from spectralcluster import SpectralClusterer

clusterer = SpectralClusterer(

min\_clusters=2,

max\_clusters=100,

p\_percentile=0.95,

gaussian\_blur\_sigma=1)

labels = clusterer.predict(X)

where X is a NumPy array of shape (n\_samples, n\_features) and the returned labels is a NumPy array of shape (n\_samples, ).

from sklearn.cluster import SpectralClustering

from sklearn.datasets.samples\_generator import make\_blobs

import matplotlib.pyplot as plt

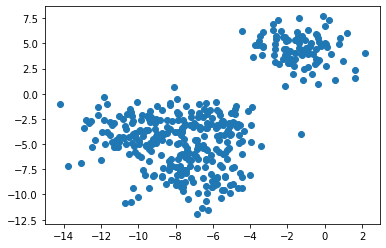
from numpy import random

random.seed(1)

x, \_ = make\_blobs(n\_samples=400, centers=4, cluster\_std=1.5)

plt.scatter(x[:,0], x[:,1])

plt.show()



Clustering with the spectral clustering and visualizing the data

sc=SpectralClustering(n\_clusters=4).fit(x)

SpectralClustering(affinity='rbf', assign\_labels='kmeans', coef0=1, degree=3,

eigen\_solver=None, eigen\_tol=0.0, gamma=1.0,

kernel\_params=None, n\_clusters=4, n\_components=None,

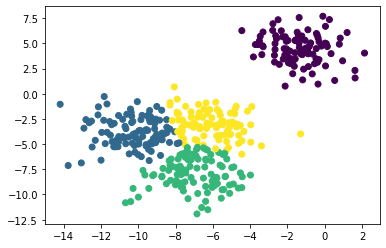
n\_init=10, n\_jobs=None, n\_neighbors=10, random\_state=None)

labels = sc.labels\_

plt.scatter(x[:,0], x[:,1], c=labels)

plt.show()

print(sc)



Let’s try changing the number of clusters.

f = plt.figure()

f.add\_subplot(2, 2, 1)

for i in range(2, 6):

sc = SpectralClustering(n\_clusters=i).fit(x)

f.add\_subplot(2, 2, i-1)

plt.scatter(x[:,0], x[:,1], s=5, c=sc.labels\_, label="n\_cluster-"+str(i))

plt.legend()

plt.show()

