MODUL DATA MINING

CLUSTERING WITH K-MEANS

Pada modul ini dijelaskan mengenai proses cluster analysis dengan menggunakan algoritma k-means dan menerapkannya dalam bahasa pemrograman python.
Diharapkan setelah mempelajari modul ini, mahasiswa mampu memahami implementasi k-means dalam suatu kasus.

EPS
6

CLUSTER ANALYSIS

Inti dari klastering adalah mengumpulkan entiti yang memiliki kesamaan sifat dalam satu golongan yang sama. Modul ini akan menjelaskan bagaimana melakukan segmentasi pelanggan dengan mempergunakan klastering. Dataset berisi informasi tentang pendapatan tahunan pelanggan (dalam $000) dan total pengeluaran (dalam $000) pada suatu situs ecommerce dalam 1 tahun.

1. Load packages yang diperlukan

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```

2. Atur styling untuk plot

```
In [2]: import seaborn as sns; sns.set()
        %matplotlib inline
        plt.rcParams['figure.figsize'] = (16, 9)
        plt.style.use('ggplot')
```

3. Load dataset

```
In [3]: dataset=pd.read_csv('/home/eka/Documents/CLV.csv')
```

4. Explore dataset

```
In [4]: dataset.head()
```

Out[4]:

	INCOME	SPEND
0	233	150
1	250	187
2	204	172
3	236	178
4	354	163

```
In [5]: len(dataset)
Out[5]: 303
```

5. Menentukan jumlah k paling optimal

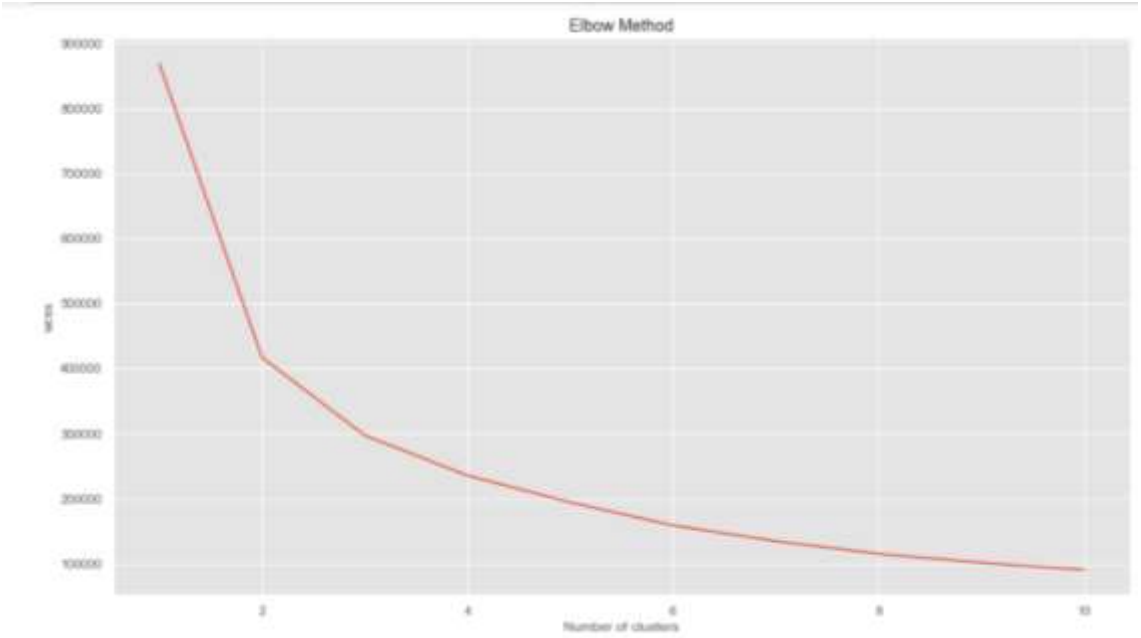
Inputan utama untuk algoritma k-means adalah jumlah k. k menunjukkan jumlah klaster/pengelompokkan yang akan dihasilkan. Ada berbagai macam metode untuk menentukan berapa jumlah k yang paling baik, modul kali ini memakai methode elbow.

Apa saja metode untuk menentukan jumlah k yang paling tepat ?

```
In [6]: X=dataset.iloc[:,[0,1]].values
```

```
In [7]: from sklearn.cluster import KMeans
        wcss = []
        for i in range(1,11):
            km=KMeans(n_clusters=i,init='k-means++', max_iter=300, n_init=10, random_state=0)
            km.fit(X)
            wcss.append(km.inertia_)
        plt.plot(range(1,11),wcss)
        plt.title('Elbow Method')
        plt.xlabel('Number of clusters')
        plt.ylabel('wcss')
        plt.show()
```

Outputnya sebagai berikut



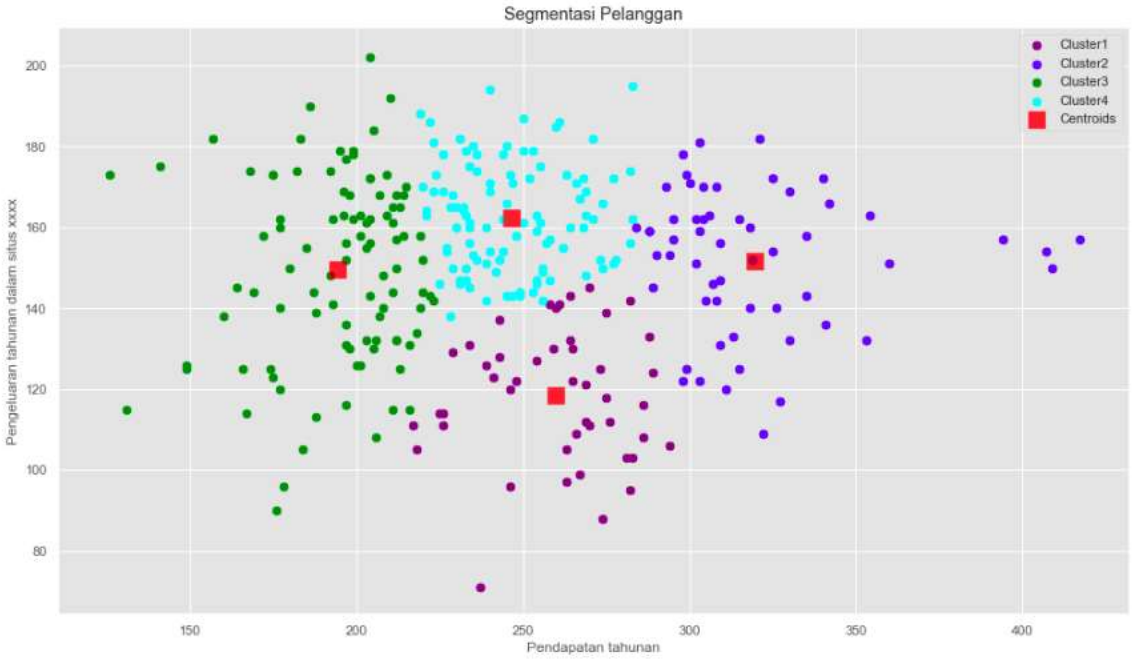
Jumlah klaster yang optimal, ditunjukkan pada bagian ‘elbow’ dari grafik diatas. Yang dicari adalah titik awal, dimana ketika jumlah klaster ditambahkan, nilai wcss tidak mengalami penurunan yang drastis. Dari grafik diatas, elbow ada pada rentang 4-6. oleh karena itu, perlu dijalankan k-means untuk masing-masing k dalam rentang tersebut.

6. Jalankan k-means dan visualisasikan data

Untuk k=4

```
In [*]: ##k=4
km4=KMeans(n_clusters=4,init='k-means++', max_iter=300, n_init=10, random_state=0)
y_means = km4.fit_predict(X)
#Visualisasi untuk k=4
plt.scatter(X[y_means==0,0],X[y_means==0,1],s=50, c='purple',label='Cluster1')
plt.scatter(X[y_means==1,0],X[y_means==1,1],s=50, c='blue',label='Cluster2')
plt.scatter(X[y_means==2,0],X[y_means==2,1],s=50, c='green',label='Cluster3')
plt.scatter(X[y_means==3,0],X[y_means==3,1],s=50, c='cyan',label='Cluster4')
plt.scatter(km4.cluster_centers_[:,0], km4.cluster_centers_[:,1],s=200,marker='s',
            c='red', alpha=0.7, label='Centroids')
plt.title('Segmentasi Pelanggan')
plt.xlabel('Pendapatan tahunan')
plt.ylabel('Pengeluaran tahunan dalam situs xxxx')
plt.legend()
plt.show()
```

Hasil :



Bagaimana untuk k=5 dan k=6 ?

7. Uji kualitas klaster yang dihasilkan

Untuk menguji kualitas klaster juga ada banyak metode, salah satunya dengan memakai silhouette coeffisien

Apa saja metode untuk menguji kualitas klaster ?

```
In [13]: from sklearn.metrics import silhouette_score
range_n_clusters = [4, 5, 6]

for n_clusters in range_n_clusters:
    clusterer = KMeans(n_clusters=n_clusters, init='k-means++',
                       max_iter=300, n_init=10, random_state=0)
    y_means = clusterer.fit_predict(X)
    silhouette_avg = silhouette_score(X, y_means)
    print("jumlah claster =", n_clusters,
          "nilai average_silhoute =", silhouette_avg)
```

```
jumlah claster = 4 nilai average_silhoute = 0.3618396084064086
jumlah claster = 5 nilai average_silhoute = 0.352199348445336
jumlah claster = 6 nilai average_silhoute = 0.35975391689328834
```

Nilai silhouette coeff ada pada rentang -1 hingga +1. Semakin mendekati nilai 1, maka semakin baik pengelompokan data dalam satu cluster. Sebaliknya jika silhouette coefficient mendekati nilai -1, maka semakin buruk pengelompokan data didalam satu cluster. Dari nilai diatas, cluster yang paling baik adalah 4.

8. Analisa klaster yang dihasilkan

Klaster 1 : Pelanggan dengan pendapatan tahunan menengah dan pengeluaran tahunan rendah

Klaster 2 : Pelanggan dengan pendapatan tahunan tinggi dan pengeluaran tahunan menengah hingga tinggi

Klaster 3 : Pelanggan dengan pendapatan tahunan rendah dan pengeluaran tahunan menengah hingga tinggi

Klaster 4 : Pelanggan dengan pendapatan tahunan menengah tetapi pengeluaran tahunan tinggi

9. Susun skenario yang mungkin, untuk:

Bagaimana skenario yang mungkin untuk mempertahankan kesetiaan pelanggan?