

Unsupervised Learning

24 April 2024



Materi Hari Ini



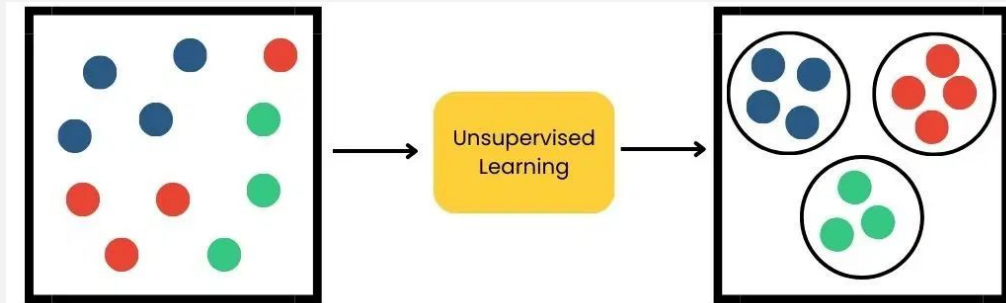
Kita akan mempelajari antara lain:

1. Unsupervised Learning
2. Clustering
3. Algoritma K-Means
4. Algoritma Agglomerative Hierarchical Clustering



Unsupervised Learning

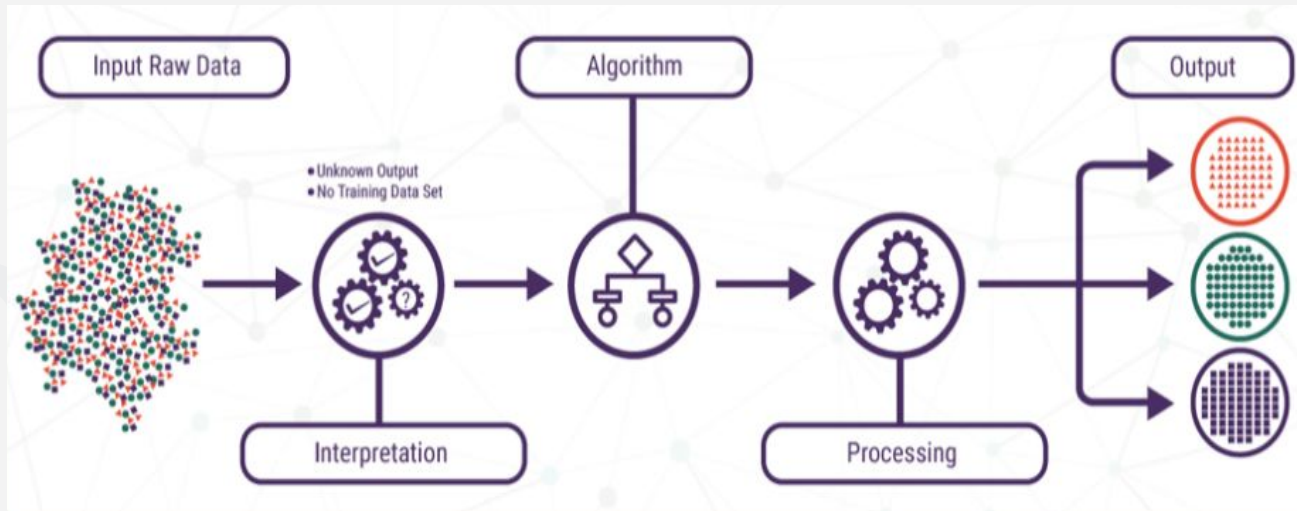
Apa dan Bagaimana



Unsupervised Learning

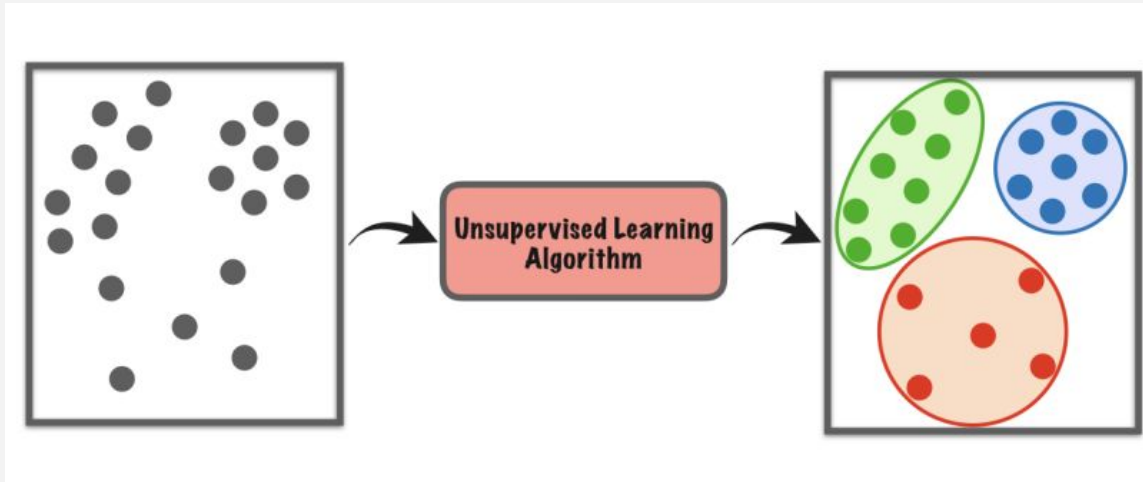


- ❖ **Unsupervised Learning**: merupakan salah satu metode di dalam algoritma machine learning yang akan melakukan proses pembelajaran data dengan tujuan untuk mengelompokkan object-object yang terdapat pada data menjadi beberapa kelompok berdasarkan dengan tingkat kemiripan yang tinggi pada featuranya (informasi pada object).



Unsupervised Learning

- ❖ Sangat berbeda bila dibandingkan dengan Supervised Learning, Unsupervised Learning merupakan jenis learning yang hanya mempunyai variabel input.
- ❖ Tujuan dari Algoritma Machine Learning pada Unsupervised Learning ini adalah untuk memodelkan struktur data dan menyimpulkan fungsi yang mendeskripsikan data tersebut.



Unsupervised Learning



Beberapa algoritma yang termasuk kedalam Unsupervised learning adalah sebagai berikut:

1. Clustering
2. Anomaly Detection
3. Autoencoder
4. Deep Belief Network
5. Principal Component Analysis (PCA)
6. Association

Unsupervised Learning



a. Euclidean Distance

$$d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

b. Manhattan Distance

$$d(x, y) = \sum_{i=1}^m |x_i - y_i|$$

c. Minkowski Distance

$$d(x, y) = \left(\sum_{i=1}^m |x_i - y_i|^r \right)^{1/r}$$

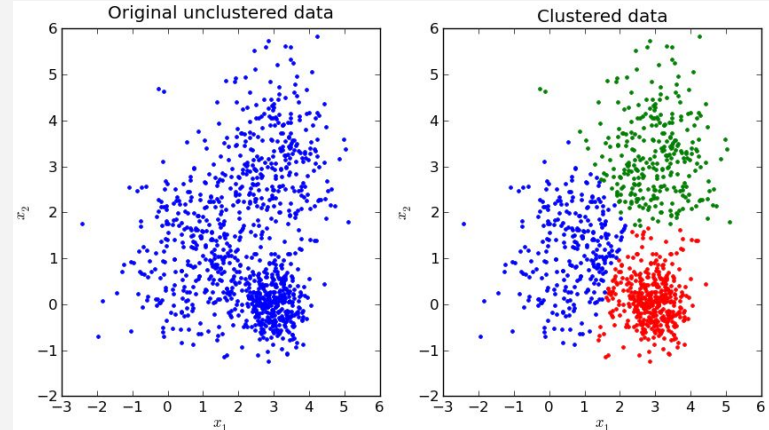
d. Chebyshev Distance

$$d(x, y) = \max_{i=1}^n |x_i - y_i|$$

- ❖ List disamping adalah beberapa formula perhitungan jarak yang umum digunakan
- ❖ Salah satu cara untuk menghitung kedekatan antar object di dalam data berdasarkan fiturnya adalah dengan menggunakan konsep jarak.
- ❖ Sehingga object berada di dalam satu cluster, berarti jarak dari sekumpulan feature object didalamnya berdekatan dibandingkan dengan object di cluster lain.

Clustering (Pengklasteran)

Apa dan Bagaimana



Clustering



- ❑ Clustering digunakan untuk pengelompokan object-object data kedalam kelompok atau cluster yang tiap cluster memiliki object-object dengan tingkat kesamaan feature antar object yang tinggi.
- ❑ Algoritma clustering dapat digunakan untuk analisis cluster yang mempunyai tujuan utama untuk mengelompokkan objek-objek berdasarkan karakteristik yang dimilikinya.
- ❑ Analisis cluster mengklasifikasikan objek sehingga setiap objek yang paling dekat kesamaannya dengan objek lain berada dalam cluster yang sama.

Clustering



Ada beberapa pendekatan yang digunakan algoritma clustering untuk mengelompokkan object-object ke dalam tiap cluster:

❑ Pendekatan Partisi

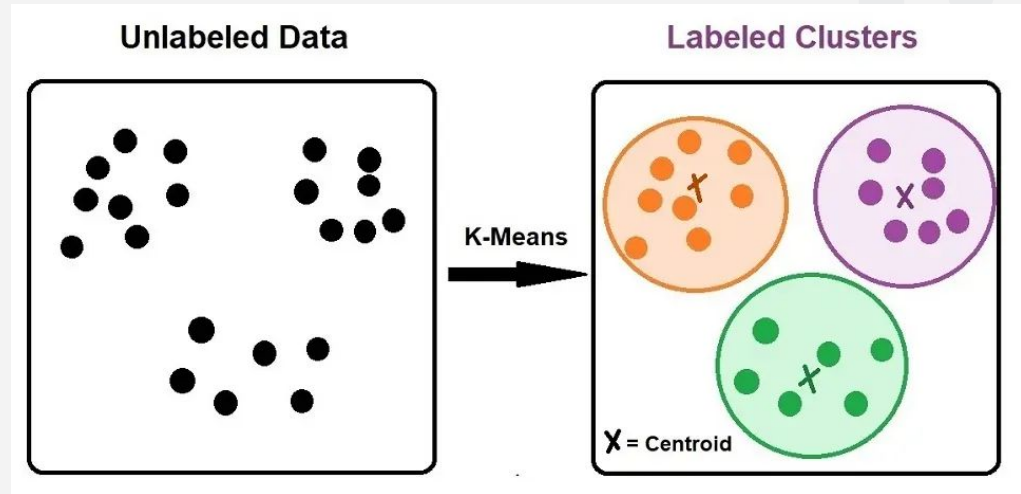
Clustering dengan pendekatan partisi atau sering disebut dengan partition-based clustering mengelompokkan data dengan memilah-milah data yang dianalisa ke dalam cluster-cluster yang ada. Contoh : K-Means

❑ Pendekatan Hirarki

Clustering dengan pendekatan hirarki atau sering disebut dengan hierarchical clustering mengelompokkan data dengan membuat suatu hirarki berupa dendogram dimana data yang mirip akan ditempatkan pada hirarki yang berdekatan dan yang tidak pada hirarki yang berjauhan. Contoh : Agglomerative Hierarchical Clustering

Algoritma K-Means

Apa dan Bagaimana



K-Means



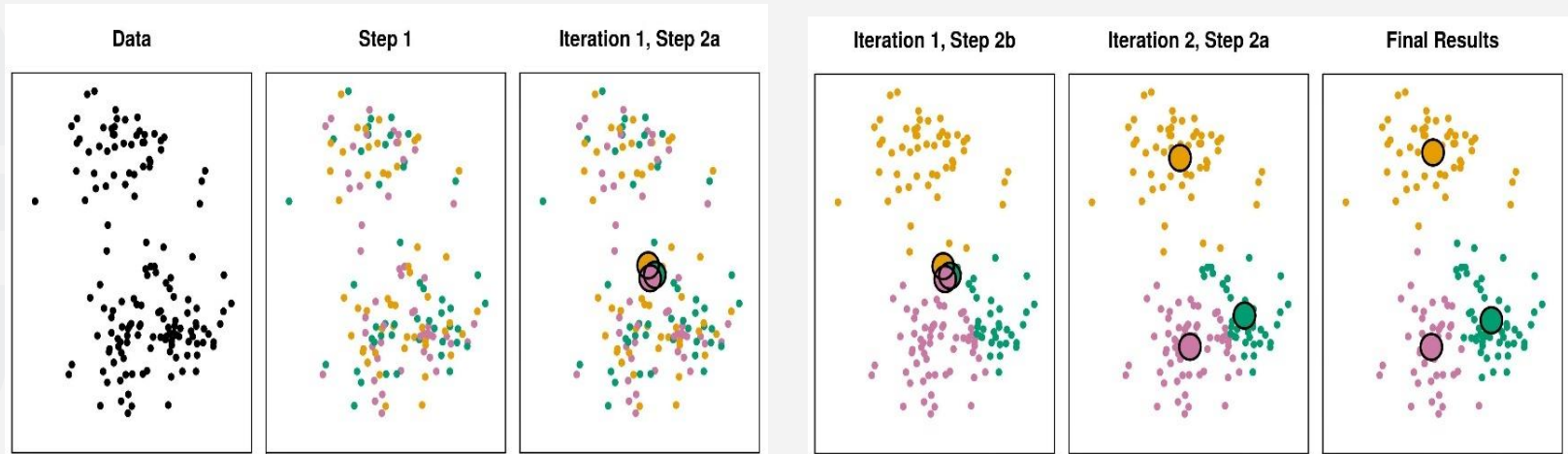
Cara kerja algoritma K-Means dalam membagi object – object di dalam data menjadi beberapa cluster adalah sebagai berikut (see visualization [here](#)):

1. Mula-mula adalah dengan membentuk sejumlah k titik, yang disebut dengan centroid (dimana nilai k merepresentasikan jumlah cluster). Langkah ini merupakan langkah untuk Memilih jumlah cluster awal (K) yang ingin dibuat.
2. Kemudian object – object di dalam data dihitung jarak kedekatan pada featurenya terhadap feature di semua centroid. Sehingga masing – masing object di dalam data akan berkumpul pada cluster yang terdekat darinya.
3. Langkah berikutnya adalah membuat centroid terbaru di tiap cluster dengan cara menghitung nilai rata – rata (mean) dari setiap feature pada seluruh object dalam satu kelompok, langkah ini bisa disebut dengan istilah penyempurnaan centroid. Sehingga terbentuklah centroid yang akan digunakan sebagai pusat cluster.

K-Means



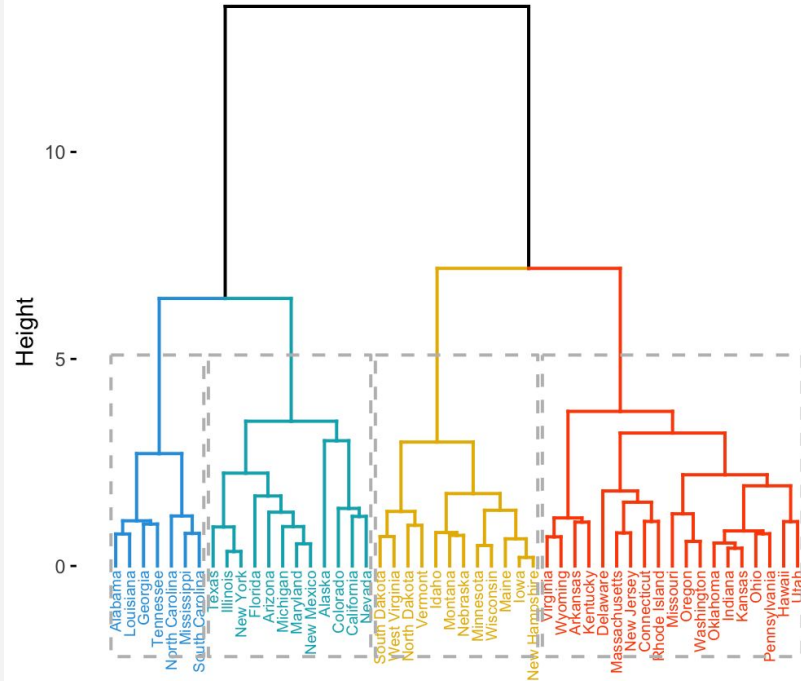
- Hitung kembali proses menghitung jarak kedekatan antar object di dalam data berdasarkan feature-nya terhadap centroid (Mengulang langkah ke-2), kemudian perbaharui kembali centroid pada masing – masing cluster berdasarkan rata – rata feature pada setiap object didalamnya. Proses ini terus dilakukan sampai centroid pada masing – masing cluster tidak berubah lagi atau telah konstan. Apabila centroid pada masing – masing cluster tidak berubah lagi maka telah selesai proses clustering dilakukan.



Algoritma Agglomerative Hierarchical Clustering

Apa dan Bagaimana

Cluster Dendrogram



Algoritma Agglomerative Hierarchical Clustering



- ❑ Secara prinsip, Agglomerative Hierarchical Clustering ini akan melakukan clustering secara berjenjang berdasarkan kemiripan tiap data. Sehingga pada akhirnya, pada ujung hierarki akan terbentuk cluster-cluster yang karakteristiknya berbeda satu sama lain, dan objek di satu cluster yang sama memiliki kemiripan satu sama lain.
- ❑ Hasil keseluruhan dari algoritma Agglomerative Hierarchical Clustering secara grafik dapat digambarkan sebagai tree, yang disebut dengan dendogram. Tree ini secara grafik menggambarkan proses penggabungan dari cluster – cluster yang ada, sehingga menghasilkan cluster dengan level yang lebih tinggi.

Algoritma Agglomerative Hierarchical Clustering



Dalam Agglomerative Hierarchical Clustering teknik pengelompokan yang paling dikenal adalah:

1. Single linkage (jarak terdekat atau tautan tunggal)

Teknik yang menggabungkan cluster-cluster menurut jarak antara anggota-anggota terdekat di antara dua cluster.

2. Average linkage (jarak rata-rata atau tautan rata-rata)

Teknik yang menggabungkan cluster-cluster menurut jarak rata-rata pasangan anggota masing-masing pada himpunan antara dua cluster.

3. Complete linkage (jarak terjauh atau tautan lengkap)

Teknik yang menggabungkan cluster-cluster menurut jarak antara anggota-anggota terjauh di antara dua cluster.

Algoritma Agglomerative Hierarchical Clustering



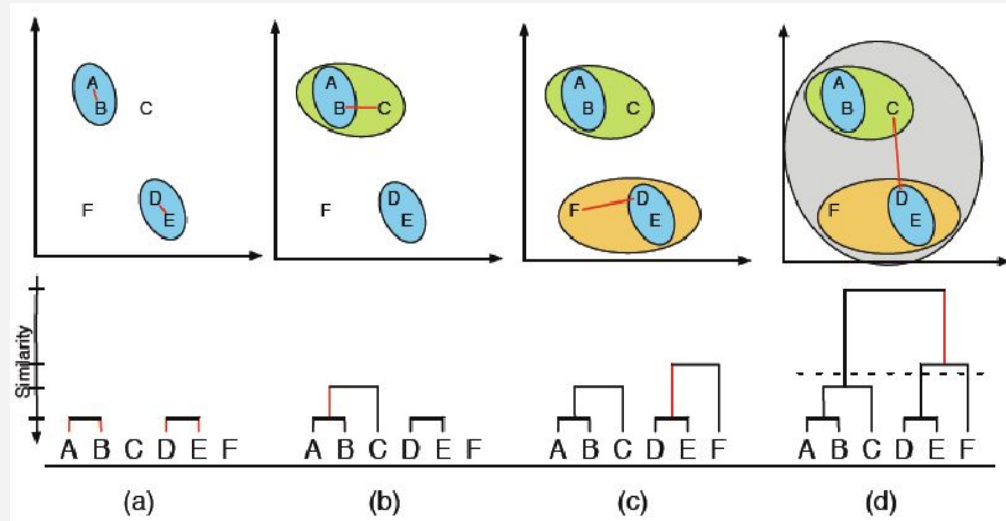
Cara kerja dari Agglomerative Hierarchical Clustering :

1. Menghitung matriks jarak pada tiap object - object di dalam data
2. Jika jarak objek a dengan b memiliki nilai jarak paling kecil dibandingkan jarak antar objek lainnya dalam matriks jarak, maka gabungkan dua cluster pada tahap pertama, sehingga object akan berpasangan dengan object yang lain yang merupakan jarak terdekat sehingga akan terjadi beberapa cluster.
3. Perbarui matriks jarak sesuai dengan teknik pengelompokkan Agglomerative Hierarchical Clustering untuk membentuk cluster terbaru.
4. Lakukan terus proses penghitungan kemiripan antar object di dalam matriks jarak sampai terbentuk cluster hanya satu, maka proses clustering telah selesai dilakukan.
5. Buat Dendrogram sebagai representasi dari proses clustering Agglomerative Hierarchical Clustering

Algoritma Agglomerative Hierarchical Clustering



5. Lakukan terus proses penghitungan kemiripan antar object didalam matriks jarak sampai terbentuk cluster hanya satu, maka proses clustering telah selesai dilakukan.
6. Jika jarak objek a dengan b memiliki nilai jarak paling kecil dibandingkan jarak antar objek lainnya dalam matriks jarak, maka gabungkan dua cluster pada tahap pertama, sehingga object akan berpasangan dengan object yang lain yang merupakan jarak terdekat sehingga akan terjadi beberapa cluster.



Implementasi

- K-Means & Agglomerative Hierarchical Clustering
- Hands-on Coding in Google Colab

```
31 def __init__(self, settings):
32     self.file = None
33     self.fingerprints = set()
34     self.logdupes = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, "requests.log"),
39                         "a")
40         self.file.seek(0)
41         self.fingerprints.update(s.request for s in self.data)
42
43 @classmethod
44 def from_settings(cls, settings):
45     debug = settings.getbool("DEBUG_LOGGING")
46     return cls(job_dir(settings), debug)
47
48 def request_seen(self, request):
49     fp = self.request_fingerprint(request)
50     if fp in self.fingerprints:
51         return True
52     self.fingerprints.add(fp)
53     if self.file:
54         self.file.write(fp + os.linesep)
55
56 def request_fingerprint(self, request):
57     return request_fingerprint(request)
```





Thank You

