Laporan Case-Based 2
Machine Learning

Dosen Pembimbing:
DEDE ROHIDIN, DRS.,MT
CII3C3-IF-44-08

Disusun oleh:
1301200421 Herjanto Janawisuta

S1 Informatika
Universitas Telkom
Bandung, Jawa Barat 2022

Kata Pengantar

Dengan mengucapkan puji dan rasa syukur kepada Allah SWT yang telah memberikan rahmat sehingga kita dapat menyelesaikan tugas dari mata kuliah Pembelajaran Mesin dengan tema “Implementasi K-Means/DBScan/Hierarchical)” dengan benar dan tepat waktu.

Untuk memenuhi nilai tugas pada mata kuliah Pembelajaran Mesin, maka dibuatkan tugas yang dapat saya selesaikan. Tidak hanya itu, tujuan dari pembuatan laporan dan pengerjaan tugas ini adalah untuk menambah wawasan tentang pembahasan Implementasi K-Means bagi kita semua.

Saya mengucapkan terima kasih kepada semua pihak mulai dari Pak Tjokorda Agung Budi Wirayuda, S.T., M.T. selaku dosen pembimbing yang telah memberikan saya tugas besar untuk membuat Algoritma K-Means.

Saya sangat menyadari laporan yang saya susun masih jauh dari kata sempurna. Tetapi saya akan terus berusaha untuk selalu menjadi lebih baik untuk kedepannya.

Pernyataan : Saya mengerjakan tugas ini dengan cara yang tidak melanggar aturan perkuliahan dan kode etik akademisi.

Bandung, 7 November 2022

Herjanto Janawisuta

BAB I
PENDAHULUAN

Case-Based 2

Scenario

Mengikuti keberhasilan tugas sebelumnya, Anda diberi kesempatan lebih lanjut untuk mengesankan Atasan atau Dosen Anda mengenai kemampuan Anda untuk mengimplementasikan algoritma unsupervised dan menganalisis data. Anda diminta untuk melakukan beberapa analisis dan menghasilkan luaran yang berguna menggunakan dataset yang tersedia online sebagai berikut:

[untuk NIM akhir GENAP gunakan data ini]
https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data

[untuk NIM akhir GANJIL gunakan data ini]
https://archive.ics.uci.edu/ml/datasets/Water+Treatment+Plant

Dataset masih terdapat missing value atau outlier. Harap lakukan perbaikan terhadap hal ini, selanjutnya anda harus menganalisa data tersebut. Jika perlu konversi variable kategori menjadi integer. Jika perlu lakukan normalisasi data melalui fitur rescaling. Jika perlu lakukan analisa elbow. Jika perlu lakukan analisa dengan plot data secara visual. Jika perlu lakukan transformasi data secara logaritmik. Dan masih banyak kemungkinan Analisa data yang dapat anda lakukan.
Hint: Anda bebas memilih satu dari tiga alat analisis data yaitu Weka, R, atau Python untuk membantu Anda menganalisis data dan menunjukkan pra-pemrosesan data yang diperlukan.

Tugas Anda

Tujuan dari tugas ini yaitu Anda diharapkan mampu menjelaskan, mengimplementasikan, menganalisis, dan mendesain teknik pembelajaran mesin unsupervised learning yaitu k means/dbscan/hierarchical.

Pertama, selidiki masalah kualitas data yang telah diberikan di atas. Jelaskan keputusan Anda mengenai pendekatan pra-pemrosesan data. Jelajahi kumpulan data dengan meringkas data menggunakan statistik dan mengidentifikasi masalah kualitas data apa pun. Tidak ada batasan jumlah ringkasan yang akan dilaporkan tetapi Anda diharapkan hanya melaporkan yang paling relevan.

Kedua, pilih salah satu dari metode unsupervised learning yang telah dipelajari yaitu kmeans/dbscan/hierarchical. Anda hanya perlu memilih satu metode untuk diterapkan. Gunakan algoritma tersebut untuk memberikan beberapa output/outcome dengan menggunakan variasi parameter, kemudian membuat laporannya dan menganalisis hasilnya.

BAB II
PEMBAHASAN

2.1 Ikhtisar Kumpulan Data yang Dipilih

Pada bagian ini saya mendapatkan data ganjil dengan nim yang diakhiri dengan “1” maka dari itu saya akan memakai data water treatment sebagai acuan pengerjaan soal.

	0	1	2	3	4	5	6	7	8	9	...	29	30	31	32	33	34	35	36	37	38
0	D-1/3/90	44101	1.50	7.8	?	407	166	66.3	4.5	2110	...	2000	?	58.8	95.5	?	70.0	?	79.4	87.3	99.6
1	D-2/3/90	39024	3.00	7.7	?	443	214	69.2	6.5	2660	...	2590	?	60.7	94.8	?	80.8	?	79.5	92.1	100
2	D-4/3/90	32229	5.00	7.6	?	528	186	69.9	3.4	1666	...	1888	?	58.2	95.6	?	52.9	?	75.8	88.7	98.5
3	D-5/3/90	35023	3.50	7.9	205	588	192	65.6	4.5	2430	...	1840	33.1	64.2	95.3	87.3	72.3	90.2	82.3	89.6	100
4	D-6/3/90	36924	1.50	8.0	242	496	176	64.8	4.0	2110	...	2120	?	62.7	95.6	?	71.0	92.1	78.2	87.5	99.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
522	D-26/8/91	32723	0.16	7.7	93	252	176	56.8	2.3	894	...	942	?	62.3	93.3	69.8	75.9	79.6	78.6	96.6	99.6
523	D-27/8/91	33535	0.32	7.8	192	346	172	68.6	4.0	988	...	950	?	58.3	97.8	83.0	59.1	91.1	74.6	90.7	100
524	D-28/8/91	32922	0.30	7.4	139	367	180	64.4	3.0	1060	...	1136	?	65.0	97.1	76.2	66.4	82.0	77.1	88.9	99
525	D-29/8/91	32190	0.30	7.3	200	545	258	65.1	4.0	1260	...	1326	39.8	65.9	97.1	81.7	70.9	89.5	87.0	89.5	99.8
526	D-30/8/91	30488	0.21	7.5	152	300	132	69.7	?	1073	...	1224	?	69.5	?	81.7	76.4	?	81.7	86.4	?

Pada data di atas masih berbentuk file *raw* yang masih butuh banyak perbaikan, dari pengurangan dimensi kolom yang tidak dibutuhkan dan juga nilai yang valuenya masih bernilai ‘?’ sehingga tidak bisa kita langsung proses untuk dilakukannya metode K-Means.

2.2 Ringkasan pra-pemrosesan data yang diusulkan

Pada bagian pra-pemrosesan data ini saya memakai beberapa metode atau teknik diharapkan bisa membuat data water-treatment menjadi lebih baik lagi sebelum dilakukan klasifikasi terhadap dataset tersebut. Pada kesempatan ini saya menggunakan metode sebagai berikut:

1) Menambah Kolom dengan list atribut yang sudah ada pada file water-treatment.names.

```
[496] df.columns = ['Date', 'Q-E', 'ZN-E', 'PH-E', 'DBO-E', 'DQO-E', 'SS-E', 'SSV-E', 'SED-E', 'COND-E', 'PH-P', 'DBO-P', 'SS-P',
                    'SSV-P', 'SED-P', 'COND-P', 'PH-D', 'DBO-D', 'DQO-D', 'SS-D', 'SSV-D', 'SED-D', 'COND-D', 'PH-S', 'DBO-S', 'DQO-S',
                    'SS-S', 'SSV-S', 'SED-S', 'COND-S', 'RD-DBO-P', 'RD-SS-P', 'RD-SED-P', 'RD-DBO-S', 'RD-DQO-S', 'RD-DBO-G', 'RD-DQO-G',
                    'RD-SS-G', 'RD-SED-G']
```

2) Mengubah data '?' menjadi NaN, dan menampilkan null value.

```
df.replace('?',np.NaN, inplace=True)
```

```
df.isnull().sum()

Date         0
Q-E         18
ZN-E         3
PH-E         0
DBO-E       23
DQO-E        6
SS-E         1
SSV-E       11
SED-E       25
COND-E       0
PH-P         0
DBO-P       40
SS-P         0
SSV-P       11
SED-P       24
COND-P       0
PH-D         0
DBO-D       28
DQO-D        9
SS-D         2
SSV-D       13
SED-D       25
COND-D       0
PH-S         1
DBO-S       23
DQO-S       18
SS-S         5
SSV-S       17
SED-S       28
COND-S       1
RD-DBO-P    62
RD-SS-P      4
RD-SED-P    27
RD-DBO-S    40
RD-DQO-S    26
RD-DBO-G    36
RD-DQO-G    25
RD-SS-G      8
RD-SED-G    31
dtype: int64
```

3) Mengubah null value menjadi nilai rata-rata.

```
imputer = SimpleImputer(missing_values=np.nan, strategy = 'mean')
df.iloc[:,1:39] = imputer.fit_transform(df.iloc[:,1:39])
```

```
df.isnull().sum()
```

```
Date        0
Q-E         0
ZN-E        0
PH-E        0
DBO-E       0
DQO-E       0
SS-E        0
SSV-E       0
SED-E       0
COND-E      0
PH-P        0
DBO-P       0
SS-P        0
SSV-P       0
SED-P       0
COND-P      0
PH-D        0
DBO-D       0
DQO-D       0
SS-D        0
SSV-D       0
SED-D       0
COND-D      0
PH-S        0
DBO-S       0
DQO-S       0
SS-S        0
SSV-S       0
SED-S       0
COND-S      0
RD-DBO-P    0
RD-SS-P     0
RD-SED-P    0
RD-DBO-S    0
RD-DQO-S    0
RD-DBO-G    0
RD-DQO-G    0
RD-SS-G     0
RD-SED-G    0
dtype: int64
```

4) Mendrop Atribut ‘Date’ dan mengubah tipe dataset menjadi float.

```
df = df.drop(['Date'], axis=1)
df = df.astype(float)
```

5) Scaling dataset dengan range 1-10.

```
#Scaling
df = ((df - df.min()) / (df.max() - df.min())) * 9 + 1
df.describe()
```

2.3 Menerapkan algoritma K-Means

Pada kesempatan ini saya menggunakan metode K-Means untuk mengerjakan tugas ini. Metode K-Means sendiri merupakan algoritma unsupervised learning yang dipakai untuk mengelompokkan dataset yang belum di label ke dalam kluster yang berbeda. Simbol K pada K-means clustering menentukan jumlah cluster yang digunakan.

Kemudian metode tersebut diaplikasikan ke dalam sebuah dataset untuk melabeli atau mengklasifikasikan data ke dalam cluster tertentu.

1) Function Initialize random centroids

```
#Initialize random centroids
def random_centroids(data,k):
  centroids = []
  for i in range(k):
    centroid = df.apply(lambda x: float(x.sample()))
    centroids.append(centroid)
  return pd.concat(centroids, axis=1)
```

Pada function di atas akan men-*generate* sebaran centroid secara acak dengan range 1 sampai 10 karena pada data ini saya telah scaling dataset ini dengan range 1 sampai dengan 10.

2) Function Get labels

```
#Label each data point
def get_labels(data, centroids):
  distances = centroids.apply(lambda x:np.sqrt(((data - x)**2).sum(axis=1)))
  return distances.idxmin(axis=1)
```

Menghitung jarak dari setiap data ke centroid atau cluster.

3) Function update centroids

```
#Update Centroids
def new_centroids(data, labels, k):
  return data.groupby(labels).apply(lambda x: np.exp(np.log(x).mean())).T
```

Menghitung geometric mean pada setiap data untuk menemukan centroid yang baru.

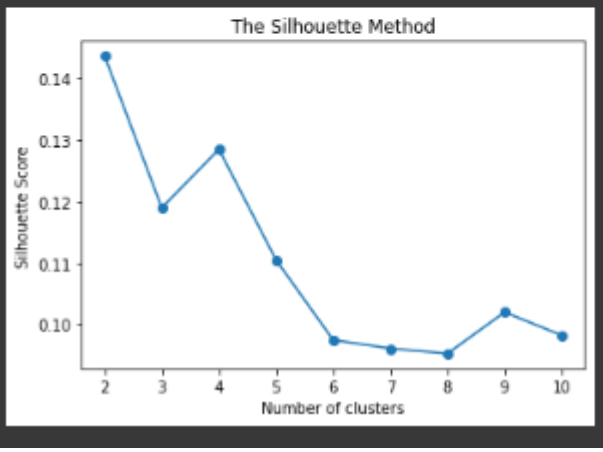
4) Function Scatter Plot

```
#Scatter Plot
def plot_clusters(data, labels, centroids, iteration):
  pca = PCA(n_components=2)
  data_2d = pca.fit_transform(data)
  centroids_2d = pca.transform(centroids.T)
  # clear_output(wait=True)
  plt.title(f'Iteration {iteration}')
  plt.scatter(x=data_2d[:,0], y=data_2d[:,1], c=labels)
  plt.scatter(x=centroids_2d[:,0], y=centroids_2d[:,1])
  plt.show()
```

Menampilkan grafik Scatter Plot dengan setiap iterasinya.

5) Silhouette Method

```
from sklearn.metrics import silhouette_score
silhouette = []
for i in range(2, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++',
                    max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(df)
    silhouette.append(silhouette_score(df, kmeans.labels_))
plt.plot(range(2, 11), silhouette, marker='o')
plt.title('The Silhouette Method')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette Score')
plt.show()
```



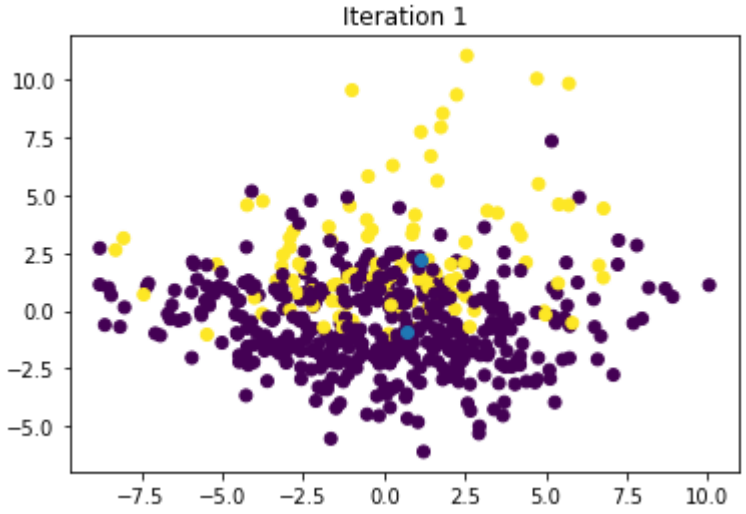
Disini, saya menggunakan Silhouette Method untuk menemukan jumlah K terbaik untuk dipakai pada algoritma K-means, dan grafik menunjukan score tertinggi berada pada K = 2.

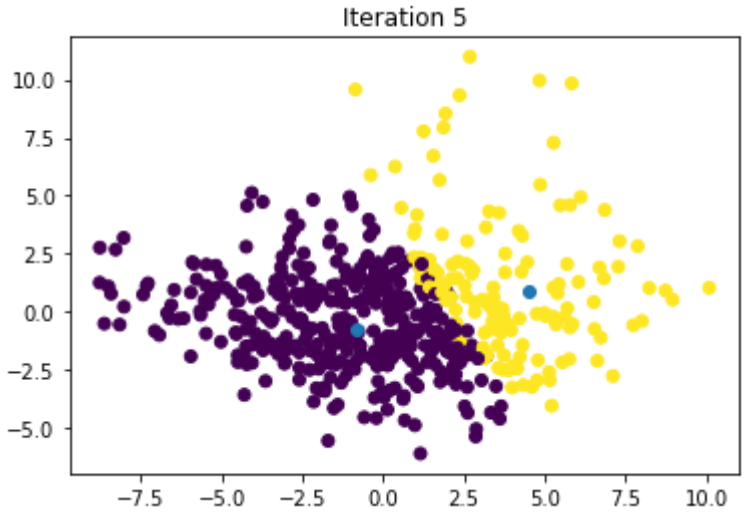
6) Memanggil seluruh function

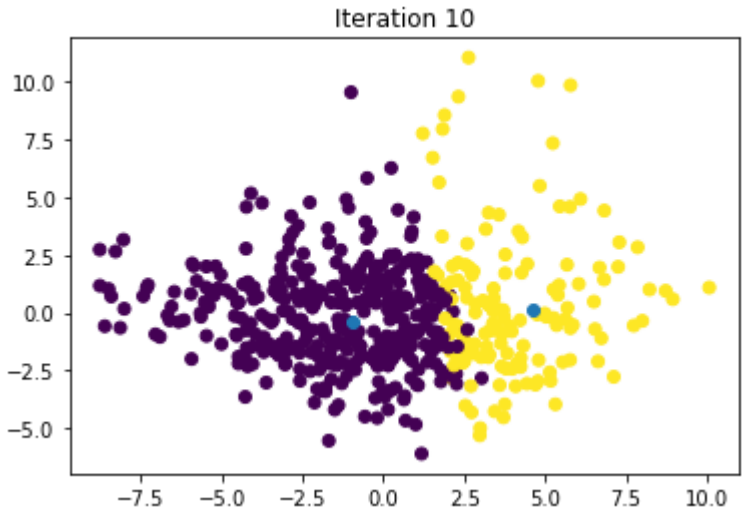
```
max_iterations = 100
k = 2

centroids = random_centroids(df, k)
old_centroids = pd.DataFrame()
iteration = 1

while iteration < max_iterations and not centroids.equals(old_centroids):
  old_centroids = centroids
  labels = get_labels(df, centroids)
  centroids = new_centroids(df, labels, k)
  plot_clusters(df, labels, centroids, iteration)
  df['clusters'] = labels
  iteration += 1
```

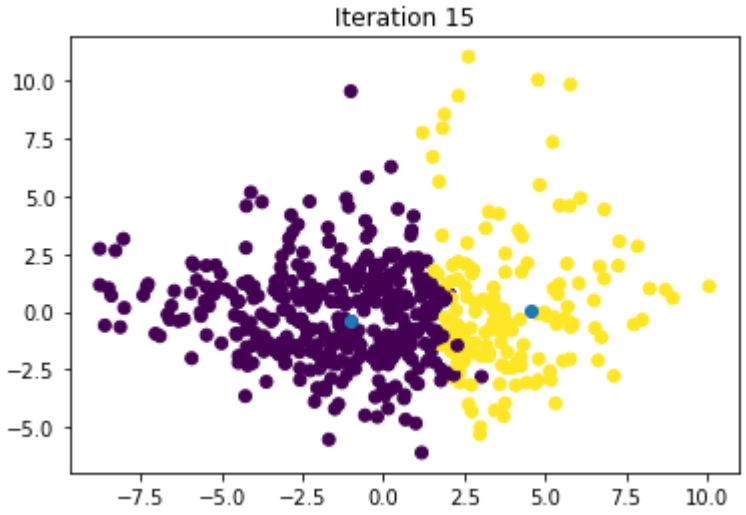
Disini saya memanggil seluruh function yang telah dibuat, dan melakukan while loop. Iterasi akan berhenti jika telah menyentuh 100, atau pada saat posisi centroid sudah tidak berubah.

2.4 Evaluasi hasil







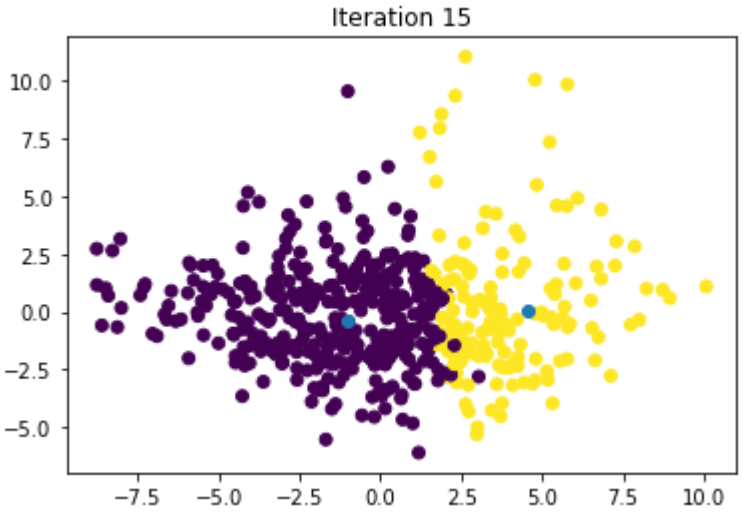

Iteration 15
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

BAB III
KESIMPULAN

Setelah kita perhatikan pada scatter plot yang telah kita visualisasikan, kita bisa melihat nilai k yang paling optimal dengan melihat persebaran data di setiap cluster ada pada nilai k=2 dengan persebaran data sebagai berikut,

```
df['clusters'].value_counts()
```

```
0    375
1    152
Name: clusters, dtype: int64
```



Saya telah mencoba untuk menggunakan Elbow Method untuk mencari K optimal, tetapi karena data yang terlalu banyak, grafik Elbow Method terlalu sulit untuk dibaca, sehingga saya menggunakan Silhouette Method

Link Colab dan Youtube:
https://colab.research.google.com/drive/1U5vQ4R_Bpn_zBCj9HDsoqfWWKOnv_DyW#scrollTo=PtAJUrTzgfdk
https://drive.google.com/drive/folders/1aW4OBjzBcVe-gt7y3xEds-rPCJ8X-2rI?usp=share_link