Soal 1

DBScan

1. Inisialisasi seluruh data dengan status unlabeled data
2. Lakukan iterasi untuk seluruh data. Untuk sebuah data X, jika X sudah berlabel, maka iterasi dilanjut ke data berikutnya, jika belum berlabel, cari seluruh tetangga dari X. Suatu tetangga merupakan data yang jaraknya kurang dari sejumlah nilai (*epsilon*).
3. Jika tetangga yang dimiliki oleh X kurang dari batas tertentu (*min_samples*), maka X diberi label *noise* (-1), kemudian lanjutkan iterasi ke data berikutnya (lanjutkan ke poin 2). Jika tetangga yang dimiliki oleh X lebih atau sama dengan batas tertentu, maka X termasuk ke dalam suatu cluster baru.
4. Lakukan iterasi untuk seluruh tetangga dari X. Untuk Y tetangga X, ubah label Y menjadi cluster yang sama dengan X.
5. Cari seluruh tetangga dari Y. Jika jumlah tetangga dari Y lebih atau sama dengan batas tertentu (*min_samples*), tambahkan seluruh tetangga dari Y ke dalam list tetangga X. Jika jumlah tetangga Y kurang dari batas tertentu, lanjutkan iterasi ke tetangga X berikutnya.
6. Ulangi poin 4 dan 5 sampai seluruh tetangga dari X sudah diproses.
7. Kembali ke poin 2 sampai seluruh data sudah dilabeli.

KMeans

Algoritma KMeans yang diimplementasikan memiliki 2 tahap utama, yaitu inisialisasi centroid dan clustering. Pelabelan data pada algoritma KMeans akan selalu memakai cara yang sama, yaitu mengelompokkan data berdasarkan centroid terdekatnya (jarak Euclidean).

Inisialisasi centroid:

1. Pilih satu data random dari dataset untuk dijadikan centroid pertama
2. Jadikan data yang paling jauh dari centroid pertama sebagai centroid kedua
3. Lakukan pelabelan data berdasarkan kedua centroid tersebut.
4. Jika jumlah cluster <= 2, inisialisasi sudah selesai. Jika jumlah cluster > 2, lanjutkan ke poin 5.
5. Tambahkan centroid cluster ke i ke dalam list centroid. Centroid dari cluster ke i merupakan titik terjauh suatu data yang termasuk ke dalam cluster ke $i - 2$ terhadap centroidnya.
6. Lakukan pelabelan data
7. Ulangi poin 5 dan 6 hingga jumlah cluster yang diharapkan tercapai.

Clustering:

1. Hitung titik tengah dari setiap centroid yang ada (menggunakan rata-rata), kemudian geser centroid ke titik tersebut.
2. Lakukan pelabelan data
3. Ulangi poin 1 dan 2 hingga tidak ada perubahan atau jumlah iterasi sudah mencapai nilai maksimal.

KMedoids

Algoritma KMedoids yang diimplementasikan memiliki 2 tahap utama, yaitu inisialisasi medoid dan clustering. Pelabelan data pada algoritma KMedoids akan selalu memakai cara yang sama, yaitu mengelompokkan data berdasarkan medoid terdekatnya (jarak Manhattan). Setiap himpunan dari suatu medoid pasti memiliki suatu nilai *cost*. Nilai *cost* tersebut didefinisikan sebagai jumlah dari seluruh Manhattan Distance dari setiap data terhadap medoidnya masing-masing.

Inisialisasi medoid:

1. Cari sepasang medoid yang memiliki *cost* paling rendah dengan cara iterasi seluruh kemungkinan.
2. Jika jumlah cluster <= 2, inisialisasi sudah selesai. Jika jumlah cluster > 2, lanjutkan ke poin 3.
3. Untuk pencarian medoid ke i, cari sebuah data yang jika dimasukkan ke dalam himpunan medoid akan memiliki *cost* paling kecil dengan cara iterasi seluruh kemungkinan.
4. Ulangi poin 3 hingga mencapai jumlah medoid yang diinginkan

Clustering:

1. Untuk setiap medoid M dalam himpunan medoid, ganti M dengan data lain kemudian hitung *cost*-nya. Iterasi seluruh data untuk menggantikan medoid tersebut. Proses penggantian dilakukan secara bergiliran dari setiap medoidnya (tidak ada penggantian lebih dari 1 medoid).
2. Setelah seluruh proses penggantian selesai, catat himpunan medoid yang memiliki cost terkecil. Jika himpunan medoid tidak berubah, maka proses clustering dihentikan. Jika himpunan medoid berubah, lanjutkan poin 1 hingga himpunan medoid tidak berubah atau jumlah iterasi sudah mencapai batas maksimal.

Soal 2

DBScan

Kelebihan:

- Karena mengelompokkan data berdasarkan densitas, bentuk cluster bisa sangat flexible. Tidak seperti KMeans dan KMedoids yang bentuknya terbatas (hanya sebatas sejenis lingkaran saja)
- Robust terhadap noise dan outlier. Dapat mengklasifikasikan suatu data sebagai noise atau outlier
- Jumlah cluster yang optimal langsung bisa didapatkan, tidak perlu lagi mencari jumlah cluster satu per satu

Kekurangan

- Sangat bergantung terhadap hyperparameter yang digunakan (epsilon, min_samples, distance metric)
- Penggunaan memori yang cukup besar
- Tidak bisa menyesuaikan jumlah cluster
- Sulit untuk data yang memiliki densitas yang sangat beragam

KMeans

Kelebihan:

- Salah satu algoritma clustering yang sangat sederhana dan mudah untuk diimplementasikan
- Mengerucut ke hasil akhir dengan cukup cepat sehingga dapat digunakan untuk dataset yang besar

Kekurangan:

- Sensitif terhadap outlier
- Sensitif terhadap metode inisialisasi, perbedaan metode inisialisasi clustering dapat mengakibatkan perbedaan terhadap hasil clustering.
- Tidak deterministic, untuk runtime yang berbeda hasilnya bisa berbeda.

KMedoids

Kelebihan:

- Lebih robust terhadap outlier dibandingkan dengan KMeans
- Dapat bekerja dengan bermacam-macam distance metric (tidak harus Euclidean)

Kekurangan:

- Berat secara komputasional, sehingga kurang optimal pada dataset yang cukup besar
- Dipengaruhi metode inisialisasi

Soal 3

- Dapat digunakan untuk mengelompokkan customer berdasarkan perilakunya
- Image segmentation, yaitu mengelompokkan pixel-pixel yang memiliki arti tertentu
- Deteksi outlier/noise
- Analisis social network, mengelompokkan user yang memiliki perilaku mirip
- Sistem rekomendasi pada platform konten, e-commerce, dan sebagainya