Due Date: See Webcampus
How to submit: Webcampus

General Guidelines:

- Please submit the codes for all the problems in **a SINGLE ipynb file** with the necessary texts to separate each problem.
- Please submit the **pdf version** of your ipynb file.

P4-1. (U & G-required) Hierarchical Clustering Dendrogram

(a) Randomly generate the following data points:

```
import numpy as np
np.random.seed(0)
X1 = np.random.randn(50,2)+[2,2]
X2 = np.random.randn(50,2)+[6,10]
X3 = np.random.randn(50,2)+[10,2]
X = np.concatenate((X1,X2,X3))
```

(b) Use **sklearn.cluster.AgglomerativeClustering** to cluster the points generated in (a). Plot your Dendrogram using different **linkage{"ward", "complete", "average", "single"}.**

Instructions: Set **distance_threshold=0, n_clusters=None** in AgglomerativeClustering**.** The default metric used to compute the linkage is 'euclidean', so you do not need to change this parameter.

P4-2. (U & G-required) Clustering the handwritten digits data

Use the hand-written digits dataset embedded in scikit-learn:

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

(a) Use the following methods to cluster the data:

- K-Means (sklearn.cluster.KMeans)
- DBSCAN (sklearn.cluster.DBSCAN)

Optimize the parameters of these methods.

(b) Evaluate these methods based on the labels of the data and discuss which method gives you the best results in terms of accuracy.

P4-3. (G-required) Clustering structured dataset

(a) Generate a swiss roll dataset:

```
from sklearn.datasets import make_swiss_roll
# Generate data (swiss roll dataset)
n_samples = 1500
noise = 0.05
X, _ = make_swiss_roll(n_samples, noise=noise)
# Make it thinner
```

```
X[:, 1] *= .5
```

(b) Use **sklearn.cluster.AgglomerativeClustering** to cluster the points generated in (a), where you set the parameters as n_clusters=6, connectivity=connectivity, linkage='ward', where

```
from sklearn.neighbors import kneighbors_graph
connectivity = kneighbors_graph(X, n_neighbors=10, include_self=False)
```

Plot the clustered data in a 3D figure and use different colors for different clusters in your figure.

(c) Use **sklearn.cluster.DBSCAN** to cluster the points generated in (a). Plot the clustered data in a 3D figure and use different colors different clusters in your figure. Discuss and compare the results of DBSCAN with the results in (b).