In [1]:

**import** **numpy** **as** **np**  
**import** **pandas** **as** **pd**  
**import** **matplotlib.pyplot** **as** **plt**

In [2]:

**from** **sklearn.datasets** **import** load\_breast\_cancer

In [3]:

cancer\_data = load\_breast\_cancer()

In [4]:

cancer\_data.keys()

Out[4]:

dict\_keys(['data', 'target', 'target\_names', 'DESCR', 'feature\_names', 'filename'])

In [5]:

print(cancer\_data['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 (perimeter^2 / 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  
 largest values) of these features were computed for each image,  
 resulting in 30 features. For instance, field 3 is Mean Radius, field  
 13 is Radius SE, field 23 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/  
  
.. topic:: References  
  
 - 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.

In [6]:

df=pd.DataFrame(cancer\_data['data'],columns=cancer\_data['feature\_names'])

In [7]:

df.head()

Out[7]:

|  | **mean radius** | **mean texture** | **mean perimeter** | **mean area** | **mean smoothness** | **mean compactness** | **mean concavity** | **mean concave points** | **mean symmetry** | **mean fractal dimension** | **...** | **worst radius** | **worst texture** | **worst perimeter** | **worst area** | **worst smoothness** | **worst compactness** | **worst concavity** | **worst concave points** | **worst symmetry** | **worst fractal dimension** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0.27760 | 0.3001 | 0.14710 | 0.2419 | 0.07871 | ... | 25.38 | 17.33 | 184.60 | 2019.0 | 0.1622 | 0.6656 | 0.7119 | 0.2654 | 0.4601 | 0.11890 |
| **1** | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0.07864 | 0.0869 | 0.07017 | 0.1812 | 0.05667 | ... | 24.99 | 23.41 | 158.80 | 1956.0 | 0.1238 | 0.1866 | 0.2416 | 0.1860 | 0.2750 | 0.08902 |
| **2** | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0.15990 | 0.1974 | 0.12790 | 0.2069 | 0.05999 | ... | 23.57 | 25.53 | 152.50 | 1709.0 | 0.1444 | 0.4245 | 0.4504 | 0.2430 | 0.3613 | 0.08758 |
| **3** | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0.28390 | 0.2414 | 0.10520 | 0.2597 | 0.09744 | ... | 14.91 | 26.50 | 98.87 | 567.7 | 0.2098 | 0.8663 | 0.6869 | 0.2575 | 0.6638 | 0.17300 |
| **4** | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0.13280 | 0.1980 | 0.10430 | 0.1809 | 0.05883 | ... | 22.54 | 16.67 | 152.20 | 1575.0 | 0.1374 | 0.2050 | 0.4000 | 0.1625 | 0.2364 | 0.07678 |

5 rows × 30 columns

In [8]:

*#df.isnull().sum()*

**MinMax Scaler**[**¶**](#gjdgxs)

In [9]:

**from** **sklearn.preprocessing** **import** MinMaxScaler

In [10]:

min\_max\_scaler = MinMaxScaler()

In [11]:

min\_max\_scaler.fit(df)

Out[11]:

MinMaxScaler(copy=True, feature\_range=(0, 1))

In [12]:

minmax\_scaled\_data=min\_max\_scaler.transform(df)

In [13]:

minmax\_scaled\_data[:1,:]

Out[13]:

array([[0.52103744, 0.0226581 , 0.54598853, 0.36373277, 0.59375282,  
 0.7920373 , 0.70313964, 0.73111332, 0.68636364, 0.60551811,  
 0.35614702, 0.12046941, 0.3690336 , 0.27381126, 0.15929565,  
 0.35139844, 0.13568182, 0.30062512, 0.31164518, 0.18304244,  
 0.62077552, 0.14152452, 0.66831017, 0.45069799, 0.60113584,  
 0.61929156, 0.56861022, 0.91202749, 0.59846245, 0.41886396]])

**Standard Normalization**[**¶**](#30j0zll)

In [14]:

**from** **sklearn.preprocessing** **import** StandardScaler

In [15]:

scaler = StandardScaler()

In [16]:

scaler.fit(df)

Out[16]:

StandardScaler(copy=True, with\_mean=True, with\_std=True)

In [17]:

df.head()

Out[17]:

|  | **mean radius** | **mean texture** | **mean perimeter** | **mean area** | **mean smoothness** | **mean compactness** | **mean concavity** | **mean concave points** | **mean symmetry** | **mean fractal dimension** | **...** | **worst radius** | **worst texture** | **worst perimeter** | **worst area** | **worst smoothness** | **worst compactness** | **worst concavity** | **worst concave points** | **worst symmetry** | **worst fractal dimension** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0.27760 | 0.3001 | 0.14710 | 0.2419 | 0.07871 | ... | 25.38 | 17.33 | 184.60 | 2019.0 | 0.1622 | 0.6656 | 0.7119 | 0.2654 | 0.4601 | 0.11890 |
| **1** | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0.07864 | 0.0869 | 0.07017 | 0.1812 | 0.05667 | ... | 24.99 | 23.41 | 158.80 | 1956.0 | 0.1238 | 0.1866 | 0.2416 | 0.1860 | 0.2750 | 0.08902 |
| **2** | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0.15990 | 0.1974 | 0.12790 | 0.2069 | 0.05999 | ... | 23.57 | 25.53 | 152.50 | 1709.0 | 0.1444 | 0.4245 | 0.4504 | 0.2430 | 0.3613 | 0.08758 |
| **3** | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0.28390 | 0.2414 | 0.10520 | 0.2597 | 0.09744 | ... | 14.91 | 26.50 | 98.87 | 567.7 | 0.2098 | 0.8663 | 0.6869 | 0.2575 | 0.6638 | 0.17300 |
| **4** | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0.13280 | 0.1980 | 0.10430 | 0.1809 | 0.05883 | ... | 22.54 | 16.67 | 152.20 | 1575.0 | 0.1374 | 0.2050 | 0.4000 | 0.1625 | 0.2364 | 0.07678 |

5 rows × 30 columns

In [18]:

scaled\_data=scaler.transform(df)

In [19]:

scaled\_data[:1,:]

Out[19]:

array([[ 1.09706398, -2.07333501, 1.26993369, 0.9843749 , 1.56846633,  
 3.28351467, 2.65287398, 2.53247522, 2.21751501, 2.25574689,  
 2.48973393, -0.56526506, 2.83303087, 2.48757756, -0.21400165,  
 1.31686157, 0.72402616, 0.66081994, 1.14875667, 0.90708308,  
 1.88668963, -1.35929347, 2.30360062, 2.00123749, 1.30768627,  
 2.61666502, 2.10952635, 2.29607613, 2.75062224, 1.93701461]])

In [20]:

**from** **sklearn.decomposition** **import** PCA

In [21]:

pca=PCA(n\_components = 2)

In [22]:

pca= pca.fit(scaled\_data)

In [23]:

x\_pca=pca.transform(scaled\_data)

In [24]:

scaled\_data.shape

Out[24]:

(569, 30)

In [25]:

x\_pca.shape

Out[25]:

(569, 2)

In [ ]:

In [26]:

pca2=PCA(n\_components = 2)  
pca2= pca2.fit(minmax\_scaled\_data)  
x\_pca2=pca2.transform(minmax\_scaled\_data)

In [27]:

minmax\_scaled\_data.shape

Out[27]:

(569, 30)

In [28]:

x\_pca2.shape

Out[28]:

(569, 2)

In [ ]: