

```
In [1]: import matplotlib
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [3]: from sklearn.datasets import load_breast_cancer
```

```
In [9]: cancer = load_breast_cancer()
```

```
In [13]: type(cancer)
```

```
Out[13]: sklearn.utils._bunch.Bunch
```

```
In [15]: cancer.keys()
```

```
Out[15]: dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names',
'filename', 'data_module'])
```

```
In [17]: print(cancer['DESCR'])
```

```
.. _breast_cancer_dataset:
```

```
Breast cancer wisconsin (diagnostic) dataset
```

```
-----
```

```
**Data Set Characteristics:**
```

```
:Number of Instances: 569
```

```
:Number of Attributes: 30 numeric, predictive attributes and the class
```

```
:Attribute Information:
```

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter² / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

The mean, standard error, and "worst" or largest (mean of the three worst/largest values) of these features were computed for each image, resulting in 30 features. For instance, field 0 is Mean Radius, field 10 is Radius SE, field 20 is Worst Radius.

- class:
 - WDBC-Malignant
 - WDBC-Benign

```
:Summary Statistics:
```

```
=====
                                     Min    Max
=====
radius (mean):                      6.981  28.11
texture (mean):                      9.71   39.28
perimeter (mean):                    43.79  188.5
area (mean):                         143.5 2501.0
smoothness (mean):                   0.053  0.163
compactness (mean):                   0.019  0.345
concavity (mean):                     0.0    0.427
concave points (mean):                0.0    0.201
symmetry (mean):                      0.106  0.304
fractal dimension (mean):              0.05   0.097
radius (standard error):              0.112  2.873
texture (standard error):              0.36   4.885
perimeter (standard error):            0.757  21.98
area (standard error):                 6.802  542.2
smoothness (standard error):           0.002  0.031
compactness (standard error):           0.002  0.135
concavity (standard error):            0.0    0.396
concave points (standard error):        0.0    0.053
symmetry (standard error):             0.008  0.079
fractal dimension (standard error):     0.001  0.03
radius (worst):                       7.93   36.04
texture (worst):                      12.02  49.54
perimeter (worst):                     50.41 251.2
```

area (worst):	185.2	4254.0
smoothness (worst):	0.071	0.223
compactness (worst):	0.027	1.058
concavity (worst):	0.0	1.252
concave points (worst):	0.0	0.291
symmetry (worst):	0.156	0.664
fractal dimension (worst):	0.055	0.208
=====	=====	=====

:Missing Attribute Values: None

:Class Distribution: 212 - Malignant, 357 - Benign

:Creator: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian

:Donor: Nick Street

:Date: November, 1995

This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.
<https://goo.gl/U2Uwz2>

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

The actual linear program used to obtain the separating plane in the 3-dimensional space is that described in:
 [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
 cd math-prog/cpo-dataset/machine-learn/WDBC/

|details-start|
 References
 |details-split|

- W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.
- O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.
- W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) 163-171.

|details-end|

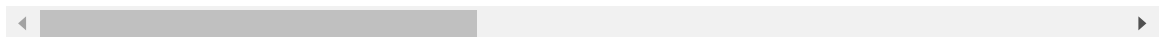
```
In [19]: df = pd.DataFrame(cancer['data'], columns=cancer['feature_names'])
```

```
In [21]: df.head()
```

```
Out[21]:
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	s
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	

5 rows × 30 columns



```
In [23]: from sklearn.preprocessing import StandardScaler
```

```
In [25]: scaler = StandardScaler()
```

```
In [27]: scaler.fit(df)
```

```
Out[27]: StandardScaler ⓘ ?
StandardScaler()
```

```
In [29]: scaled_data = scaler.transform(df)
```

```
In [33]: #PCA
from sklearn.decomposition import PCA
```

```
In [37]: pca = PCA(n_components=2)
```

```
In [39]: pca.fit(scaled_data)
```

```
Out[39]: PCA ⓘ ?
PCA(n_components=2)
```

```
In [41]: x_pca = pca.transform(scaled_data)
```

```
In [43]: scaled_data.shape
```

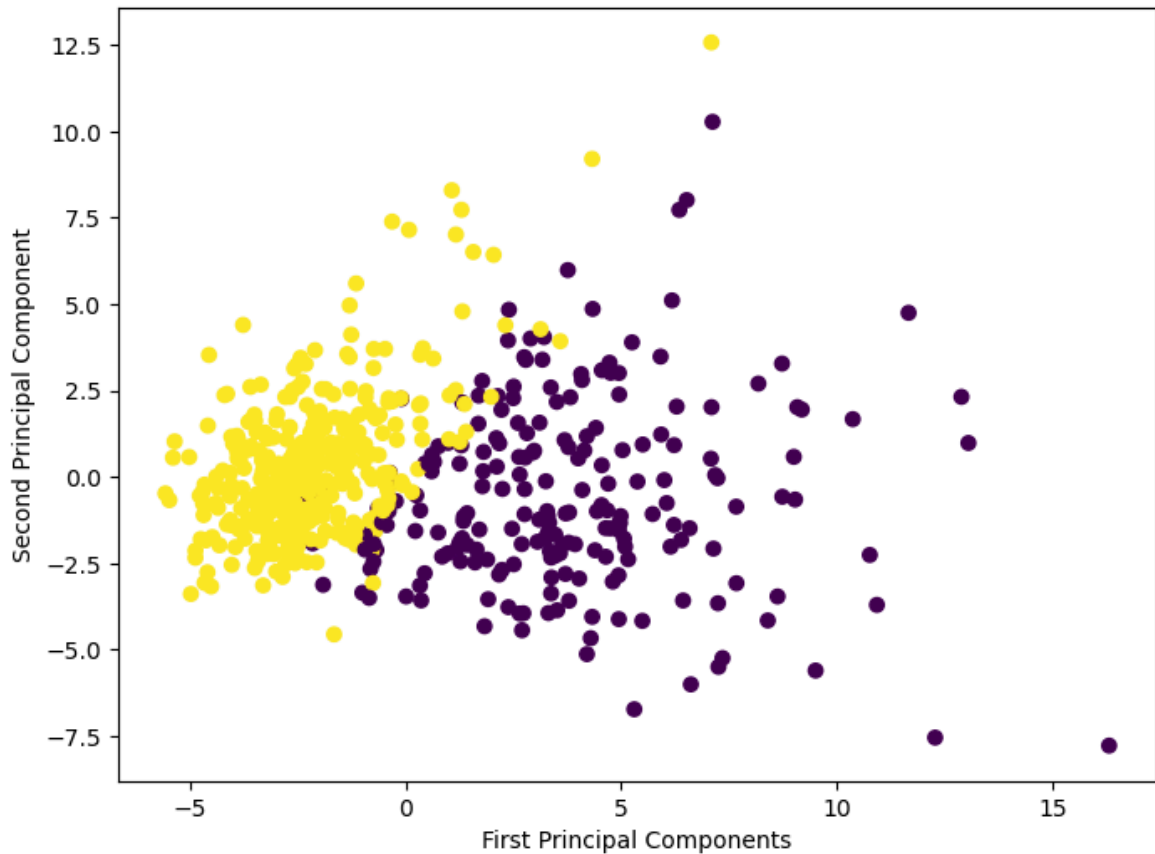
```
Out[43]: (569, 30)
```

```
In [45]: x_pca.shape
```

```
Out[45]: (569, 2)
```

```
In [53]: plt.figure(figsize=(8,6))
plt.scatter(x_pca[:,0], x_pca[:,1], c=cancer['target'])
plt.xlabel('First Principal Components')
plt.ylabel('Second Principal Component')
```

```
Out[53]: Text(0, 0.5, 'Second Principal Component')
```



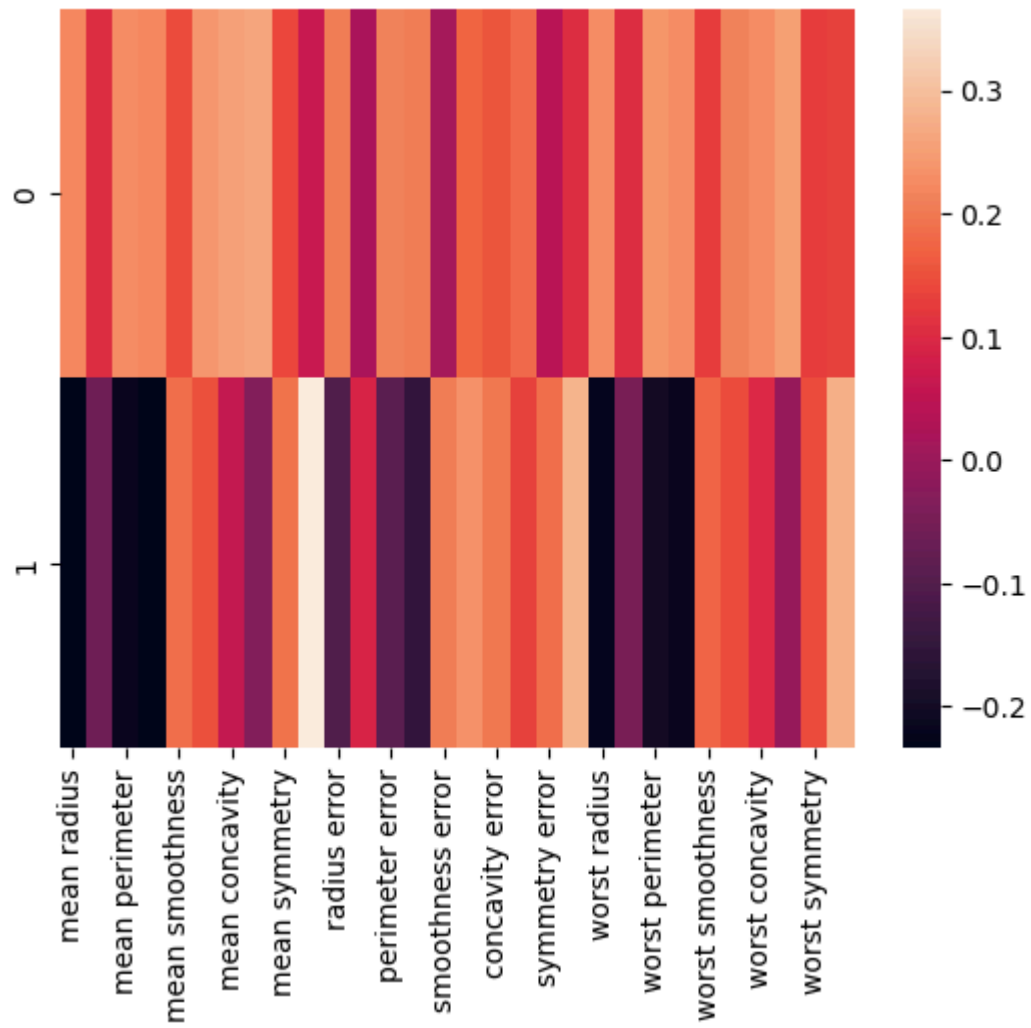
```
In [55]: pca.components_
```

```
Out[55]: 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]])
```

```
In [57]: df_comp = pd.DataFrame(pca.components_, columns=cancer['feature_names'])
```

```
In [61]: sns.heatmap(df_comp)
```

```
Out[61]: <Axes: >
```



In []: