

Machine learning: Model optimization and validation

Katarzyna Michalowska, SINTEF
Digital



Outline

1. Motivation
2. Data splitting
3. Data leakage
4. Hyperparameter tuning
5. Demo
6. Evaluation

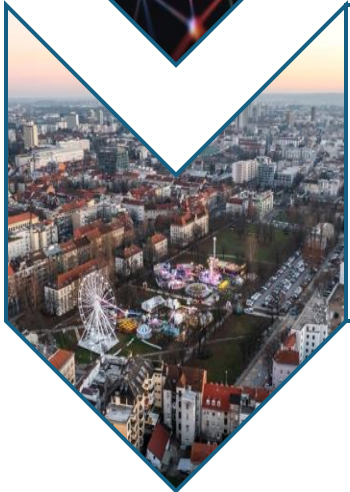
Motivation



The central goal of ML



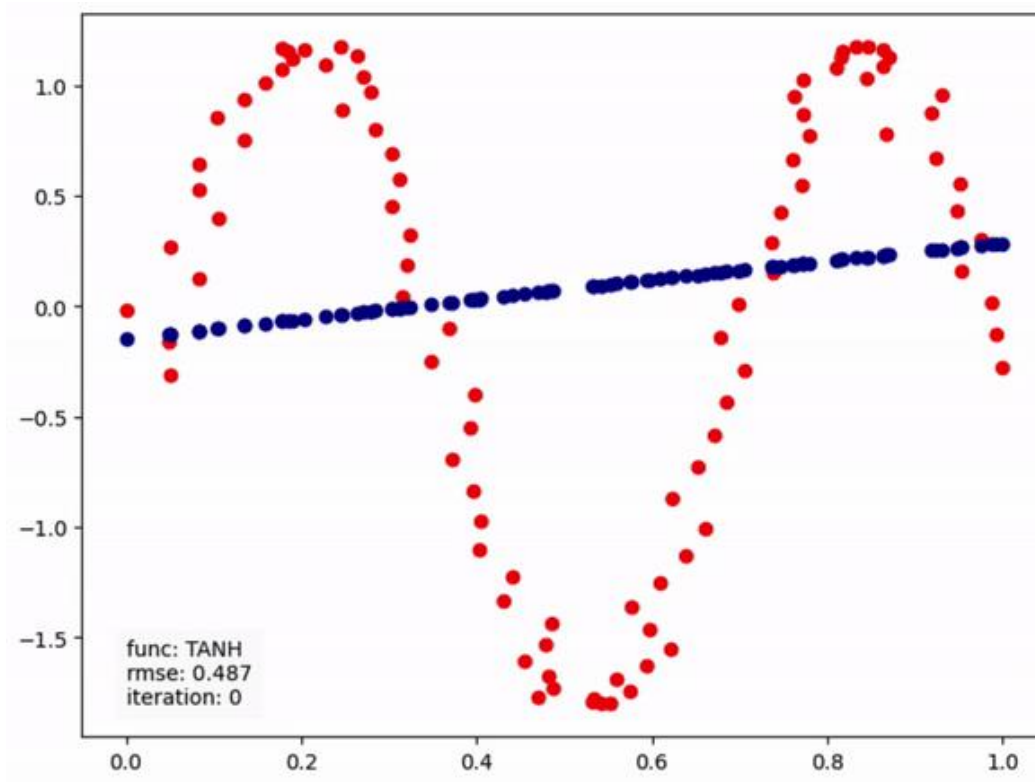
Learn patterns from data that generalize well to unseen data.



Develop models that can make accurate and reliable predictions in the real world.

Machine learning models

Functions/structures with parameters that we learn (approximate).



Model Type	Learnable Parameters
Linear regression	Coefficients (weights), bias
Decision tree	Split thresholds, tree structure
Random forest	Splits and structures of all individual trees
SVM	Support vectors, coefficients, bias
Neural network	Weights and biases in each layer
K-means	Centroid coordinates

Machine learning models

STRENGTHS:

1. Can learn complex patterns directly from raw data.
2. Can achieve high accuracy and approximate data.

Automatic learning, applicability across many domains.

RISKS:

1. Can learn spurious correlations.
2. Can memorize noise.

If not trained carefully, can behave unexpectedly and generalize poorly to unseen samples.

Machine learning models

Overfitting

RISKS:

1. Can learn **spurious correlations**.

2. Can **memorize noise**.

If not trained carefully, **can behave unexpectedly and generalize poorly to unseen samples.**

Example: Husky or wolf?



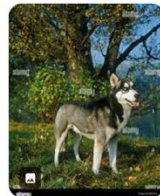
Husky or wolf?: Training data



Alamy
Siberian Husky, Adult Laying on G...



Unsplash
Adult white and gray Siberian husky l...



Alamy
Siberian Husky Do...



Alamy
Siberian Husky, Ad...



Unsplash
Adult white and gray S



iStock
250+ Portrait Of Old Husky Dog Stock...



PickPik
Royalty-Free photo: Adult malamute si...



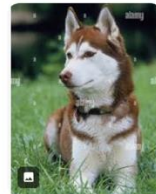
iStock
50+ Siberian Husky Laying In Flowers ...



Alamy
Siberian husky dog...



Dreamstime.com
9,331 Husky Adult Stock Photos - ...



Alamy
Husky laying on gr...



Shutterstock
45+ Thousand Husky On Gras...



iStock
20+ Siberian Husky And Beagle Pl...



Reddit
My beautiful



iStock
13,800+ Wolf In Snow Stock Photos, ...



Etsy
Winter Wolf Painting, Wol...



iStock
20+ Two Wolves Walking In The Sno...



Unsplash
Snow Wolf Pictures | Download Fre



Jackson Hole Wildlife Safaris
Winter Wolves of Yellowstone - Jackson Hole ...



Pinterest
Wolf in the snow #anim...



Ardea - Wildlife Pets...
Gray Wolf in Snowy...



ArtPhotoLimited
European wolf in the snow - European w...



Adobe Stock
Winter Wolf Images – Browse 196,8...



Collin Bogle - W ma...
Snow Plow - Runni...

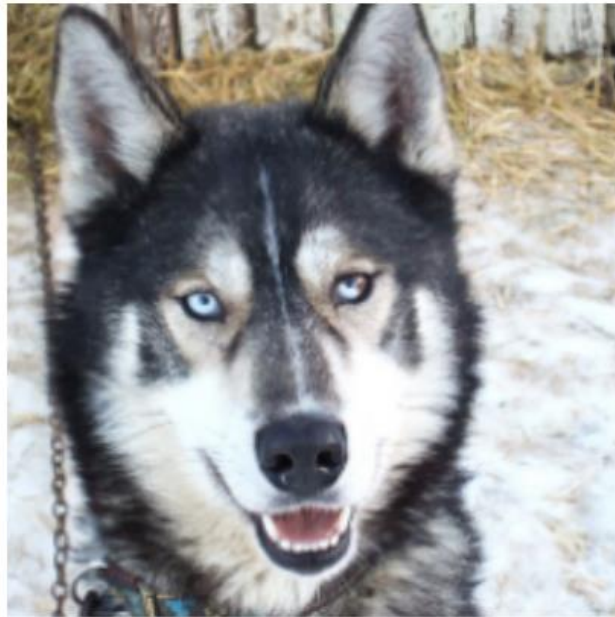


Pixels
Grey wolf in snow with wolf in dista...

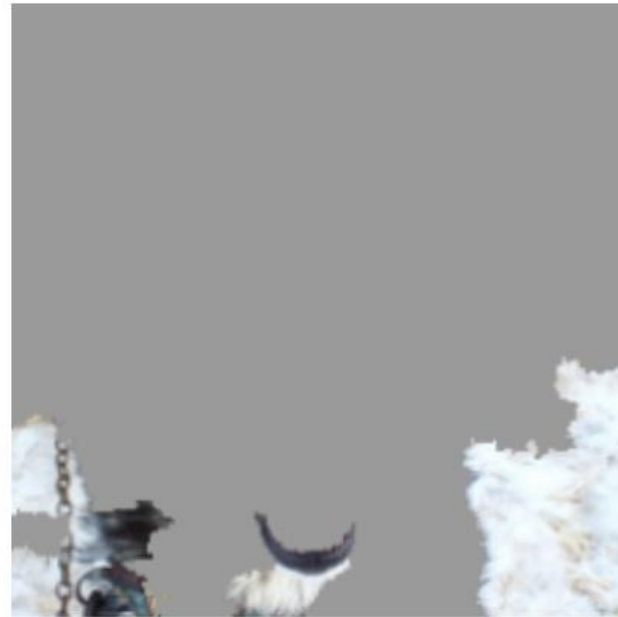


eBay · W magazynie
WOLF GLOSSY POSTER PICTURE ...

Husky in snow = Wolf



(a) Husky classified as wolf



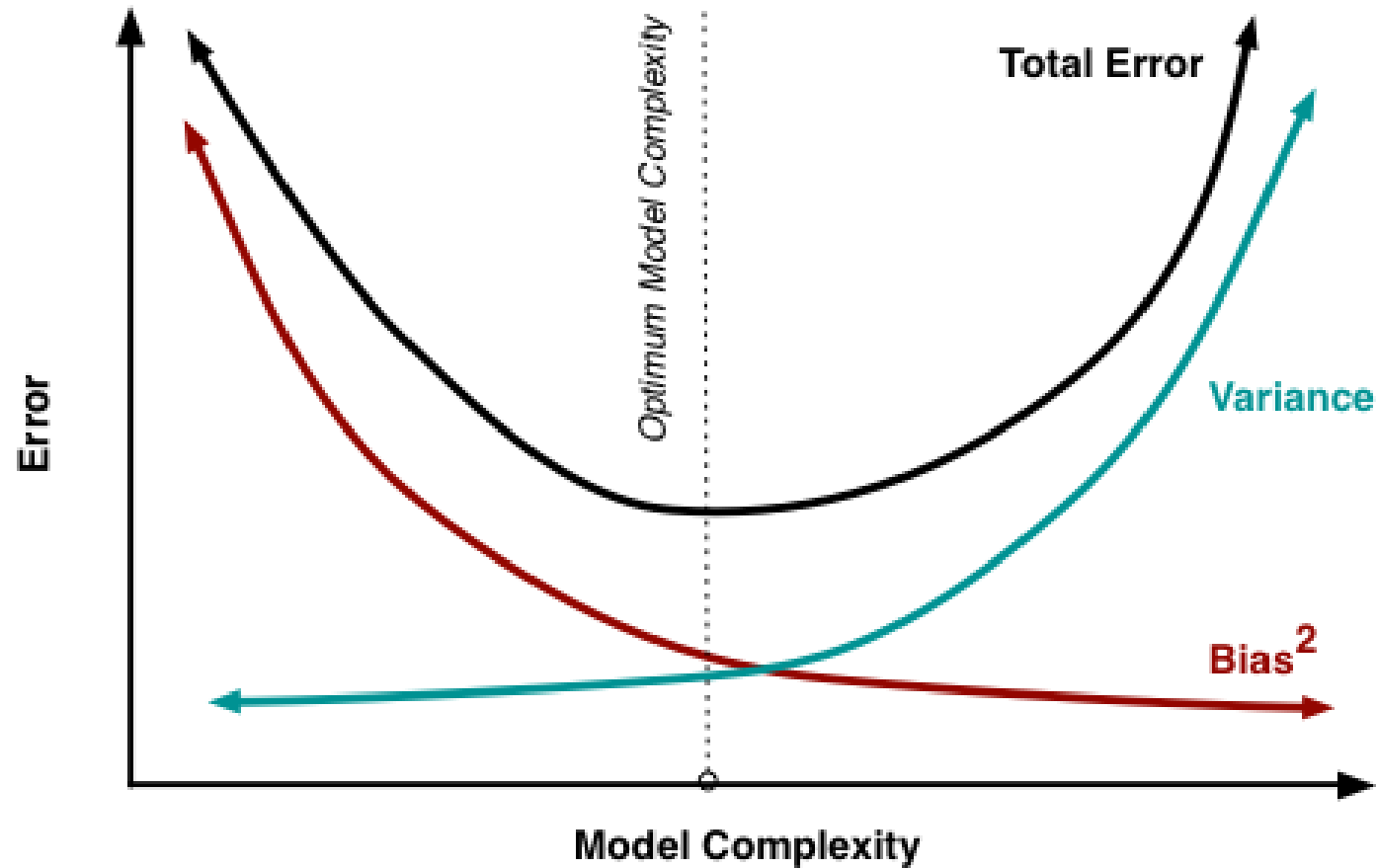
(b) Explanation

⑩ = Max speed 10 km/h



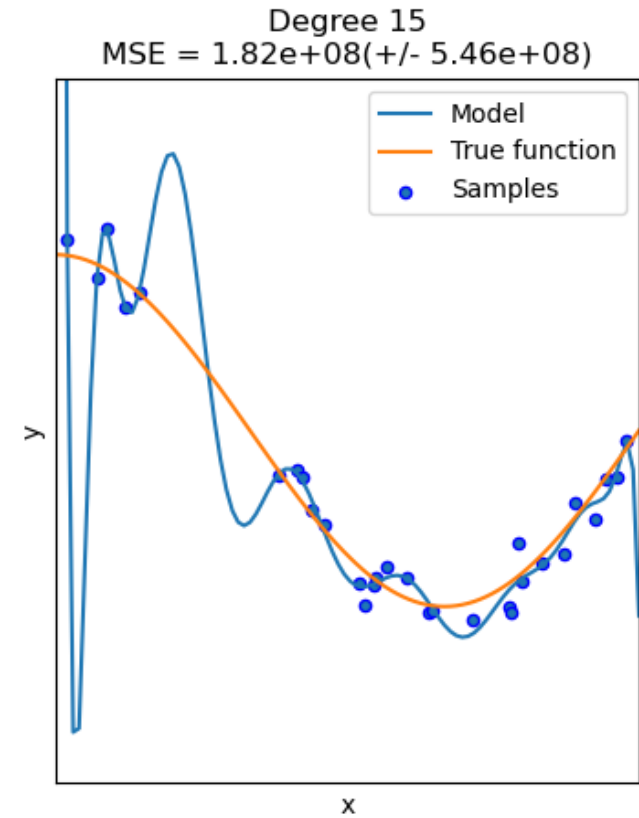
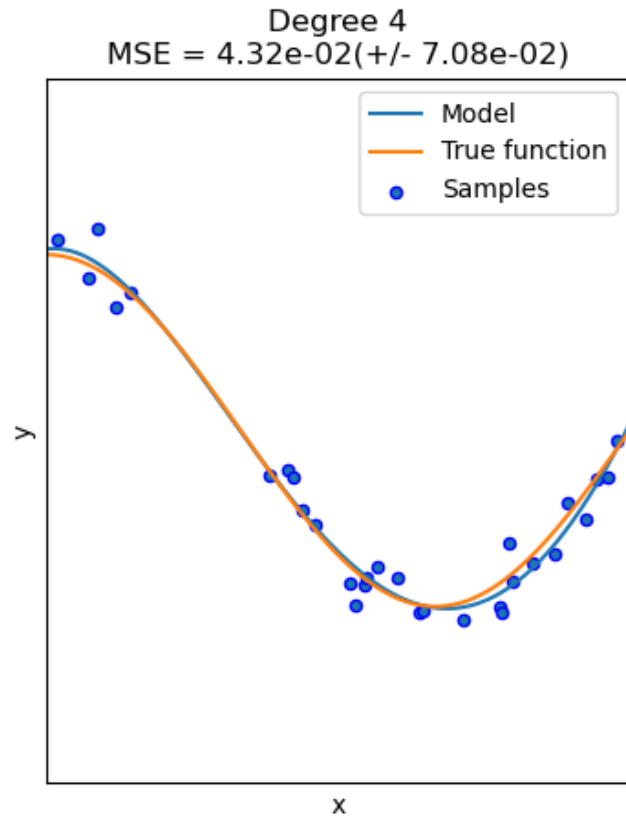
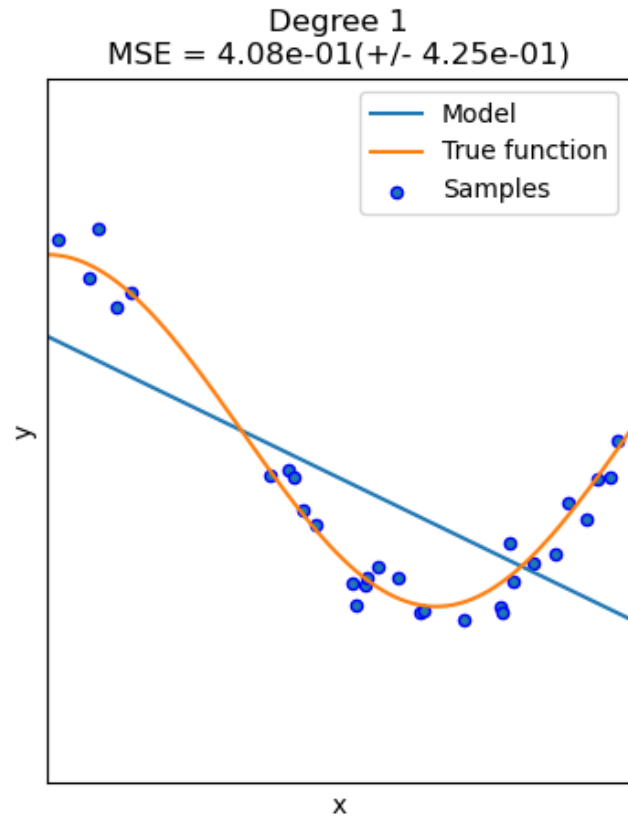
Illustration of an actual problem in cars with cruising control (2021).

Bias-variance trade-off

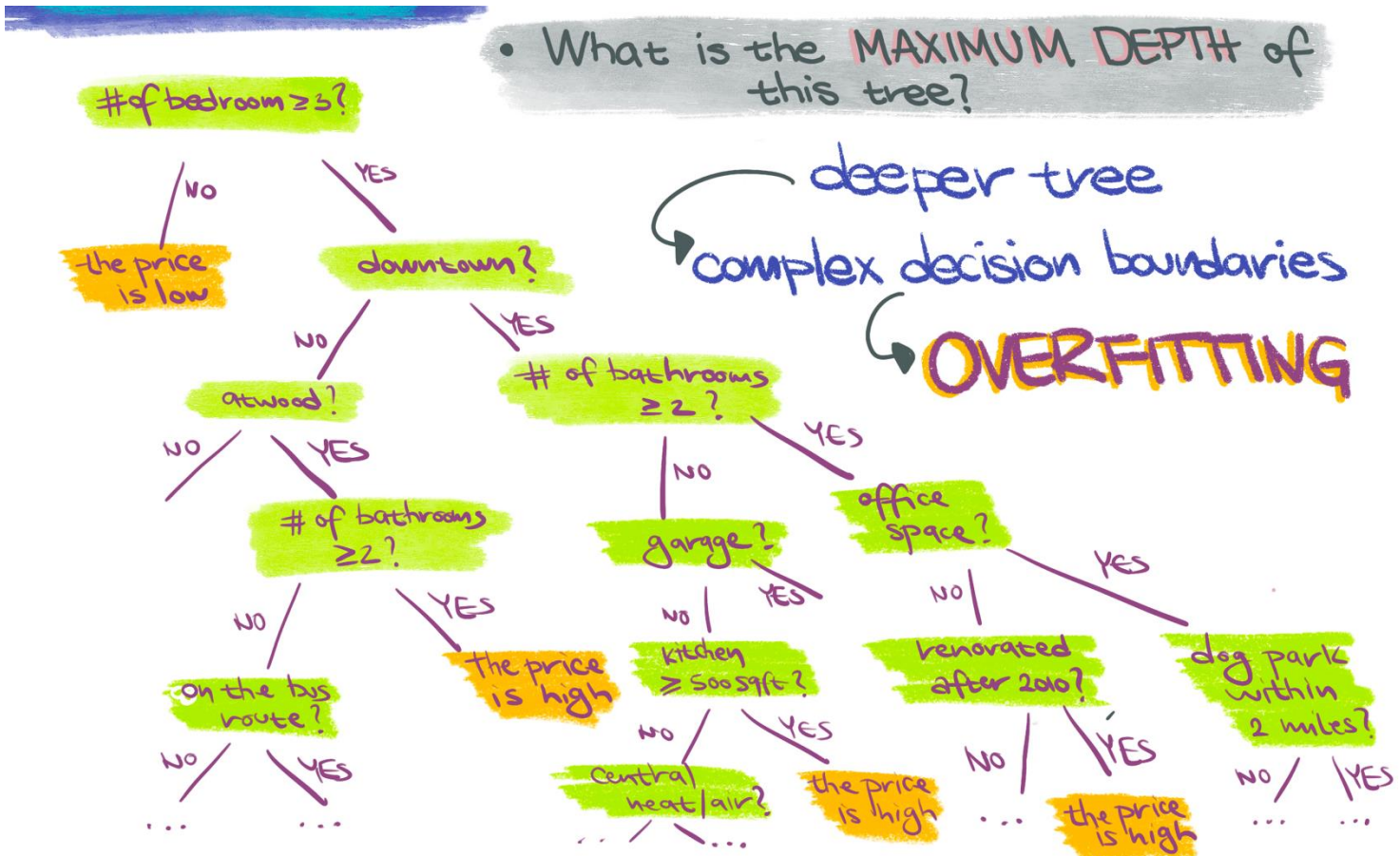


*Heavily overparametrized models (neural networks) can have high complexity and generalize well if designed and trained correctly

Overfitting in linear regression



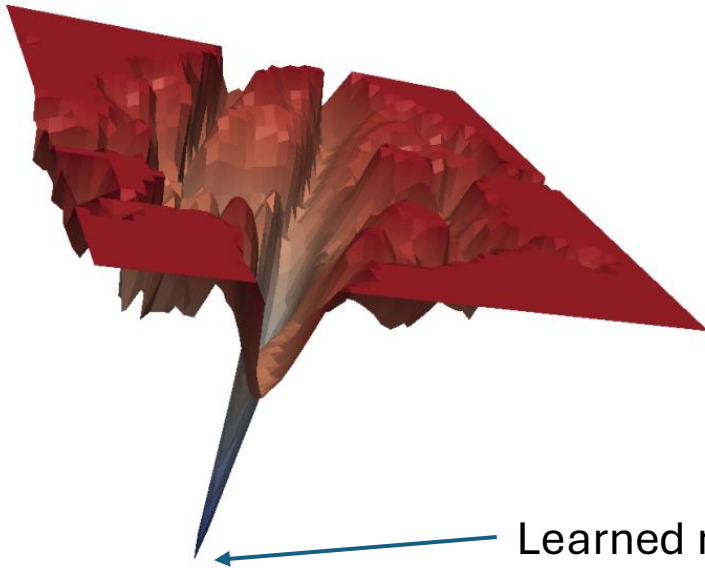
Overfitting in decision tree



Overfitting in a neural network

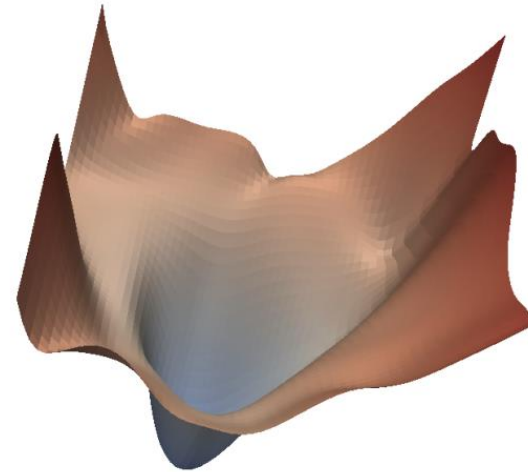
Loss landscapes of two trained neural networks

1.



Overfitting: Small data perturbations lead to sharp increase in loss

2.



Learned minimum

No overfitting: Small data perturbations don't influence the performance

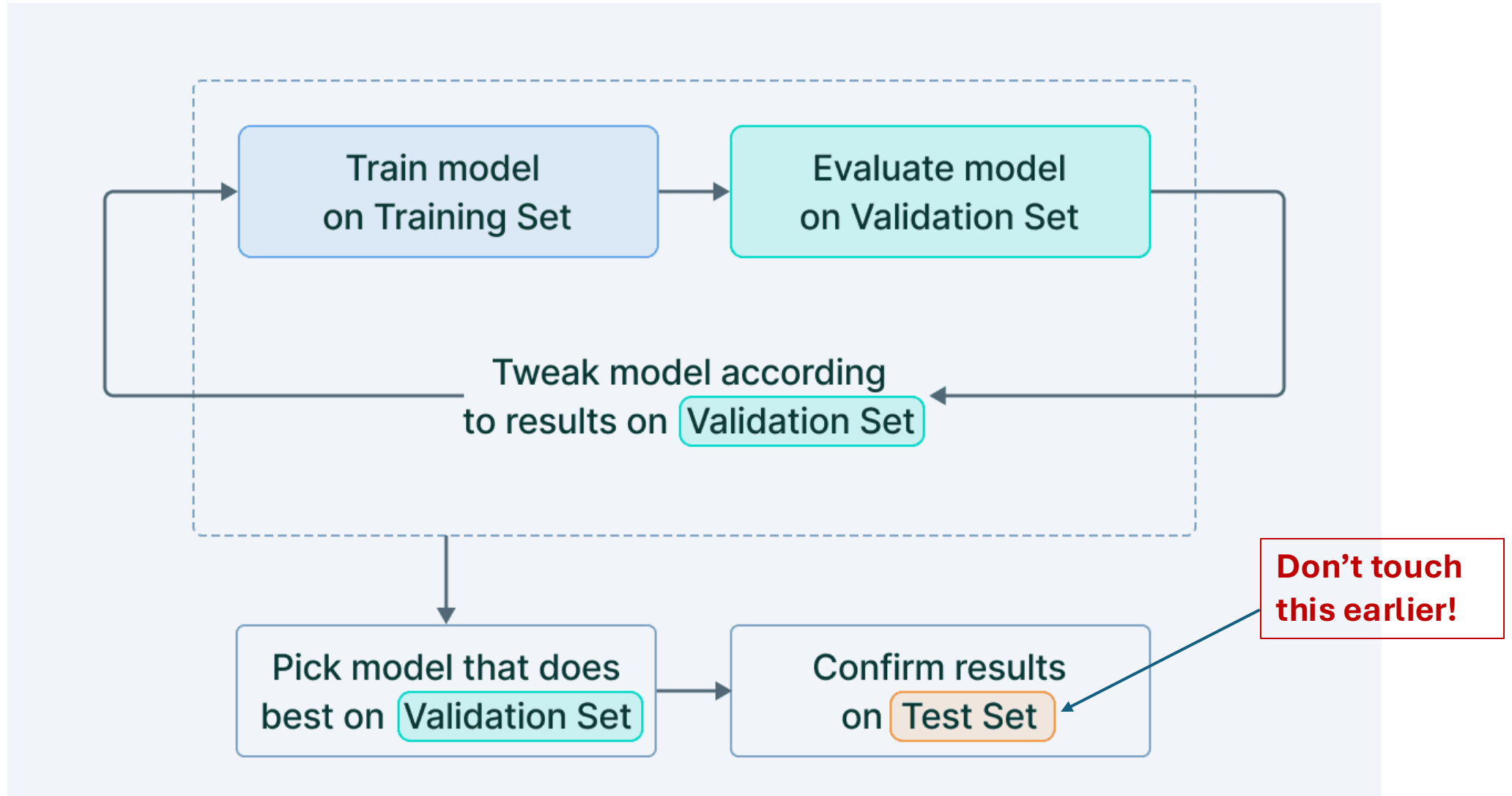
Quiz 1:

- A. Come up with a scenario of overfitting:
 - What was the predictive goal?
 - What is the overfitted (memorised) pattern?
 - What would you have liked the model to focus on instead?
- B. Which do you think is more dangerous in practice - underfitting or overfitting? Why?
- C. What does it mean if your model has zero training error?

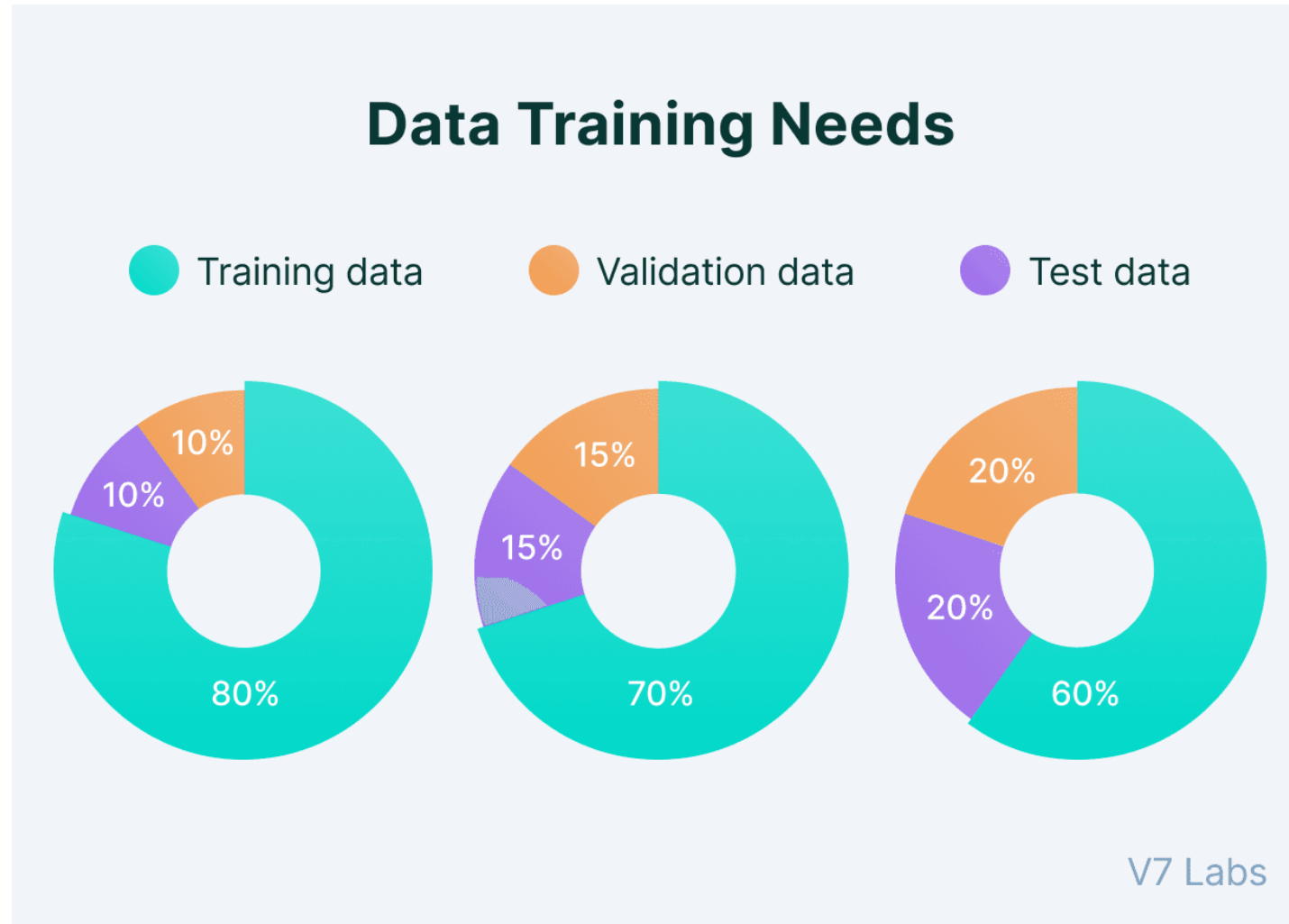
Data splitting



Machine learning pipeline

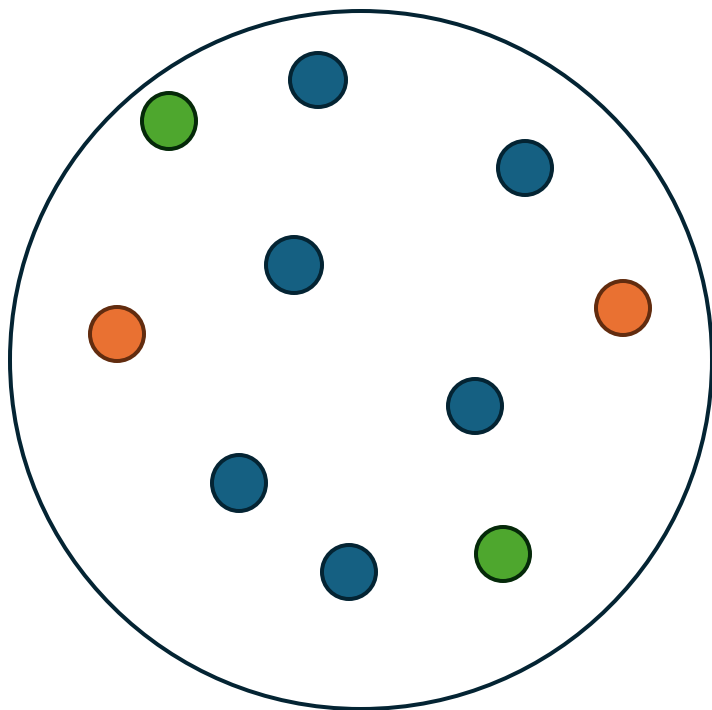


Training-validation-test

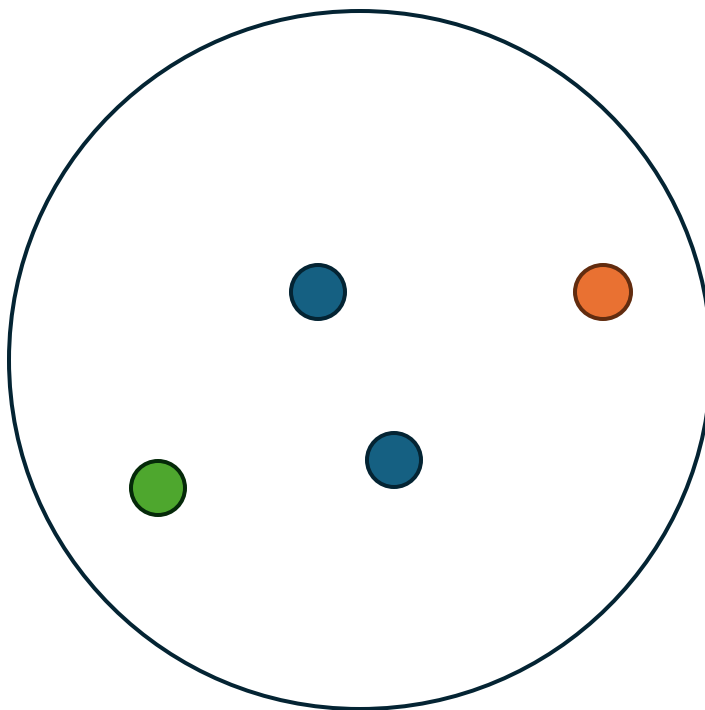


Stratified splitting

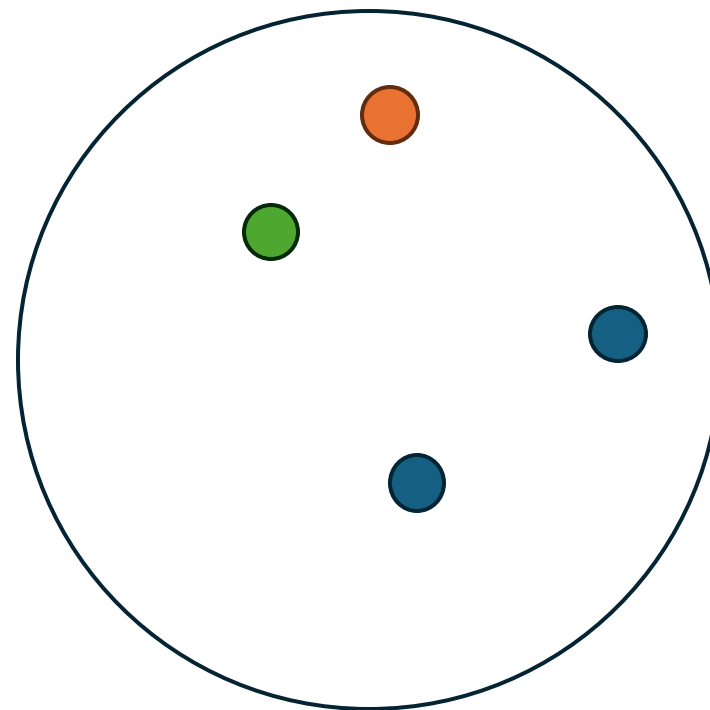
Train



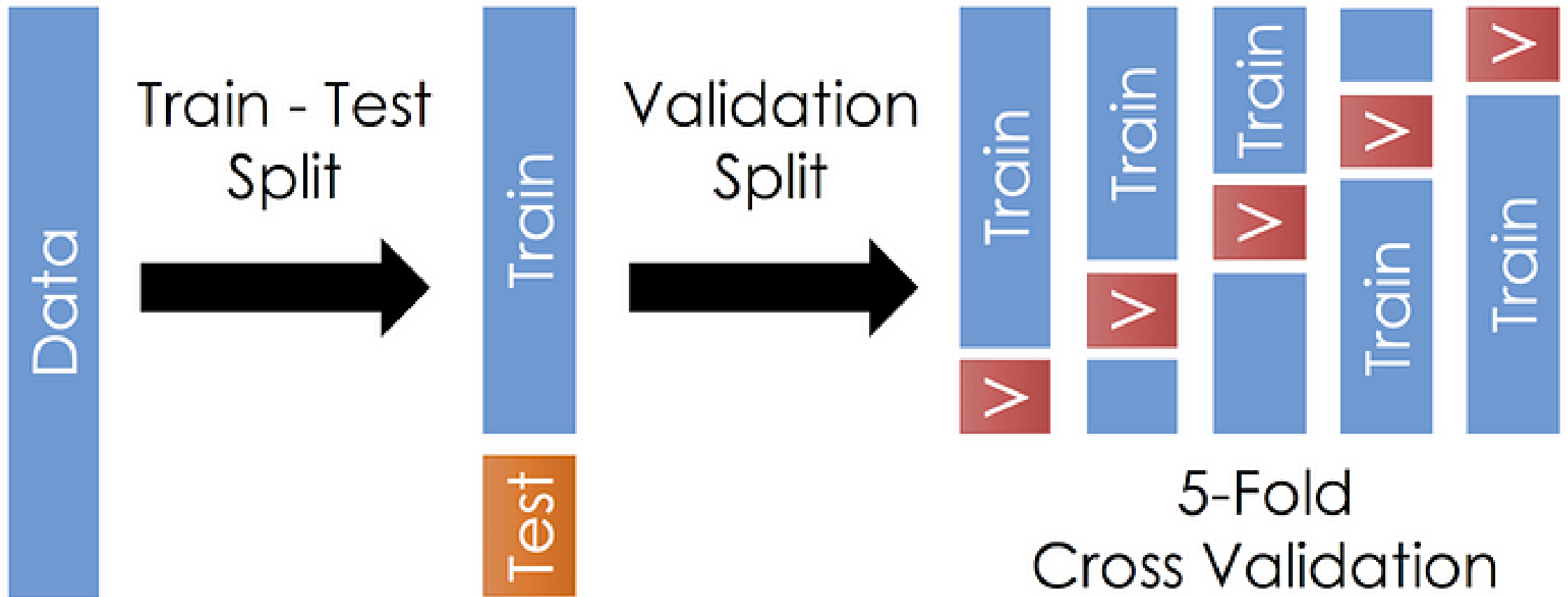
Validation



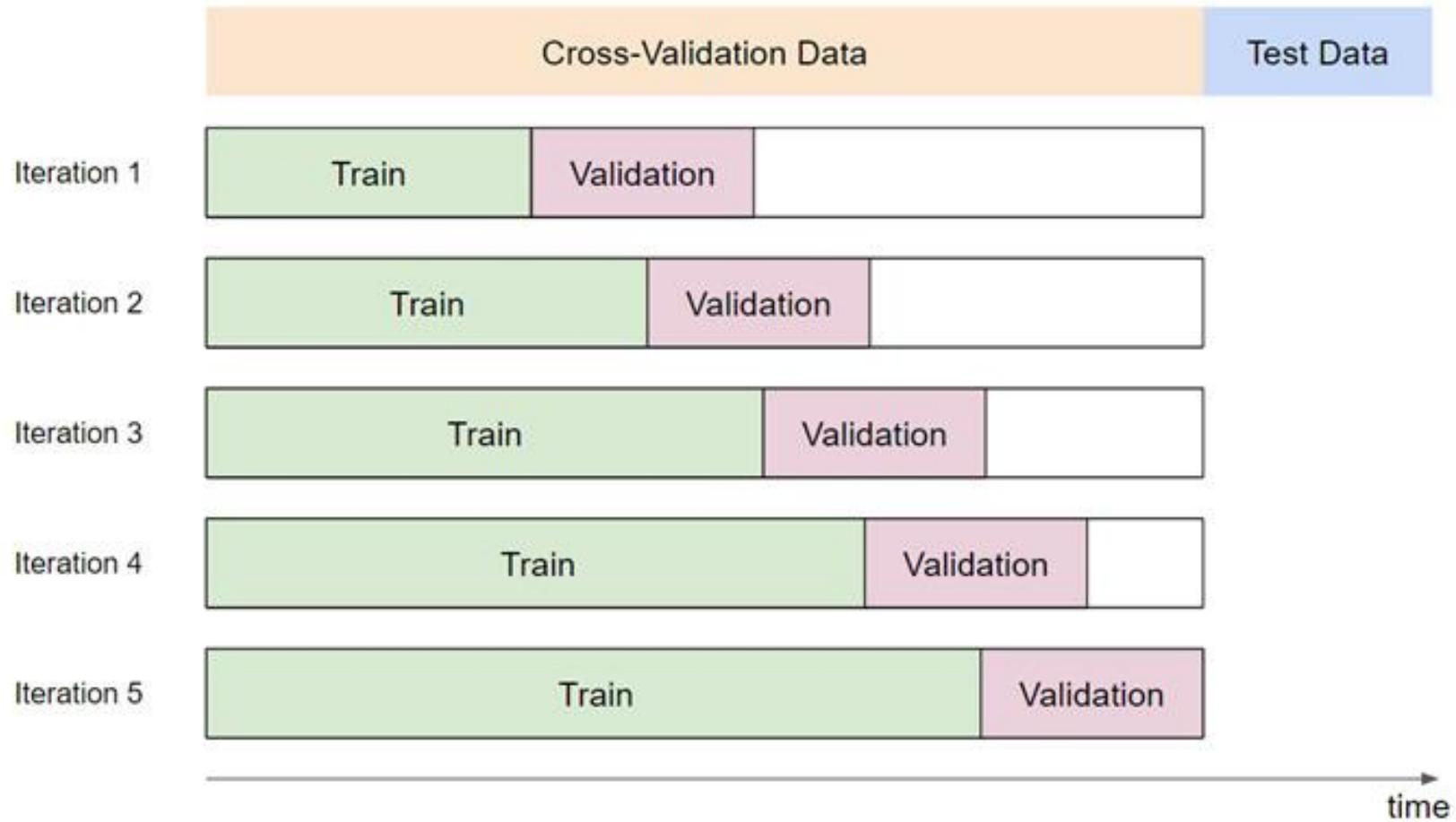
Test



Cross-validation (k-fold)



Timeseries data splits



Leave-One-Out CV (LOOCV)

- k-fold where k is the number of data points N
- Each “fold” uses 1 sample for validation and all others for training

Pros	Cons
<ul style="list-style-type: none">• Full use of the training data	<ul style="list-style-type: none">• Very computationally heavy (train model N times)• Usually does not outperform regular k-fold

Verdict: Use when there is very little data (100s)

Quiz 2:

A. What is the benefit of using:

- 90% training + 5% validation + 5% testing
- 70% training + 15% validation + 15% testing

B. In what way is cross-validation better than a single train-validation-test split?

Data leakage

Data leakage

- **Definition:** Information not available at prediction time is used in training
- **Result:** Over-optimistic, unreliable test scores
- **Silent failure:** You won't notice until the model fails in production

There are very many ways this can happen!

Scenario A

You want to predict a student's final exam score.

You include the following features:

- Lecture attendance rate to date
- Midterm score
- Final course grade
- Participation score
- Help sessions attended

Quiz 3: Where is the leak?

Scenario A: Leak

“Final course grade” ***includes*** the final exam grade!

Results:

- The model achieves small error in **training and validation**
- The model can't be used in real life, because “Final course grade” **won't be available** during testing!

Lesson: Use only the features available at deployment!

Scenario B

You want to predict electricity consumption at each hour.

You include the following features:

- Temperature
- Location
- Humidity
- Day of week
- Hour of day

You randomly split the data into training and test sets.

Quiz 3: Where is the leak?

Scenario B: Leak

Randomly splitting hourly data leads to **highly correlated samples** in train and test - in fact, they're nearly duplicates.

Results:

- Model seems to perform extremely well — low test error
- When deployed on truly unseen time periods (e.g., next week), performance **collapses**

Lesson: Use truly independent samples in data splits!

Scenario C

You want to predict how many days a patient will stay in the hospital. You include the following features:

- Age
- Diagnosis code
- Initial lab results
- Number of previous admissions

Some features have missing values, so you impute (fill in) missing values using the mean of the dataset. You randomly split the data into training and test sets.

Quiz 3: Where is the leak?

Scenario C: Leak

By imputing **before splitting** into train/test, you include test data in the imputation statistics.

Results:

- The model learns from patterns that include information from the test set
- Performance may drop when the model is used on truly unseen data

Lesson: Always perform imputation **after splitting the data**.

Scenario D

You want to predict the average power consumption in a 1-hour window based on sensor readings.

To increase the training sample, you generate 1-hour sliding windows that shift by 10 minutes at a time.

You randomly split the dataset into training and test sets.

Quiz 3: Where is the leak?

Scenario D: Leak

The sliding windows **overlap by up to 50 minutes**, meaning most windows in the training data include copies of the testing data.

Results:

- Very low error on training and validation data
- Overfitting: The model fails in real life

Lesson: Split the data into train/val/test sets before further dividing them into time windows. The test data should be in the future!

Scenario E

You want to predict whether a patient has a disease based on gene expression data.

Each sample has thousands of gene features. You analyse your entire dataset and reduce dimensionality by selecting the **top 500 most predictive genes**. You then split the dataset into training and test sets.

Quiz 3: Where is the leak?

Scenario E: Leak

You selected the top genes using the **entire dataset**, including the test set.

Results:

- The test set **influenced the choice of features**
- Model is indirectly optimized to perform well on test data hence the test metrics are not reliable

Lesson: Use only training data for feature selection.

Scenario F

You want to predict whether a user will click on an ad.

Each row in your dataset is a record of a user viewing a specific ad, along with:

- User demographics
- Ad content features
- Time of day
- Whether the user clicked

You perform a standard random train/test split at the observation level.

Quiz 3: Where is the leak?

Scenario F: Leak

The **same users** appear in both training and test sets.

Result:

- The model learns **user-specific behavior** instead of general patterns
- Test set performance looks great
- On truly new users, performance **drops significantly**

Lesson: Split the data into sets at the user level!

Data leakage prevention!

1. Rigorously separate training from validation and test at all stages.
2. All data transformations should be fit on training data only, then applied to validation/test.
3. Consider if each feature is genuinely available at prediction time.
4. Time series: always train on past data and validate on future data – never randomly shuffle in a way that future leaks into past.
5. Sanity-check if high validation performance might be “too good to be true”!

Hyperparameter tuning



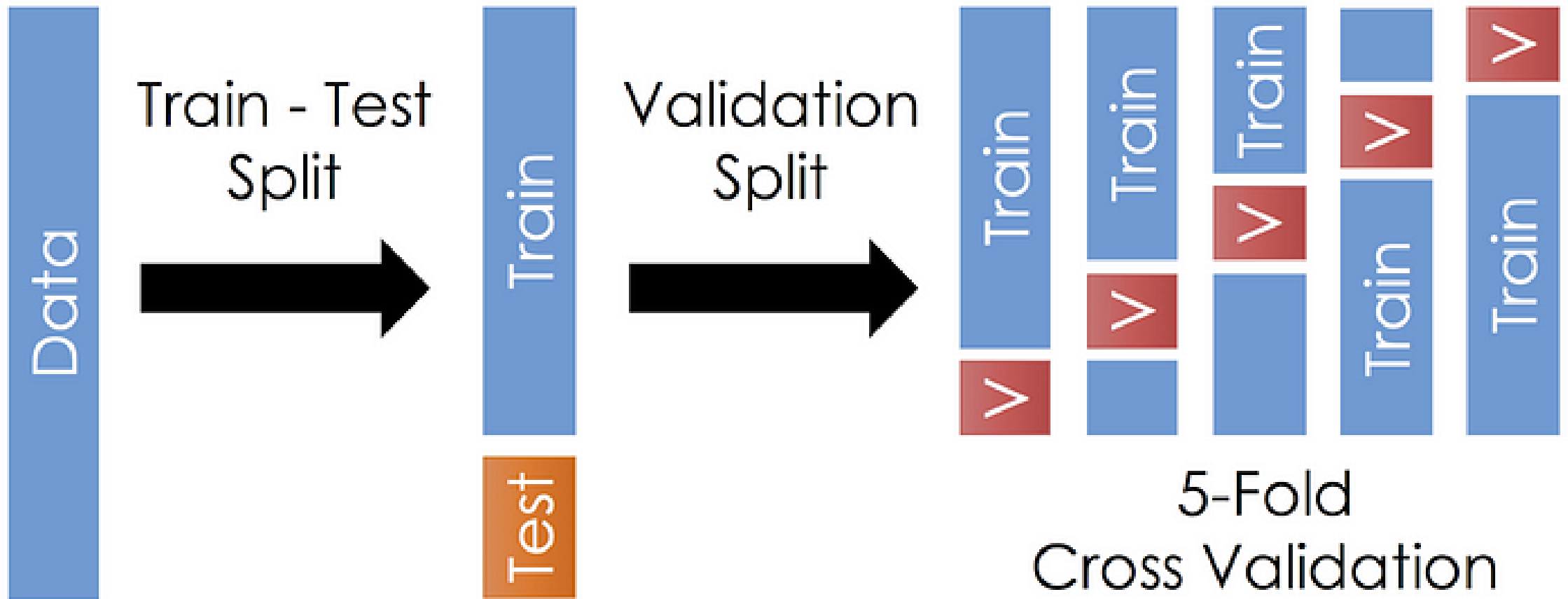
Parameters vs. hyperparameters

Model	Parameters: Learned during training	Hyperparameters: Set before training
Decision Tree	<ul style="list-style-type: none">- Variable split thresholds at nodes- Leaf predictions	<ul style="list-style-type: none">- Max depth- Min samples per split
Neural Network	Weights and biases of each neuron	<ul style="list-style-type: none">- Learning rate- Number of layers- Units per layer- Activation function- Batch size
Random Forest	Same as in decision tree	<ul style="list-style-type: none">- Number of trees- Max features per tree

Why tune hyperparameters?

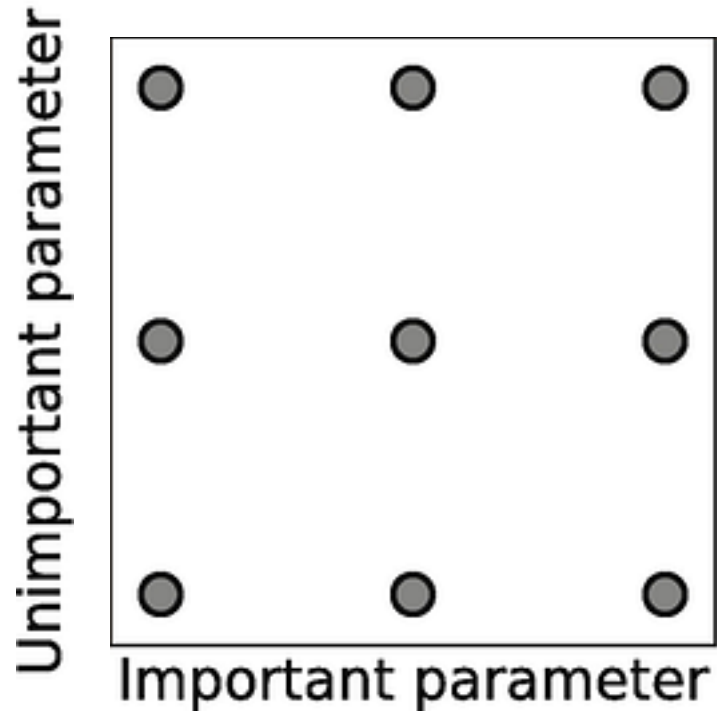
- Bias-variance trade-off: Some hyperparameters control model complexity.
- There is no universal best setting for all datasets
- Best setups are found empirically

Hyperparameter search

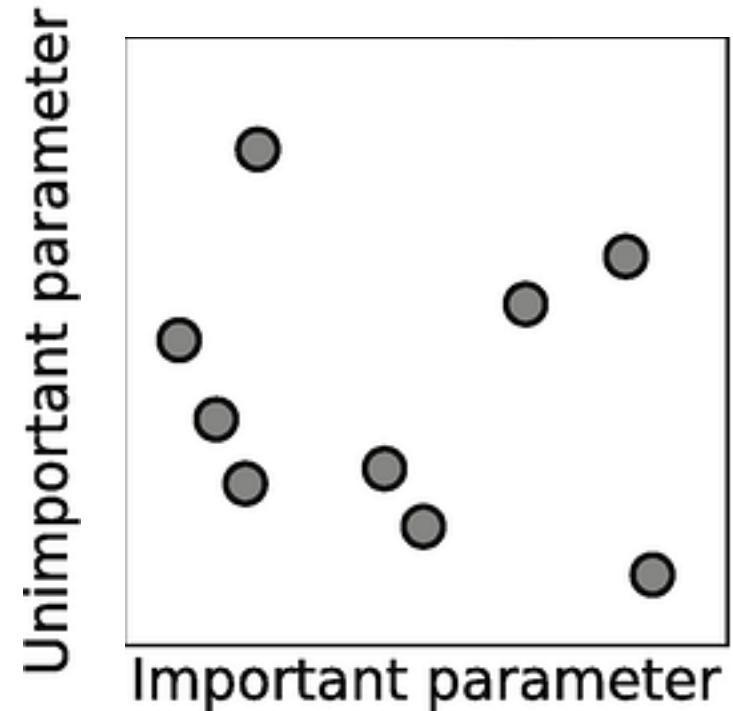


Grid search vs. random search

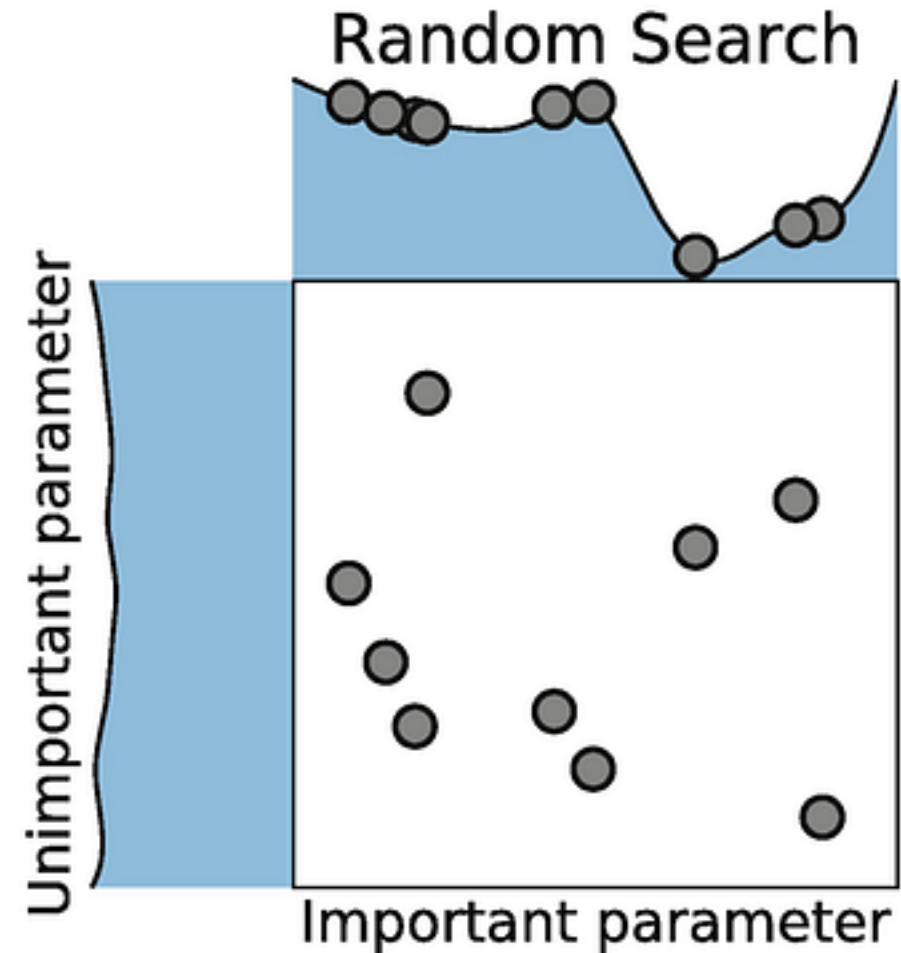
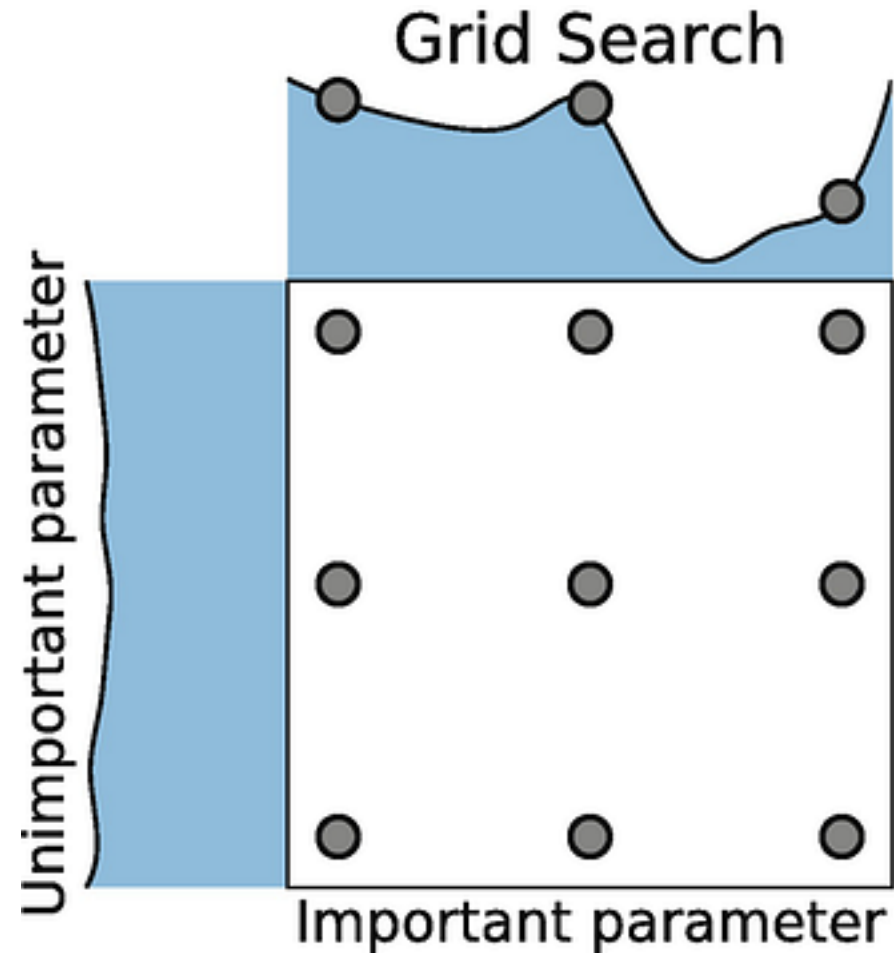
Grid Search



Random Search



Grid search vs. random search



Hyperparameters are interdependent

Neural networks:

- Initialization tactic depends on activation functions
- Larger training batches often require lower learning rates
- ...

Tree models:

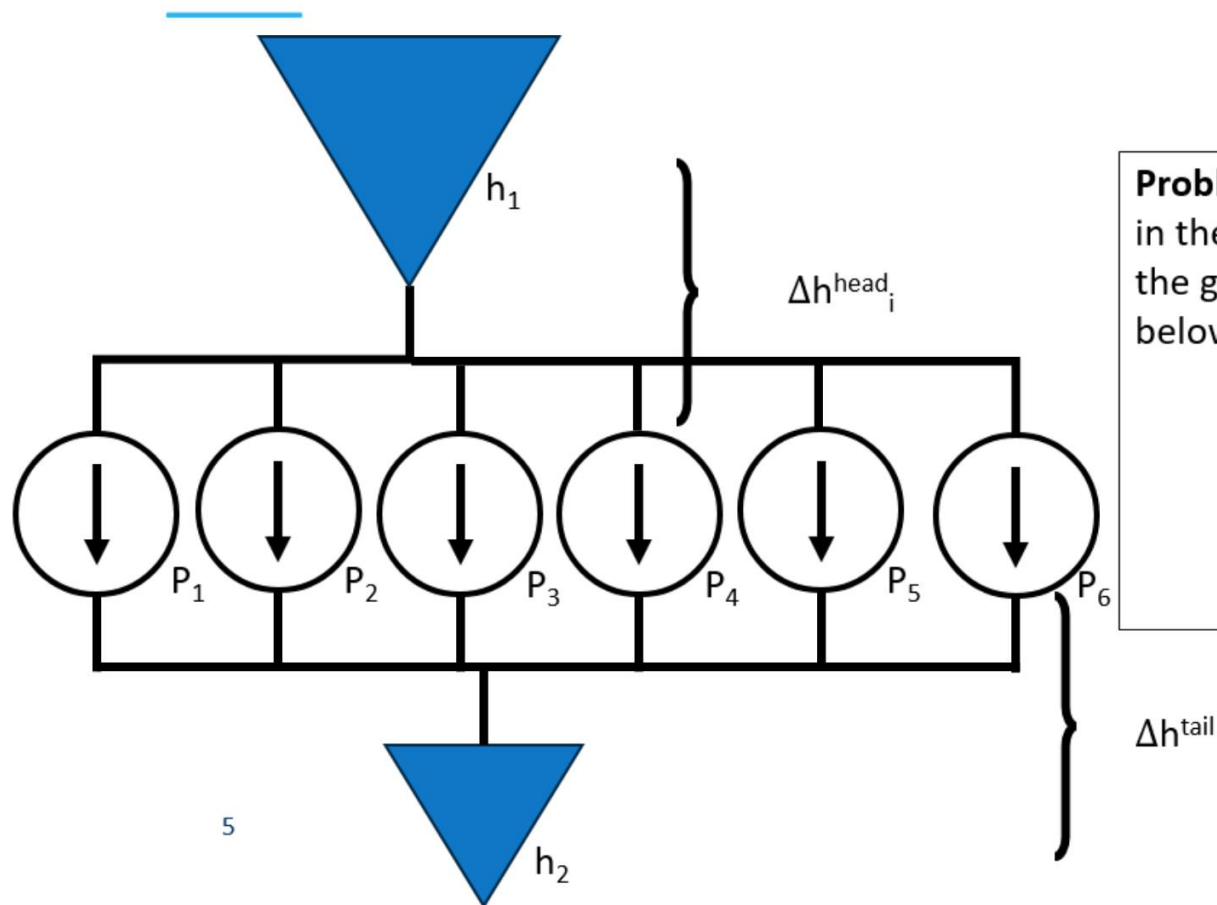
- Use more trees if you use few features
- Don't overrestrict: max depth, max nodes, max features...
- ...

Quiz 4

- A. When would you prefer grid search over random search?
- B. You run a hyperparameter search, and most combinations result in poor performance. Only a few models perform reasonably well. What can you do?

Demo + Quiz 5

Water power plant – larger system



Problem: Given reservoir height h_1 and h_2 , produced power in the generators P_1 - P_6 , estimate the tunnel losses above the generators $\Delta h^{\text{head}_1} - \Delta h^{\text{head}_6}$ and the common loss below, Δh^{tail}

Evaluation



Baseline

The "**minimum bar**" your model must beat - if it can't outperform the baseline, it's probably not worth using.

For example:

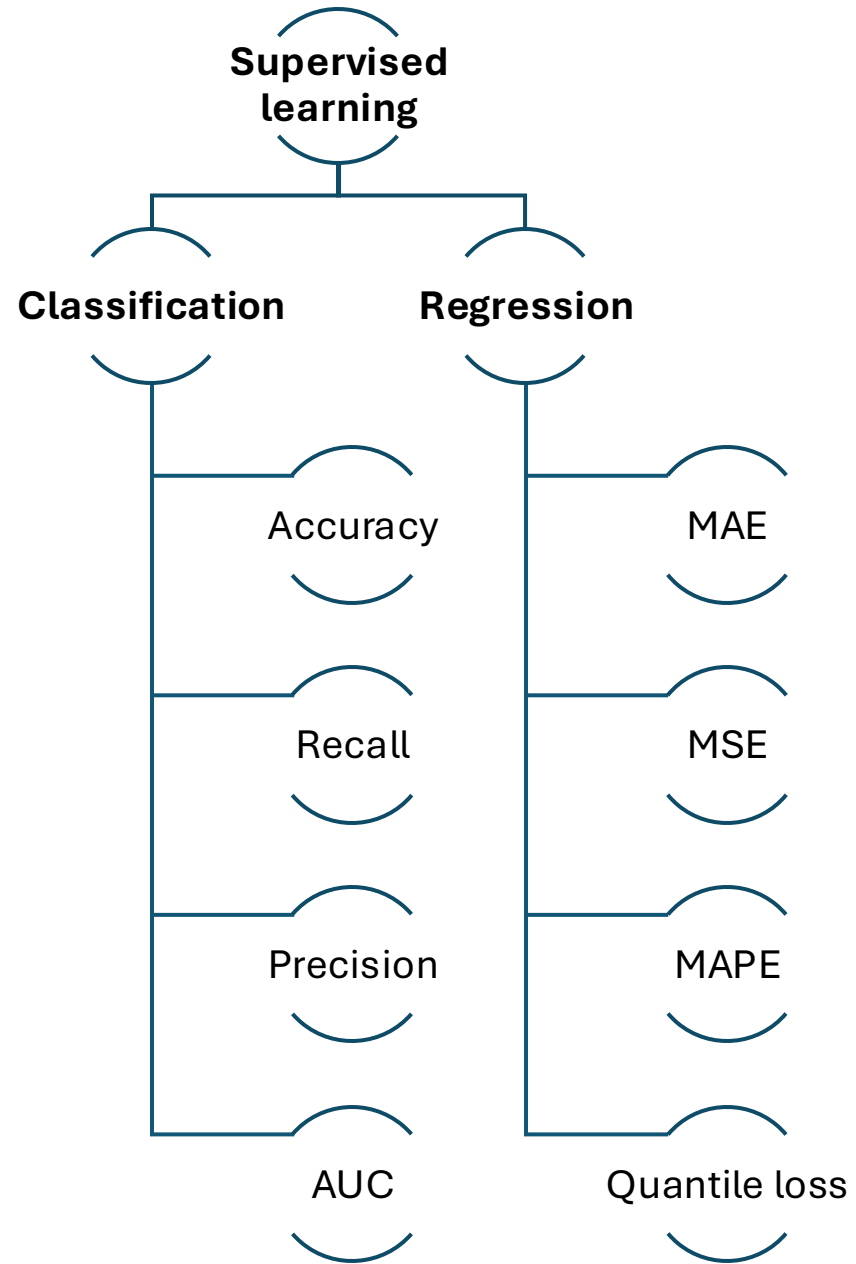
- Human evaluation
- Existing models or systems
- Dummy baselines

“Dummy” baselines (the absolute minimum)

- Regression:
 - Predict a the mean, median, constant value...
 - Linear regression model
 - Timeseries prediction (if last timestep included as input): predict the last value
- Classification:
 - Predict the majority class

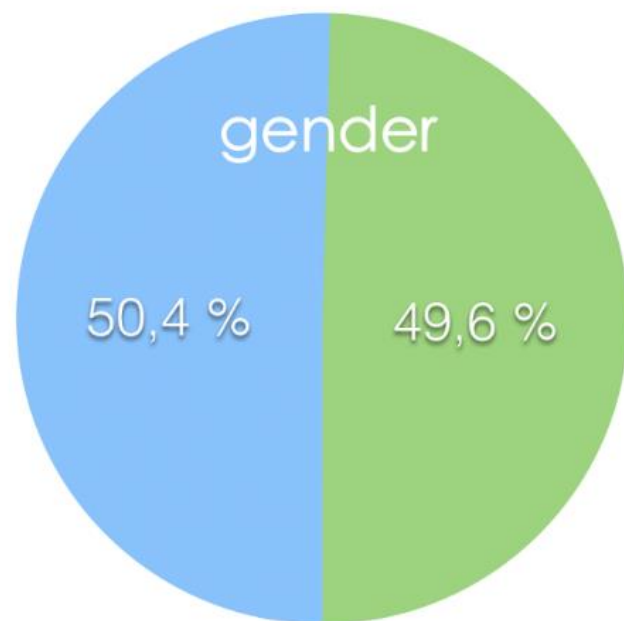
Many metrics

No one way to evaluate models!
When to use each?
Interpretation is key!



Accuracy can be misleading

✓ Balanced Dataset

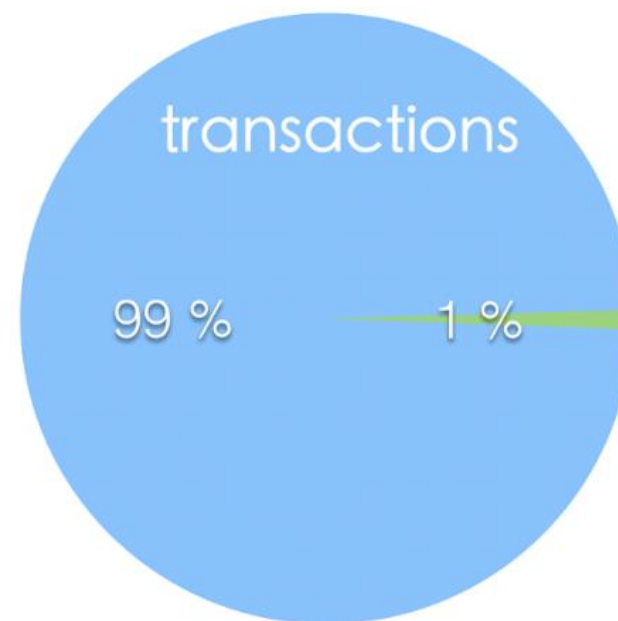


● male

● female

Majority class prediction: **“male”**
Accuracy: **50,4%**

✗ Unbalanced Dataset



● normal

● fraudulent

Majority class prediction: **“normal”**
Accuracy: **99%**

Precision and recall: Use **together!**

- **Precision** = "Of all predicted positives, how many were correct?"
 - **Recall** = "Of all actual positives, how many were found?"
1. Raise alarm anytime at slight suspicion (many false alarms!):
High recall, but low precision
 2. Raise alarm only when you are 100% sure (many missed alarms!):
High precision, but low recall

MAE vs. MSE

Mean squared error:

- Penalize larger errors
- Best for optimization in general: Prioritize improving the algorithm where it struggles the most
- Sensitive to outliers

Mean average error:

- Treat errors equally, regardless of their amplitude
- Better when outliers are present
- Direct explanation of errors (off by x value)

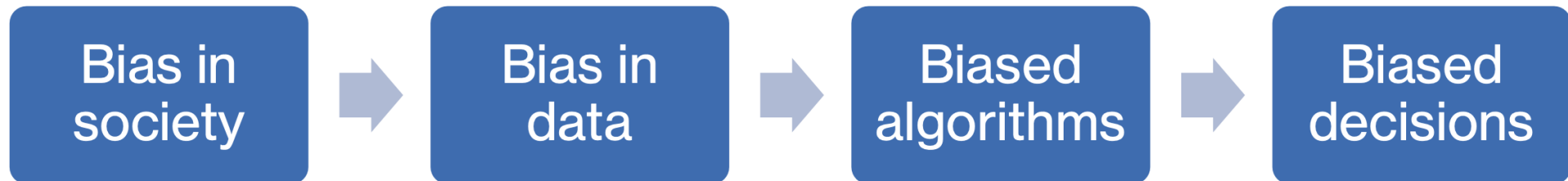
Quiz 6

- A. You are developing a model to predict whether a person has an increased risk of developing cancer. Which metric would you care about most and why?
- B. You are building a spam filter. Which metric do you care about most?
- C. You have built a model that predicts house prices in a neighborhood. Your model has $\text{MAE} = \$75,000$. You calculate that the average absolute deviation of house prices from the mean is $\$70,000$. What does it mean?

Evaluation: Other considerations

Fairness evaluation

- The process of measuring whether a model treats different groups **equitably** - especially across sensitive attributes like race, gender, age, or income.



Example: Optum's healthcare algorithm

- An algorithm used to prioritize care for patients in the U.S.
- It used **healthcare spending** as a proxy for health needs.
- Black patients received **lower care priority** because they historically **receive less medical spending**, even when equally sick.

[nature](#) > [news](#) > article

NEWS | 24 October 2019 | Update [26 October 2019](#)

Millions of black people affected by racial bias in health-care algorithms

Study reveals rampant racism in decision-making software used by US hospitals – and highlights ways to correct it.

[Heidi Ledford](#)



Source: Nature

Example: Amazon's reicruting tool (2014-2018)

- AI model to screen resumes for tech jobs.
- The model learned from historical data where mostly **male candidates** were hired.
- **Penalized resumes with phrases like: “women’s chess club”** and language more commonly used by females.

Amazon scraps misogynistic AI recruiting tool

The experiment exposed limitations of machine learning and how unreliable algorithms trained on potentially biased data can be

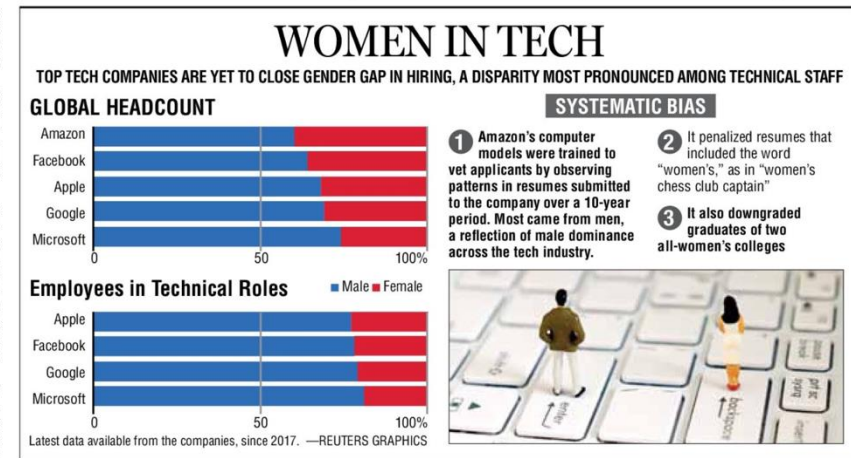
San Francisco: Amazon Inc's machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.

The team had been building computer programs since 2014 to review job applicants' resumes with the aim of mechanizing the search for top talent, five people familiar with the effort.

Automation has been key to Amazon's e-commerce dominance, be it inside warehouses or driving pricing decisions. The company's experimental hiring tool used artificial intelligence to give job candidates scores ranging from one to five stars — much like shoppers rate products on Amazon, some of the people said.

“Everyone wanted this holy grail,” one of the people said. “They literally wanted it to be an engine where I’m going to give you 100 resumes, it will spit out the top five, and we’ll hire those.”

But by 2015, the company realized its new custom was



That is because Amazon's computer models were trained to vet applicants by observing

across the tech industry.

In effect, Amazon's system taught itself that male candi-

downgraded graduates of two all-women's colleges, according to people familiar with the mat-

these particular terms. But that was no guarantee that the machines would not devise other

mately disbanded the team by the start of last year because executives lost hope for the project, according to the people, who spoke on condition of anonymity. Amazon's recruiters looked at the recommendations generated by the tool when searching for new hires, but never relied solely on those rankings, they said.

Amazon declined to comment on the recruiting engine or its challenges, but it said that it is committed to workplace diversity and equality.

The company's experiment offers a case study in the limitations of machine learning. About 55 per cent of US human resources managers said that AI would play a role in recruitment within the next five years.

However, doubts have been raised about how reliable algorithms trained on potentially biased data will be.

“How to ensure that the algorithm is fair, how to make sure the algorithm is really interpretable and explainable

Source: Reuters

Example: Face recognition

Face recognition algorithms were 65% accurate on Black females, vs. 95-100% on the rest of population (2019)

98.7%



**DARKER
MALES**

68.6%



**DARKER
FEMALES**

100%



**LIGHTER
MALES**

92.9%



**LIGHTER
FEMALES**

Causes of AI bias: Biased data

- Unfair sampling: Lack of data due to exclusion
- Unfair labelling: Wrong labels due to prejudice
- Small data: Too little data to represent groups and their intersections

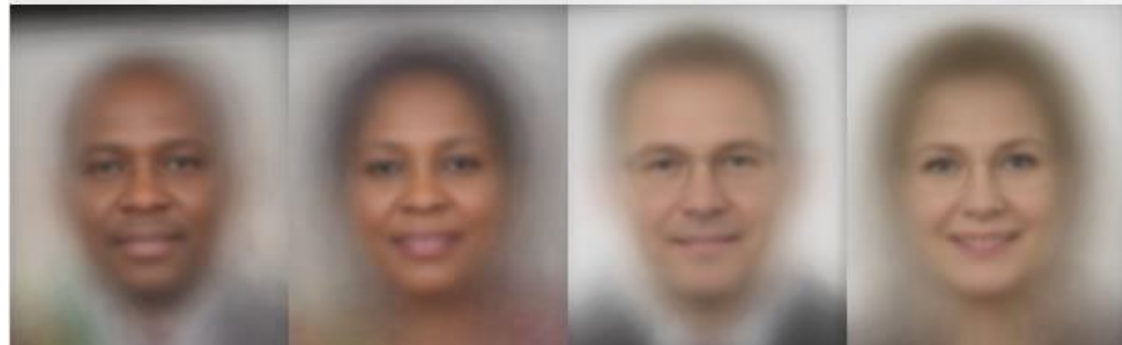
Causes of AI bias: Model design

- Aggregation, despite differences:
 - ‘One model fits all’ and biased data
 - Over-simplistic models
- Sensitive data (and correlates)
 - Predictive variables/text/image
 - Target variables
- Model evaluation
 - Algorithms are evaluated on their overall effectiveness, not fairness

Towards fair algorithms

Good data

Representative and balanced across various backgrounds, for example different genders or race, and their intersections, for example Black females.



Source: gendershades.org

Towards fair algorithms

Good data

Model design

- Exclusion of sensitive information and its correlates
- Thoughtful design of predictive targets
- Other: Correcting biases with transfer learning

Towards fair algorithms

Good data

Model design

Algorithmic design

- Designing algorithms to be more interpretable and transparent
- Other: Guided learning processes for fairness

Towards fair algorithms

Good data

Model design

Algorithmic design

Fair evaluation

Evaluating models on their fairness across different groups and group intersections rather than the overall models' accuracy.

Towards fair algorithms

Good data

Model design

Algorithmic design

Fair evaluation

Education

Understanding and responsibility for algorithms
Applying fair algorithm design guidelines
Regulations for fair AI

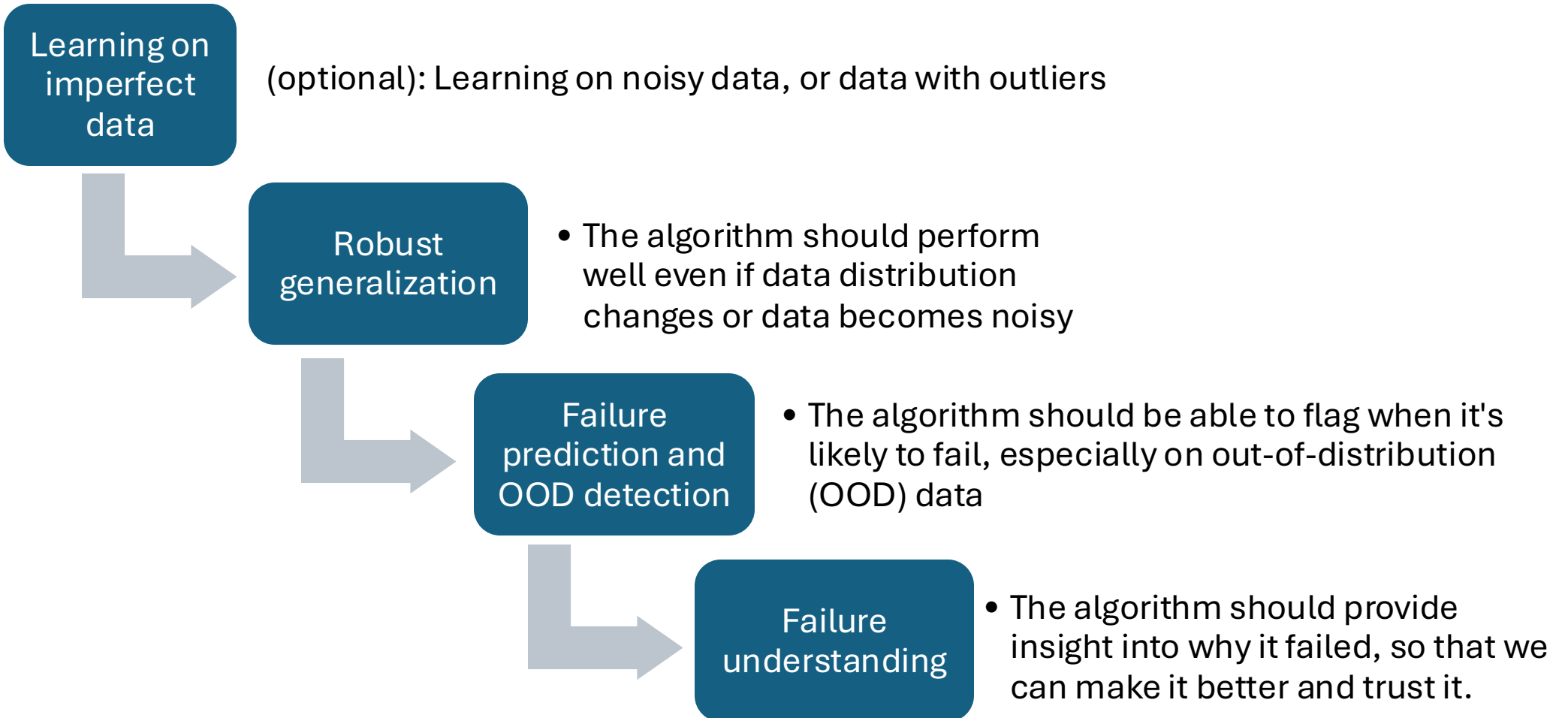
Model robustness

The capacity of a model to sustain stable predictive performance in the face of variations and changes in the input data.

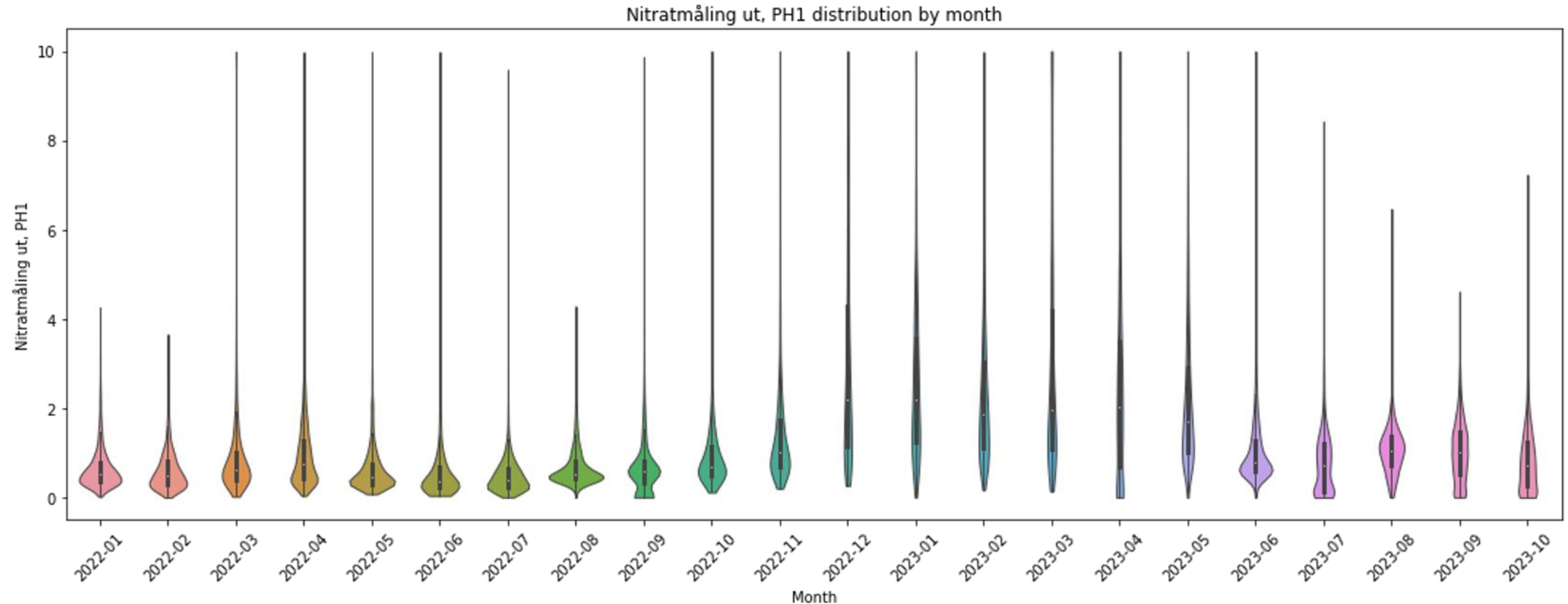
Applied: The ability of ML models to maintain stable performance across varied and unexpected environmental conditions.

- *robustness \neq accuracy*
- *robustness \neq i.d.d. generalization*


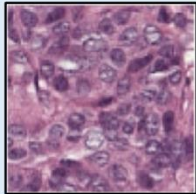
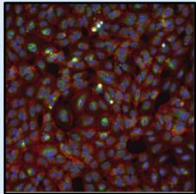
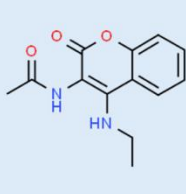
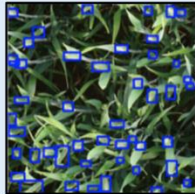



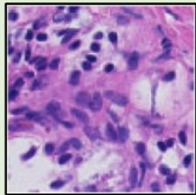
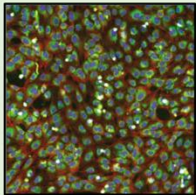
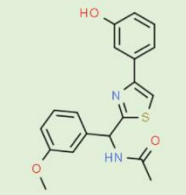



The elements of ML robustness



Distribution of target variable by month



Distribution shifts in the wild

	Domain generalization					Subpopulation shift	Domain generalization + subpopulation shift			
Dataset	iWildCam	Camelyon17	RxRx1	OGB-MolPCBA	GlobalWheat	CivilComments	FMoW	PovertyMap	Amazon	Py150
Input (x)	camera trap photo	tissue slide	cell image	molecular graph	wheat image	online comment	satellite image	satellite image	product review	code
Prediction (y)	animal species	tumor	perturbed gene	bioassays	wheat head bbox	toxicity	land use	asset wealth	sentiment	autocomplete
Domain (d)	camera	hospital	batch	scaffold	location, time	demographic	time, region	country, rural-urban	user	git repository
# domains	323	5	51	120,084	47	16	16 x 5	23 x 2	2,586	8,421
# examples	203,029	455,954	125,510	437,929	6,515	448,000	523,846	19,669	539,502	150,000
Train example						<div>What do Black and LGBT people have to do with bicycle licensing?</div>			<div>Overall a solid package that has a good quality of construction for the price.</div>	<div><code>import numpy as np</code> ... <code>norm=np.____</code></div>
Test example						<div>As a Christian, I will not be patronizing any of those businesses.</div>			<div>I *loved* my French press, it's so perfect and came with all this fun stuff!</div>	<div><code>import subprocess as sp</code> <code>p=sp.Popen()</code> <code>stdout=p.____</code></div>
Adapted from	Beery et al. 2020	Bandi et al. 2018	Taylor et al. 2019	Hu et al. 2020	David et al. 2021	Borkan et al. 2019	Christie et al. 2018	Yeh et al. 2020	Ni et al. 2019	Raychev et al. 2016

Quiz 7

- A. What does it mean for a machine learning model to be fair? Can a model be accurate but unfair?
- B. How can historical biases in data lead to unfair model behavior, even if the algorithm doesn't explicitly use sensitive data?
- C. What does it mean for a model to be robust? Describe a scenario where lack of robustness could cause real-world problems.
- D. What are the risks of deploying a non-robust model in a high-stakes environment like healthcare or finance? How would you mitigate them?