

Welcome to CS152 - Neural Networks!

Spring 2025

Prof. Gabe Hope

Warm-up discussion

- What is an application, program or system that you *at least suspect* uses neural networks in some way?
 - Why is this system useful? Does it have an important benefit for society?
 - What risks does this system have? Is it used maliciously?
- **But first: Introductions!**
 - **Going around:** Introduce yourself with your name, year and major
 - *Optionally:* Is there something you are particularly excited to learn about in this class?

Applications of neural networks

Prediction

Predict output given an input

Image classification

Example: Google lens

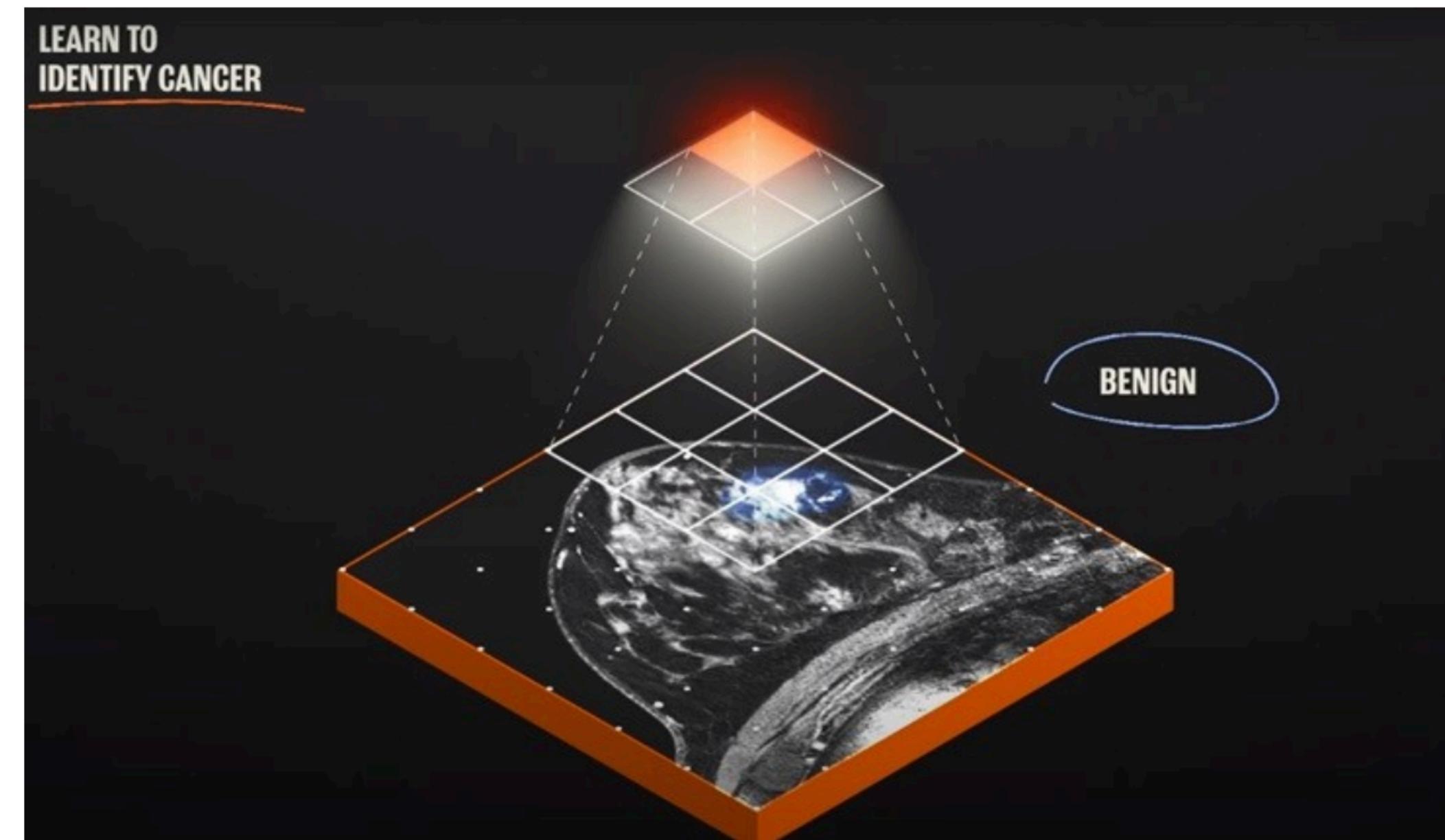


Identify plants and animals

Find out what plant is in your friend's apartment, or what kind of dog you saw in the park.

Medical diagnosis

Example: Breast cancer diagnosis from Microsoft Research



Transforming Breast Cancer Detection
with AI

Weather prediction

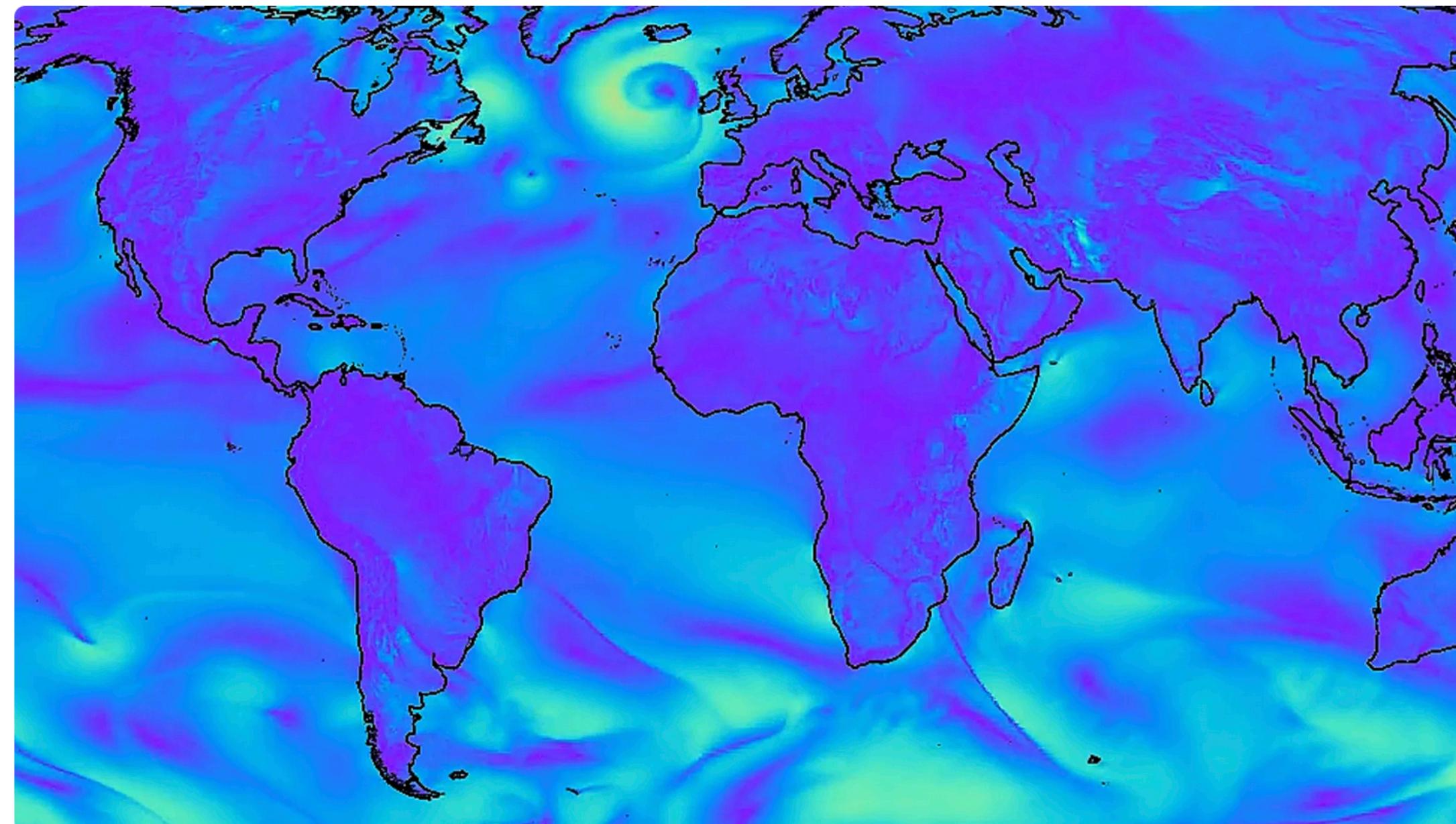
Example: GraphCast weather forecasting

GraphCast: AI model for faster and
more accurate global weather
forecasting

14 NOVEMBER 2023

Remi Lam on behalf of the GraphCast team

Share



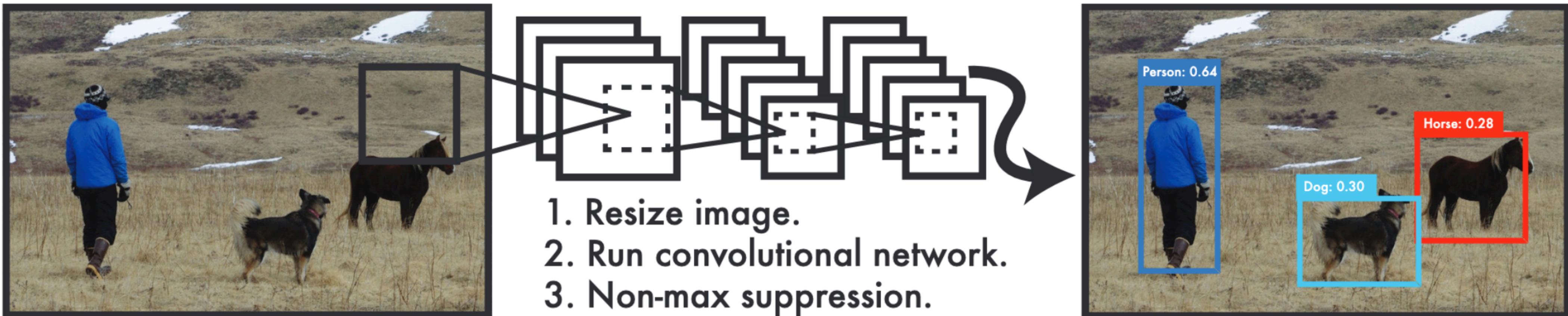
Detection

Find instances of a pattern

Object detection

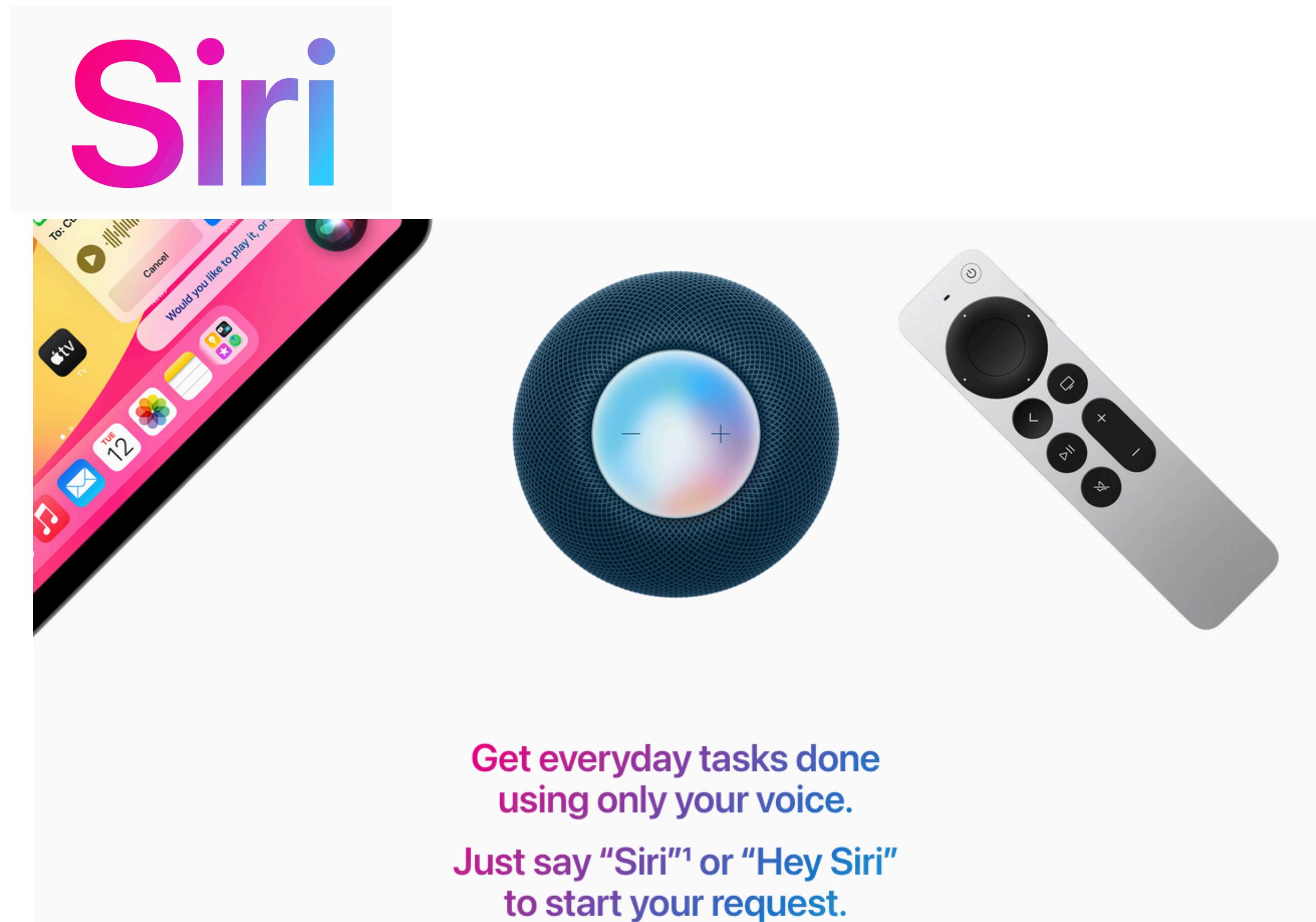
Example: YOLO (You Only Look Once)

You Only Look Once:
Unified, Real-Time Object Detection



Activation phrase recognition

Example: Siri

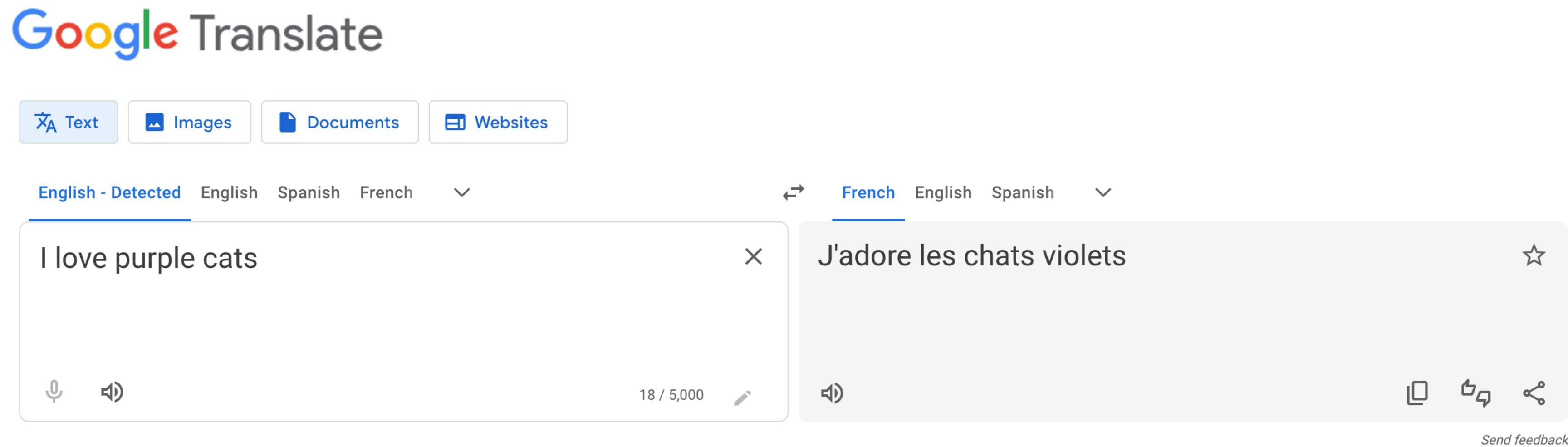


Transformation

Improve or translate an input in some way

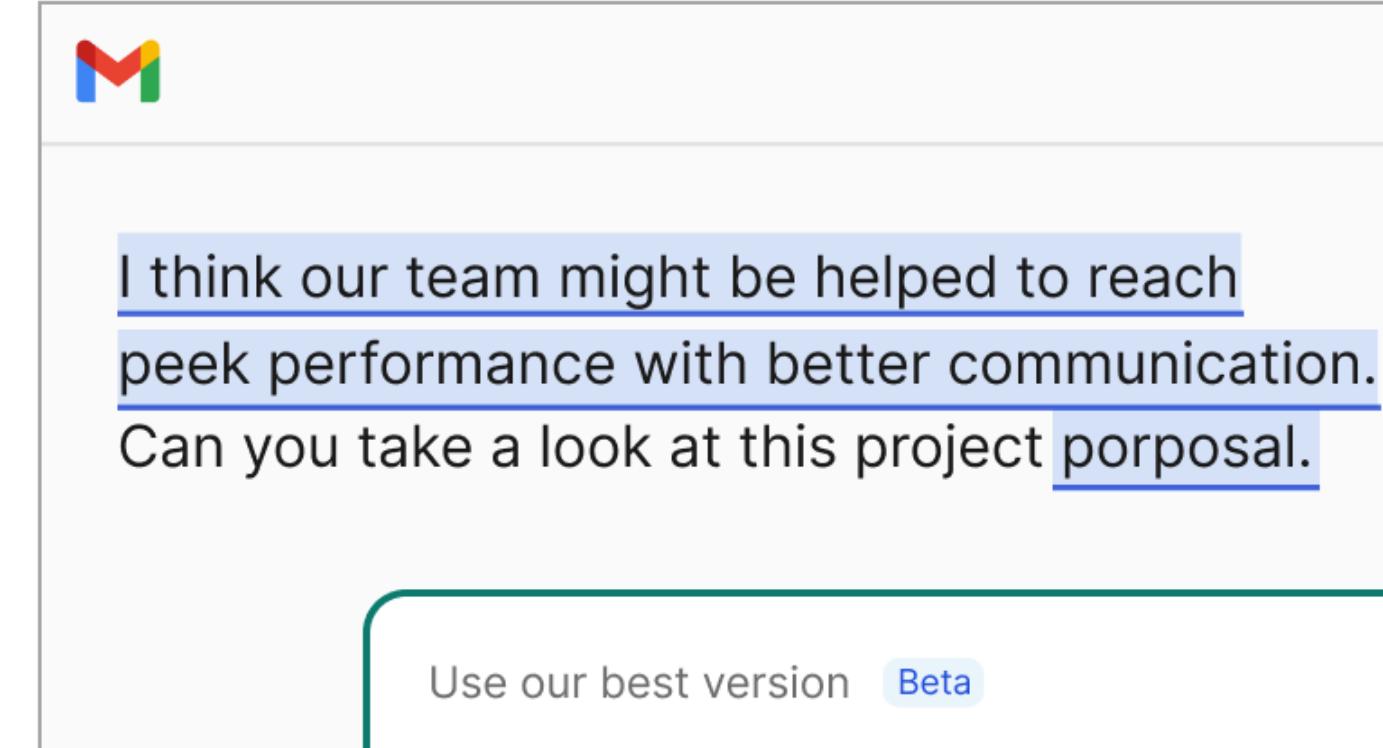
Language translation

Example: Google Translate



Grammar checking

Example: Grammarly



The screenshot shows a Gmail interface with a Grammarly extension overlay. The message body contains the following text:

I think our team might be helped to reach
peak performance with better communication.
Can you take a look at this project porposal.

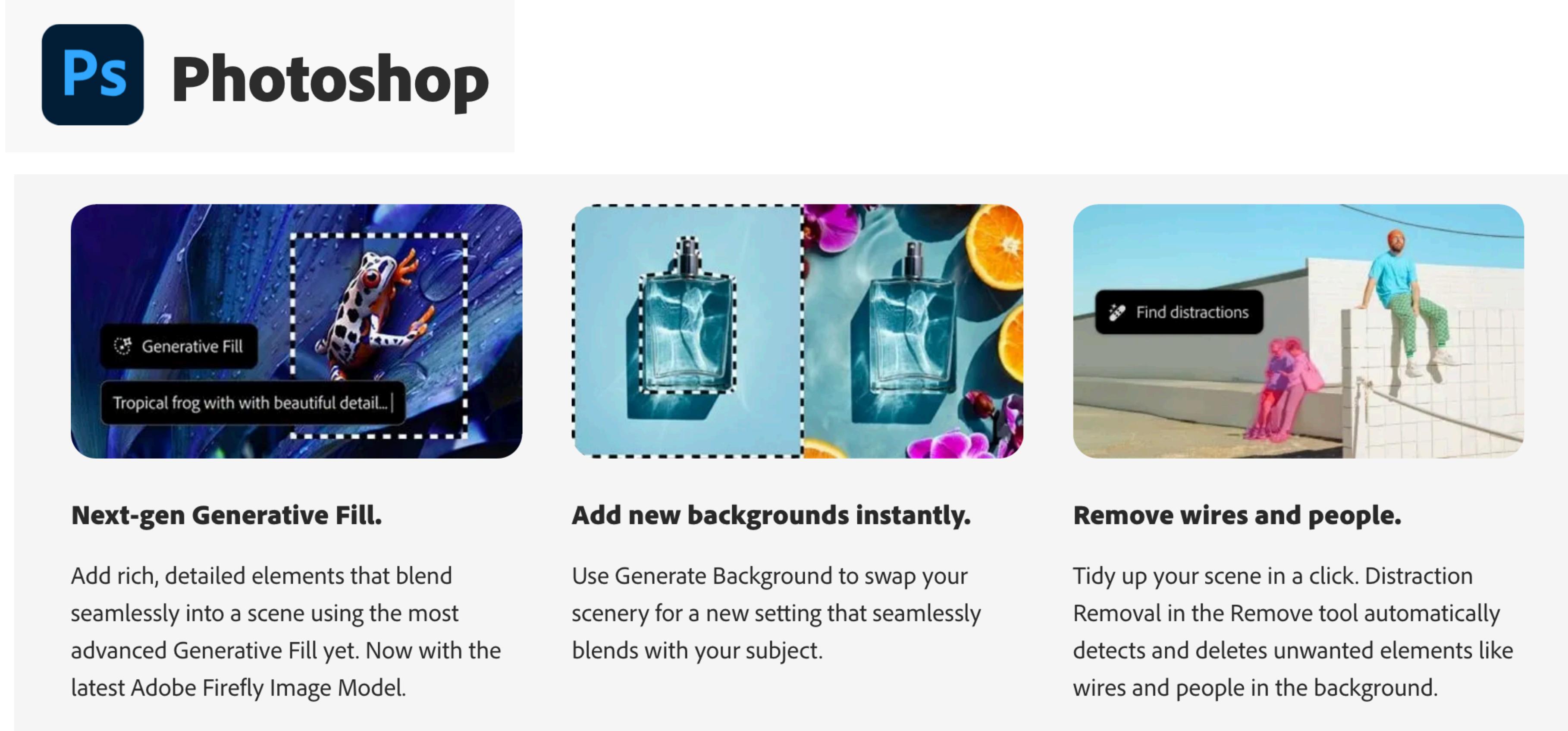
A callout box from Grammarly highlights the word "porposal" with a red underline and suggests the corrected spelling "proposal". The callout also includes a "Beta" label and a "Dismiss" button.

Better communication can help our team reach peak performance. Can you take a look at this project proposal?

[Dismiss](#) [See more in Grammarly](#)

Photo enhancement

Example: AI tools in Photoshop



The screenshot shows the Photoshop interface with the title "Photoshop" and the "Ps" logo. Below the title, there are three examples demonstrating AI tools:

- Generative Fill:** A tropical frog on a blue surface. A dashed selection box highlights the frog, with a tooltip "Generative Fill" and a descriptive text "Tropical frog with with beautiful detail...".
- Generate Background:** Two perfume bottles on a blue background. A dashed selection box highlights the bottles, illustrating how the background can be swapped.
- Remove tool:** A man sitting on a ledge. A tooltip "Find distractions" points to a person sitting on the ground. This demonstrates how the tool automatically detects and removes unwanted elements like wires and people.

Next-gen Generative Fill.
Add rich, detailed elements that blend seamlessly into a scene using the most advanced Generative Fill yet. Now with the latest Adobe Firefly Image Model.

Add new backgrounds instantly.
Use Generate Background to swap your scenery for a new setting that seamlessly blends with your subject.

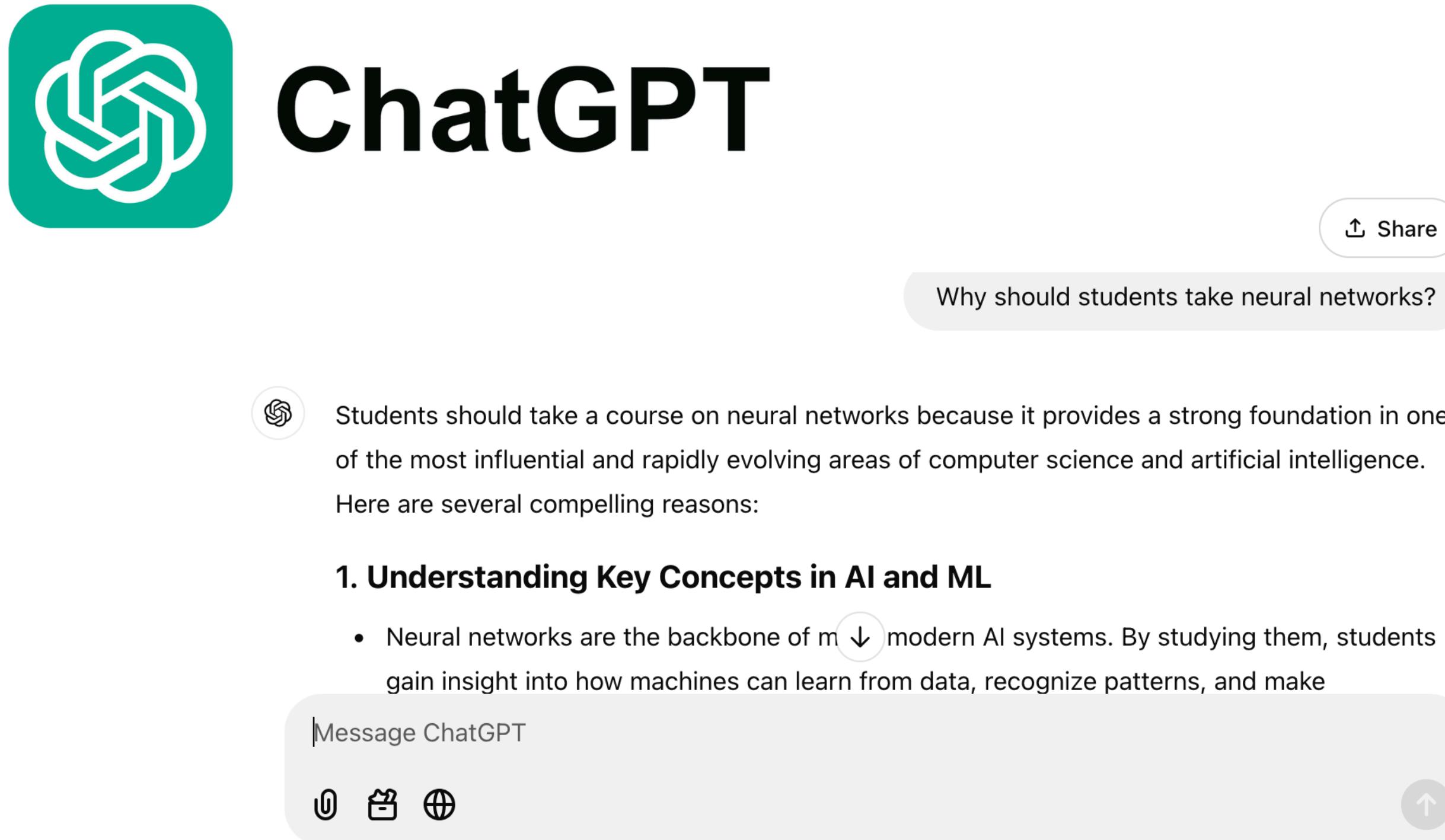
Remove wires and people.
Tidy up your scene in a click. Distraction Removal in the Remove tool automatically detects and deletes unwanted elements like wires and people in the background.

Generation

Create new data

Language models

Example: ChatGPT



The image shows a screenshot of the ChatGPT interface. At the top left is the teal ChatGPT logo. To its right, the word "ChatGPT" is written in a large, bold, black sans-serif font. In the top right corner, there is a "Share" button with a small upward arrow icon. Below the main title, a question is displayed in a light gray rounded rectangle: "Why should students take neural networks?". A small circular icon with a white swirl pattern is positioned to the left of the first paragraph of text. This text explains that neural networks provide a strong foundation in one of the most influential areas of computer science and artificial intelligence, listing several compelling reasons. Below this, a section titled "1. Understanding Key Concepts in AI and ML" is shown, followed by a bulleted list of points. At the bottom of the interface is a message input field containing the placeholder text "Message ChatGPT". To the right of the input field is a send button featuring a small upward arrow icon.

Students should take a course on neural networks because it provides a strong foundation in one of the most influential and rapidly evolving areas of computer science and artificial intelligence. Here are several compelling reasons:

1. Understanding Key Concepts in AI and ML

- Neural networks are the backbone of modern AI systems. By studying them, students gain insight into how machines can learn from data, recognize patterns, and make

Message ChatGPT

Image models

Example: Dall-e 3

DALL-E 3



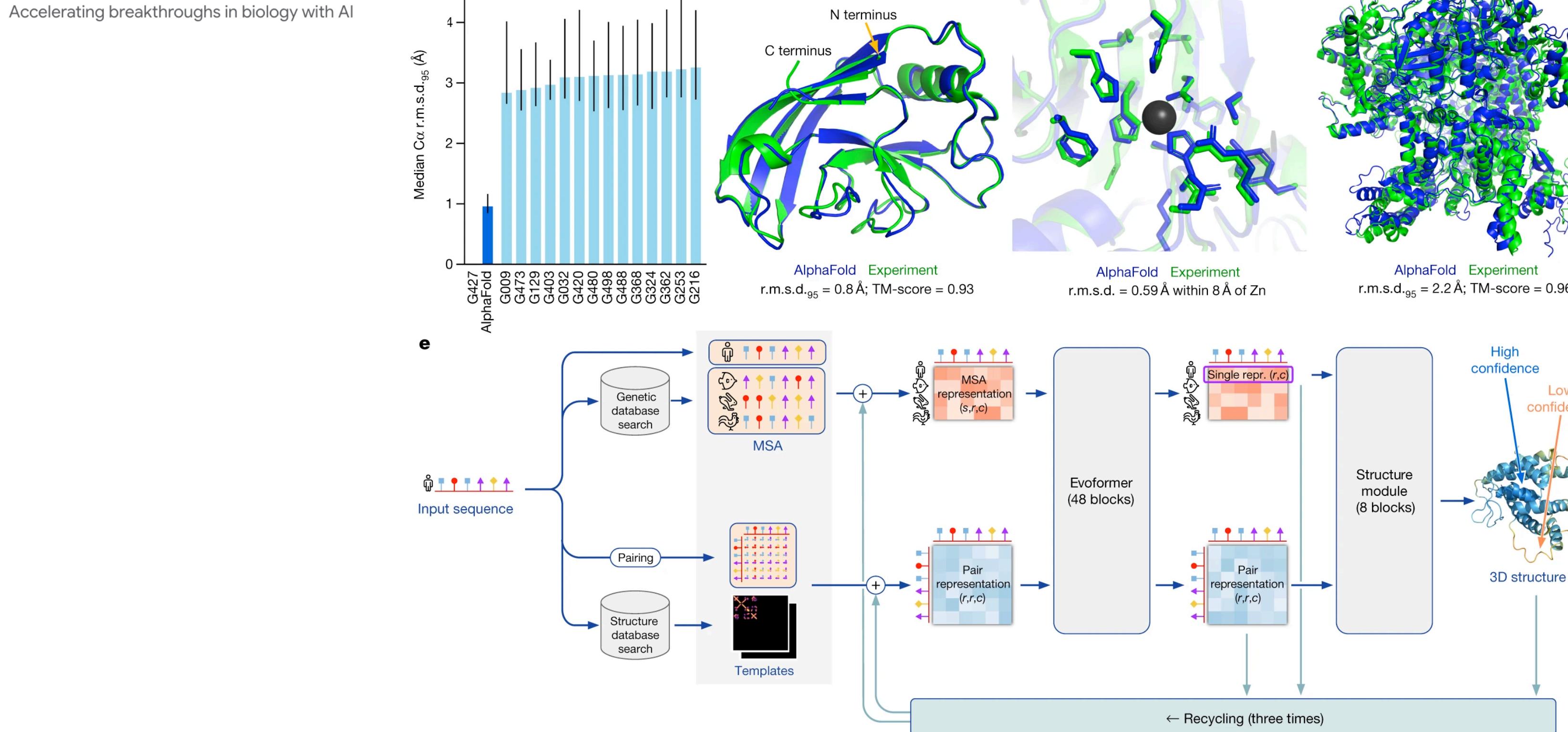
Approximation

Approximate complex physical systems

Molecular dynamics simulation

Example: AlphaFold

AlphaFold

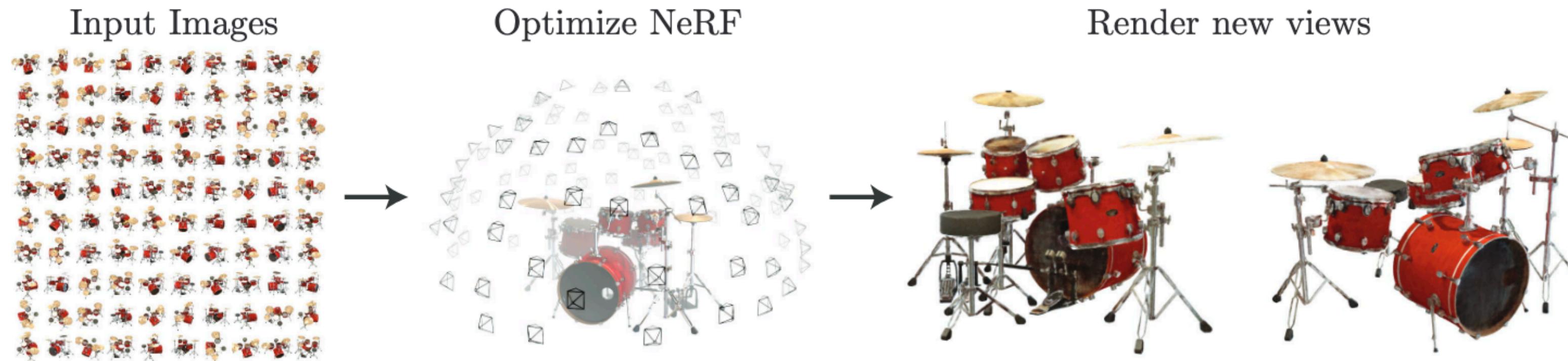


Multi-view simulation

Example: Neural Radiance Fields

NeRF

Representing Scenes as Neural Radiance Fields for View Synthesis



Neural networks have implications

'Impossible' to create AI tools like ChatGPT without copyrighted material, OpenAI says

intelligence firms over the products

Explained: Generative AI's environmental impact
Rapid development and deployment of powerful generative AI models comes with environmental consequences, including increased electricity demand and water consumption.

Adam Zewe | MIT News
January 17, 2025

The Risks of Artificial Intelligence in Weapons Design
Researchers outline dangers of developing AI-powered autonomous weapons

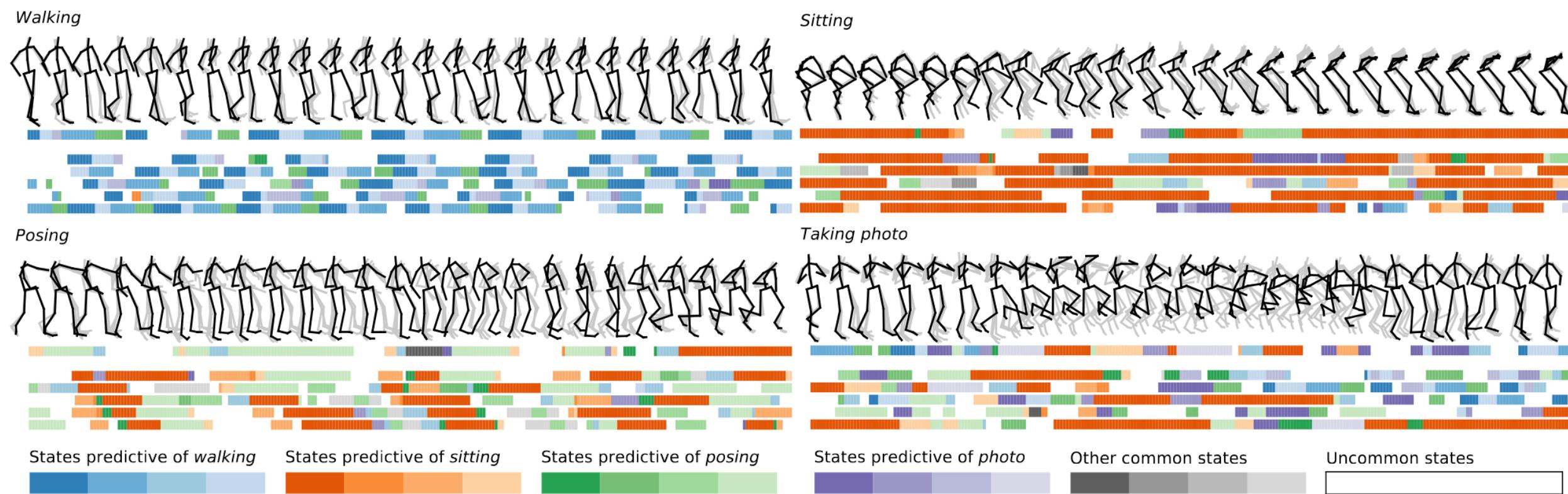
By CATHERINE CARUSO | August 7, 2024 | Research
6 min read

Machine Bias

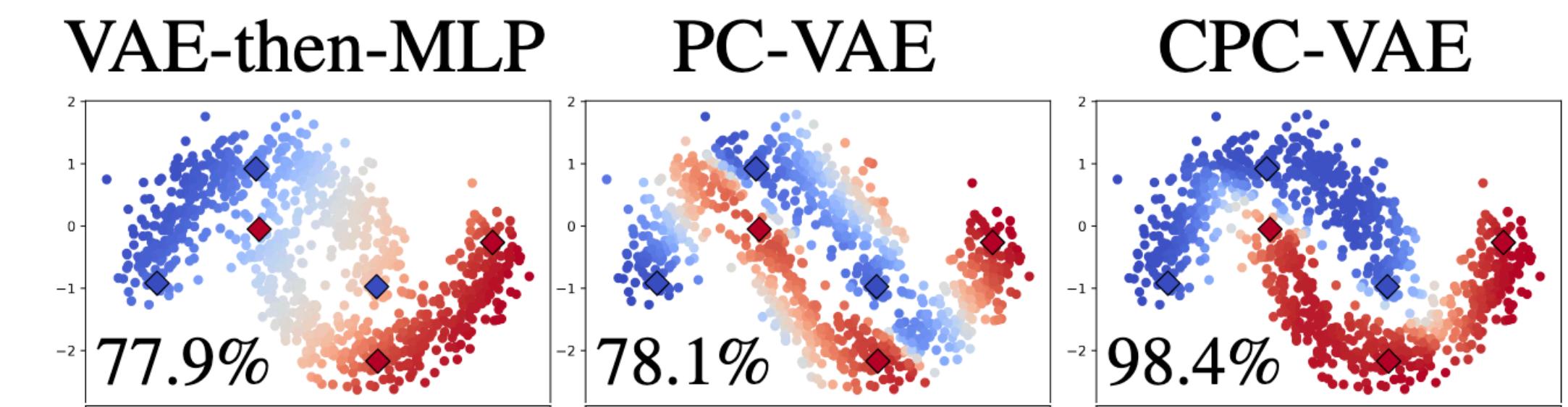
There's software used across the country to predict future criminals. And it's biased against blacks.

My work

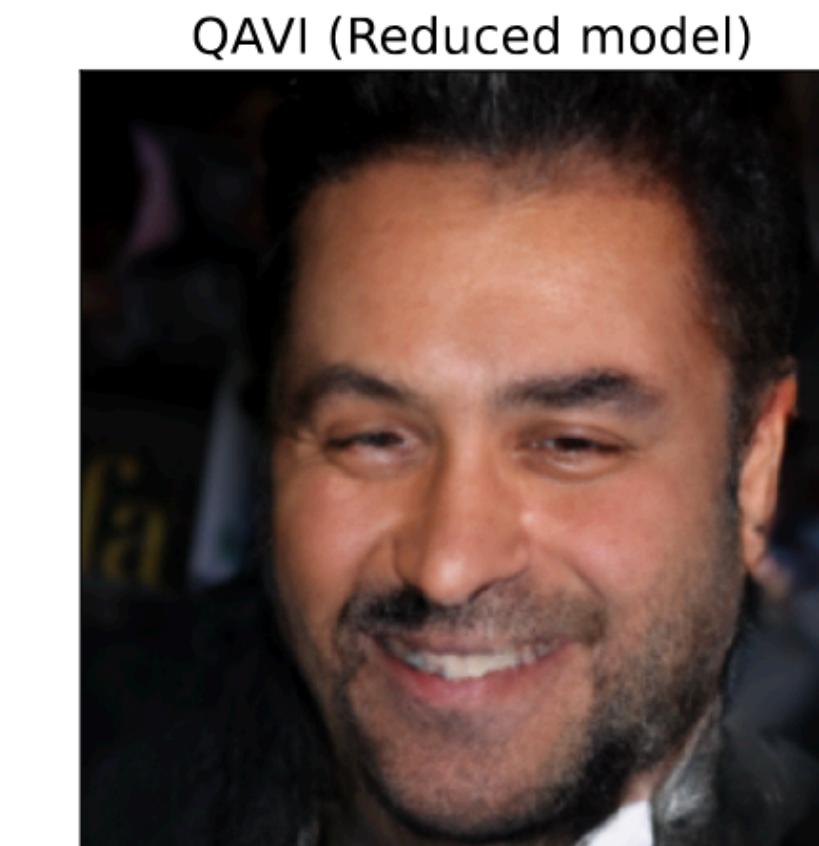
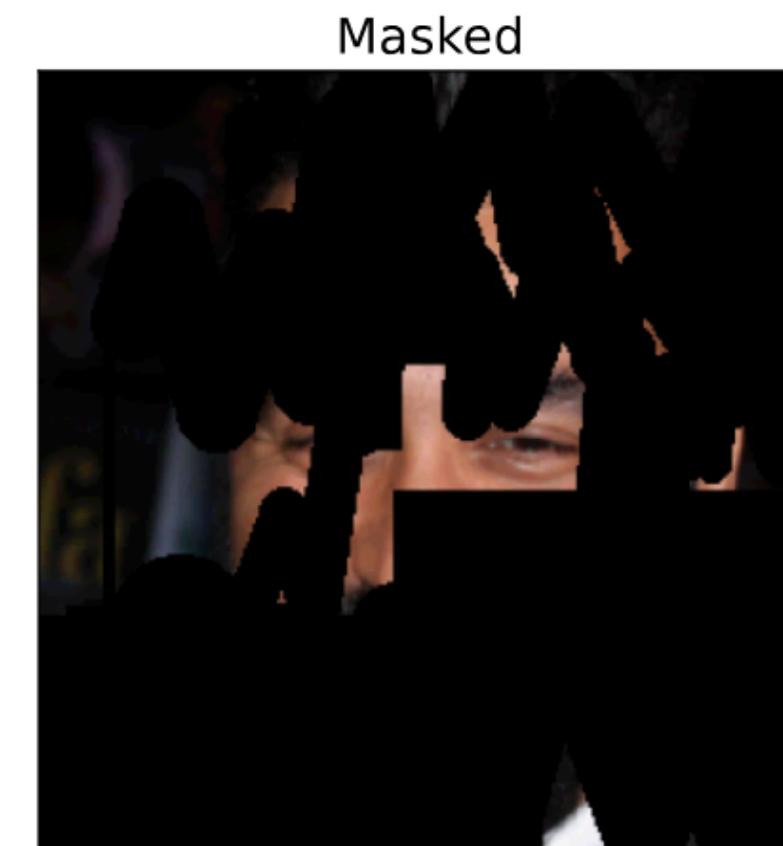
Timeseries analysis



Semi-supervised learning



Imputation

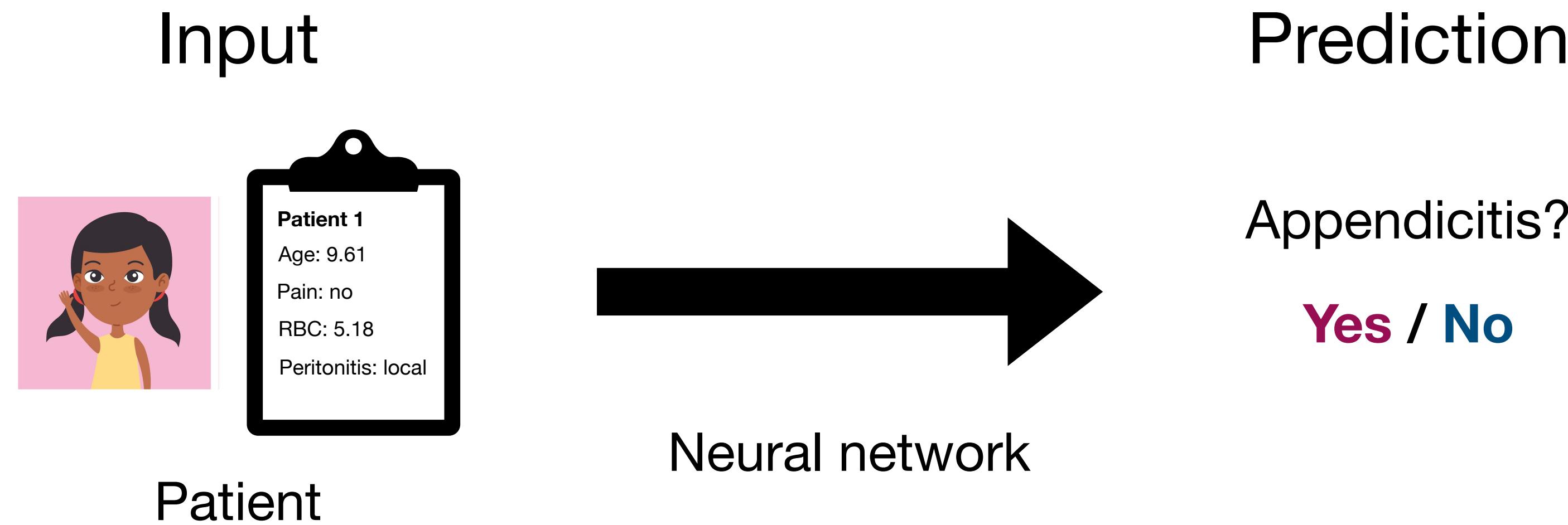


Course Logistics (website)

Prediction problems

Setup

Example: Appendicitis diagnosis



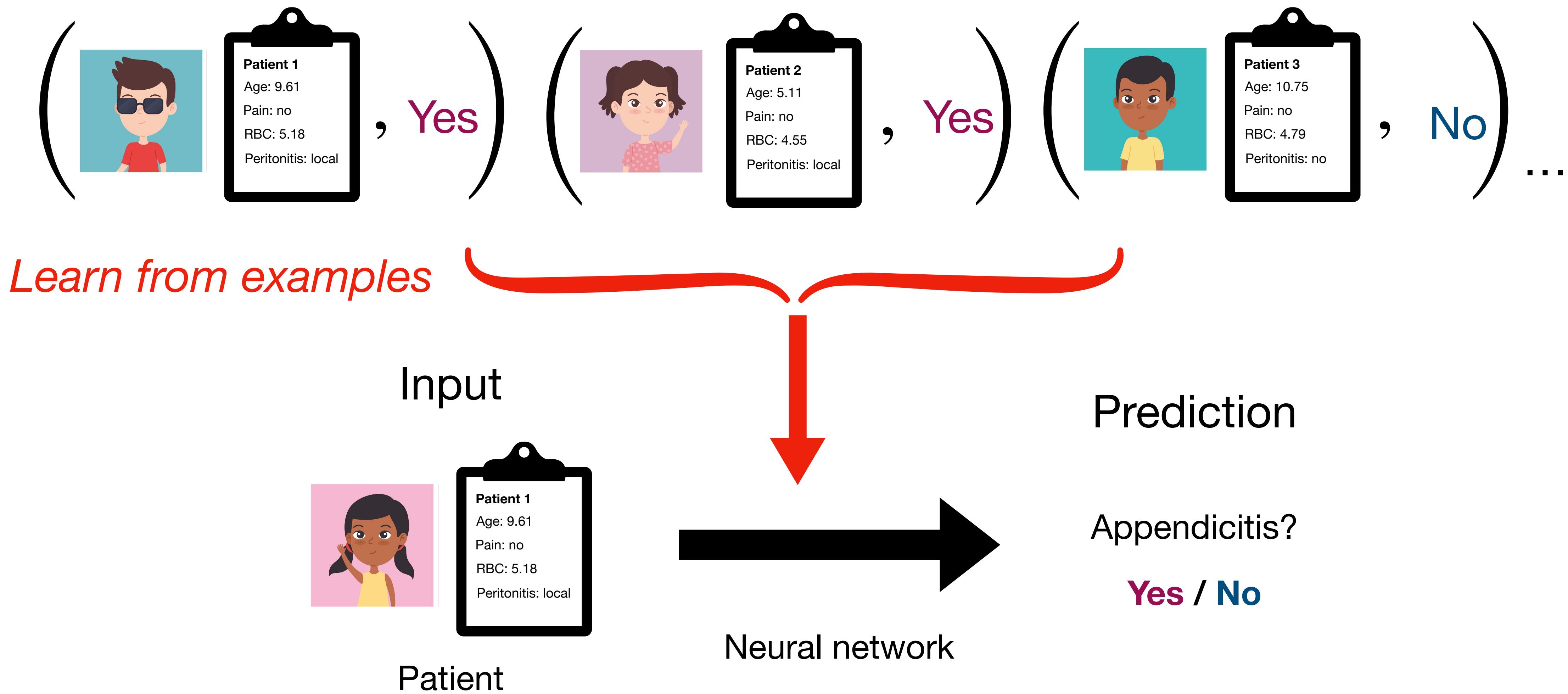
Some notation:

Input: \mathbf{x} \longrightarrow Output: y , $y = f(\mathbf{x})$

predict *Prediction function*

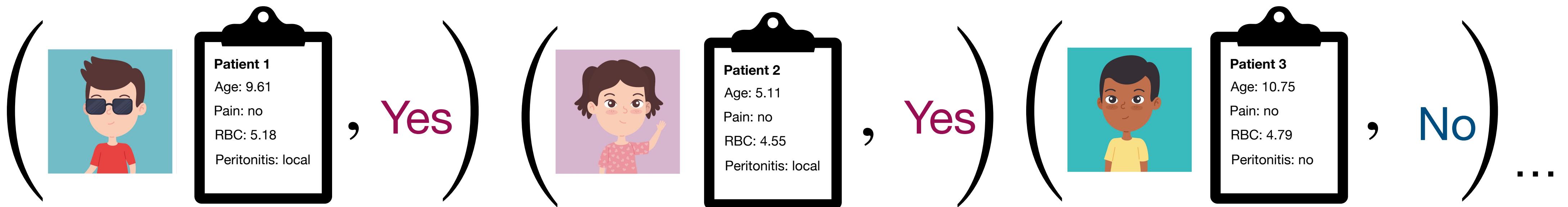
Preview: Machine learning

How do we find a good prediction function?



Dataset

Set of known inputs and outputs



Some notation:

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_N, y_N)\}$$

Types of data

Tabular

Records: e.g. Excel, SQL, Pandas

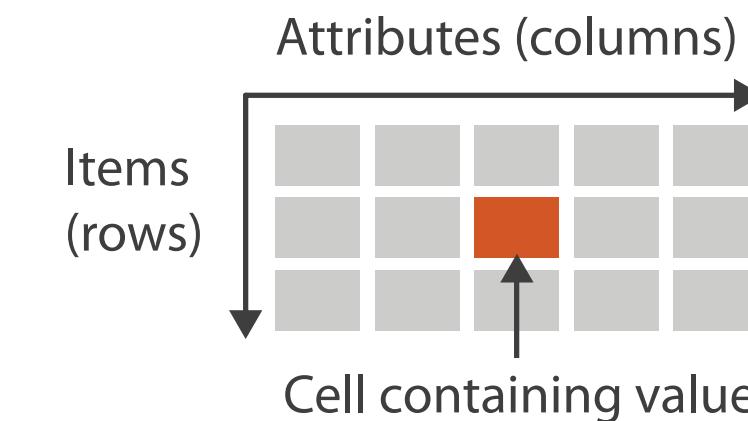
Patients with abdominal pain						
	Age	Appendix Size	Migratory Pain	RBC Count	RBC Urine	Peritonitis
0	9.61	9.0	no	5.18	high	local
1	5.11	7.0	no	4.55	medium	local
2	10.75	5.0	no	4.79	none	no
3	10.51	9.0	no	5.03	none	local
4	7.3	6.2	yes	4.64	low	no
5	15.21	8.5	yes	4.62	low	no
6	15.83	12.0	yes	4.33	high	no
7	9.58	7.0	yes	5.04	low	generalized
8	10.37	5.5	no	4.8	none	no
9	16.66	9.0	yes	5.31	none	no
10	14.52	4.5	yes	4.9	none	no
11	10.74	9.0	no	5.66	none	local
12	12.41	3.7	no	5.49	none	no
13	6.67	3.5	no	5.27	none	no
14	14.36	9.0	yes	4.84	low	local
15	9.04	5.3	yes	4.92	low	no
16	12.43	12.0	yes	4.62	none	generalized

→ Data and Dataset Types

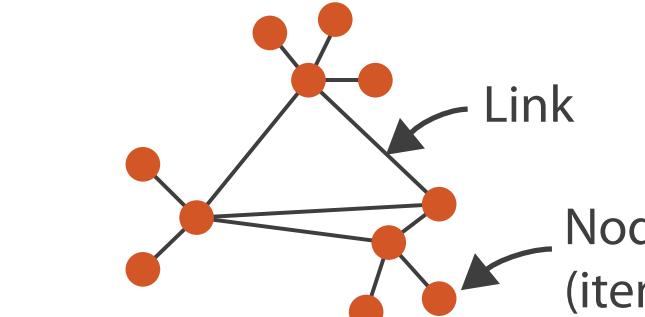
Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Clusters, Sets, Lists
Attributes	Links	Positions	Positions	Items

→ Dataset Types

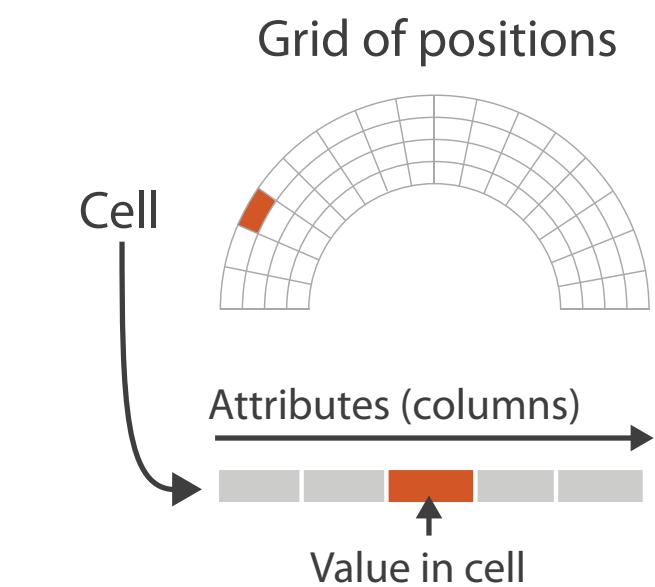
→ Tables



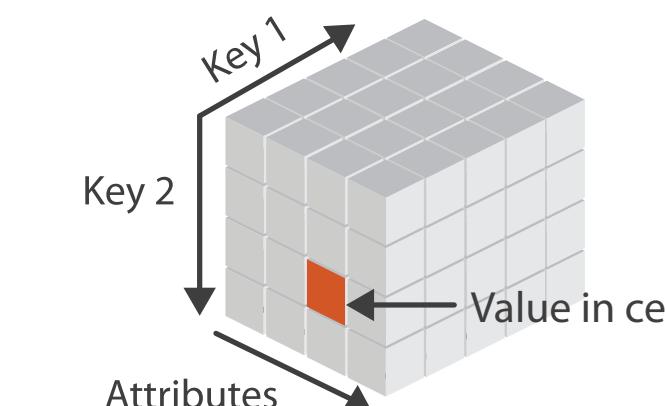
→ Networks



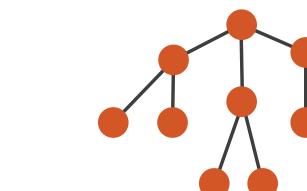
→ Fields (Continuous)



→ Multidimensional Table



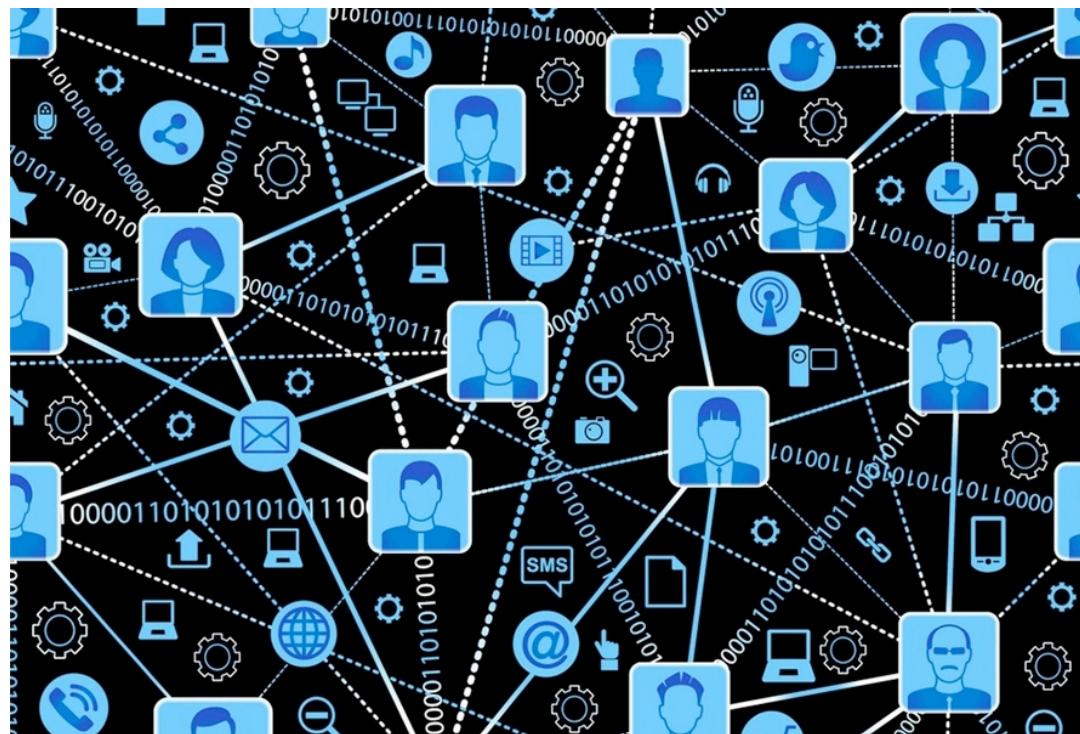
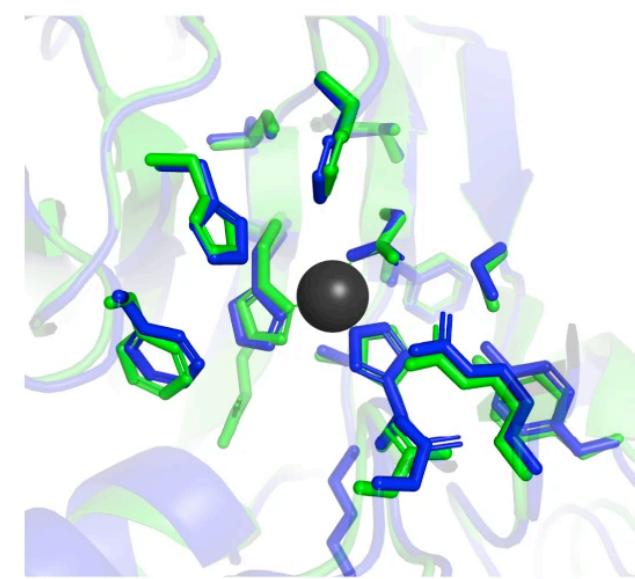
→ Trees



Credit: Tamara Munzer

Types of data Networks

Molecules, social networks, etc.

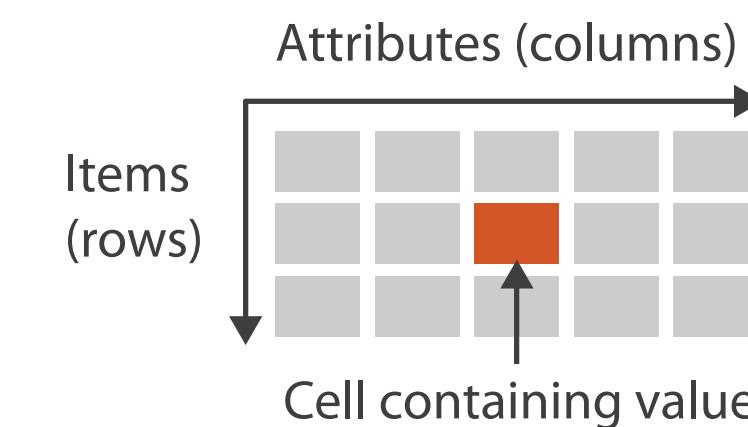


→ Data and Dataset Types

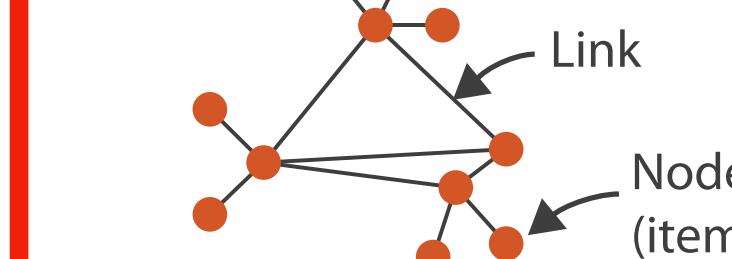
Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Clusters, Sets, Lists
Attributes	Links	Positions	Positions	Items

→ Dataset Types

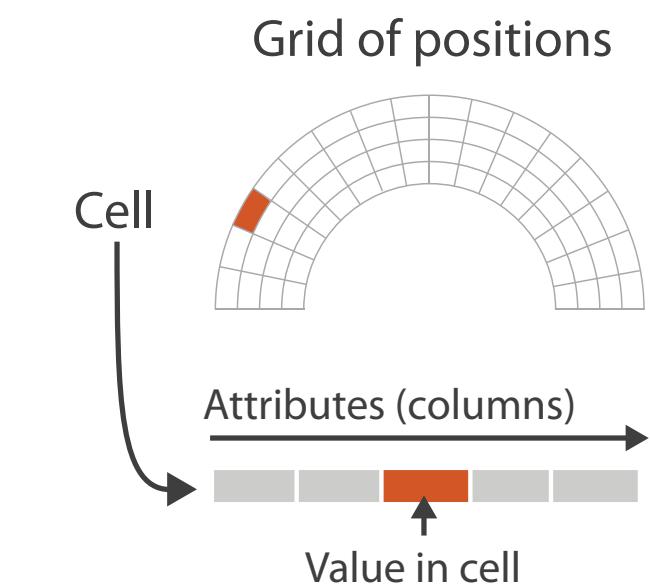
→ Tables



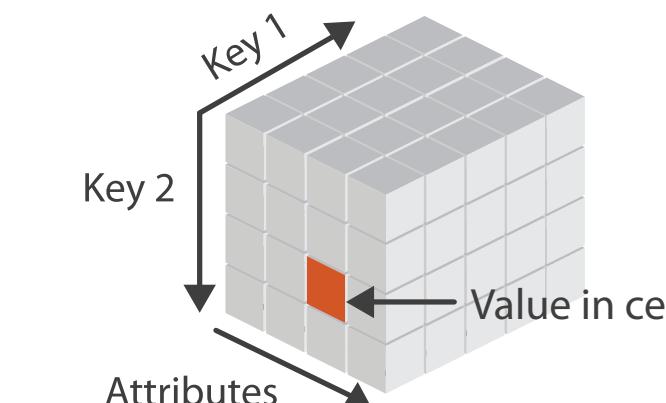
→ Networks



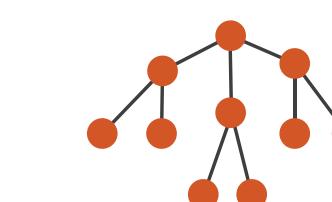
→ Fields (Continuous)



→ Multidimensional Table



→ Trees



Credit: Tamara Munzer

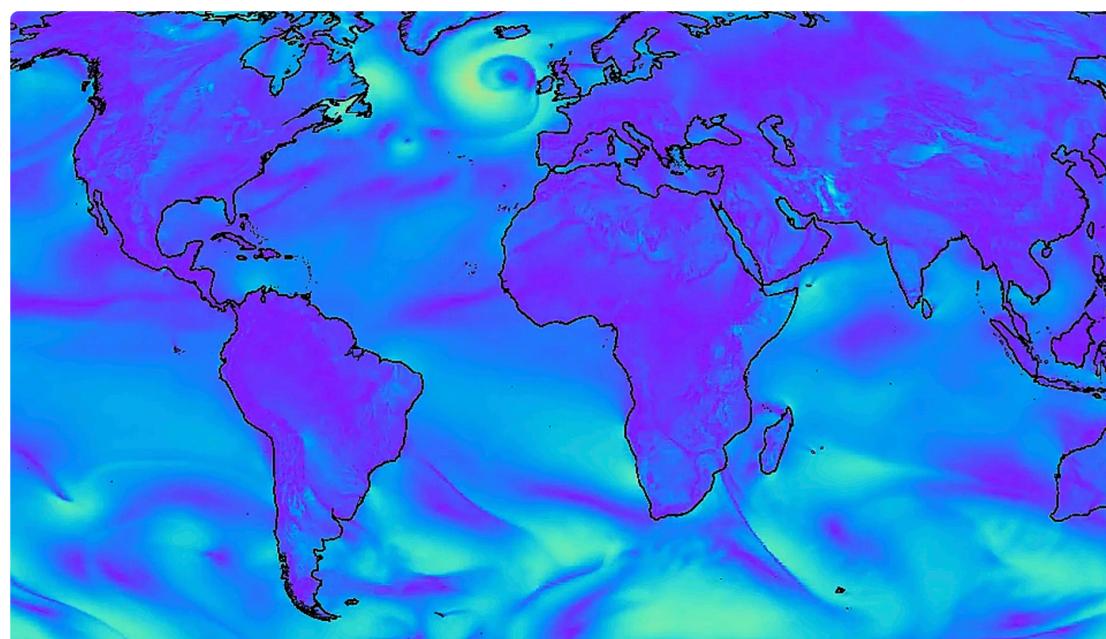
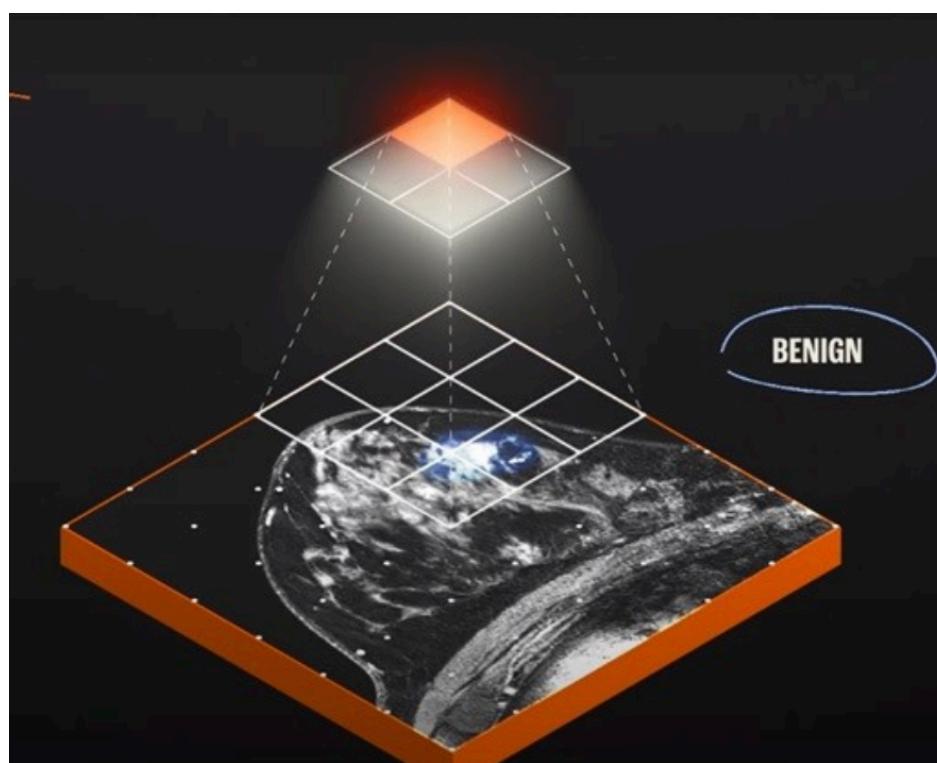
Types of data

Fields

Images, audio, medical scans, etc.

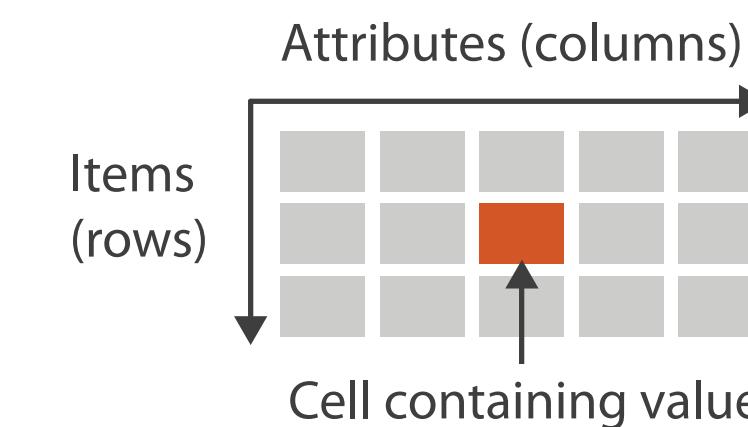
→ Data and Dataset Types

Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Clusters, Sets, Lists
Attributes	Links	Positions	Positions	Items

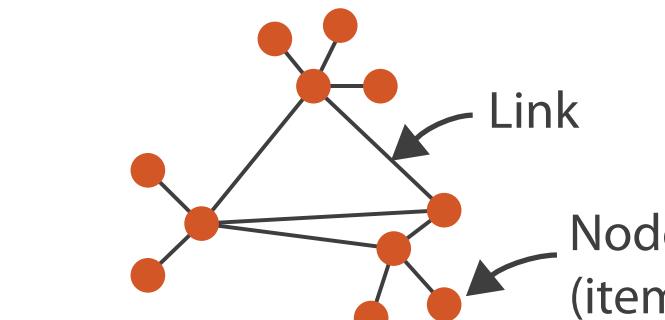


→ Dataset Types

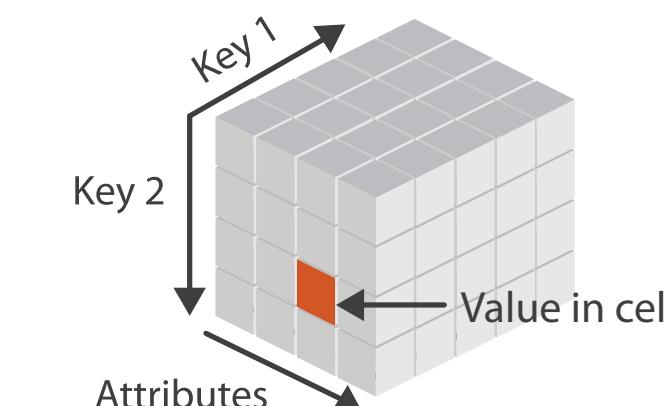
→ Tables



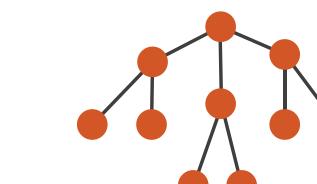
→ Networks



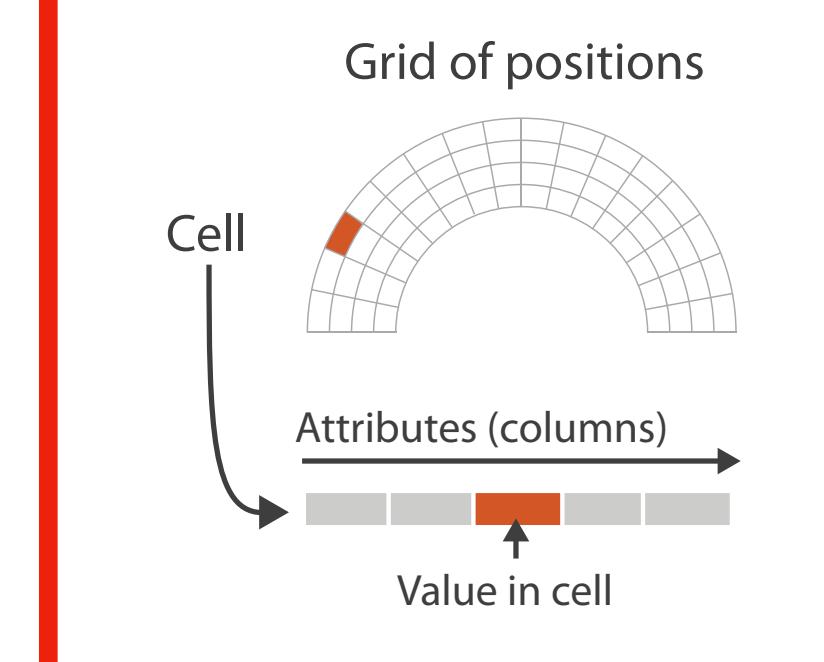
→ Multidimensional Table



→ Trees



→ Fields (Continuous)



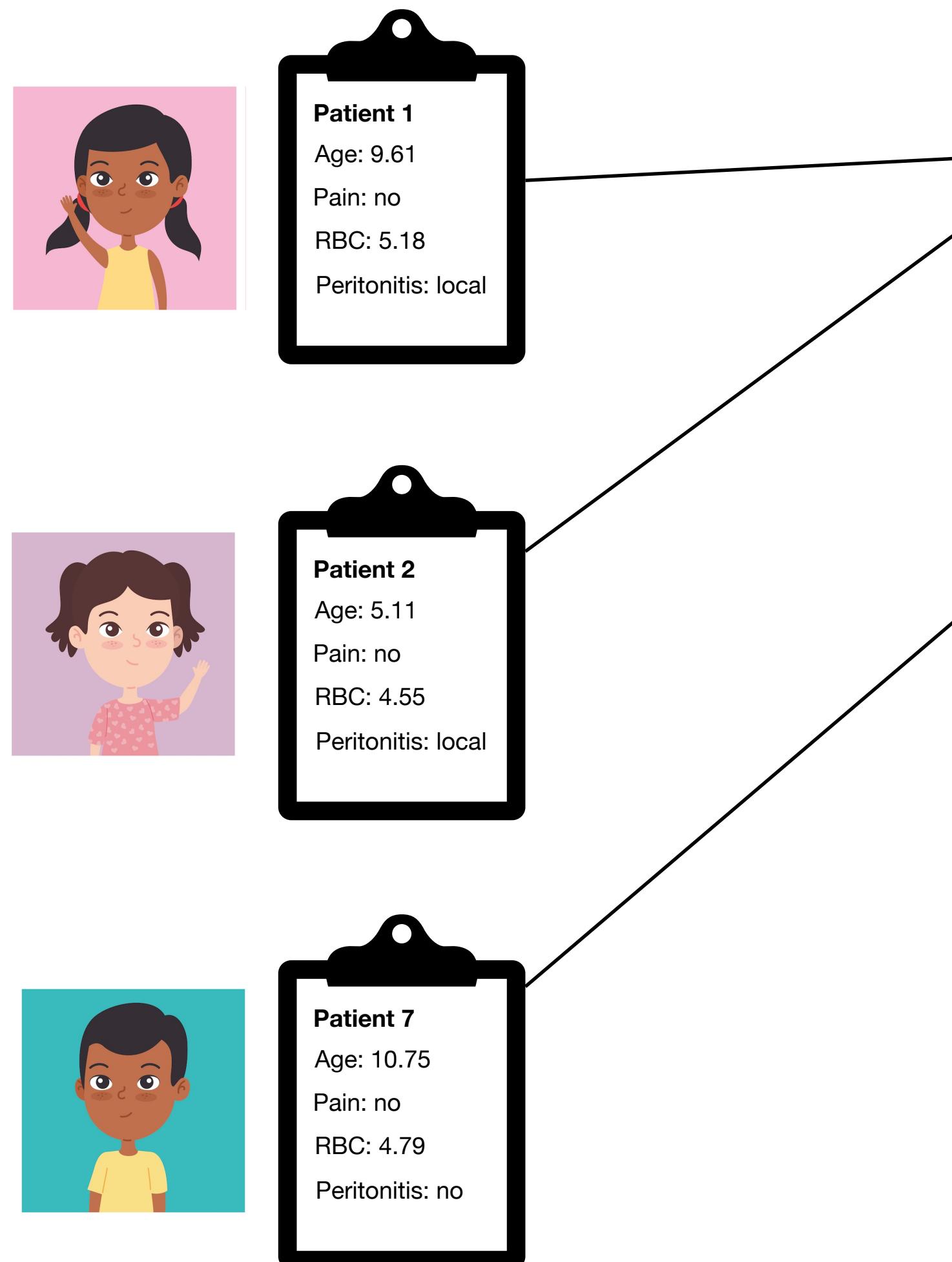
Credit: Tamara Munzer

Tabular data

Patients with abdominal pain

	Age	Appendix Size	Migratory Pain	RBC Count	RBC Urine	Peritonitis
0	9.61	9.0	no	5.18	high	local
1	5.11	7.0	no	4.55	medium	local
2	10.75	5.0	no	4.79	none	no
3	10.51	9.0	no	5.03	none	local
4	7.3	6.2	yes	4.64	low	no
5	15.21	8.5	yes	4.62	low	no
6	15.83	12.0	yes	4.33	high	no
7	9.58	7.0	yes	5.04	low	generalized
8	10.37	5.5	no	4.8	none	no
9	16.66	9.0	yes	5.31	none	no
10	14.52	4.5	yes	4.9	none	no
11	10.74	9.0	no	5.66	none	local
12	12.41	3.7	no	5.49	none	no
13	6.67	3.5	no	5.27	none	no
14	14.36	9.0	yes	4.84	low	local
15	9.04	5.3	yes	4.92	low	no
16	12.43	12.0	yes	4.62	none	generalized

Tabular data



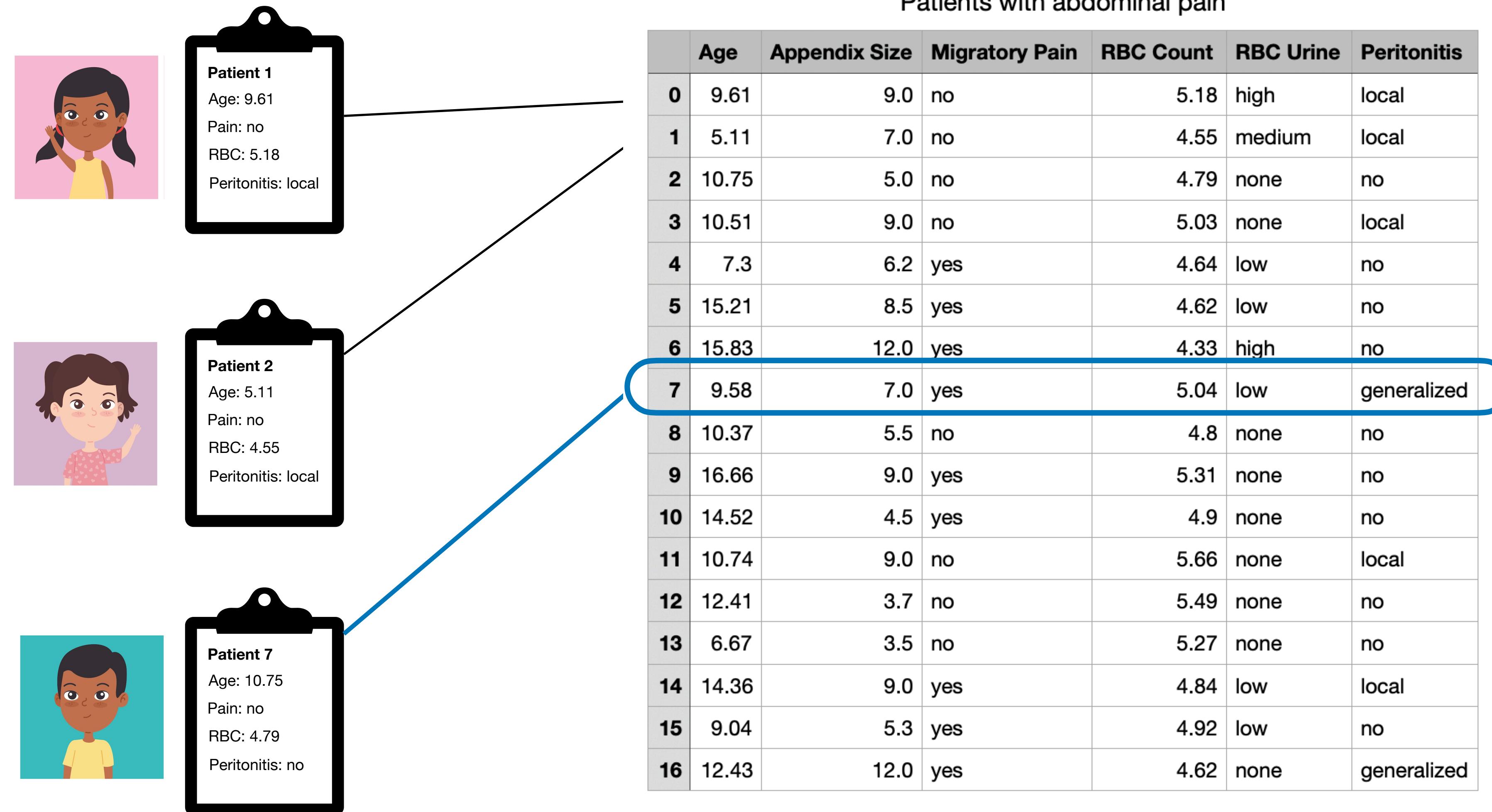
Patients with abdominal pain

	Age	Appendix Size	Migratory Pain	RBC Count	RBC Urine	Peritonitis
0	9.61	9.0	no	5.18	high	local
1	5.11	7.0	no	4.55	medium	local
2	10.75	5.0	no	4.79	none	no
3	10.51	9.0	no	5.03	none	local
4	7.3	6.2	yes	4.64	low	no
5	15.21	8.5	yes	4.62	low	no
6	15.83	12.0	yes	4.33	high	no
7	9.58	7.0	yes	5.04	low	generalized
8	10.37	5.5	no	4.8	none	no
9	16.66	9.0	yes	5.31	none	no
10	14.52	4.5	yes	4.9	none	no
11	10.74	9.0	no	5.66	none	local
12	12.41	3.7	no	5.49	none	no
13	6.67	3.5	no	5.27	none	no
14	14.36	9.0	yes	4.84	low	local
15	9.04	5.3	yes	4.92	low	no
16	12.43	12.0	yes	4.62	none	generalized

Patients

Table

Tabular data

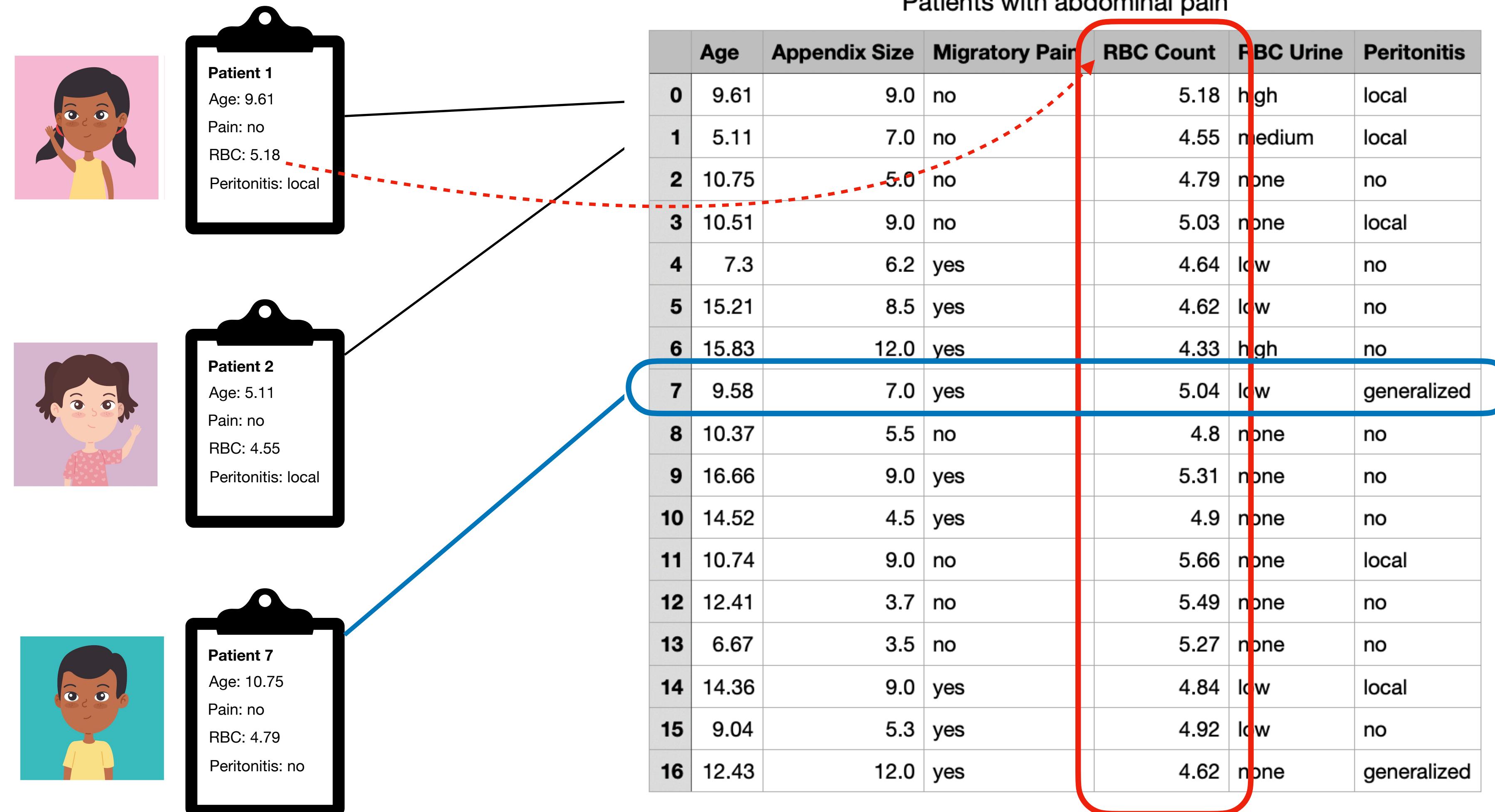


Patients

Table

Observation

Tabular data



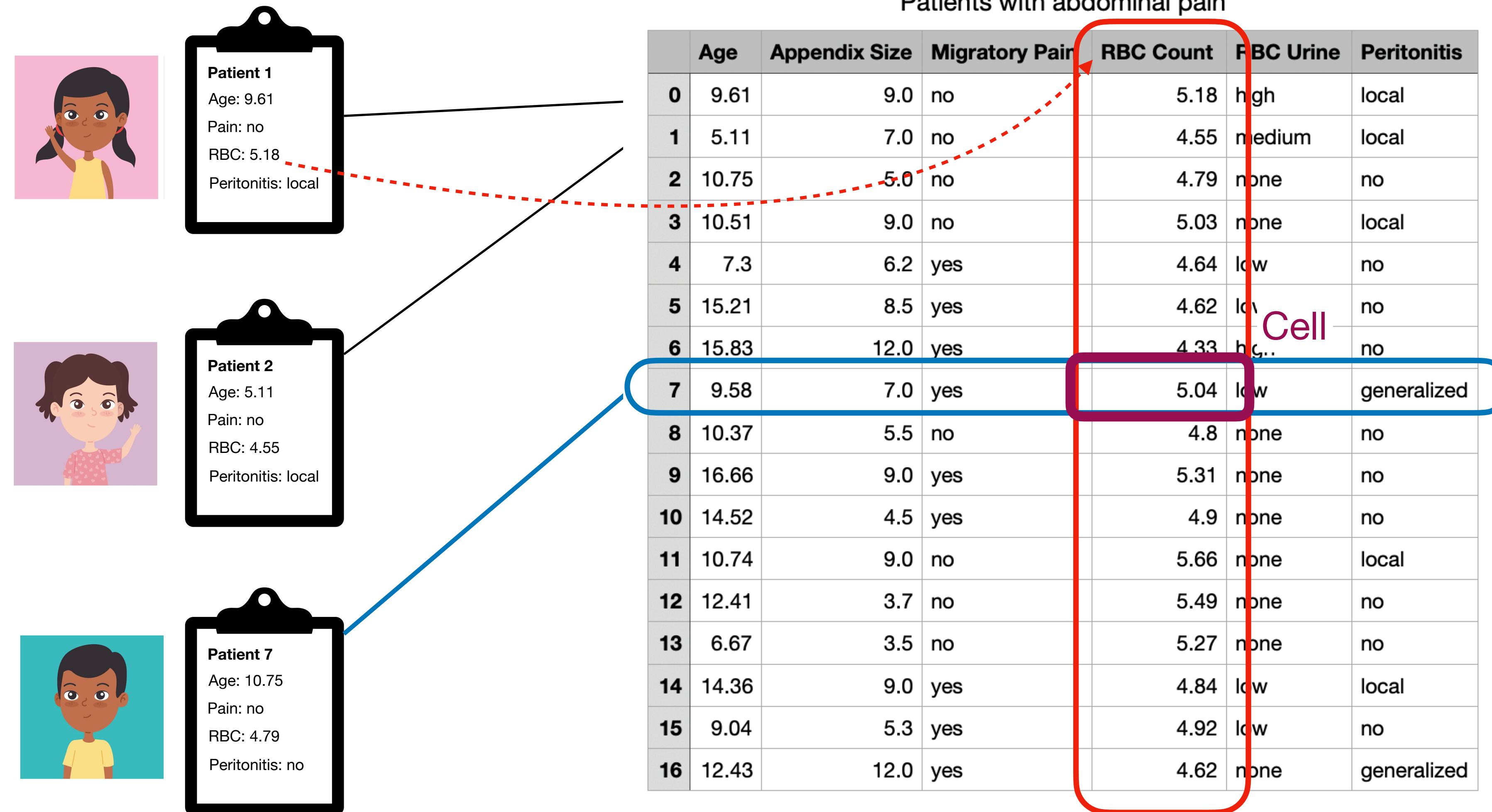
Patients

Table

Observation

Feature

Tabular data



Patients

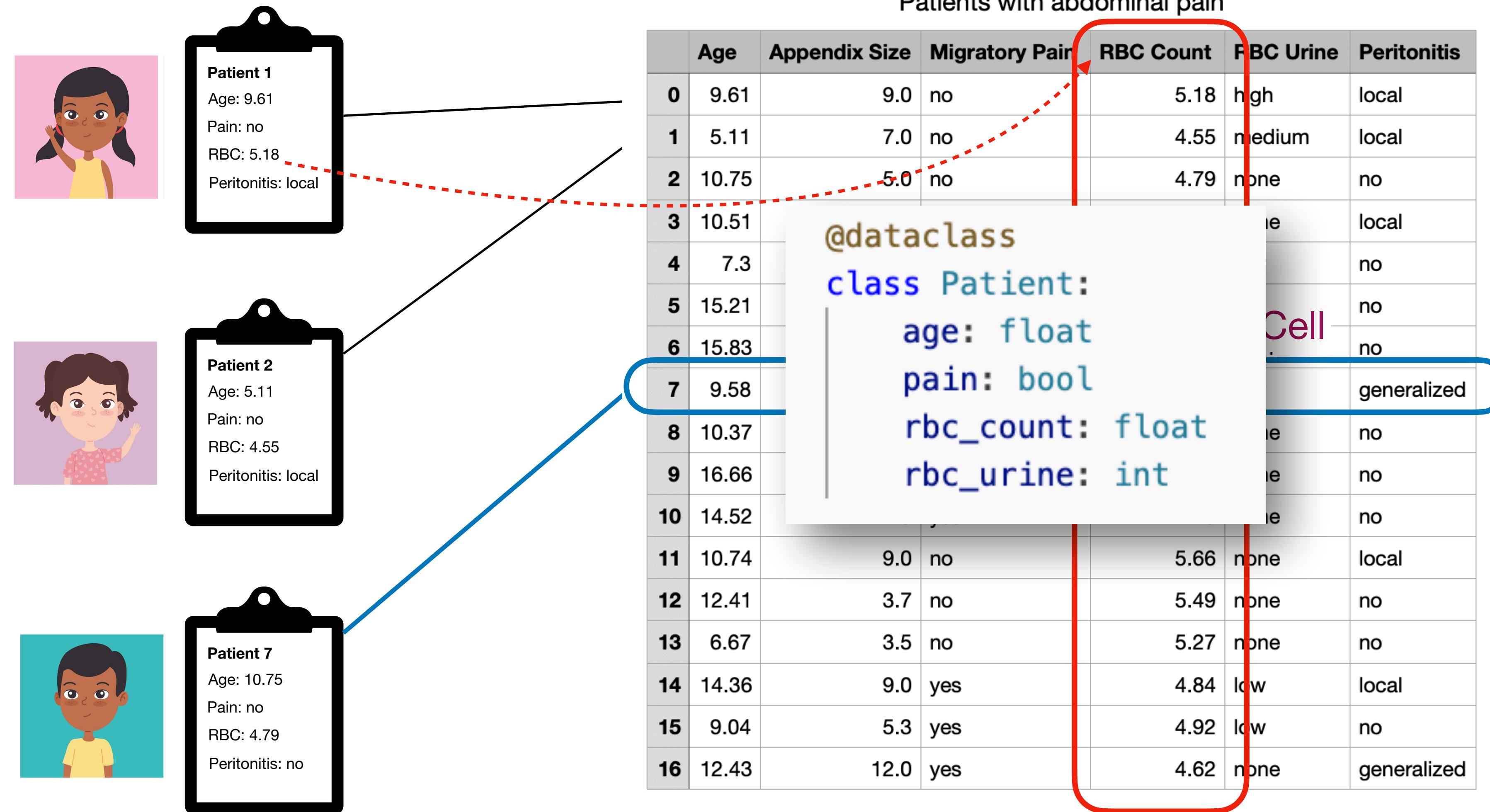
Table

Observation

Cell

Feature

Tabular data



Patients

Table

Observation

Feature

Types of features

Patients with abdominal pain

	Age	Appendix Size	Migratory Pain	RBC Count	RBC Urine	Peritonitis
0	9.61	9.0	no	5.18	high	local
1	5.11	7.0	no	4.55	medium	local
2	10.75	5.0	no	4.79	none	no
3	10.51	9.0	no	5.03	none	local
4	7.3	6.2	yes	4.64	low	no
5	15.21	8.5	yes	4.62	low	no
6	15.83	12.0	yes	4.33	high	no
7	9.58	7.0	yes	5.04	low	generalized
8	10.37	5.5	no	4.8	none	no
9	16.66	9.0	yes	5.31	none	no
10	14.52	4.5	yes	4.9	none	no
11	10.74	9.0	no	5.66	none	local
12	12.41	3.7	no	5.49	none	no
13	6.67	3.5	no	5.27	none	no
14	14.36	9.0	yes	4.84	low	local
15	9.04	5.3	yes	4.92	low	no
16	12.43	12.0	yes	4.62	none	generalized

Quantitative

E.g. 3, 2.7, -43, 8.2, etc.

Categorical (or Boolean)

E.g. True/False, {"local", "generalized", ...}

Ordinal

E.g. {low, medium, high...}

Types of features

Patients with abdominal pain

	Age	Appendix Size	Migratory Pain	RBC Count	RBC Urine	Peritonitis
0	9.61	9.0	no	5.18	high	local
1	5.11	7.0	no	4.55	medium	local
2	10.75	5.0	no	4.79	none	no
3	10.51	9.0	no	5.03	none	local
4	7.3	6.2	yes	4.64	low	no
5	15.21	8.5	yes	4.62	low	no
6	15.83	12.0	yes	4.33	high	no
7	9.58	7.0	yes	5.04	low	generalized
8	10.37	5.5	no	4.8	none	no
9	16.66	9.0	yes	5.31	none	no
10	14.52	4.5	yes	4.9	none	no
11	10.74	9.0	no	5.66	none	local
12	12.41	3.7	no	5.49	none	no
13	6.67	3.5	no	5.27	none	no
14	14.36	9.0	yes	4.84	low	local
15	9.04	5.3	yes	4.92	low	no
16	12.43	12.0	yes	4.62	none	generalized

Quantitative

E.g. 3, 2.7, -43, 8.2, etc.

Categorical (or Boolean)

E.g. True/False, {"local", "generalized", ...}

Ordinal

E.g. {low, medium, high...}

Types of features

Patients with abdominal pain						
	Age	Appendix Size	Migratory Pain	RBC Count	RBC Urine	Peritonitis
0	9.61	9.0	no	5.18	high	local
1	5.11	7.0	no	4.55	medium	local
2	10.75	5.0	no	4.79	none	no
3	10.51	9.0	no	5.03	none	local
4	7.3	6.2	yes	4.64	low	no
5	15.21	8.5	yes	4.62	low	no
6	15.83	12.0	yes	4.33	high	no
7	9.58	7.0	yes	5.04	low	generalized
8	10.37	5.5	no	4.8	none	no
9	16.66	9.0	yes	5.31	none	no
10	14.52	4.5	yes	4.9	none	no
11	10.74	9.0	no	5.66	none	local
12	12.41	3.7	no	5.49	none	no
13	6.67	3.5	no	5.27	none	no
14	14.36	9.0	yes	4.84	low	local
15	9.04	5.3	yes	4.92	low	no
16	12.43	12.0	yes	4.62	none	generaliz

Easier to assume everything is quantitative!

Quantitative

E.g. 3, 2.7, -43, 8.2, etc.

Categorical (or Boolean)

E.g. True/False, {"local", "generalized", ...}

Ordinal

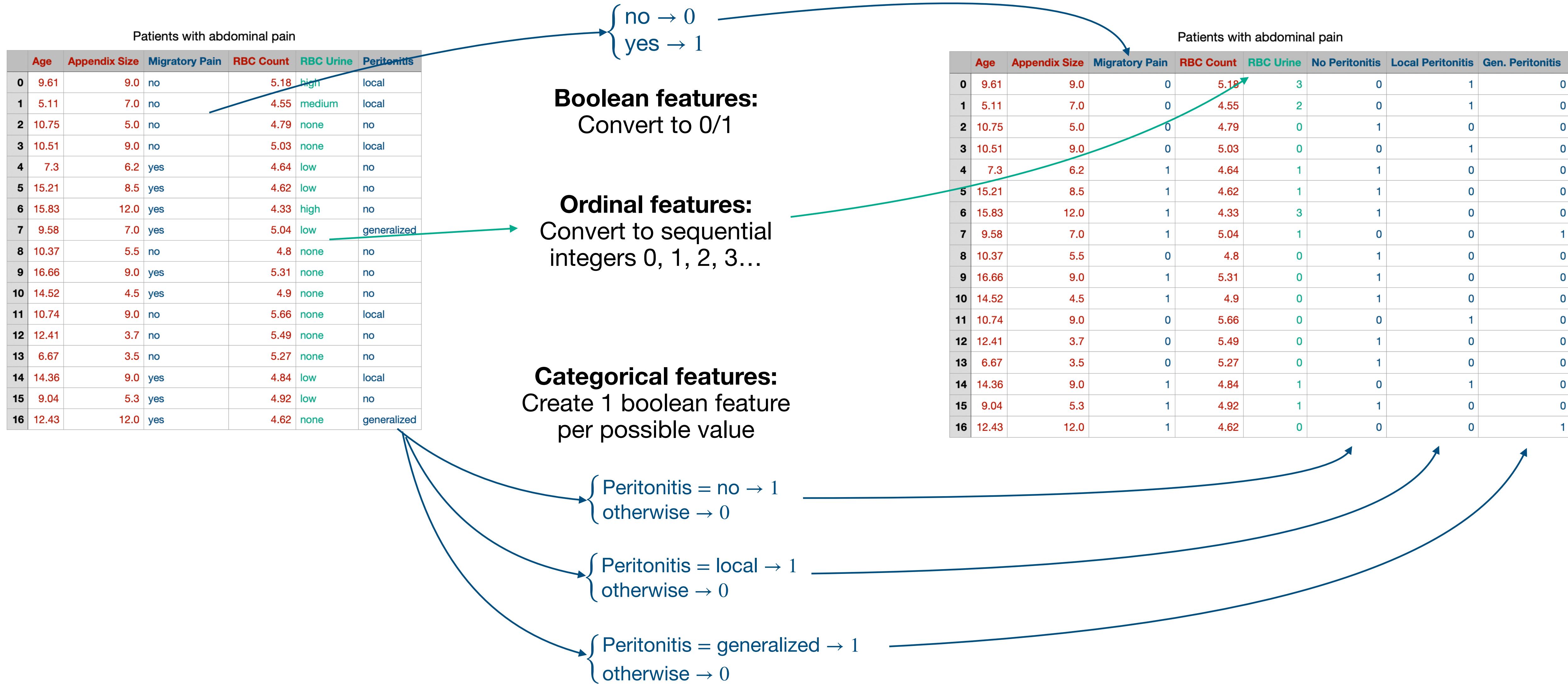
E.g. {low, medium, high...}

Input: $\mathbf{x} \xrightarrow[predict]{} \text{Output}$

$y, \quad y = f(\mathbf{x})$

Prediction function

Converting non-quantitative features



Mathematical abstraction

As table

Observation

	Patients with abdominal pain						
	Age	Appendix Size	Height	Weight	RBC Count	Temperature	WBC Count
0	16.66	9.0	174.0	65.0	5.31	36.6	6.6
1	10.74	9.0	146.0	57.5	5.66	37.3	10.2
2	9.04	5.3	134.0	29.4	4.92	36.0	5.1
3	10.75	5.0	155.0	54.5	4.79	37.7	10.3
4	7.3	6.2	123.0	23.5	4.64	37.4	21.1
5	5.11	7.0	116.0	22.0	4.55	40.2	19.4
6	14.36	9.0	163.0	50.0	4.84	37.5	14.3
7	9.61	9.0	140.0	29.2	5.18	38.7	14.3
8	15.83	12.0	153.0	59.0	4.33	36.7	12.8
9	9.58	7.0	132.0	24.7	5.04	38.4	13.5
10	10.37	5.5	156.0	39.0	4.8	37.4	5.6
11	14.52	4.5	181.0	55.0	4.9	37.0	9.0
12	12.41	3.7	150.5	42.5	5.49	37.2	9.1
13	6.67	3.5	124.0	38.5	5.27	39.6	16.8
14	15.21	8.5	155.0	85.0	4.62	36.8	12.4
15	12.43	12.0	157.0	46.0	4.62	37.1	16.4
16	10.51	9.0	134.5	27.0	5.03	37.4	12.8

Feature

$N \times d$ Matrix: $X \in \mathbb{R}^{N \times d}$

Observation

$X =$

As matrix

Feature

$$X = \begin{bmatrix} 16.66 & 9.0 & 174.0 & 65.0 & 5.31 & 36.6 & 6.6 \\ 10.74 & 9.0 & 146.0 & 57.5 & 5.66 & 37.3 & 10.2 \\ 9.04 & 5.3 & 134.0 & 29.4 & 4.92 & 36.0 & 5.1 \\ 10.75 & 5.0 & 155.0 & 54.5 & 4.79 & 37.7 & 10.3 \\ 7.3 & 6.2 & 123.0 & 23.5 & 4.64 & 37.4 & 21.1 \\ 5.11 & 7.0 & 116.0 & 22.0 & 4.55 & 40.2 & 19.4 \\ 14.36 & 9.0 & 163.0 & 50.0 & 4.84 & 37.5 & 14.3 \\ 9.61 & 9.0 & 140.0 & 29.2 & 5.18 & 38.7 & 14.3 \\ 15.83 & 12.0 & 153.0 & 59.0 & 4.33 & 36.7 & 12.8 \\ 9.58 & 7.0 & 132.0 & 24.7 & 5.04 & 38.4 & 13.5 \\ 10.37 & 5.5 & 156.0 & 39.0 & 4.8 & 37.4 & 5.6 \\ 14.52 & 4.5 & 181.0 & 55.0 & 4.9 & 37.0 & 9.0 \\ 12.41 & 3.7 & 150.5 & 42.5 & 5.49 & 37.2 & 9.1 \\ 6.67 & 3.5 & 124.0 & 38.5 & 5.27 & 39.6 & 16.8 \\ 15.21 & 8.5 & 155.0 & 85.0 & 4.62 & 36.8 & 12.4 \\ 12.43 & 12.0 & 157.0 & 46.0 & 4.62 & 37.1 & 16.4 \\ 10.51 & 9.0 & 134.5 & 27.0 & 5.03 & 37.4 & 12.8 \end{bmatrix}$$

$\underbrace{\hspace{10cm}}$

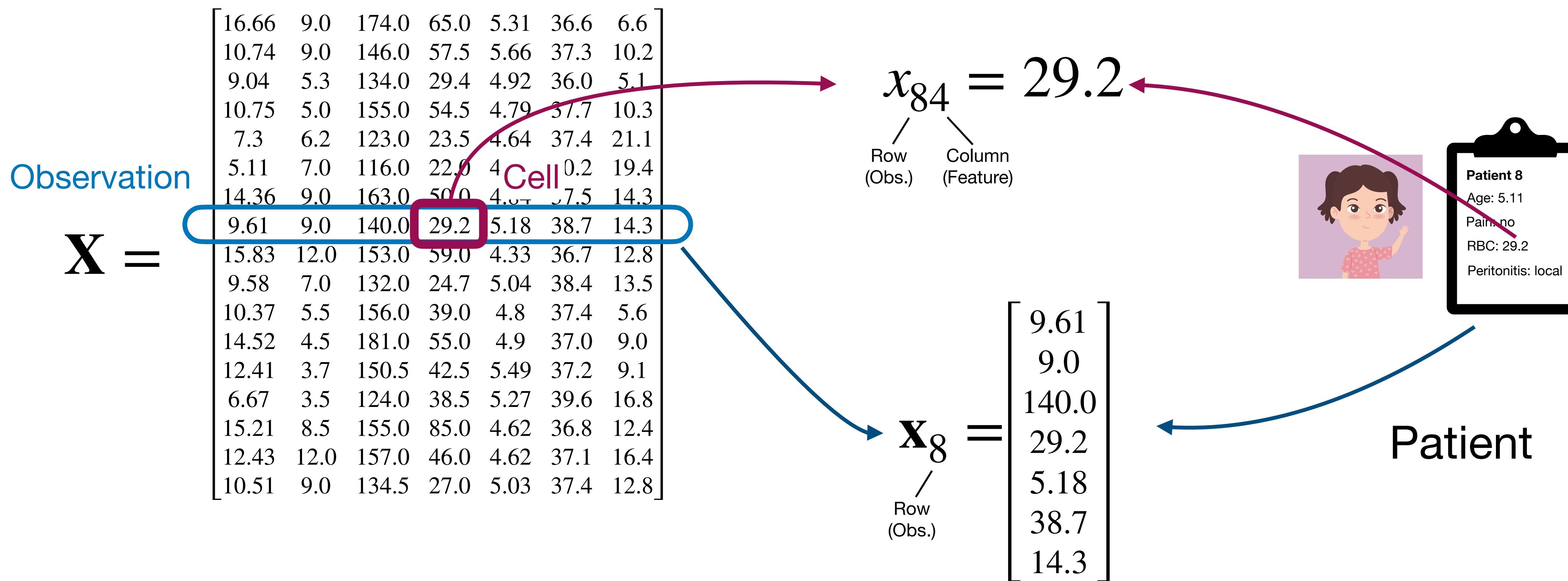
d

(features)

N
(Obs.)

Matrix, vector and scalar notation

As matrix



Notation warning!

As matrix

Observation

$\mathbf{X} =$

16.66	9.0	174.0	65.0	5.31	36.6	6.6
10.74	9.0	146.0	57.5	5.66	37.3	10.2
9.04	5.3	134.0	29.4	4.92	36.0	5.1
10.75	5.0	155.0	54.5	4.79	37.7	10.3
7.3	6.2	123.0	23.5	4.64	37.4	21.1
5.11	7.0	116.0	22.0	4.55	40.2	19.4
14.36	9.0	163.0	50.0	4.84	37.5	14.3
9.61	9.0	140.0	29.2	5.18	38.7	14.3
15.83	12.0	153.0	59.0	4.33	36.7	12.8
9.58	7.0	132.0	24.7	5.04	38.4	13.5
10.37	5.5	156.0	39.0	4.8	37.4	5.6
14.52	4.5	181.0	55.0	4.9	37.0	9.0
12.41	3.7	150.5	42.5	5.49	37.2	9.1
6.67	3.5	124.0	38.5	5.27	39.6	16.8
15.21	8.5	155.0	85.0	4.62	36.8	12.4
12.43	12.0	157.0	46.0	4.62	37.1	16.4
10.51	9.0	134.5	27.0	5.03	37.4	12.8

Vector

Notation:

$$\mathbf{x}_8 = \begin{matrix} / \\ \text{Row} \\ (\text{Obs.}) \end{matrix} \begin{bmatrix} 9.61 \\ 9.0 \\ 140.0 \\ 29.2 \\ 5.18 \\ 38.7 \\ 14.3 \end{bmatrix}$$

In Python:

```
x[7]
✓ 0.0s
Python
array([ 9.61,  9. , 140. , 29.2 , 5.18, 38.7 , 14.3 ])
```

Column vector ($d \times 1$ matrix)

Notation:

$$\mathbf{x}_8 = \begin{bmatrix} 9.61 \\ 9.0 \\ 140.0 \\ 29.2 \\ 5.18 \\ 38.7 \\ 14.3 \end{bmatrix}$$

In Python:

```
x[7][:, None]
✓ 0.0s
Python
[[ 9.61]
 [ 9. ]
 [140. ]
 [ 29.2 ]
 [ 5.18]
 [ 38.7 ]
 [ 14.3 ]]
```

Row vector ($1 \times d$ matrix)

Notation:

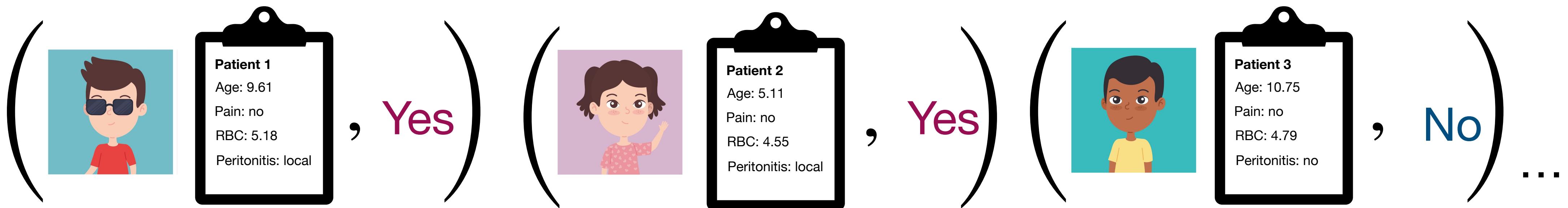
$$\mathbf{x}_8^T = [9.61 \ 9.0 \ 140.0 \ 29.2 \ 5.18 \ 38.7 \ 14.3]$$

In Python:

```
x[7][None, :]
✓ 0.0s
Python
array([[ 9.61,  9. , 140. , 29.2 , 5.18, 38.7 , 14.3 ]])
```

Dataset

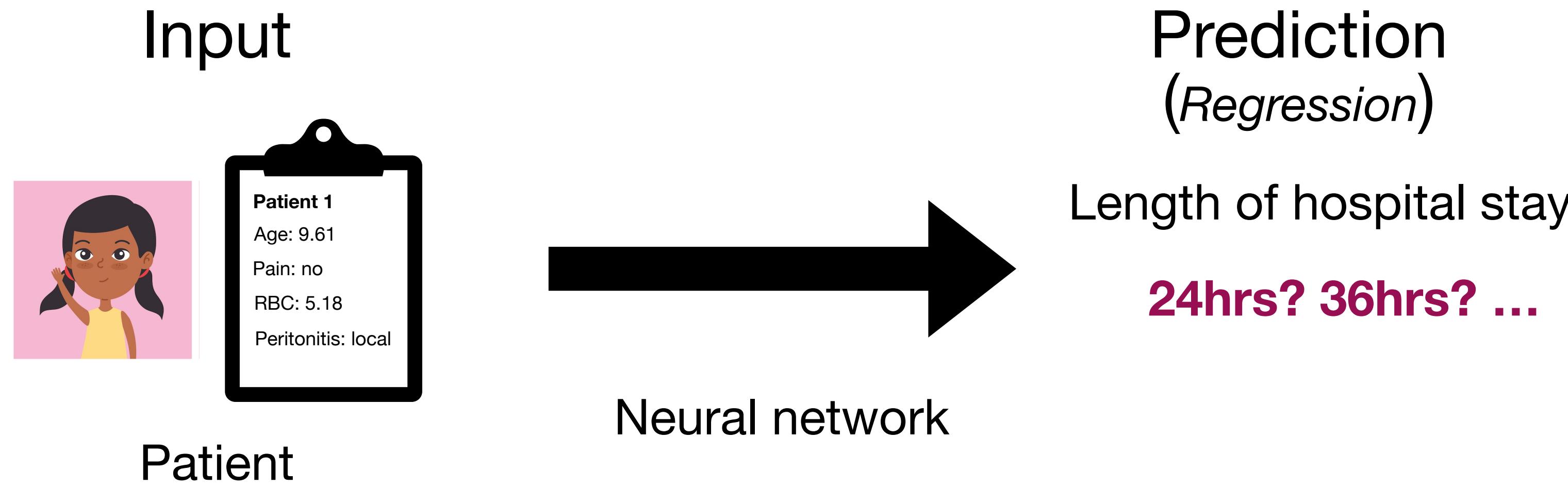
Set of known inputs and **outputs**



Some notation:

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_N, y_N)\}$$

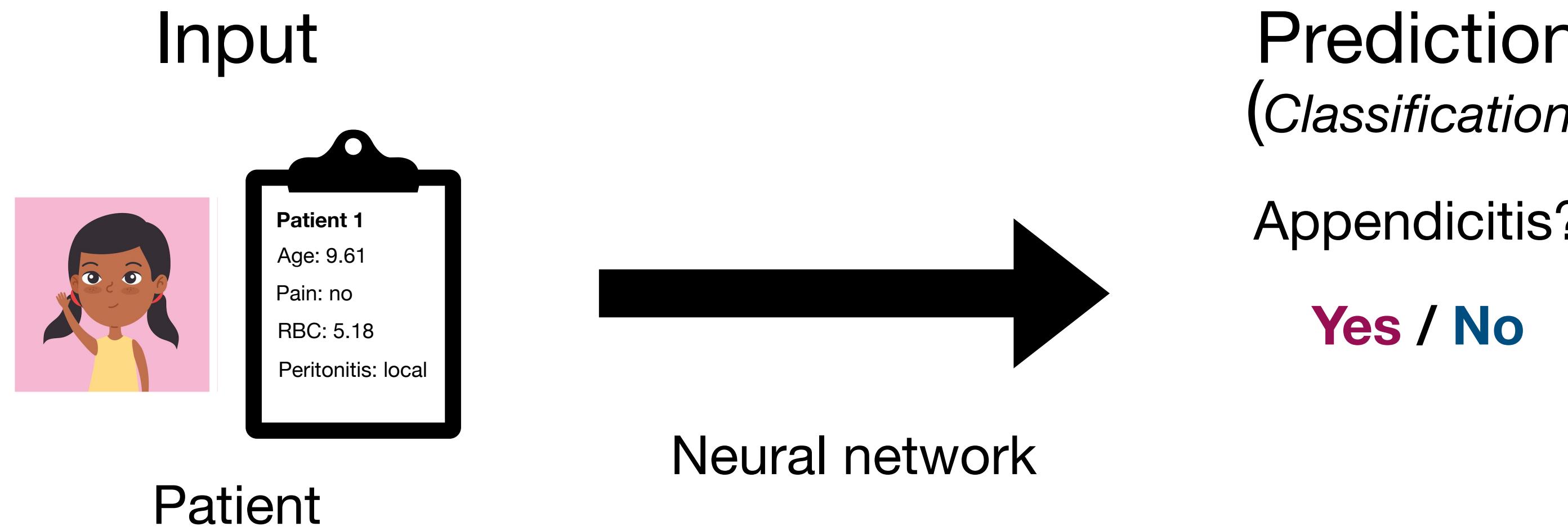
Quantitative outputs: *Regression*



$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_N, y_N)\}$$

Output domain: $y \in \mathbb{R}$

Categorical outputs: *Classification*

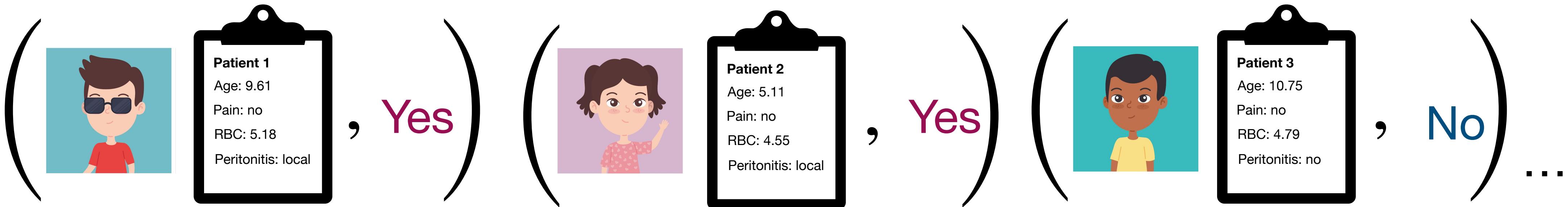


$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_N, y_N)\}$$

Output domain: $y \in \{0,1\}$

Dataset

Set of known inputs and **outputs**



Input table

	Age	Appendix Size	Migratory Pain	RBC Count	RBC Urine	Peritonitis
0	9.61	9.0	no	5.18	high	local
1	5.11	7.0	no	4.55	medium	local
2	10.75	5.0	no	4.79	none	no
3	10.51	9.0	no	5.03	none	local
4	7.3	6.2	yes	4.64	low	no
5	15.21	8.5	yes	4.62	low	no
6	15.83	12.0	yes	4.33	high	no
7	9.58	7.0	yes	5.04	low	generalized
8	10.37	5.5	no	4.8	none	no
9	16.66	9.0	yes	5.31	none	no
10	14.52	4.5	yes	4.9	none	no
11	10.74	9.0	no	5.66	none	local
12	12.41	3.7	no	5.49	none	no
13	6.67	3.5	no	5.27	none	no
14	14.36	9.0	yes	4.84	low	local
15	9.04	5.3	yes	4.92	low	no
16	12.43	12.0	yes	4.62	none	generalized

Output table

	Diagnosis
0	appendicitis
1	no appendicitis
2	no appendicitis
3	no appendicitis
4	appendicitis
5	no appendicitis
6	no appendicitis
7	no appendicitis
8	no appendicitis
9	appendicitis
10	appendicitis
11	no appendicitis
12	no appendicitis
13	no appendicitis
14	appendicitis
15	no appendicitis
16	appendicitis

Mathematical abstraction

Labels become a vector

Output table

Diagnosis	
	Diagnosis
0	appendicitis
1	no appendicitis
2	no appendicitis
3	no appendicitis
4	appendicitis
5	no appendicitis
6	no appendicitis
7	no appendicitis
8	no appendicitis
9	appendicitis
10	appendicitis
11	no appendicitis
12	no appendicitis
13	no appendicitis
14	appendicitis
15	no appendicitis
16	appendicitis

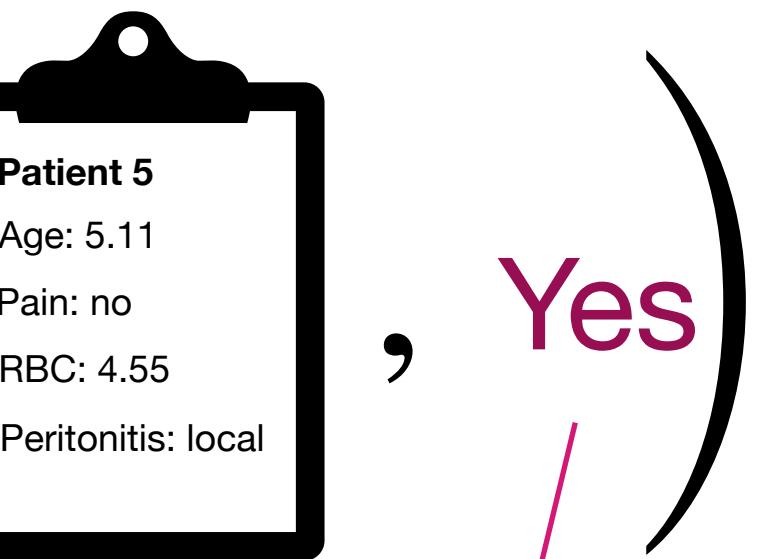
Converted

Diagnosis	
	Diagnosis
0	1
1	0
2	0
3	0
4	1
5	0
6	0
7	0
8	0
9	1
10	0
11	0
12	0
13	0
14	1
15	0
16	1

$$\mathbf{y} =$$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \boxed{1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

N
(Obs.)



Yes

$$y_5 = 1$$

Index
(Obs.)