

PRAKTIKUM DATA WAREHOUSING DAN DATA MINING
MODUL 13
PRINCIPAL COMPONENT ANALYSIS



Disusun oleh:
Adinda Aulia Hapsari
L200220037

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2024

Setelah kegiatan selesai, lembar kerja ini dicetak (di-print) dan dikumpulkan ke asisten.	(Diisi oleh Asisten)
NIM : L200220037	Nilai Praktek :
Nama : Adinda Aulia Hapsari	
Nama Asisten : Diva Halimah	Tanda Tangan :
Tanggal Praktikum : 27 Desember 2024	

KEGIATAN PRAKTIKUM

Contoh Kasus:

Terdapat sebuah dataset tentang review dari sebuah platform travel TripAdvisor, yang berjumlah 980 reviews. Kemudian dataset tersebut terdapat 11 features sebagai berikut:

Atribut 1: Unique user id	Merupakan atribut yang berisikan id user
Atribut 2: Average user feedback on art galleries	Merupakan atribut yang berisikan nilai rata-rata feedback dari art gallery
Atribut 3: Average user feedback on dance clubs	Merupakan atribut yang berisikan nilai rata-rata feedback dari dance club
Atribut 4: Average user feedback on juice bars	Merupakan atribut yang berisikan nilai rata-rata feedback dari juice bars
Atribut 5: Average user feedback on restaurants	Merupakan atribut yang berisikan nilai rata-rata feedback dari restoran
Atribut 6: Average user feedback on museums	Merupakan atribut yang berisikan nilai rata-rata feedback dari museum
Atribut 7: Average user feedback on resorts	Merupakan atribut yang berisikan nilai rata-rata feedback dari resorts
Atribut 8: Average user feedback on parks/picnic spots	Merupakan atribut yang berisikan nilai rata-rata feedback dari taman dan tempat wisata
Atribut 9: Average user feedback on beaches	Merupakan atribut yang berisikan nilai rata-rata feedback dari Pantai
Atribut 10: Average user feedback on theaters	Merupakan atribut yang berisikan nilai rata-rata feedback dari teater
Atribut 11: Average user feedback on religious institutions	Merupakan atribut yang berisikan nilai rata-rata feedback dari lembaga keagamaan

Hipotesis:

Bagaimana mendapatkan dataset dengan dimensi yang lebih rendah dengan menggunakan algoritma PCA berdasarkan features yang telah diketahui pada dataset?

13.4.1 Mengimport Library

Meng-import library yang diperlukan, yaitu library pandas, numpy, dan PCA. Untuk meng-import library yang akan digunakan, kita meng import library pandas, numpy, dan PCA.

```
[2]: import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
```

13.4.2 Membaca Dataset

Membaca dataset dari sebuah directory. Disini, kita mengambil dataset tripadvisor.csv yang diambil dari platform gitea. Dataset tersebut akan tersimpan di variable data sebagai sebuah dataframe.

```
[3]: data = pd.read_csv("tripadvisor.csv")
print(data)
```

	User ID	Category 1	Category 2	Category 3	Category 4	Category 5	\
0	User 1	0.93	1.80	2.29	0.62	0.80	
1	User 2	1.02	2.20	2.66	0.64	1.42	
2	User 3	1.22	0.80	0.54	0.53	0.24	
3	User 4	0.45	1.80	0.29	0.57	0.46	
4	User 5	0.51	1.20	1.18	0.57	1.54	
..	
975	User 976	0.74	1.12	0.30	0.53	0.88	
976	User 977	1.25	0.92	1.12	0.38	0.78	
977	User 978	0.61	1.32	0.67	0.43	1.30	
978	User 979	0.93	0.20	0.13	0.43	0.30	
979	User 980	0.93	0.56	1.13	0.51	1.34	
	Category 6	Category 7	Category 8	Category 9	Category 10		
0	2.42	3.19	2.79	1.82	2.42		
1	3.18	3.21	2.63	1.86	2.32		
2	1.54	3.18	2.80	1.31	2.50		
3	1.52	3.18	2.96	1.57	2.86		
4	2.02	3.18	2.78	1.18	2.54		
..		
975	1.38	3.17	2.78	0.99	3.20		
976	1.68	3.18	2.79	1.34	2.80		
977	1.78	3.17	2.81	1.34	3.02		
978	0.40	3.18	2.98	1.12	2.46		
979	2.36	3.18	2.87	1.34	2.40		

[980 rows x 11 columns]

13.4.3 Menghapus Kolom yang Tidak Diperlukan

Disini kita akan mengeliminasi feature yang tidak diperlukan, dalam hal ini adalah User ID, karena User ID tidak akan berpengaruh pada hasil clustering. Pada kode di bawah ini, kita mengaplikasikan fungsi drop pada variable data yang menyimpan data frame. Fungsi drop

disini bertujuan untuk menghapus kolom User ID. Setelah User ID dihapus, dataframe disimpan ke dalam variable X. Kemudian variable X dicetak.

```
[5]: #Menghapus Kolom yang Tidak Diperlukan
X = data.drop(columns=['User ID'])
print(X)
```

	Category 1	Category 2	Category 3	Category 4	Category 5	Category 6	\
0	0.93	1.80	2.29	0.62	0.80	2.42	
1	1.02	2.20	2.66	0.64	1.42	3.18	
2	1.22	0.80	0.54	0.53	0.24	1.54	
3	0.45	1.80	0.29	0.57	0.46	1.52	
4	0.51	1.20	1.18	0.57	1.54	2.02	
..	
975	0.74	1.12	0.30	0.53	0.88	1.38	
976	1.25	0.92	1.12	0.38	0.78	1.68	
977	0.61	1.32	0.67	0.43	1.30	1.78	
978	0.93	0.20	0.13	0.43	0.30	0.40	
979	0.93	0.56	1.13	0.51	1.34	2.36	
	Category 7	Category 8	Category 9	Category 10			
0	3.19	2.79	1.82	2.42			
1	3.21	2.63	1.86	2.32			
2	3.18	2.80	1.31	2.50			
3	3.18	2.96	1.57	2.86			
4	3.18	2.78	1.18	2.54			
..			
975	3.17	2.78	0.99	3.20			
976	3.18	2.79	1.34	2.80			
977	3.17	2.81	1.34	3.02			
978	3.18	2.98	1.12	2.46			
979	3.18	2.87	1.34	2.40			

[980 rows x 10 columns]

13.4.4 Menampilkan Jumlah Fitur

Kode di bawah ini akan menampilkan jumlah fitur yang ada pada dataset setelah User ID dieliminasi. Fungsi shape menampilkan bentuk atau ordo dari sebuah dataset. Untuk mengetahui jumlah baris pada dataset X, kita menggunakan X.shape[0] sedangkan apabila kita ingin mengetahui jumlah kolom pada dataset X, kita menggunakan X.shape[1] pada kode Python tersebut. Kemudian jumlah kolom kita simpan pada variable num_features, yang akan kita cetak. Dalam kasus ini, kita memiliki 10 kolom/fitur.

```
[9]: # Menampilkan Jumlah Fitur
num_features = X.shape[1]
print("Jumlah fitur input:", num_features, "fitur")

Jumlah fitur input: 10 fitur
```

13.4.5 Implementasi PCA dengan Jumlah Fitur Awal

Selanjutnya, kita jalankan kode untuk PCA dengan cara memanggil fungsi PCA. Adapun fungsi PCA memiliki parameter wajib yaitu: `n_components` yang berarti jumlah fitur yang ada pada dataset. Disini, dikarenakan jumlah fitur pada dataset setelah User ID dieliminasi adalah sejumlah 10 fitur, maka kita menggunakan `n_components = 10` yang berarti kita akan membuat 10 principal components menggunakan teknik PCA.

```
[10]: #Implementasi PCA dengan Jumlah Fitur Awal
      pca = PCA(n_components=num_features)
      pca.fit(X)
```

[10]: PCA

PCA(n_components=10)

13.4.6 Menampilkan Hasil Variance pada Tiap Principal Components

Kemudian, kita menampilkan hasil variance pada tiap principal components. Pada kode di bawah ini kita menerapkan fungsi `enumerate` untuk mendapatkan indeks dan data dari setiap elemen di dalam sebuah list. Elemen pada list `pca.explained_variance_ratio_` indeksnya akan tersimpan dalam variable `i` sedangkan data akan tersimpan dalam variable `j`. Setelah itu, pada setiap iterasi, program akan mencetak nilai variance untuk setiap principal components. Hasilnya, semakin rendah urutan principal components, nilai variance semakin tinggi.

```
[11]: #Menampilkan Hasil Variance pada Tiap Principal Components
      for i,j in enumerate(pca.explained_variance_ratio_):
          print("Fitur independen ke-", (i+1),
                "menghasilkan variance ratio sebesar",
                round(j,7))
```

Fitur independen ke- 1 menghasilkan variance ratio sebesar 0.4252009
Fitur independen ke- 2 menghasilkan variance ratio sebesar 0.1772314
Fitur independen ke- 3 menghasilkan variance ratio sebesar 0.1245329
Fitur independen ke- 4 menghasilkan variance ratio sebesar 0.0731861
Fitur independen ke- 5 menghasilkan variance ratio sebesar 0.0693468
Fitur independen ke- 6 menghasilkan variance ratio sebesar 0.0538007
Fitur independen ke- 7 menghasilkan variance ratio sebesar 0.0412973
Fitur independen ke- 8 menghasilkan variance ratio sebesar 0.0258732
Fitur independen ke- 9 menghasilkan variance ratio sebesar 0.0095227
Fitur independen ke- 10 menghasilkan variance ratio sebesar 8e-06

13.4.7 Menampilkan Beberapa Principal Component Pertama Dengan Cumulative Explained Ratio Minimal 90%

Pada kode di bawah ini kita menambahkan kondisi dimana kita akan mengambil nilai variance kumulatif dari principal components jika sudah mencapai angka minimal 0.9. Disini, kita

menambahkan variable `cumulative_variance` untuk menyimpan nilai variance kumulatif untuk principal components, juga variable `num_pc` untuk menyimpan jumlah principal components saat iterasi dijalankan.

Pada setiap iterasi, nilai `num_pc` bertambah satu dan nilai `cumulative_variance` bertambah sesuai dengan nilai variance pada setiap principal components; program akan mencetak nilai variance untuk setiap principal components untuk cumulative variance maksimal 0.9.

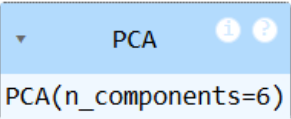
```
[12]: #Menampilkan Beberapa Principal Component Pertama Dengan
      #Cumulative Explained Ratio Minimal 90%
      cumulative_variance = 0
      num_pc = 0
      for i,j in enumerate(pca.explained_variance_ratio_):
          if cumulative_variance < 0.9:
              num_pc += 1
              cumulative_variance += j
              print("Fitur independen ke-",
                    (i+1),
                    "menghasilkan variance ratio sebesar",
                    round(j,7))
```

```
Fitur independen ke- 1 menghasilkan variance ratio sebesar 0.4252009
Fitur independen ke- 2 menghasilkan variance ratio sebesar 0.1772314
Fitur independen ke- 3 menghasilkan variance ratio sebesar 0.1245329
Fitur independen ke- 4 menghasilkan variance ratio sebesar 0.0731861
Fitur independen ke- 5 menghasilkan variance ratio sebesar 0.0693468
Fitur independen ke- 6 menghasilkan variance ratio sebesar 0.0538007
```

13.4.8 Implementasi PCA dengan Jumlah Fitur yang Dikurangi

Selanjutnya, kita jalankan kode untuk PCA untuk mengurangi dimensi pada dataset. Disini kita akan menggunakan fungsi PCA dengan `n_components` sejumlah 6 dikarenakan jumlah yang dapat mewakili 90% dari dataset adalah sebanyak 6 fitur, maka kita menggunakan `n_components = 6` yang berarti kita akan membuat 6 principal components menggunakan teknik PCA.

```
[14]: #Implementasi PCA dengan Jumlah Fitur yang Dikurangi
      pca_reduced = PCA(n_components=num_pc)
      pca_reduced.fit(X)
```

[14]: 

TUGAS

Terdapat dataset pada GPS Trajectories yang dapat diunduh pada halaman GPS+Trajectories berikut: <http://archive.ics.uci.edu/ml/datasets/>

Setelah itu, bukalah dataset `go_track_tracks.csv`.

Pada dataset tersebut terdapat 163 data dengan atribut sebagai berikut:

1. id: id dari objek
2. id_android: perangkat yang digunakan untuk membaca objek
3. speed: kecepatan rata-rata (km/h)
4. time: waktu tempuh perjalanan (h)
5. distance: jarak total (km)
6. rating: rating lalu lintas perjalanan. (3- baik, 2- normal, 1-buruk).
7. rating_bus: rating bus (1 – Penumpang bus sedikit, 2 – Penumpang Bus cukup banyak, 3- Penumpang Bus banyak).
8. rating_weather: rating cuaca (1- hujan, 2- cerah,).
9. mobil_atau_bus: (1 - mobil, 2 bus)
10. linha: informasi tentang bus yang melakukan jalur tersebut

Kemudian kerjakanlah soal-soal berikut ini:

```
[1]: import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
```

```

id      id_android      speed      time      distance      rating      rating_bus \
0      1      0      19.210586      0.138049      2.652      3      0
1      2      0      30.848229      0.171485      5.290      3      0
2      3      1      13.560101      0.067699      0.918      3      0
3      4      1      19.766679      0.389544      7.700      3      0
4      8      0      25.807401      0.154801      3.995      2      0
..      ...      ...      ...      ...      ...      ...
158      38081      24      30.051732      0.218756      6.574      2      0
159      38082      24      30.173788      0.255387      7.706      3      0
160      38084      25      1.153772      0.013001      0.015      1      3
161      38090      26      0.843223      0.007116      0.006      3      1
162      38092      27      1.372998      0.016752      0.023      3      1

rating_weather      car_or_bus      linha
0      0      1      NaN
1      0      1      NaN
2      0      2      NaN
3      0      2      NaN
4      0      1      NaN
..      ...      ...      ...
158      0      1      carro
159      0      1      carro
160      2      2      721 - CASTELO BRANCO SUISSA
161      2      2      002 - FERNANDO COLLOR DIA
162      2      2      060 - PADRE PEDRO CAMPUS

[163 rows x 10 columns]
```

1. Tentukan berapa jumlah fitur input yang digunakan untuk PCA. Bagaimana cara anda mendapatkan nilai tersebut?

Menghapus fitur id, id android dan linha karena tidak akan berpengaruh pada hasil clustering. Sisa fitur adalah 7.

```
[3]: X = data.drop(columns = ['id', 'id_android', 'linha'])
      print(X)
```

	speed	time	distance	rating	rating_bus	rating_weather	\
0	19.210586	0.138049	2.652	3	0	0	
1	30.848229	0.171485	5.290	3	0	0	
2	13.560101	0.067699	0.918	3	0	0	
3	19.766679	0.389544	7.700	3	0	0	
4	25.807401	0.154801	3.995	2	0	0	
..
158	30.051732	0.218756	6.574	2	0	0	
159	30.173788	0.255387	7.706	3	0	0	
160	1.153772	0.013001	0.015	1	3	2	
161	0.843223	0.007116	0.006	3	1	2	
162	1.372998	0.016752	0.023	3	1	2	
	car_or_bus						
0	1						
1	1						
2	2						
3	2						
4	1						
..	...						
158	1						
159	1						
160	2						
161	2						
162	2						

[163 rows x 7 columns]

```
num_features = X.shape[1]
print('Jumlah fitur input :', num_features, 'fitur')
```

Jumlah fitur input : 7 fitur

2. Tuliskan algoritma PCA dengan n_components sebesar jumlah fitur input.

```
[11]: #2
      pca = PCA(n_components = num_features)
      pca.fit(X)
```

```
[11]: PCA
      PCA(n_components=7)
```

3. Setelah itu, tampilkan nilai variance ratio untuk setiap principal components. Kemudian, tentukan ada berapa fitur independen yang dapat memenuhi 90% cumulative variance ratio.


```
[12]: #3
for i, j in enumerate(pca.explained_variance_ratio_):
    print('Fitur independen ke-',
          (i+1),
          'Menghasilkan variance ratio sebesar',
          round(j, 7))
```

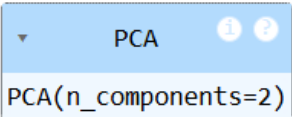
Fitur independen ke- 1 Menghasilkan variance ratio sebesar 0.8850815
 Fitur independen ke- 2 Menghasilkan variance ratio sebesar 0.1090902
 Fitur independen ke- 3 Menghasilkan variance ratio sebesar 0.0036053
 Fitur independen ke- 4 Menghasilkan variance ratio sebesar 0.0015101
 Fitur independen ke- 5 Menghasilkan variance ratio sebesar 0.0003889
 Fitur independen ke- 6 Menghasilkan variance ratio sebesar 0.0002053
 Fitur independen ke- 7 Menghasilkan variance ratio sebesar 0.0001187

```
[13]: #Menampilkan Beberapa Principal Component Pertama Dengan
#Cumulative Explained Ratio Minimal 90%
cumulative_variance = 0
num_pc = 0
for i, j in enumerate(pca.explained_variance_ratio_):
    if cumulative_variance < 0.9:
        num_pc += 1
        cumulative_variance += j
    print('Fitur independen ke_',
          (i+1),
          'Menghasilkan variance ratio sebesar',
          round(j, 7))
```

Fitur independen ke_ 1 Menghasilkan variance ratio sebesar 0.8850815
 Fitur independen ke_ 2 Menghasilkan variance ratio sebesar 0.1090902

4. Cetaklah Data pada beberapa principal components pertama dengan Cumulative Explained Ratio Minimal 90%.

```
[14]: #Implementasi PCA dengan Jumlah Fitur yang Dikurangi
pca_reduced = PCA(n_components = num_pc)
pca_reduced.fit(X)
```

[14]:  PCA
 PCA(n_components=2)