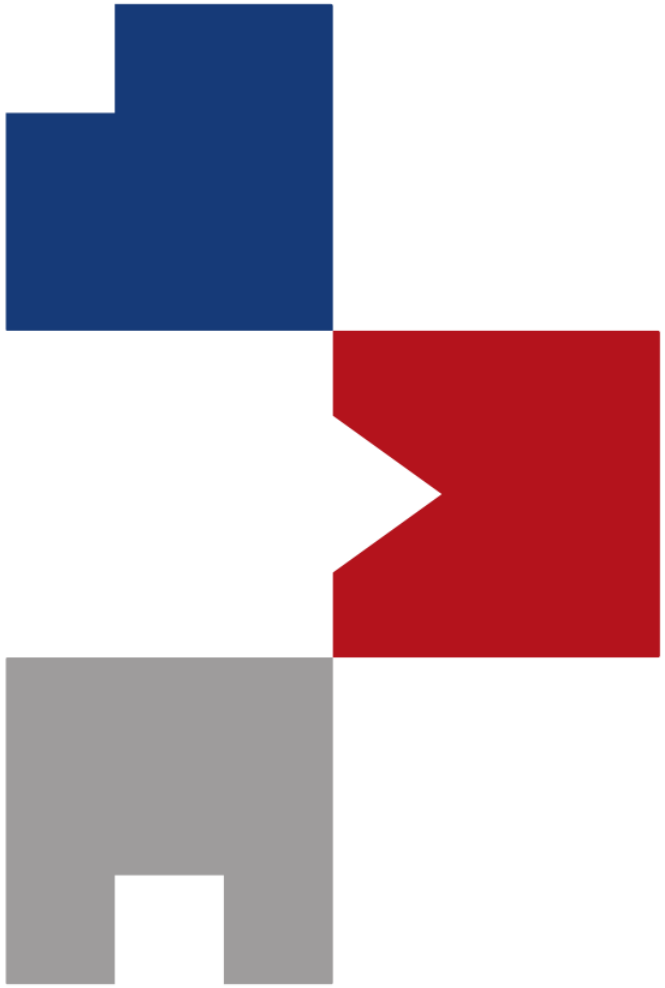
Note) Slides and example codes are available:
https://github.com/mint-lab/dl_tutorial



ML Lab #2:
Breast Cancer Classification with Cross-validation

Sunglok Choi, Assistant Professor, Ph.D.
Computer Science and Engineering Department, SEOULTECH
sunglok@seoultech.ac.kr | https://mint-lab.github.io/

Overview

- **Prerequisite**
  - Anacodna (Individual Edition)

- **Practice) Breast Cancer Classification with Cross-validation**
  - The given data
  - Expected results
  - Practice with the skeleton code
    - Step #1) Find your best classifier

- **Assignment**
  - Mission: Find your best classifier

Practice) Breast Cancer Classification

- The given data: [Breast Cancer Wisconsin (Diagnostic) Data Set]
  - Classes (#: **2**): *Malignant* (M; 악성종양 in Korean), *Benign* (B; 양성종양)
  - Attributes: **30** real numbers (except ID and target class)
    - Radius
    - Texture
    - Perimeter
    - Area
    - …
  - The number of data: **569** (M: 212, B: 357)
  - Note) Load the dataset using scikit-learn [API]

```
from sklearn import datasets
wdbc = datasets.load_breast_cancer()
```

Practice) Breast Cancer Classification

- Expected results
  - The default classifier: Decision tree (`tree.DecisionTreeClassifier`)
    - `Accuracy @ training data: 1.000`
    - `Accuracy @ test data: 0.919`
    - `Your score: 12`

- Evaluation (Total score: 20)
  - Your score = 10 + 100 x (your accuracy @ test data – 0.9)

Practice) Breast Cancer Classification

- The given skeleton code (`wdbc_classification_cv.py`)
  - Step #1) Find your best classifier

```python
import numpy as np
from sklearn import (datasets, tree, model_selection)

if __name__ == '__main__':
    # Load a dataset
    wdbc = datasets.load_breast_cancer()

    # Train a model
    model = tree.DecisionTreeClassifier() # TODO
    cv_results = model_selection.cross_validate(model, wdbc.data, wdbc.target, cv=5, return_train_score=True)

    # Evaluate the model
    acc_train = np.mean(cv_results['train_score'])
    acc_test = np.mean(cv_results['test_score'])
    print(f'* Accuracy @ training data: {acc_train:.3f}')
    print(f'* Accuracy @ test data: {acc_test:.3f}’)
    print(f'* Your score: {max(10 + 100 * (acc_test - 0.9), 0):.0f}')
```

Assignment

- Mission
  - Find your best classifier using the skeleton code (`wdbc_classification_cv.py`)
    - Note) Please think about a situation when training your model needs 1 hour or 1 day or 1 week or 1 month.
  - Submit your code (`wdbc_classification_cv.py`) and its accuracy (`wdbc_classification_cv.png`)

- Condition
  - Please follow the above filename convention.
  - You can start from scratch (without using the given skeleton code).
    - However, you should use the given data.
  - You can freely change the given skeleton code if necessary.

- Submission
  - Deadline: **November 15, 2022 23:59** (firm deadline; no extension)
  - Where: e-Class > Assignments
  - Score: Max 20 points