Principal Component Analysis

DMPR: Lab activities for Week 04

Subhasis Ray

February 16,

2023

In this class we shall explore principal component analysis for data reduction and visualization.

1 Import libraries

```
#%%
import numpy as np
from scipy import signal as sig
import h5py as h5
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.decomposition import PCA
```

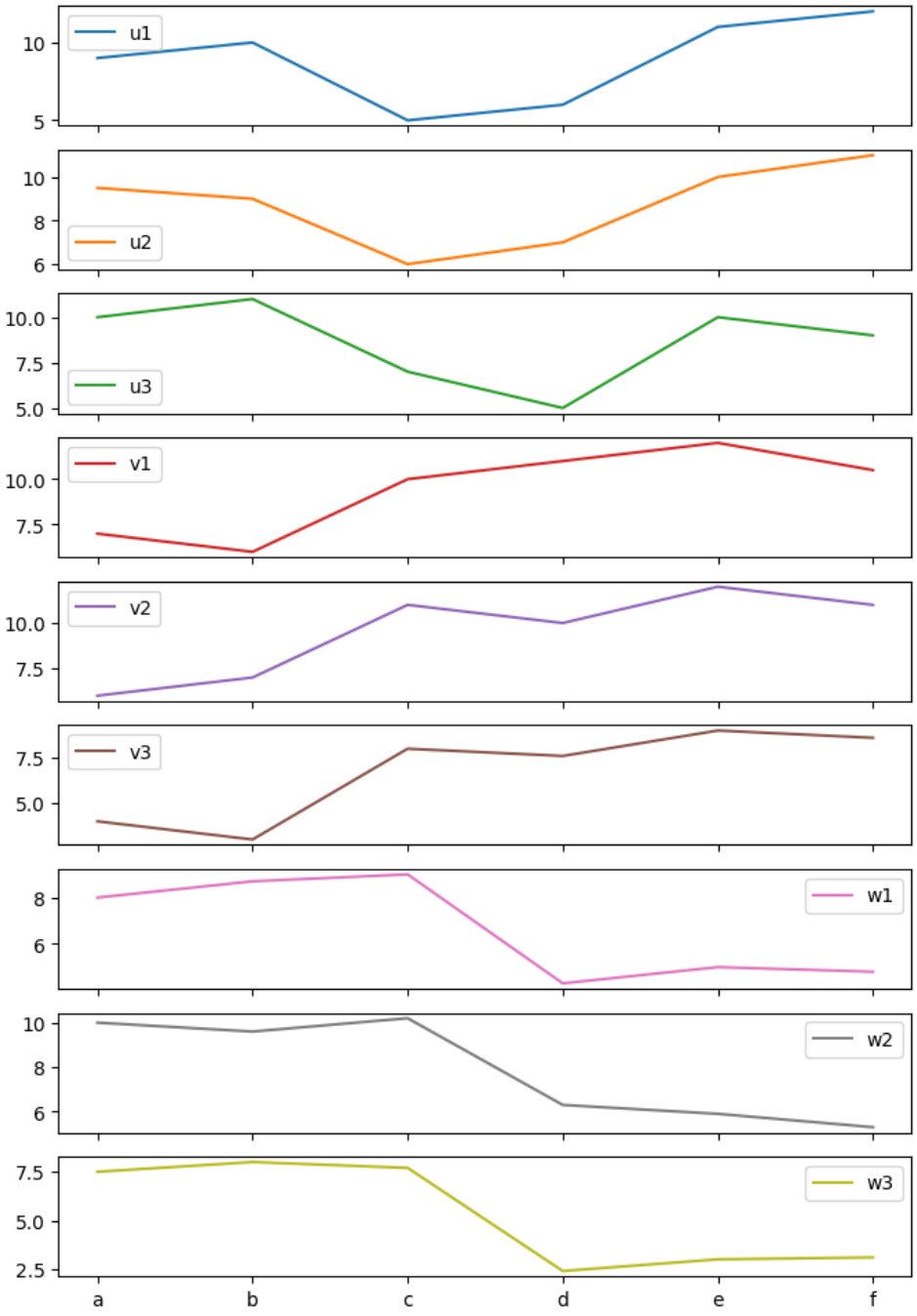
2 Dummy data (inspired by the nature article)

```
data = pd.DataFrame(data=[[9.0, 10.0, 5.0, 6.0, 11.0, 12.0],
                          [9.5, 9.0, 6.0, 7.0, 10.0, 11.0],
                          [10.0, 11.0, 7.0, 5.0, 10.0, 9.0],
                          [7.0, 6.0, 10.0, 11.0, 12.0, 10.5],
                          [6.0, 7.0, 11.0, 10.0, 12.0, 11.0],
                          [4.0, 3.0, 8.0, 7.6, 9.0, 8.6],
                          [8.0, 8.7, 9.0, 4.3, 5.0, 4.8],
                          [10.0, 9.6, 10.2, 6.3, 5.9, 5.3],
                          [7.5, 8.0, 7.7, 2.4, 3.0, 3.1]],
                    index=['u1', 'u2', 'u3', 'v1', 'v2', 'v3',
    ↪   'w1', 'w2', 'w3'], columns=['a', 'b', 'c', 'd', 'e', 'f'])
```

3 Display the data: use dataframe.plot.line: 5 marks

Plot the data so that the rows are plotted as lines on sequential axes (9 axes in a column). Thus in the top axis, you will plot the values of a, b, c, d, e and f on the y axis with the column index on the x axis.

Ans:



4 Create a PCA object and then fit the data

```
#%% Do PCA

pca =
PCA(copy=True) ret
= pca.fit(data)

print('Components:', ret.components_)
print('Explained variance:', ret.explained_variance_)
```

Ans: Explained variance: [2.57365648e+01
1.14455187e+01 4.18511082e+00 5.35648553e-01
1.41018719e-01 1.30828465e-02]

5 Now create another PCA object passing n_components=3 and see the results: 5 marks

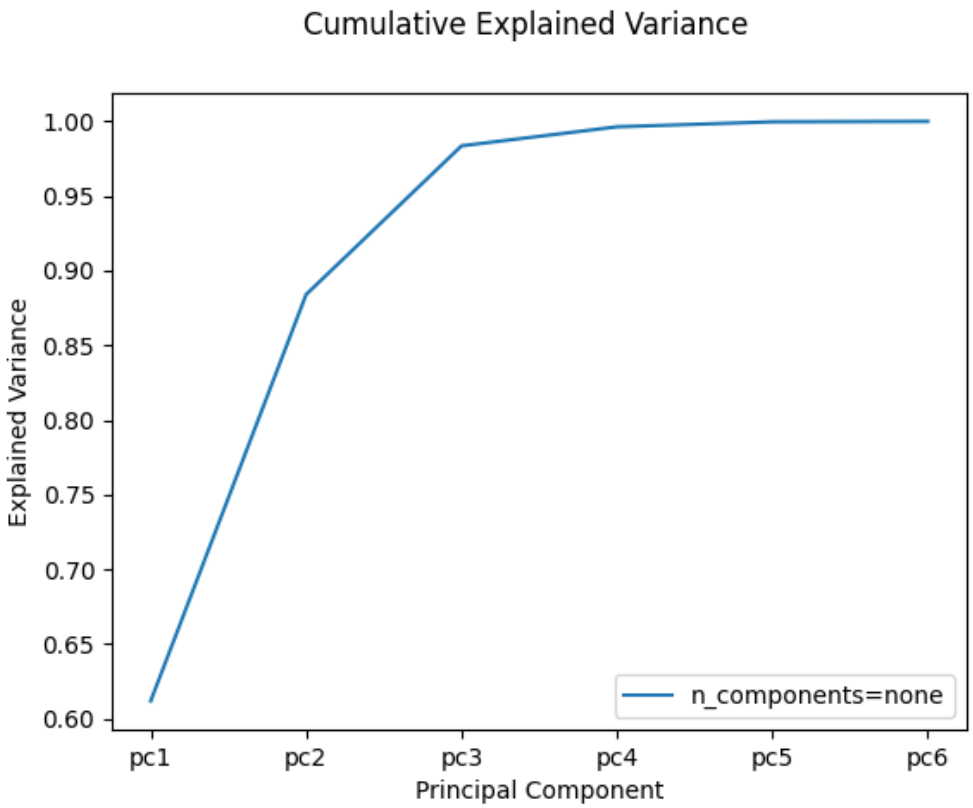
Also print the number of components and the amount of explained variance.

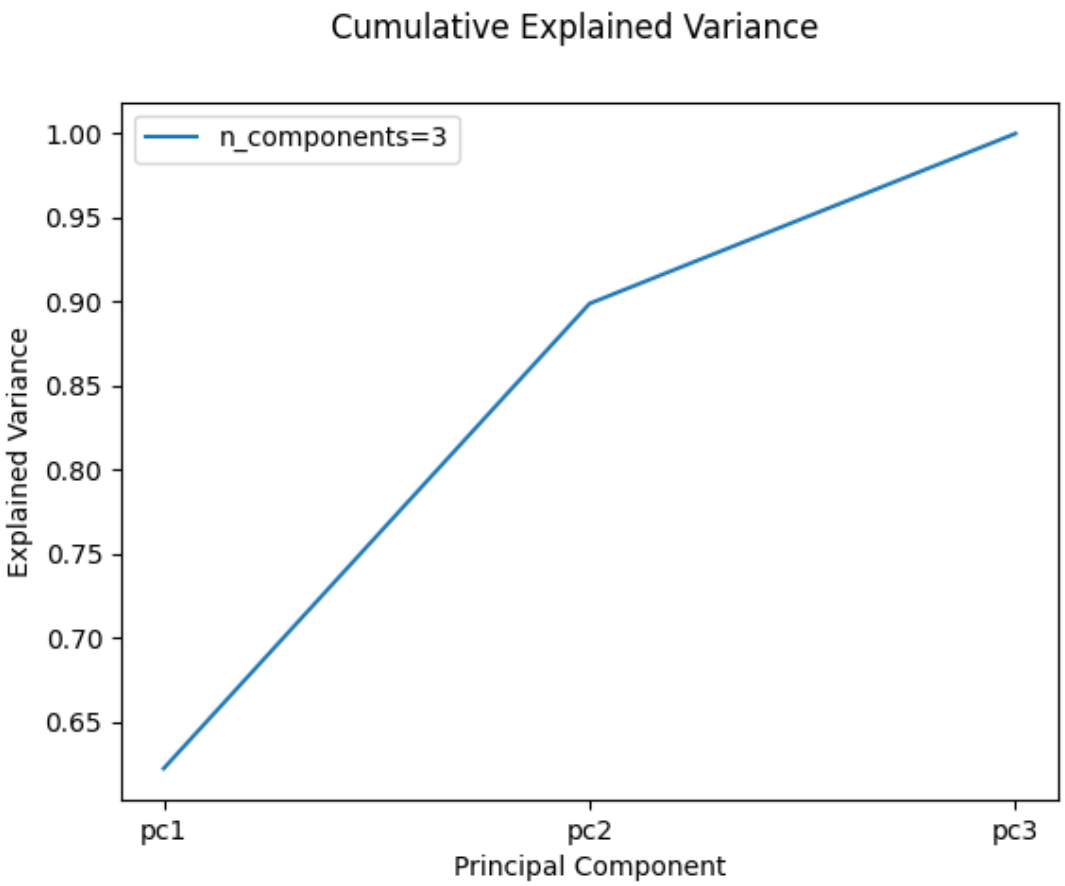
Ans: Explained variance: [25.73656481 11.4455187 4.18511082]

6 Plot the explained variance by principal components in a cumulative manner: 5 marks

Plot the cumulative explained variance as a fraction of the total.

Ans:


Cumulative Explained Variance
Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
pc1
pc2
pc3
pc4
pc5
pc6
Principal Component
n_components=none



7 Now transform the data using the 3 component PCA

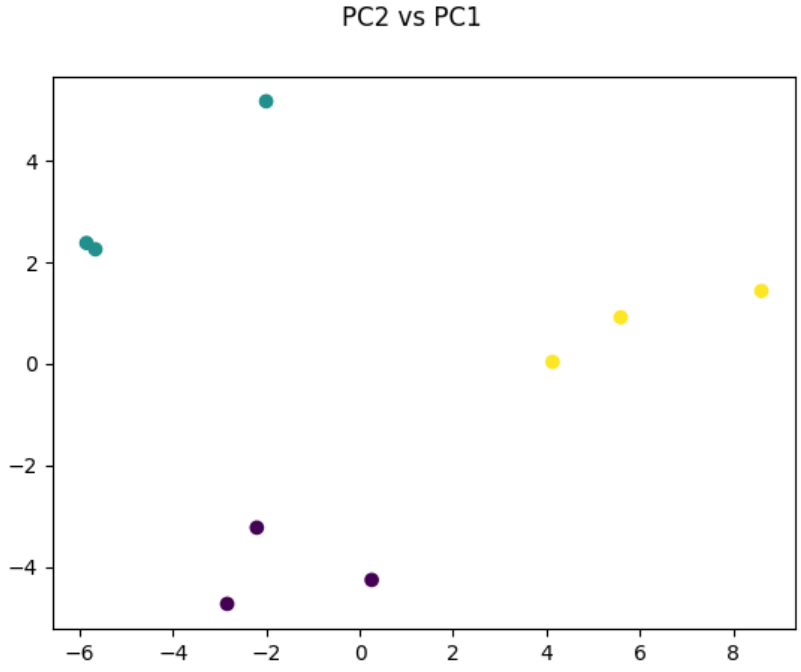
```
#%% Transform to PC axes
transformed = pca.transform(data)
print(data.shape,
transformed.shape)
fig, axes = plt.subplots(nrows=transformed.shape[0], ncols=1)
```

```
for ii in range(transformed.shape[0]): axes[ii].plot(transformed[ii])
```

8 Now plot the first two PCs: 5 marks

Create a scatter plot of the first two principal components (PC2 vs PC1), pass a list of numbers between 0 and 1 for coloring the three groups of rows (u, v, w, so you should have 3 colors, each repeated thrice). What pattern do you observe?

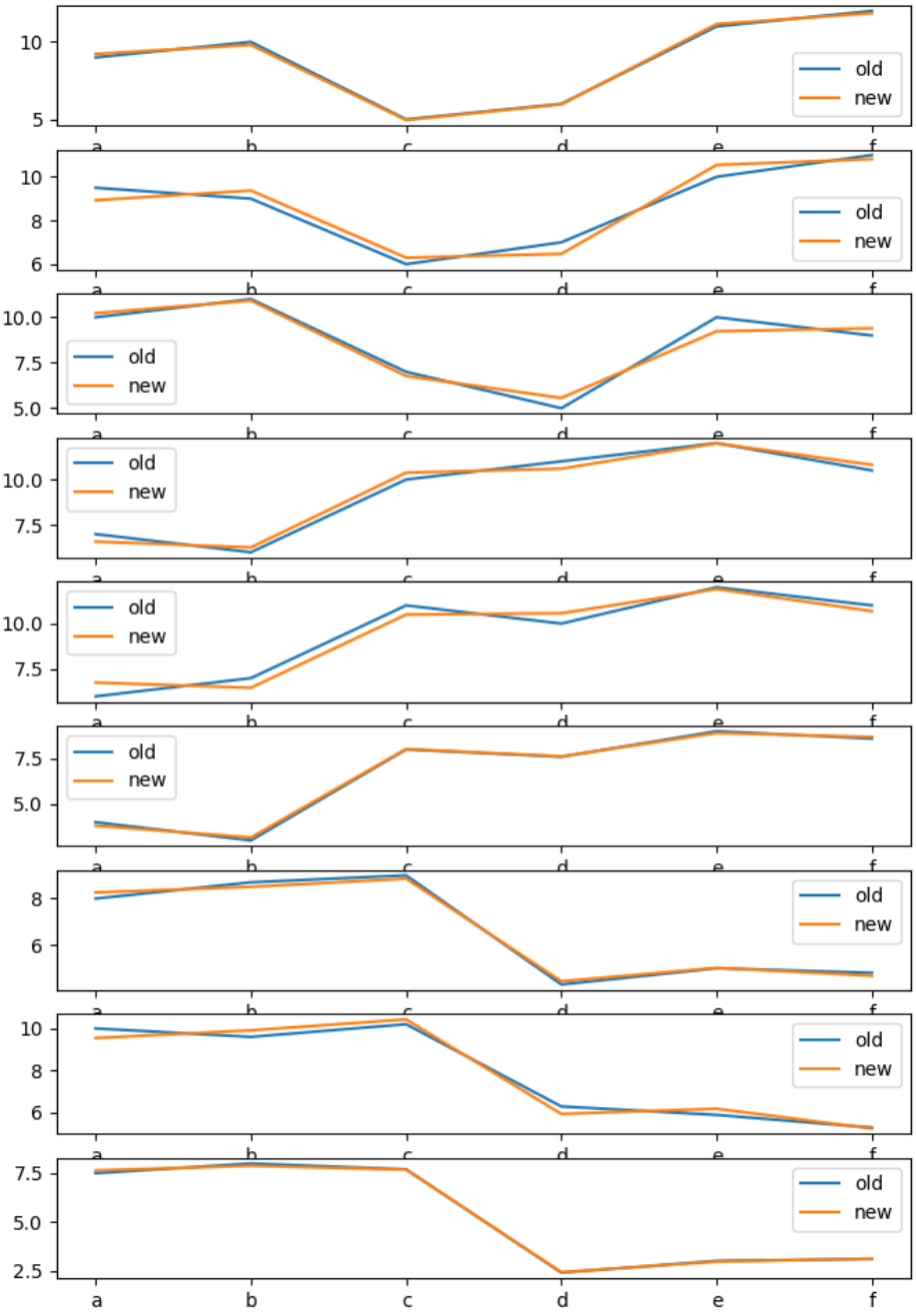
Ans: It seems like there are three clusters as should be after PCA such that similar data points can be distinguished from one another



9 Do the inverse transform on the reduced data and plot it along with the original: 5 marks

Lookup PCA.inverse_transform.

Ans:


old
new
a
b
c
d
e
f

10 Compute the errors: 5 marks

What is the squared error for each data point after you back converted from the principle components?

Ans: 0.6131112169751007

For each data point:

u1 0.136199

u2 1.163962

u3 1.176982

v1 0.646562

v2 1.546318

v3 0.080107

w1 0.162481

w2 0.568129

w3 0.037262