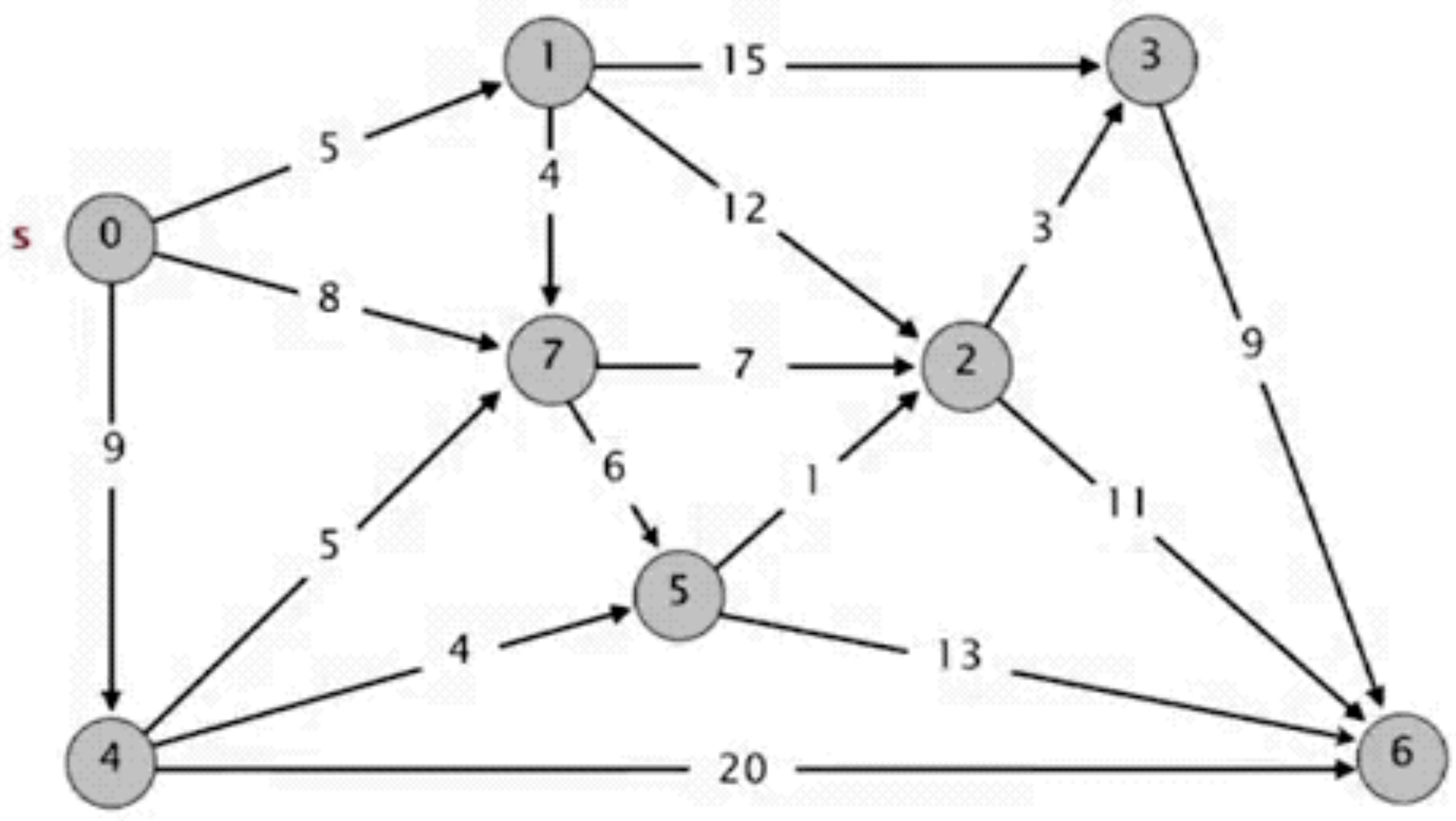
Graph Theory III

Seminar 14.

Minimum distances
Dijkstra algorithm



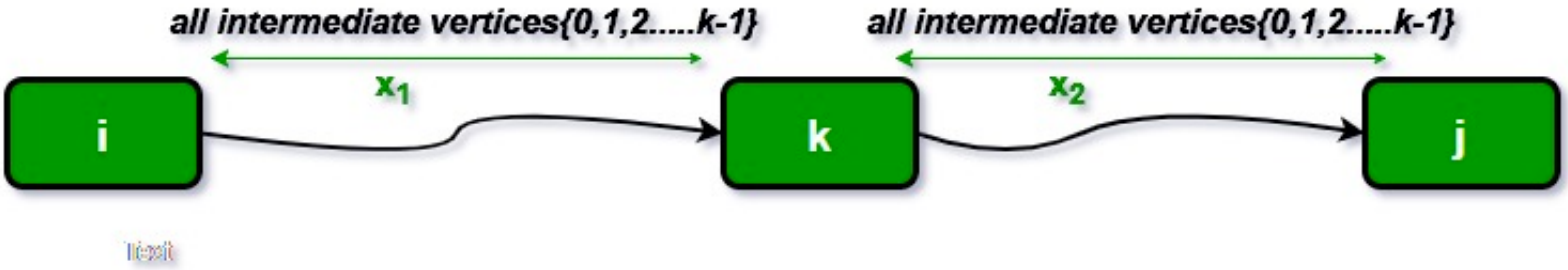
an edge-weighted DAG

0→1	5.0
0→4	9.0
0→7	8.0
1→2	12.0
1→3	15.0
1→7	4.0
2→3	3.0
2→6	11.0
3→6	9.0
4→5	4.0
4→6	20.0
4→7	5.0
5→2	1.0
5→6	13.0
7→5	6.0

Minimum distances

Floyd Warshall Algorithm

- is an all pair shortest path algorithm
- unlike Dijkstra and Bellman Ford which are single source shortest path algorithms



Minimum distances

Floyd Warshall Algorithm

- initialize Solution matrix $\boldsymbol{S}$ = Adjacency matrix $\boldsymbol{G}$
- update $\boldsymbol{S}$ by considering all vertices as intermediate
- **idea**: pick all vertices one by one and update all shortest paths which include the picked vertex as an intermediate vertex in the shortest path.

Minimum distances

Floyd Warshall Algorithm

- **idea**: pick all vertices one by one and update all shortest paths which include the picked vertex as an intermediate vertex in the shortest path.

- For every pair **(i, j)** of the source and destination, there are two possible cases:

  - **k** vertex is not an intermediate vertex in shortest path from **i** to **j** => keep the value of **dist[I][j]**

  - **k** vertex is an intermediate vertex in shortest path from **i** to **j** => update the value of **dist[i][j]**

$$\mathrm{dist}[i][j] \overset{?}{>} \mathrm{dist}[i][k] + \mathrm{dist}[k][j] \longrightarrow \mathrm{dist}[i][k] + \mathrm{dist}[k][j]$$

Disjoint set union

Kruskal's algorithm

- Iterate through edges in non-decreasing order
- Check for safe edge => Add to the MST subgraph if TRUE

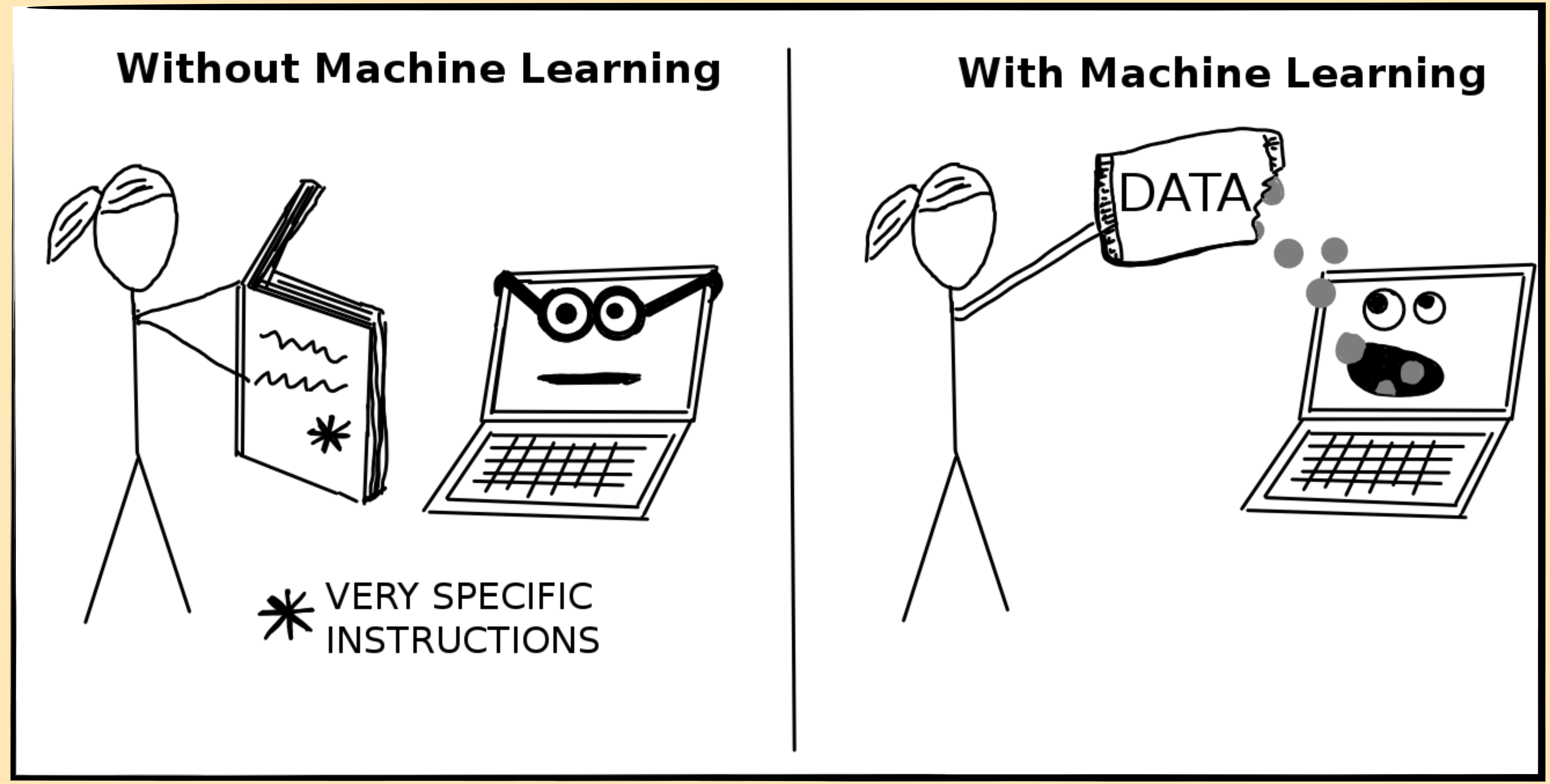
find() – identify the root index of particular node

union() – join two subsets

kruskal() – take edges - check for cycles - unite

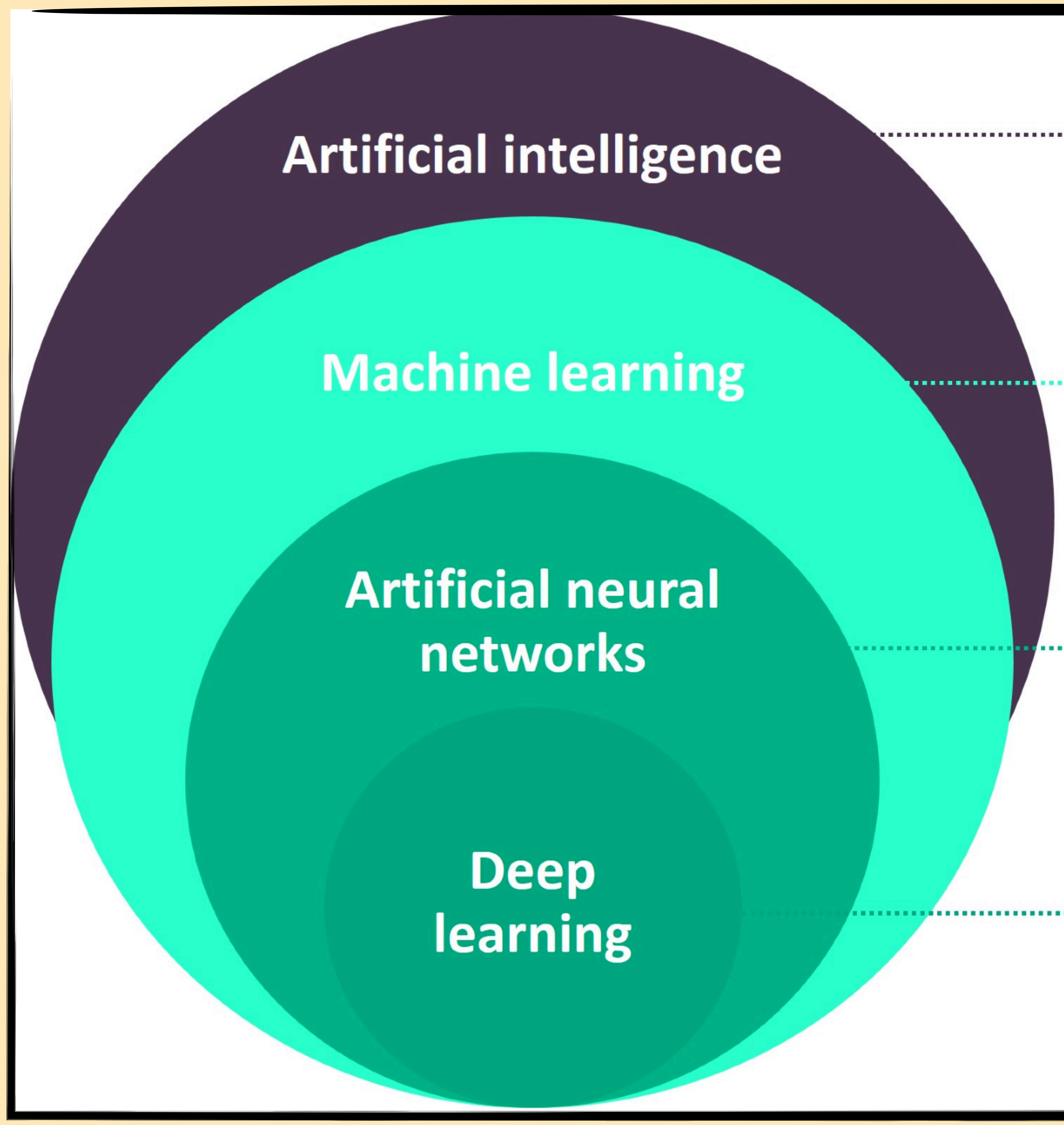
Machine Learning

1. Introduction



1. Introduction

Differences with DL



Artificial intelligence (AI)
Any techniques that enable machines to solve a task in a way like humans do

Machine learning (ML)
Algorithms that allow computers to learn from examples without being explicitly programmed

Artificial neural networks (ANN)
Brain-inspired machine learning models

Deep learning (DL)
A subset of ML which uses deep artificial neural networks as models and automatically builds a hierarchy of data representations

1. Introduction

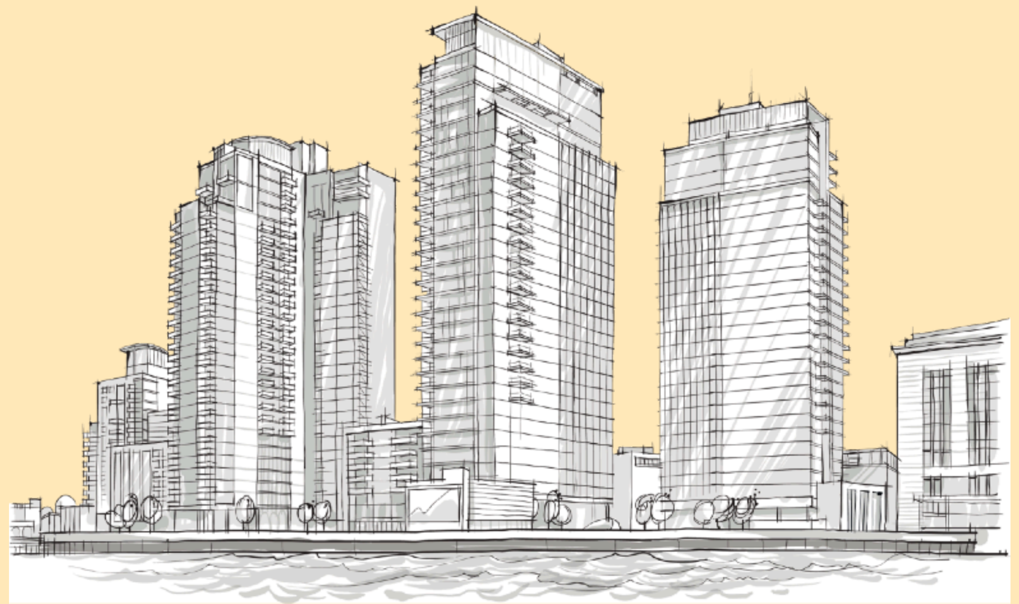
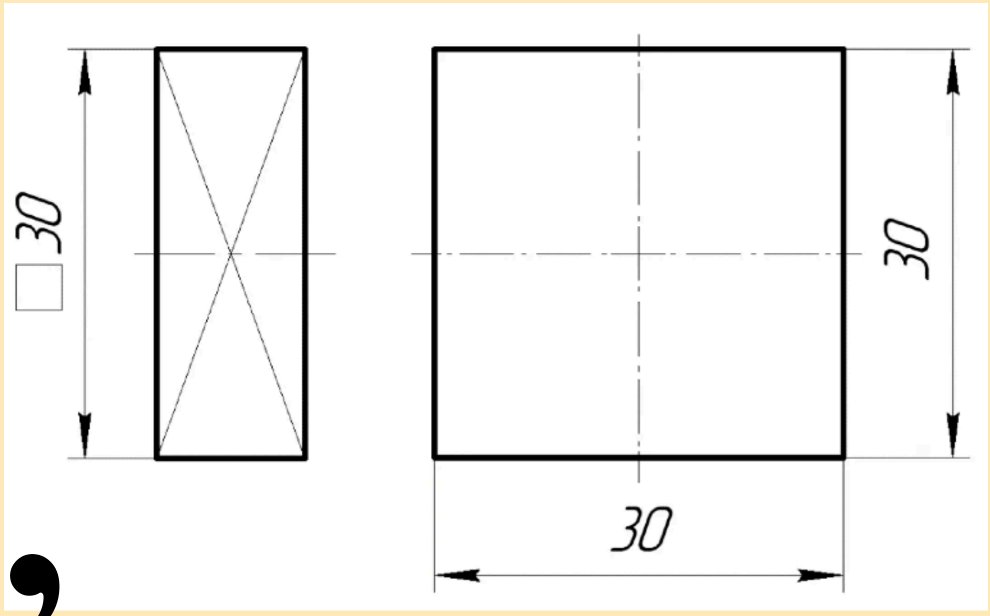
Basic Definitions

- **ML** studies ways to extract patterns from a limited number of examples
- **Object,** x - what for do we make predictions
- Object is identical to its **feature** description. Hence, $\boldsymbol{x}$ is a combination of all object's features
- The value we want to predict is called **target variable,** $\boldsymbol{y}$

1. Introduction

Basic Definitions

age of building   area of flat   time to subway

( , 

,)

w_1 w_2 w_3

$$x_i \in \mathbb{X} = (x_i)_{i=0}^{n}$$

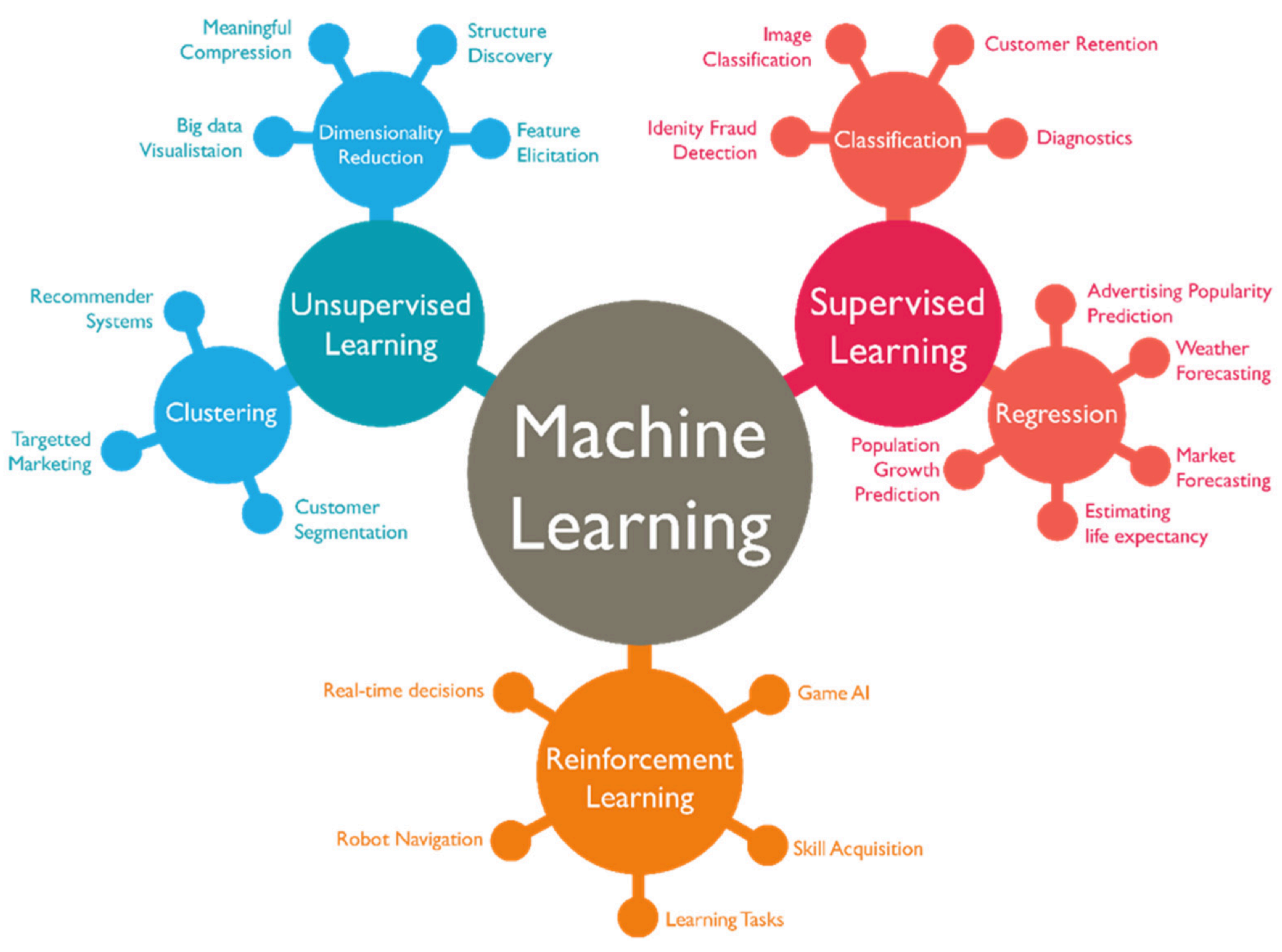
$$y_i \in \mathbb{Y} = (y_i)_{i=0}^{n}$$

1. Introduction

Types of task: supervised learning

- **Classification:**
  - *Binary:* $\mathbb{Y} = \{0,1\}$
  - *Multi-class:* $\mathbb{Y} = \{1,\ldots,K\}$
  - *Multi-label:* $\mathbb{Y} = \{0,1\}^k$

- **Regression**



2. Task formulation

Matrix form

- **Training data:**
  - ***Matrix*** *"objects-features": $X \in \mathbb{R}^{\ell \times d}$, где ℓ - число объектов, d - признаков*

Year	Population	Violent crime total	Murder and nonnegligent Manslaughter	Legacy rape /1	Revised rape /2	Robbery	Aggravated assault	Property crime total	Burglary	Larceny-theft	Motor vehicle theft	Violent Crime rate
1960	179323175	288460	9110	17190		107840	154320	3095700	912100	1855400	328200	160.9
1961	182992000	289390	8740	17220		106670	156760	3198600	949600	1913000	336000	158.1
1962	185771000	301510	8530	17550		110860	164570	3450700	994300	2089600	366800	162.3
1963	188483000	316970	8640	17650		116470	174210	3792500	1086400	2297800	408300	168.2
1964	191141000	364220	9360	21420		130390	203050	4200400	1213200	2514400	472800	190.6
1965	193526000	387390	9960	23410		138690	215330	4352000	1282500	2572600	496900	200.2

- ***Task:*** $\alpha : \mathbb{X} \to \mathbb{Y}$ - elaborate an algorithm to predict *target* from *features*
  - Algorithm's quality control: $Q(\alpha, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (\alpha(x_i) - y_i)^2$ - ***Loss function***

2. Task formulation

Loss function

- ***Task:*** $\alpha : \mathbb{X} \rightarrow \mathbb{Y}$ - elaborate an algorithm to predict *target* from *features*
    - Algorithm's quality control: $Q(\alpha, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (\alpha(x_i) - y_i)^2$ - ***Loss function***

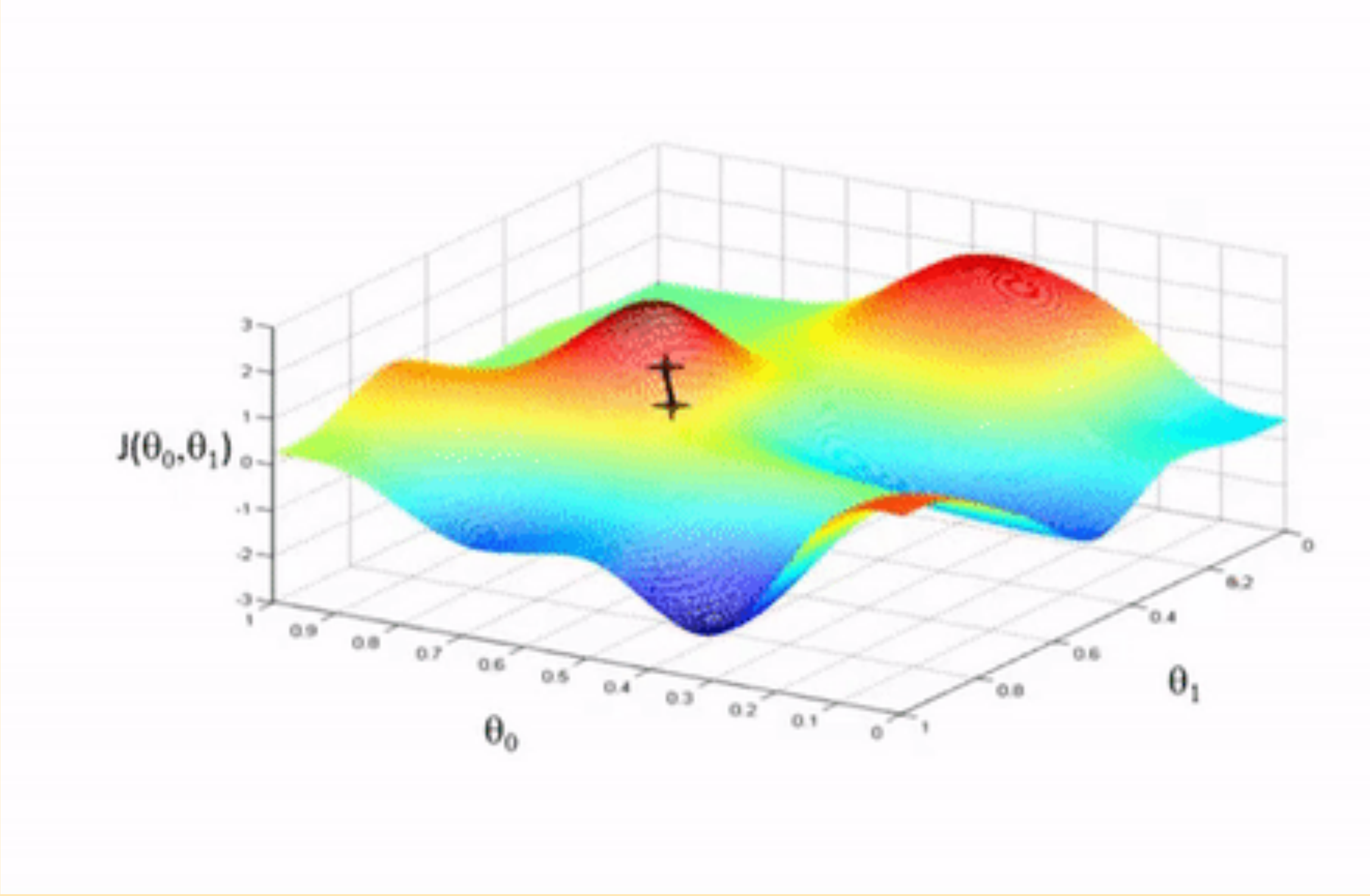
- ***Loss function -*** measure of the error's extent for one particular sample: $L(y, z) = (y - z)^2$

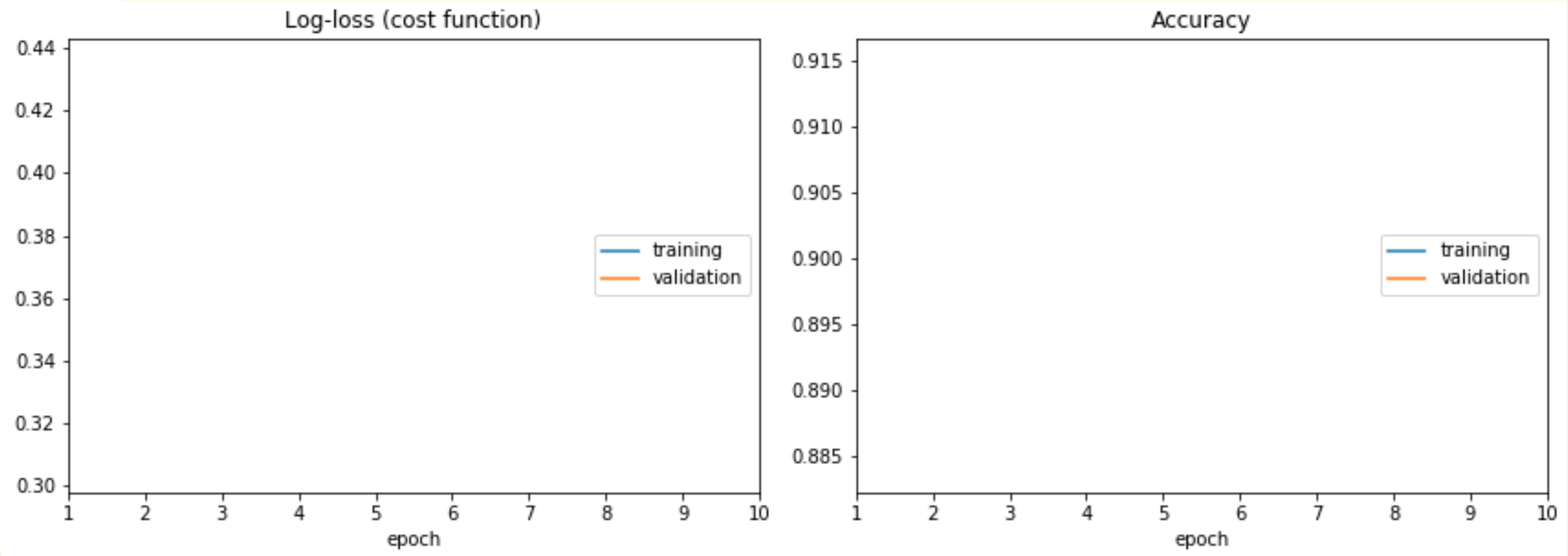
$$Q(\alpha, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (\alpha(x_i) - y_i)^2$$

$$x = (x_{\boldsymbol{метро}}, x_{\boldsymbol{слышимость}} \ldots) \rightarrow a(x) = w_o + w_1 x_1 + \ldots + w_d x_d$$

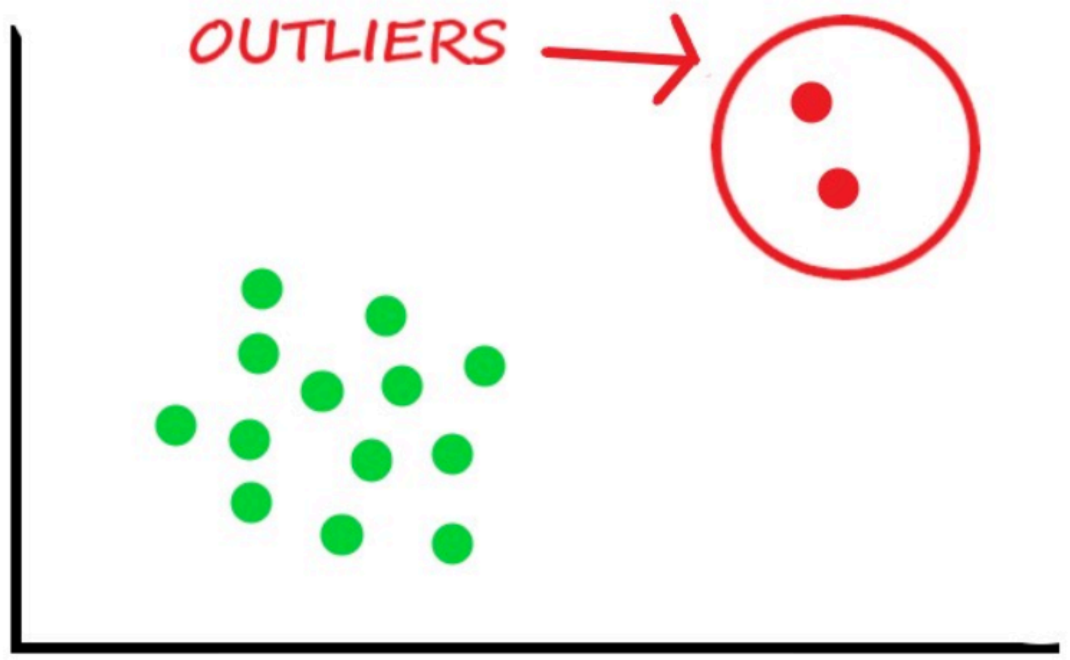
3. Linear models

- ***Linear models family:*** $\mathscr{A} = \{a(x) = w_o + w_1 x_1 + \ldots + w_d x_d \,|\, w_0, w_1, \ldots, w_d \in \mathbb{R}\}$
- ***MSE** to find best:* $\frac{1}{\ell} \sum (w_0 + \sum w_j x_{ij} - y_i)^2 \to \min\limits_{w_0, w_1, \ldots}$



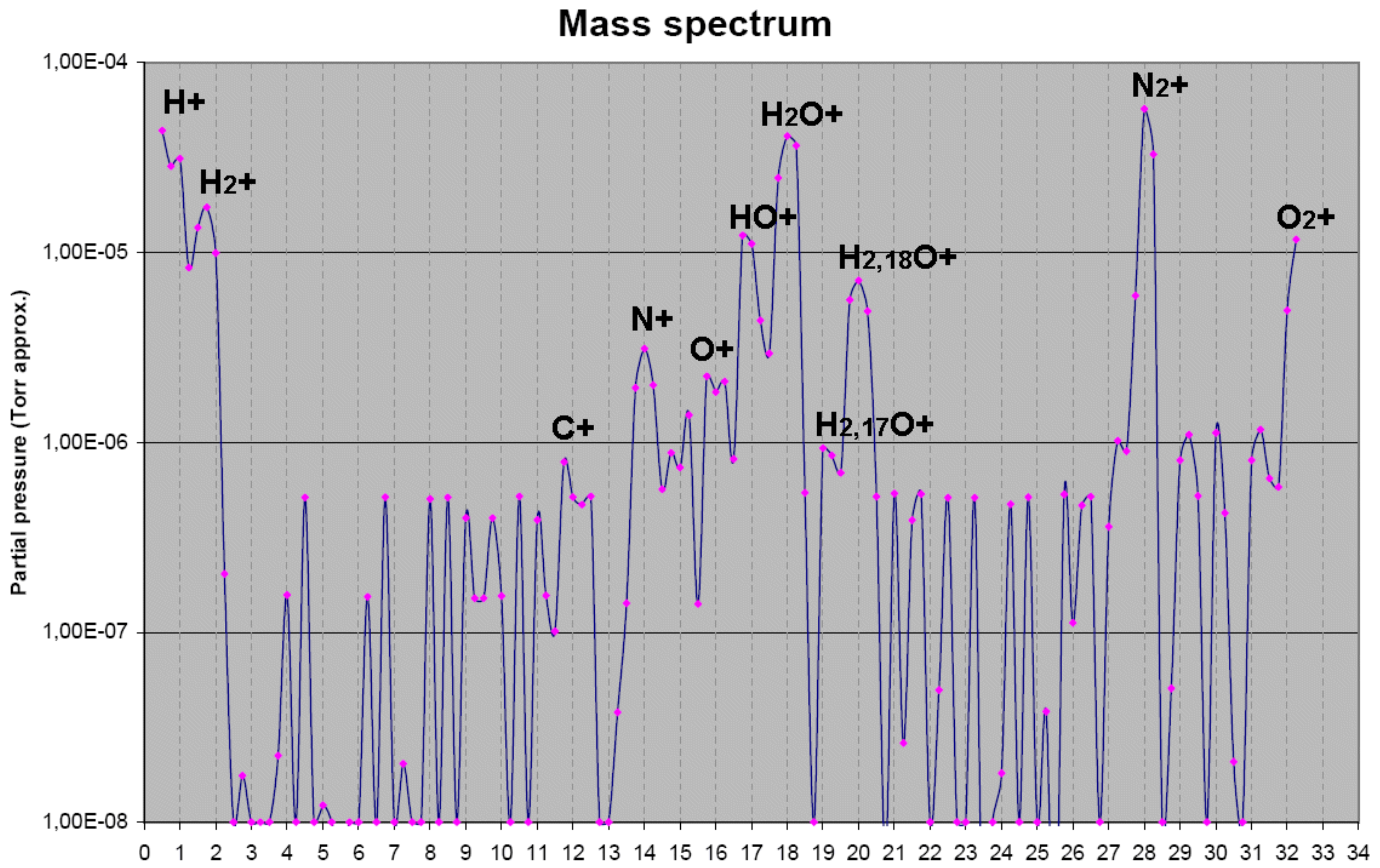


4. Preprocessing

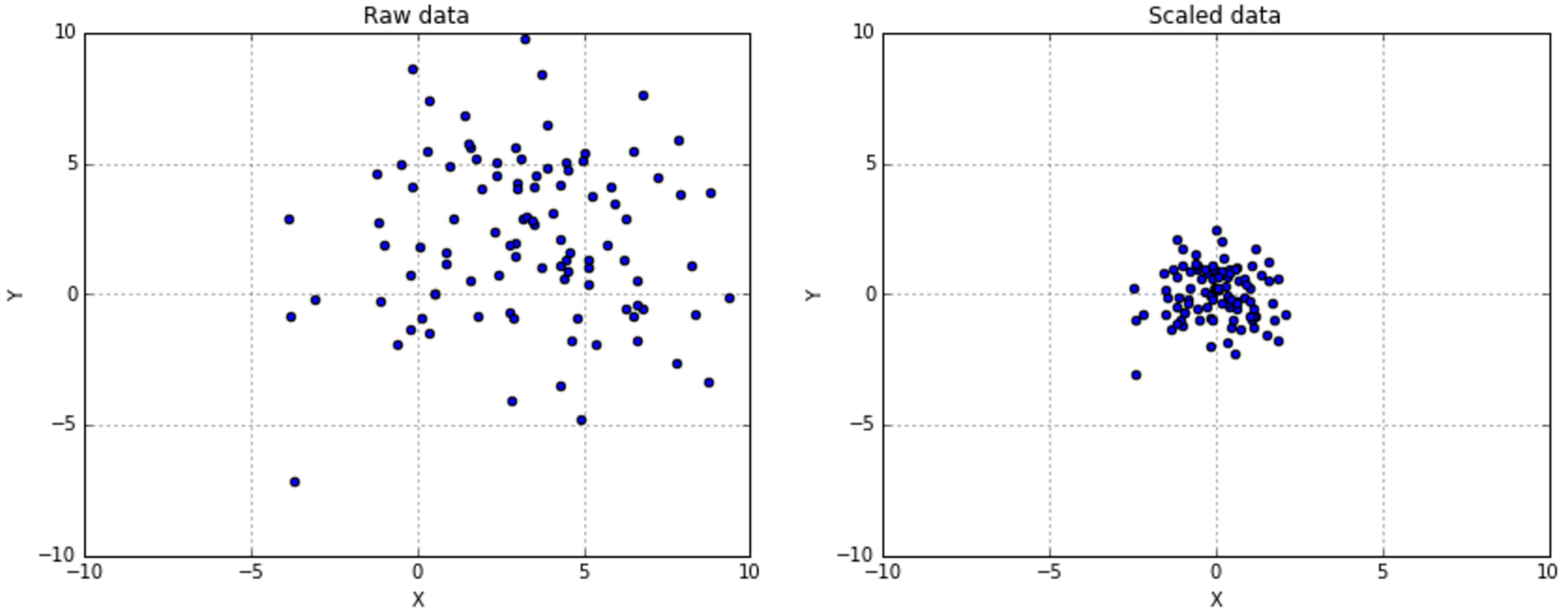


Remove outliers

Reduce noise



Scale



5. ML phases

I. Formulate the task

II. Identify features

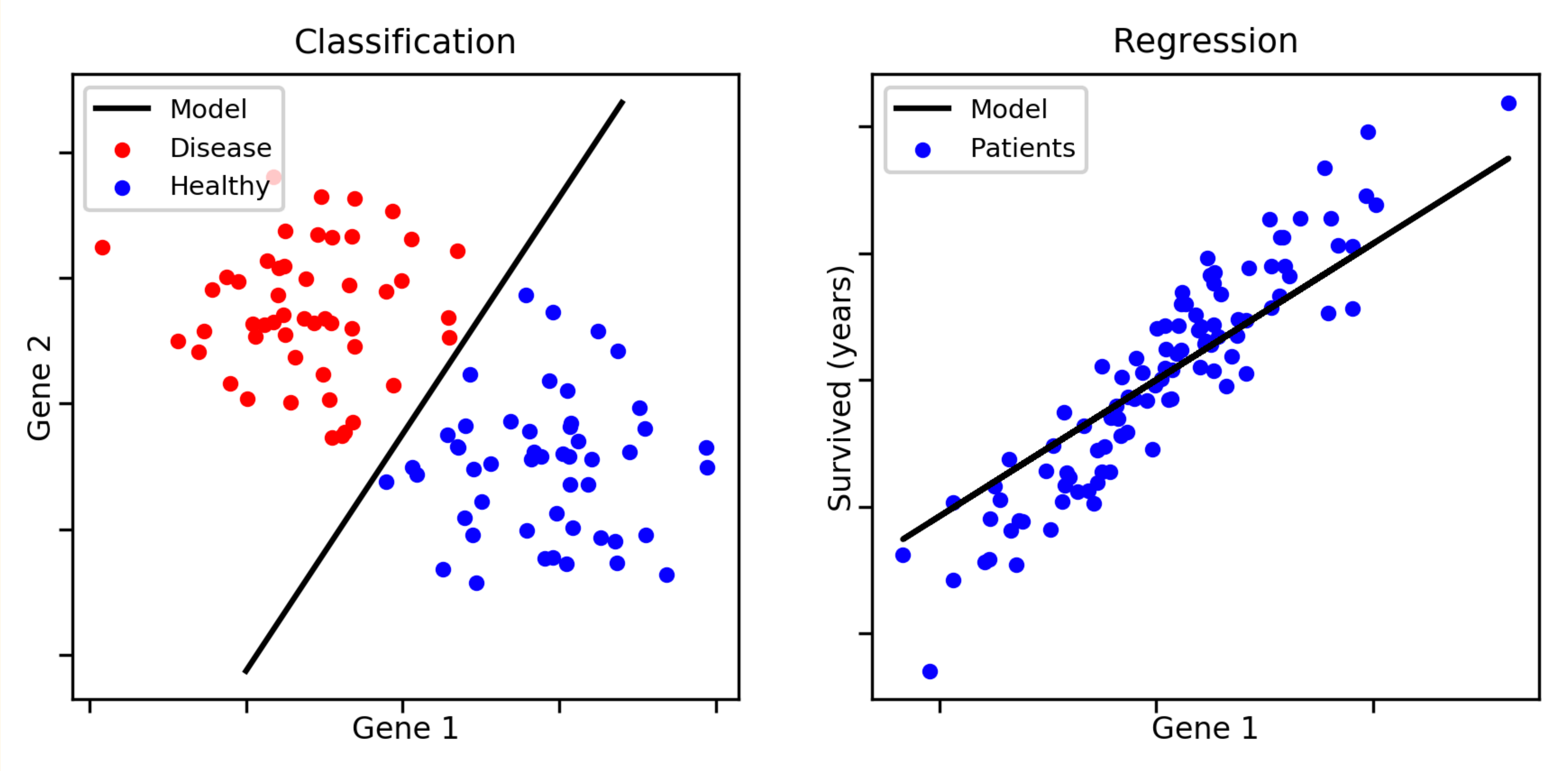
III. Create the dataset

IV. Select the loss function

V. Preprocess data

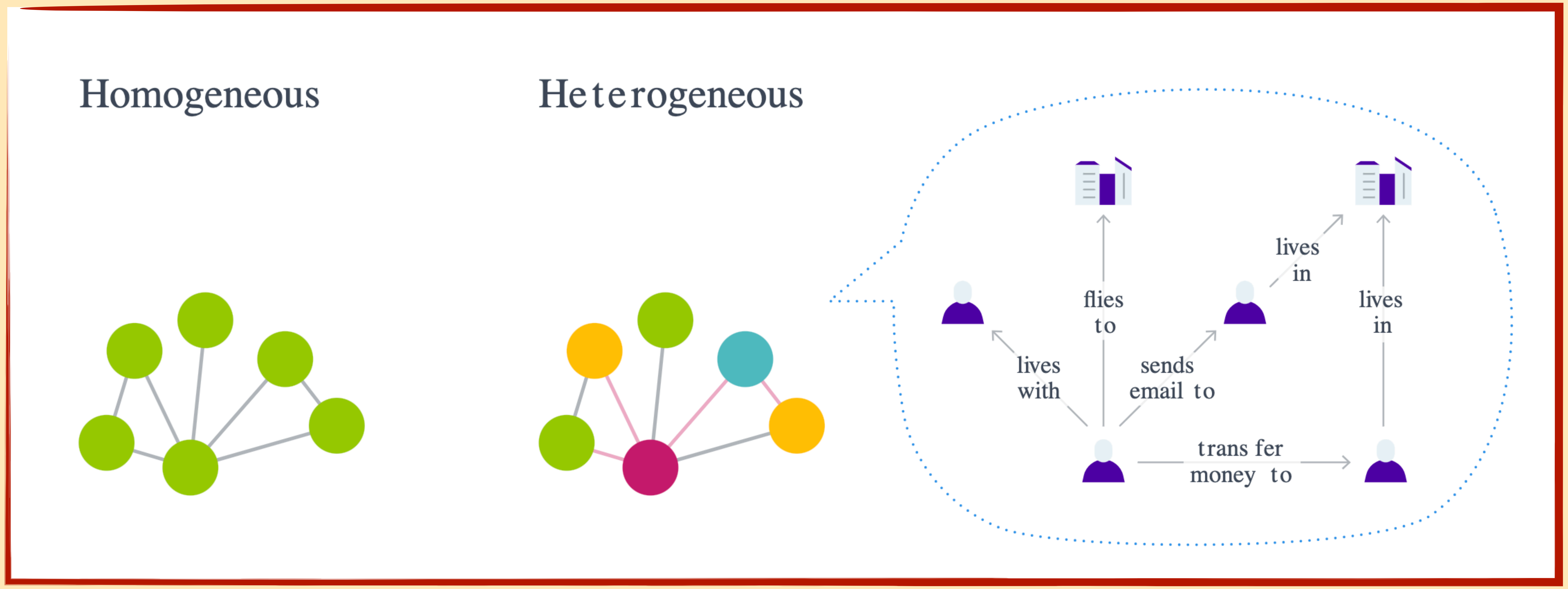
VI. Build the model & Train it

VII. Estimate model's quality

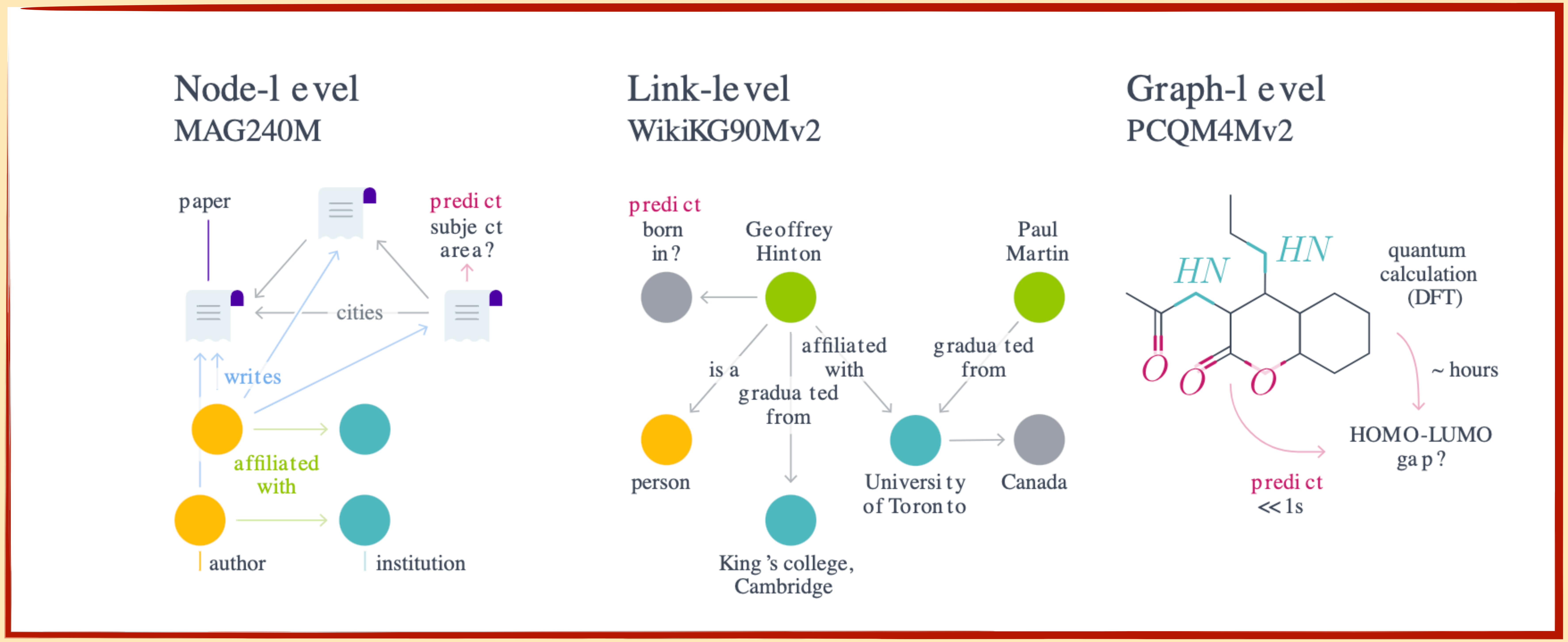


Graphs & ML

Types of graphs

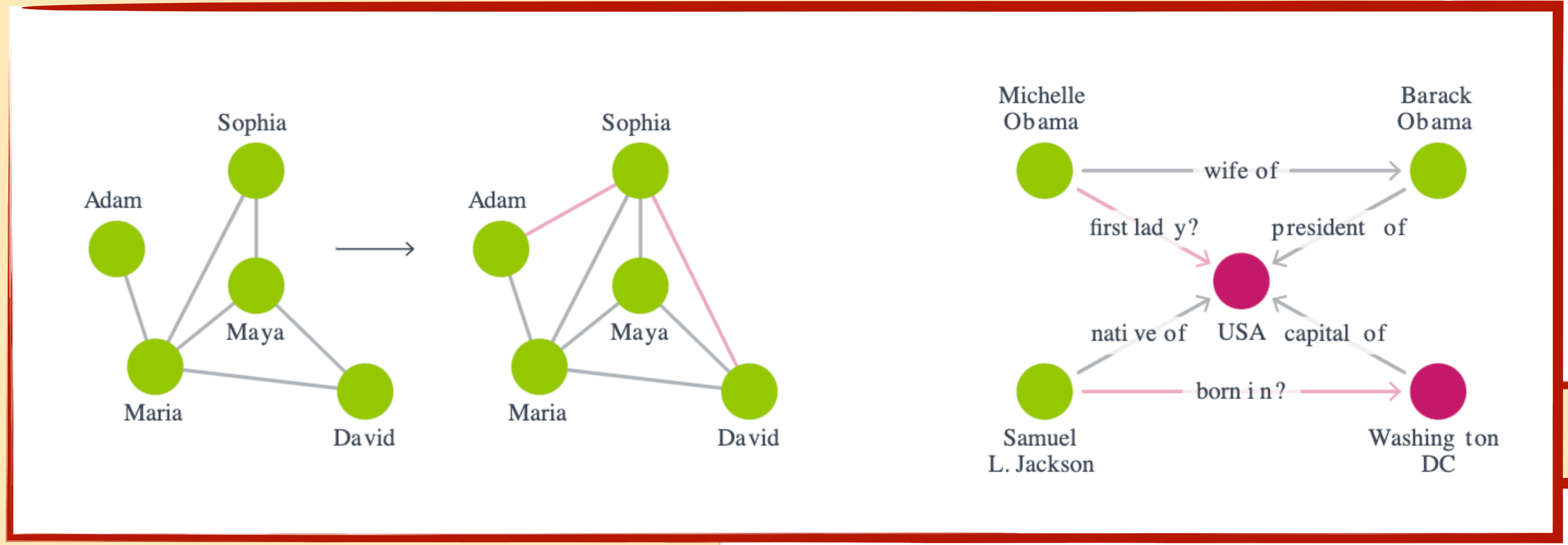


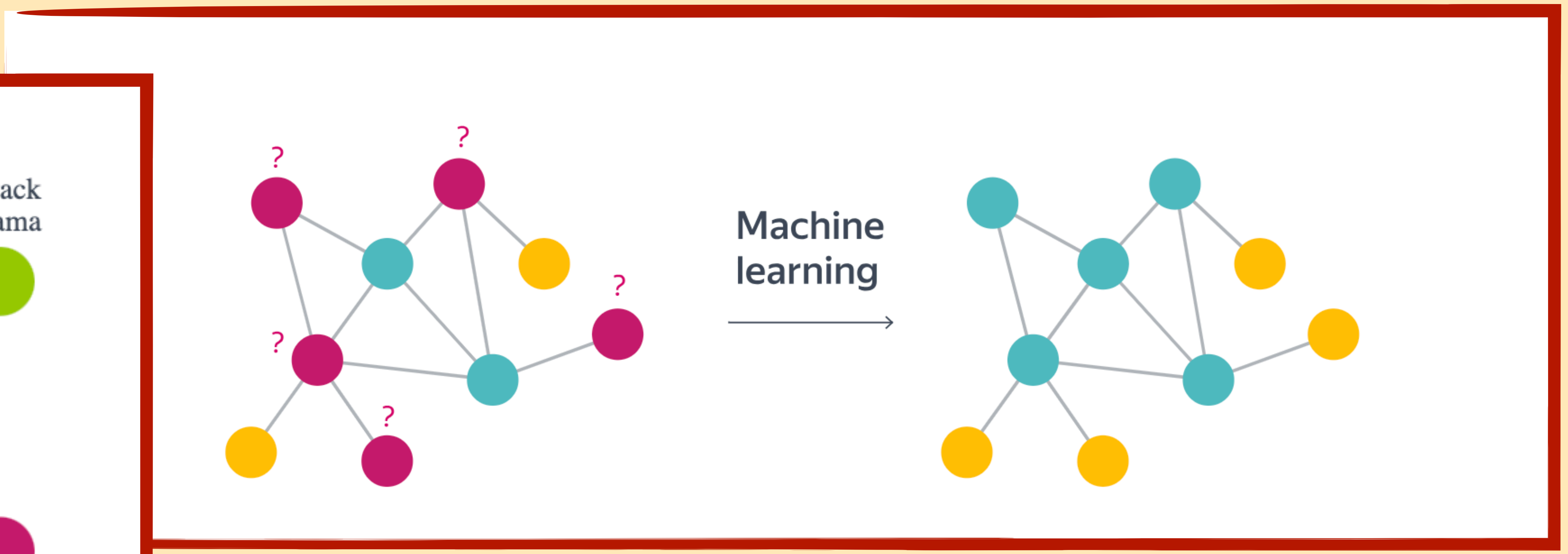
Graphs & ML
Types of tasks

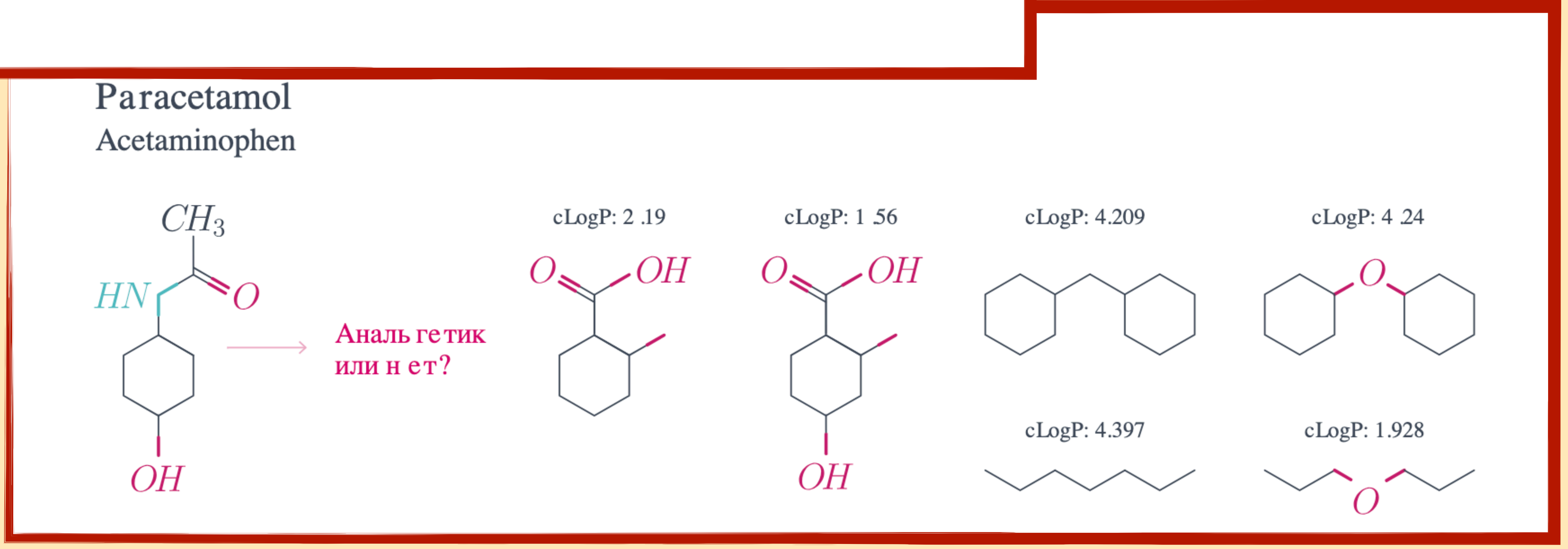


Graphs & ML

Types of tasks



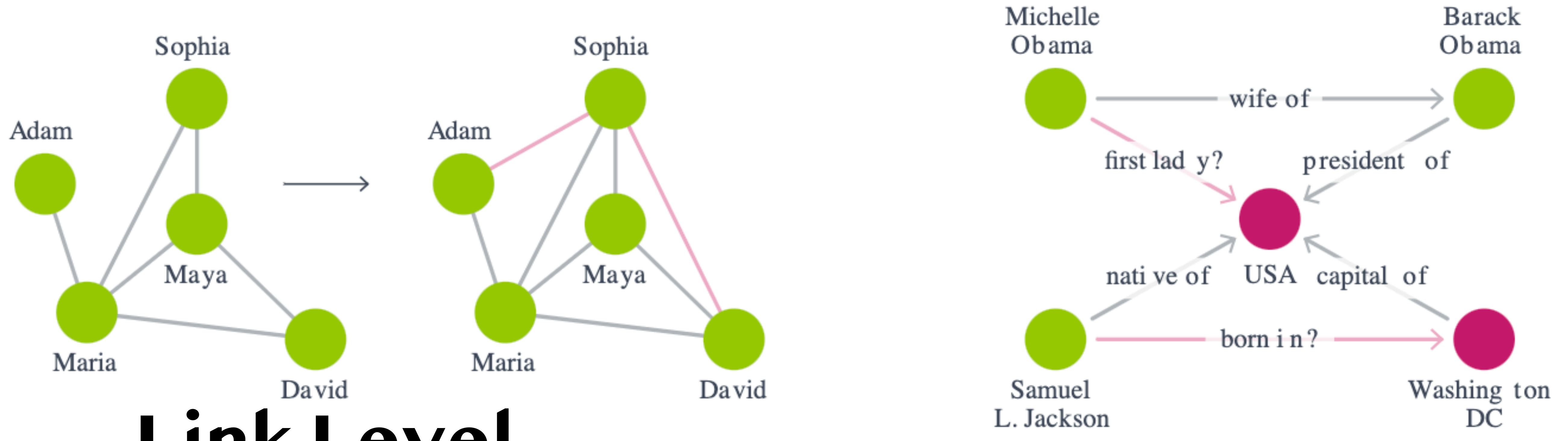




Graphs & ML

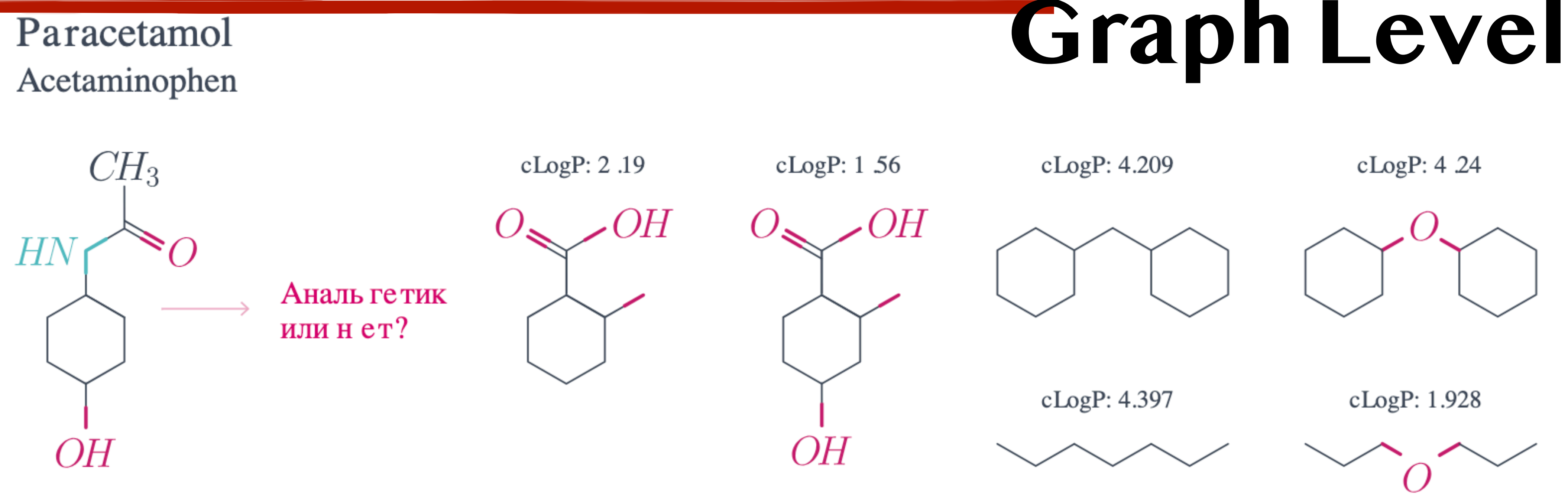
Types of tasks

Node Level



Link Level

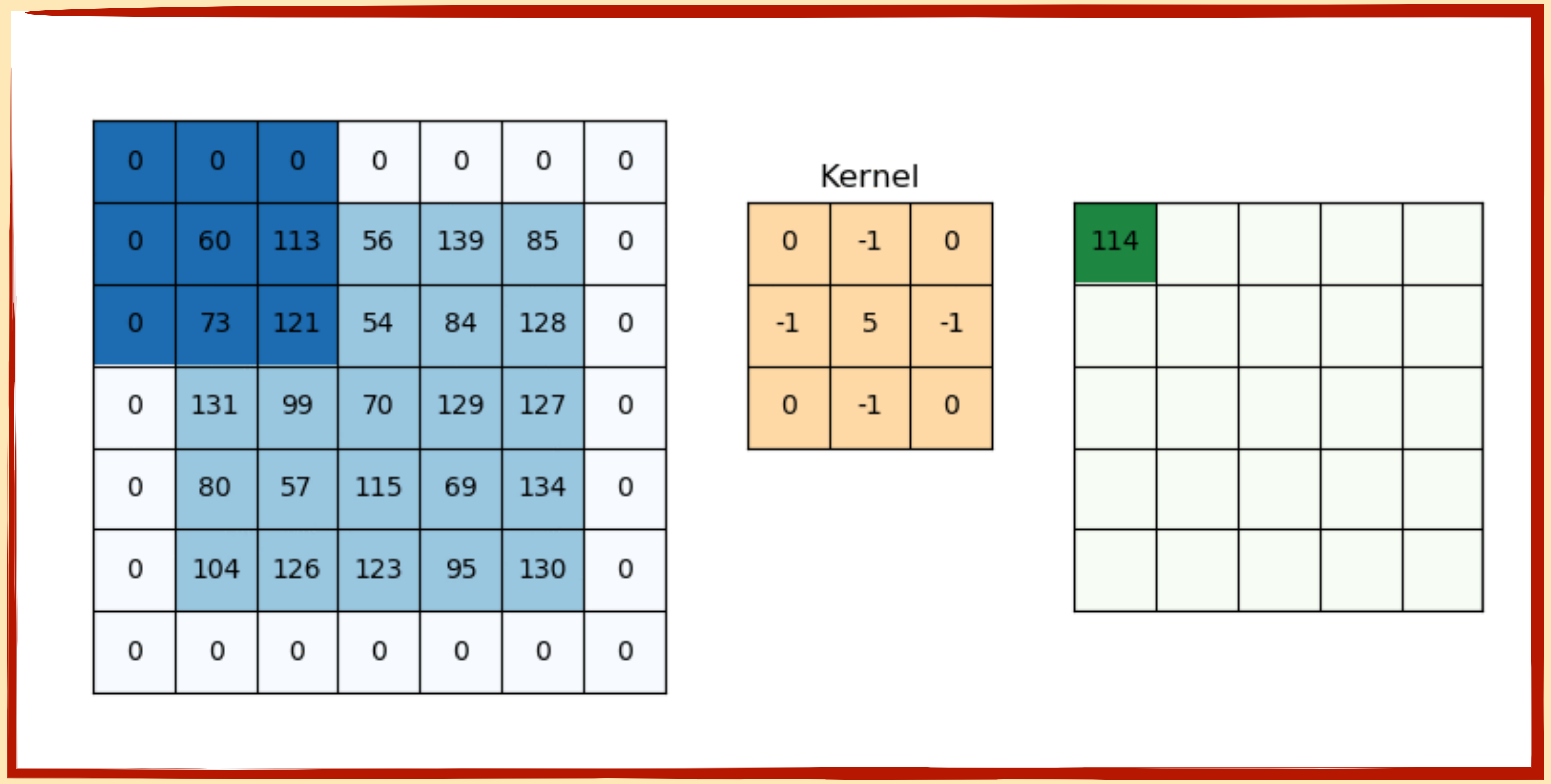
Graph Level



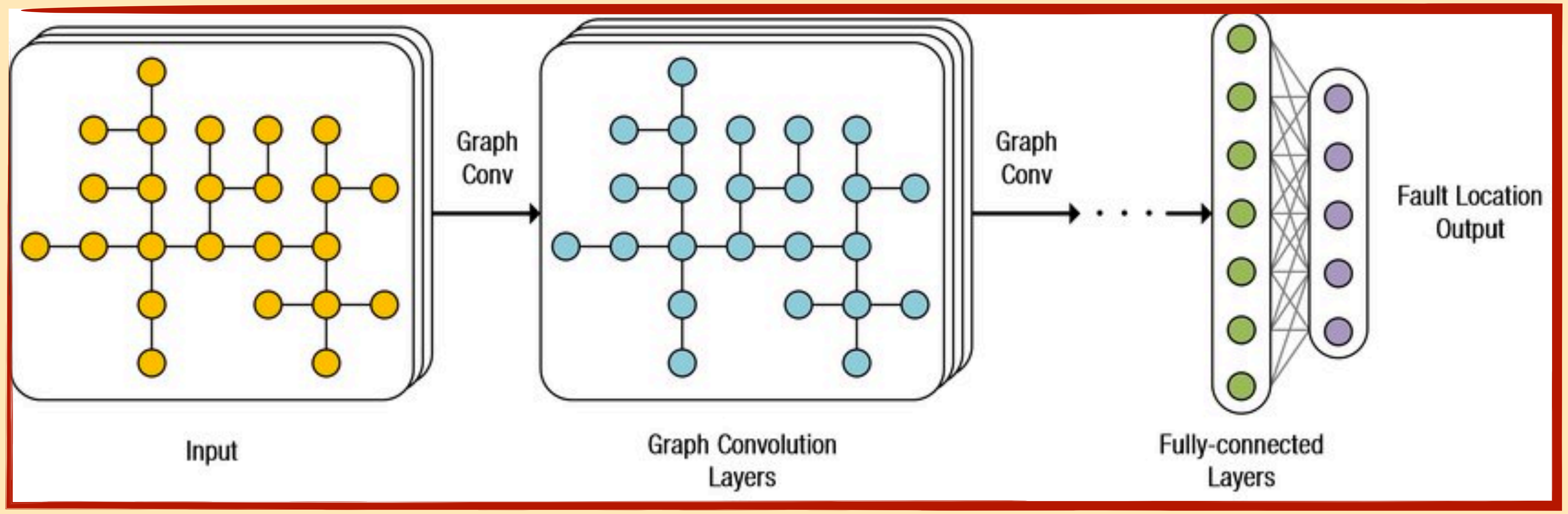
Graphs & ML

Neural Network: Arch

For images



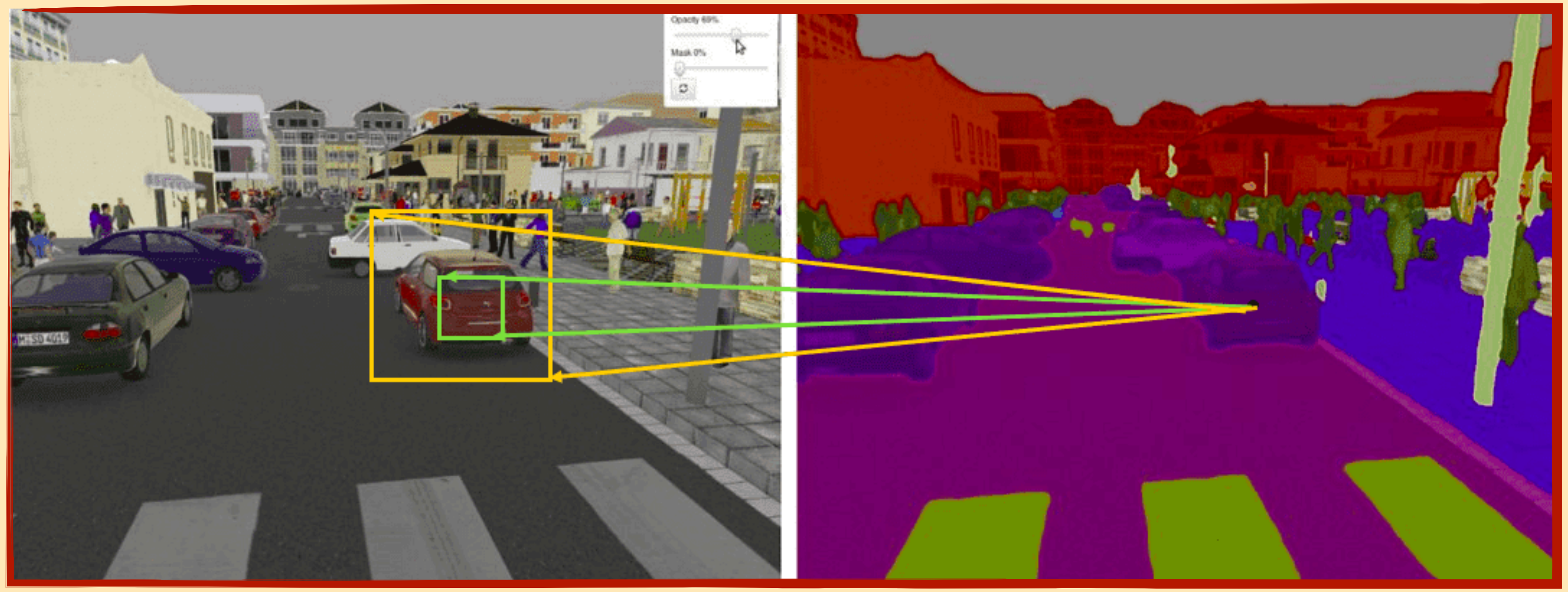
For graphs

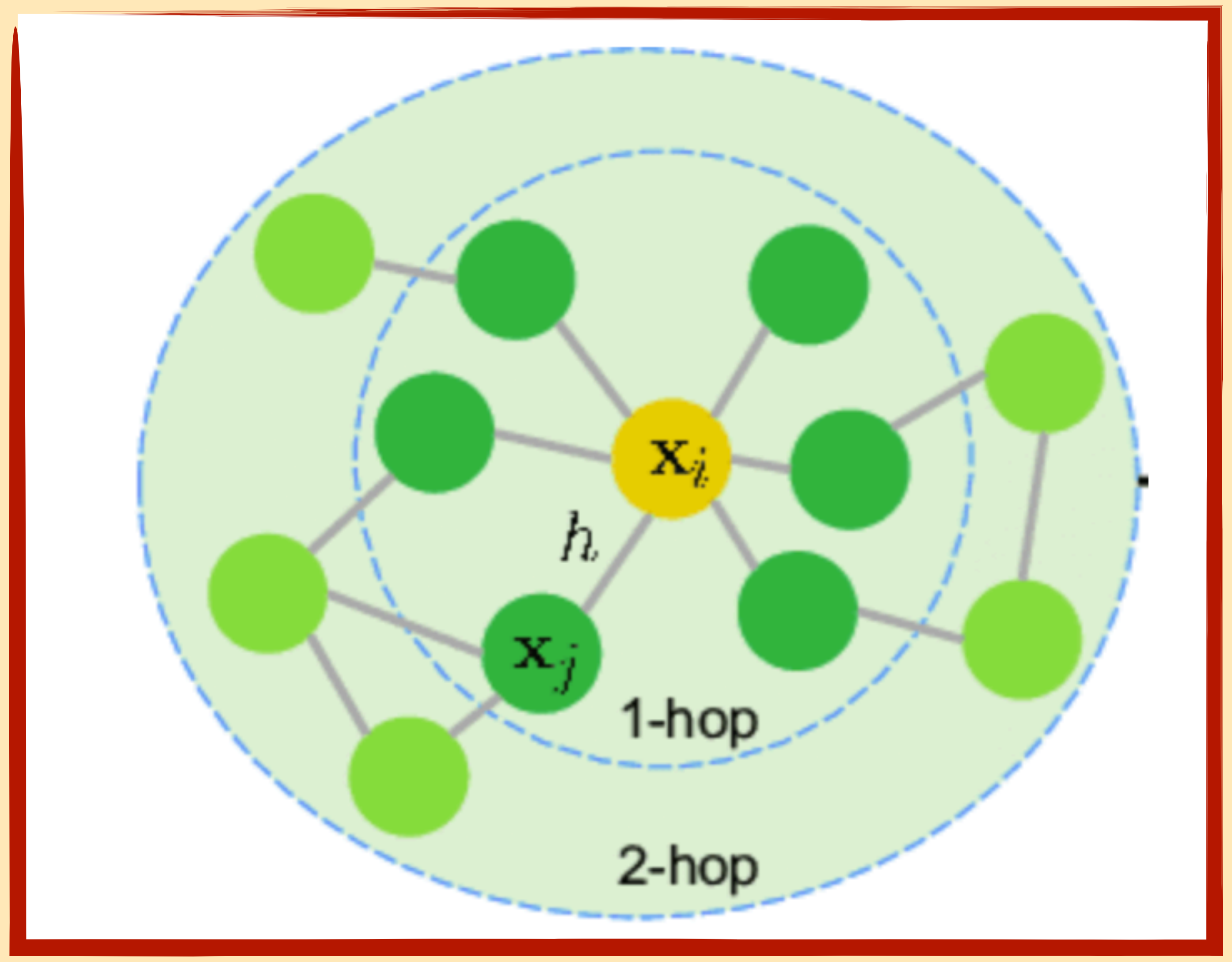


Graphs & ML

Neural Network: receptive field

For images





For graphs