Modul Praktikum

Kecerdasan Buatan

6

Unsupervised Learning

Unsupervised Learning

Pada algoritma unsupervised learning, data tidak memiliki label secara eksplisit dan model mampu belajar dari data dengan menemukan pola yang implisit. Sangat berbeda dengan supervised learning, unsupervised learning merupakan jenis learning yang hanya mempunyai variabel input tapi tidak mempunyai variabel output yang berhubungan. Tujuan dari Machine Learning ini adalah untuk memodelkan struktur data dan menyimpulkan fungsi yang mendeskripsikan data tersebut.

Unsupervised learning digunakan untuk analisis dan cluster dataset tanpa label. Algoritma ini menemukan pola tersembunyi atau mengelompokkan data tanpa intervensi manusia. Pengelompokan data dilakukan berdasarkan karakteristik data yang mirip. Dalam unsupervised learning memiliki 2 pendekatan yaitu clustering dan association rules.

Unsupervised learning adalah salah satu tipe algoritma machine learning yang digunakan untuk menarik kesimpulan dari dataset. Metode ini hanya akan mempelajari suatu data berdasarkan kedekatannya saja atau yang biasa disebut dengan clustering. Metode unsupervised learning yang paling umum adalah analisis cluster, yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data.

Cara Kerja Unsupervised Learning

Cara kerja algoritma ini yaitu dengan mencari pola tersembunyi (eksplisit) dari dataset yang diberikan. Unsupervised machine learning bekerja dengan menganalisis data yang tidak berlabel untuk menemukan pola tersembunyi dan menentukan korelasinya.

Pendekatan ini tidak menggunakan data training dan data test untuk melakukan prediksi maupun klasifikasi dengan tujuan mengelompokkan objek yang hampir sama dalam suatu area tertentu. Beberapa contoh algoritma yang dapat digunakan dalam unsupervised learning seperti, K-Means, Hierarchical clustering, DBSCAN, dan Fuzzy C-Means.

Contoh Unsupervised Learning

Salah satu contoh implementasi unsupervised learning adalah clustering. Dengan teknologi clustering, sebuah perusahaan dapat mengidentifikasi customer segmen yang berpotensial untuk menjual produk mereka. Perusahaan dapat mengidentifikasi customer segmen yang paling mungkin menggunakan layanan atau membeli produk mereka. Perusahaan juga dapat mengevaluasi segmen pelanggan lalu memutuskan untuk menjual produk guna memaksimalkan keuntungan mereka.

Contohnya ketika seorang Data Analyst ingin mengelompokkan client dari salah satu provider hosting di Indonesia berdasarkan kemiripan sifat dalam hal pendapatan umur, hobi, dan jenis pekerjaannya. Maka untuk mengelompokkan customer berdasarkan kemiripan sifat, machine learning tidak memerlukan data training. Melainkan menggunakan data yang ada langsung bisa mengelompokkan customer-customer tersebut.

Clustering

Klaster (cluster) adalah sebuah grup yang memiliki kemiripan tertentu. Pengklasteran adalah sebuah metode machine learning unsupervised untuk mengelompokkan objek-objek yang memiliki kemiripan (memiliki karakteristik yang mirip), ke dalam sebuah klaster. Karena termasuk kategori unsupervised, maka dataset yang digunakan model clustering tidak memiliki label.

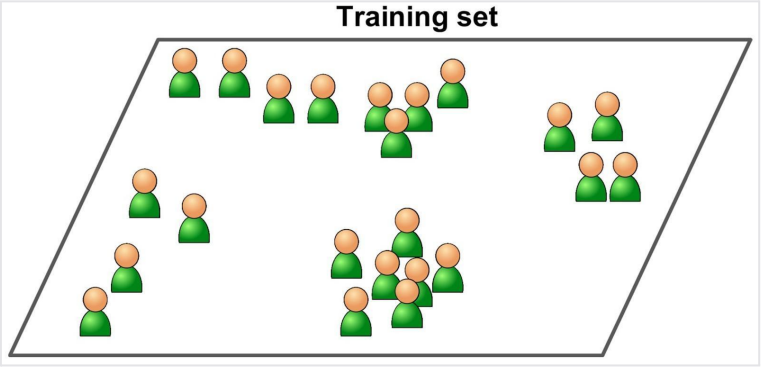
Semakin besar tingkat kemiripan/similarity (atau homogenitas) di dalam satu grup dan semakin besar tingkat perbedaan di antara grup, maka semakin baik (atau lebih berbeda) clustering tersebut. pengelompokan untuk pemahaman atau pengelompokan untuk penggunaan merupakan 2 tujuan dari clustering.

Proses pengelompokan untuk tujuan pemahaman hanya sebagai proses awal untuk kemudian dilanjutkan dengan pekerjaan seperti summarization (rata-rata, standar deviasi, pelabelan kelas untuk setiap kelompok sehingga dapat digunakan sebagai data training dalam klasifikasi supervised learning.

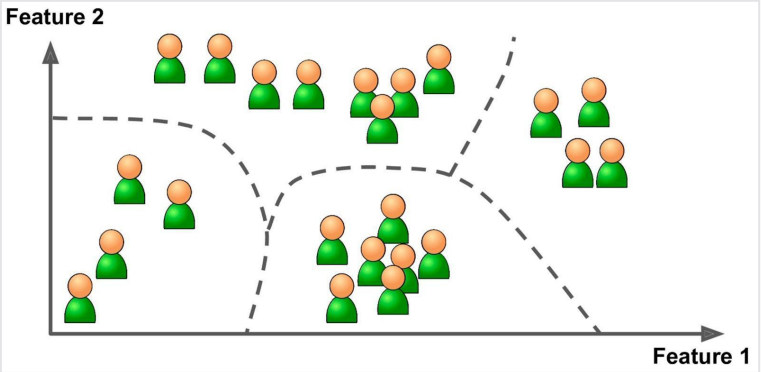
Proses pengelompokan untuk tujuan penggunaan biasanya mencari prototipe kelompok yang paling representatif terhadap data, memberikan abstraksi dari setiap proyek data dalam kelompok dimana sebuah data terletak di dalamnya.

Contohnya adalah ketika kita memiliki data pengunjung web toko online kita seperti gambar di bawah. Kemudian kita ingin mengembangkan sebuah model yang bisa

mengelompokkan pengunjung yang memiliki kemiripan. Misalnya, diketahui bahwa 80% pengunjung toko online Anda adalah perempuan, sementara 20% nya adalah laki-laki. 60% dari pengunjung perempuan mengunjungi toko online Anda pada hari kerja, sementara sisanya berkunjung pada akhir minggu. Contoh lain, 40% pengunjung toko online Anda berasal dari Pulau Jawa, 55% berasal dari pulau lain di seluruh Indonesia, dan 5% sisanya berasal dari luar negeri. Tujuan pengelompokkan kemiripan ini adalah agar kita mengetahui target market yang sesuai untuk setiap kelompok.



Sebuah model pengklasteran akan membandingkan atribut setiap pengunjung lalu membuat sebuah klaster yang diisi oleh pengunjung yang memiliki kemiripan karakteristik/atribut yang tinggi.



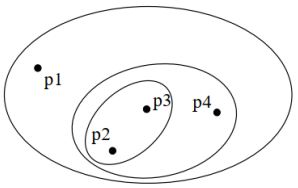
Contoh di atas dikenal juga sebagai customer segmentation, salah satu kasus yang populer di industri, di mana bisnis mengelompokkan pelanggan agar bisa memberikan penawaran yang sesuai untuk setiap kelompok. Misal, kelompok pengunjung wanita dengan rentang usia 25 sampai 35 tahun tentu akan memiliki selera yang berbeda dengan pengunjung wanita pada rentang usia 40 tahun ke atas. Customer segmentation ini penting agar setiap target kelompok mendapatkan

penawaran yang sesuai sehingga dapat memberikan kontribusi positif terhadap revenue toko.

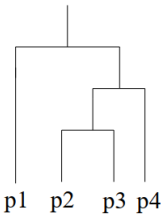
Jenis - jenis Clustering

Ada banyak pengelompokan clustering yang dapat digunakan. Beberapa diantaranya adalah exclusive vs non-exclusive, fuzzy vs non-fuzzy, partial vs complete, dan heterogeneous vs homogeneous. Pada praktikum ini akan dibahas tentang pengelompokkan berdasarkan struktur kelompok.

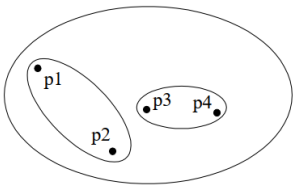
- **Struktur Kelompok**
  - Hierarchical
    sekumpulan cluster bersarang yang terorganisir sebagai hierarki pohon.



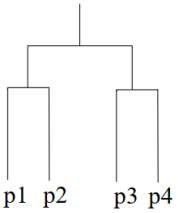
Traditional Hierarchical Clustering



Traditional Dendrogram

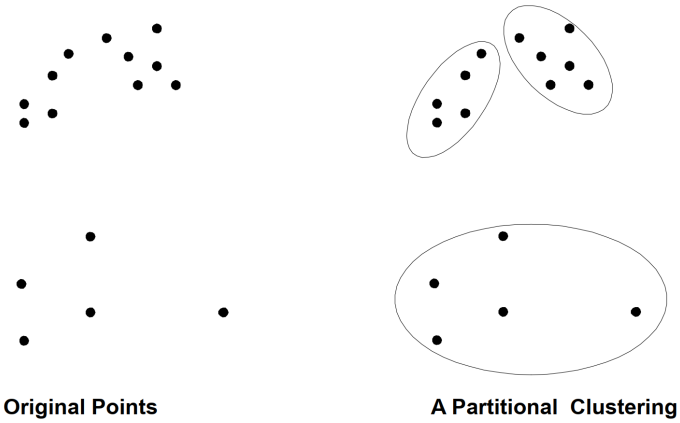


Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

- Partitioning
  Pembagian objek data ke cluster yang tidak saling tindih. Sehingga satu data hanya masuk ke satu cluster.



Original Points A Partitional Clustering

Algoritma Clustering

Ada banyak tipe algoritma clustering yang dapat digunakan. diantaranya adalah sebagai berikut.

- Density-based
  Data dikelompokkan dengan daerah berkonsentrasi data tinggi dikelilingi daerah berkonsentrasi data rendah. Baik digunakan untuk menghindari outliers.
- Distribution-based
  Semua data point dianggap bagian dari cluster berdasarkan probabilitasnya. Pembagian cluster-nya berdasarkan jarak antara data point ke center.
- Centroid-based
  Algoritma yang cepat dan efisien. Membagi data point ke dalam cluster berdasarkan jarak ke centroid. centroid adalah titik tengah dari suatu objek, dalam hal ini titik tengah dari cluster.
- Hierarchical-based
  Biasanya digunakan pada data yang bentuknya hierarki. Seperti database perusahaan atau taxonomi.

K-Means Clustering

Algoritma ini menemukan kelompok data dengan nilai squared error antara rata-rata empiris dari cluster dan point di cluster minimum. Algoritma ini termasuk dalam clustering dengan pendekatan **partitional**. Setiap cluster berhubungan dengan centroid (point tengah). Setiap point dimasukkan pada cluster dengan centroid terdekat. Pada algoritma ini nilai K == cluster harus ditentukan.

Centroid pada umumnya dipilih secara random. Biasanya centroid merupakan rata-rata dari point pada cluster. Untuk menentukan setiap data point masuk ke cluster mana, digunakan pengukuran kedekatan(jarak) dengan Euclidean distance, cosine similarity, korelasi, dan lain sebagainya.

Clustering Bunga Iris (2D)

```
# Import KMeans
from sklearn.cluster import KMeans

#Read data (Data Iris)
df =
pd.read_csv("https://raw.githubusercontent.com/Opensourcefordatascience/Data-sets/
master/Iris_Data.csv")

#Buat objek dari k means dengan jumlah cluster : 3
model = KMeans(n_clusters=3)

# Pilih atribut yang akan dilatih
points = df[["sepal_length","petal_length"]]

# Lakukan fit terhadap model
model.fit(points)
# Tentukan label untuk cluster
labels = model.labels_

print(labels)
```

Output:

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 2 1 2 2 2 2 1 2 2 2 2
 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 1 2
 2 1]
```

Evaluasi Hasil Pemodelan

```
# Buat dataframe dengan label dan spesies sebagai kolom
dfKmeans = pd.DataFrame({'labels': labels, 'Species': df.species})

# buat crosstab: ct
ct = pd.crosstab(dfKmeans["labels"],dfKmeans["Species"])

# Print ct
print(ct)
```

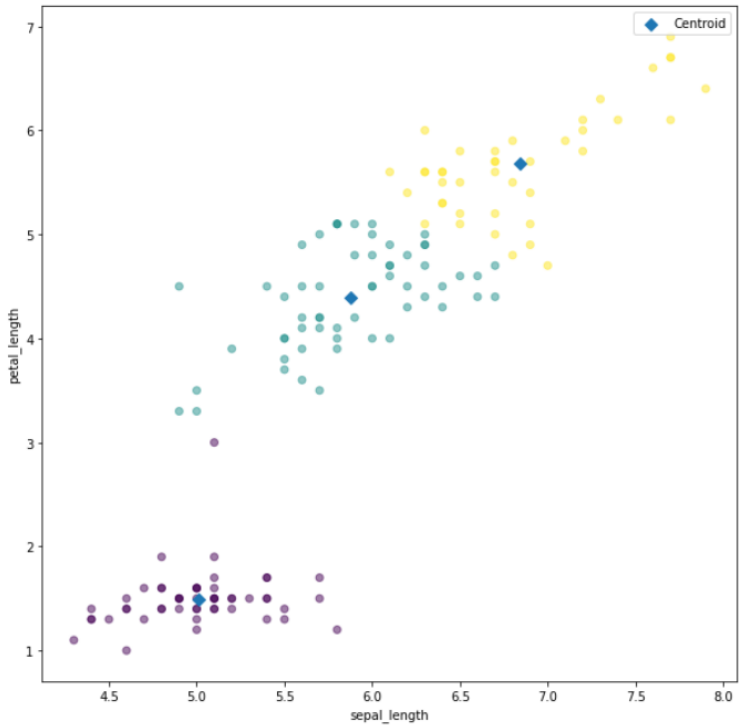
Output

Species	Iris-setosa	Iris-versicolor	Iris-virginica
labels			
0	0	4	37
1	50	1	0
2	0	45	13

Melihat centroid dari model yang sudah dibuat

```
# import visualisasi
import matplotlib.pyplot as plt

# mengambil semua baris pada kolom tertentu (dalam bentuk array)
xs = points.iloc[:,0]
ys = points.iloc[:,1]

# Menampung koordinat dari tiap centroid
centroids = model.cluster_centers_

centroids_x = centroids[:,0]
centroids_y = centroids[:,1]

# visualisasi cluster
plt.figure(figsize=(10,10))
plt.scatter(xs,ys,alpha=0.5,c=labels)
plt.scatter(centroids_x,centroids_y,marker="D",s=50,label="Centroid")
plt.xlabel("sepal_length")
plt.ylabel("petal_length")
plt.legend()
plt.show()
```

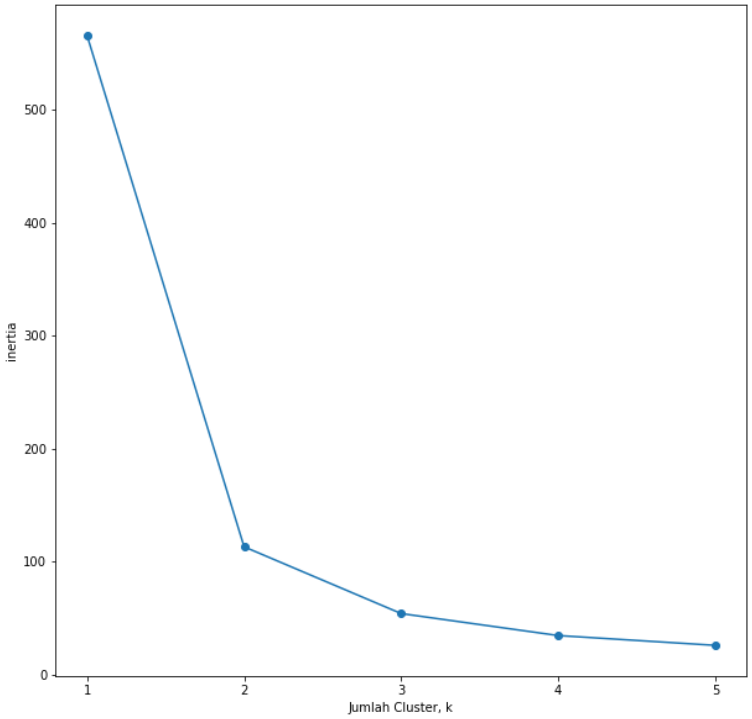
Output:



Menentukan Jumlah Cluster (Elbow Method)

```
ks = range(1, 6)
inertias = []

for k in ks:
    model = KMeans(n_clusters=k)
    model.fit(points)
    inertias.append(model.inertia_)

# Plot ks vs inertias
plt.figure(figsize=(10,10))
plt.plot(ks, inertias, '-o')
plt.xlabel('Jumlah Cluster, k')
plt.ylabel('inertia')
plt.xticks(ks)
plt.show()
```

Output

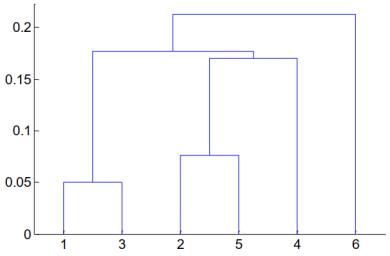


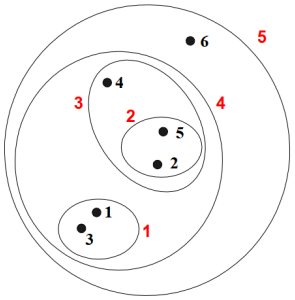
Hierarchical Clustering

Algoritma ini menghasilkan sekumpulan cluster bersarang dalam bentuk hirarki pohon. Algoritma ini dapat divisualisasikan menggunakan dendogram atau bubble. Algoritma ini tidak harus menentukan jumlah cluster. Jumlah cluster yang diinginkan didapatkan dengan memotong dendogram pada level yang tepat.

Terdapat 2 tipe hierarchical clustering, yaitu:

- Agglomerative





  - Dimulai dengan setiap point sebagai cluster sendiri-sendiri
  - Pada setiap langkah menggabungkan pasangan point terdekat hingga hanya tersisa satu cluster
- Divisive
  - Dimulai dengan satu cluster yang menampung semua point
  - Pada setiap langkah membagi cluster hingga setiap cluster hanya memiliki 1 point

Algoritma hierarchical tradisional menggunakan similarity atau distance matrix. Untuk menentukan similarity antar cluster dapat digunakan nilai Min(Single Link/Linkage), Max, Group Average, Distance antar centroid, atau metode lain seperti squared error.

Contoh penerapan hierarchical Agglomerative dengan similarity menggunakan nilai Min(Linkage).

```
# memasukkan library
from matplotlib import pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage, complete
import scipy
import matplotlib
import numpy as np
import pandas as pd

# mengecek versi library
print(f'versi matplotlib {matplotlib.__version__}', # 3.5.1
      f'versi scipy {scipy.__version__}', # 1.8.0
      f'versi numpy {np.__version__}', # 1.22.2
      f'versi pd {pd.__version__}', # 1.4.1
      sep='\n')

# mengambil dataset
dataset =
pd.read_csv("https://raw.githubusercontent.com/Opensourcefordatascience/Data-se
ts/master/Iris_Data.csv")
dataset.head(5)

# mengambil semua baris pada kolom pertama dan kedua
points = dataset.iloc[:,[0,1]].values
points

# menentukan algoritma similarity yang digunakan
linkage_hiearchical = linkage(points, method='ward')

# visualisasi menggunakan dendogram
plt.figure(figsize=(25, 10), facecolor="white")
dendrogram(linkage_hiearchical)
plt.title('Dendrogram')
plt.xlabel('Sepal')
plt.ylabel('Euclidean distances')
plt.show()

# import algoritma clustering
from sklearn.cluster import AgglomerativeClustering

# menentukan jenis hierarchy yang digunakan
hierarchical_cluster = AgglomerativeClustering(n_clusters=3,
```

```
affinity='euclidean', linkage='ward')

# training model
predicted_hierachical_clustering = hierarchical_cluster.fit_predict(points)

# visualisasi hasil model
plt.scatter(points[predicted_hierachical_clustering==0, 0],
points[predicted_hierachical_clustering==0, 1], s=30, c='red', label ='Cluster
1')
plt.scatter(points[predicted_hierachical_clustering==1, 0],
points[predicted_hierachical_clustering==1, 1], s=30, c='blue', label ='Cluster
2')
plt.scatter(points[predicted_hierachical_clustering==2, 0],
points[predicted_hierachical_clustering==2, 1], s=30, c='green', label
='Cluster 3')
plt.title('Clusters of Customers (Hierarchical Clustering Model)')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.show()
```

Source

1. DQLab
2. https://www.uc.ac.id/ict/perbedaan-supervised-learning-and-unsupervised-learning/
3. https://www.techtarget.com/searchenterpriseai/definition/supervised-learning