import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

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

##############################################################################

# Generate sample data

np.random.seed(0)

batch\_size = 45

centers = [[1, 1], [-1, -1], [1, -1]]

n\_clusters = len(centers)

X, labels\_true = make\_blobs(n\_samples=3000, centers=centers, cluster\_std=0.7)

##############################################################################

# Compute clustering with Means

k\_means = KMeans(init='k-means++', n\_clusters=3, n\_init=10)

k\_means.fit(X)

k\_means\_labels = k\_means.labels\_

k\_means\_cluster\_centers = k\_means.cluster\_centers\_

k\_means\_labels\_unique = np.unique(k\_means\_labels)

##############################################################################

# Plot result

colors = ['#4EACC5', '#FF9C34', '#4E9A06']

plt.figure()

plt.hold(True)

for k, col in zip(range(n\_clusters), colors):

my\_members = k\_means\_labels == k

cluster\_center = k\_means\_cluster\_centers[k]

plt.plot(X[my\_members, 0], X[my\_members, 1], 'w',

markerfacecolor=col, marker='.')

plt.plot(cluster\_center[0], cluster\_center[1], 'o', markerfacecolor=col,

markeredgecolor='k', markersize=6)

plt.title('KMeans')

plt.grid(True)

plt.show()

**OUTPUT**

