Due Date: See Webcampus
How to submit: Webcampus

General Guidelines:

- Please prepare a **typed** report that describes what you did. The report should be as concise as possible while providing all necessary information required to replicate your plots.
- For each problem, please provide, at the end of your report, a commented version of your python code files. **Python Notebook files are preferred. You may put the codes for all the problems in a SINGLE ipynb file with necessary texts to separate each problem.**

P3-1. Revisit Text Documents Classification
Use the 20 newsgroups dataset embedded in scikit-learn:

```
from sklearn.datasets import fetch_20newsgroups
```

(See https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html#sklearn.datasets.fetch_20newsgroups)

(a) Load the following 4 categories from the 20 newsgroups dataset: categories = ['rec.autos', 'talk.religion.misc', 'comp.graphics', 'sci.space'].

(b) Build classifiers using the following methods:

- Support Vector Machine (sklearn.svm.LinearSVC)
- Naive Bayes classifiers (sklearn.naive_bayes.MultinomialNB)
- K-nearest neighbors (sklearn.neighbors.KNeighborsClassifier)
- Random forest (sklearn.ensemble.RandomForestClassifier)
- AdaBoost classifier (sklearn.ensemble.AdaBoostClassifier)

Optimize the hyperparameters of these methods and compare the results of these methods.

P3-2. Recognizing hand-written digits
Use the hand-written digits dataset embedded in scikit-learn:

```
from sklearn import datasets
digits = datasets.load_digits()
```

(a) Develop a multi-layer perceptron classifier to recognize images of hand-written digits. To build your classifier, you can use:

sklearn.neural_network.MLPClassifier

(See https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier)

Instructions: use **sklearn.model_selection.train_test_split** to split your dataset into random train and test subsets, where you set **test_size=0.5**.

(b) Optimize the hyperparameters of your neural network to maximize the classification accuracy. Show the confusion matrix of your neural network. Discuss and compare your results

with the results using a support vector classifier (see https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html#sphx-glr-auto-examples-classification-plot-digits-classification-py).

P3-3. Nonlinear Support Vector Machine

(a) Randomly generate the following 2-class data points

```
import numpy as np
np.random.seed(0)
X = np.random.rand(300, 2)*10-5
Y = np.logical_xor(X[:, 0] > 0, X[:, 1] > 0)
```

(b) Develop a nonlinear SVM binary classifier (sklearn.svm.NuSVC).

(c) Plot these data points and the corresponding decision boundaries, which is similar to the figure in the slide 131 in Chapter 4.