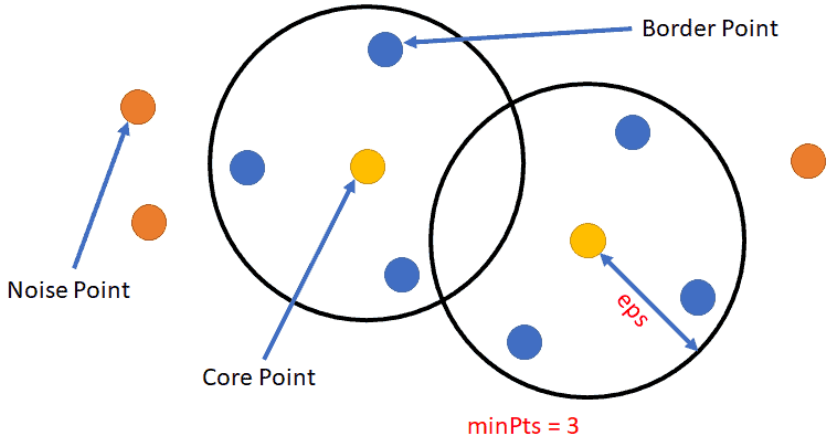
DBSCAN

Salman Hanif - 13523056

Cara Kerja Algoritma

Algoritma DBSCAN bekerja dengan mengelompokkan titik-titik data yang jaraknya berdekatan dengan berdasarkan jarak maksimal/jari-jari dari titik data pusat, dan satu kumpulan daata bisa dianggap satu klaster berdasarkan minimum jumlah data di sekitarnya, sedangkan data-data yang tidak masuk daerah/klaster mana pun dianggap noise atau outlier. Berbeda dengan K-Means yang jumlah klaster nya ditentukan manual dan terbatas, DBSCAN bekerja mandiri menentukan jumlah klaster nya. Jarak daatanya sendiri bisa dicari dengan rumus euclidean, manhattan, dan minkowski.



Alur kerja fungsi fit dan transform :

Fit :

1. Setiap titik data dalam X diberi label awal -2, yang berarti belum dikunjungi atau belum ditetapkan. Sebuah clusterId diinisialisasi dengan 0 untuk menandai cluster pertama
2. Algoritma mengiterasi setiap titik data. Jika sebuah titik sudah memiliki label selain -2 (sudah diklasifikasikan), proses akan dilewati. Untuk setiap titik yang belum diklasifikasi, algoritma akan mencari semua tetangganya dalam radius eps menggunakan metode _getNeighbors.
3. Jika jumlah tetangga dari titik saat ini kurang dari minSamples, titik tersebut dianggap sebagai noise dan diberi label -1. Jika jumlah tetangga lebih besar atau sama dengan minSamples, titik tersebut adalah titik inti.

4. Jika sebuah titik diidentifikasi sebagai titik inti, algoritma akan memanggil metode _expandCluster untuk membentuk cluster baru. Algoritma secara berulang akan memeriksa setiap tetangga. Jika tetangga tersebut juga merupakan titik inti, semua tetangganya juga akan ditambahkan ke antrian untuk diperiksa. Proses ini berlanjut hingga tidak ada lagi titik yang dapat dijangkau dalam cluster yang sama. Proses berlanjut hingga semua titik diklasifikasikan

Predict:

1. Sebuah array predictedLabels diinisialisasi, di mana setiap titik data baru diberi label awal -1
2. Algoritma mengiterasi setiap titik dalam X, Untuk setiap titik baru, algoritma mencari titik terdekat (baik titik inti maupun titik batas) dari data yang telah dilatih (self.X_). Metode _getDistance digunakan untuk menghitung jarak antara titik baru dan setiap titik yang ada di data latih.
3. Jika titik baru berada dalam jarak eps dari sebuah titik inti atau titik batas yang sudah ada, titik baru akan ditetapkan ke cluster yang sama dengan titik terdekat tersebut. Jika ditemukan lebih dari satu titik terdekat dalam eps, titik baru akan ditetapkan ke cluster dari titik yang paling dekat.
4. Hasil akhirnya adalah sebuah array predictedLabels yang berisi label cluster untuk setiap titik data baru.

Hasil Evaluasi model dari hasil scratch dan dari library. Jelaskan perbedaan

Dalam evaluasi, model saya dapat bekerja dengan baik melakukan clustering pada data, sama baiknya dengan DBSCAN dari sklearn pada eps dan minSamples yang sama. Jarak default yang di-set pun sama-sama menggunakan Euclidean.

Pengujian hanya dilakukan pada satu dataset yang sama, presisi kedua model mungkin bisa berbeda jika dalam kasus data yang lebih kompleks.

Improvement yang bisa dilakukan

Pada fungsional sudah berjalan dengan cukup baik. Improvement yang bisa dilakukan mungkin pada optimalisasi performa saja, seperti looping perhitungan jarak yang cukup memakan waktu mungkin bisa diefisienkan dengan suatu library untuk menghitung seluruh jarak.

Penambahan visualisasi cluster untuk dataset 2 dan 3 fitur bisa ditambahkan meski kurang terpakai karena jarang dataset sekecil itu.