pnpe2rbgq

April 17, 2023

1 Principal Component Analysis

PCA is a widely covered machine learning method on the web, and there are some great articles about it, but many spend too much time in the weeds on the topic, when most of us just want to know how it works in a simplified way.

Principal component analysis can be broken down into five steps. I'll go through each step, providing logical explanations of what PCA is doing and simplifying mathematical concepts such as standardization, covariance, eigenvectors and eigenvalues without focusing on how to compute them.

[5]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
```

[6]:
```
from sklearn.datasets import load_breast_cancer
```

[7]:
```
cancer = load_breast_cancer()
```

[8]:
```
cancer.keys()
```

[8]:
```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names',
'filename', 'data_module'])
```

[10]:
```
df = pd.DataFrame(cancer['data'],columns=cancer['feature_names'])
```

[11]:
```
df.head()
```

[11]:
```
   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0        17.99         10.38          122.80     1001.0          0.11840
1        20.57         17.77          132.90     1326.0          0.08474
2        19.69         21.25          130.00     1203.0          0.10960
3        11.42         20.38           77.58      386.1          0.14250
4        20.29         14.34          135.10     1297.0          0.10030

   mean compactness  mean concavity  mean concave points  mean symmetry  \
0           0.27760          0.3001             0.14710          0.2419
1           0.07864          0.0869             0.07017          0.1812
2           0.15990          0.1974             0.12790          0.2069
3           0.28390          0.2414             0.10520          0.2597
4           0.13280          0.1980             0.10430          0.1809

   mean fractal dimension  …  worst radius  worst texture  worst perimeter  \
0                 0.07871  …         25.38          17.33           184.60
1                 0.05667  …         24.99          23.41           158.80
2                 0.05999  …         23.57          25.53           152.50
3                 0.09744  …         14.91          26.50            98.87
4                 0.05883  …         22.54          16.67           152.20

   worst area  worst smoothness  worst compactness  worst concavity  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   worst concave points  worst symmetry  worst fractal dimension
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678

[5 rows x 30 columns]
```

[12]:
```python
from sklearn.preprocessing import StandardScaler
```

[13]:
```python
scaler = StandardScaler()
scaler.fit(df)
```

[13]:
```
StandardScaler()
```

[14]:
```python
scaled_data = scaler.transform(df)
```

[15]:
```python
from sklearn.decomposition import PCA
```

[16]:
```python
pca = PCA(n_components=2)
```

[17]:
```python
pca.fit(scaled_data)
```

[17]:
```
PCA(n_components=2)
```

[18]:
```python
x_pca = pca.transform(scaled_data)
```

```
[19]: scaled_data.shape
```

```
[19]: (569, 30)
```

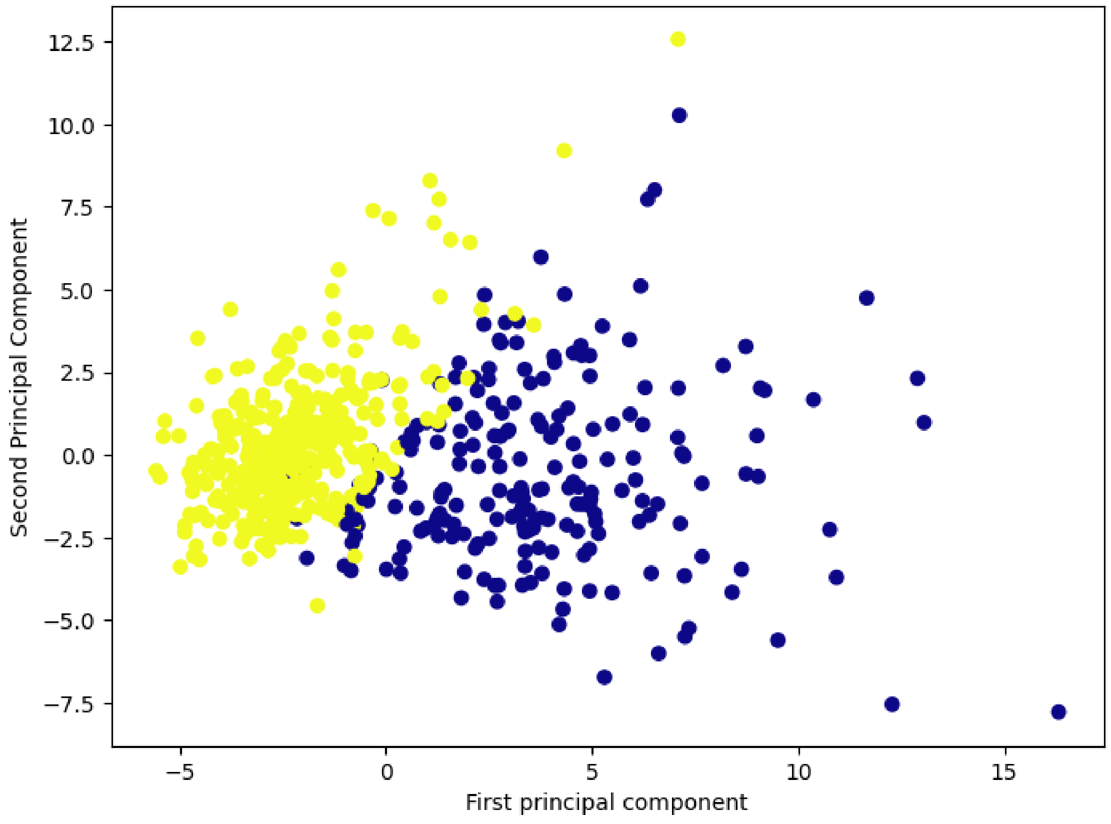
```
[ ]:
```

```
[20]: x_pca.shape
```

```
[20]: (569, 2)
```

```
[21]: plt.figure(figsize=(8,6))
      plt.scatter(x_pca[:,0],x_pca[:,1],c=cancer['target'],cmap='plasma')
      plt.xlabel('First principal component')
      plt.ylabel('Second Principal Component')
```

```
[21]: Text(0, 0.5, 'Second Principal Component')
```



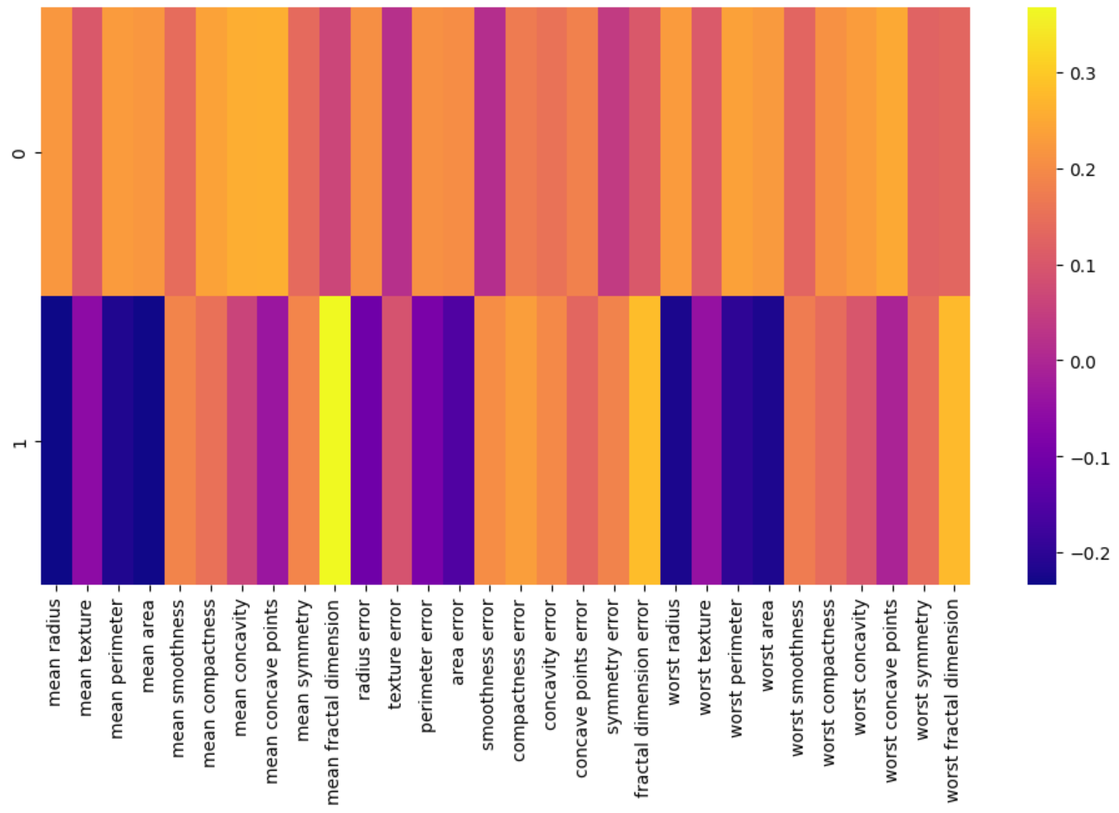
```
[22]: pca.components_
```

```
[22]: array([[ 0.21890244,  0.10372458,  0.22753729,  0.22099499,  0.14258969,
               0.23928535,  0.25840048,  0.26085376,  0.13816696,  0.06436335,
         0.20597878,  0.01742803,  0.21132592,  0.20286964,  0.01453145,
         0.17039345,  0.15358979,  0.1834174 ,  0.04249842,  0.10256832,
         0.22799663,  0.10446933,  0.23663968,  0.22487053,  0.12795256,
         0.21009588,  0.22876753,  0.25088597,  0.12290456,  0.13178394],
       [-0.23385713, -0.05970609, -0.21518136, -0.23107671,  0.18611302,
         0.15189161,  0.06016536, -0.0347675 ,  0.19034877,  0.36657547,
        -0.10555215,  0.08997968, -0.08945723, -0.15229263,  0.20443045,
         0.2327159 ,  0.19720728,  0.13032156,  0.183848  ,  0.28009203,
        -0.21986638, -0.0454673 , -0.19987843, -0.21935186,  0.17230435,
         0.14359317,  0.09796411, -0.00825724,  0.14188335,  0.27533947]])
```

[23]:
```python
df_comp = pd.DataFrame(pca.components_,columns=cancer['feature_names'])
```

[24]:
```python
plt.figure(figsize=(12,6))
sns.heatmap(df_comp,cmap='plasma',)
```

[24]:
```
<Axes: >
```


mean radius
mean texture
mean perimeter
mean area
mean smoothness
mean compactness
mean concavity
mean concave points
mean symmetry
mean fractal dimension
radius error
texture error
perimeter error
area error
smoothness error
compactness error
concavity error
concave points error
symmetry error
fractal dimension error
worst radius
worst texture
worst perimeter
worst area
worst smoothness
worst compactness
worst concavity
worst concave points
worst symmetry
worst fractal dimension