



# Machine Learning

## Lecture 1: intro to ML

**Radoslav Neychev**

Spring 2021

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)

# I am Makhmoodjon Sodikov

- Senior ML Research Engineer, Team Lead
- 3+ years in Academic Research and Lecturing
- 4+ years in AI Industry
- 20+ RU/EU hackathons
- 100+ chords on e-guitar 🎸👉!



# I am Makhmoodjon Sodikov

- Senior ML Research Engineer, Team Lead
- 3+ years in Academic Research and Lecturing
- 4+ years in AI Industry
- 20+ RU/EU hackathons
- 100+ chords on e-guitar 🎸👉!

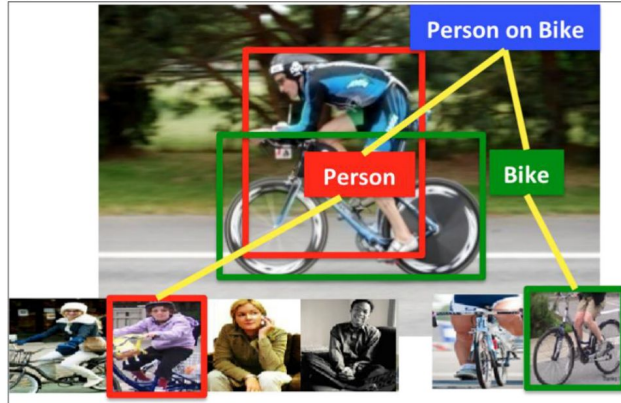


Motivation, historical overview and  
current state of ML and AI

# Machine Learning applications



- Object detection
- Action classification
- Image captioning
- ...

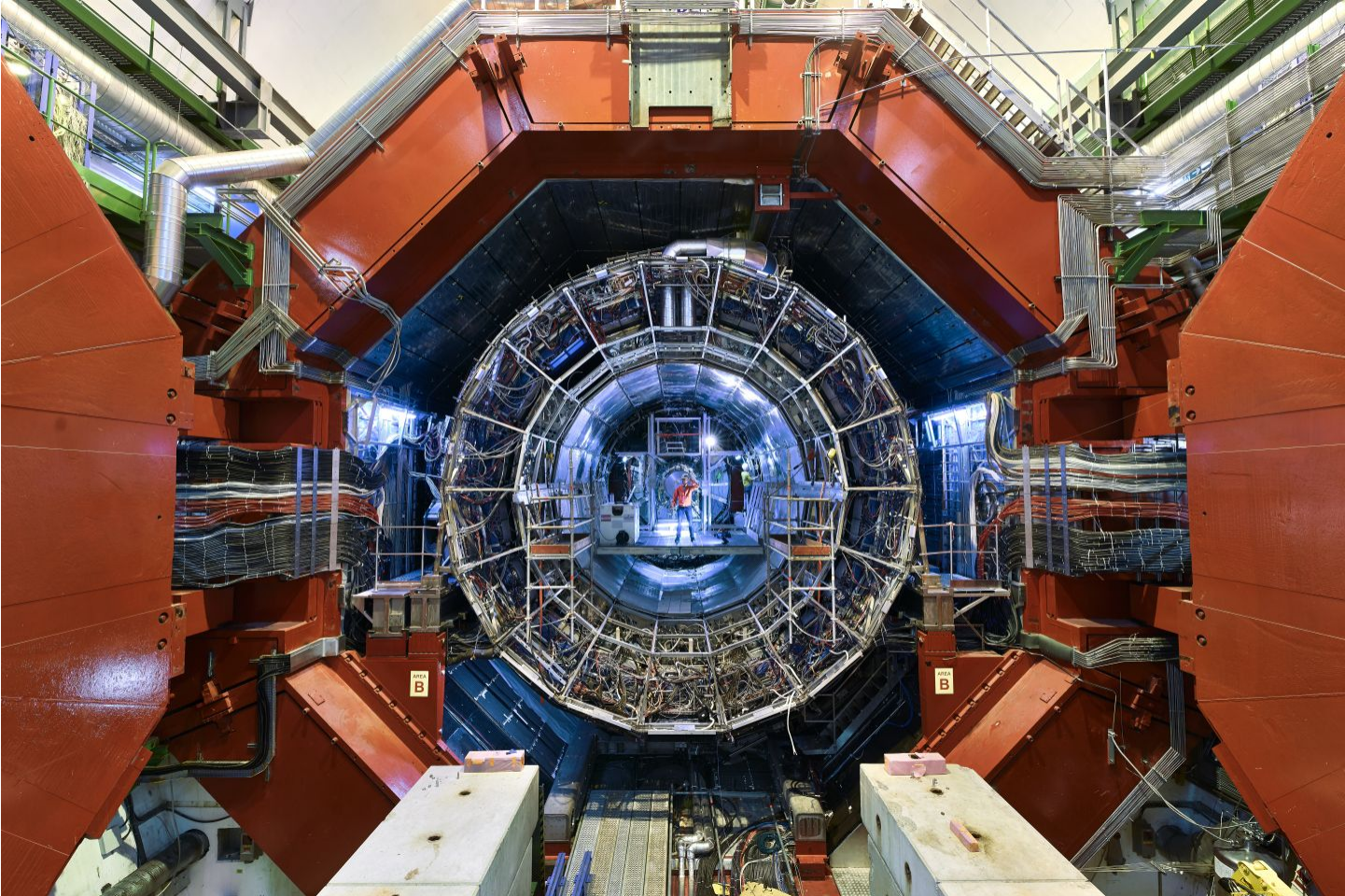


# Machine Learning applications





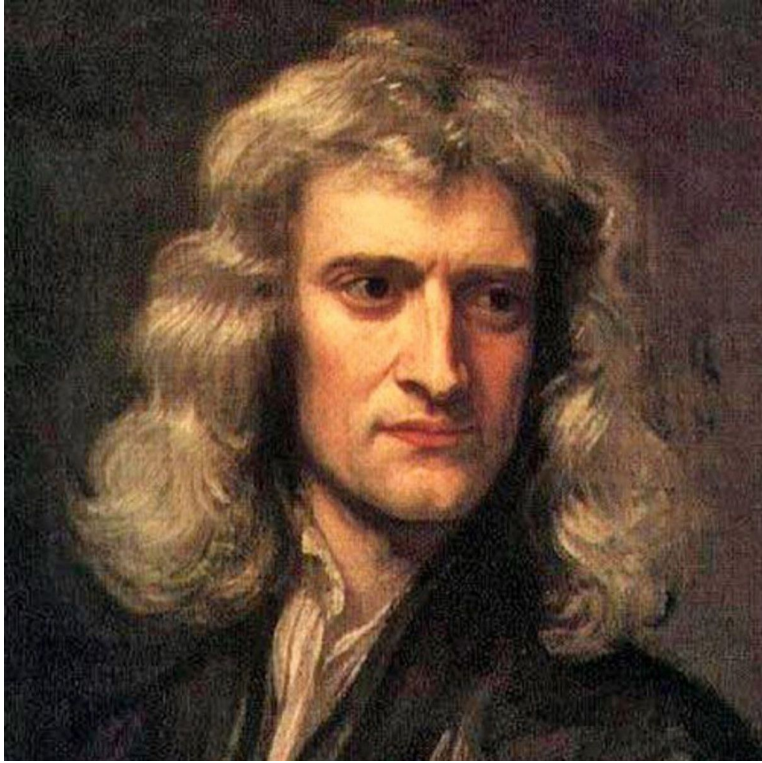
# Machine Learning applications





Data → Knowledge

Long before the ML

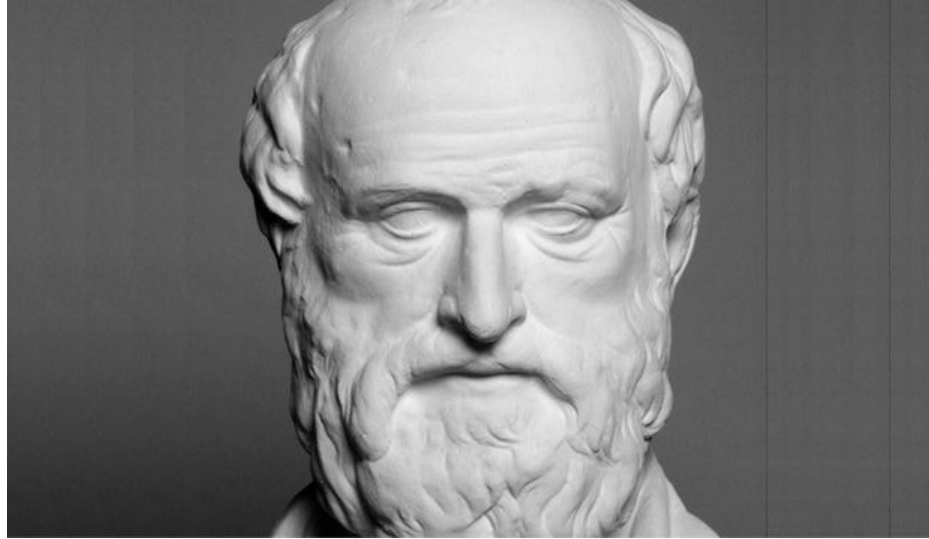


Isaac Newton



Johannes Kepler

Long before the ML



Eratosthenes

ML thesaurus

# 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

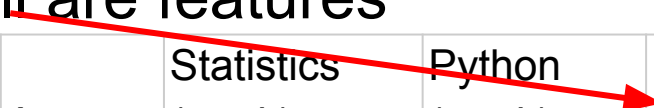
These all are features



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

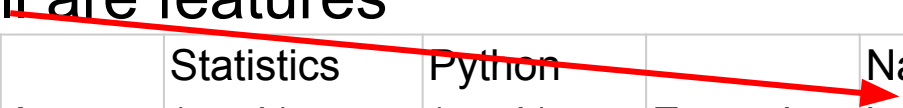


Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE



# 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

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)*

# 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.*

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:

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

# 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:*

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

# 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	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:*

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



# 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.*

***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  $\mathbf{y}$  is vector of targets):

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

***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  $\mathbf{y}$  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.

## Recap:

- Dataset
- Observation (datum)
- Feature
- Design matrix
- Target
- Prediction
- Model
- Loss function
- Parameter
- Hyperparameter



# Maximum Likelihood Estimation

Denote dataset generated by distribution with parameter  $\theta$

***Likelihood*** function:

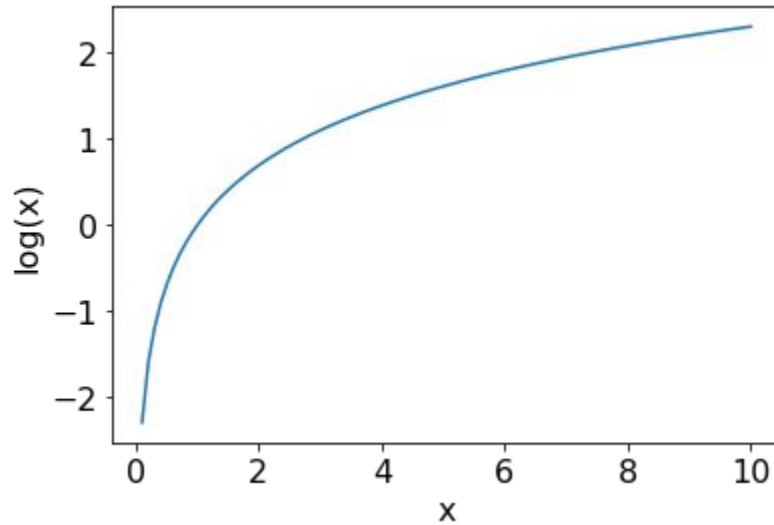
$$L(\theta|X, Y) = P(X, Y|\theta)$$

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

**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



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**

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

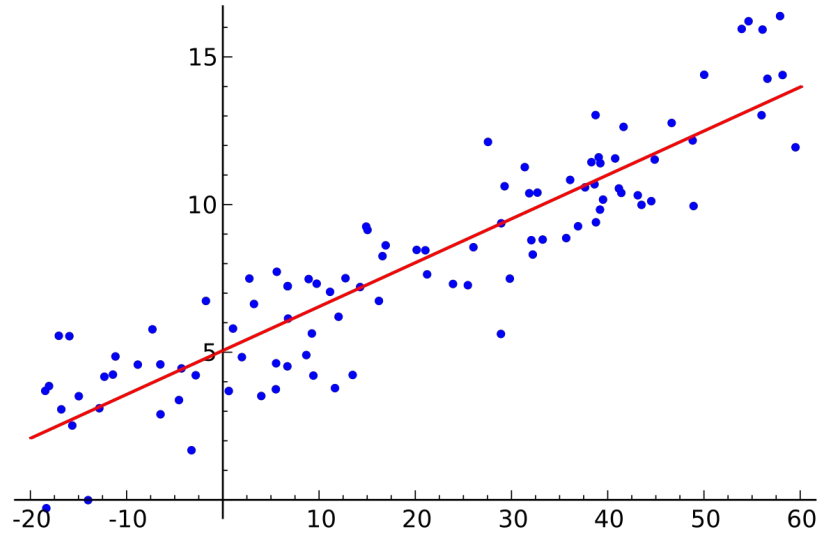
# Machine Learning problems overview

# 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  $i$

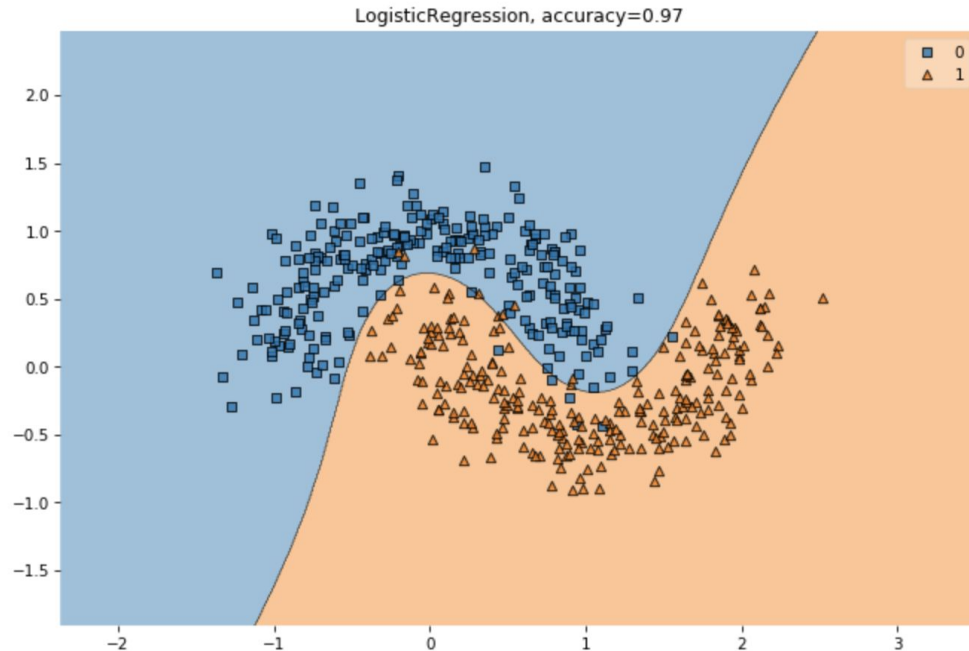
Estimate of  
the regression  
intercept

Estimate of the  
regression slope

Value of X for  
observation  $i$

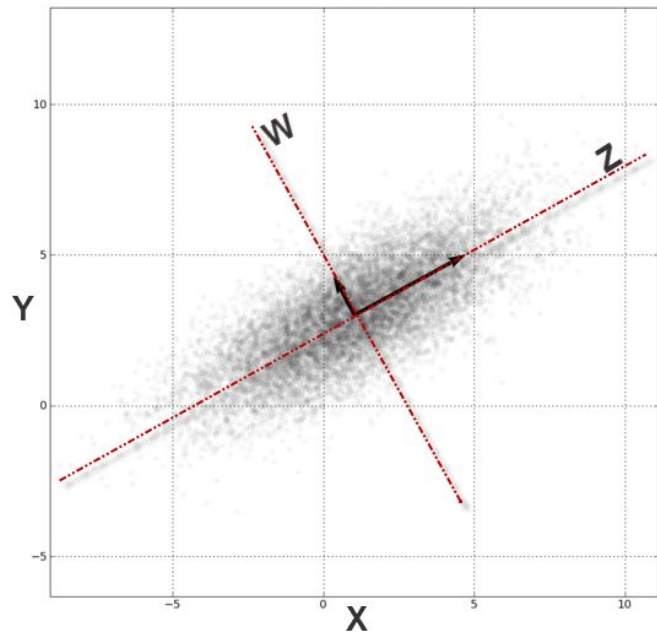
$$\hat{Y}_i = b_0 + b_1 X_i$$

- Regression problem
- Classification problem





- Regression problem
- Classification problem
- Dimensionality reduction



# Naïve Bayes classifier

# Naïve Bayes classifier

Let's denote:

- Training set  $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^n$  , where
  - $\mathbf{x}_i \in \mathbb{R}^p$  ,  $y_i \in \{C_1, \dots, C_k\}$  for k-class classification

# Bayes' theorem

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

or, in our case

$$P(y_i = C_k | \mathbf{x}_i) = \frac{P(\mathbf{x}_i | y_i = C_k)P(y_i = C_k)}{P(\mathbf{x}_i)}$$

# Naïve Bayes classifier

Let's denote:

- Training set  $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^n$  , where
  - $\mathbf{x}_i \in \mathbb{R}^p$  ,  $y_i \in \{C_1, \dots, C_K\}$  for K-class classification

$$P(y_i = C_k | \mathbf{x}_i) = \frac{P(\mathbf{x}_i | y_i = C_k) P(y_i = C_k)}{P(\mathbf{x}_i)}$$

Naïve assumption: features are **independent**

# Naïve Bayes classifier

$$P(y_i = C_k | \mathbf{x}_i) = \frac{P(\mathbf{x}_i | y_i = C_k) P(y_i = C_k)}{P(\mathbf{x}_i)}$$

Naïve assumption: features are **independent**:

$$P(\mathbf{x}_i | y_i = C_k) = \prod_{l=1}^p P(x_i^l | y_i = C_k)$$

# Naïve Bayes classifier

$$P(y_i = C_k | \mathbf{x}_i) = \frac{P(\mathbf{x}_i | y_i = C_k) P(y_i = C_k)}{\cancel{P(\mathbf{x}_i)}}$$

Optimal class label:

$$C^* = \arg \max_k P(y_i = C_k | \mathbf{x}_i)$$

To find maximum we even do not need the denominator

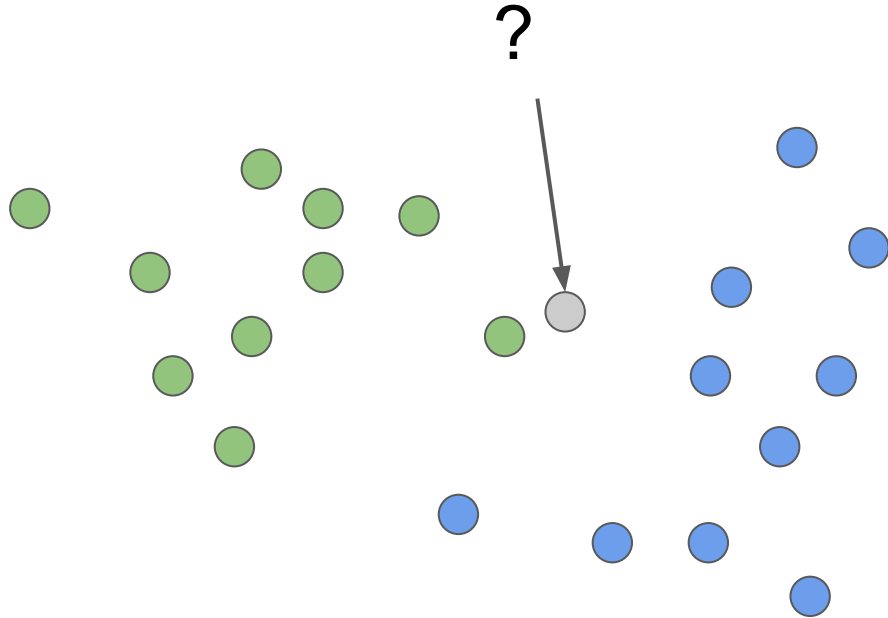
But we need it to get probabilities

kNN – k Nearest Neighbors



# kNN - k Nearest Neighbours

# kNN - k Nearest Neighbours



# k Nearest Neighbors Method

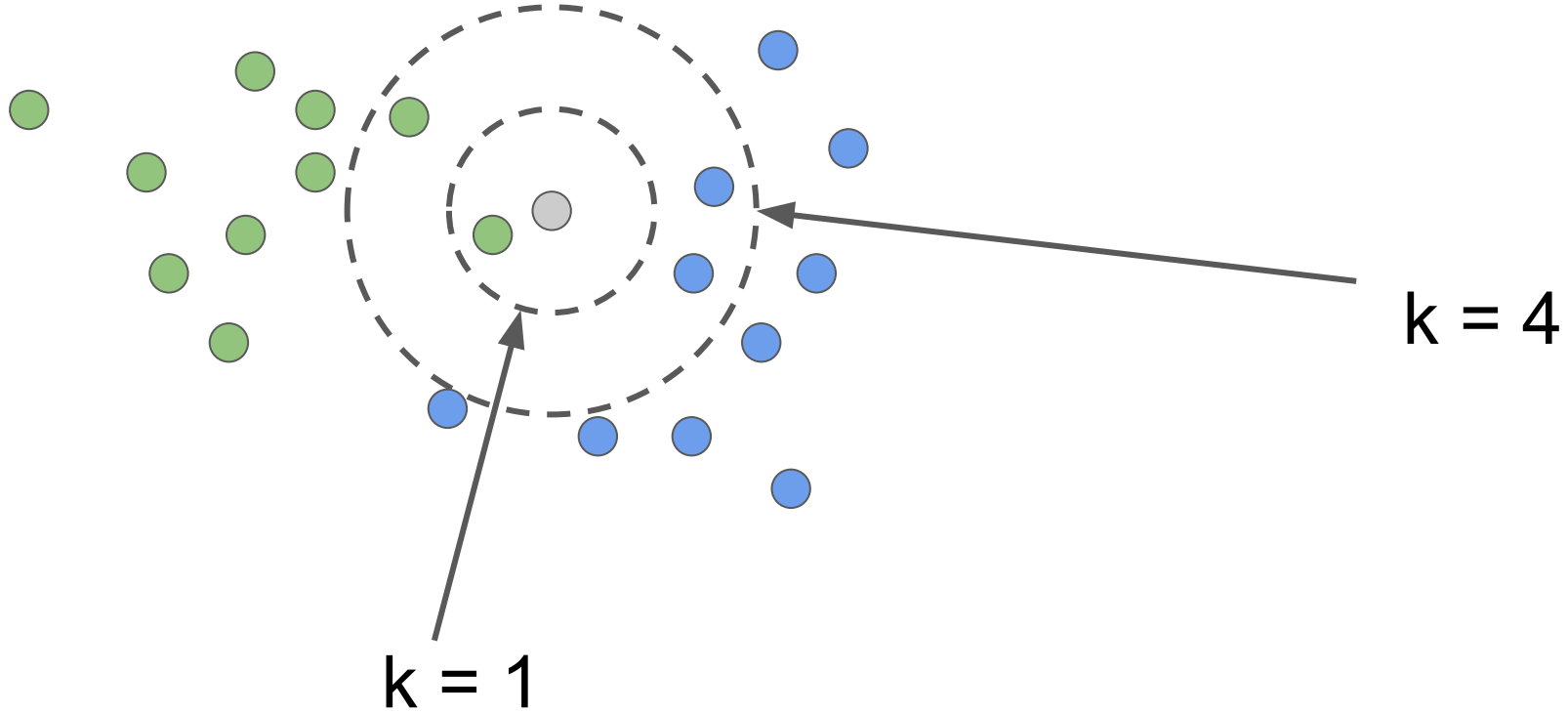
Given a *new observation*:

1. Calculate the distance to each of the samples in the dataset.
2. Select  $k$  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

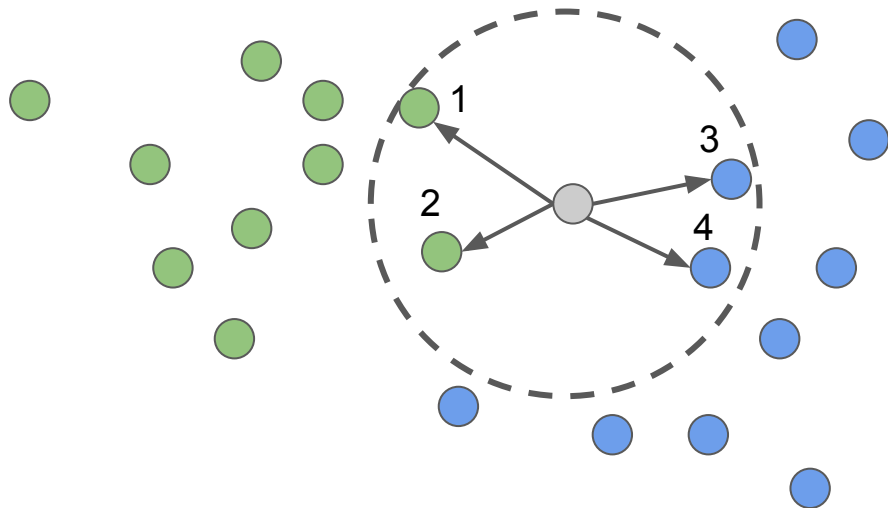


## How to make it better?

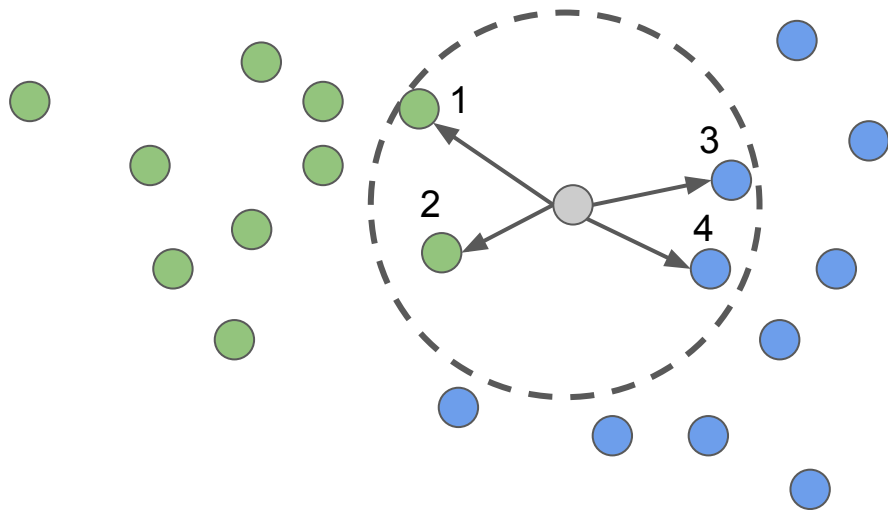
- The number of neighbors  $k$  (it is a ***hyperparameter***)
- The distance measure between samples
  - a. Hamming
  - b. Euclidean
  - c. cosine
  - d. Minkowski distances
  - e. etc.
- Weighted neighbours

$k = 4$

Weighted kNN



$k = 4$

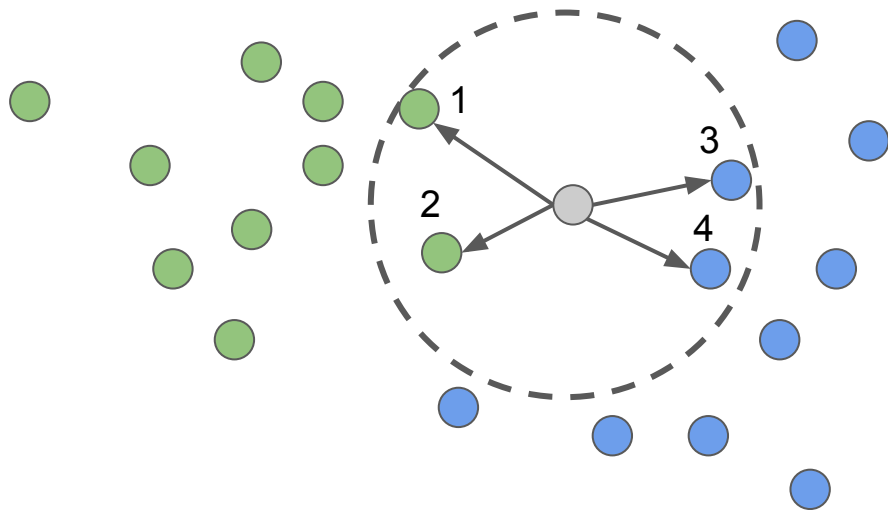


## Weighted kNN

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



$k = 4$



## Weighted kNN

- Weights can be adjusted according to the neighbors order,

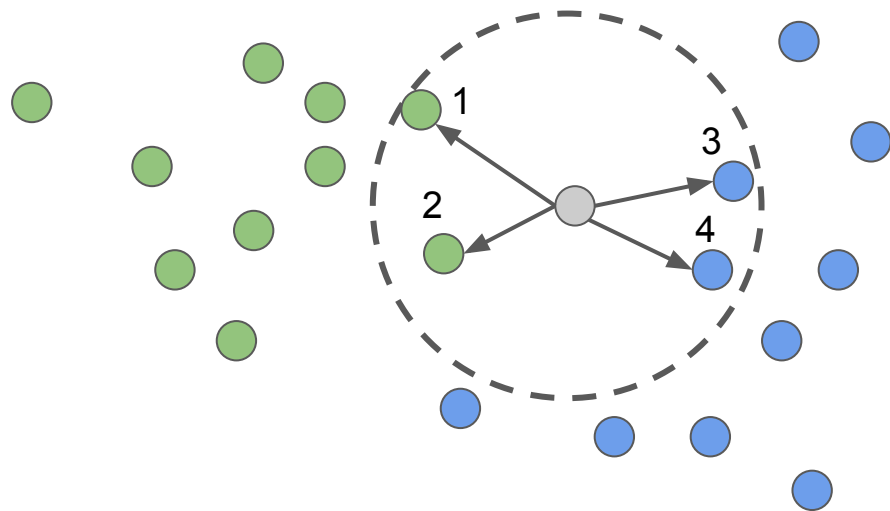
$$w(\mathbf{x}_{(i)}) = w_i$$

- or on the distance itself

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

$k = 4$

## Weighted kNN



- Weights can be adjusted according to the neighbors order,

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

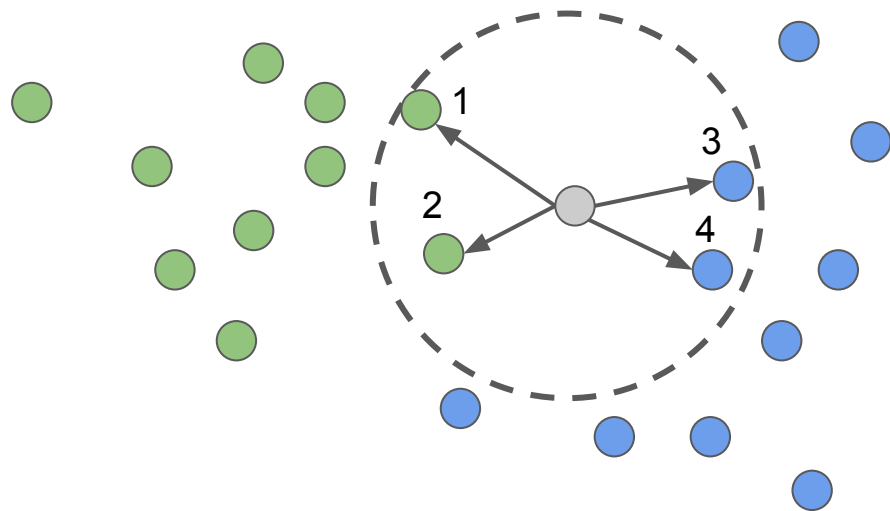
- or on the distance itself

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

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

$k = 4$

## Weighted kNN



- Weights can be adjusted according to the neighbors order,

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

- or on the distance itself

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

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

- 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

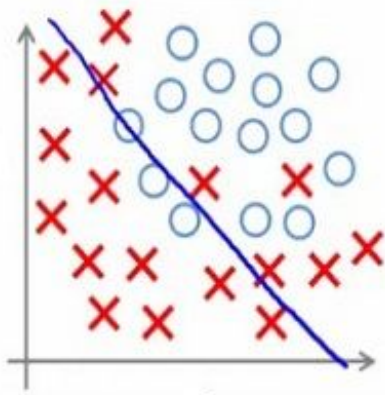
# Model validation and evaluation

# Supervised learning problem statement

Let's denote:

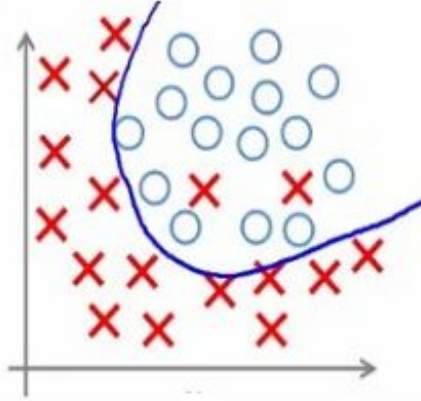
- Training set  $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^n$  , where
  - $(\mathbf{x} \in \mathbb{R}^p, y \in \mathbb{R})$  for regression
  - $\mathbf{x}_i \in \mathbb{R}^p$  ,  $y_i \in \{+1, -1\}$  for binary classification
- Model  $f(\mathbf{x})$  predicts some value for every object
- Loss function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

# Overfitting vs. underfitting

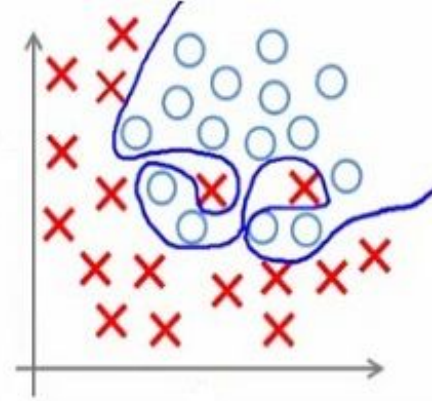


**Under-fitting**

(too simple to  
explain the  
variance)



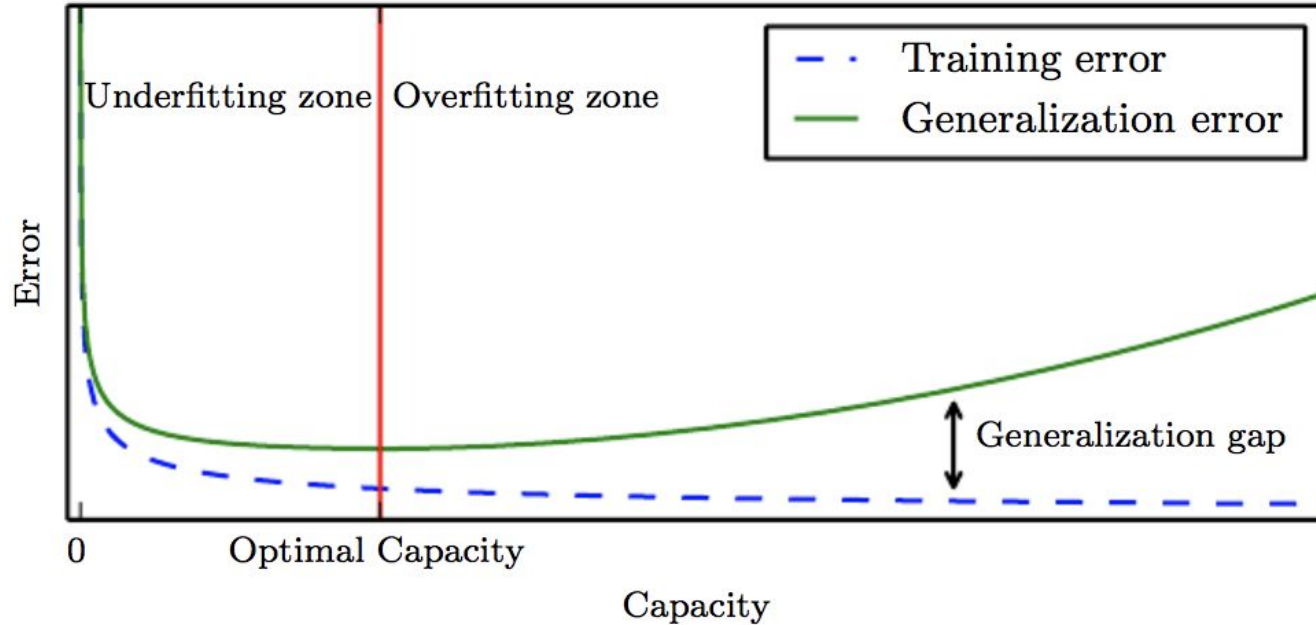
**Appropriate-fitting**



**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

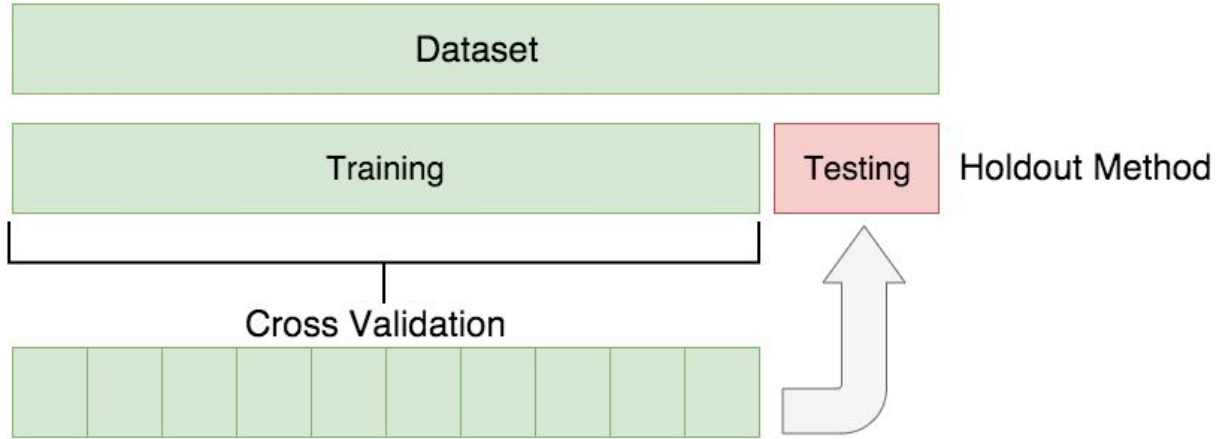


# Evaluating the quality

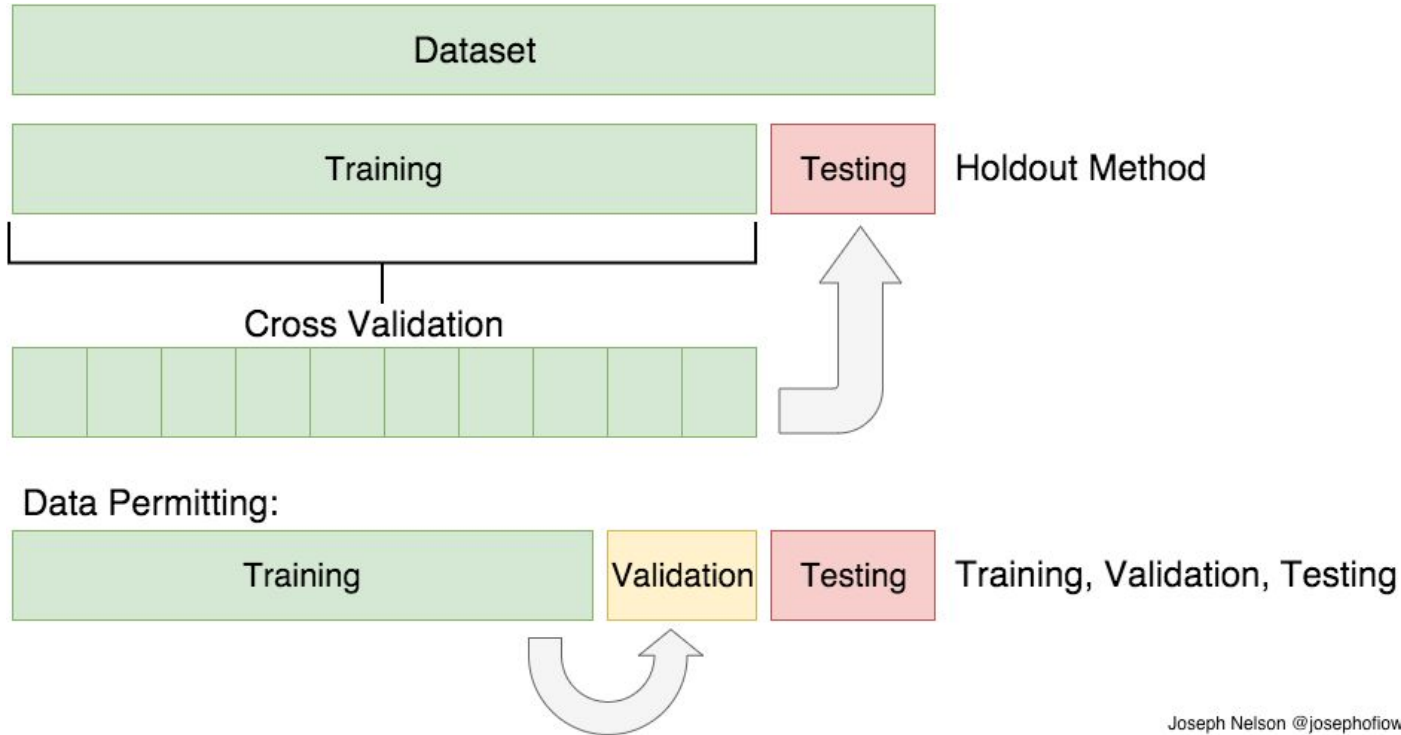


Is it good enough?

# Evaluating the quality



# Evaluating the quality



Joseph Nelson @josephofiowa

# Cross-validation

