13522103 - Steven Tjhia

Density-Based Spatial Clustering of Applications with Noise

1. Jelaskan cara kerja dari algoritma tersebut! (boleh dalam bentuk *pseudocode* ataupun narasi)
   ⇒ Pertama-tama, model akan menyimpan informasi epsilon, minimum *sample*, dan metrik jarak yang digunakan. Epsilon disini akan menentukan "radius" ketika model melakukan "scan', sedangkan minimum *sample* disini akan menentukan apakah *sample* tersebut akan menjadi *core point* atau tidak. Jika terdapat setidaknya sebanyak {minimum *sample*} di dalam radius epsilon dari suatu *sample*, maka *sample* tersebut akan menjadi *core point*.
   Selanjutnya model akan memilih *core point* secara acak dan memulai "scan". Model akan memulai cluster baru dan melabeli sampel tersebut dengan cluster baru itu. Selanjutnya semua sampel yang berada dalam jangkauan radius epsilon dari sampel saat ini akan dipilih menjadi sampel yang sedang dicek saat ini (*currentCluster*) dan melabeli sampel tersebut dengan cluster tersebut, jadi perlahan-lahan *cluster* seperti menyebar sampai sudah tidak ada lagi sampel yang bisa dilabeli dalam jangkauan radius epsilon dari suatu sampel. Setelah hal tersebut selesai, selanjutnya akan memulai *cluster* baru pada sampel yang belum dilabeli dengan suatu *cluster* apapun. Selanjutnya lakukan hal yang sama seperti sebelumnya yaitu "menyebarkan" *cluster* tersebut sampai tidak ada yang bisa disebarkan lagi.

2. Bandingkan hasil evaluasi model from scratch dan *library*, bagaimana hasil perbandingannya? Jika ada perbedaan, jelaskan alasannya!

Waktu Eksekusi	
Scratch	***Library***
83 detik	0.370 detik
Jumlah Cluster yang dihasilkan (outlier tidak dihitung)	
Scratch	***Library***
4 *cluster*	11 *cluster*

Silhouette Score	
Scratch	***Library***
0.129	0.074

Berdasarkan data hasil percobaan tersebut, dapat terlihat bahwa waktu eksekusi yang diperlukan oleh model yang dibuat sendiri (sangat) jauh lebih lama dibandingkan dengan waktu eksekusi yang diperlukan model dari *library*. Hal tersebut dikarenakan model yang dibuat sendiri tidak mengimplementasikan optimasi kecepatan algoritma, sehingga untuk data yang banyak otomatis akan memerlukan waktu eksekusi yang lebih lama juga. Untuk hasil jumlah clusternya pun berbeda cukup jauh meskipun menggunakan hyperparameter yang sama. Hal tersebut dapat dikarenakan adanya perbedaan implementasi algoritma, terutama pada penentuan *core point*, *border point*, sampai proses "penyebaran" *cluster*. Untuk *silhouette score* yang didapatkan dari model yang dibuat sendiri bernilai lebih besar daripada hasil *library* (semakin besar nilai *silhouette score* maka semakin bagus clusteringnya) hal tersebut dapat dikarenakan jumlah *cluster* yang dihasilkan model *library* sangat banyak yaitu 11 yang mungkin dapat dikarenakan aspek yang telah dijelaskan sebelumnya.