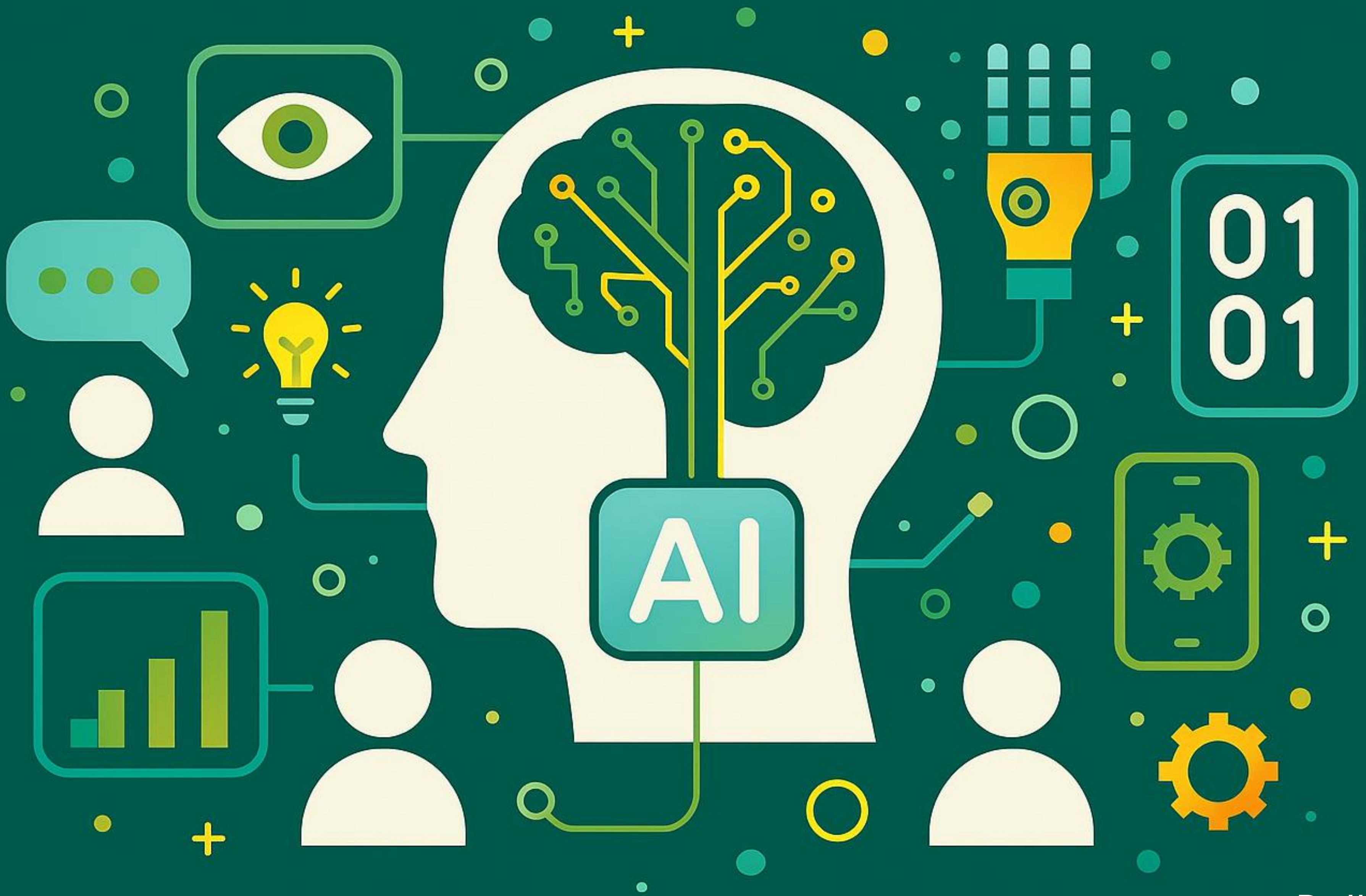
Model evaluation



Dr Ibrahim Alsaggaf

Attribution Requirement

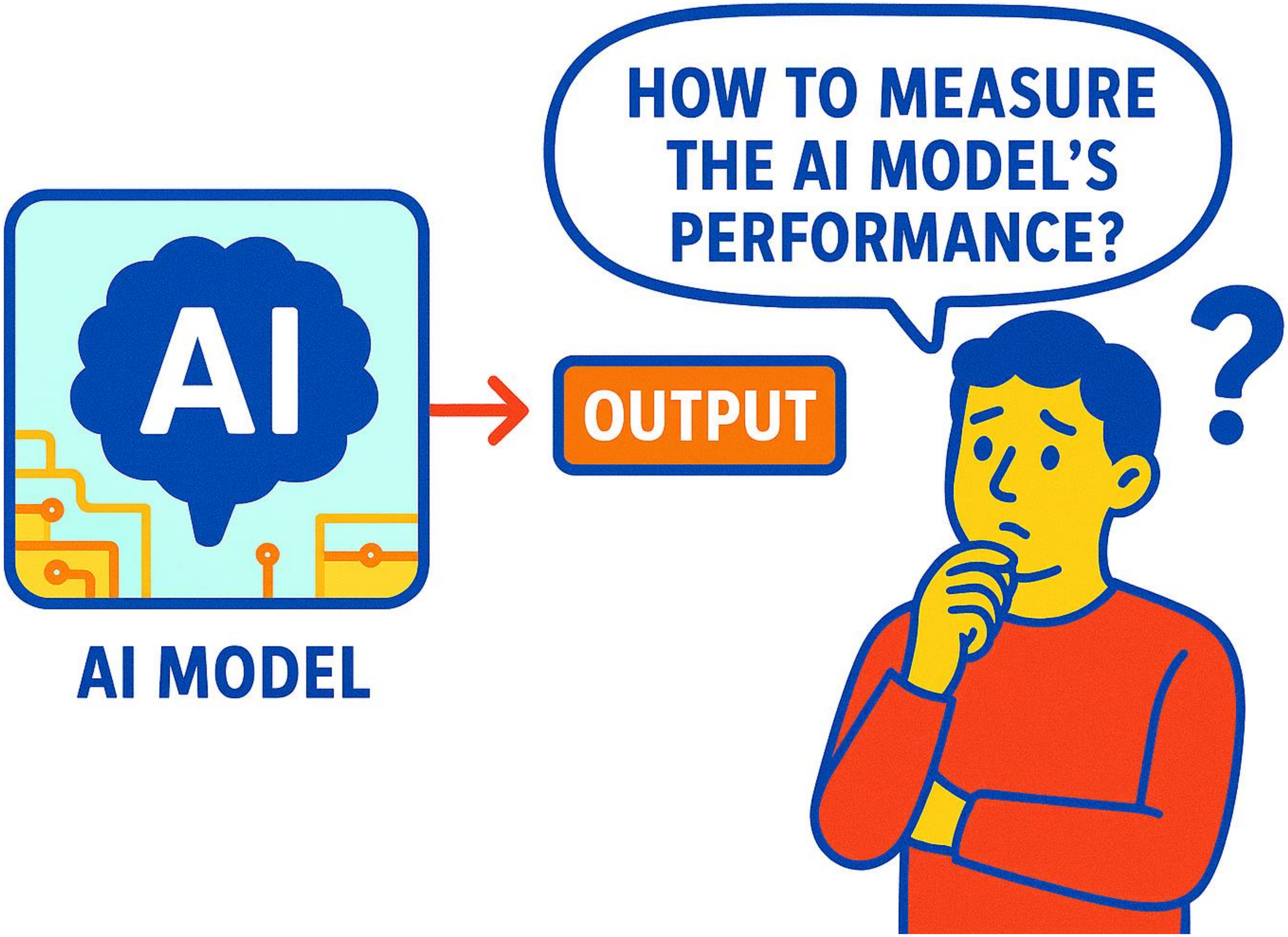
When using, sharing, or adapting this document for any individual, group, or organisation, proper citation of the original author is required. Please cite as follows:

Alsaggaf, I. (2025) *Introduction to Artificial Intelligence*. Available at: https://github.com/ibrahimsaggaf/Introduction-to-Artificial-Intelligence (Accessed: [insert date]).

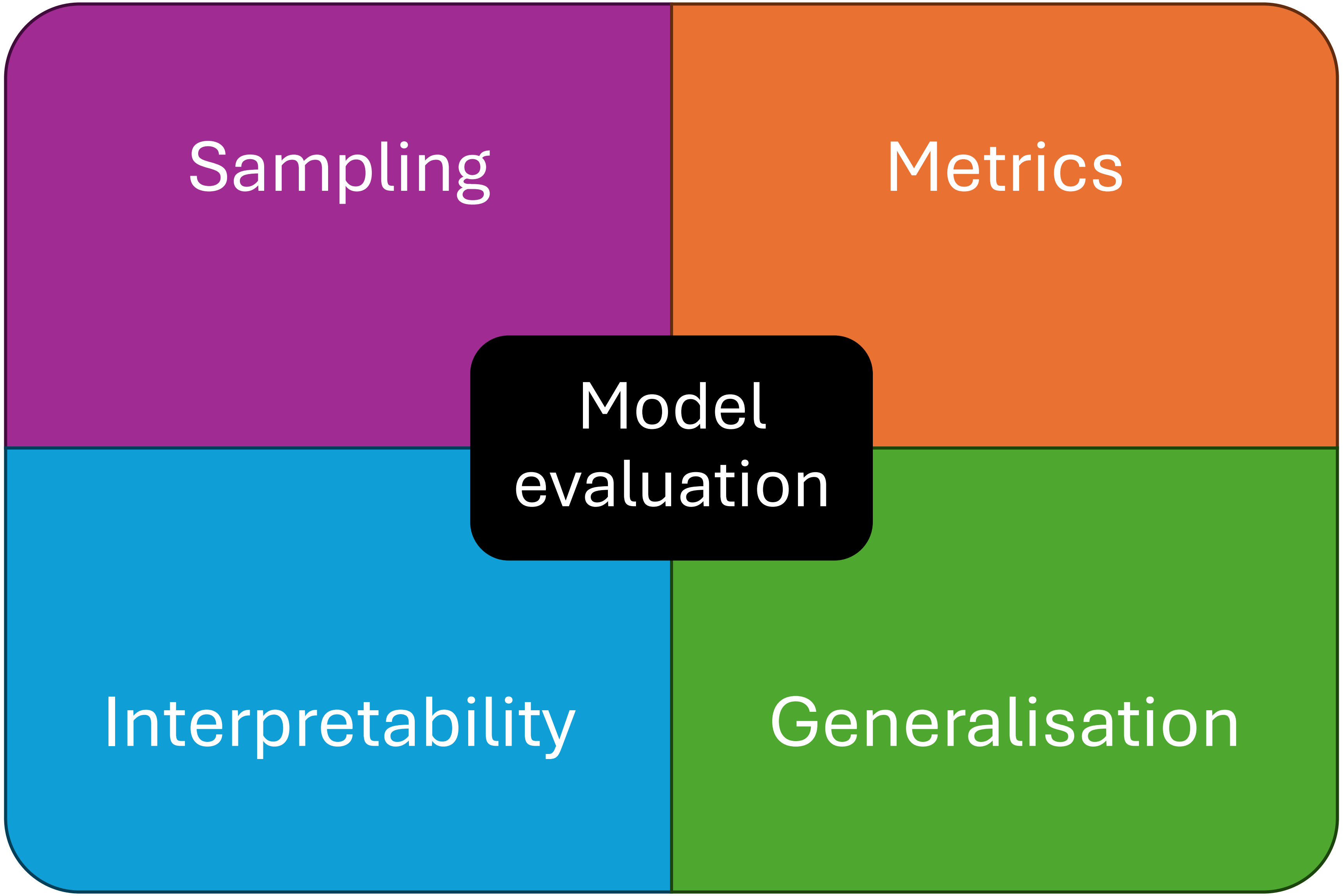
Content

- Classification metrics
- Regression metrics
- Bias-variance trade-off
- Sampling
- Q&A

Lab session: Model evaluation in classification settings



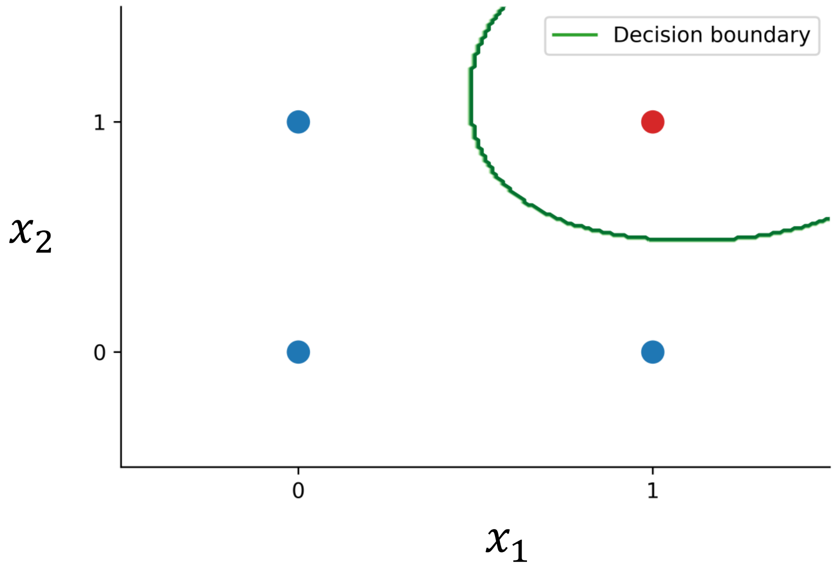
Generated using https://copilot.microsoft.com


Sampling
Metrics
Model evaluation
Interpretability
Generalisation

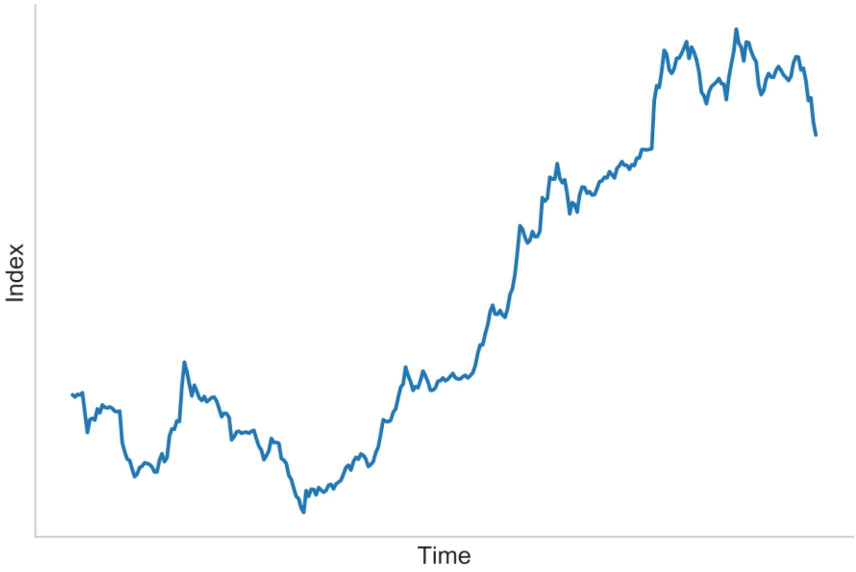
AND problem

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

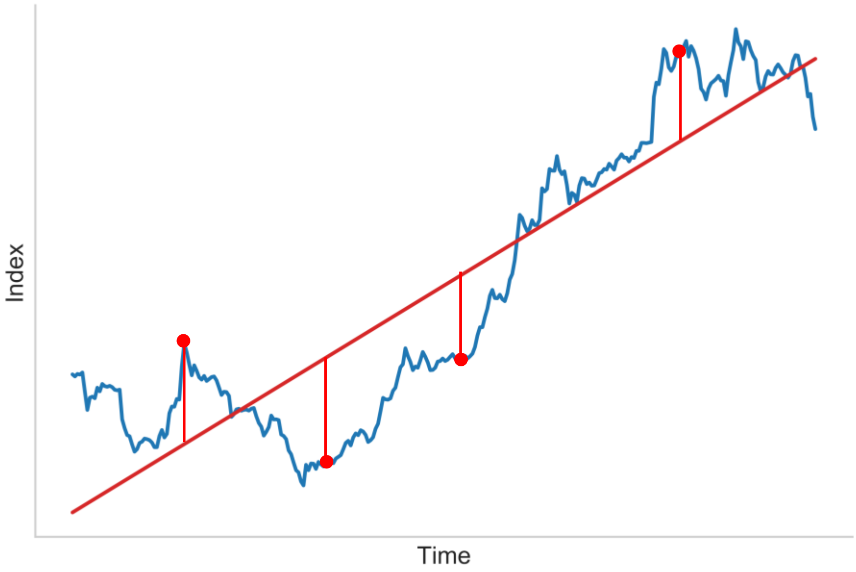
Classification tasks
(y is categorical)



VS



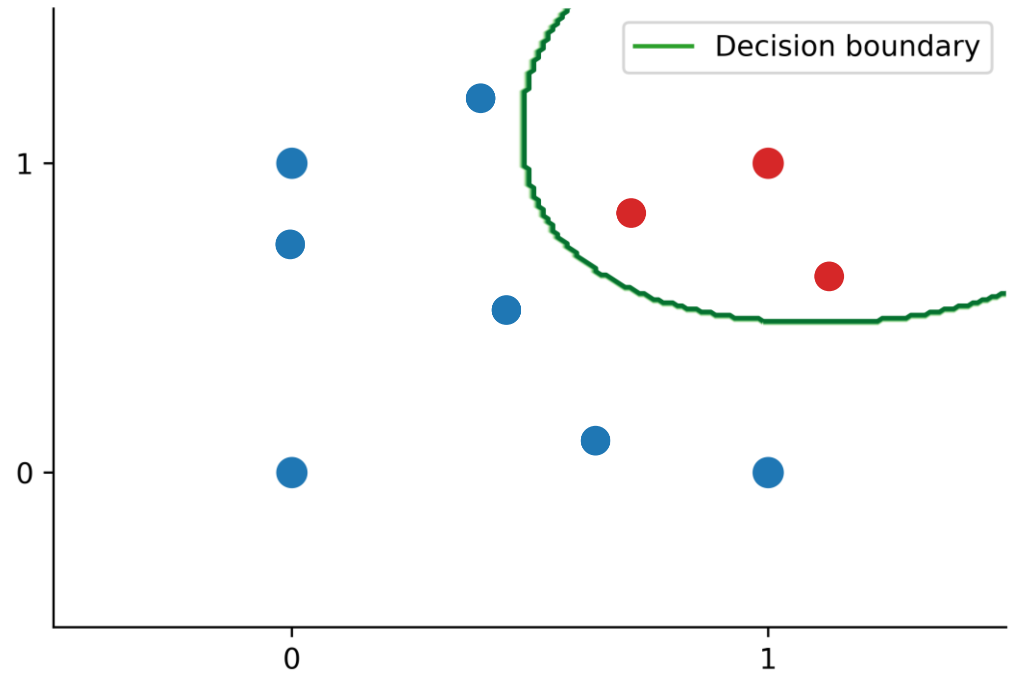
Regression tasks
(y is continuous)



Classification metrics

$$Accuracy = \frac{number\ of\ correctly\ classified\ instances}{Total\ number\ of\ instance}$$

$$Error = \frac{number\ of\ incorrectly\ classified\ instances}{Total\ number\ of\ instance}$$



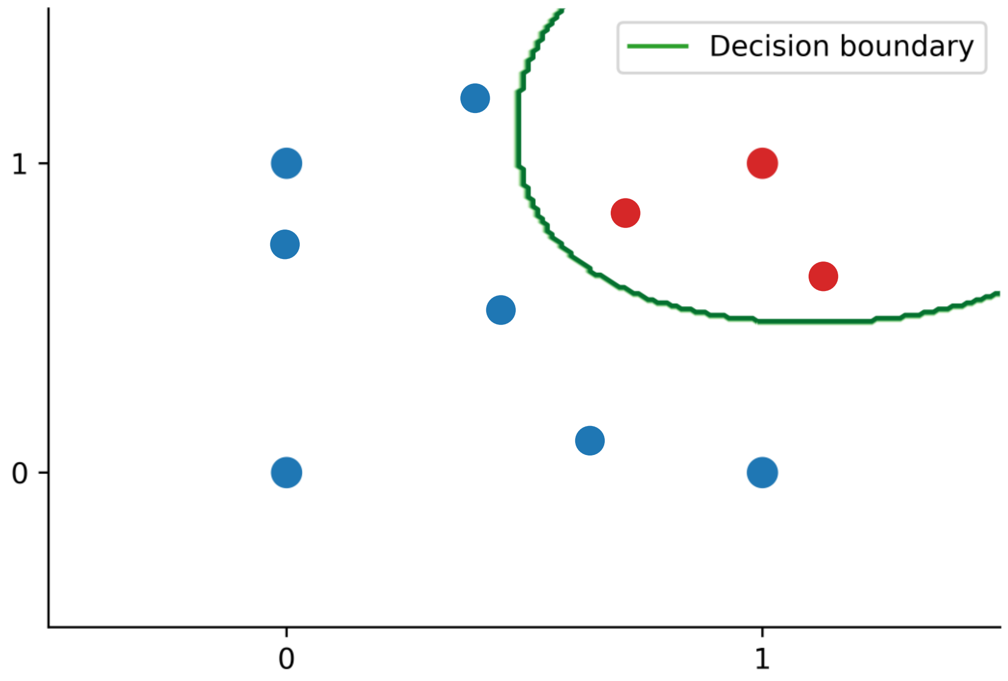
Classification metrics

$$Accuracy = \frac{number\ of\ correctly\ classified\ instances}{Total\ number\ of\ instance}$$

$$Error = \frac{number\ of\ incorrectly\ classified\ instances}{Total\ number\ of\ instance}$$

$$Accuracy = \frac{10}{10} = 1.0$$

$$Error = \frac{0}{10} = 0.0$$



Classification metrics

Confusion matrix in binary classification

		Prediction	
		1	0
Truth	1	TP	FN
	0	FP	TN

TP: True Positive

FN: False Negative

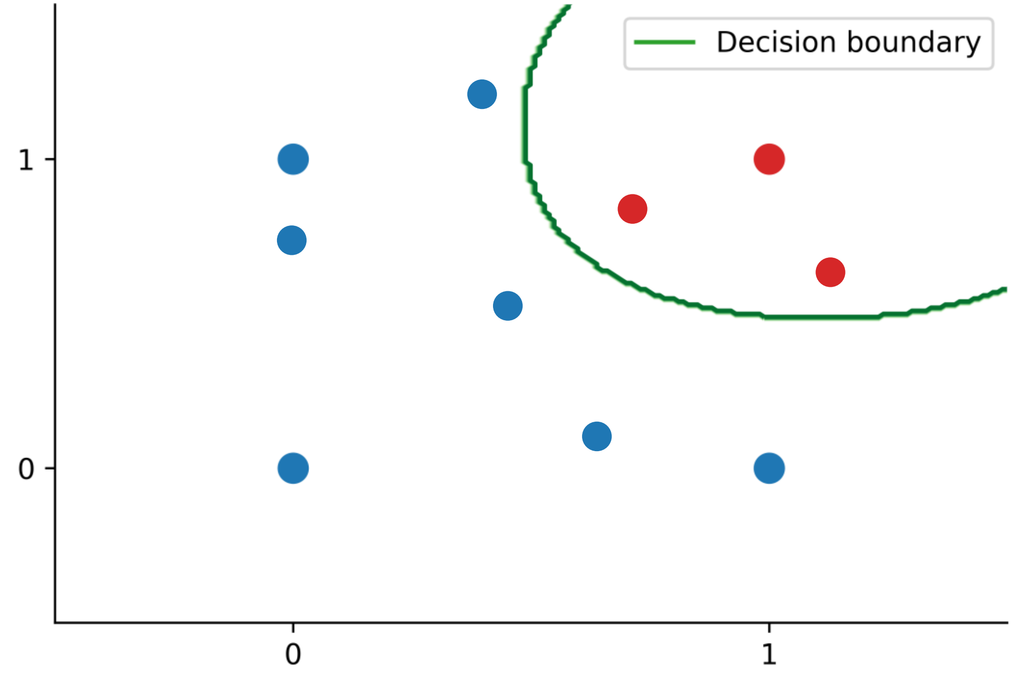
FP: False Positive

TN: True Negative

Classification metrics

Confusion matrix in binary classification

		Prediction	
		1	0
Truth	1	3	0
	0	0	7



Classification metrics

Confusion matrix in binary classification

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Error = \frac{FP + FN}{TP + FP + TN + FN}$$

$$Accuracy = \frac{3 + 7}{3 + 0 + 7 + 0} = \frac{10}{10} = 1.0$$

$$Error = \frac{0 + 0}{3 + 0 + 7 + 0} = \frac{0}{10} = 0.0$$

		Prediction	
		1	**0**
Truth	**1**	3	0
	0	0	7

Classification metrics

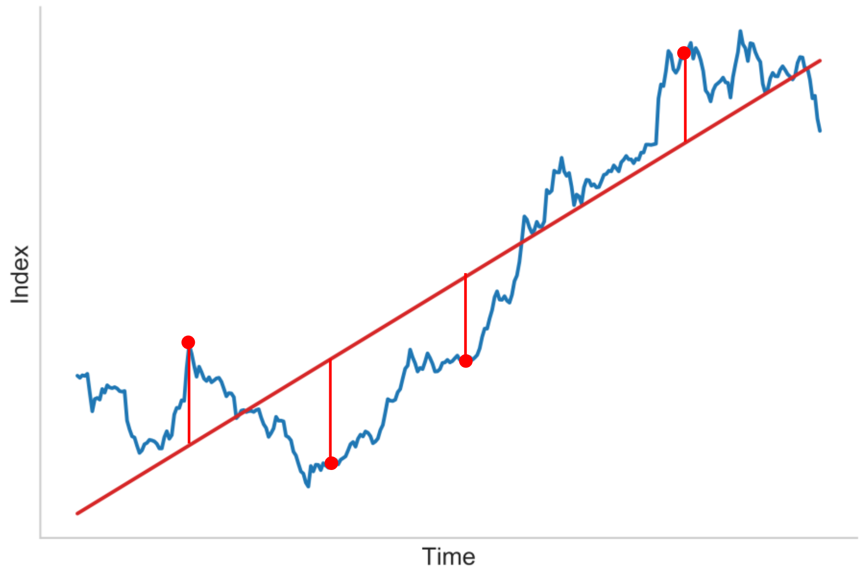
When to use

Metric	Binary	Multi-class	Imbalanced data
Accuracy	✓	✓	
F1	✓	✓	✓
Precision	✓	✓	✓
Recall	✓	✓	✓
AUC	✓		✓
MCC	✓	✓	✓
…			

Regression metrics

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Where y_i is the i^{th} true value and $\hat{y}_i$ is its prediction



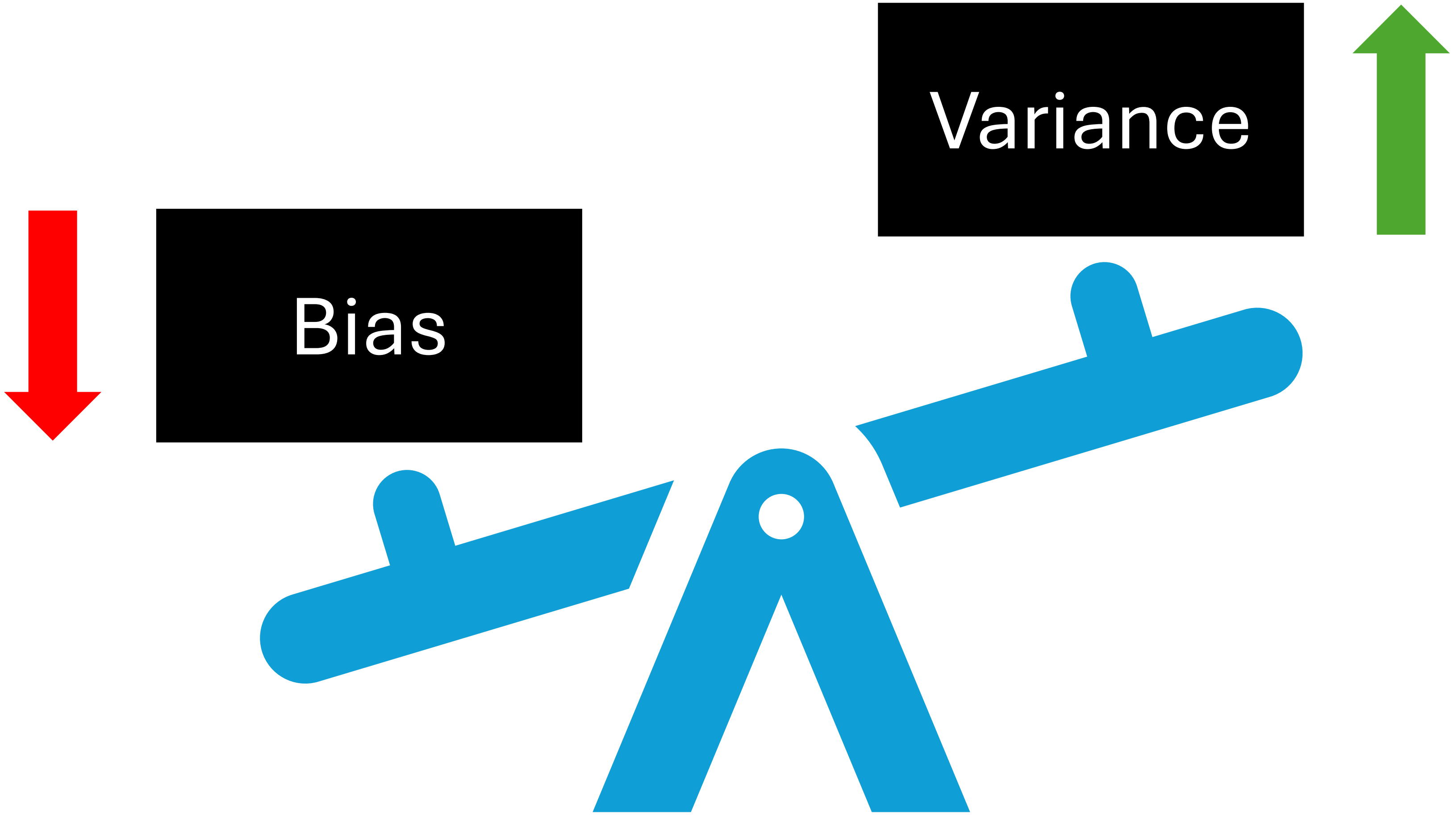
When to use

Metric	High penalty	Outliers
MSE	✓	
RMSE	✓	
MAE		✓
R-squared		✓
…		

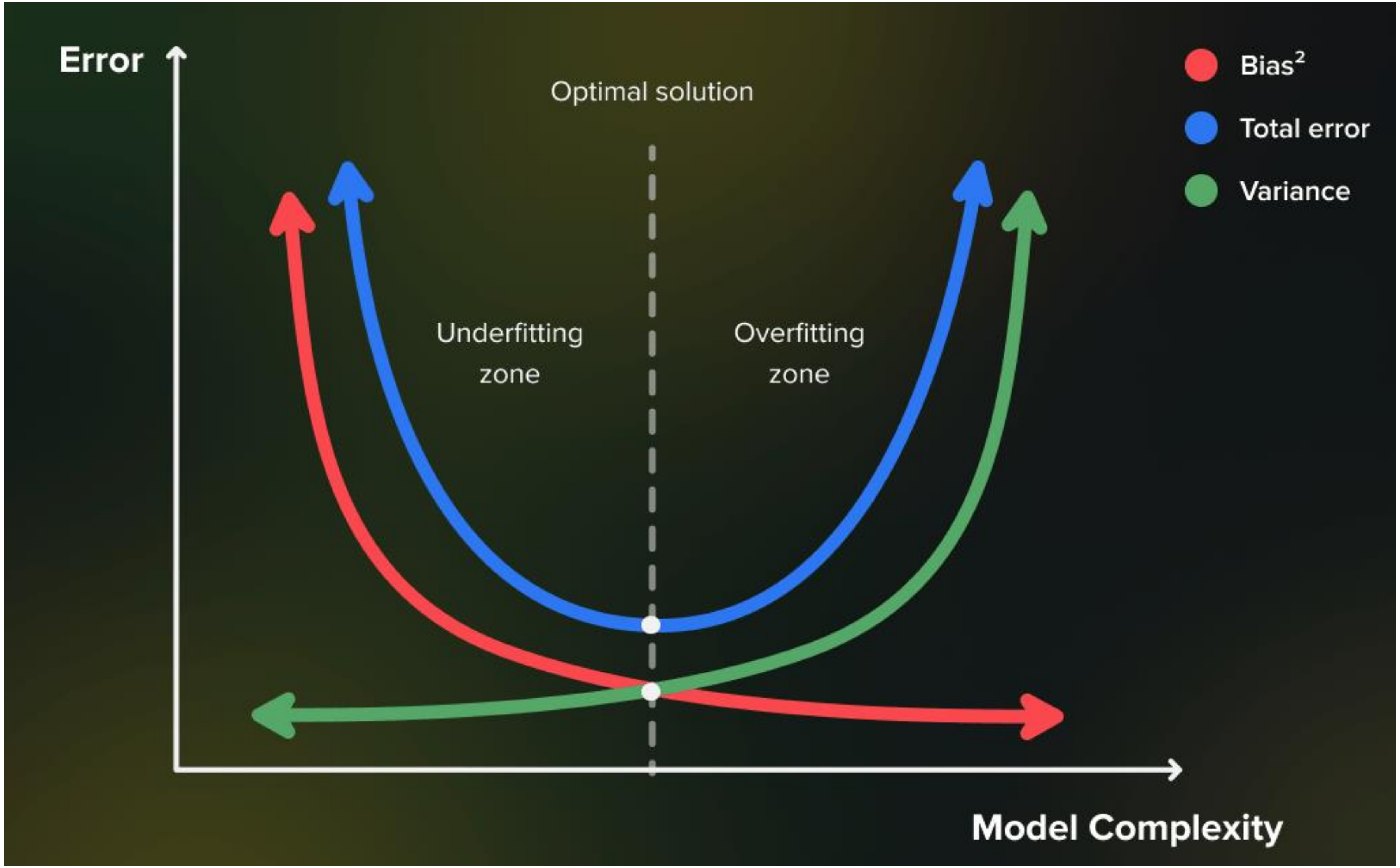
Bias-variance trade-off

Bias	Error
Variance	Sensitive to input

Bias-variance trade-off

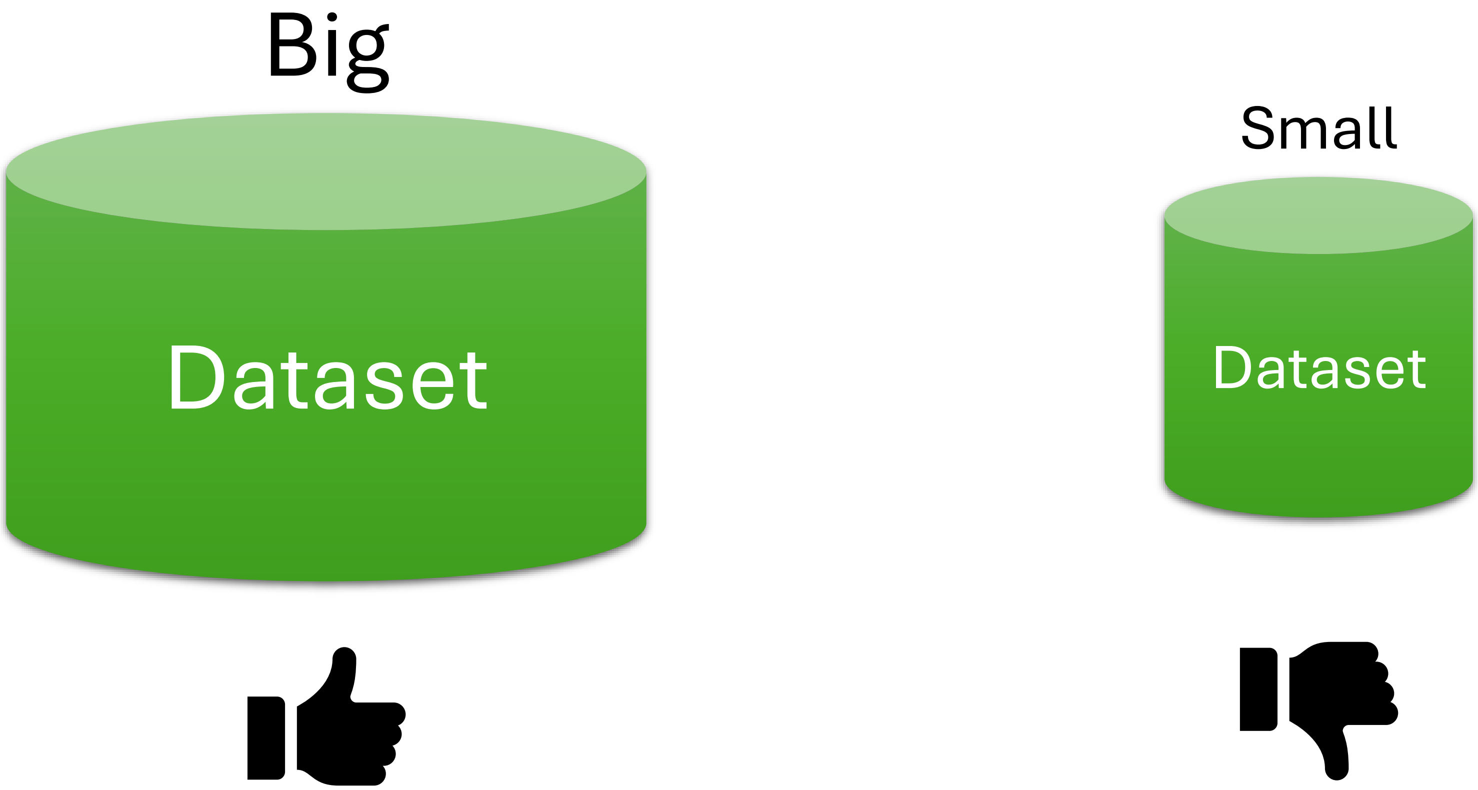


Bias-variance trade-off

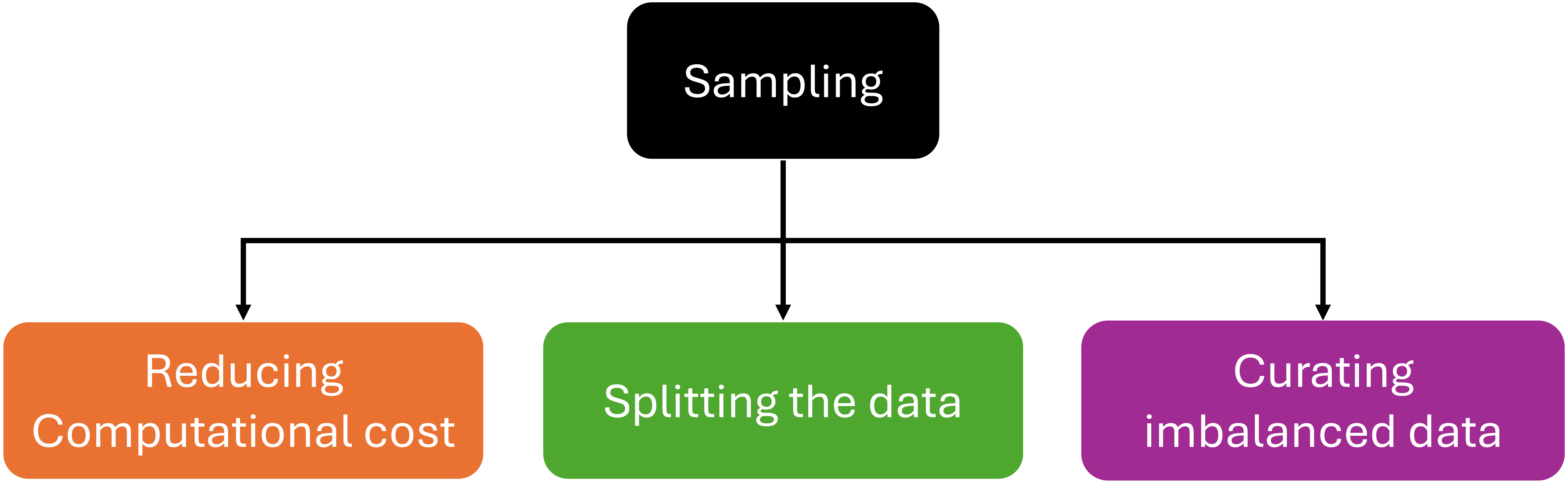


https://serokell.io/blog/bias-variance-tradeoff

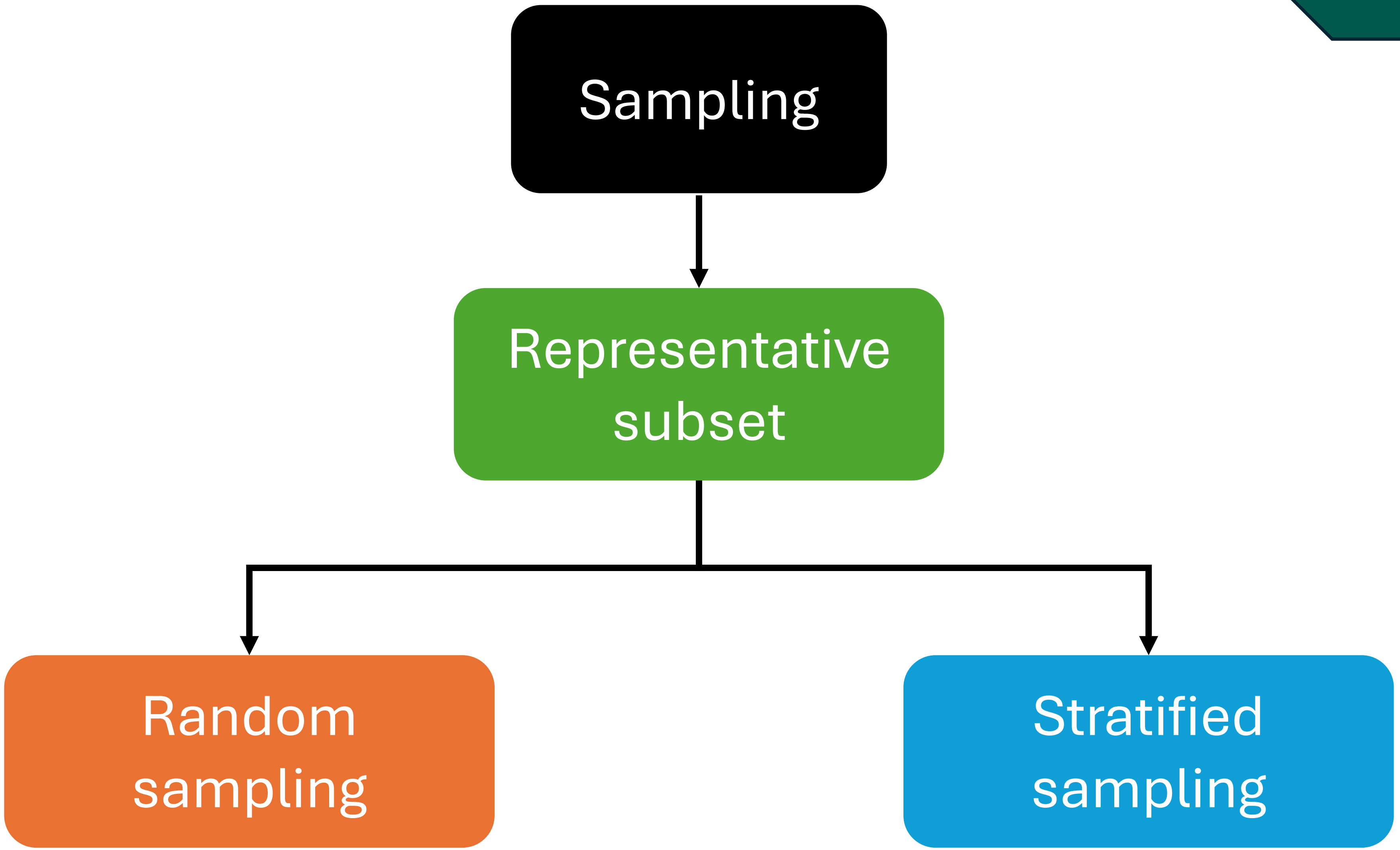
Sampling



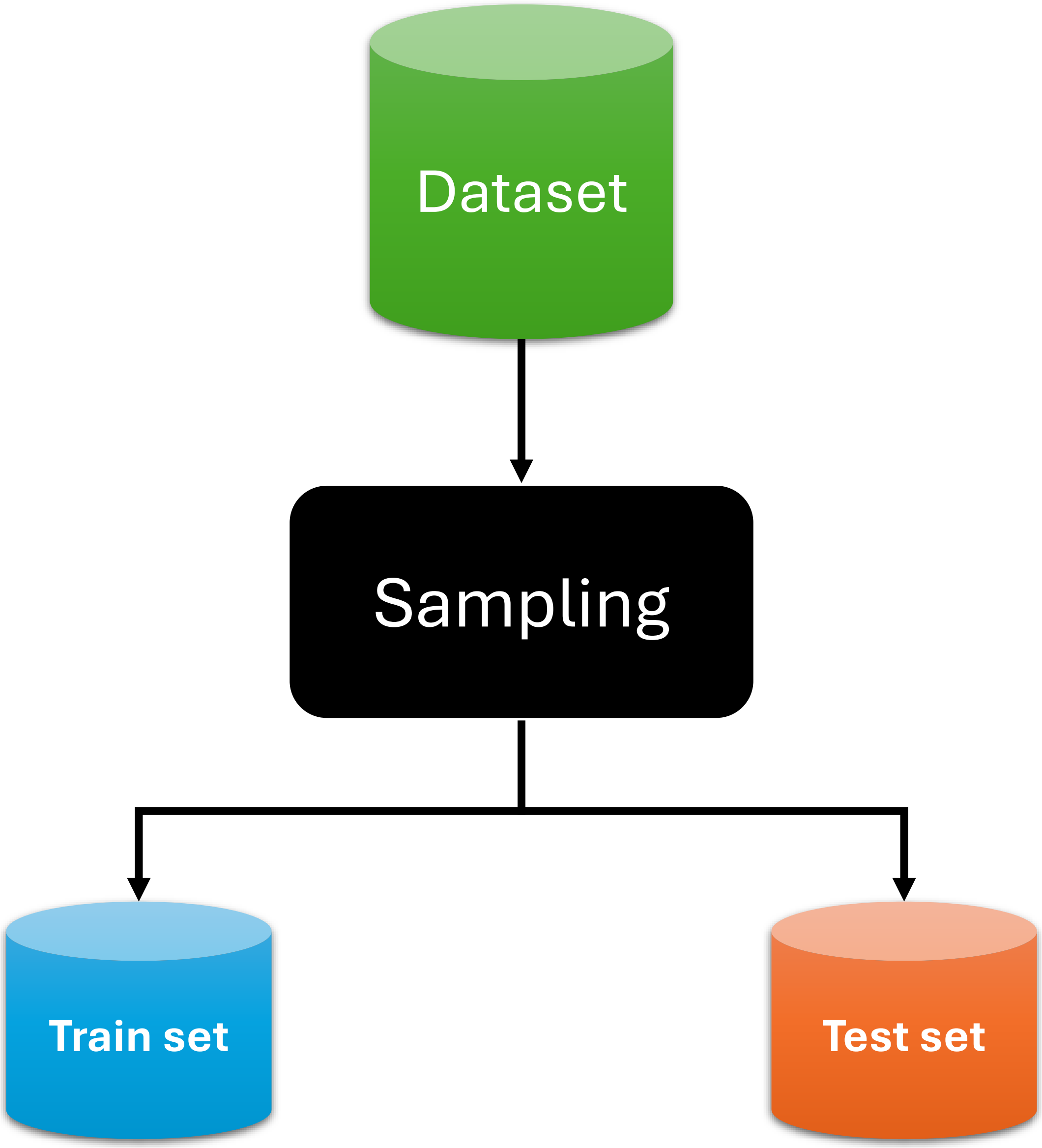
Why?


Sampling
Reducing
Computational cost
Splitting the data
Curating
imbalanced data

Sampling

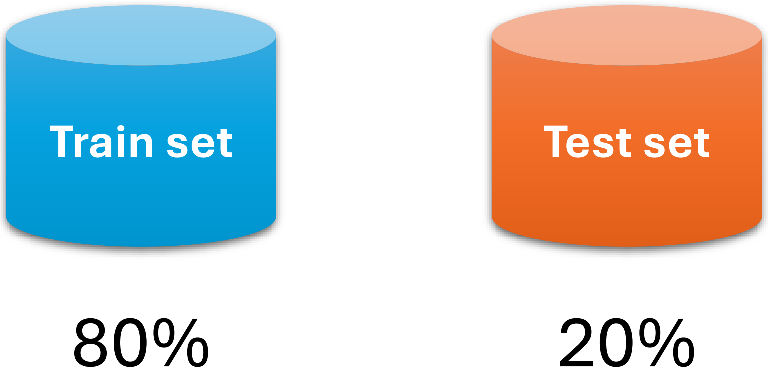

Sampling
Representative subset
Random sampling
Stratified sampling

Sampling

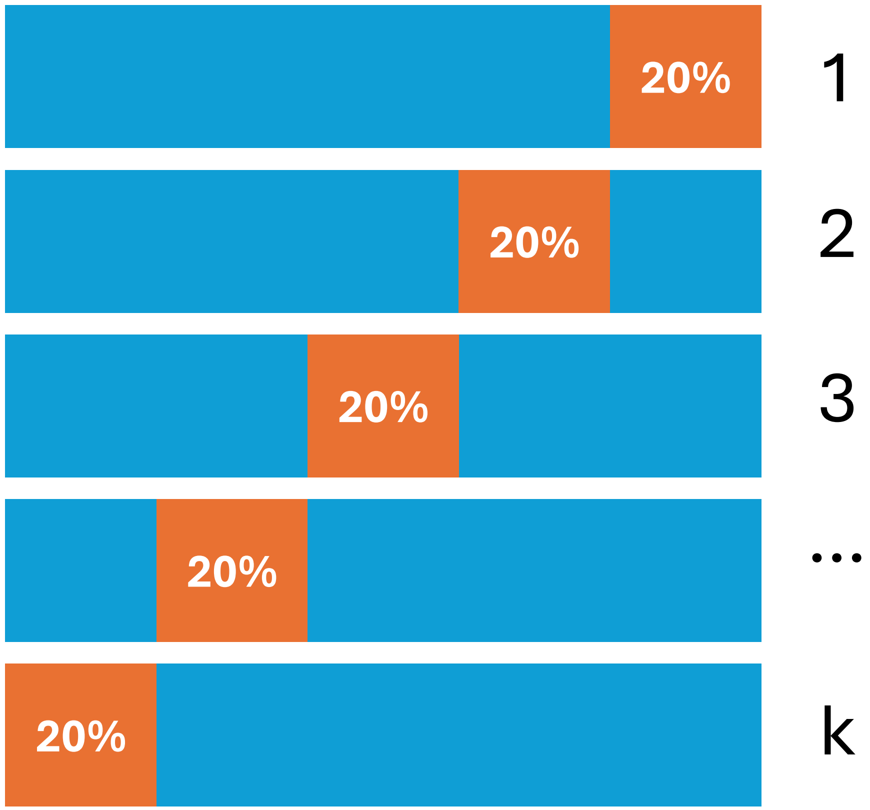

Dataset
Sampling
Train set
Test set

Sampling

Single split



K-fold cross validation



K-fold cross validation

Data size		Computational cost: Cheap	Computational cost: Expensive
	Small	5-folds	5-folds
	Big	10-folds	5-fold or 10-folds

Q&A

Lab Time

Lab 4: Model evaluation in classification settings

Lab 4: Dataset

MNIST Dataset

The MNIST database of handwritten digits

Download Raw Dataset

Dataset Statistics

1. Color: Grey-scale
2. Sample Size: 28x28

The number of categories of MNIST is 10, that is 0-9, 10 digits.

The Number of Samples per Category for MNIST

CATEGORY	0	1	2	3	4	5	6	7	8	9	TOTAL
#Training Samples	5,923	6,742	5,958	6,131	5,842	5,421	5,918	6,265	5,851	5,949	60,000
#Testing Samples	980	1,135	1,032	1,010	982	892	958	1,028	974	1,009	10,000

Samples

Step 1

- Download the Lab4 directory from the GitHub repository
  https://github.com/ibrahimsaggaf/Introduction-to-Artificial-Intelligence
- Open the Lab4 directory in Visual Studio Code.
- The Lab4 directory contains 5 files:
  - main.py
  - model.py
  - network.py
  - utils.py
  - requirements.txt

Take your time examining these files.

Step 2

- Download the MNIST dataset from https://drive.google.com/file/d/1eEKzfmEu6WKdRlohBQiqi3PhW_uIVJVP/view
- Unzip the downloaded file MNIST_CSV.zip and move the 2 csv files into the lab4 directory:
  - train_mnist.csv ($\approx 104\ MB$)
  - test_mnist.csv ($\approx 17.4\ MB$)

Step 3

- Create and activate a virtual environment under the name "lab4_env" (see Lab 1)
- Install the below libraries inside the virtual environment using a requirements file:
  - ❑ Scikit-learn
  - ❑ Deep learning library Pytorch
  - ❑ Visualisation library Matplotlib

By running the following command:
pip install -r requirements.txt

Step 4

- Run the command:
  python main.py

This command evaluates a model's performance in multi-class classification settings:

1. Load the training and testing MNIST datasets.
2. Create a small and large MLP (Neural Networks) models.
3. Train the model on the training set and measure the loss (error) in both training and testing sets using Cross Entropy loss.
4. Plot the learning curves and save the figure.

File Edit Selection View Go ··· Lab4

EXPLORER

LAB4
- __pycache__
- lab4_env
- learning_curve.jpg
- main.py
- model.py
- network.py
- requirements.txt
- test_mnist.csv
- train_mnist.csv
- utils.py

OUTLINE

TIMELINE

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE powershell

```
(lab4_env) PS G:\My Drive\Training course\Introduction to Artificial Intelligence\Model evaluation\Lab4> python main.py
Small>> Epoch: 0, Train loss: 0.5499, Test loss: 0.3359
Small>> Epoch: 1, Train loss: 0.3289, Test loss: 0.2958
Small>> Epoch: 2, Train loss: 0.3000, Test loss: 0.2819
Small>> Epoch: 3, Train loss: 0.2862, Test loss: 0.2752
Small>> Epoch: 4, Train loss: 0.2776, Test loss: 0.2713
Small>> Epoch: 5, Train loss: 0.2717, Test loss: 0.2689
Small>> Epoch: 6, Train loss: 0.2672, Test loss: 0.2674
Small>> Epoch: 7, Train loss: 0.2636, Test loss: 0.2663
Small>> Epoch: 8, Train loss: 0.2607, Test loss: 0.2656
Small>> Epoch: 9, Train loss: 0.2582, Test loss: 0.2651
Small>> Epoch: 10, Train loss: 0.2561, Test loss: 0.2648
Small>> Epoch: 11, Train loss: 0.2542, Test loss: 0.2647
Small>> Epoch: 12, Train loss: 0.2526, Test loss: 0.2646
Small>> Epoch: 13, Train loss: 0.2511, Test loss: 0.2647
Small>> Epoch: 14, Train loss: 0.2498, Test loss: 0.2647
Small>> Epoch: 15, Train loss: 0.2486, Test loss: 0.2649
Small>> Epoch: 16, Train loss: 0.2475, Test loss: 0.2651
Small>> Epoch: 17, Train loss: 0.2465, Test loss: 0.2653
Small>> Epoch: 18, Train loss: 0.2456, Test loss: 0.2656
Small>> Epoch: 19, Train loss: 0.2447, Test loss: 0.2658
Small>> Epoch: 20, Train loss: 0.2439, Test loss: 0.2661
Small runtime: 0.4171 minutes
Large>> Epoch: 0, Train loss: 0.2103, Test loss: 0.1362
Large>> Epoch: 1, Train loss: 0.0894, Test loss: 0.0943
Large>> Epoch: 2, Train loss: 0.0571, Test loss: 0.0934
Large>> Epoch: 3, Train loss: 0.0423, Test loss: 0.1053
Large>> Epoch: 4, Train loss: 0.0312, Test loss: 0.0980
Large>> Epoch: 5, Train loss: 0.0244, Test loss: 0.1060
Large>> Epoch: 6, Train loss: 0.0244, Test loss: 0.0888
Large>> Epoch: 7, Train loss: 0.0183, Test loss: 0.1008
Large>> Epoch: 8, Train loss: 0.0159, Test loss: 0.0966
Large>> Epoch: 9, Train loss: 0.0149, Test loss: 0.0923
Large>> Epoch: 10, Train loss: 0.0140, Test loss: 0.0954
Large>> Epoch: 11, Train loss: 0.0110, Test loss: 0.0893
Large>> Epoch: 12, Train loss: 0.0119, Test loss: 0.0887
Large>> Epoch: 13, Train loss: 0.0092, Test loss: 0.1008
Large>> Epoch: 14, Train loss: 0.0086, Test loss: 0.1006
```

⊗ 0 ⚠ 0 Signed out

Step 5

- Inspect the printed output along with the generated figure labelled "learning_curve.jpg".

Lab 4: Model evaluation in classification settings

Congrats! 👍

[✓] Train a small and large MLP (Neural Networks) models

[✓] Model evaluation in classification settings

[✓] Plotting training and testing learning curves

[✓] Underfitting and overfitting investigation

Quiz 3

Q1: Which of the following classification metrics is most resilient to class imbalance?

A) Coefficient of determination
B) Mean Squared Error
C) Matthews Correlation Coefficient
D) None of the above

Q2: In this lab, what conclusions can be drawn from the learning curves generated by the give code?

A) The loss values obtained with large models are lower than those obtained with small models

B) With respect to the bias–variance trade-off, large models reach the optimal solution much faster than small models

C) As training progresses, large models are more prone to overfitting compared to small models

D) All of the above

Reading list

Bias-Variance Tradeoff in Machine Learning
https://serokell.io/blog/bias-variance-tradeoff

An Overview of Classification Model Metrics
https://medium.com/@ml_dl_explained/an-overview-of-classification-model-metrics-8e25432d36ea

Regression Metrics for Machine Learning
https://machinelearningmastery.com/regression-metrics-for-machine-learning/