Modul Praktikum

Pembelajaran Mesin
(Machine Learning)

Pertemuan 3
K-Means Clustering

K-Means Clustering

Praktikum minggu ini akan membahas mengenai salah satu algoritma dalam unsupervised learning atau clustering, yaitu k-means clustering. K-means clustering merupakan algoritma yang mengelompokkan data menjadi K cluster berdasarkan jarak terdekatnya dengan centroid masing-masing cluster.

Pada praktikum ini akan dicontohkan cara melakukan clustering menggunakan sebuah dataset dan dataset yang di-generate secara otomats menggunakan make_blobs.

I. Melakukan Clustering Menggunakan Sample Dataset

Dataset yang digunakan sebagai contoh adalah xclara.csv. Dataset tersebut terdiri atas 3000 record dan setiap record berisi 2 fitur. Secara natural dataset tersebut membentuk 3 cluster.

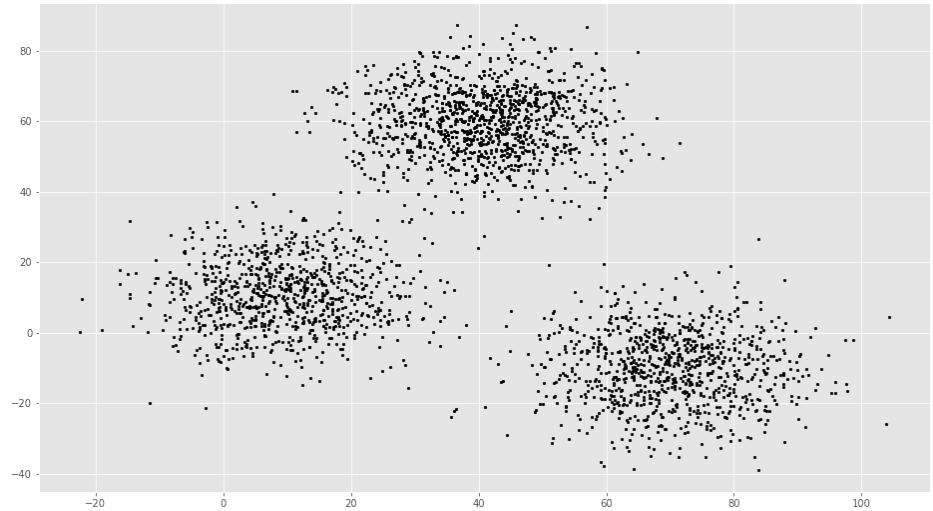
1. Import Dataset

```
%matplotlib inline
from copy import deepcopy
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')

# Import dataset
# Sesuaikan dengan lokasi file xclara.csv di local komputer
>>>data = pd.read_csv('xclara.csv')
>>>print(data.shape)
(3000, 2)
>>> print(dataset.head(20))
```

```
          V1         V2
0   2.072345  -3.241693
1  17.936710  15.784810
2   1.083576   7.319176
3  11.120670  14.406780
4  23.711550   2.557729
5  24.169930  32.024780
6  21.665780   4.892855
7   4.693684  12.342170
8  19.211910  -1.121366
9   4.230391  -4.441536
10  9.127130  23.605720
11  0.407503  15.297050
12  7.314846   3.309312
13 -3.438403 -12.025270
14 17.639350  -3.212345
15  4.415292  22.815550
16 11.941220   8.122487
17  0.725853   1.806819
18  8.185273  28.132600
19 -5.773587   1.024800
>>>
```

```
# Plot dataset
f1 = data['V1'].values
f2 = data['V2'].values
X = np.array(list(zip(f1, f2)))
plt.scatter(f1, f2, c='black', s=7)
```



2. Melakukan clustering

   Untuk melakukan clustering dengan K-Means clustering kita akan menggunakan library sklearn.cluster.

```
from sklearn.cluster import KMeans
# Menentukan jumlah cluster
kmeans = KMeans(n_clusters=3)
# Fitting input data
```

```
kmeans = kmeans.fit(X)
# Mendapatkan cluster labels
labels = kmeans.predict(X)
# Mendapatkan nilai centroid
C = kmeans.cluster_centers_
# Mencetak nilai centroid
print(C)
[[  9.4780459   10.686052  ]
 [ 69.92418447 -10.11964119]
 [ 40.68362784  59.71589274]]
```

3. Plot hasil clustering

```
plt.scatter(X[:, 0], X[:, 1], s=7, c=labels)
plt.scatter(C[:, 0], C[:, 1], marker='*', s=200, c='#050505')
```

II. Melakukan Clustering Menggunakan Generate Dataset

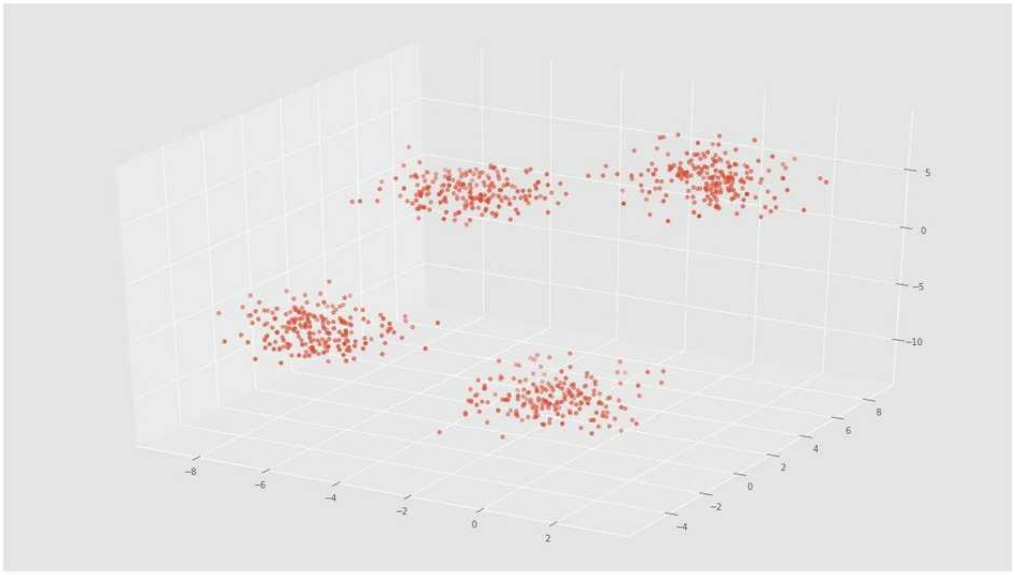
Pada contoh kali ini, datset yang akan di-cluster di-generate secara otomatis menggunakan make_blobs.

1. Men-generate dataset

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

plt.rcParams['figure.figsize'] = (16, 9)
```

```
# Men-generate dataset yang terkelompok dalam 4 cluster
X, y = make_blobs(n_samples=800, n_features=3, centers=4)
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1], X[:, 2])
```

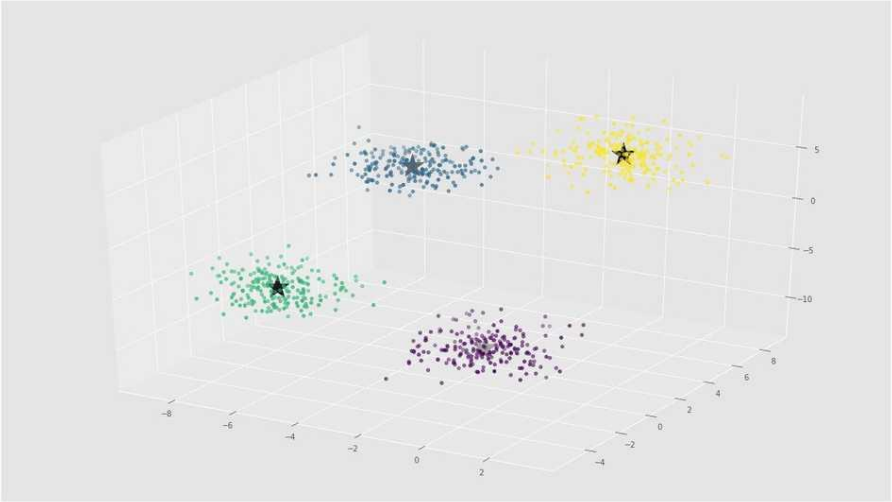


2. Melakukan clustering

```
# Initializing KMeans
kmeans = KMeans(n_clusters=4)
# Fitting with inputs
kmeans = kmeans.fit(X)
# Predicting the clusters
labels = kmeans.predict(X)
# Getting the cluster centers
C = kmeans.cluster_centers_
```

3. Plot hasil clustering

```
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=y)
ax.scatter(C[:, 0], C[:, 1], C[:, 2], marker='*', c='#050505', s=1000)
```



III. Melakukan Evaluasi Hasil Clustering

Pada kedua contoh di atas, kita telah mengetahui jumlah k (cluster) yang tepat untuk melakukan clustering. Pada kasus di dunia nyata, kita mungkin tidak mengetahui dengan tepat berapa jumlah cluster yang tepat untuk dataset yang akan kita cluster. Untuk itu, kita perlu melakukan evaluasi hasil clustering menggunakan beberapa kemungkinan nilai k dan menggunakan metric yang bersesuaian untuk clustering. Sebagai contoh kita akan mengevaluasi hasil clustering pada contoh yang pertama.

```
for k in range (1, 10):
    # Menentukan jumlah cluster
    kmeans = KMeans(n_clusters=k, random_state=1)
    # Fitting input data
    kmeans = kmeans.fit(X)
    # Mendapatkan cluster labels
    labels = kmeans.predict(X)
    # Menghitung jumlahan jarak antara setiap sampel dengan cluster
centroid-nya (SSE)
    interia = kmeans.inertia_
    print "k:",k, " cost:", interia
print ""
```

Evaluasi hasil cluster menggunakan silhouette coefficient

```
from sklearn.metrics.cluster import silhouette_score
silhouette_score(X, labels)
```

Tugas!

1. Lakukanlah clustering menggunakan dataset iris seperti yang digunakan pada praktikum sebelumnya!
2. Lakukan evaluasi hasil clustering menggunakan inertia dan silhouette coefficient!