

Lecture 23: Introduction to machine learning

Instructor: Michael Szell

Nov 22, 2023



Today you will see an overview of machine learning

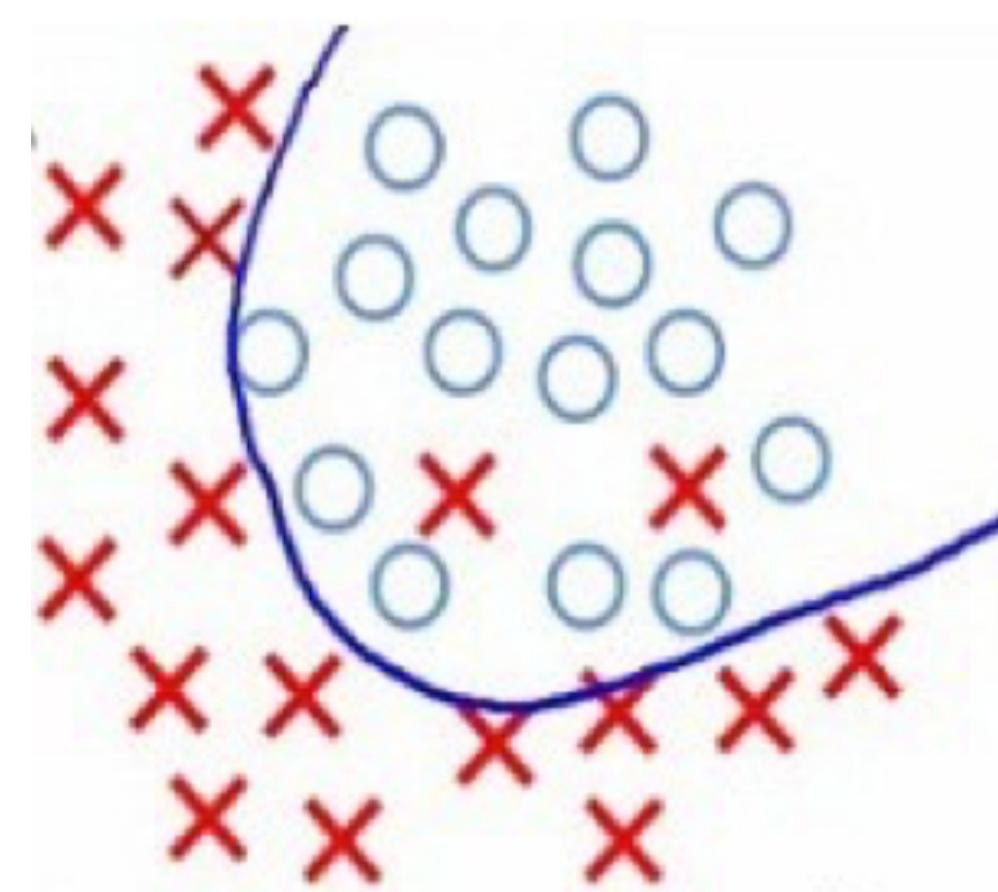
Difference to traditional
programming



Fundamental definitions
and techniques



Model evaluation



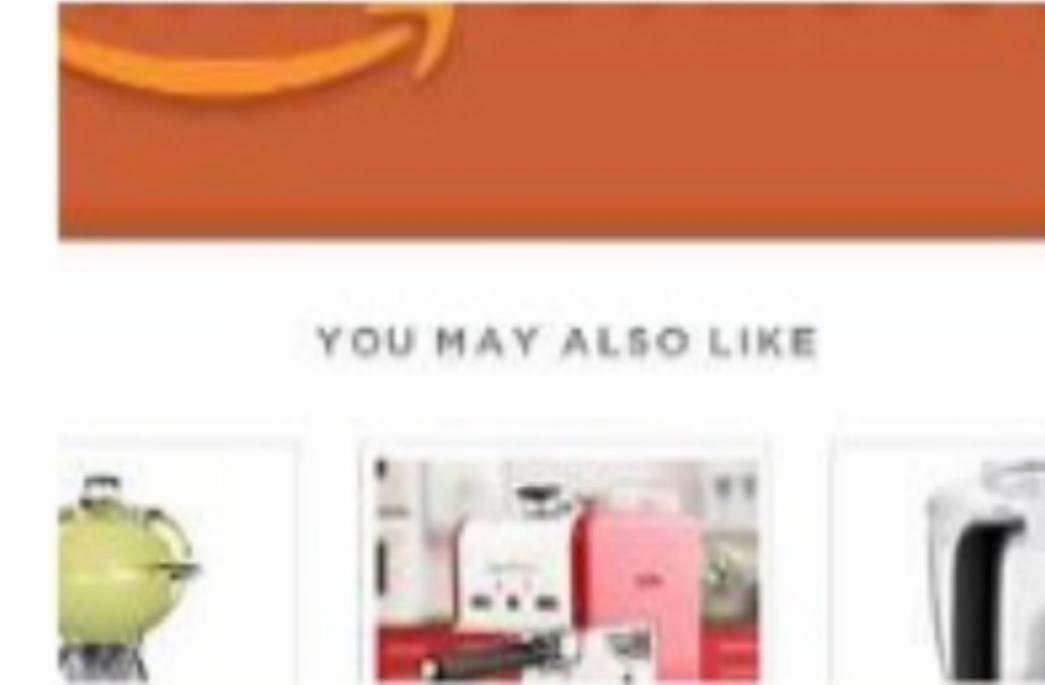
Machine learning is everywhere



Facial Recognition



Spam Detection



Recommendations



Medical Diagnosis



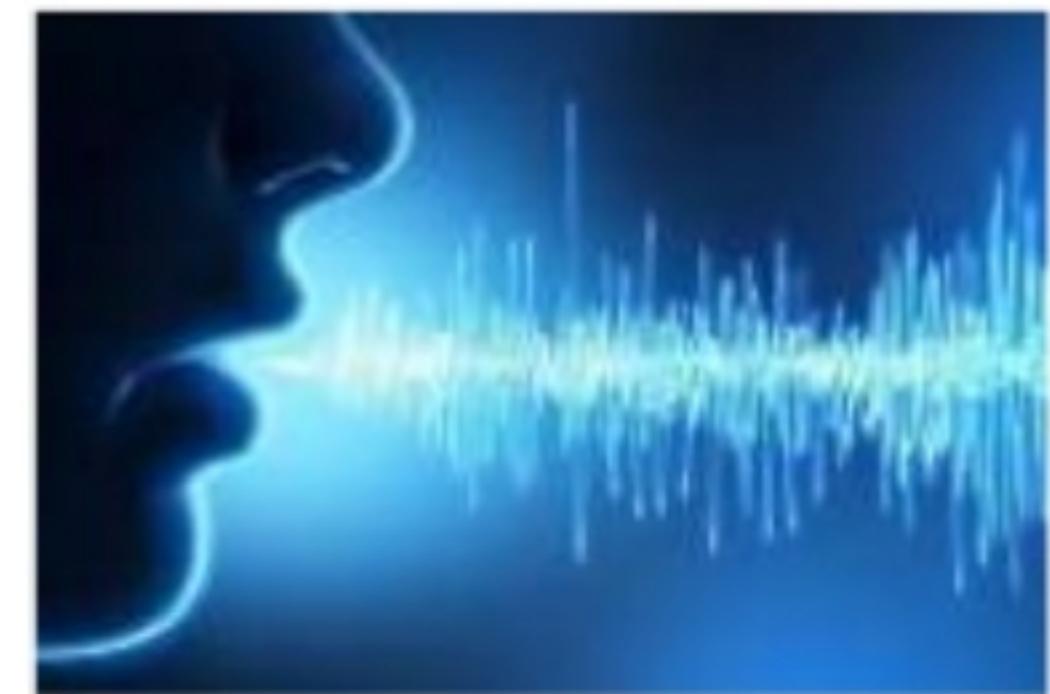
Smart Cars



Fraud Detection



Online Search



Speech

There is no common definition of machine learning

“Field of study that gives computers the ability to learn without being explicitly programmed”. Arthur Samuel (1959)



Creating and using models that are learned from data

Traditional programming
versus
Machine learning

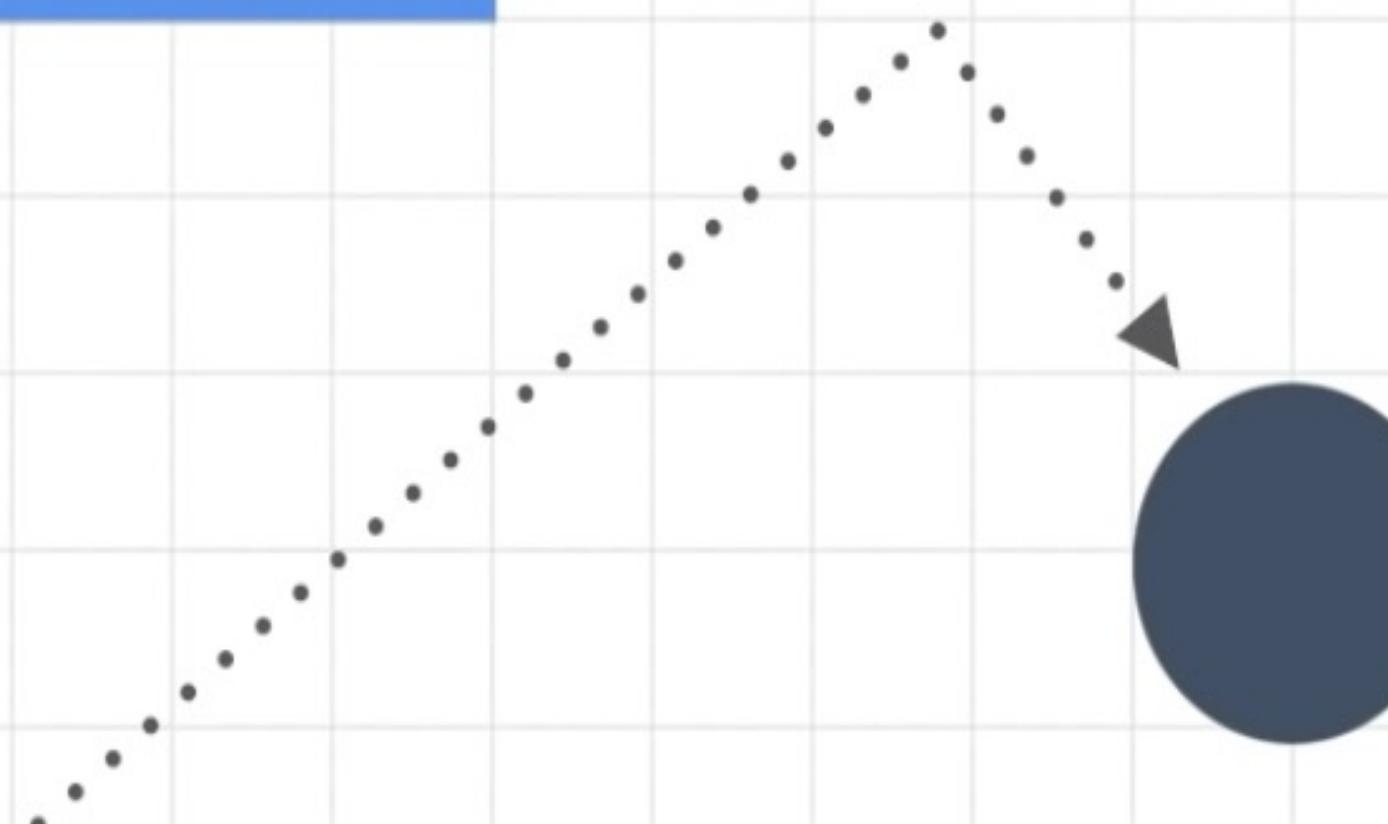
Traditional programming





```
if (ball.collide(brick)){  
    removeBrick();  
    ball.dy=-1*(ball.dy);  
}
```





```
if (ball.collide(brick)){  
    removeBrick();  
    ball.dy=-1*(ball.dy);  
}
```



Traditional programming



Traditional programming vs Machine learning



Activity recognition in traditional programming



```
if(speed<4){  
    status=WALKING;  
}
```

Activity recognition in traditional programming



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```

Activity recognition in traditional programming



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```

Activity recognition in traditional programming



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```



// Uh oh

Activity recognition in machine learning



0101001010100101
0101001010101001
0111010100101010
0101010010101001
010100101010

Label =
WALKING



1010100101001010
1010101010010010
0100010010011111
010101111010100
100111101011

Label =
RUNNING



1001010011111010
1011101010111010
1011101010101111
0101010111111110
001111010101

Label =
BIKING



111111111010011
1010011111010111
1101010101110101
0101011101010101
010100111110
Label = GOLFING
(Sort of)

Traditional programming of a data relation

X = -1, 0, 1, 2, 3, 4

Y = -2, 1, 4, 7, 10, 13

Traditional programming of a data relation

X = -1, 0, 1, 2, 3, 4

Y = -2, 1, 4, 7, 10, 13

Y = 3X + 1

Traditional programming of a data relation

```
getResult(x)
{
    return (3 * x) + 1
}
```

```
y = getResult(x);
```

Machine learning to learn a data relation

```
getResult(x)
{
    p1, p2 = learn();
    return (p1*x)+p2;
}

y = getResult(x);
```

How are things learned?

Supervised learning

Unsupervised learning

Reinforcement learning

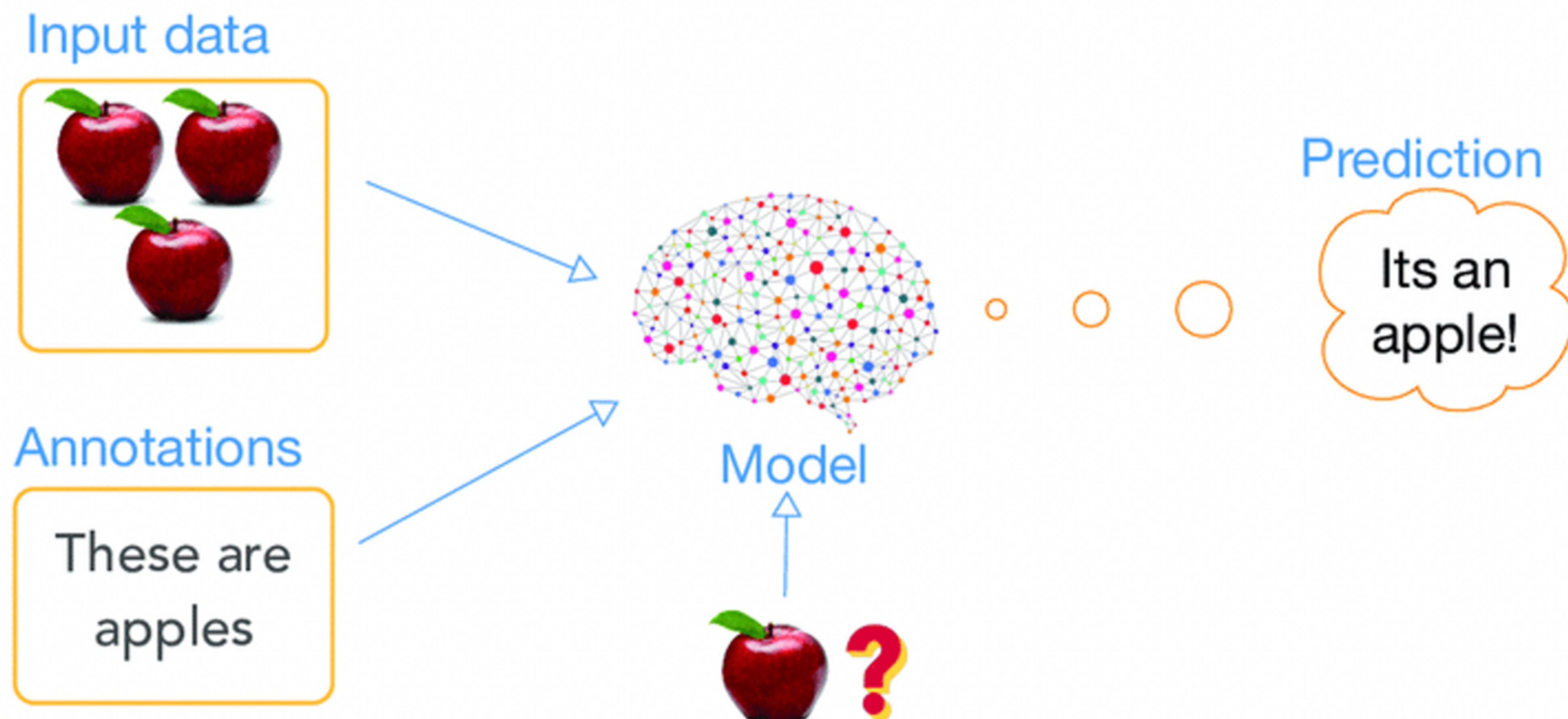
How are things learned?

Supervised learning

Unsupervised learning

Reinforcement learning

Supervised learning

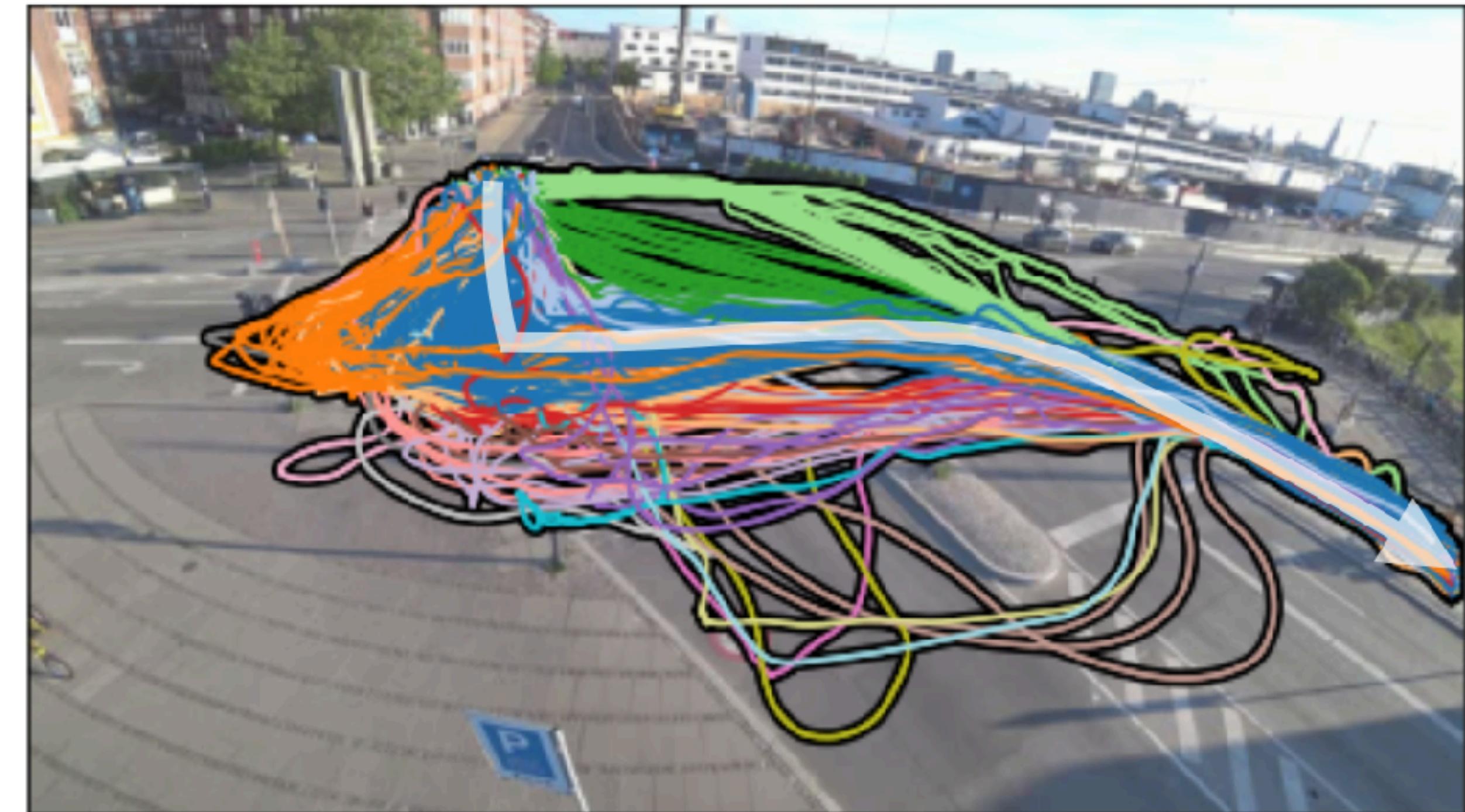


Supervised learning

Design

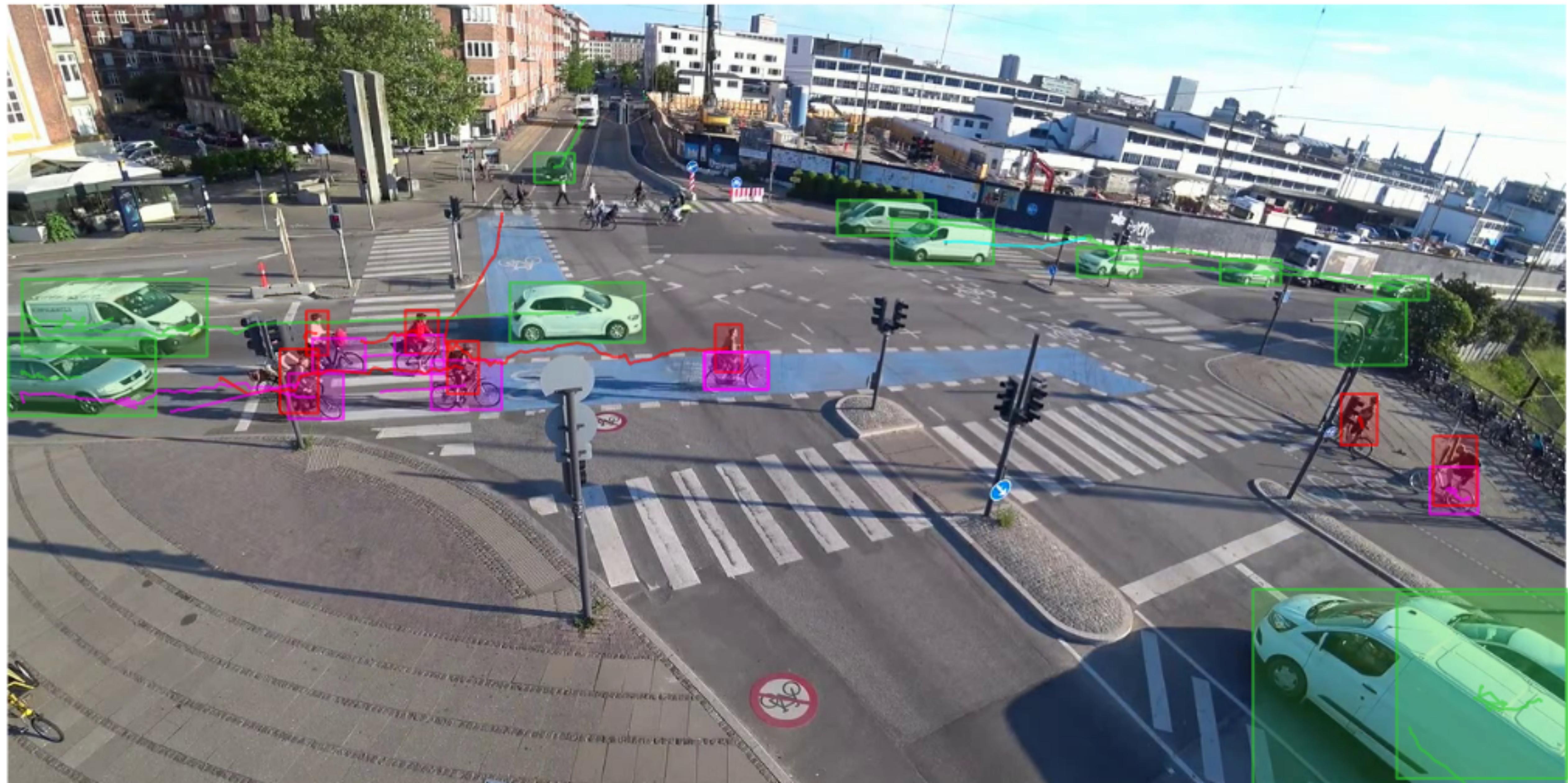


Reality



Breum, Kostic & Szell. Computational Desire Line Analysis of Cyclists on the Dybbølsbro Intersection in Copenhagen, Transport Findings 56683 (2022)

Supervised learning



Supervised learning

Instructions for a data annotation tool to label traffic participants



How are things learned?

Supervised learning

A model learns from a labeled dataset with guidance

Unsupervised learning

Reinforcement learning

How are things learned?

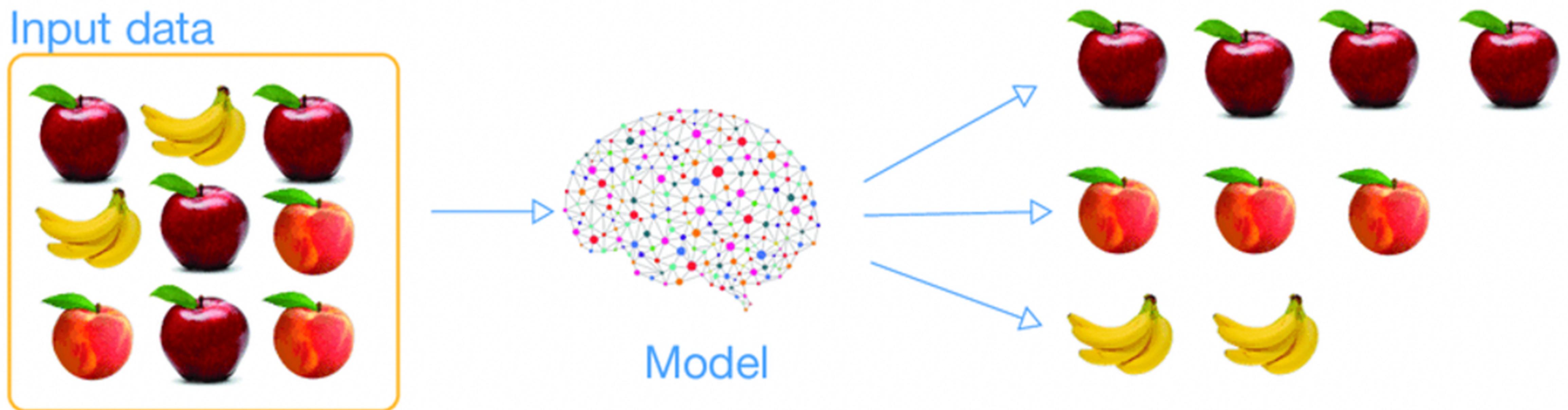
Supervised learning

A model learns from a labeled dataset with guidance

Unsupervised learning

Reinforcement learning

Unsupervised learning



How are things learned?

Supervised learning

A model learns from a labeled dataset with guidance

Unsupervised learning

A model learns from an unlabeled dataset without guidance

Reinforcement learning

How are things learned?

Supervised learning

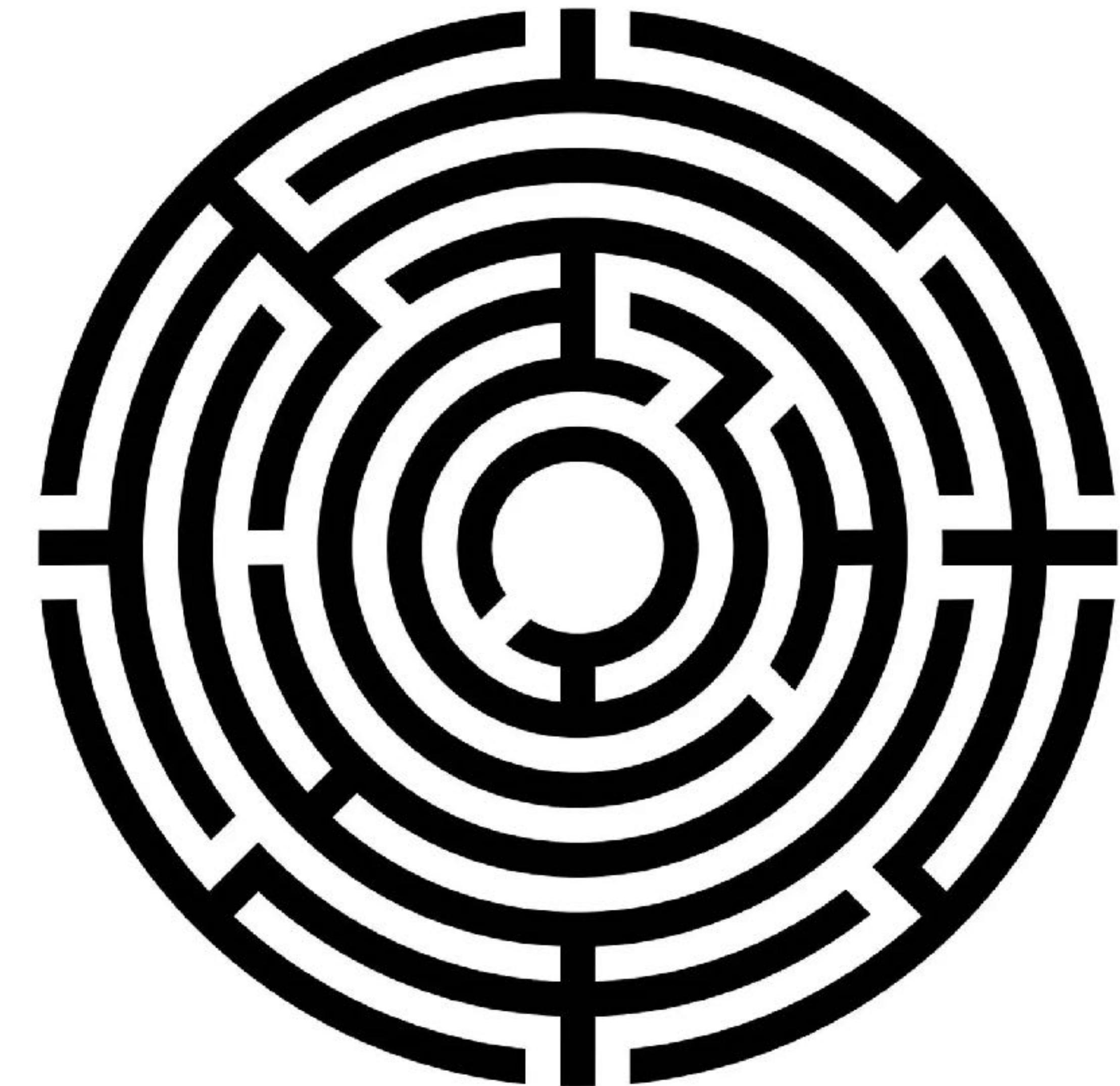
A model learns from a labeled dataset with guidance

Unsupervised learning

A model learns from an unlabeled dataset without guidance

Reinforcement learning

Reinforcement learning



DAY1



DAY10



How are things learned?

Supervised learning

A model learns from a labeled dataset with guidance

Unsupervised learning

A model learns from an unlabeled dataset without guidance

Reinforcement learning

A model interacts with an environment and learns by trial-and-error

What is a model?

In machine learning:

A program that has been trained to recognize certain patterns in data or to make predictions

What is a model?

In machine learning:

A program that has been trained to recognize certain patterns in data or to make predictions

Outside of machine learning:

A specification of a mathematical or probabilistic relationship between different variables

Required choices for machine learning models

Training data and evaluation method

Feature representation / engineering

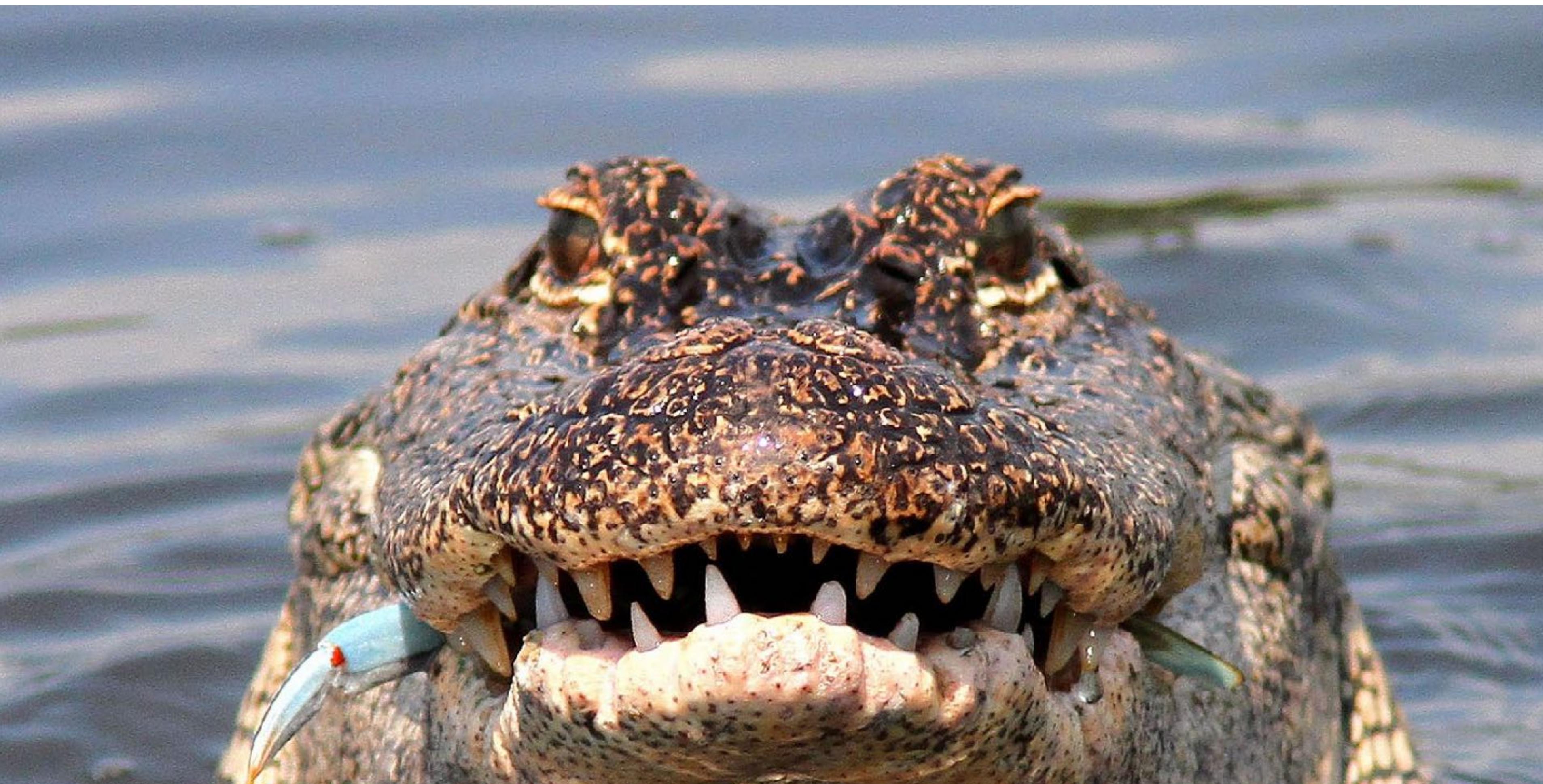
Today

Distance metric for feature vectors

Objective function and constraints

Optimisation method for learning the model

Labeling reptiles



Labeling reptiles

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes



Model: not enough data to generalize

Labeling reptiles

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlensnake	TRUE	TRUE	TRUE	TRUE	0	Yes



Model: egg-laying & has scales & poisonous & cold-blooded & has no legs

Labeling reptiles

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlesnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes



Model: egg-laying & has scales & poisonous & cold-blooded & has no legs

Labeling reptiles

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlesnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes



Model: egg-laying & has scales & poisonous & cold-blooded & has no legs

Labeling reptiles

Features				Features		Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlensnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes



Model: has scales & cold-blooded & has no legs

Labeling reptiles

Features				Features		Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlensnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Chicken	TRUE	TRUE	FALSE	FALSE	2	No



Model: has scales & cold-blooded & has no legs

Labeling reptiles

Features				Features		Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlensnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Chicken	TRUE	TRUE	FALSE	FALSE	2	No
Alligator	TRUE	TRUE	FALSE	TRUE	4	Yes



Model: has scales & cold-blooded & has no legs

Labeling reptiles

Name	Features			Features		Label
	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlensnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Chicken	TRUE	TRUE	FALSE	FALSE	2	No
Alligator	TRUE	TRUE	FALSE	TRUE	4	Yes



Model: has scales & cold-blooded & has 0 or 4 legs

Labeling reptiles

Features				Features		Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlensnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Chicken	TRUE	TRUE	FALSE	FALSE	2	No
Alligator	TRUE	TRUE	FALSE	TRUE	4	Yes
Dart frog	TRUE	FALSE	TRUE	FALSE	4	No



Model: has scales & cold-blooded & has 0 or 4 legs

Labeling reptiles

Features				Features		Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlesnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Chicken	TRUE	TRUE	FALSE	FALSE	2	No
Alligator	TRUE	TRUE	FALSE	TRUE	4	Yes
Dart frog	TRUE	FALSE	TRUE	FALSE	4	No
Salmon	TRUE	TRUE	FALSE	TRUE	0	No
Python	TRUE	TRUE	FALSE	TRUE	0	Yes

Model: has scales & cold-blooded & has 0 or 4 legs



Labeling reptiles

Name	Features			Features		Label
	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlesnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Chicken	TRUE	TRUE	FALSE	FALSE	2	No
Alligator	TRUE	TRUE	FALSE	TRUE	4	Yes
Dart frog	TRUE	FALSE	TRUE	FALSE	4	No
Salmon	TRUE	TRUE	FALSE	TRUE	0	No
Python	TRUE	TRUE	FALSE	TRUE	0	Yes

Model: has scales & cold-blooded & has 0 or 4 legs



Not enough data

Labeling reptiles

Name	Features			Features		Label
	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlesnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Chicken	TRUE	TRUE	FALSE	FALSE	2	No
Alligator	TRUE	TRUE	FALSE	TRUE	4	Yes
Dart frog	TRUE	FALSE	TRUE	FALSE	4	No
Salmon	TRUE	TRUE	FALSE	TRUE	0	No
Python	TRUE	TRUE	FALSE	TRUE	0	Yes

Model: has scales & cold-blooded & has 0 or 4 legs



Labeling reptiles

	Features			Features		Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Cobra	TRUE	TRUE	TRUE	TRUE	0	Yes
Rattlesnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Chicken	TRUE	TRUE	FALSE	FALSE	2	No
Alligator	TRUE	TRUE	FALSE	TRUE	4	Yes
Dart frog	TRUE	FALSE	TRUE	FALSE	4	No
Salmon	TRUE	TRUE	FALSE	TRUE	0	No
Python	TRUE	TRUE	FALSE	TRUE	0	Yes

Model: has scales & cold-blooded

What would be a machine
learning approach?

Feature engineering: Turning raw data into feature vectors

A **feature vector** is a numeric representation of an object.

Feature engineering: Turning raw data into feature vectors

A feature vector is a numeric representation of an object.

- Which features to include?
- How to measure distance between training records?
- How to weight features?

Feature engineering: Turning raw data into feature vectors

A feature vector is a numeric representation of an object.

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Rattlensnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Dart frog	TRUE	FALSE	TRUE	FALSE	4	No

Feature engineering: Turning raw data into feature vectors

A feature vector is a numeric representation of an object.

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# Legs	Reptile
Rattlensnake	TRUE	TRUE	TRUE	TRUE	0	Yes
Boa Constrictor	FALSE	TRUE	FALSE	TRUE	0	Yes
Dart frog	TRUE	FALSE	TRUE	FALSE	4	No

bool bool bool bool int

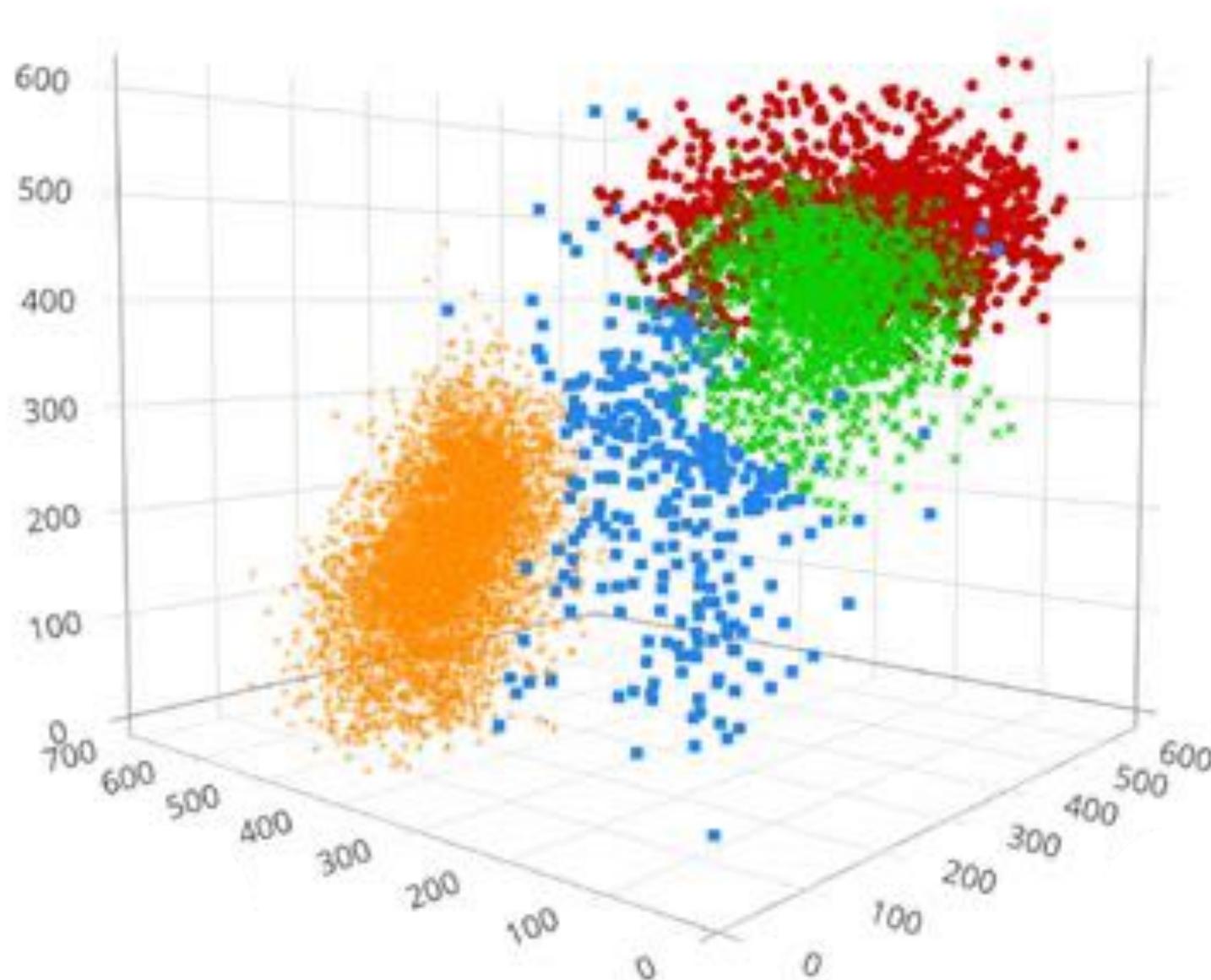
```
rattlesnake = [1,1,1,1,0]
```

```
boa = [0,1,0,1,0]
```

```
frog = [1,0,1,0,4]
```

Feature engineering: Turning raw data into feature vectors

The feature vectors represent points in a high-dimensional space.



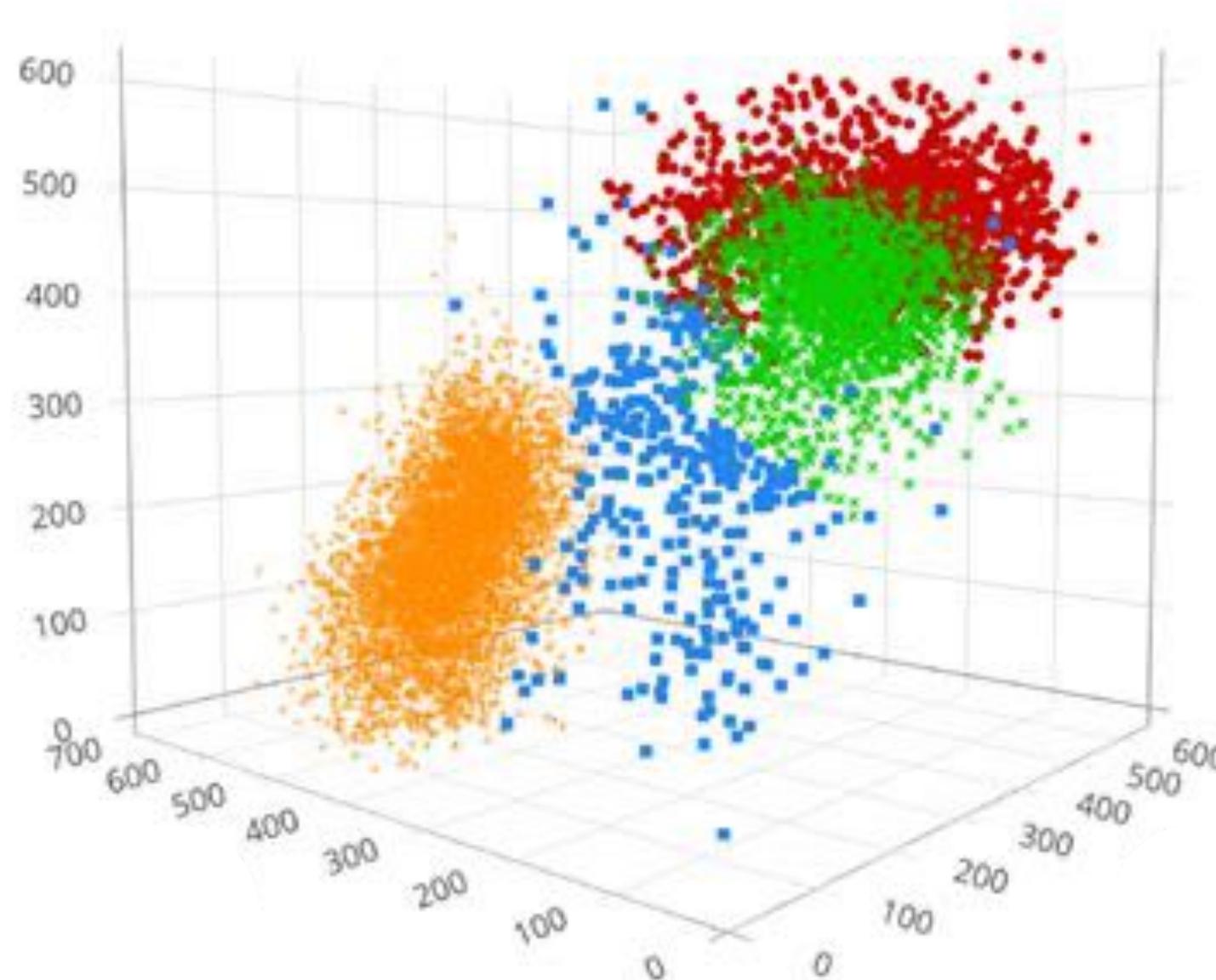
rattlesnake = [1,1,1,1,0]

boa = [0,1,0,1,0]

frog = [1,0,1,0,4]

Feature engineering: Turning raw data into feature vectors

The feature vectors represent points in a high-dimensional space.



How distant are they to each other?
Do they cluster?

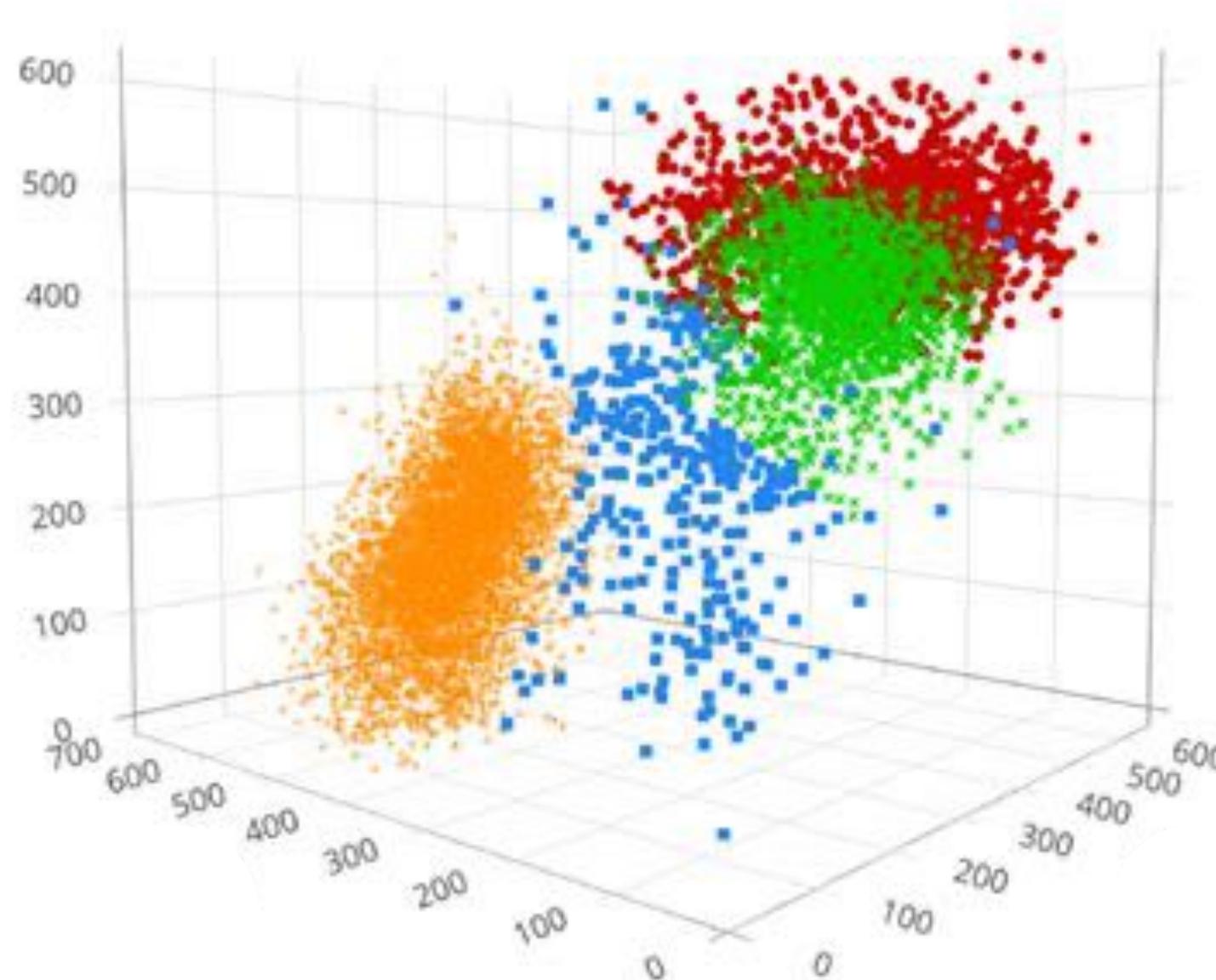
rattlesnake = [1,1,1,1,0]

boa = [0,1,0,1,0]

frog = [1,0,1,0,4]

Feature engineering: Turning raw data into feature vectors

The feature vectors represent points in a high-dimensional space.



How distant are they to each other?
Do they cluster?

If so, find a classifier surface that
optimally separates labeled clusters.

rattlesnake = [1,1,1,1,0]

boa = [0,1,0,1,0]

frog = [1,0,1,0,4]

Defining distance

A **distance measure** $d(A, B)$ has the properties:

1) Symmetry:

$$d(A, B) = d(B, A)$$

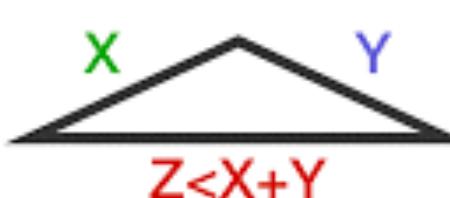
2) Identification:

$$d(A, B) = 0 \quad \text{only if} \quad A = B$$

3) Non-negativity:

$$d(A, B) \geq 0$$

4) Triangle inequality: $d(A, C) \leq d(A, B) + d(B, C)$



Euclidian distance

$$\mathbf{p} = (p_1, p_2, \dots, p_n)$$

$$\mathbf{q} = (q_1, q_2, \dots, q_n)$$

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Euclidian distance

$$\mathbf{p} = (p_1, p_2, \dots, p_n)$$

$$\mathbf{q} = (q_1, q_2, \dots, q_n)$$

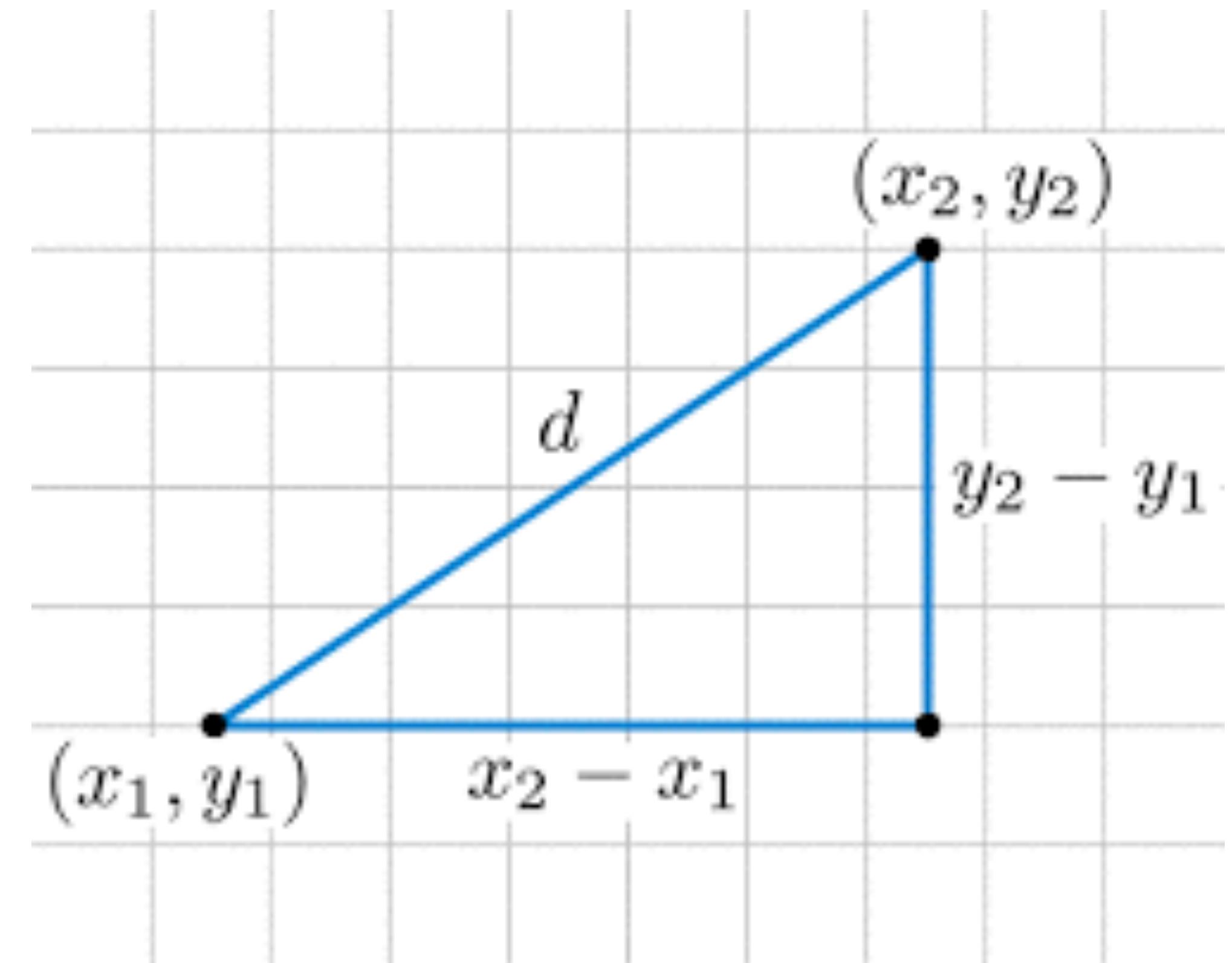
$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

In 2 dimensions, a common notation is:



$$\mathbf{p} = (x_1, y_1)$$

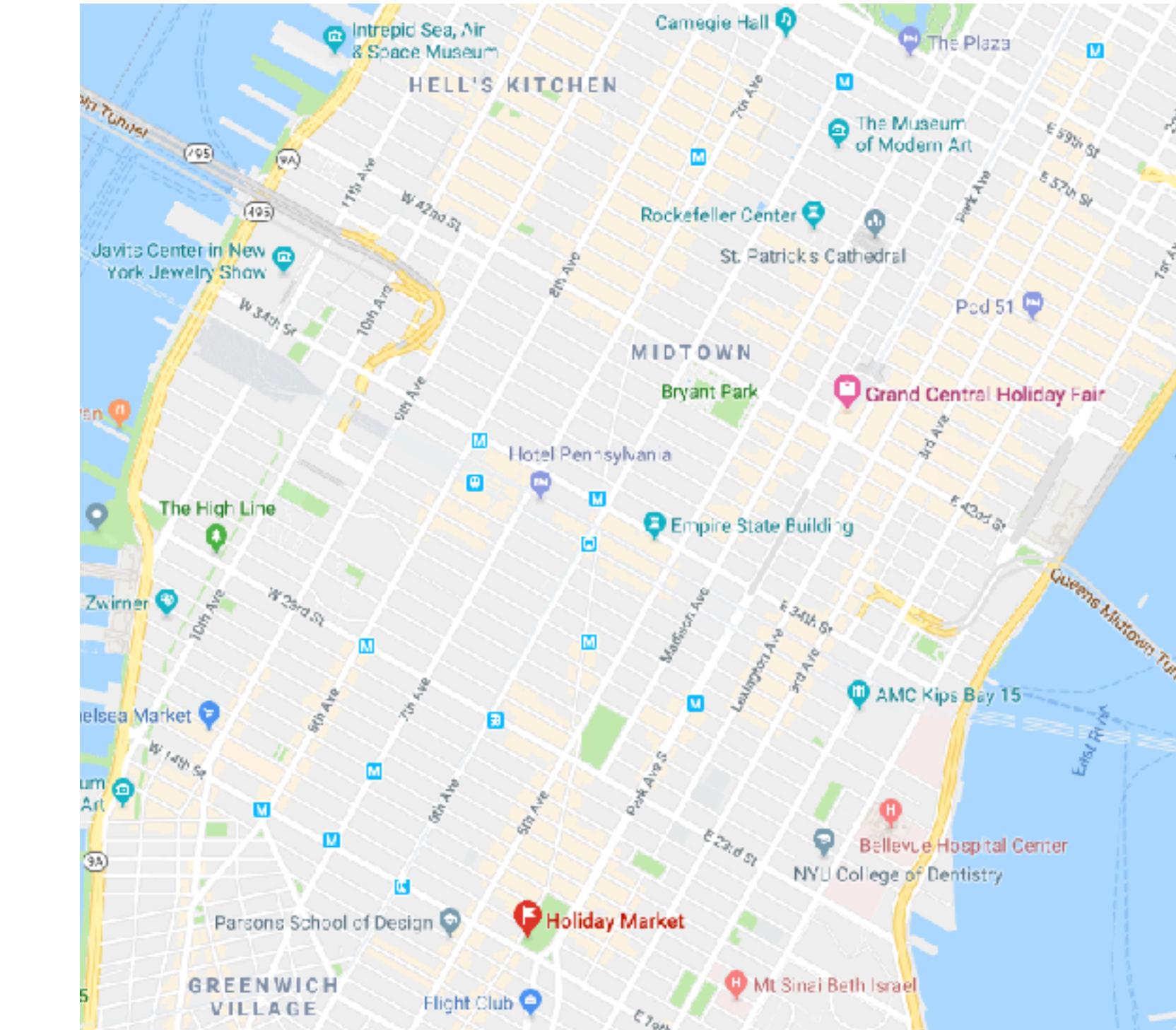
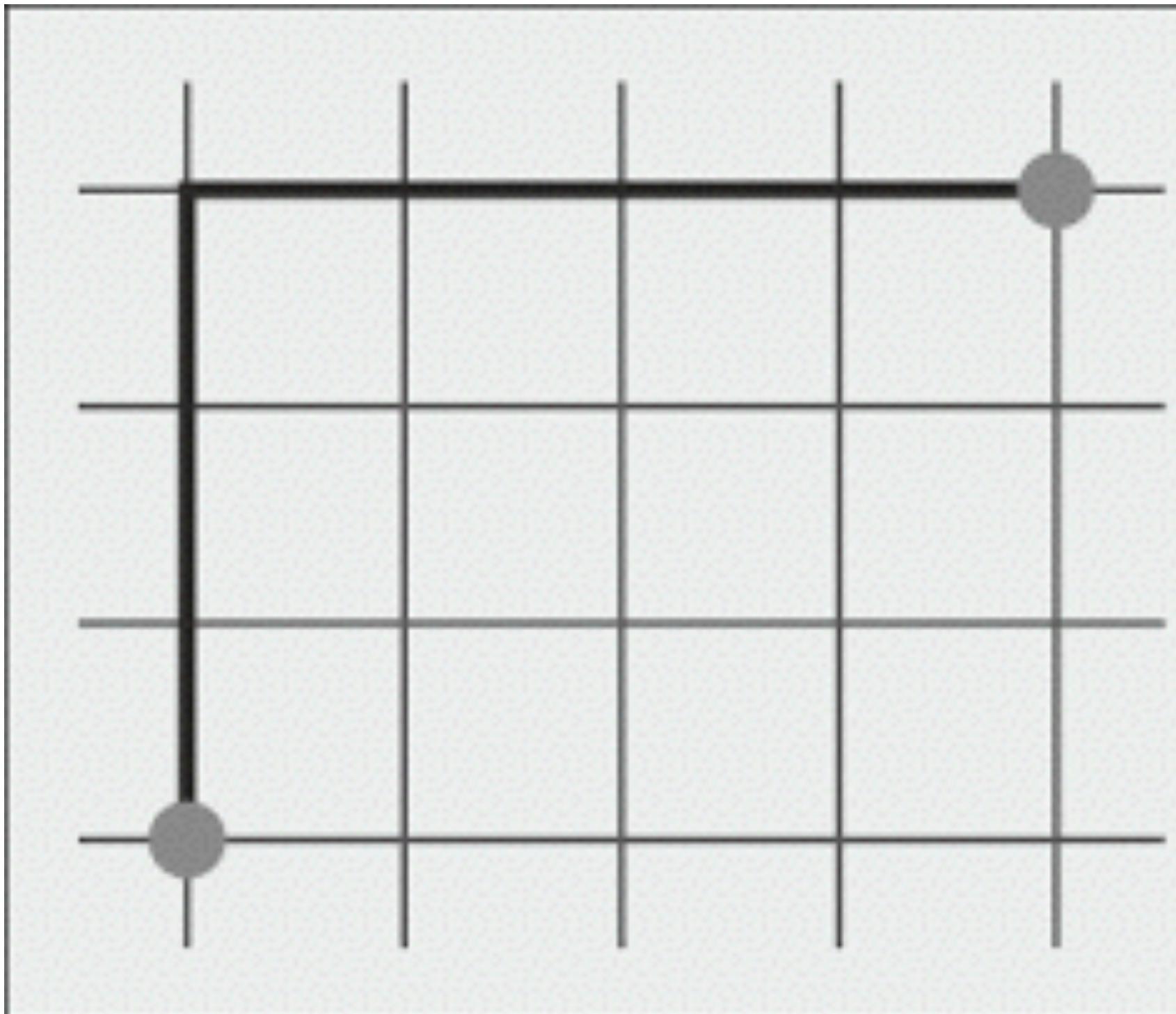
$$\mathbf{q} = (x_2, y_2)$$



Manhattan distance

Also called: taxicab, L_1

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n |p_i - q_i|$$



Chebyshev distance

Also called: maximum, chessboard, L_∞

$$d(\mathbf{p}, \mathbf{q}) = \max_i(|p_i - q_i|)$$

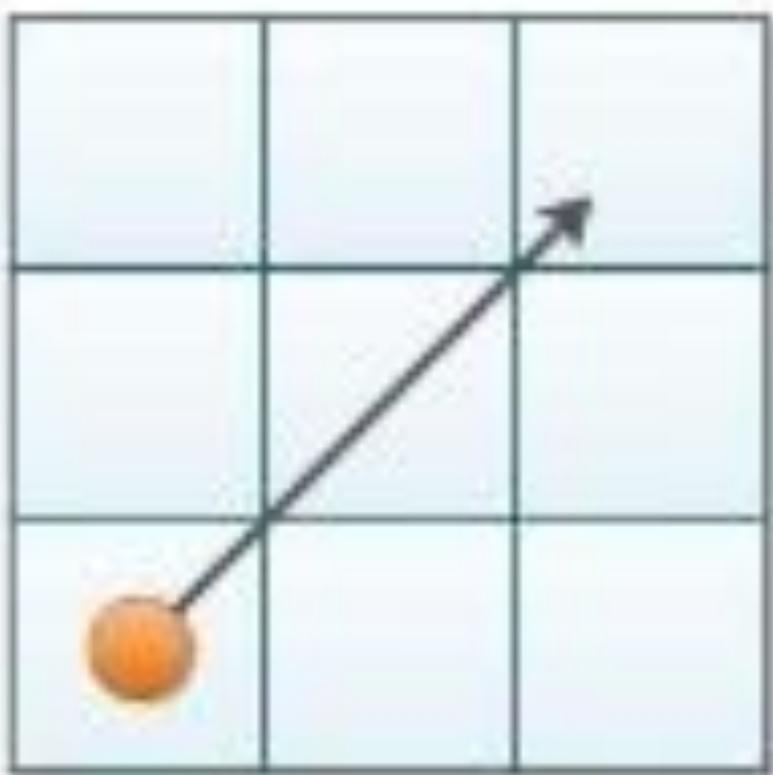
	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1	1	1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

The Minkowski distance generalises these distances

$$d(\mathbf{p}, \mathbf{q}) = \left(\sum_{i=1}^n |p_i - q_i|^r \right)^{1/r}$$

$$r = 2$$

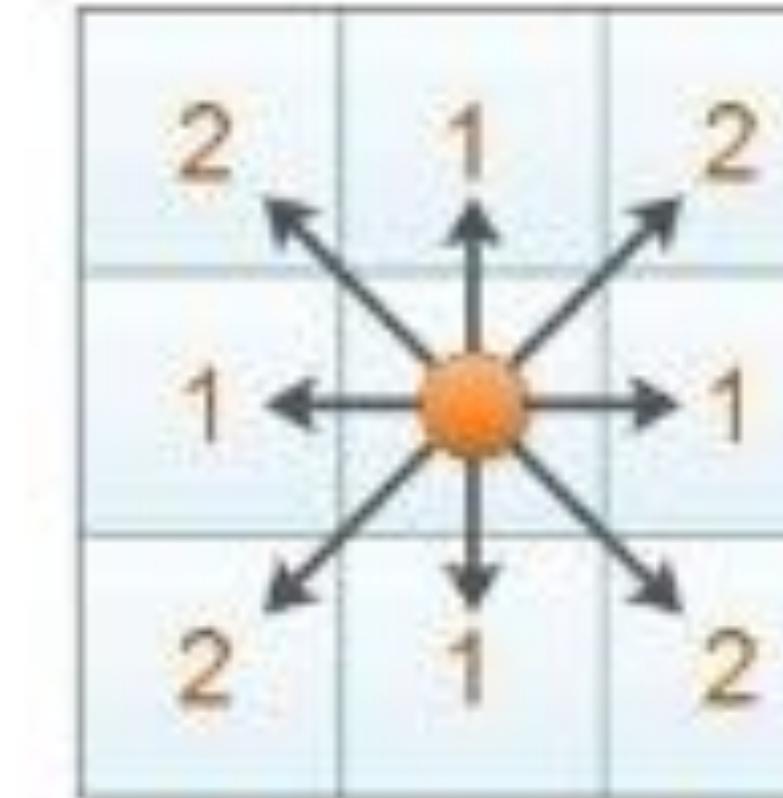
Euclidean Distance



$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$r = 1$$

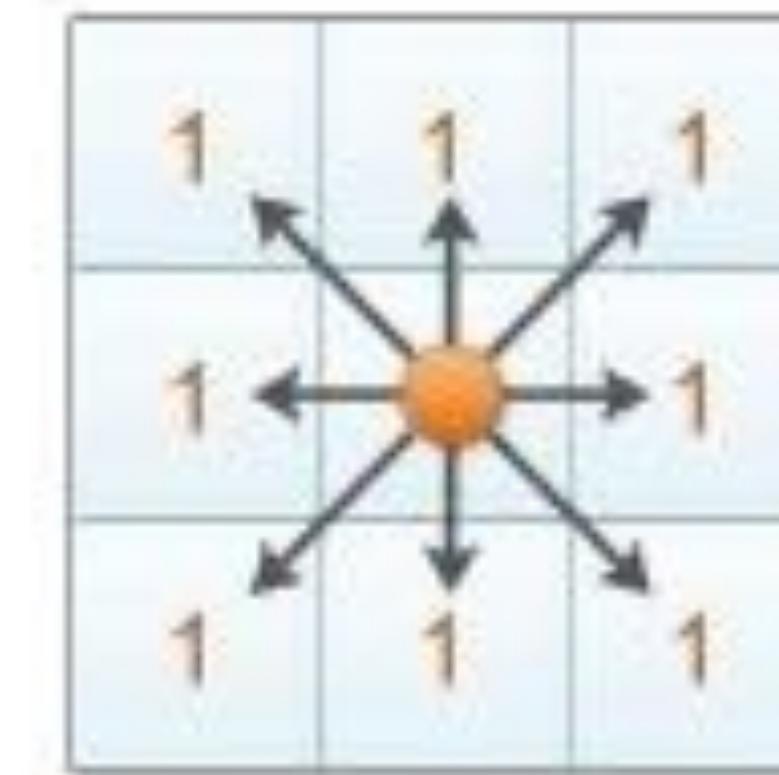
Manhattan Distance



$$|x_1 - x_2| + |y_1 - y_2|$$

$$r = \infty$$

Chebyshev Distance



$$\max(|x_1 - x_2|, |y_1 - y_2|)$$

Hamming distance

Also called: edit

$$d(\mathbf{karolin}, \mathbf{kerstin}) =$$

Hamming distance

Also called: edit

$$d(\text{karolin}, \text{kerstin}) = 3$$

Hamming distance

Also called: edit

$$d(\mathbf{karolin}, \mathbf{kerstin}) = 3$$

$$\begin{aligned}\mathbf{x} &= (1, 0, 0, 0, 0, 0, 0, 0, 0) \\ \mathbf{y} &= (0, 0, 0, 0, 0, 0, 1, 0, 1)\end{aligned}$$

$$d(\mathbf{x}, \mathbf{y}) = 3$$

Hamming distance

Also called: edit

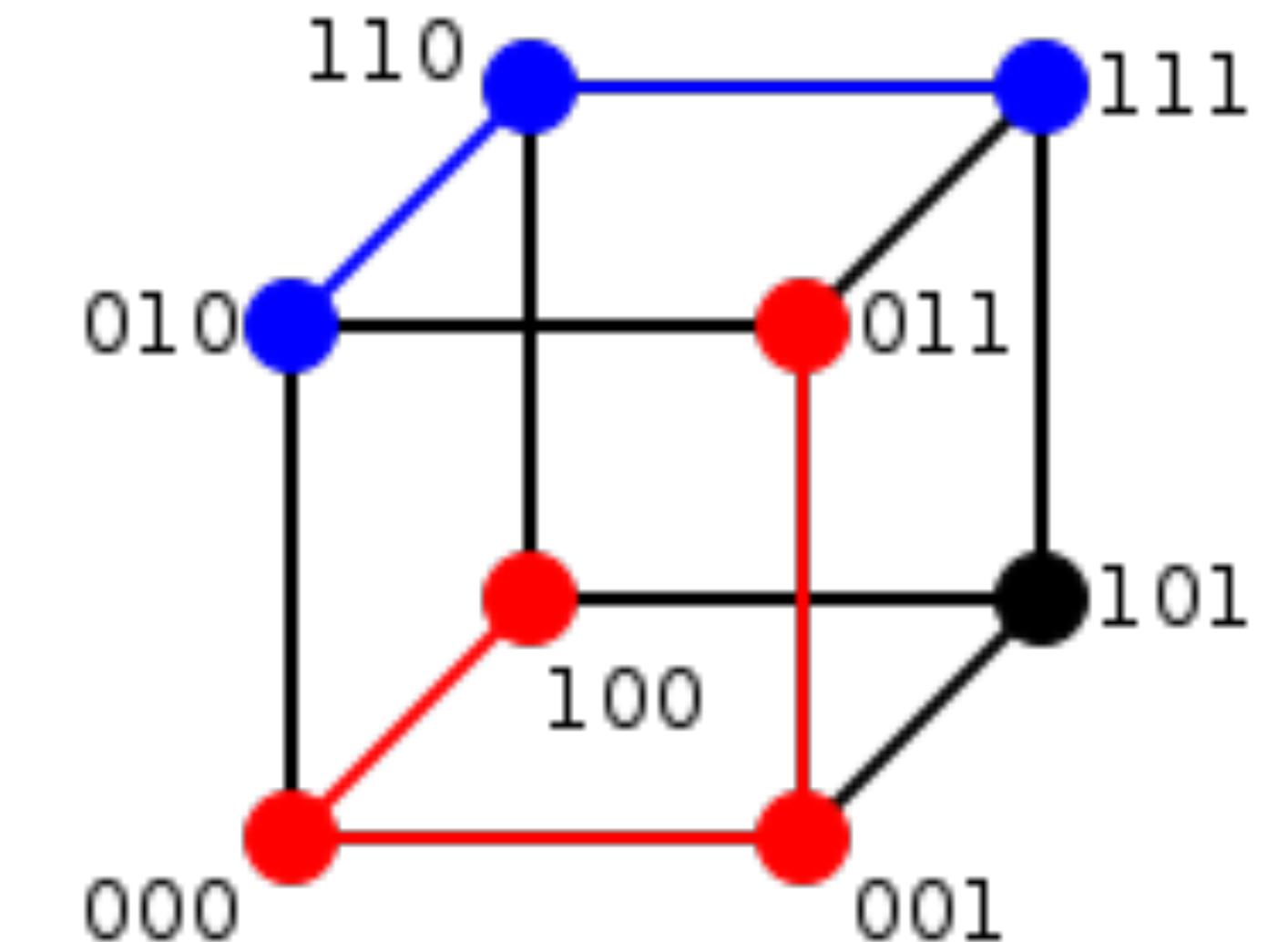
$$d(\text{karolin}, \text{kerstin}) = 3$$

$$\begin{aligned}\mathbf{x} &= (1, 0, 0, 0, 0, 0, 0, 0, 0) \\ \mathbf{y} &= (0, 0, 0, 0, 0, 0, 1, 0, 1)\end{aligned}$$

$$d(\mathbf{x}, \mathbf{y}) = 3$$

For binary vectors this is the Manhattan distance:

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n |p_i - q_i|$$



Euclidian distance between animals

Blackboard

```
rattlesnake = [1,1,1,1,0]
```

```
boa = [0,1,0,1,0]
```

```
frog = [1,0,1,0,4]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

	rattle snake	boa	frog
rattle snake	0	?	
boa	?	0	
frog			0

Euclidian distance between animals

```
rattlesnake = [1,1,1,1,0]
```

```
boa = [0,1,0,1,0]
```

```
frog = [1,0,1,0,4]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

	rattle snake	boa	frog
rattle snake	0	1.41	
boa	1.41	0	
frog			0

Euclidian distance between animals

```
rattlesnake = [1,1,1,1,0]
```

```
boa = [0,1,0,1,0]
```

```
frog = [1,0,1,0,4]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

	rattle snake	boa	frog
rattle snake	0	1.41	4.24
boa	1.41	0	4.47
frog	4.24	4.47	0

Euclidian distance between animals

```
rattlesnake = [1,1,1,1,0]
```

```
boa = [0,1,0,1,0]
```

```
frog = [1,0,1,0,4]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

	rattle snake	boa	frog
rattle snake	0	1.41	4.24
boa	1.41	0	4.47
frog	4.24	4.47	0

Rattlesnake and boa are closer to each other than to the frog. Nice.

Euclidian distance between animals

```
rattlesnake = [1,1,1,1,0]  
boa = [0,1,0,1,0]  
frog = [1,0,1,0,4]  
alligator = [1,1,0,1,4]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

	rattle snake	boa	frog	alligator
rattle snake	0	1.41	4.24	4.12
boa	1.41	0	4.47	4.12
frog	4.24	4.47	0	1.73
alligator	4.12	4.12	1.73	0

Euclidian distance between animals

```
rattlesnake = [1,1,1,1,0]  
boa = [0,1,0,1,0]  
frog = [1,0,1,0,4]  
alligator = [1,1,0,1,4]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

The alligator is closer to frog than to the snakes. Why?

	rattle snake	boa	frog	alligator
rattle snake	0	1.41	4.24	4.12
boa	1.41	0	4.47	4.12
frog	4.24	4.47	0	1.73
alligator	4.12	4.12	1.73	0

Euclidian distance between animals

```
rattlesnake = [1,1,1,1,0]  
boa = [0,1,0,1,0]  
frog = [1,0,1,0,4]  
alligator = [1,1,0,1,4]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

The alligator is closer to frog than to the snakes. Why?

Number of legs dominate instead of matching features!

	rattle snake	boa	frog	alligator
rattle snake	0	1.41	4.24	4.12
boa	1.41	0	4.47	4.12
frog	4.24	4.47	0	1.73
alligator	4.12	4.12	1.73	0

Euclidian distance between animals

```
rattlesnake = [1,1,1,1,0]  
boa = [0,1,0,1,0]  
frog = [1,0,1,0,1]  
alligator = [1,1,0,1,1]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Solution: Binarize legs

	rattle snake	boa	frog	alligator
rattle snake	0	1.41	1.73	1.41
boa	1.41	0	2.24	1.41
frog	1.73	2.24	0	1.73
alligator	1.41	1.41	1.73	0

Euclidian distance between animals

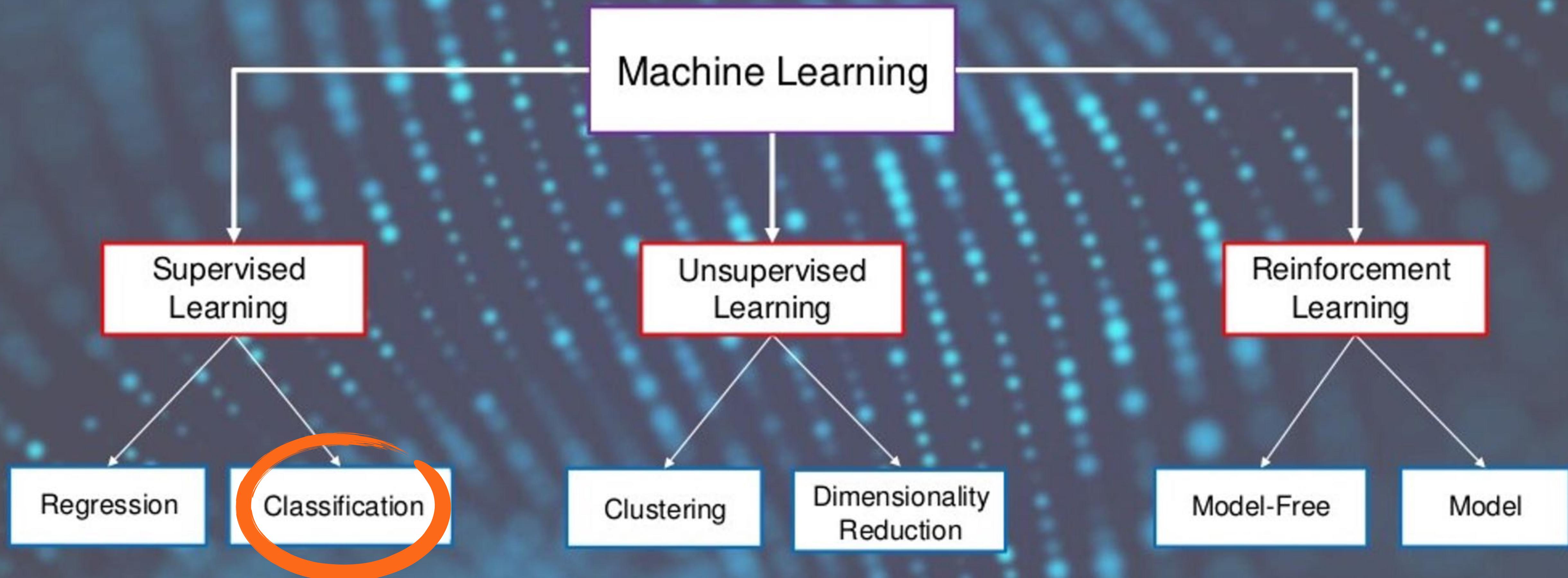
```
rattlesnake = [1,1,1,1,0]  
boa = [0,1,0,1,0]  
frog = [1,0,1,0,1]  
alligator = [1,1,0,1,1]
```

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

	rattle snake	boa	frog	alligator
rattle snake	0	1.41	1.73	1.41
boa	1.41	0	2.24	1.41
frog	1.73	2.24	0	1.73
alligator	1.41	1.41	1.73	0

The alligator is now closer to the snakes than to the frog. Nice.

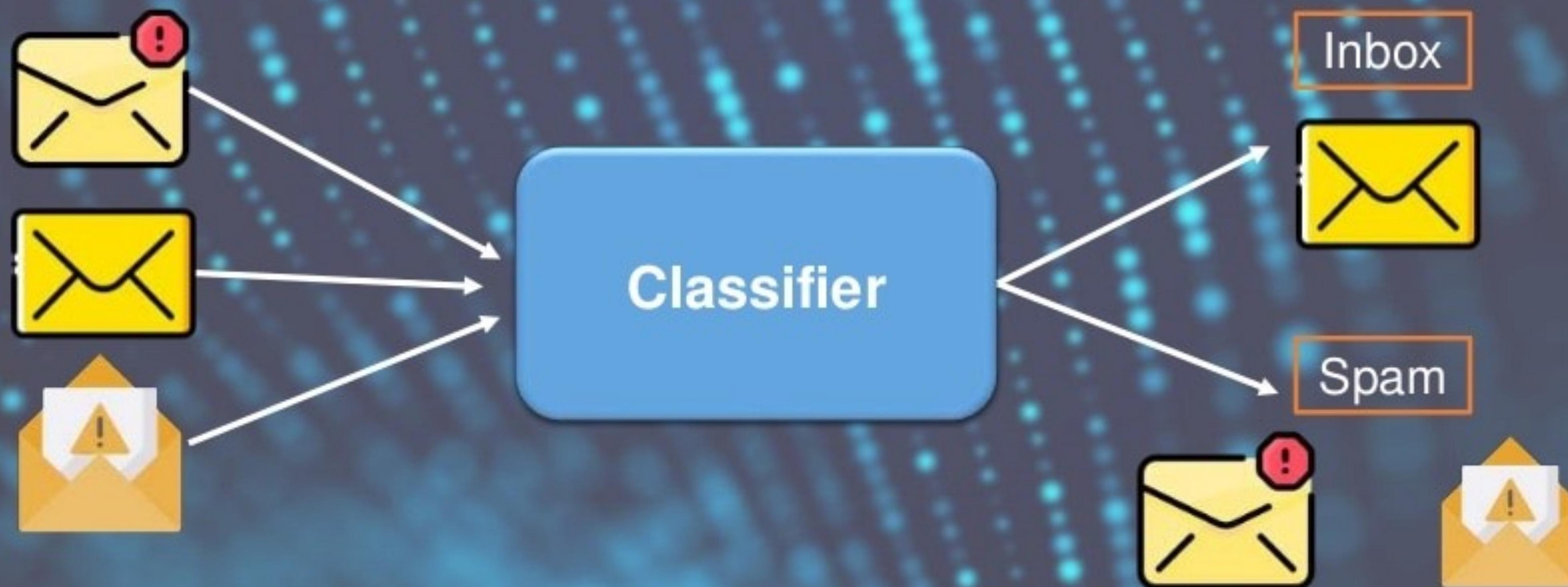
Type of Machine Learning



What is Classification?

Classification is the process of classifying data into different categories based on some of their common characteristics

Example: A system to **classify spam emails**



Need for Confusion Matrices

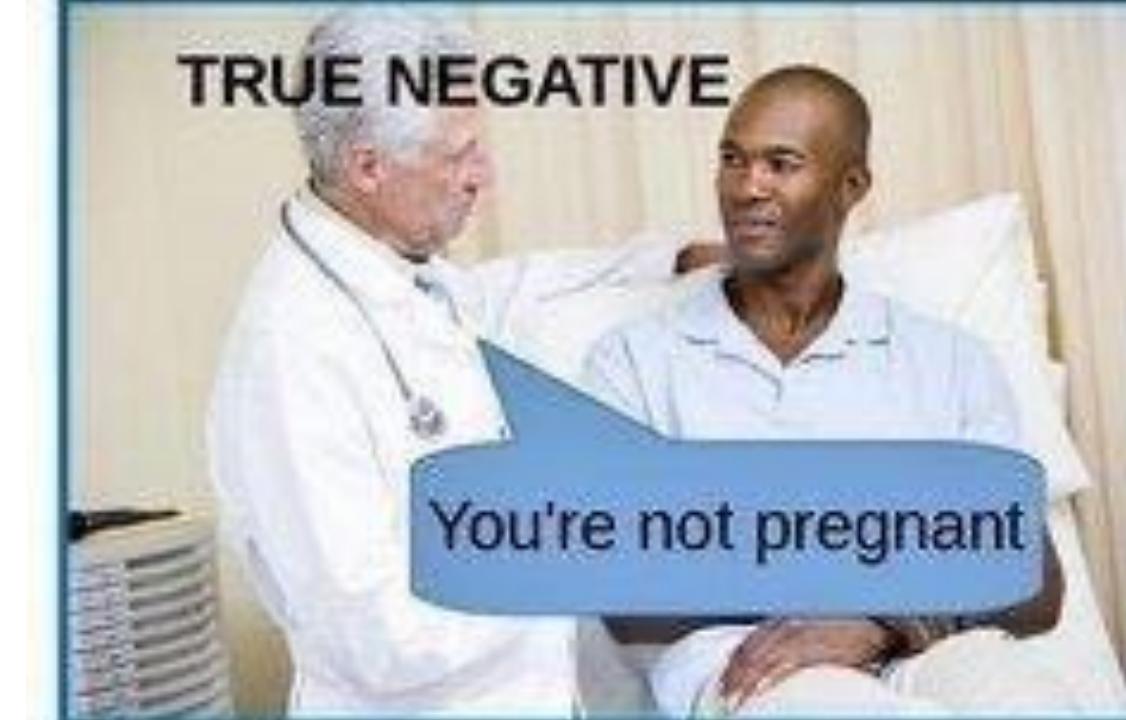
During classification, we also have to overcome the limitations of accuracy. Accuracy can be misleading for classification problems. If there is a significant class imbalance, a model might predict the majority class for all cases and have a high accuracy score



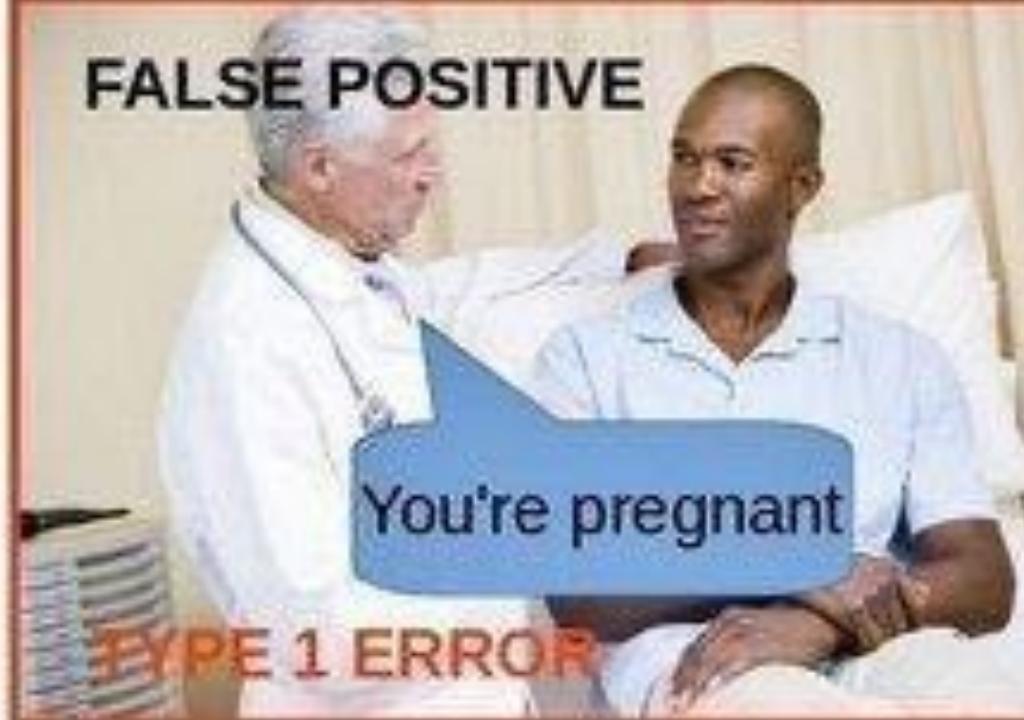
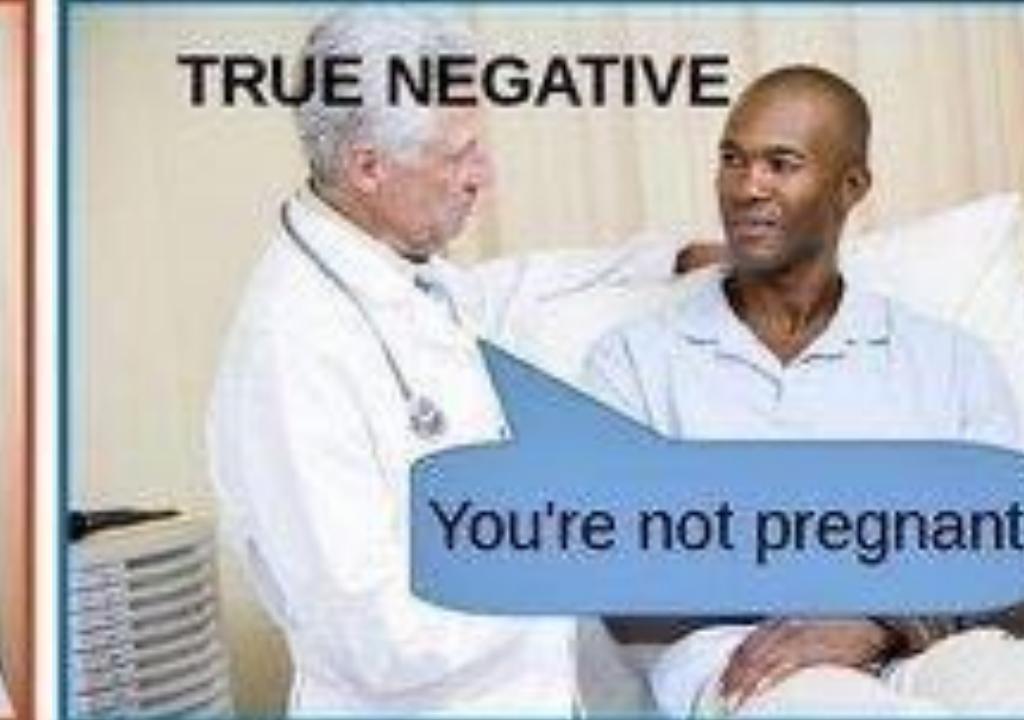
The confusion matrix shows results from binary classification

		Predicted	
		Positive	Negative
Actual	Positive	 A photograph of a pregnant woman sitting in a chair, facing a female doctor who is smiling and pointing towards her belly. A blue speech bubble overlay on the image contains the text "You're pregnant". Above the photo, the words "TRUE POSITIVE" are written in bold capital letters.	
	Negative		

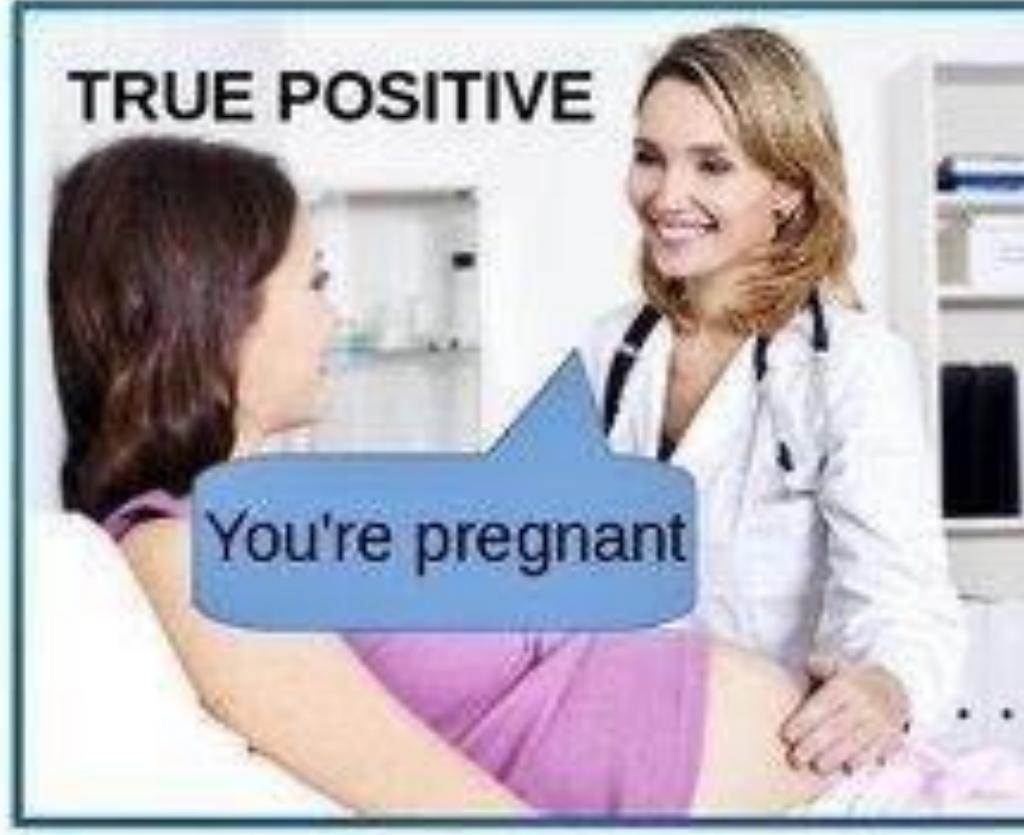
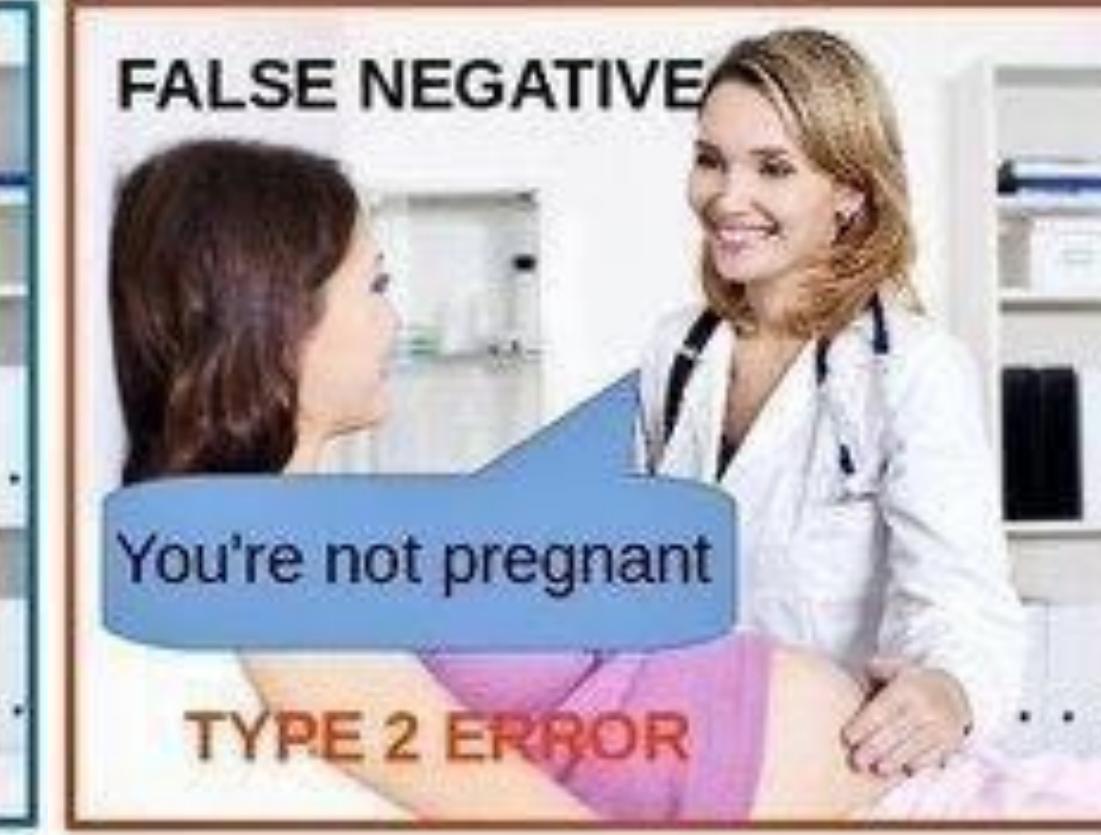
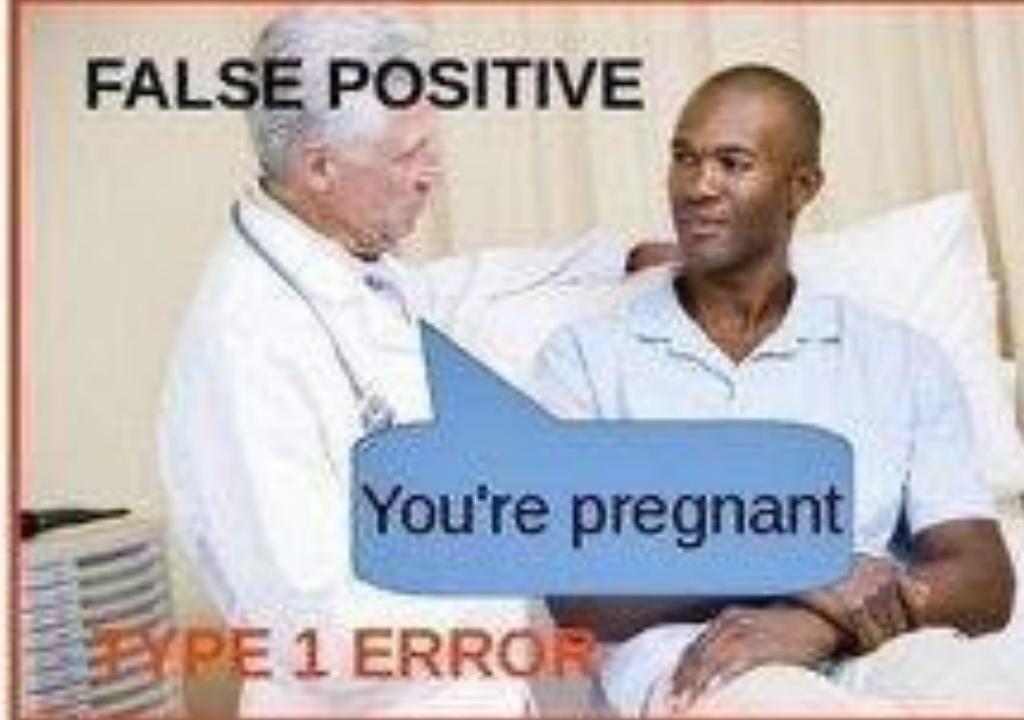
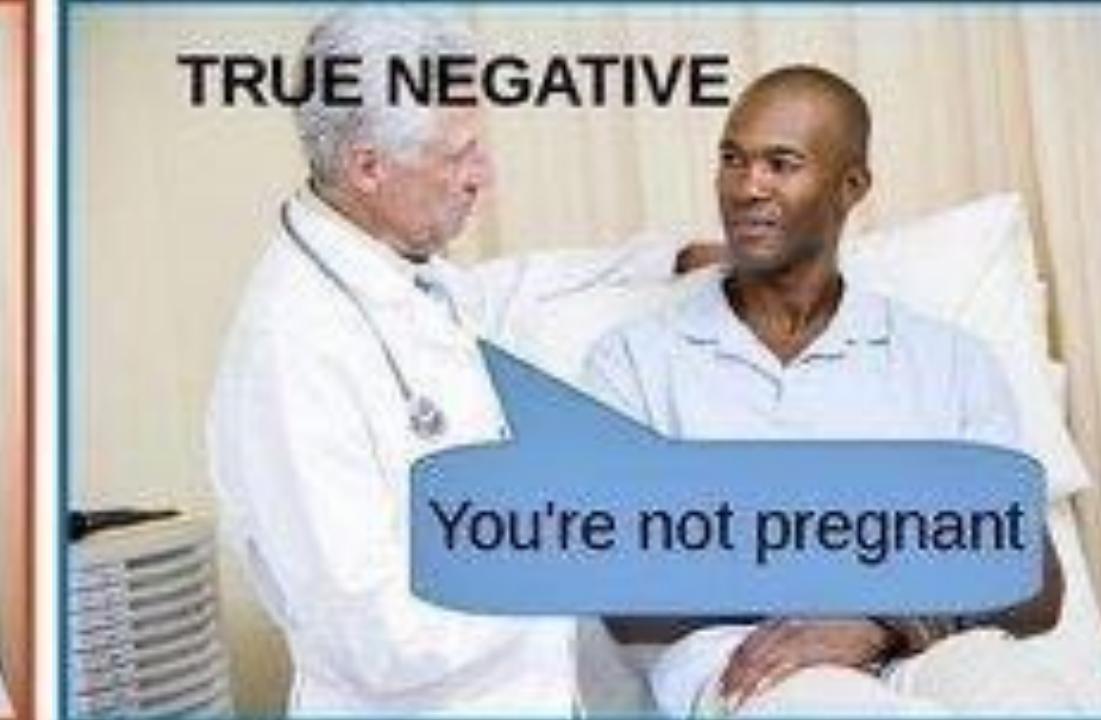
The confusion matrix shows results from binary classification

		Predicted	
		Positive	Negative
Actual	Positive	 A photograph of a pregnant woman in a white top and pink pants sitting in a chair. A female doctor in a white coat and stethoscope around her neck is smiling and pointing towards the woman's belly. A blue speech bubble in the foreground says "You're pregnant". The text "TRUE POSITIVE" is displayed at the top left of the photo.	
	Negative	 A photograph of a man in a white shirt sitting in a chair. An older male doctor in a white coat and stethoscope around his neck is smiling and pointing towards the man's belly. A blue speech bubble in the foreground says "You're not pregnant". The text "TRUE NEGATIVE" is displayed at the top left of the photo.	

The confusion matrix shows results from binary classification

		Predicted	
		Positive	Negative
Actual	Positive		
	Negative		

The confusion matrix shows results from binary classification

		Predicted	
		Positive	Negative
Actual	Positive		
	Negative		

The confusion matrix shows results from binary classification

		Predicted	
		Positive	Negative
Actual	Positive	TRUE POSITIVE TP You're pregnant	FALSE NEGATIVE FN You're not pregnant TYPE 2 ERROR
	Negative	FALSE POSITIVE FP You're pregnant TYPE 1 ERROR	TRUE NEGATIVE TN You're not pregnant

Performance metrics help judge model performance

- Accuracy
 - Precision Positive predictive value (PPV)
 - Sensitivity Recall, true positive rate (TPR)
 - Specificity Selectivity, true negative rate (TNR)
-
- and many more...

Accuracy: How often is it right?

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{\begin{array}{c} \text{Green} \\ \text{Grey} \\ \text{Red} \end{array}}{\begin{array}{c} \text{Green} \\ \text{Grey} \\ \text{Red} \end{array}}$$

Precision: How often is it right when predicting positive?

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{\begin{array}{|c|c|}\hline \text{Green} & \text{Grey} \\ \hline \text{Grey} & \text{Green} \\ \hline \end{array}}{\begin{array}{|c|c|}\hline \text{Green} & \text{Red} \\ \hline \text{Red} & \text{Green} \\ \hline \end{array}}$$

$$\text{Precision} = \frac{\text{True positives}}{\text{Predicted positives}} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\begin{array}{|c|}\hline \text{Green} \\ \hline \end{array}}{\begin{array}{|c|c|}\hline \text{Green} & \text{Grey} \\ \hline \text{Red} & \text{Grey} \\ \hline \end{array}}$$

Sensitivity: How often is it right when actually positive?

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{\begin{array}{|c|c|}\hline \text{Green} & \text{Grey} \\ \hline \text{Grey} & \text{Green} \\ \hline\end{array}}{\begin{array}{|c|c|}\hline \text{Green} & \text{Red} \\ \hline \text{Red} & \text{Green} \\ \hline\end{array}}$$

$$\text{Precision} = \frac{\text{True positives}}{\text{Predicted positives}} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\begin{array}{|c|}\hline \text{Green} \\ \hline \text{Grey} \\ \hline\end{array}}{\begin{array}{|c|c|}\hline \text{Green} & \text{Red} \\ \hline \text{Red} & \text{Grey} \\ \hline\end{array}}$$

$$\text{Sensitivity} = \frac{\text{True positives}}{\text{Actual positives}} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\begin{array}{|c|}\hline \text{Green} \\ \hline \text{Grey} \\ \hline\end{array}}{\begin{array}{|c|c|}\hline \text{Green} & \text{Red} \\ \hline \text{Red} & \text{Grey} \\ \hline\end{array}}$$

Specificity: How often is it right when actually negative?

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{\begin{array}{|c|c|}\hline \text{Green} & \text{Grey} \\ \hline \text{Grey} & \text{Green} \\ \hline\end{array}}{\begin{array}{|c|c|}\hline \text{Green} & \text{Red} \\ \hline \text{Red} & \text{Green} \\ \hline\end{array}}$$

$$\text{Precision} = \frac{\text{True positives}}{\text{Predicted positives}} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\begin{array}{|c|}\hline \text{Green} \\ \hline \text{Grey} \\ \hline\end{array}}{\begin{array}{|c|c|}\hline \text{Green} & \text{Red} \\ \hline \text{Red} & \text{Grey} \\ \hline\end{array}}$$

$$\text{Sensitivity} = \frac{\text{True positives}}{\text{Actual positives}} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\begin{array}{|c|}\hline \text{Green} \\ \hline \text{Grey} \\ \hline\end{array}}{\begin{array}{|c|c|}\hline \text{Green} & \text{Red} \\ \hline \text{Red} & \text{Grey} \\ \hline\end{array}}$$

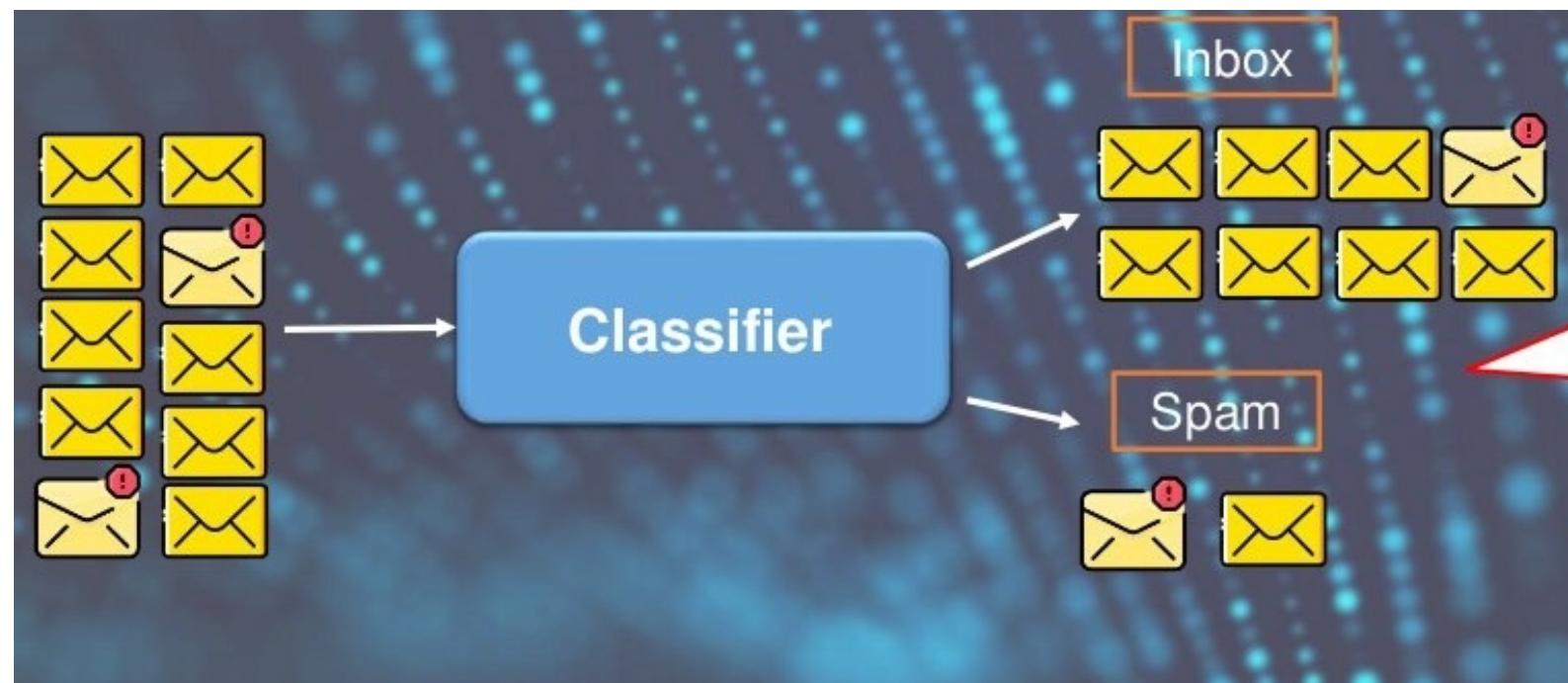
$$\text{Specificity} = \frac{\text{True negatives}}{\text{Actual negatives}} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\begin{array}{|c|}\hline \text{Grey} \\ \hline \text{Green} \\ \hline\end{array}}{\begin{array}{|c|c|}\hline \text{Red} & \text{Green} \\ \hline \text{Grey} & \text{Green} \\ \hline\end{array}}$$

Example: Spam filter

Blackboard



Example: Spam filter



$$\text{Accuracy} = \frac{8}{10} = 80\%$$

		Predicted	
		Positive	Negative
Actual	Positive	1	1
	Negative	1	7

$$\text{Precision} = \frac{1}{2} = 50\%$$

$$\text{Sensitivity} = \frac{1}{2} = 50\%$$

$$\text{Specificity} = \frac{7}{8} = 87.5\%$$

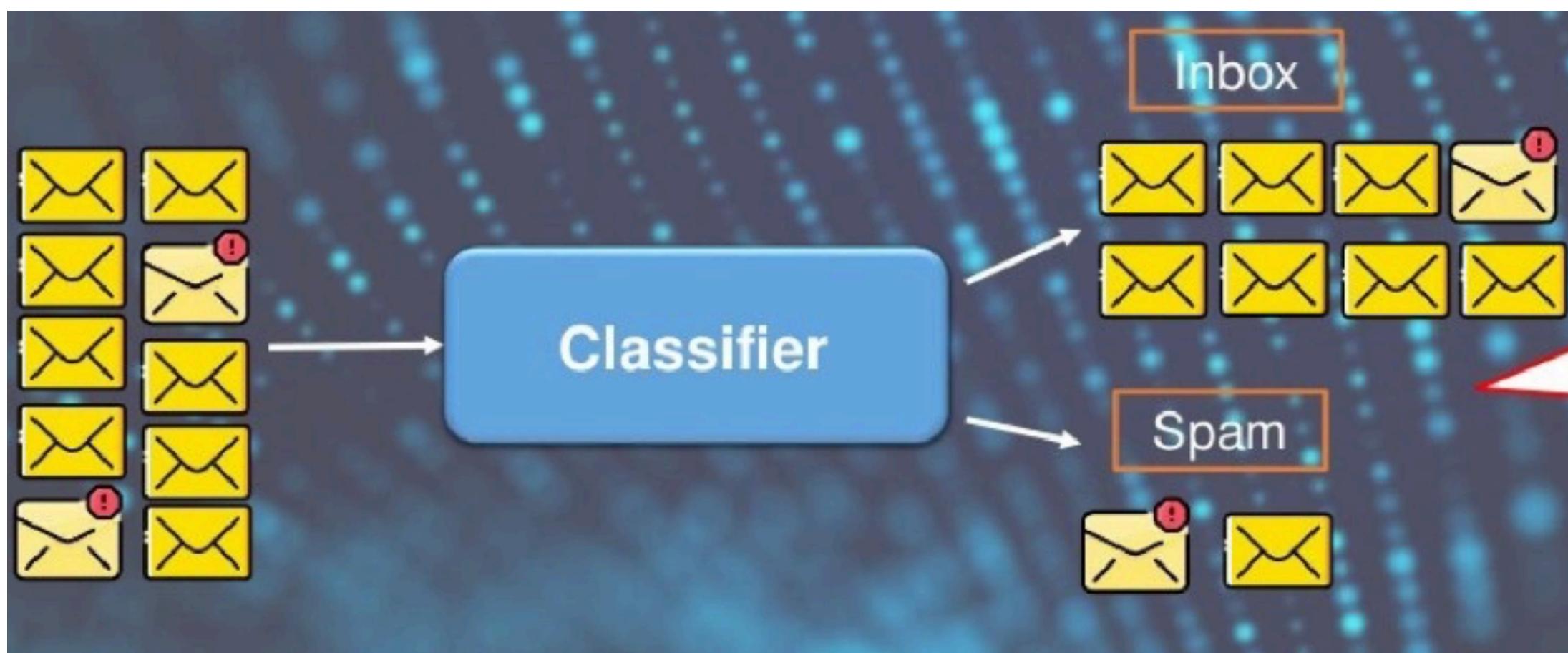
There is a trade-off between Sensitivity and Specificity

Example: Spam filter

What do we want?

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$



There is a trade-off between Sensitivity and Specificity

Example: Spam filter

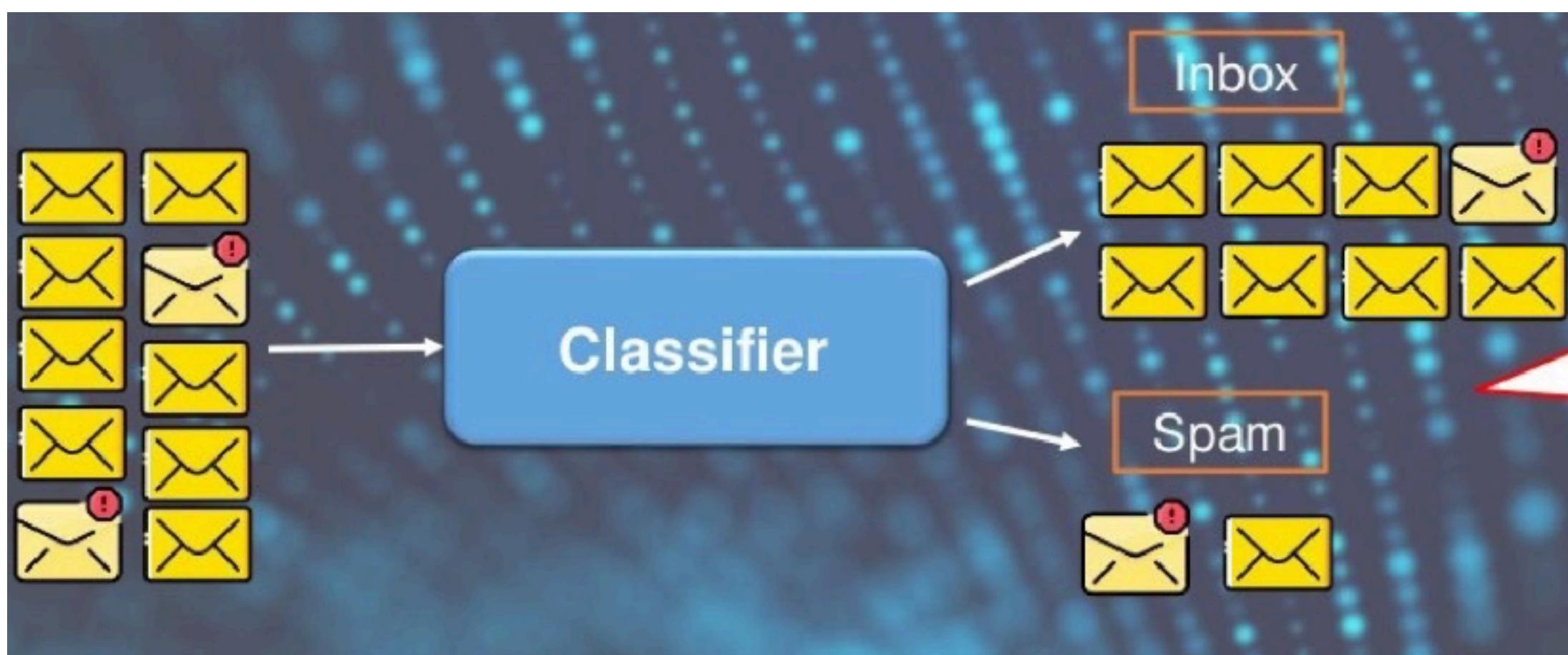
What do we want?

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

Low: It's OK if some spam slips into our inbox

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$

High: Legitimate email should never land in spam



There is a trade-off between Sensitivity and Specificity

Example: Spam filter

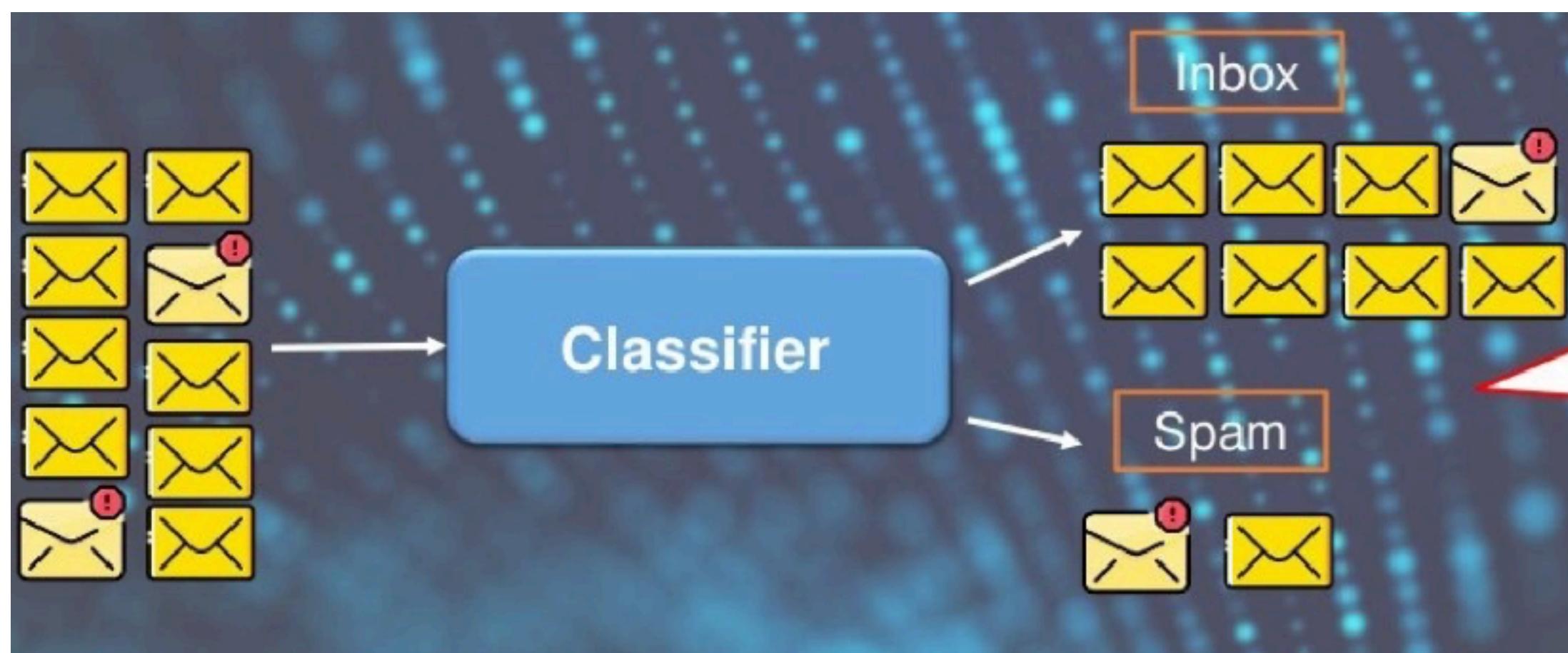
What do we want?

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

Low: It's OK if some spam slips into our inbox

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$

High: Legitimate email should never land in spam



Here important: Precision

If we predict spam,
it should be spam!

There is a trade-off between Sensitivity and Specificity

Example: Security scanner

What do we want?

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$



There is a trade-off between Sensitivity and Specificity

Example: Security scanner

What do we want?

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

High: Want to catch all terrorists

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$

Low: OK if some false alarms
(but not too many..)



There is a trade-off between Sensitivity and Specificity

Example: Pregnancy test

What do we want?

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$



There is a trade-off between Sensitivity and Specificity

Example: Pregnancy test

What do we want?

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

High: Do not want to miss a pregnancy

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$

High: Do not want to make non-pregnant women go to doctor



There is a trade-off between Sensitivity and Specificity

Example: Pregnancy test

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

What do we want?

High: Do not want to miss a pregnancy

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$

High: Do not want to make non-pregnant women go to doctor



Difficult in practice:

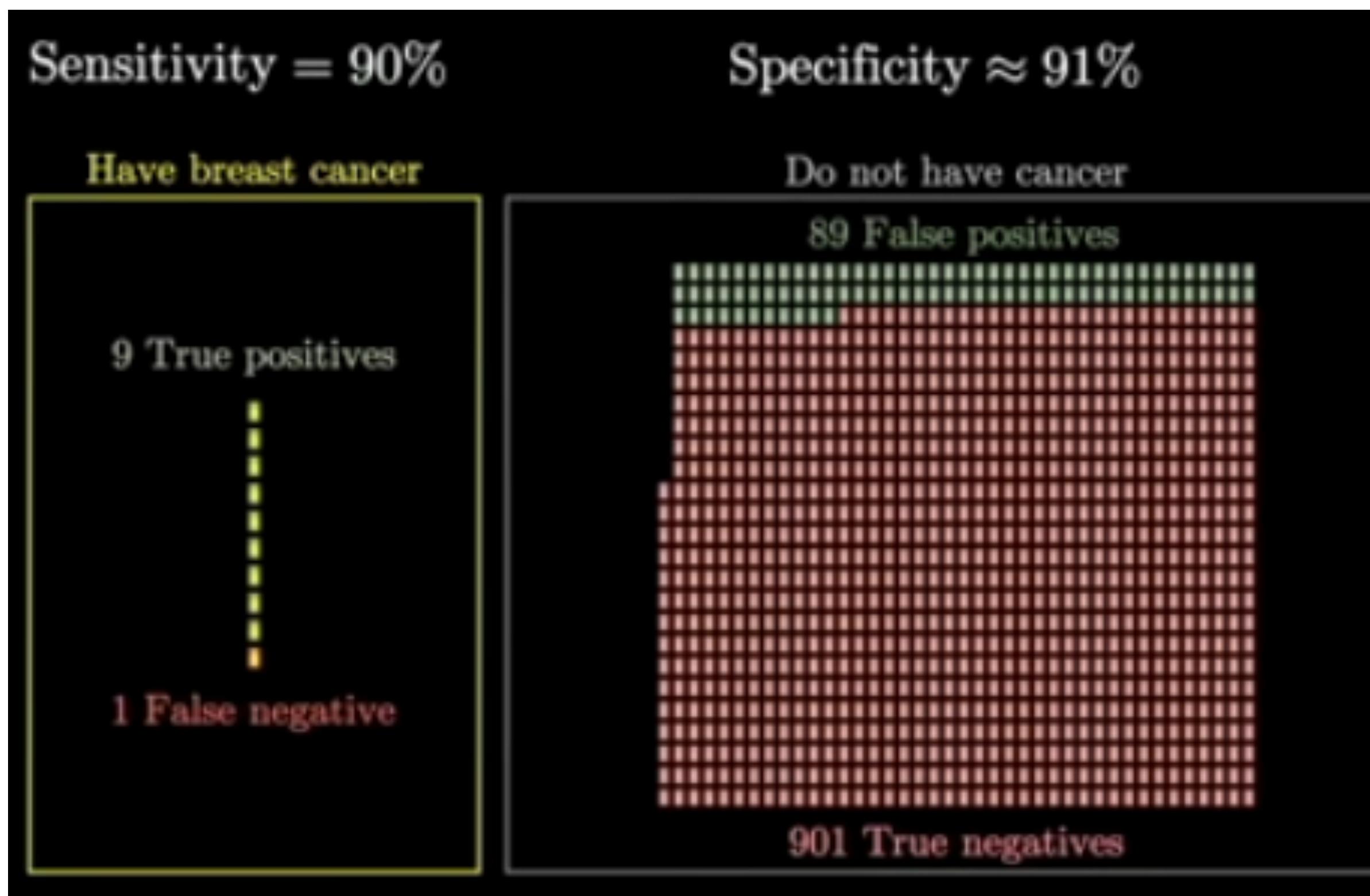
Clinical performance of hCG pregnancy test kits with 99 urine and 99 serum samples							
Test	Sensitivity	Specificity	False +	False -	True +	True -	Accuracy
Urine tests							
Tandem	90.6	100	0	8	77	14	91.9
NovoClone	70.6	92.9	1	25	60	13	73.7
Abbott	84.7	100	0	13	72	14	86.9
Serum tests							
Tandem	89.4	85.7	2	9	76	12	88.9
NovoClone	67.1	100	0	28	57	14	71.7
Abbott	94.1	78.6	3	5	80	11	91.9

Alfthan, Björkes, Tiitinen, Stenman. Specificity and detection limit of ten pregnancy tests, Scan J clin lab inv 216: 105-13 (1993)

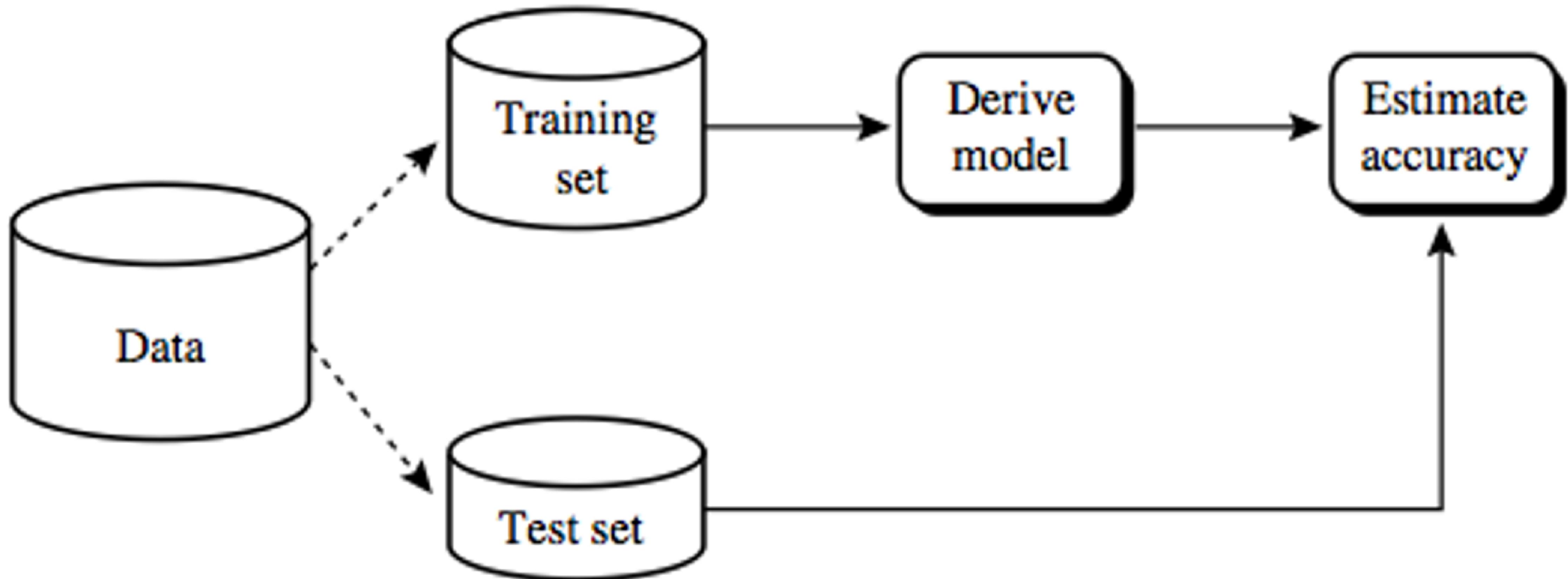
"Paradoxes" arise in certain situations: Medical test paradox

Low prevalence: 1 in 100 have breast cancer

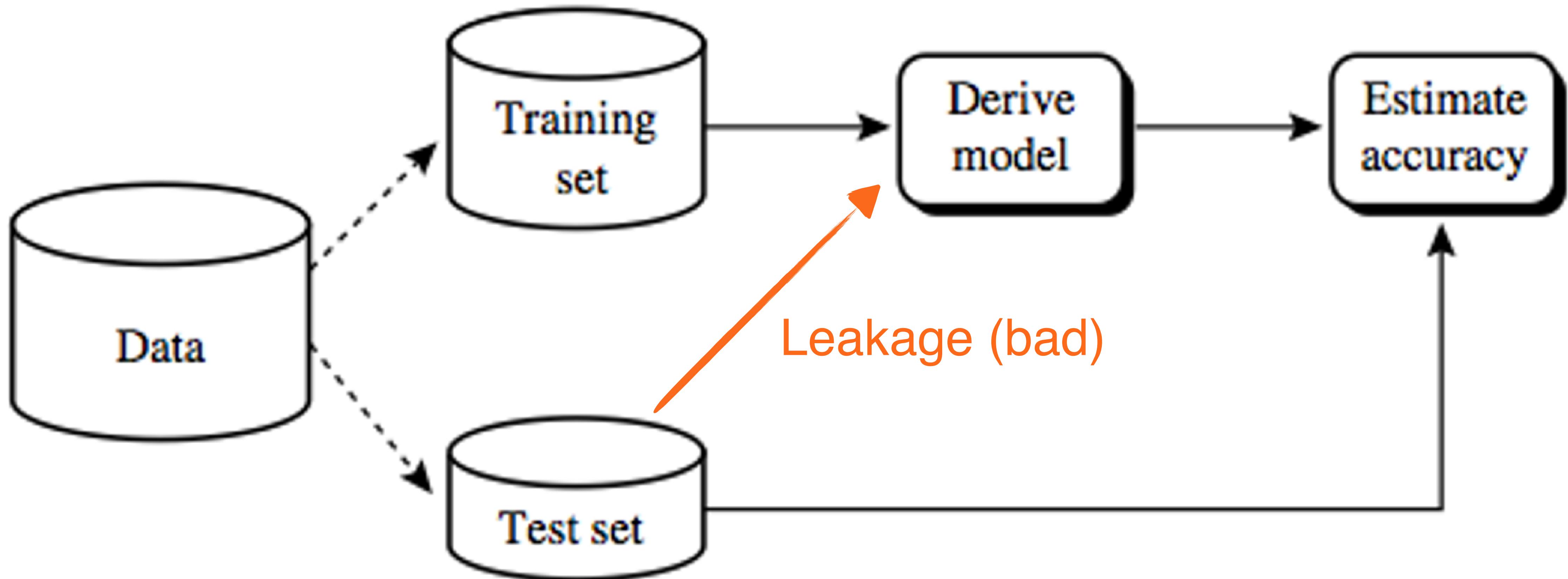
Despite a test giving correct results to >90% of patients, getting a positive result is unlikely (9%) to imply breast cancer



To train and check a model we split up data into training and test



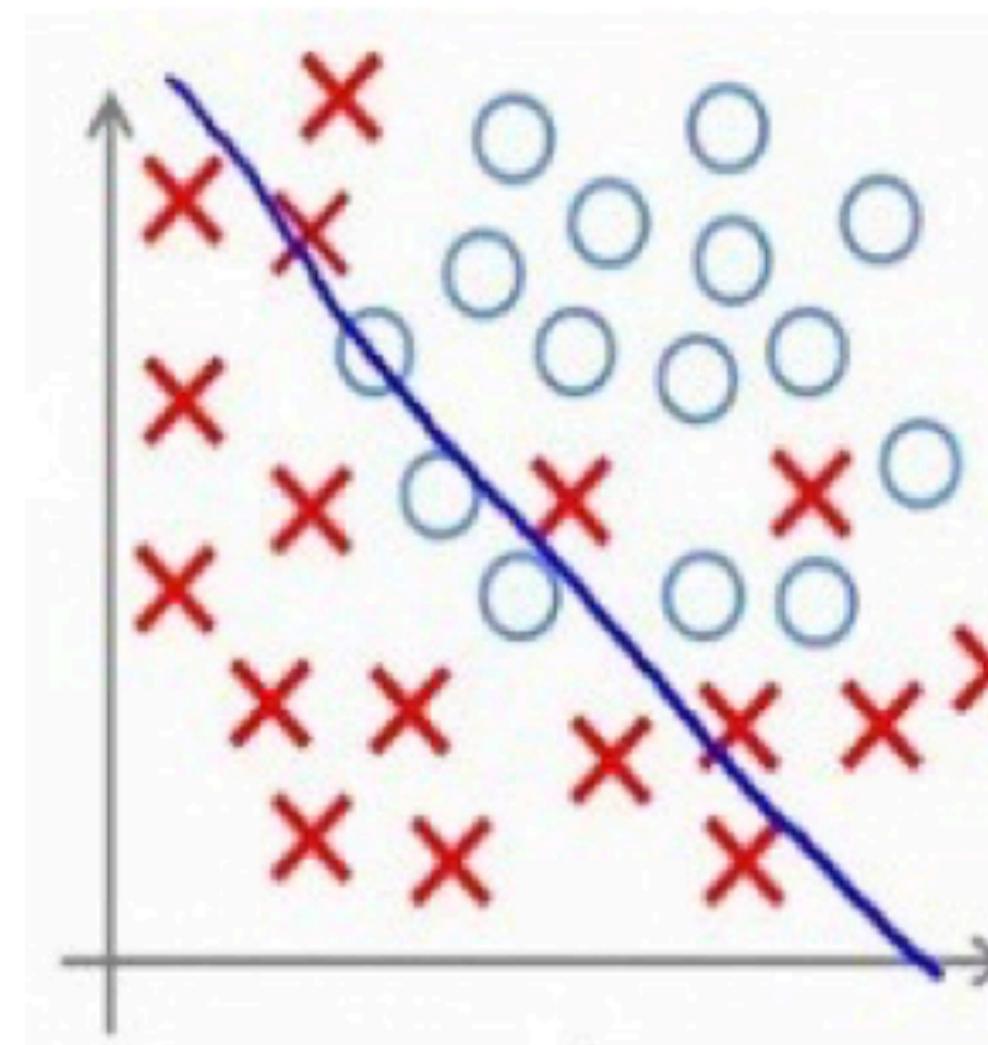
To train and check a model we split up data into training and test



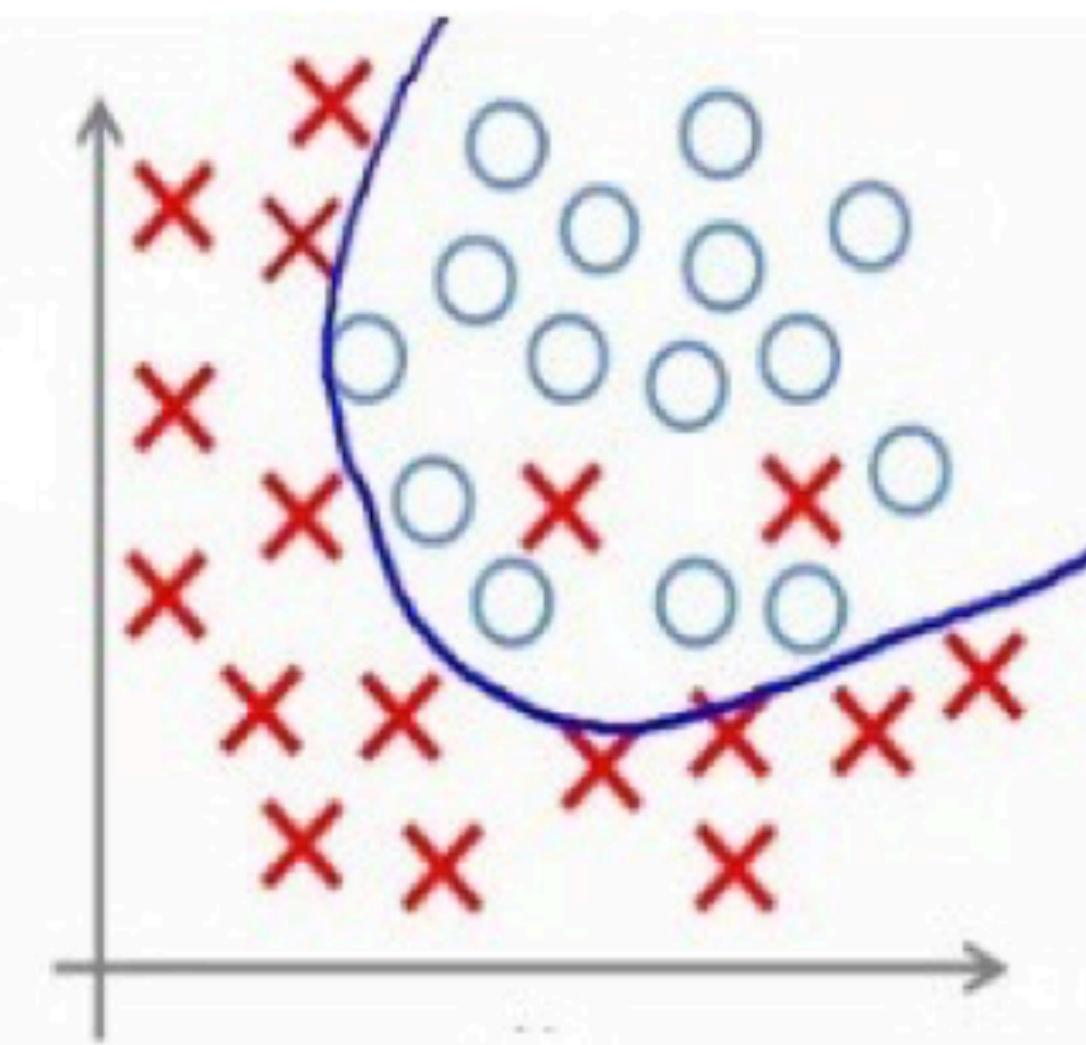
There is a trade-off between underfitting and overfitting

Underfitting: The model does not perform well for both training and testing data

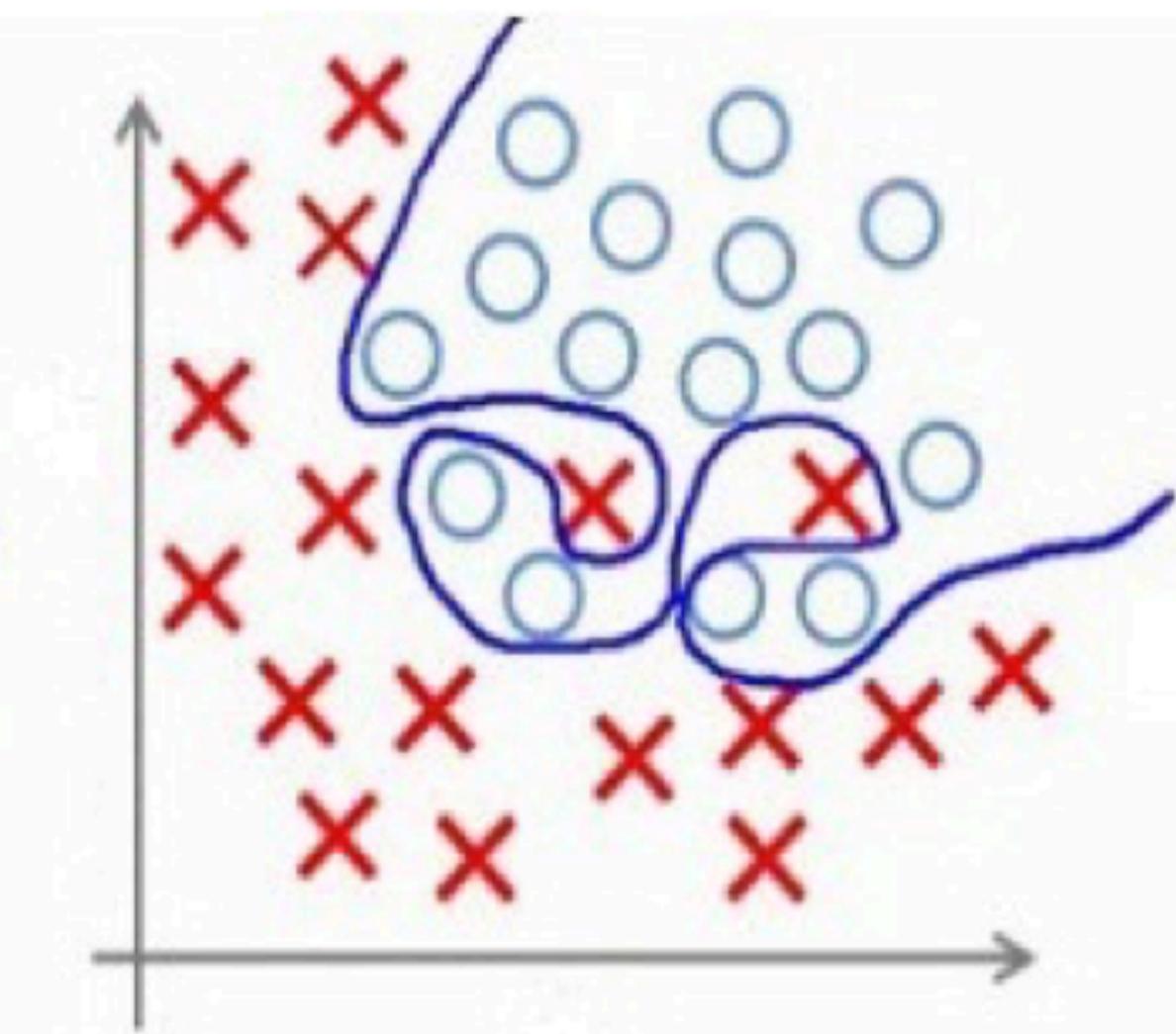
Overfitting: The model performs well for training, but not for testing data



Underfitting

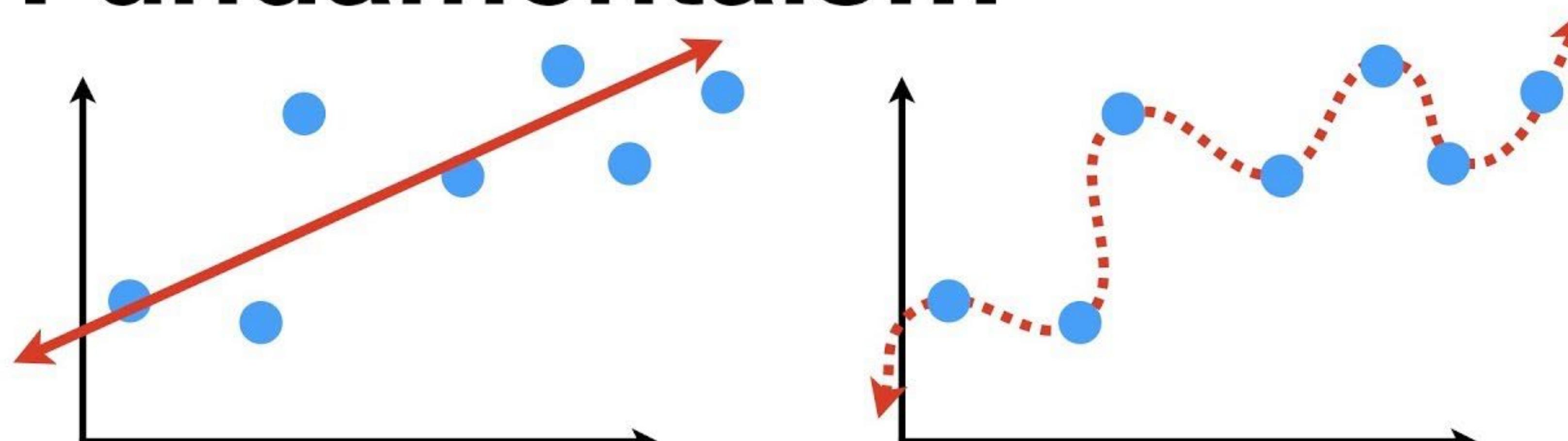


Overfitting



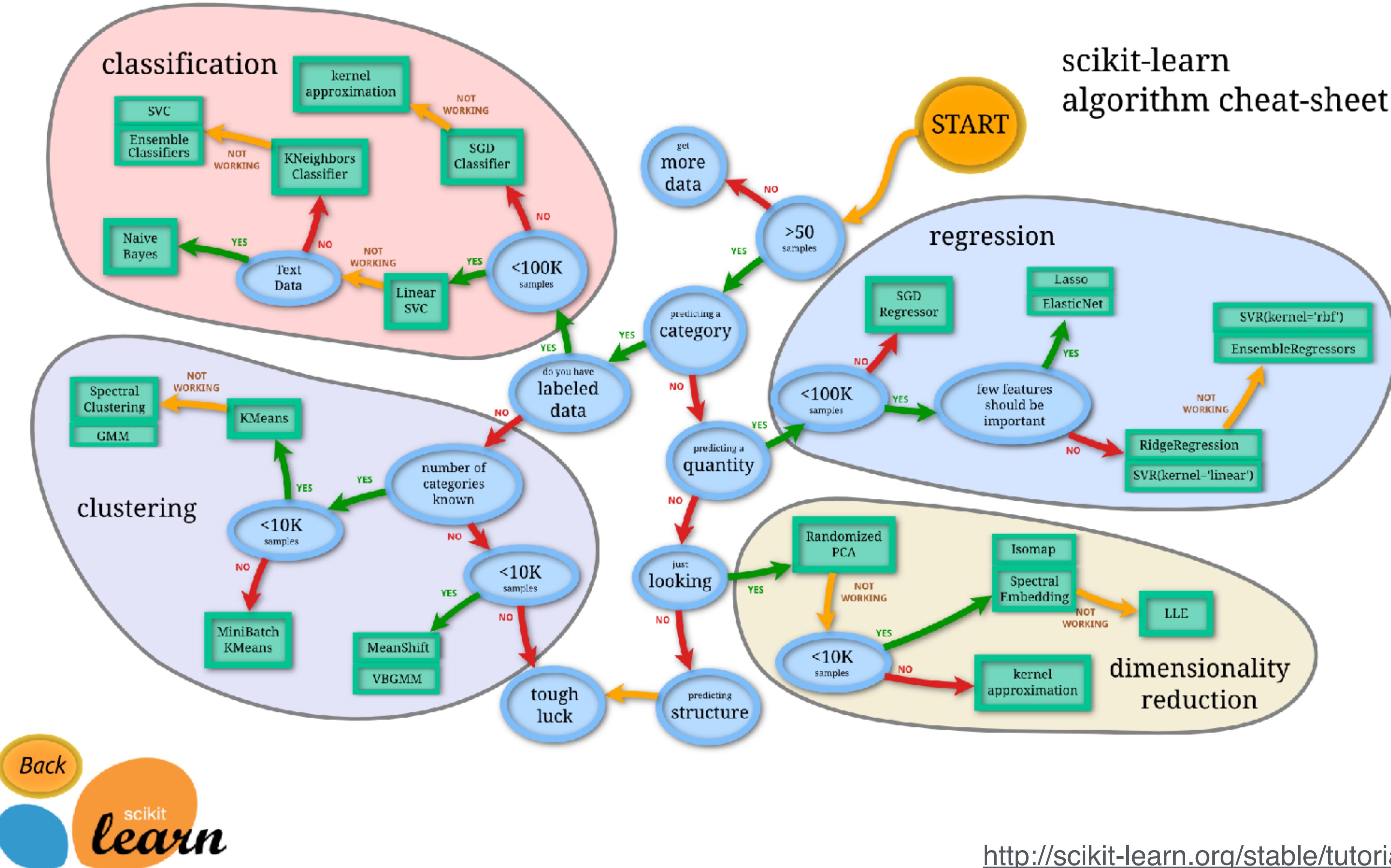
Jupyter

Machine Learning Fundamentals...



...Bias and Variance!!!

scikit-learn is the standard Python library for ML



Take home messages

Machine learning tries to find patterns or rules in data

Choice of features and distance measure influences results

Performance metrics, like the confusion matrix, help evaluate models