

# Intro to ML Naïve Bayes

Radoslav Neychev



fall 2024



# Outline

- 
1. Prerequisites
  2. Course structure
  3. ML and AI overview
  4. Thesaurus and notation
  5. Datasets
  6. Maximum Likelihood Estimation
  7. Some Machine Learning problems
    - a. Classification
    - b. Regression
    - c. Dimensionality reduction
  8. Naïve Bayes classifier

# Prerequisites

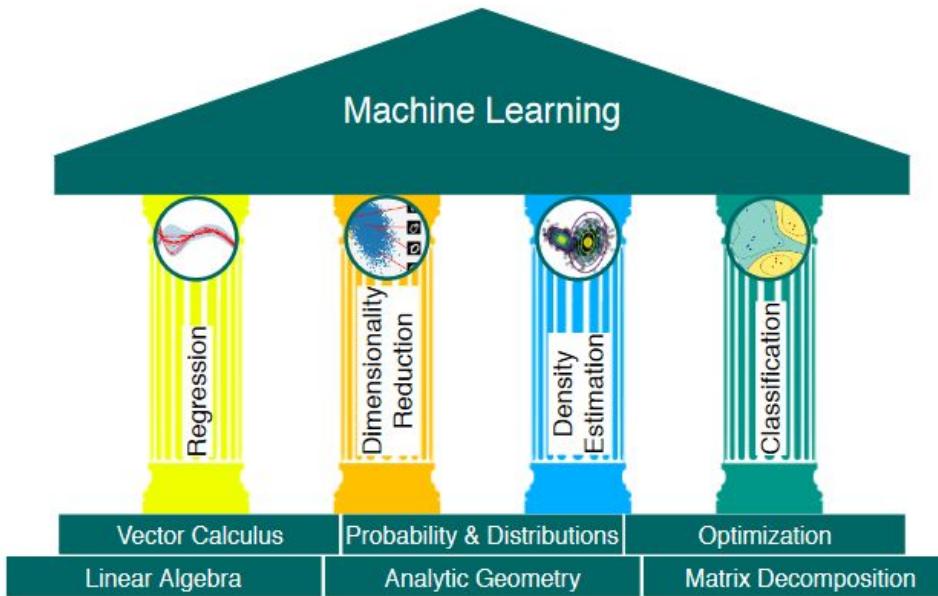
---

girafe  
ai

01



# Math requirements



Materials to refresh necessary math:

- <https://github.com/qirafe-ai/math-basics-for-ai/>
- <https://mml-book.github.io/book/mml-book.pdf>

# Math requirements

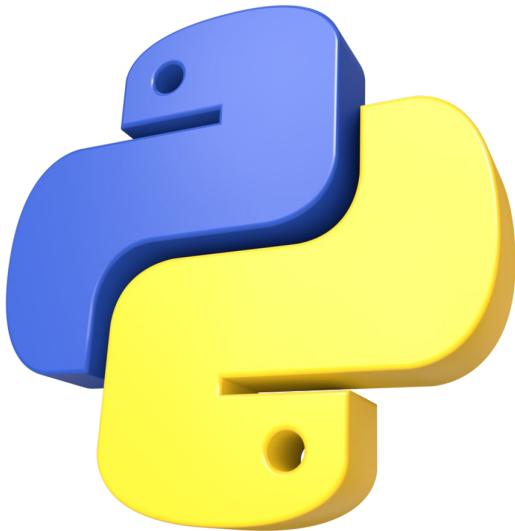


Basic topics we expect you to know:

- Linear algebra
  - Matrices
  - Linear functions, transforms
  - Matrix decompositions
- Calculus
  - Matrix differentiation
- Statistics
  - Distributions
  -
- Basics of optimization
  - Extremum
  - Convexity



# Programming requirements



Tools:

- Python
- Numpy, Scipy
- Jupyter notebooks
- VS Code
- UNIX terminal
- git

Good places to study Python:

<https://snakify.org/en/> or <https://pythontutor.ru/>

# Course structure

---

girafe  
ai

02



# Course structure (planned)

1. Intro
2. Linear models
3. Trees and ensembles
4. Neural networks
  - a. CNN
  - b. RNN
  - c. Attention
5. Working with data types
  - a. Images
  - b. Text
  - c. Timeseries
6. Other topics
  - a. Geometrical ML
  - b. Auto ML

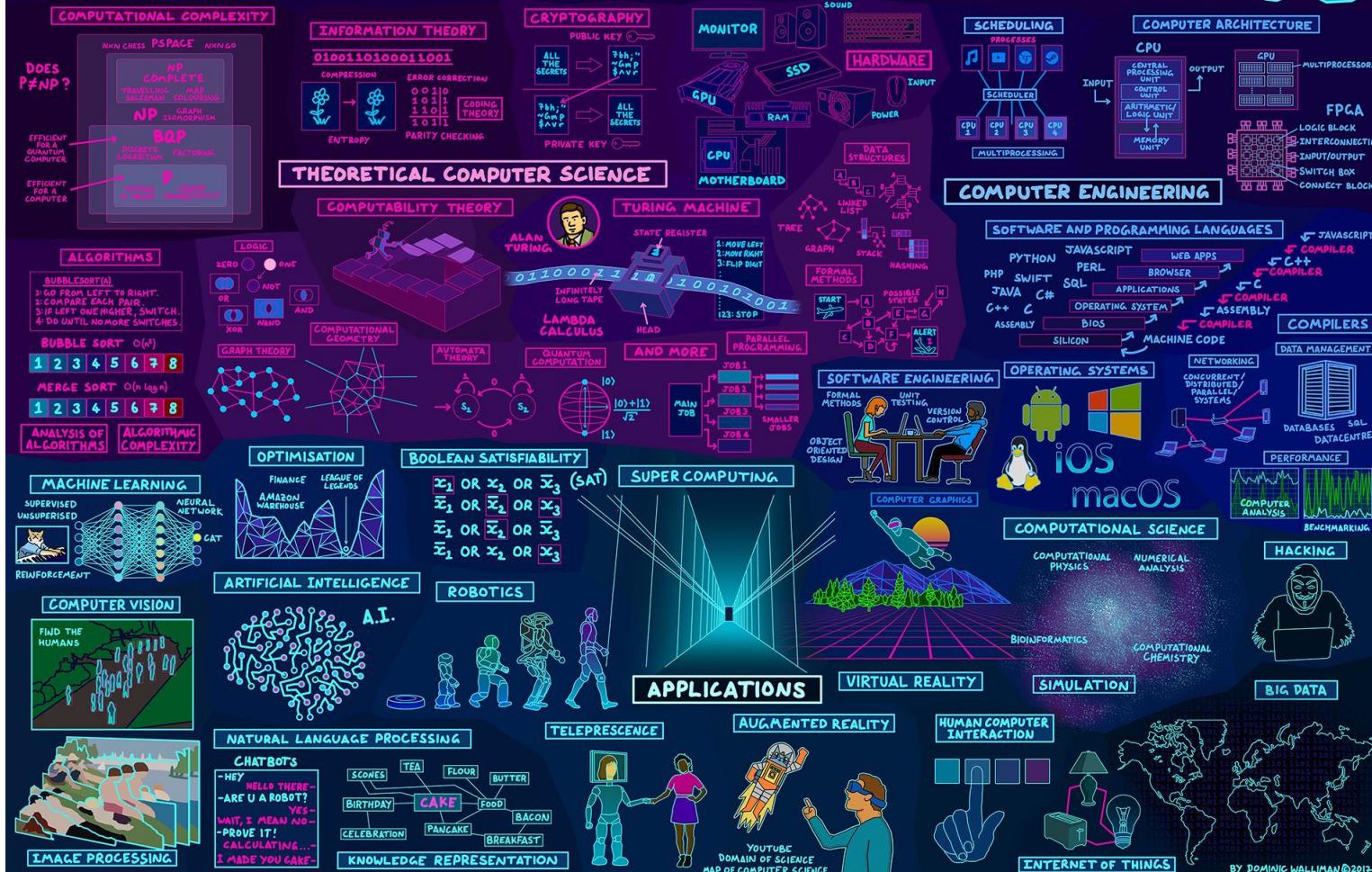
# ML and AI overview

---

girafe  
ai

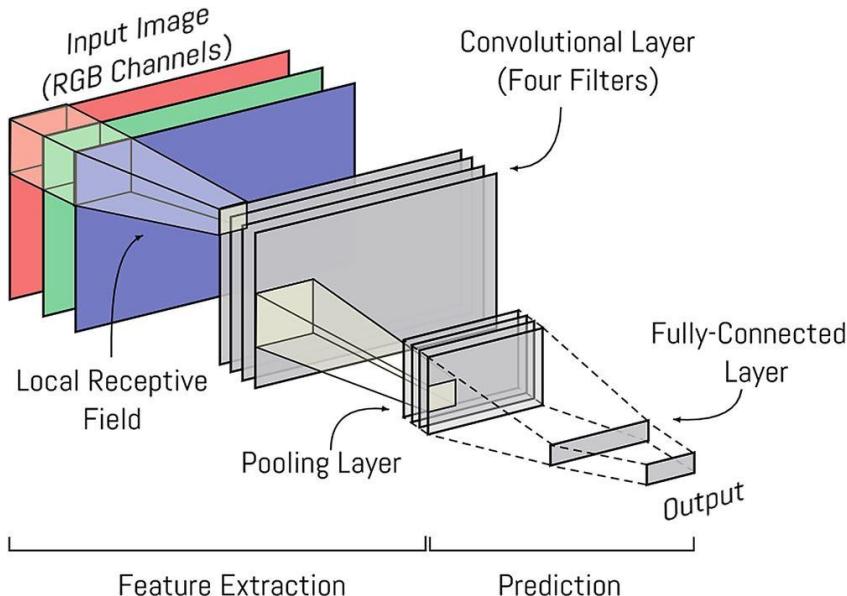
03

# MAP OF COMPUTER SCIENCE





# Computer Vision



Basics:

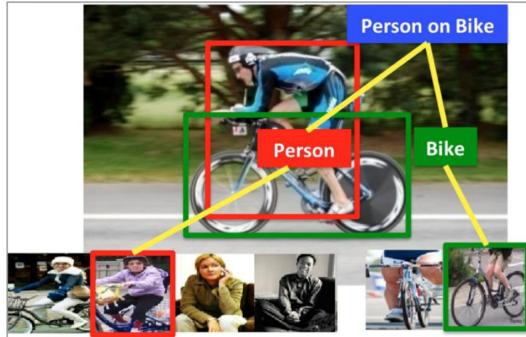
- Classical CV (filters, border detectors)
- Convolutional Neural Networks



# Computer Vision

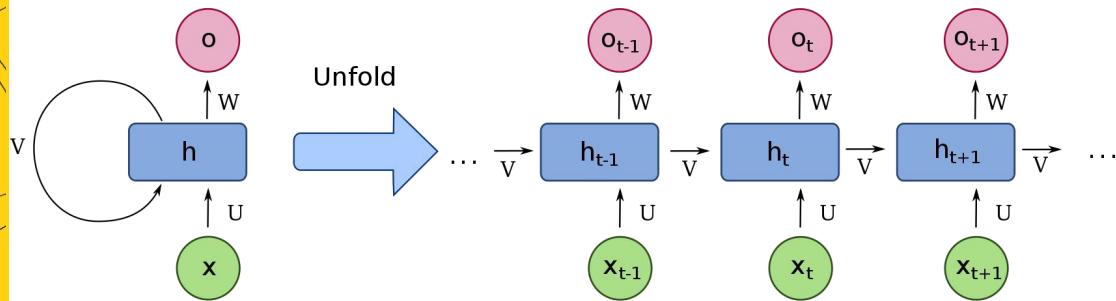
Some achievements:

- Object detection
- Semantic segmentation
- Generative models





# Natural Language Processing



Basics:

- Language models
- Recurrent Neural Networks
- Attention module



# Natural Language Processing

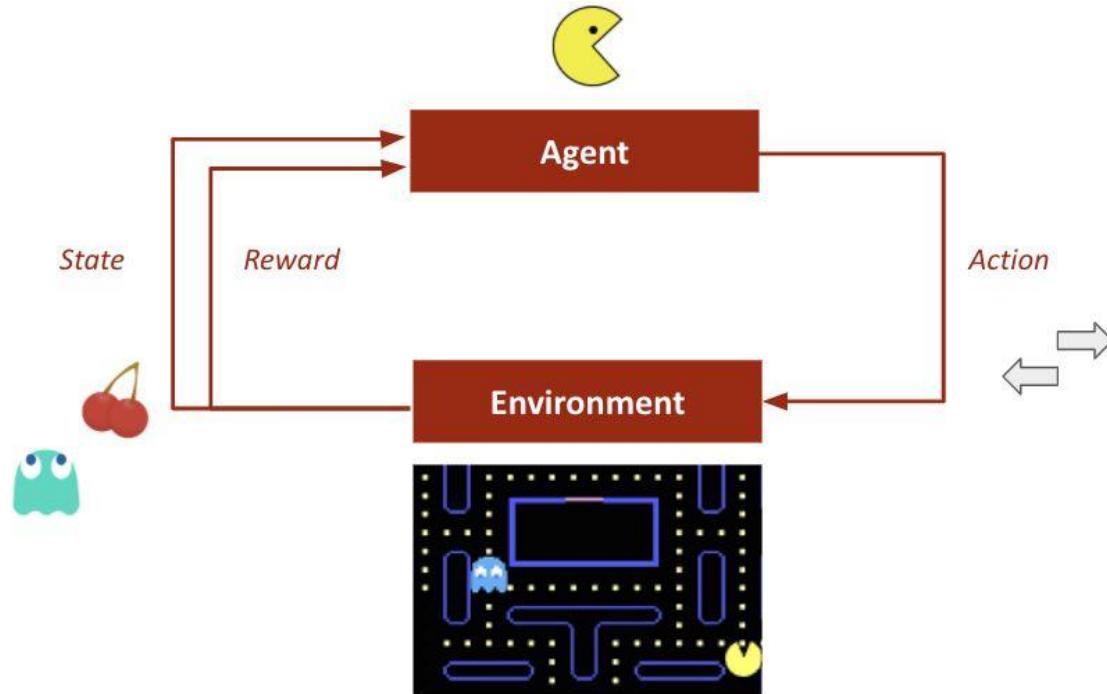
Some achievements:

- Machine translation
- Texts classification
- Texts generation





# Reinforcement Learning



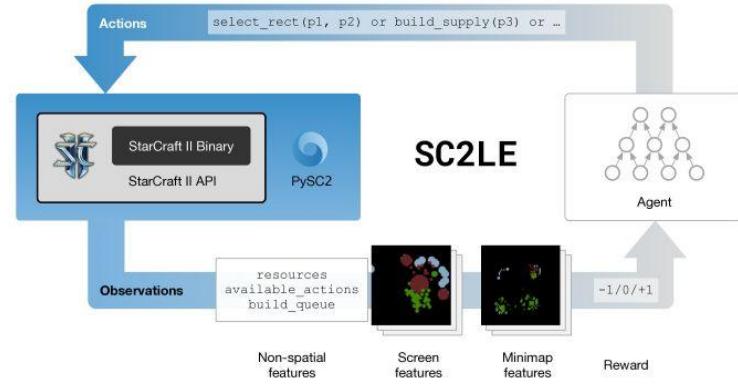
Basics:

- Q-learning
- DQN
- REINFORCE

# Reinforcement Learning

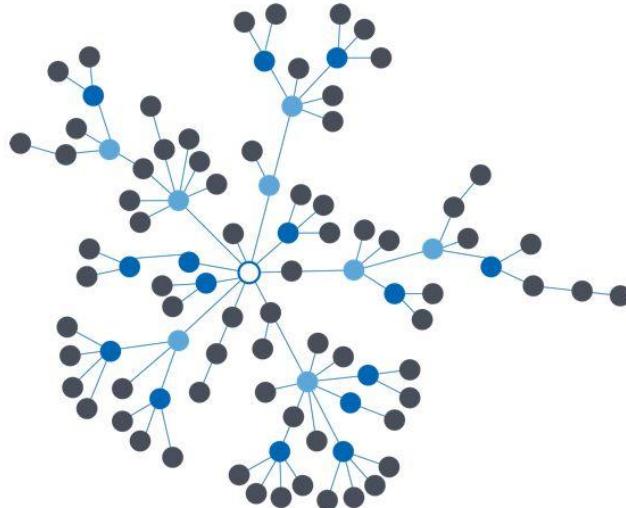
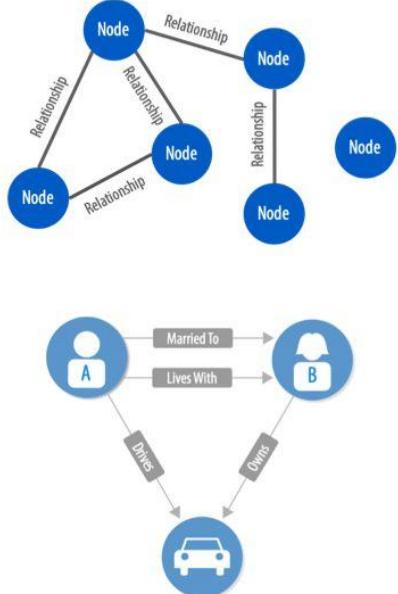
Achievements:

- Alpha Go
- OpenAI Five
- DeepMind Star Craft 2





# Machine Learning on Graphs



Basics:

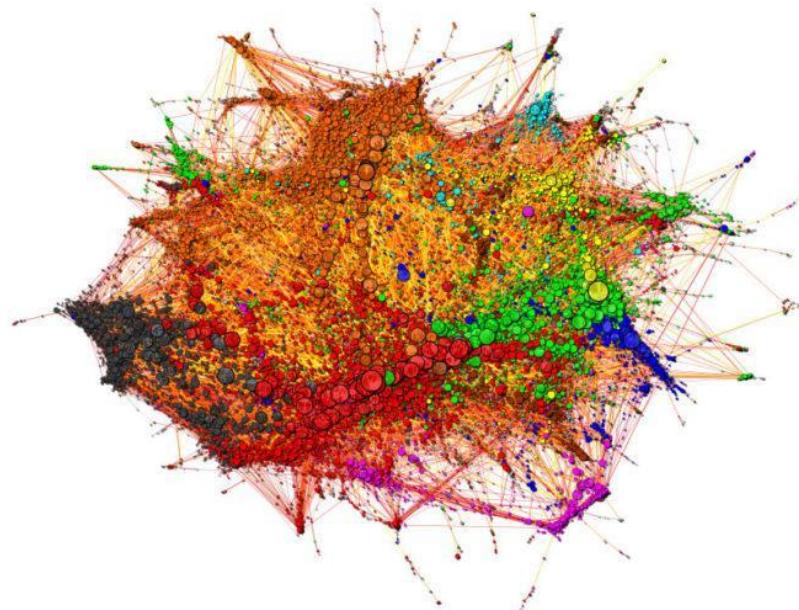
- Random graphs
- Small world model
- Graphs convolutions



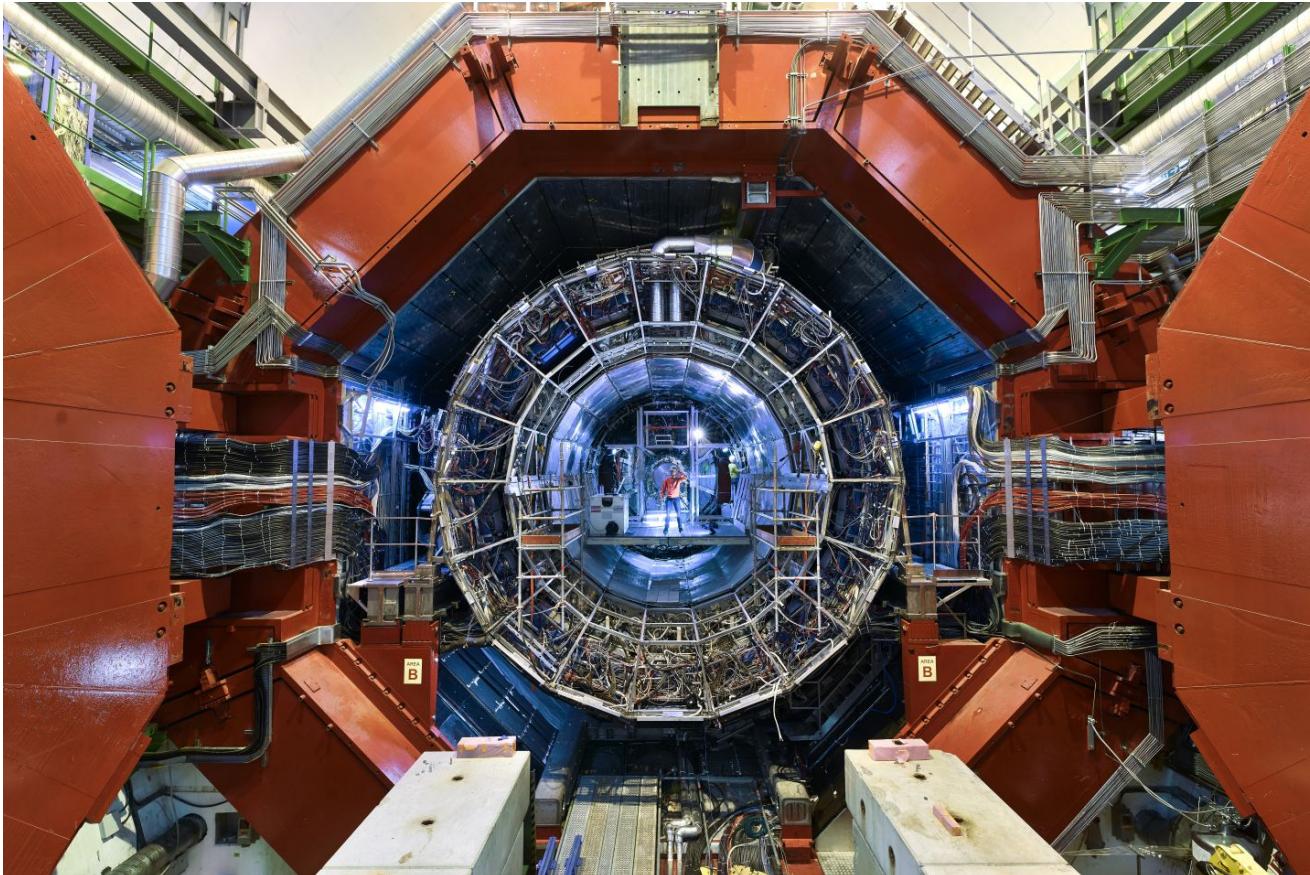
# Machine Learning on Graphs

Some achievements:

- Communities detection
- Recommender systems



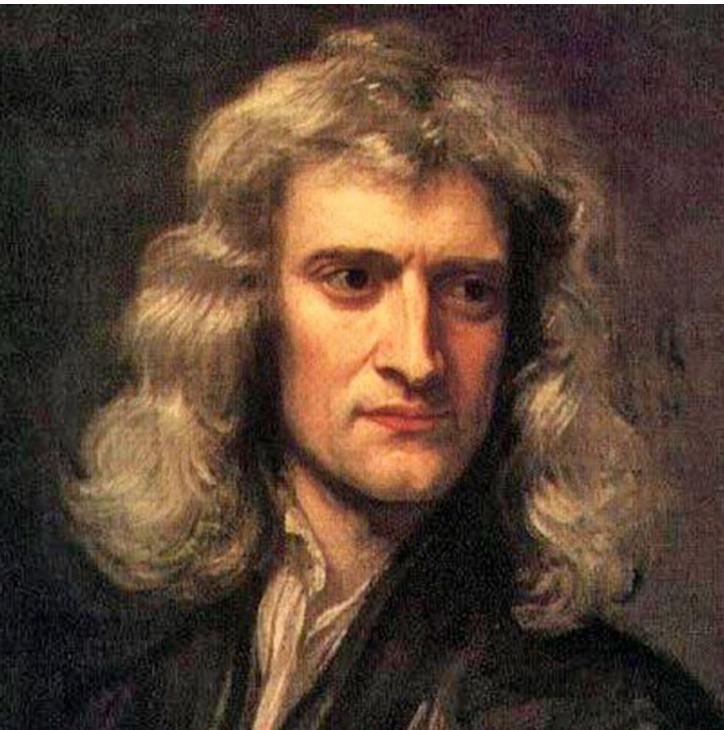
# Machine Learning applications



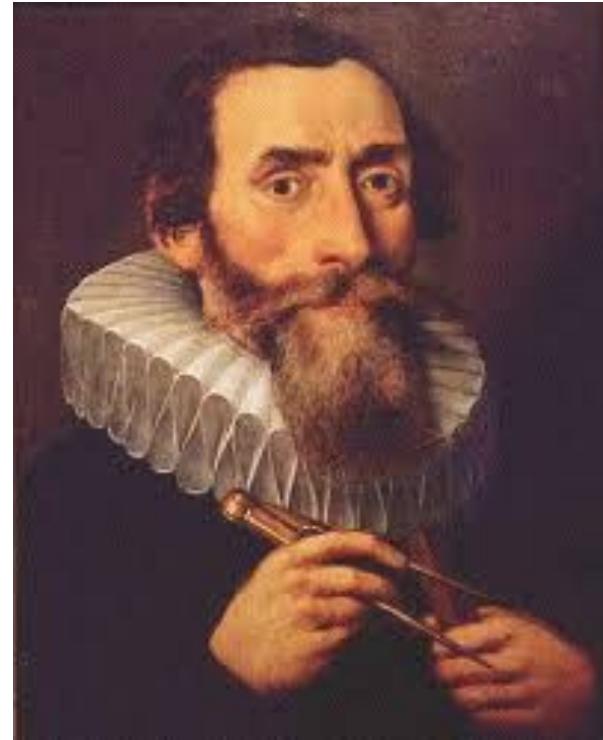


Data → Knowledge

# Long before the ML

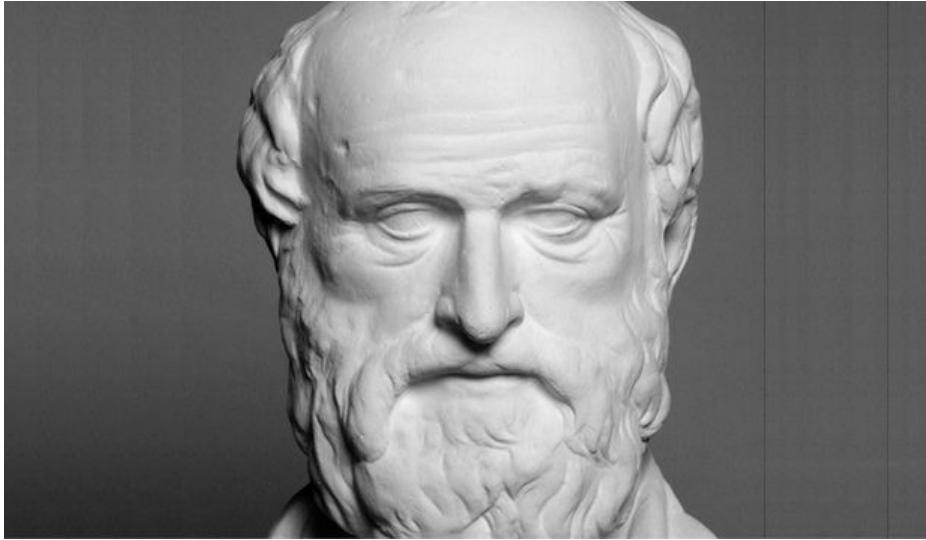


Isaac Newton



Johannes Kepler

# Long before the ML



Eratosthenes

# ML thesaurus

---

girafe  
ai

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



# 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 **feature matrix or design matrix** contains all the observations and their 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

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



# Matrix notation: 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

Feature matrix is usually denoted as  $X \in R^{n \times p}$

where  $n$  is number of objects in dataset and  $p$  is number of properties



# 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)



# 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

**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

# Matrix notation: target



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

Target matrix is usually denoted as  $Y \in R^n$

where  $n$  is number of objects in dataset



# 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:

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



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

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



# 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  $\mathbf{y}$  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

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

# 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
- Model
- Prediction
- Loss function
- Parameter
- Hyperparameter

# Datasets

---

girafe  
ai

03



# Datasets search

Nowadays there are tons of data available on the Internet.

It covers most of the cases you can think of.

So the main problem is to search for the right one!

Let's overview some ways to collect datasets.

# Google dataset search



<https://datasetsearch.research.google.com/>

Contains main info about dataset

and links to sources.

Not all datasets are easily available

The screenshot shows the Google Dataset Search interface. The search bar at the top contains the query "cancer cell segmentation". Below the search bar, there are several filter buttons: "Last updated", "Download format", "Croissant", "Usage rights", "Topic", "Provider", and "Free". The main search results area displays "100+ data sets found". The first result is a dataset from Zenodo titled "Cell Colony Image Segmentation Dataset 1 for T-47D Breast Cancer Cells". It includes links to zenodo.org and explore.openaire.eu, and download options for txt and zip files. The second result is a dataset from Mendeley titled "CCAgT: Images of Cervical Cells with AgNOR Stain Technique". It includes links to data.mendeley.com, and download options for txt and zip files. The third result is a dataset from Figshare titled "Segmentation of organelles in isotropic electron microscopy...". It includes links to janelia.figshare.com, and download options for bin files. The right side of the interface provides detailed information for the first dataset, including its unique identifier (<https://doi.org/10.5281/zenodo.4593510>), the date it was updated (Mar 11, 2021), the provider (Zenodo), authors (Delmon Arous, Stefan Schrunner, Ingunn Hanson, Nina F.J. Edin, Eirik Malinen, Delmon Arous, Stefan Schrunner, Ingunn Hanson, Nina F.J. Edin, Eirik Malinen), and a note about the license (Attribution 4.0 (CC BY 4.0)).

# Kaggle

<https://www.kaggle.com/datasets>



## Datasets

Explore, analyze, and share quality data. [Learn more](#) about data types, creating, and collaborating.

+ New Dataset

Your Work

Search datasets

All datasets Computer Science Education Classification Computer Vision NLP Data Visualization Pre-Trained Model

### Trending Datasets

SI - 100	Moderate
101 - 150	Unhealthy for S
151 - 200	Unhealthy

2023 Air Quality Data for CBSAs

Nikki Perry · Updated 20 hours ago  
Usability 10.0 · 166 kB

16



Smite Item Statistics Data

Matt OP · Updated a day ago  
Usability 10.0 · 15 kB  
1 File (CSV)

16



Machine Learning Engineer Salary in 2024

Chopper53 · Updated 21 hours ago  
Usability 10.0 · 110 kB  
1 File (CSV)

8



FuelConsumption

Krupa Dharamshi · Updated a month ago  
Usability 10.0 · 6 kB  
1 File (CSV)

15

# Hugging Face



HF

# Papers with code



<https://paperswithcode.com/datasets>



9761 dataset results

Search for datasets

Best match

Filter by Modality

Images	2661
Texts	2546
Videos	862
Audio	394
Medical	336
3D	303

**CIFAR-10 (Canadian Institute for Advanced Research, 10 classes)**  
The CIFAR-10 dataset (Canadian Institute for Advanced Research, 10 classes) is a subset of the Tiny Images dataset and consists of 60000 32x32 color images. The images are labelled...  
14,073 PAPERS • 98 BENCHMARKS

**ImageNet**  
The ImageNet dataset contains 14,197,122 annotated images according to the WordNet hierarchy. Since 2010 the dataset is used in the ImageNet Large Scale Visual Recognition...  
13,418 PAPERS • 40 BENCHMARKS

**MS COCO (Microsoft Common Objects in Context)**  
The MS COCO (Microsoft Common Objects in Context) dataset is a large-scale object detection, segmentation, key-point detection, and captioning dataset. The dataset consists of...  
10,128 PAPERS • 92 BENCHMARKS

# UCI ML repository



<https://archive.ics.uci.edu/>

Historical source of datasets.

Most of them collected in 90s

and 00s.

Welcome to the UC Irvine Machine Learning Repository

We currently maintain 664 datasets as a service to the machine learning community. Here, you can donate and find datasets used by millions of people all around the world!

[VIEW DATASETS](#) [CONTRIBUTE A DATASET](#)

### Popular Datasets

	<b>Iris</b> A small classic dataset from Fisher, 1936. One of the earliest known datasets used for... <a href="#">Classification</a> <span>150 Instances</span> <span>4 Features</span>
	<b>Heart Disease</b> 4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach <a href="#">Classification</a> <span>303 Instances</span> <span>13 Features</span>
	<b>Dry Bean</b> Images of 13,611 grains of 7 different registered dry beans were taken with a high-res... <a href="#">Classification</a> <span>13.61K Instances</span> <span>16 Features</span>
	<b>Rice (Cammao and Osmancik)</b> A total of 3810 rice grain's images were taken for the two species, processed and feat... <a href="#">Classification</a> <span>3.81K Instances</span> <span>7 Features</span>

### New Datasets

	<b>PhiUSIIL Phishing URL (Website)</b> PhiUSIIL Phishing URL Dataset is a substantial dataset comprising 134,850 legitimate ... <a href="#">Classification</a> <span>235.8K Instances</span> <span>54 Features</span>
	<b>RT-IoT2022</b> The RT-IoT2022, a proprietary dataset derived from a real-time IoT infrastructure, is in... <a href="#">Classification, Regres...</a> <span>123.12K Instances</span> <span>84 Features</span>
	<b>Regensburg Pediatric Appendicitis</b> This repository holds the data from a cohort of pediatric patients with suspected app... <a href="#">Classification</a> <span>782 Instances</span> <span>59 Features</span>
	<b>National Poll on Healthy Aging (NPHA)</b> This is a subset of the NPHA dataset filtered down to develop and validate machine le... <a href="#">Classification</a> <span>714 Instances</span> <span>15 Features</span>

# Standardization initiative



Croissant dataset standard

<https://mlcommons.org/working-groups/data/croissant/>

<https://research.google/blog/croissant-a-metadata-format-for-ml-ready-datasets/>

The collage includes:

- Hugging Face:** A screenshot showing the "titanic-survival" dataset card, highlighting the "Croissant" file format.
- kaggle:** A screenshot showing the "titanic\_1" dataset page, with a prominent "Download (11 kB)" button and a "Export metadata as Croissant" link.
- OpenML:** A screenshot showing the "titanic" dataset page, with a "Download" button and a "Load" button.
- Google Search:** A screenshot of a Google search results page for "titanic". The first result is a link to "Croissant Editor" with the text: "Croissant is a format for Machine Learning datasets Learn more about this at mlcommons.org/croissant".
- Search:** A large image of a croissant with the word "Search" written vertically next to it.
- Create:** A screenshot of the "Croissant Editor" interface, showing a form to create a new dataset named "Titanic".
- Load:** A screenshot of the "Croissant Editor" interface, showing a "Load" section with a "TensorFlow Datasets" icon.
- Code Block:** A code snippet demonstrating how to use the TensorFlow Datasets API to build a Croissant dataset:

```
import tensorflow_datasets as tfds
builder = tfds.dataset_builders.CroissantBuilder(
    json_url='https://raw.githubusercontent.com/mlcommons/croissant/main/datasets/titanic/titanic.json',
    file_format='array-record',
)
builder.download_and_prepare()
ds = builder.as_data_source()
print(ds['default'][0])
```

# Standardization initiative



Croissant dataset standard

<https://mlcommons.org/working-groups/data/croissant/>

<https://research.google/blog/croissant-a-metadata-format-for-ml-ready-datasets/>

The collage includes:

- A screenshot of the Hugging Face dataset card for "titanic-survival", showing the Croissant format option.
- A screenshot of the Kaggle interface, showing a dataset card for "titanic\_1" with a "Download (11 kB)" button and a "Export metadata as Croissant" link.
- A screenshot of the OpenML interface, showing a dataset card for "titanic" with a "Download" button.
- A screenshot of a Google search results page for "titanic", where the Croissant format is highlighted in the search filters.
- A central graphic with the words "Search", "Create", "Load", and "Download" arranged around a central icon of a croissant.
- A screenshot of the Croissant Editor interface, showing a dataset named "Titanic" with a description of the original dataset.
- A code snippet in a terminal window demonstrating the use of TensorFlow's `tfds` library to build a Croissant dataset from a URL:

```
import tensorflow_datasets as tfds
builder = tfds.dataset_builders.CroissantBuilder(
    json_url='https://raw.githubusercontent.com/mlcommons/croissant/main/datasets/titanic/croissant.json',
    file_format='array_record',
)
builder.download_and_prepare()
ds = builder.as_data_source()
print(ds['default'][0])
```

# Maximum Likelihood Estimation

---

girafe  
ai

04



# Parametric and nonparametric models

Nonparametric statistics is a type of statistical analysis that makes minimal assumptions about the underlying distribution of the data being studied. Often these models are infinite-dimensional, rather than finite dimensional, as is parametric statistics.

Nonparametric statistics can be used for descriptive statistics or statistical inference. Nonparametric tests are often used when the assumptions of parametric tests are evidently violated.

[© Common knowledge site](#)



# Likelihood maximization

Consider the most simple case of discrete features and target.

Denote dataset  $X, Y$  generated by distribution with parameter  $\theta$

Likelihood of a parameter is defined as probability of sampling this particular data in case underlying distribution is defined by this parameter.

Maximization of likelihood means we choose the most probable parameters having this particular dataset

$$L(\theta|X, Y) = P(X, Y|\theta) \rightarrow \max_{\theta}$$

Note that likelihood is not probability function of  $\theta$



# i.i.d. property

We can employ i.i.d property of data samples to split probability of the whole dataset into independent problems

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

Then we apply logarithm function to both parts of equation above

$$\log P(X, Y | \theta) = \sum_i \log P(x_i, y_i | \theta)$$

The latter expression is easier to operate with:  
later we will predict log-probability of each object directly

# Log-likelihood equivalence



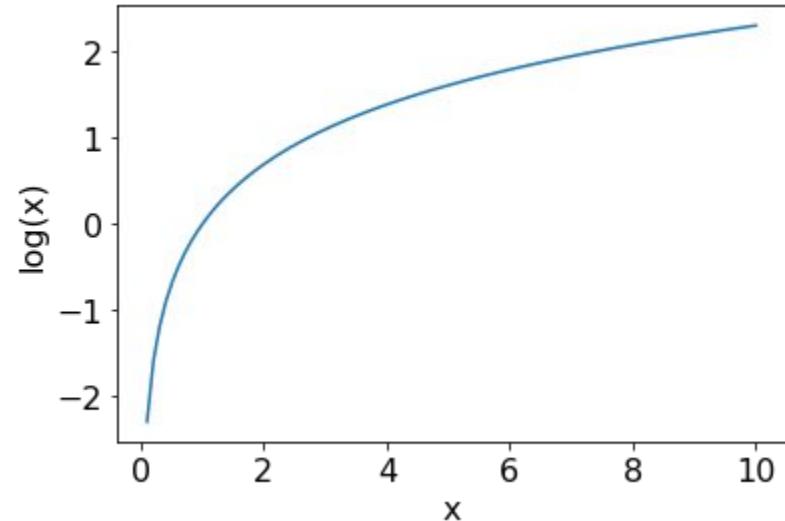
Since logarithm is a convex function on open set, it preserves maximum of expression when applied, so that

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

and

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

have the same solutions in terms of  $\theta$



# Maximum Likelihood Estimation



$$\hat{\theta} = \arg \max_{\theta} L(\theta | X, Y)$$

is called maximum likelihood estimation of model parameters.

In optimization theory functions are usually minimized, so the same problem could be reformulated using **Negative Log-Likelihood (NLL)** loss

$$\hat{\theta} = \arg \min_{\theta} - \sum_i \log P(x_i, y_i | \theta)$$



# Note

Here we formulate MLE in terms of probability which means we assume finite number of parameter values.

Defining MLE for infinite parameters set is left as easy exercise for you.

$$\hat{\theta} = \arg \min_{\theta} - \sum_i \log P(x_i, y_i | \theta)$$

# Machine Learning problems overview

---

girafe  
ai

05



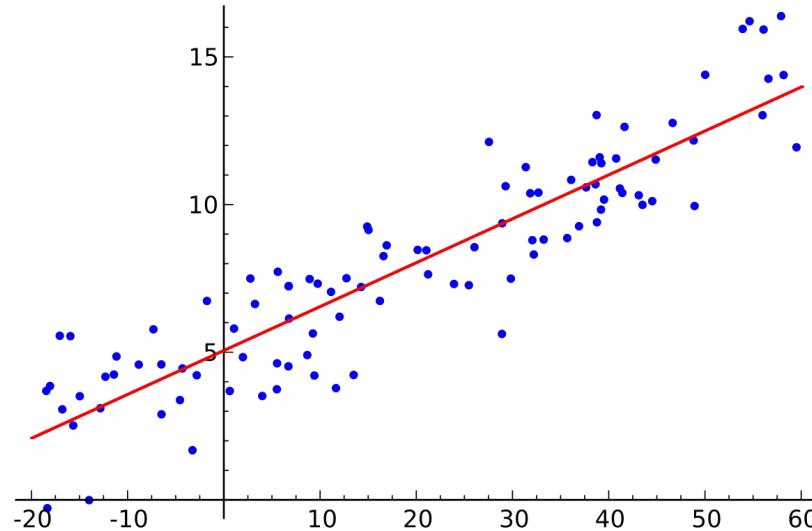
# Supervised learning problem statement

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

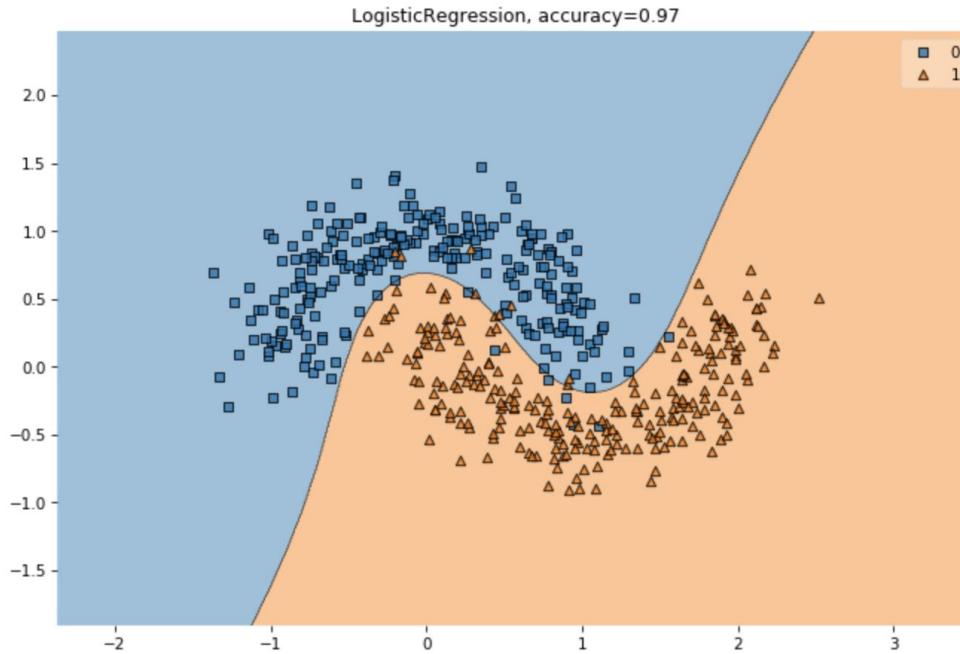
Estimate of the  
regression slope

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

Value of X for  
observation i

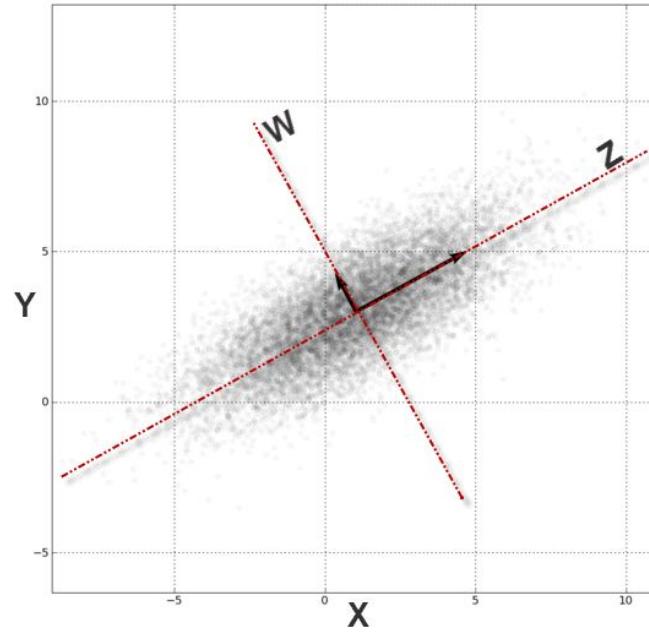


- Regression problem
- Classification problem





- Regression problem
- Classification problem
- Dimensionality reduction



# Naïve Bayes classifier

---

girafe  
ai

06



# 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

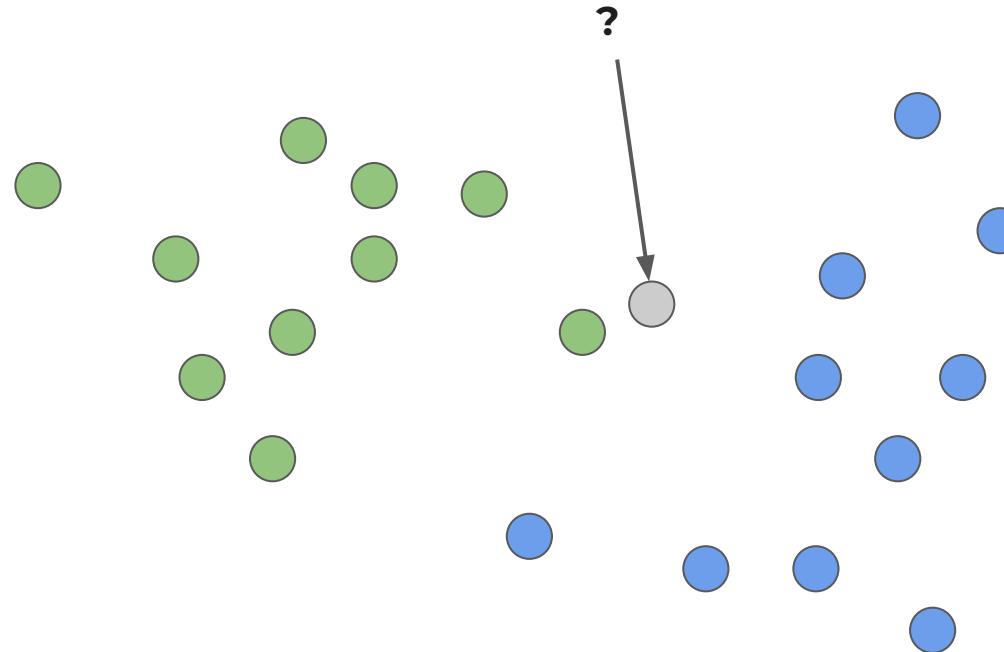
# k Nearest Neighbors

---

girafe  
ai

07

# Intuition





# kNN model

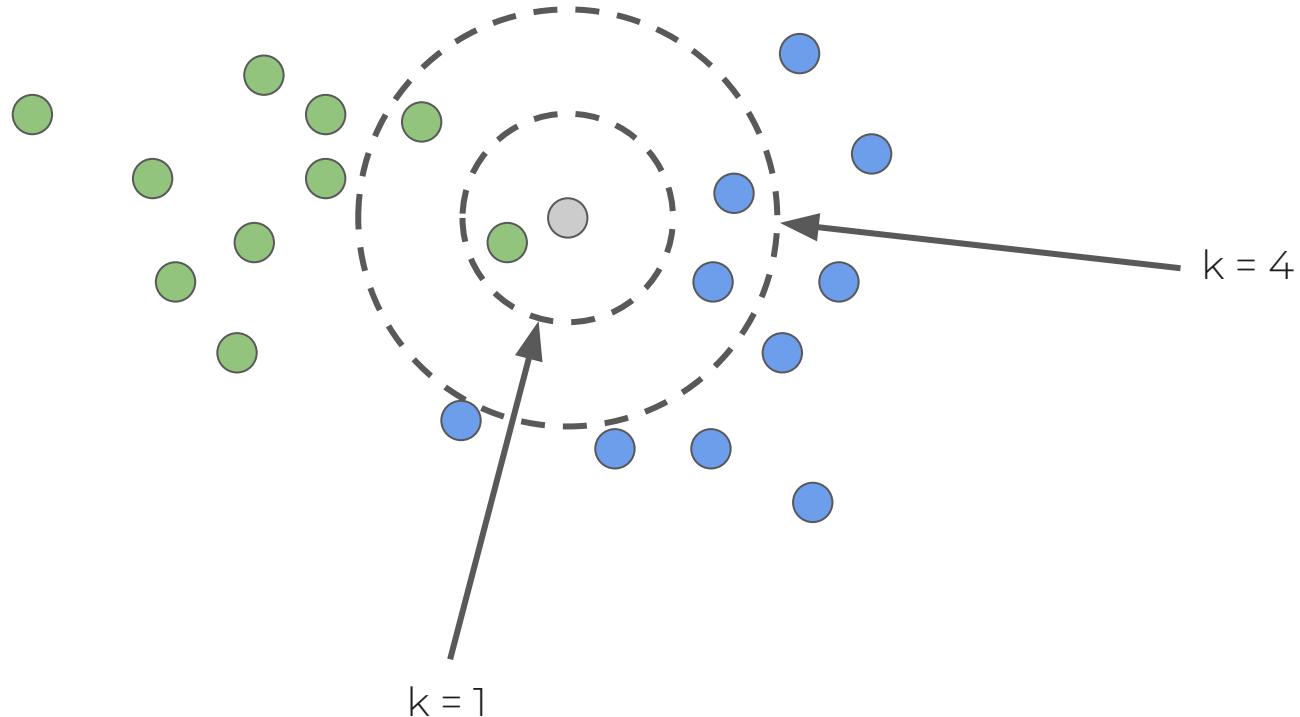
Given a new observation:

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?



1. The number of neighbors  $k$





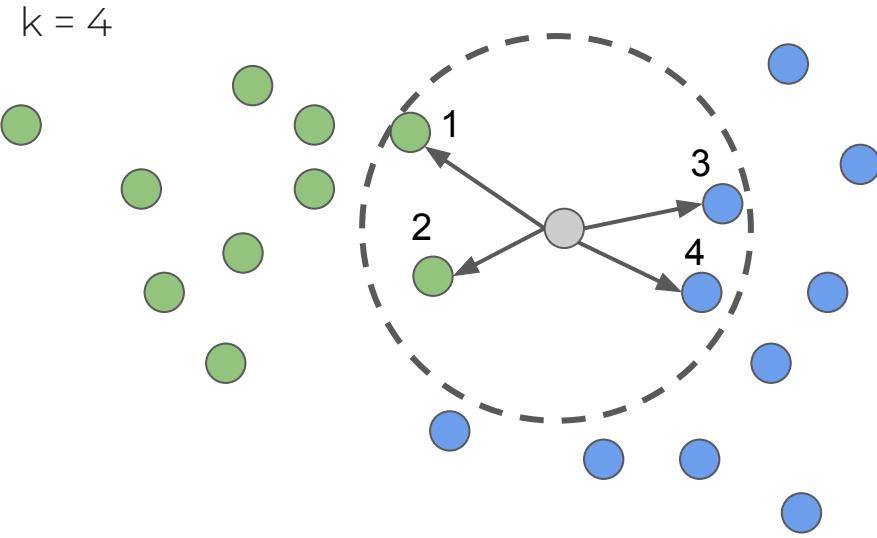
# How to make it better?

1. The number of neighbors k
2. The distance measure between samples
  - a. Euclidean
  - b. Minkowski distances
  - c. cosine
  - d. Hamming
  - e. etc.
3. Weighted neighbours

They are **hyperparameters** for kNN model.



# Weighted kNN



- Weights can be adjusted according to the neighbors order

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

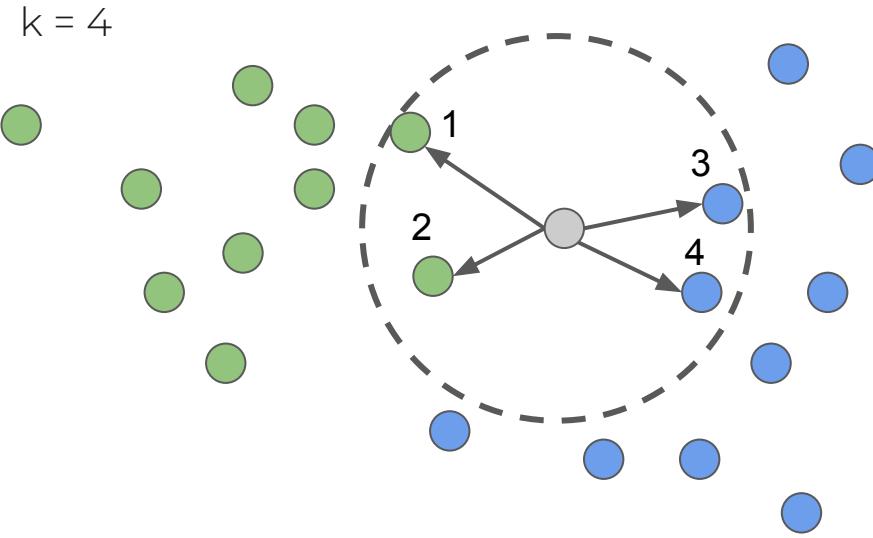
- or on the distance itself

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

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



# Weighted kNN



- Weights can be adjusted according to the neighbors order,

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

- or on the distance itself

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

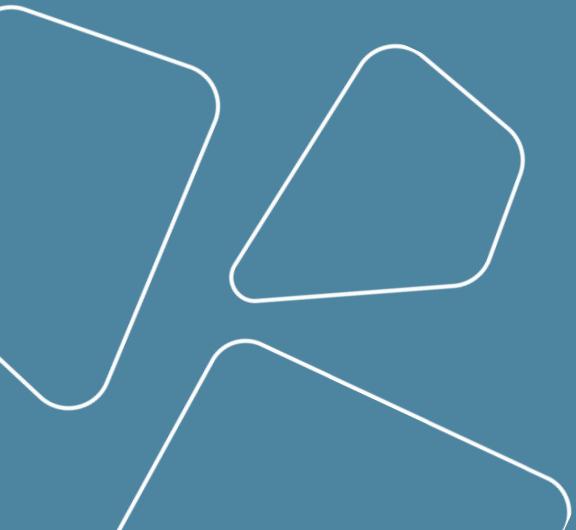
$$p_{\text{blue}} = \frac{w(\mathbf{x}_3) + w(\mathbf{x}_4)}{w(\mathbf{x}_1) + w(\mathbf{x}_2) + w(\mathbf{x}_3) + w(\mathbf{x}_4)}$$



# Takeouts

- 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

# Revise

- 
- A decorative graphic in the bottom-left corner consists of several white-outlined geometric shapes on a teal background. It includes a large irregular polygon on the left, a smaller pentagon-like shape in the center, and some curved lines at the bottom.
- 1. ML and AI overview
  - 2. Thesaurus and notation
  - 3. Maximum Likelihood Estimation
  - 4. Some Machine Learning problems
    - a. Classification
    - b. Regression
    - c. Dimensionality reduction
  - 5. Naïve Bayes classifier
  - 6. k Nearest Neighbours (kNN)

# Thanks for attention!

Questions?



girafe  
ai

