PCA- (digits)

December 18, 2022

[16]:
```
import pandas as pd
from sklearn.datasets import load_digits
```

[17]:
```
dataset = load_digits()
dataset.keys()
```

[17]:
```
dict_keys(['data', 'target', 'frame', 'feature_names', 'target_names', 'images',
'DESCR'])
```

[18]:
```
dataset.data[0]
```

[18]:
```
array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
       15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
       12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
        0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
       10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.])
```

[19]:
```
dataset.data[0].reshape(8,8)
```

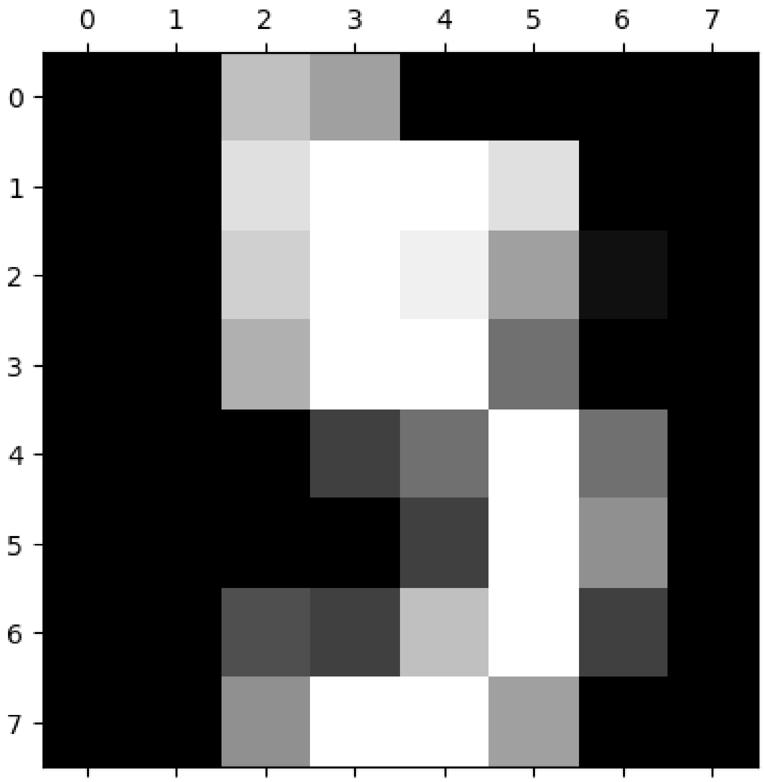
[19]:
```
array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
       [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
       [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
       [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
       [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
       [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
       [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
       [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
```

[20]:
```
from matplotlib import pyplot as plt
%matplotlib inline

plt.gray()
plt.matshow(dataset.data[5].reshape(8,8))
```

[20]:
```
<matplotlib.image.AxesImage at 0x213aefc6580>
```

```
<Figure size 640x480 with 0 Axes>
```



[21]:
```
import numpy as np
np.unique(dataset.target)
```

[21]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

[22]:
```
pd.DataFrame(dataset.data,columns =dataset.feature_names)
```

[22]:
```
      pixel_0_0  pixel_0_1  pixel_0_2  pixel_0_3  pixel_0_4  pixel_0_5  \
0           0.0        0.0        5.0       13.0        9.0        1.0
1           0.0        0.0        0.0       12.0       13.0        5.0
2           0.0        0.0        0.0        4.0       15.0       12.0
3           0.0        0.0        7.0       15.0       13.0        1.0
4           0.0        0.0        0.0        1.0       11.0        0.0
…             …          …          …          …          …          …
1792        0.0        0.0        4.0       10.0       13.0        6.0
1793        0.0        0.0        6.0       16.0       13.0       11.0
1794        0.0        0.0        1.0       11.0       15.0        1.0
1795        0.0        0.0        2.0       10.0        7.0        0.0
1796        0.0        0.0       10.0       14.0        8.0        1.0

      pixel_0_6  pixel_0_7  pixel_1_0  pixel_1_1  …  pixel_6_6  pixel_6_7  \
0           0.0        0.0        0.0        0.0  …        0.0        0.0
```

```
1           0.0        0.0        0.0        0.0  …        0.0        0.0
2           0.0        0.0        0.0        0.0  …        5.0        0.0
3           0.0        0.0        0.0        8.0  …        9.0        0.0
4           0.0        0.0        0.0        0.0  …        0.0        0.0
…         …        …        …        …  …        …        …
1792        0.0        0.0        0.0        1.0  …        4.0        0.0
1793        1.0        0.0        0.0        0.0  …        1.0        0.0
1794        0.0        0.0        0.0        0.0  …        0.0        0.0
1795        0.0        0.0        0.0        0.0  …        2.0        0.0
1796        0.0        0.0        0.0        2.0  …        8.0        0.0

      pixel_7_0  pixel_7_1  pixel_7_2  pixel_7_3  pixel_7_4  pixel_7_5  \
0           0.0        0.0        6.0       13.0       10.0        0.0
1           0.0        0.0        0.0       11.0       16.0       10.0
2           0.0        0.0        0.0        3.0       11.0       16.0
3           0.0        0.0        7.0       13.0       13.0        9.0
4           0.0        0.0        0.0        2.0       16.0        4.0
…         …        …        …        …        …        …
1792        0.0        0.0        2.0       14.0       15.0        9.0
1793        0.0        0.0        6.0       16.0       14.0        6.0
1794        0.0        0.0        2.0        9.0       13.0        6.0
1795        0.0        0.0        5.0       12.0       16.0       12.0
1796        0.0        1.0        8.0       12.0       14.0       12.0

      pixel_7_6  pixel_7_7
0           0.0        0.0
1           0.0        0.0
2           9.0        0.0
3           0.0        0.0
4           0.0        0.0
…         …        …
1792        0.0        0.0
1793        0.0        0.0
1794        0.0        0.0
1795        0.0        0.0
1796        1.0        0.0

[1797 rows x 64 columns]
```

[23]:

```
df=pd.DataFrame(dataset.data,columns =dataset.feature_names)
df.head()
```

[23]:

```
   pixel_0_0  pixel_0_1  pixel_0_2  pixel_0_3  pixel_0_4  pixel_0_5  \
0        0.0        0.0        5.0       13.0        9.0        1.0
1        0.0        0.0        0.0       12.0       13.0        5.0
2        0.0        0.0        0.0        4.0       15.0       12.0
3        0.0        0.0        7.0       15.0       13.0        1.0
```

```
4        0.0        0.0        0.0        1.0       11.0        0.0

   pixel_0_6  pixel_0_7  pixel_1_0  pixel_1_1  …  pixel_6_6  pixel_6_7  \
0        0.0        0.0        0.0        0.0  …        0.0        0.0
1        0.0        0.0        0.0        0.0  …        0.0        0.0
2        0.0        0.0        0.0        0.0  …        5.0        0.0
3        0.0        0.0        0.0        8.0  …        9.0        0.0
4        0.0        0.0        0.0        0.0  …        0.0        0.0

   pixel_7_0  pixel_7_1  pixel_7_2  pixel_7_3  pixel_7_4  pixel_7_5  \
0        0.0        0.0        6.0       13.0       10.0        0.0
1        0.0        0.0        0.0       11.0       16.0       10.0
2        0.0        0.0        0.0        3.0       11.0       16.0
3        0.0        0.0        7.0       13.0       13.0        9.0
4        0.0        0.0        0.0        2.0       16.0        4.0

   pixel_7_6  pixel_7_7
0        0.0        0.0
1        0.0        0.0
2        9.0        0.0
3        0.0        0.0
4        0.0        0.0

[5 rows x 64 columns]
```

```
[24]: df.describe()
```

```
[24]:        pixel_0_0    pixel_0_1    pixel_0_2    pixel_0_3    pixel_0_4  \
      count     1797.0  1797.000000  1797.000000  1797.000000  1797.000000
      mean         0.0     0.303840     5.204786    11.835838    11.848080
      std          0.0     0.907192     4.754826     4.248842     4.287388
      min          0.0     0.000000     0.000000     0.000000     0.000000
      25%          0.0     0.000000     1.000000    10.000000    10.000000
      50%          0.0     0.000000     4.000000    13.000000    13.000000
      75%          0.0     0.000000     9.000000    15.000000    15.000000
      max          0.0     8.000000    16.000000    16.000000    16.000000

               pixel_0_5    pixel_0_6    pixel_0_7    pixel_1_0    pixel_1_1  …  \
      count  1797.000000  1797.000000  1797.000000  1797.000000  1797.000000  …
      mean      5.781859     1.362270     0.129661     0.005565     1.993879  …
      std       5.666418     3.325775     1.037383     0.094222     3.196160  …
      min       0.000000     0.000000     0.000000     0.000000     0.000000  …
      25%       0.000000     0.000000     0.000000     0.000000     0.000000  …
      50%       4.000000     0.000000     0.000000     0.000000     0.000000  …
      75%      11.000000     0.000000     0.000000     0.000000     3.000000  …
      max      16.000000    16.000000    15.000000     2.000000    16.000000  …
```

```
         pixel_6_6    pixel_6_7    pixel_7_0    pixel_7_1    pixel_7_2  \
count  1797.000000  1797.000000  1797.000000  1797.000000  1797.000000
mean      3.725097     0.206455     0.000556     0.279354     5.557596
std       4.919406     0.984401     0.023590     0.934302     5.103019
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     1.000000
50%       1.000000     0.000000     0.000000     0.000000     4.000000
75%       7.000000     0.000000     0.000000     0.000000    10.000000
max      16.000000    13.000000     1.000000     9.000000    16.000000

         pixel_7_3    pixel_7_4    pixel_7_5    pixel_7_6    pixel_7_7
count  1797.000000  1797.000000  1797.000000  1797.000000  1797.000000
mean     12.089037    11.809126     6.764051     2.067891     0.364496
std       4.374694     4.933947     5.900623     4.090548     1.860122
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%      11.000000    10.000000     0.000000     0.000000     0.000000
50%      13.000000    14.000000     6.000000     0.000000     0.000000
75%      16.000000    16.000000    12.000000     2.000000     0.000000
max      16.000000    16.000000    16.000000    16.000000    16.000000

[8 rows x 64 columns]
```

```
[25]: x =df
```

```
[26]: y=dataset.target
```

```
[27]: y
```

```
[27]: array([0, 1, 2, …, 8, 9, 8])
```

```
[28]: from sklearn.preprocessing import StandardScaler
```

```
[29]: scaler=StandardScaler()
```

```
[30]: x_scaled = scaler.fit_transform(x)
```

```
[31]: x_scaled
```

```
[31]: array([[ 0.        , -0.33501649, -0.04308102, …, -1.14664746,
              -0.5056698 , -0.19600752],
             [ 0.        , -0.33501649, -1.09493684, …,  0.54856067,
              -0.5056698 , -0.19600752],
             [ 0.        , -0.33501649, -1.09493684, …,  1.56568555,
               1.6951369 , -0.19600752],
             …,
             [ 0.        , -0.33501649, -0.88456568, …, -0.12952258,
              -0.5056698 , -0.19600752],
             [ 0.        , -0.33501649, -0.67419451, …,  0.8876023 ,
              -0.5056698 , -0.19600752],
             [ 0.        , -0.33501649,  1.00877481, …,  0.8876023 ,
              -0.26113572, -0.19600752]])
```

[32]:
```
from sklearn.model_selection import train_test_split
```

[33]:
```
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y,test_size=0.
 ↪2,random_state=30)
```

[34]:
```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(x_train,y_train)
model.score(x_test,y_test)
```

[34]:
```
0.9722222222222222
```

[35]:
```
from sklearn.decomposition import PCA
pca =PCA(0.95)
x_pca = pca.fit_transform(x)
x_pca.shape
```

[35]:
```
(1797, 29)
```

[36]:
```
pca.explained_variance_ratio_
```

[36]:
```
array([0.14890594, 0.13618771, 0.11794594, 0.08409979, 0.05782415,
       0.0491691 , 0.04315987, 0.03661373, 0.03353248, 0.03078806,
       0.02372341, 0.02272697, 0.01821863, 0.01773855, 0.01467101,
       0.01409716, 0.01318589, 0.01248138, 0.01017718, 0.00905617,
       0.00889538, 0.00797123, 0.00767493, 0.00722904, 0.00695889,
       0.00596081, 0.00575615, 0.00515158, 0.0048954 ])
```

[37]:
```
pca.n_components_
```

[37]:
```
29
```

[38]:
```
x_train_pca,x_test_pca,y_train,y_test = train_test_split(x_pca,y,test_size=0.
 ↪2,random_state=30)
```

[39]:
```
model = LogisticRegression(max_iter=1000)
model.fit(x_train_pca,y_train)
model.score(x_test_pca,y_test)
```

[39]:
```
0.9694444444444444
```

[ ]:
```

```