Nama : Silverius Sony Lembang
NIM : H071191002

KLASTERISASI

Klasterisasi adalah model *machine learning* yang termasuk dalam kategori *unsupervised learning.* Hal ini karena pada dasarnya *data* yang akan diolah menggunakan model klasterisasi tidak memiliki label. Klasterisa adalah model machine learning yang bertujuan untuk membentuk beberapa grup data sehingga setiap data yang berada pada grup yang sama memiliki kemiripan dan sangat berbeda dengan data pada grup yang lain. Klasterisasi secara umum digunakan dalam menyelesaikan masalah yang berkaitan dengan segmentasi pasar, segmentasi gambar, dan pemetaan zona wilayah.

Klasterisasi yang adalah sebuah model machine learning memiliki beberapa algoritma yang digunakan dalam menyelesaikan masalah. Beberapa contoh dari algoritma klasterisasi adalah K-means, K-medoids, Hierarchical clustering, dan DBScan.

a. K-Means

K-Means adalah algoritma iteratif yang melakukan proses klasterisasi sebanyak K klaster dimana setiap data point hanya akan menjadi bagian dari salah satu klaster yang terbentuk. Algoritma ini akan mengelompokkan data yang mirip kedalam satu kelompok dengan tetap memisahkannya dari kelompok yang berbeda sebanyak k kelompok yang telah ditentukan diawal. Penentuan klaster dari setiap data point dapat ditentukan dengan menghitung jarak data point ke masing masing centroid dan mengambil jarak terkecil sebagai klusternya. Centroid adalah pusat dari sebuah klaster.

Tahapan-tahapan dalam algoritma K-Means adalah sebagai berikut.

1. Tentukan banyak k klaster
2. Tentukan letak dari masing-masing centroid secara acak
3. Pilih salah satu data point dalam dataset dan tentukan klasternya berdasarkan centroid terdekat. Lakukan proses ini untuk semua data point dalam dataset hingga semua data sudah dikelompokkan klaster tertentu
4. Hitung rata-rata dari semua data point dalam klaster untuk menentukan posisi centroid.
5. Ulangi langkah 3-4 hingga posisi semua centroid tidak berubah

Kelemahan dari K-Means adalah sangat sensitif terhadap keberadaan outlier. Hal ini dapat memengaruhi posisi centroid sehingga cluster yang terbentuk bisa menjadi kurang optimal.

b. K-Medoids

K-Medoids adalah algoritma klasterisasi yang merupakan pengembangan dari algoritma K-Means. K-Medoids dikembangakan setelah melihat kekurangan dari algoritma K-Means yang sangat sensitif terhadap keberadaan outlier. Perbedaan urama yang antara K-Means dan K-Medoids adalah pada K-Medoids, pusat klaster yang ditentukan adalah sebuah data point yang terdapat pada sekumpulan data point yang kemudian disebut sebagai medoids. Medoids ini adalah sebuah data point yang memiliki rata-rata ketidaksamaan yang terendah di dalam klaster. Hal ini berbeda dengan K-Means yang menggunakan rata-rata dari anggota klaster untuk menentukan posisi dari centroidnya.

Secara umum tahapan-tahapan yang dilalui dalam K-Medoids mirip dengan yang ada dalam metode K-Means. Tahapan-tahapan tersebut sebagai berikut.

1. Tentukan jumlah k kluster

2. Inisisalisasi k medoids awal secara random dari data point dari dataset
3. Untuk setiap data point yang bukan medoids, hitung jarak dari data point tersebut ke masing-masing medoid. Jarak terendah merupakan klaster sementara dari data point tersebut.
4. Untuk setiap data dalam klaster, lakukan update terhadap medoid dengan menganggap setiap data point sebagai calon medoid. Medoid ditentukan dengan mencari jarak rata-rata dari calon medoid ke seluruh data point dalam klaster. Jarak rata-rata terendah menandakan data point tersebut sebagai medoid berikutnya menggantikan medoid sebelumnya. Lakukan untuk setiap medoid pada klaster lainnya
5. Ulangi langkah 3-4 hingga posisi dari medoid tidak mengalami perubahan atau mencapai batas iterasi tertentu.

c. Hierarchical Clustering

Hierarchical Clustering adalah metode klasterisasi yang digunakan untuk membentuk sebuah pohon klaster yang disebut sebagai dendrogram. Dalam hierarchical clustering, terdapat dua jenis metode yang dapat diterapkan yakni.

1. Agglomerative Clustering

   Agglomerative clustering bekerja dengan menganggap bahwa setiap data point merupakan sebuah klaster tersendiri. Sehingga jika terdapat n data point, maka terdapat n klaster saat inisisalisasi. Proses selanjutnya adalah menggabungkan data point yang memiliki kemiripan menjadi kedalam satu klaster. Hal ini dilakukan hingga terbentuk 1 klaster yang memuat seluruh data point.
2. Divisive Clustering

   Divisive clustering melakukan hal yang terbalik dari Agglomerative Clustering. Jika pada agglomerative clustering menggunakan pendekatan bottom-top, maka pada divisive

clustering menggunakan pendekatan top-down yakni saat inisialisasi hanya terdapat 1 klaster. Kemudian klaster tersebut akan dipisahkan hingga terbentuk sebanyak n klaster dimana n adalah jumlah data point dalam dataset. Proses pemisahan klaster ini dapat menggunakan algoritma K-Means.

Dalam menetukan kedekatan antar klaster seperti pada Agglomerative Clustering, maka terdapat tolak ukur yang dapat digunakan sebagai berikut.

1. Single linkage, yakni kedekatan antar 2 klaster dihitung dari jarak minimum antara dua data point yang berbeda klaster.
2. Complete linkage, yakni kedekatan antar 2 klaster dihitung berdasarkan jarak terjauh antara 2 data point yang berbeda klaster.
3. Centroid linkage, yakni kedekatan 2 klaster diukur dari jarak antara masing masing centroid dari 2 klaster tersebut.
4. Average linkage, yakni kedekatan 2 klaster diukur dari rata-rata jarak antara pasangan-pasangan data point dari kedua klaster

d. DB Scan

DBSCAN merupakan singkatan dari Density-Based Spatial Clustering of Applications with Noise. Metode ini bekerja dengan membuat klaster berdasarkan kepadatan dari data point. Dalam metode ini, kita tidak perlu menentukan jumlah klaster secara eksplisit. Namun dalam ini kita memerlukan parameter ϵdan minPts. ϵmerupakan parameter yang digunakan untuk menentukan radius disekitar data point tertentu. minPts adalah parameter yang digunakan untuk menentukan seberapa banyak data point dalam circle/hypersphere agar menjadi core point. Core point adalah data point yang memiliki jumlah data point tetangga yang lebih dari atau sama dengan minPts. Istilah lain yang digunakan adalah border point yakni data point yang memiliki jumlah data point tetangga yang kurang dari minPts namun terhubung ke sedikitnya 1 core point.

Yang terakhir adalah noise yakni data point yang tidak termasuk dalam core point dan border point.

Langkah langkah yang dilalui dalam algoritma ini sebagai berikut.

1. Tentukan core point dari dataset
2. Pilih satu core point, untuk setiap tetangganya kelompokkan sebagai satu klaster
3. Cari semua density-connected point secara rekursif dan kelompokkan sebagai kelompok yang sama dengan core point. A dan B disebut density connected jika terdapat data point C yang memenuhi minPTS dan A serta B termasuk kedalam minPTS tersebut
4. Iterasi seluruh data point yang belum dikunjungi. Untuk core point yang belum mempunyai klaster maka core point tersebut akan membentuk klaster baru dan prosesnya sama seperti langkah ke 3
5. Data point yang tidak memiliki klaster didefenisikan sebagai noise