学習メモ

In [ ]:

・クラスタリングとは目的関数を設定し、最大化(最小化)するパラメータを探索する教師なし学習の手法である
・クラスタリングの良さを目的関数として定式化する

演習

In [158]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
import xgboost as xgb
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.model_selection import GridSearchCV
```

In [38]:

```
k_data = pd.read_csv("week8cluster.csv", header=None)
```

In [39]:

```
k_data.head()
```

Out[39]:

	0	1
0	5.798165	4.811611
1	5.694314	4.367983
2	7.154786	4.325657
3	3.835942	3.316601
4	4.018064	3.288468

In [41]:

```
data = np.array(k_data)
```

In [55]:

```
data_new = data[:,1]
```

In [56]:

```
data_new[]
```

Out[56]:

```
(150,)
```

In [57]:

```
data = np.loadtxt("week8cluster.csv",delimiter=",")
```

In [87]:

```
d_size, n_features = data.shape
num_class = 3
max_iter = 200
```

In [85]:

```
centroids = data[np.random.choice(d_size, num_class)]
new_centroids = np.zeros((num_class, n_features))
cluster = np.zeros(d_size)
```

In [86]:

```
print(new_centroids)
print(centroids)
print(cluster)
```

```
[[0. 0.]
 [0. 0.]
 [0. 0.]]
[[-0.83731342  2.10210998]
 [ 3.33266262  4.01067262]
 [ 1.90246657  7.23846262]]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0.]
```

In [142]:

```
def k_means(num_class,data):
    d_size, n_features = data.shape
    centroids = data[np.random.choice(d_size, num_class)]
    new_centroids = np.zeros((num_class, n_features))
    cluster = np.zeros(d_size)
    max_iter = 300

    for epoch in range(max_iter):

        for i in range(d_size):
            distances = np.sum((centroids - data[i]) ** 2, axis=1)
            cluster[i] = np.argsort(distances)[0]

        for j in range(num_class):
            new_centroids[j] = data[cluster==j].mean(axis=0)

        if np.sum(new_centroids == centroids) ==num_class:
            break
        centroids = new_centroids

    return cluster
```

In [145]:

```
cluster = k_means(3,  data = np.loadtxt("week8cluster.csv",delimiter=","))
```

In [150]:

```
cluster
```

Out[150]:

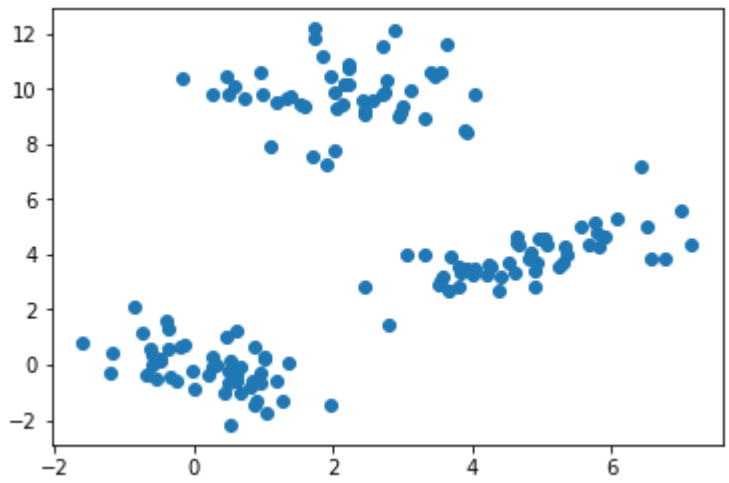
```
array([2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 0.,
       2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2.,
       2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

In [147]:

```
plt.scatter(data[:,0], data[:,1])
```

Out[147]:

<matplotlib.collections.PathCollection at 0x1a18a3cc18>



In [153]:

```
plt.figure()
for (i, label) in enumerate(cluster):
    if label == 0:
        plt.scatter(data[i, 0], data[i, 1], c='red')
    elif label == 1:
        plt.scatter(data[i, 0], data[i, 1], c='blue')
    elif label == 2:
         plt.scatter(data[i, 0], data[i, 1], c='green')
plt.show()
```