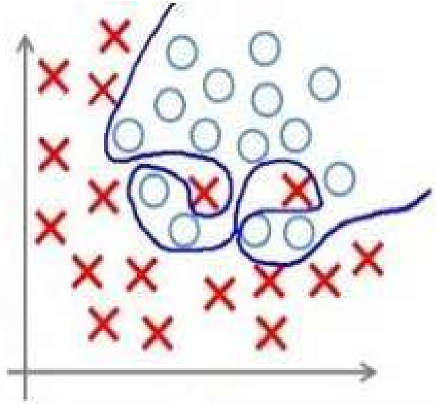


Under-fitting

(too simple to explain the variance)



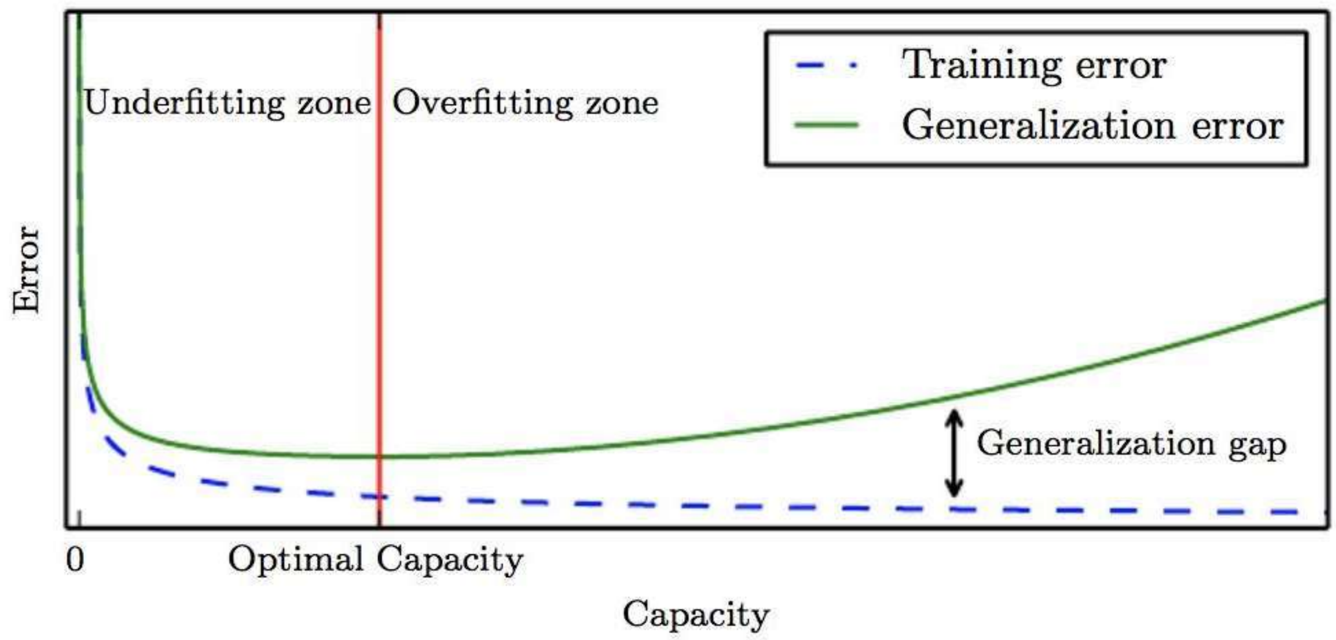
Appropriate-fitting



Over-fitting

(forcefitting -- too good to be true)

# Overfitting vs. underfitting



# Overfitting vs. underfitting

- We can control overfitting / underfitting by altering model’s capacity (ability to fit a wide variety of functions):
- select appropriate hypothesis space
- learning algorithm’s effective capacity may be less than the representational capacity of the model family

# Evaluating the quality


Image credit: Joseph Nelson @josephofiowa

# Evaluating the quality


Is it good enough?

Image credit: Joseph Nelson @josephofiowa

# Evaluating the quality

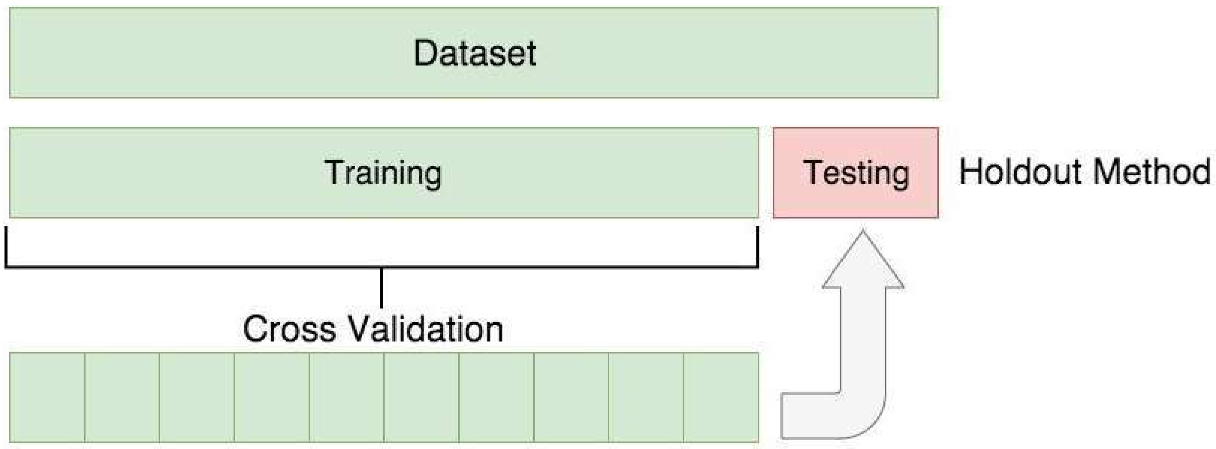



Image credit: Joseph Nelson @josephofiowa

# Evaluating the quality

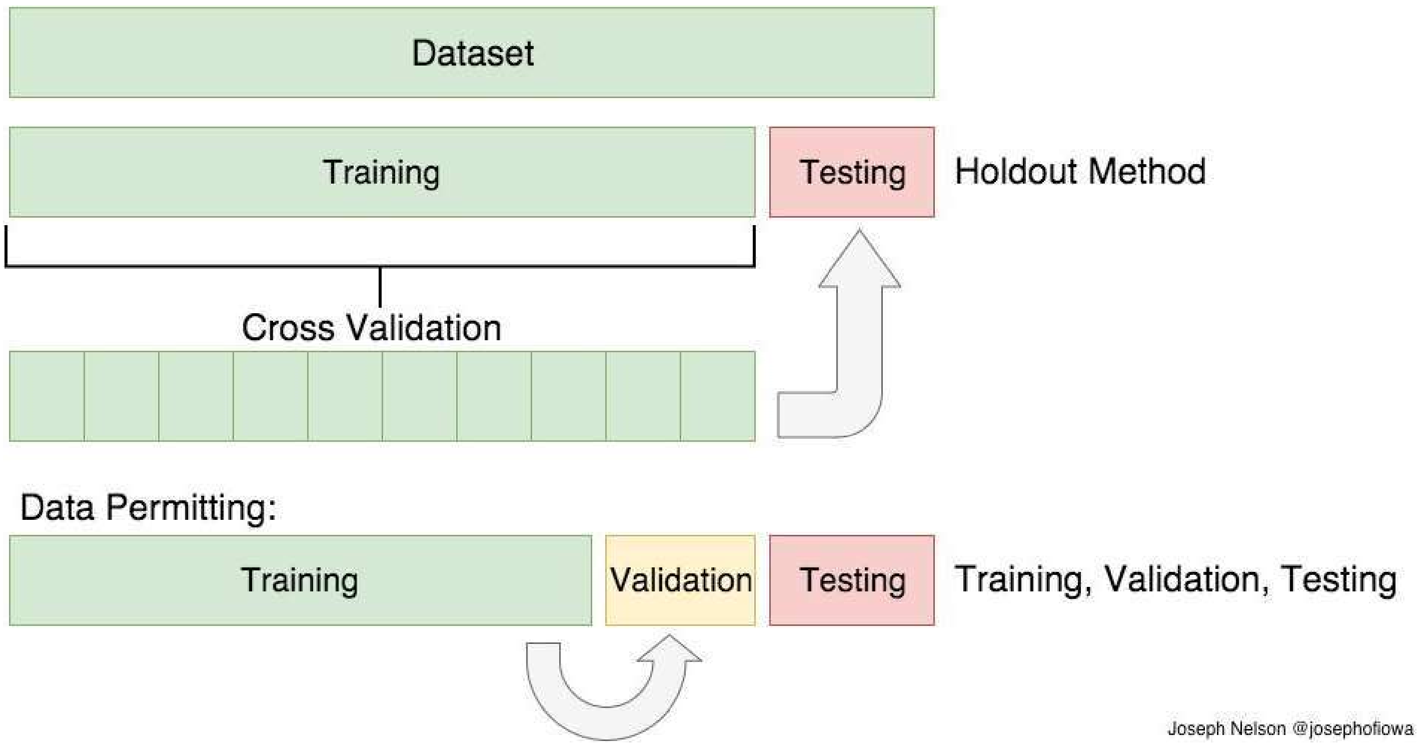



Image credit: Joseph Nelson @josephofiowa

# Cross-validation