TUGAS LAB IF540 MACHINE LEARNING

WEEK 03 : Principal Component Analysis

Semester Ganjil 2022/2023

In [138…
```python
import datetime
import uuid

# Fill in your name and NIM
myName = "Christopher Darren"
myNIM = "00000054804"
```

In [2]:
```python
myDate = datetime.datetime.now()
myDevice = str(uuid.uuid1())

# Header
print("Name: \t\t{}".format(myName))
print("NIM: \t\t{}".format(myNIM))
print("Start: \t\t{}".format(myDate))
print("Device ID: \t{}".format(myDevice))
```

```
Name:           Christopher Darren
NIM:            00000054804
Start:          2023-02-23 10:25:42.431844
Device ID:      c09423f1-b329-11ed-a8f4-f02f74a116e8
```

Dataset yang dipakai:

1. 60,000+ Images of Cars – sumber : https://www.kaggle.com/datasets/prondeau/the-car-connection-picture-dataset?select=Acura_ILX_2013_28_16_110_15_4_70_55_179_39_FWD_5_4_4dr_Bbw.jpg
2. Red Wine Quality – sumber : https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009

Hasil kerja

In [139…
```python
# Your codes are here (replace the following codes)
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns; sns.set()
```

In [140…
```python
#reading data

df= pd.read_csv(r"D:\SEMESTER 4\IF540 Machine Learning\LAB\week3\winequality-red.csv")
df.head(5)
```

Out[140]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

In [141…
```python
df.info()
```

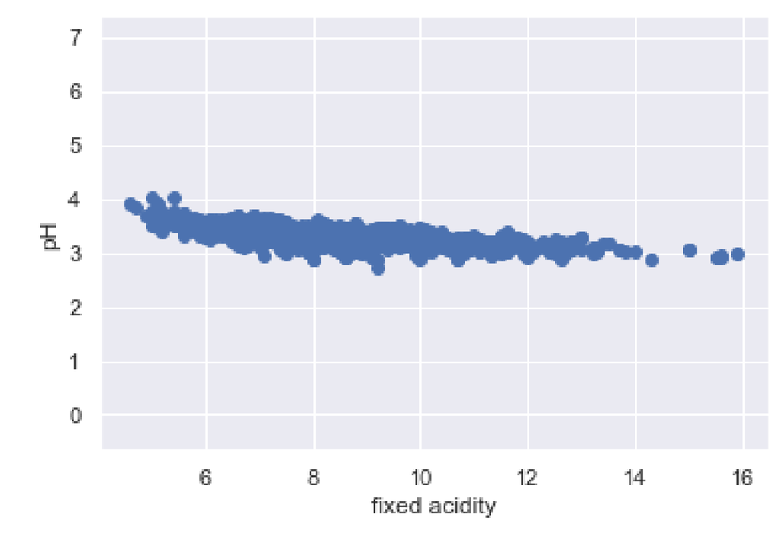
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

In [142…
```python
df.describe()
```

Out[142]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

In [143…
```python
plt.scatter(df.iloc[:,0], df.iloc[:, 8])
plt.xlabel("fixed acidity")
plt.ylabel("pH")
plt.axis('equal');
```



In [144…
```python
print(df)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076   
1               7.8             0.880         0.00             2.6      0.098   
2               7.8             0.760         0.04             2.3      0.092   
3              11.2             0.280         0.56             1.9      0.075   
4               7.4             0.700         0.00             1.9      0.076   
...             ...               ...          ...             ...        ...   
1594            6.2             0.600         0.08             2.0      0.090   
1595            5.9             0.550         0.10             2.2      0.062   
1596            6.3             0.510         0.13             2.3      0.076   
1597            5.9             0.645         0.12             2.0      0.075   
1598            6.0             0.310         0.47             3.6      0.067   

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0  0.99780  3.51       0.56   
1                    25.0                  67.0  0.99680  3.20       0.68   
2                    15.0                  54.0  0.99700  3.26       0.65   
3                    17.0                  60.0  0.99800  3.16       0.58   
4                    11.0                  34.0  0.99780  3.51       0.56   
...                   ...                   ...      ...   ...        ...   
1594                 32.0                  44.0  0.99490  3.45       0.58   
1595                 39.0                  51.0  0.99512  3.52       0.76   
1596                 29.0                  40.0  0.99574  3.42       0.75   
1597                 32.0                  44.0  0.99547  3.57       0.71   
1598                 18.0                  42.0  0.99549  3.39       0.66   

      alcohol  quality  
0         9.4        5  
1         9.8        5  
2         9.8        5  
3         9.8        6  
4         9.4        5  
...       ...      ...  
1594     10.5        5  
1595     11.2        6  
1596     11.0        6  
1597     10.2        5  
1598     11.0        6  

[1599 rows x 12 columns]
```

In [145…
```python
for data in df:
    print(data)
```

```
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
```

In [146…
```python
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(df[['fixed acidity','pH']].dropna())
```

Out[146]:
```
PCA(n_components=2)
```

In [147…
```python
print(pca.components_)
```

```
[[ 0.99815587 -0.06070307]
 [ 0.06070307  0.99815587]]
```

In [148…
```python
print(pca.explained_variance_)
```

```
[3.04258119 0.01267038]
```

In [149…
```python
#df.iloc[:,1]
df
```

Out[149]:

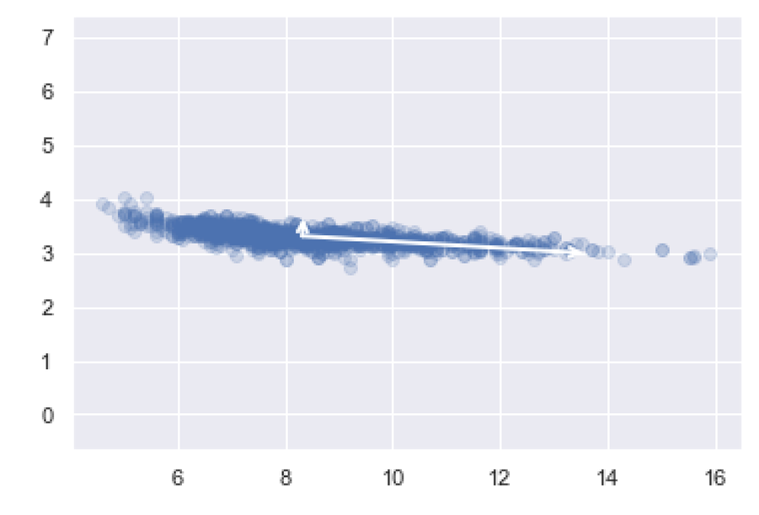
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

In [150…
```python
def draw_vector(v0, v1, ax=None):
    ax = ax or plt.gca()
    arrowprops=dict(arrowstyle="->", linewidth=2, shrinkA=0, shrinkB=0,)
    ax.annotate('', v1, v0, arrowprops=arrowprops)

plt.scatter(df["fixed acidity"], df["pH"], alpha=0.2)
for length, vector in zip(pca.explained_variance_, pca.components_):
    v = vector * 3 * np.sqrt(length)
    draw_vector(pca.mean_, pca.mean_+v)
plt.axis("equal")
```

Out[150]:
```
(4.034999999999999, 16.465, 2.6765000000000003, 4.0735)
```

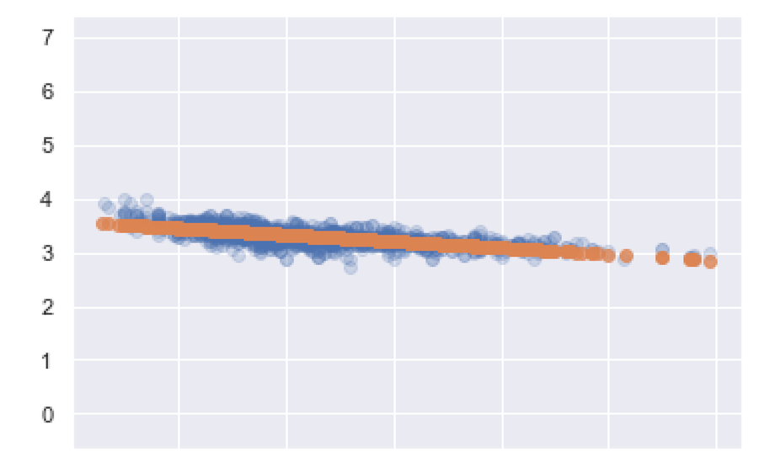


PCA as dimensionality reduction

In [151…
```python
pca = PCA(n_components=1)
pca.fit(df[['fixed acidity','pH']])
df_pca = pca.transform(df[['fixed acidity','pH']])
print("original shape: ", df[['fixed acidity','pH']].shape)
print("transformed shape: ", df_pca.shape)
```

```
original shape:  (1599, 2)
transformed shape:  (1599, 1)
```

In [152…
```python
df_new = pca.inverse_transform(df_pca)
plt.scatter(df['fixed acidity'], df['pH'], alpha=0.2)
plt.scatter(df_new[:, 0], df_new[:, 1], alpha=0.8)
plt.axis('equal');
```



Choosing the number of component(using picture)

In [153…
```python
#!pip install Pillow
```

In [154…
```python
from PIL import Image
```

In [155…
```python
#import os
```

In [156…
```python
#menginput gambar
image_ = Image.open("D:\SEMESTER 4\IF540 Machine Learning\LAB\Acura_ILX_2013_28_16_110_15_4_70_55_179_39_FWD_5_4_4dr_Bbw.jpg")
```

In [157…
```python
image_ = np.array(image_)
print(image_.shape)
nsamples, nx , ny = image_.shape
img = image_.reshape((nsamples, nx*ny))
```

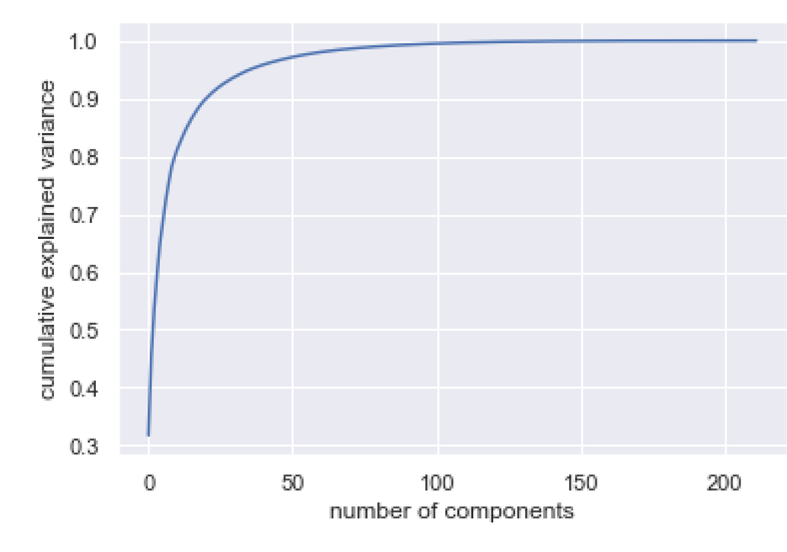
```
(212, 320, 3)
```

In [137…
```python
Image.fromarray(image_)
```

Out[137]:

Calculating the cumulative explained variance ratio:

In [158…
```python
pca = PCA().fit(img)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance');
```



PCA as Noise Filtering

In [199…
```python
np.random.seed(42)
noisy = np.random.normal(img, 50)
noisy.shape
```

Out[199]:
```
(212, 960)
```

In [200…
```python
Image.fromarray(noisy.astype(np.uint8))
```

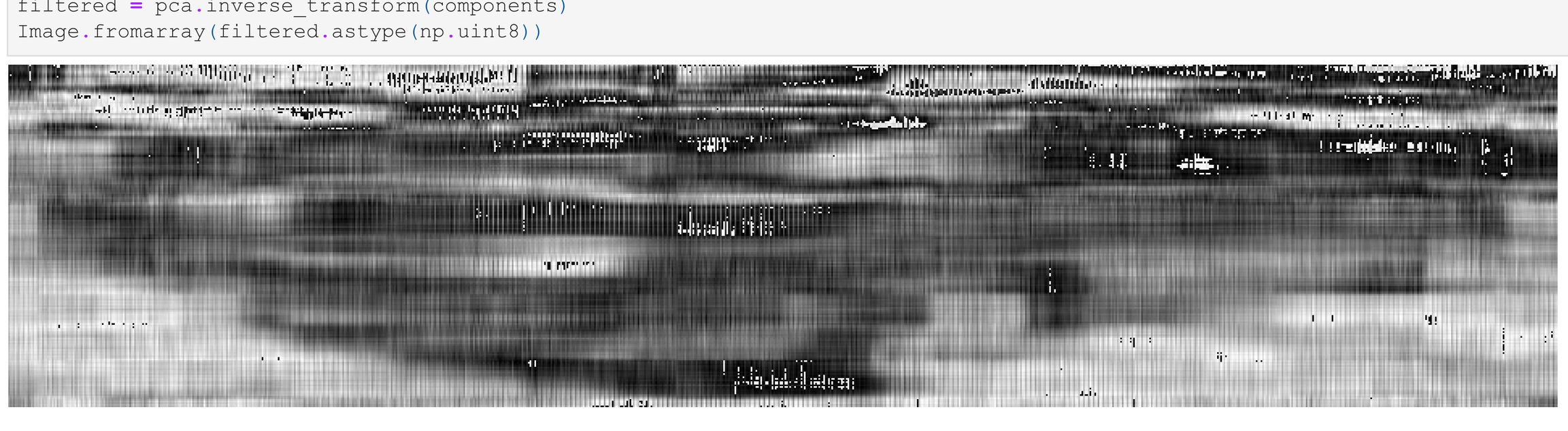
Out[200]:

In [201…
```python
pca = PCA(0.50).fit(noisy)
pca.n_components_
```

Out[201]:
```
8
```

In [202…
```python
components = pca.transform(noisy)
filtered = pca.inverse_transform(components)
Image.fromarray(filtered.astype(np.uint8))
```

Out[202]:



Kesimpulan

Berikan simpulan yang dilakukan dari hasil kerja menggunakan algoritma dan 2 dataset yang dipilih. Simpulan bisa berkisar antara (bisa di modifikasi):

```
PCA digunakan untuk mempelajari dimensity reduction data.
PCA juga bisa digunakan untuk noise filtering data.
mahasiswa dilatih kembali bagaimana caranya untuk membuat plot secara benar.
```

In [204…
```python
# Footer
myDate = datetime.datetime.now()
print("I certify that this is my own work.")
print("Signed by:")
print("Name: \t{}".format(myName))
print("NIM: \t{}".format(myNIM))
print("Time-stamp:\t{}".format(myDate))
```

```
I certify that this is my own work.
Signed by:
Name:   Christopher Darren
NIM:    00000054804
Time-stamp:     2023-03-01 21:30:03.076547
```

Save the notebook, then convert the notebook to html (by running the next code).

In [205…
```
!jupyter nbconvert --to html "./IF540_Kelas EL_00000054804_Christopher Darren_Week03.ipynb" --output-dir="./"
```

```
[NbConvertApp] Converting notebook ./IF540_Kelas EL_00000054804_Christopher Darren_Week03.ipynb to html
[NbConvertApp] Writing 1304972 bytes to IF540_Kelas EL_00000054804_Christopher Darren_Week03.html
```

Next step:

- convert the generated html file to PDF using the online tool: https://www.sejda.com/html-to-pdf
- choose the following settings:
  - Page size: One long page
  - Page Orientation: auto
  - Use print stylesheet
- Submit your ipython notebook and PDF files

Markdown basics https://markdown-guide.readthedocs.io/en/latest/basics.html#