# Artificial Intelligence

Taro Sekiyama

National Institute of Informatics (NII)
sekiyama@nii.ac.jp

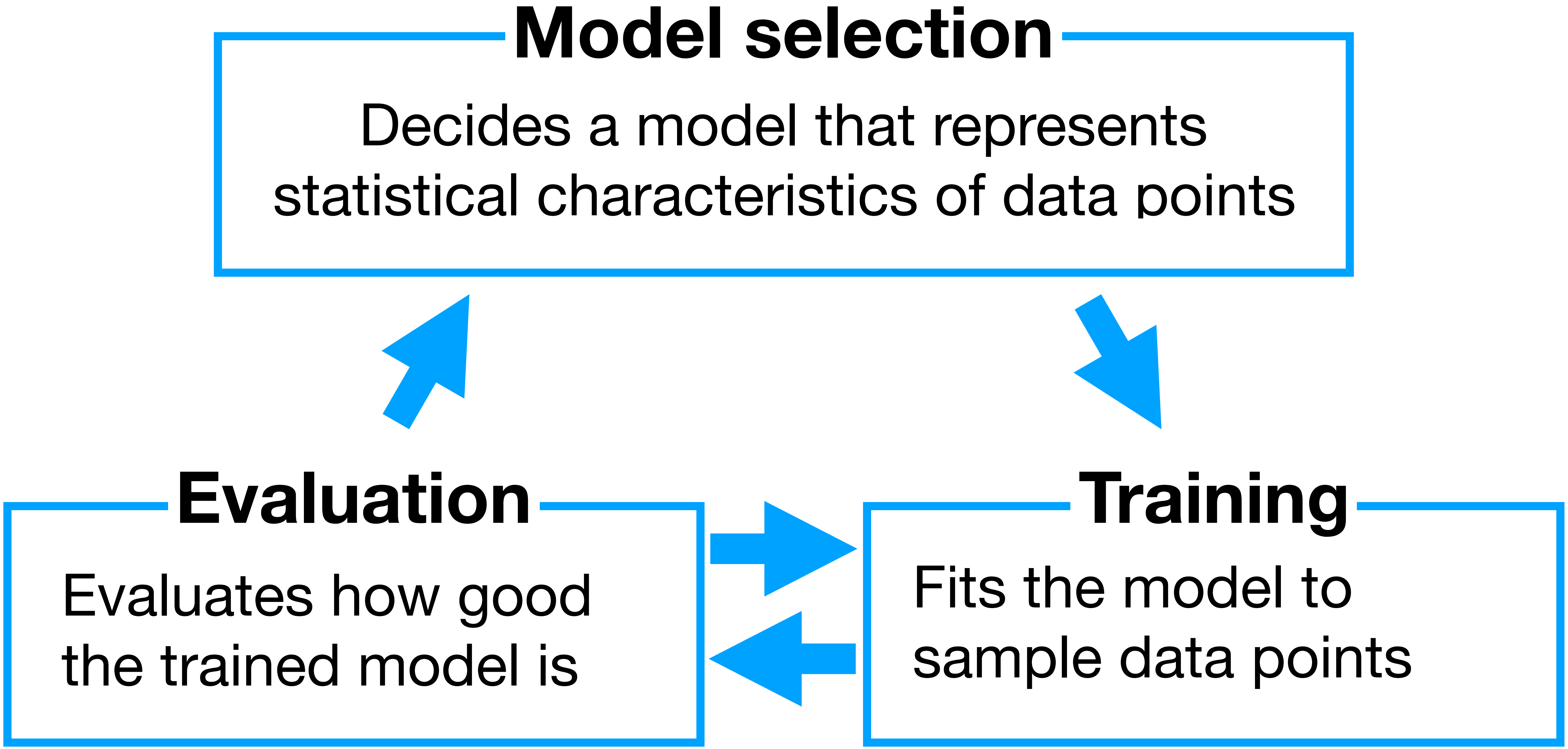
# Agenda

- Development phases for machine learning
- K-nearest neighbors algorithm
- Case studies

# Machine learning (ML)

- Technology to learn the statistical characteristics of data points

# Development cycle for ML



**Model selection**

Decides a model that represents statistical characteristics of data points

**Evaluation**

Evaluates how good the trained model is

**Training**

Fits the model to sample data points

# Model selection

- Decides how to capture the statistical characteristics of sample data
- We have to select:
    - Machine learning algorithms
    - Options (a.k.a. hyperparameters) of the selected algorithm
- The performance of a model depends on a task and a dataset
    - Important to investigate them deeply

# Training

- Fits the ***parameters*** of a selected ML model with a dataset (a set of sample data)
  - Parameters make the model so expressive that it can be applied to various tasks
  - Getting good performance needs optimization of the parameters for a specific task
- Parameters are fitted according to ***features*** of data points

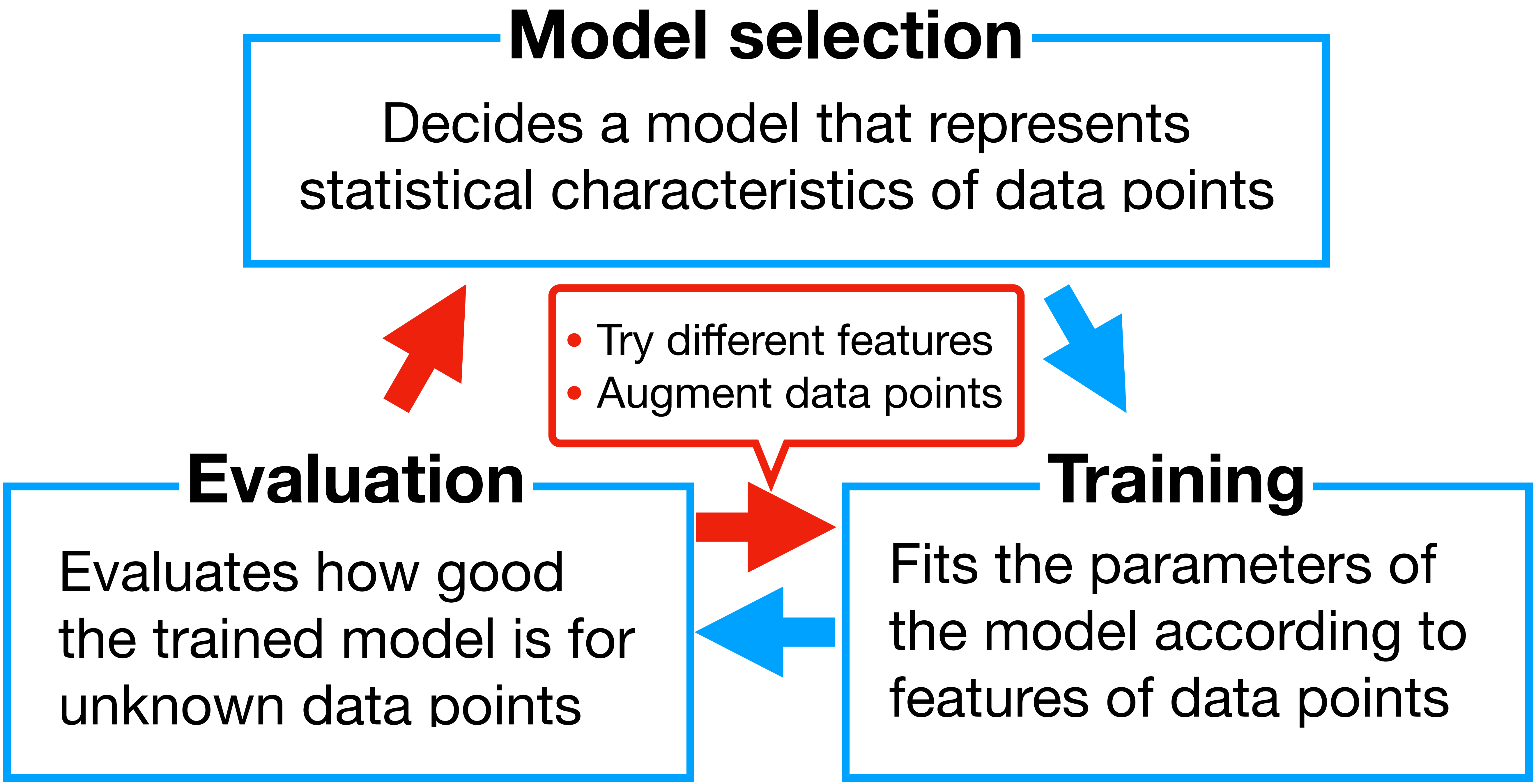
# Features

- Properties capturing the characteristics of data points
- Example
  - For images:
    width, height, RGB value for each pixel, …
  - For texts:
    word frequency, part of speech, length, …
- Often represented by n-dimensional vectors over real numbers (that is, elements in $\mathbb{R}^n$)

# Evaluation

- Tests the performance of the trained model for data ***not appearing in the training***
  - Important to check how good the model is for unknown data
- Metrics for evaluation
  - Accuracy for classification
    - The number of correctly predicted data out of all
  - Root Mean Square Error (RMSE) for regression

# Development cycle for ML



**Model selection**

Decides a model that represents statistical characteristics of data points

- Try different features
- Augment data points

**Training**

Fits the parameters of the model according to features of data points

**Evaluation**

Evaluates how good the trained model is for unknown data points

# Case study

Task: Classification of flower species

- Dataset: Iris flower dataset
- Algorithm: K-nearest neighbors (K-NN)
- Evaluation metric: accuracy

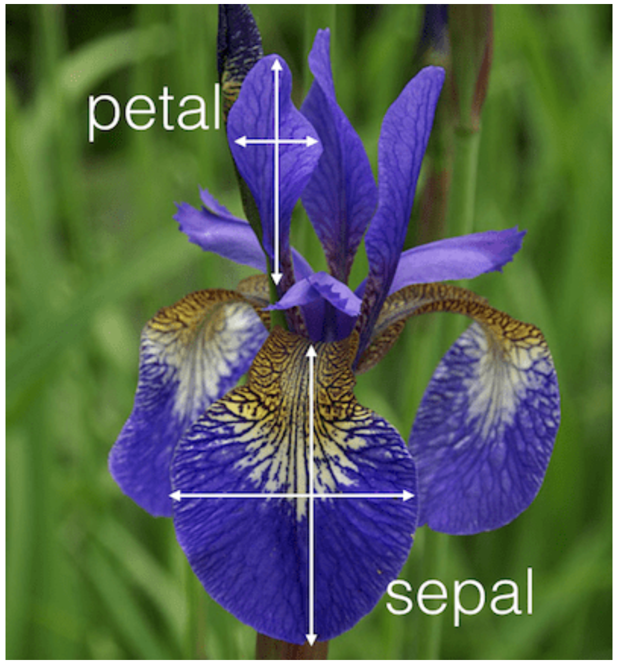
# Classification dataset

- A dataset of classification consists of pairs (X, Y) such that
  - X: inputs to a classifier
    - Features identifying a data point
  - Y: Expected outputs from a classifier
    - Category of X (called ***label***)

# Iris flower dataset

- 150 data points

- Input X is a tuple of:
  (1) sepal length (2) sepal width
  (3) petal length  (4) petal width



http://blog.kaggle.com/2015/04/22/scikit-learn-video-3-machine-learning-first-steps-with-the-iris-dataset/

- Output Y is a number denoting one of:
  (1) setosa
  (2) versicolor
  (3) virginica

Iris Setosa

Iris Versicolor

Iris Virginica

https://inclass.kaggle.com/alexisbcook/distributions

# Case study

Task: Classification of flower species

- Dataset: Iris flower dataset
- Algorithm: K-nearest neighbors (K-NN)
- Evaluation metric: accuracy

# K-nearest neighbors

Prediction by votes from training data points nearest to an input

- Training phase
  - Holding a given training dataset
- Classification phase
  - Returns the most frequent label among K training data points nearest to an input
    - K is a hyperparameter that controls how many data points join voting
    - Euclid distance is used to determine “nearest” usually

# Case study

Task: Classification of flower species

- Dataset: Iris flower dataset
- Algorithm: K-nearest neighbors (K-NN)
- Evaluation metric: accuracy
  - Calculates the number of correctly predicted data points in a test dataset

# Programming environment

- This course uses **Python** (ver. 3)
  - Jupyter notebook
    - Tool to enable interactive programming
  - numpy / scikit-learn / pandas / matplot
    - Useful libraries for ML programming
- **Anaconda** (ver. 2019.07) provides all in one package

# Outline

1. Basics of Python programming
2. Implementing the case study: classification of Iris species in Python

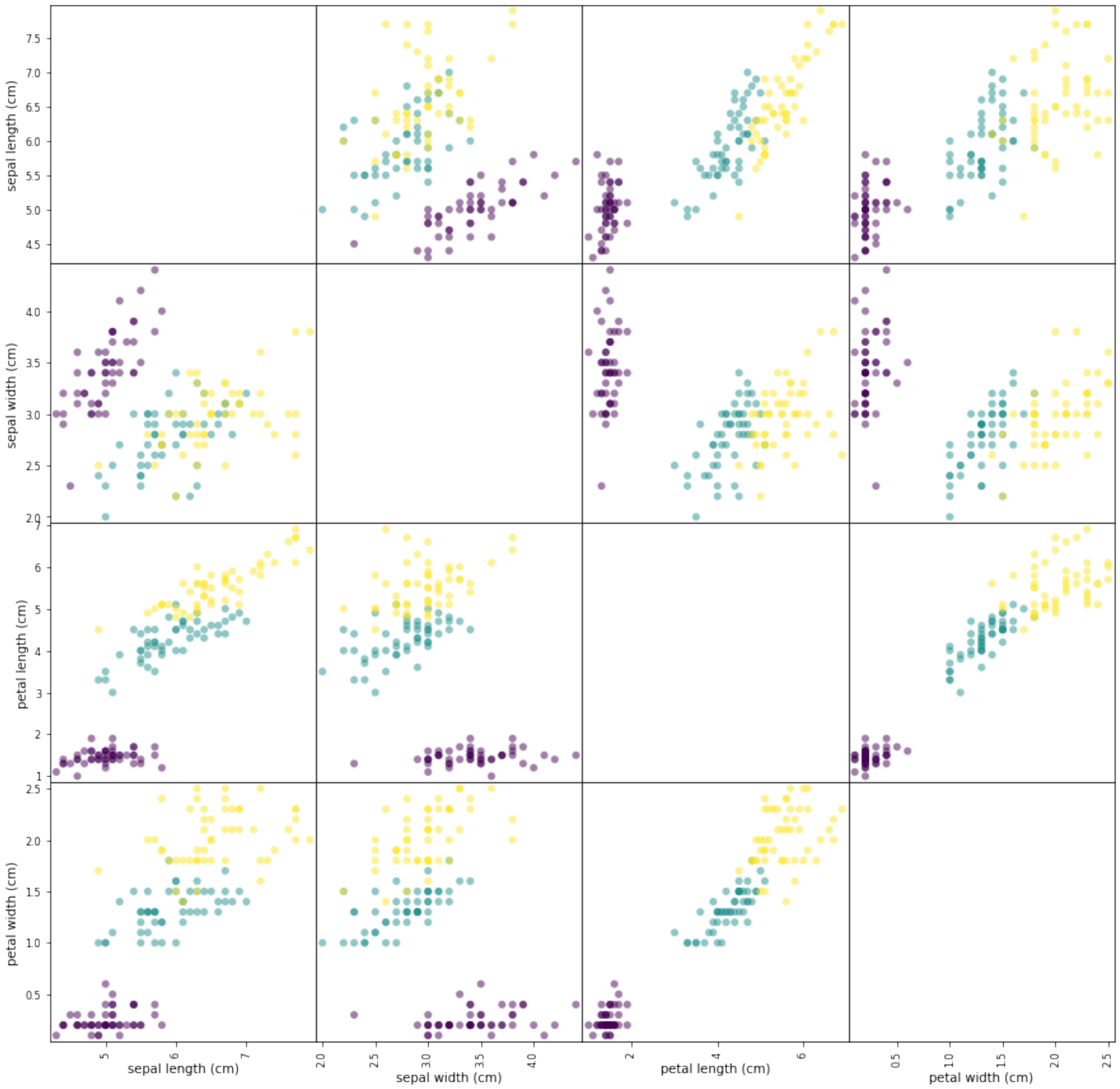
# Basics of Python

1. Primitive values: numbers, strings, Booleans
2. If / while statements
3. Data structures
    - Lists
    - Dictionaries
    - Objects
4. Use of libraries

■ FYI: Quick references are found by googling with “python cheat sheet”

# Why K-NN works well?

Thanks to the features capturing the characteristics of Iris data points well



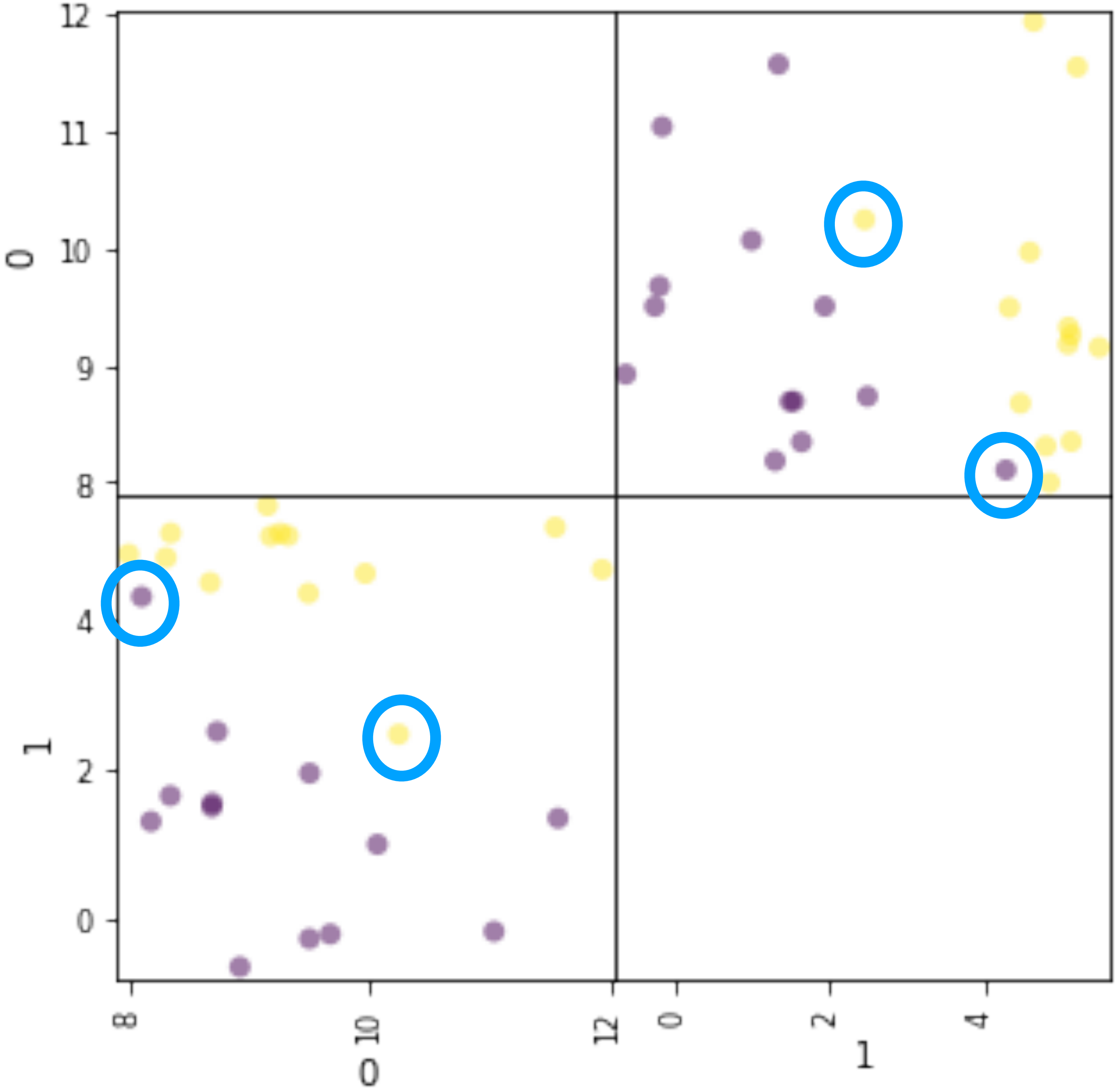
# Case study 2

Task: Classification of flower species

- Dataset: ***Forge dataset***
  - Need to install “mglearn” library
  - Type “pip install mglearn” in the terminal
- Algorithm: K-nearest neighbors (K-NN)
- Evaluation metric: accuracy

# Features in Forge

- There are found noisy data points



# Problem of K-NN

- Weak for noisy data points
- Weak for sparse data points with highly-dimensional features
  - Differences of highly-dimensional features are eliminated by conversion to a real number (low-dimensional repr.)
- Computational cost of prediction with a huge training dataset
  - K-NN needs to compute distances from all the training data points
  - Practical datasets consists of:
    - 1M~ data points
    - 1K~ features