PCA and scaling

October 21, 2022

[40]:
```
import pandas as pd
import numpy as np
```

[41]:
```
from sklearn.datasets import load_breast_cancer
```

[42]:
```
cancer_data= load_breast_cancer()
```

[43]:
```
print(cancer_data)
```

```
{'data': array([[1.799e+01, 1.038e+01, 1.228e+02, …, 2.654e-01, 4.601e-01,
        1.189e-01],
       [2.057e+01, 1.777e+01, 1.329e+02, …, 1.860e-01, 2.750e-01,
        8.902e-02],
       [1.969e+01, 2.125e+01, 1.300e+02, …, 2.430e-01, 3.613e-01,
        8.758e-02],
       …,
       [1.660e+01, 2.808e+01, 1.083e+02, …, 1.418e-01, 2.218e-01,
        7.820e-02],
       [2.060e+01, 2.933e+01, 1.401e+02, …, 2.650e-01, 4.087e-01,
        1.240e-01],
       [7.760e+00, 2.454e+01, 4.792e+01, …, 0.000e+00, 2.871e-01,
        7.039e-02]]), 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0,
       0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
```

```
       0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1,
       1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1]), 'frame': None,
'target_names': array(['malignant', 'benign'], dtype='<U9'), 'DESCR': '..
_breast_cancer_dataset:\n\nBreast cancer wisconsin (diagnostic)
dataset\n--------------------------------------------\n\n**Data Set
Characteristics:**\n\n    :Number of Instances: 569\n\n    :Number of
Attributes: 30 numeric, predictive attributes and the class\n\n    :Attribute
Information:\n        - radius (mean of distances from center to points on the
perimeter)\n        - texture (standard deviation of gray-scale values)\n
- perimeter\n        - area\n        - smoothness (local variation in radius
lengths)\n        - compactness (perimeter^2 / area - 1.0)\n        - concavity
(severity of concave portions of the contour)\n        - concave points (number
of concave portions of the contour)\n        - symmetry\n        - fractal
dimension ("coastline approximation" - 1)\n\n        The mean, standard error,
and "worst" or largest (mean of the three\n        worst/largest values) of
these features were computed for each image,\n        resulting in 30 features.
For instance, field 0 is Mean Radius, field\n        10 is Radius SE, field 20
is Worst Radius.\n\n        - class:\n                - WDBC-Malignant\n
- WDBC-Benign\n\n    :Summary Statistics:\n\n
===================================== ====== ======\n
Min    Max\n    ===================================== ====== ======\n    radius
(mean):                        6.981  28.11\n    texture (mean):
9.71   39.28\n    perimeter (mean):                       43.79  188.5\n    area
(mean):                          143.5  2501.0\n    smoothness (mean):
0.053  0.163\n    compactness (mean):                     0.019  0.345\n
concavity (mean):                     0.0    0.427\n    concave points (mean):
0.0    0.201\n    symmetry (mean):                       0.106  0.304\n
fractal dimension (mean):             0.05   0.097\n    radius (standard error):
0.112  2.873\n    texture (standard error):              0.36   4.885\n
perimeter (standard error):           0.757  21.98\n    area (standard error):
6.802  542.2\n    smoothness (standard error):           0.002  0.031\n
compactness (standard error):         0.002  0.135\n    concavity (standard
error):           0.0    0.396\n    concave points (standard error):      0.0
0.053\n    symmetry (standard error):            0.008  0.079\n    fractal
dimension (standard error):   0.001  0.03\n    radius (worst):
7.93   36.04\n    texture (worst):                       12.02  49.54\n
perimeter (worst):                    50.41  251.2\n    area (worst):
185.2  4254.0\n    smoothness (worst):                    0.071  0.223\n
compactness (worst):                  0.027  1.058\n    concavity (worst):
0.0    1.252\n    concave points (worst):                0.0    0.291\n
```

```
symmetry (worst):                       0.156  0.664\n    fractal dimension
(worst):            0.055  0.208\n    ====================================
====== ======\n\n    :Missing Attribute Values: None\n\n    :Class Distribution:
212 - Malignant, 357 - Benign\n\n    :Creator:  Dr. William H. Wolberg, W. Nick
Street, Olvi L. Mangasarian\n\n    :Donor: Nick Street\n\n    :Date: November,
1995\n\nThis is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic)
datasets.\nhttps://goo.gl/U2Uwz2\n\nFeatures are computed from a digitized image
of a fine needle\naspirate (FNA) of a breast mass.  They
describe\ncharacteristics of the cell nuclei present in the image.\n\nSeparating
plane described above was obtained using\nMultisurface Method-Tree (MSM-T) [K.
P. Bennett, "Decision Tree\nConstruction Via Linear Programming." Proceedings of
the 4th\nMidwest Artificial Intelligence and Cognitive Science Society,\npp.
97-101, 1992], a classification method which uses linear\nprogramming to
construct a decision tree.  Relevant features\nwere selected using an exhaustive
search in the space of 1-4\nfeatures and 1-3 separating planes.\n\nThe actual
linear program used to obtain the separating plane\nin the 3-dimensional space
is that described in:\n[K. P. Bennett and O. L. Mangasarian: "Robust
Linear\nProgramming Discrimination of Two Linearly Inseparable
Sets",\nOptimization Methods and Software 1, 1992, 23-34].\n\nThis database is
also available through the UW CS ftp server:\n\nftp ftp.cs.wisc.edu\ncd math-
prog/cpo-dataset/machine-learn/WDBC/\n\n.. topic:: References\n\n   - W.N.
Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction \n     for
breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on \n
Electronic Imaging: Science and Technology, volume 1905, pages 861-870,\n
San Jose, CA, 1993.\n   - O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast
cancer diagnosis and \n     prognosis via linear programming. Operations
Research, 43(4), pages 570-577, \n     July-August 1995.\n   - W.H. Wolberg,
W.N. Street, and O.L. Mangasarian. Machine learning techniques\n     to diagnose
breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) \n
163-171.', 'feature_names': array(['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'], dtype='<U23'), 'filename':
'breast_cancer.csv', 'data_module': 'sklearn.datasets.data'}
```

[44]:
```
data= pd.DataFrame(cancer_data.data, columns= cancer_data.feature_names)
```

[45]:
```
data.head()
```

[45]:
```
   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]
```

[46]:
```
data.describe()
```

[46]:
```
       mean radius  mean texture  mean perimeter    mean area  \
count   569.000000    569.000000      569.000000   569.000000
mean     14.127292     19.289649       91.969033   654.889104
std       3.524049      4.301036       24.298981   351.914129
min       6.981000      9.710000       43.790000   143.500000
25%      11.700000     16.170000       75.170000   420.300000
50%      13.370000     18.840000       86.240000   551.100000
75%      15.780000     21.800000      104.100000   782.700000
```

```
max      28.110000     39.280000      188.500000  2501.000000

       mean smoothness  mean compactness  mean concavity  mean concave points  \
count       569.000000        569.000000      569.000000           569.000000
mean          0.096360          0.104341        0.088799             0.048919
std           0.014064          0.052813        0.079720             0.038803
min           0.052630          0.019380        0.000000             0.000000
25%           0.086370          0.064920        0.029560             0.020310
50%           0.095870          0.092630        0.061540             0.033500
75%           0.105300          0.130400        0.130700             0.074000
max           0.163400          0.345400        0.426800             0.201200

       mean symmetry  mean fractal dimension  …  worst radius  \
count     569.000000              569.000000  …    569.000000
mean        0.181162                0.062798  …     16.269190
std         0.027414                0.007060  …      4.833242
min         0.106000                0.049960  …      7.930000
25%         0.161900                0.057700  …     13.010000
50%         0.179200                0.061540  …     14.970000
75%         0.195700                0.066120  …     18.790000
max         0.304000                0.097440  …     36.040000

       worst texture  worst perimeter   worst area  worst smoothness  \
count     569.000000       569.000000   569.000000        569.000000
mean       25.677223       107.261213   880.583128          0.132369
std         6.146258        33.602542   569.356993          0.022832
min        12.020000        50.410000   185.200000          0.071170
25%        21.080000        84.110000   515.300000          0.116600
50%        25.410000        97.660000   686.500000          0.131300
75%        29.720000       125.400000  1084.000000          0.146000
max        49.540000       251.200000  4254.000000          0.222600

       worst compactness  worst concavity  worst concave points  \
count         569.000000       569.000000            569.000000
mean            0.254265         0.272188              0.114606
std             0.157336         0.208624              0.065732
min             0.027290         0.000000              0.000000
25%             0.147200         0.114500              0.064930
50%             0.211900         0.226700              0.099930
75%             0.339100         0.382900              0.161400
max             1.058000         1.252000              0.291000

       worst symmetry  worst fractal dimension
count      569.000000               569.000000
mean         0.290076                 0.083946
std          0.061867                 0.018061
min          0.156500                 0.055040
```

```
25%          0.250400                 0.071460
50%          0.282200                 0.080040
75%          0.317900                 0.092080
max          0.663800                 0.207500

[8 rows x 30 columns]
```

[47]:
```
target= pd.DataFrame( cancer_data.target, columns=['target'])
```

[48]:
```
from sklearn.preprocessing import StandardScaler
```

[49]:
```
scaler= StandardScaler()
scaler.fit(data[['mean area']])
```

[49]:
```
StandardScaler()
```

[50]:
```
#data['mean area']= scaler.fit(data[['mean area']])
```

[51]:
```
data['mean area'] = scaler.transform(data[['mean area']])
```

[52]:
```
data.describe()
```

[52]:
```
       mean radius  mean texture  mean perimeter     mean area  \
count   569.000000    569.000000      569.000000  5.690000e+02
mean     14.127292     19.289649       91.969033 -1.900452e-16
std       3.524049      4.301036       24.298981  1.000880e+00
min       6.981000      9.710000       43.790000 -1.454443e+00
25%      11.700000     16.170000       75.170000 -6.671955e-01
50%      13.370000     18.840000       86.240000 -2.951869e-01
75%      15.780000     21.800000      104.100000  3.635073e-01
max      28.110000     39.280000      188.500000  5.250529e+00

       mean smoothness  mean compactness  mean concavity  mean concave points  \
count       569.000000        569.000000      569.000000           569.000000
mean          0.096360          0.104341        0.088799             0.048919
std           0.014064          0.052813        0.079720             0.038803
min           0.052630          0.019380        0.000000             0.000000
25%           0.086370          0.064920        0.029560             0.020310
50%           0.095870          0.092630        0.061540             0.033500
75%           0.105300          0.130400        0.130700             0.074000
max           0.163400          0.345400        0.426800             0.201200

       mean symmetry  mean fractal dimension  …  worst radius  \
count     569.000000              569.000000  …    569.000000
mean        0.181162                0.062798  …     16.269190
std         0.027414                0.007060  …      4.833242
min         0.106000                0.049960  …      7.930000
```

```
25%         0.161900                0.057700  …     13.010000
50%         0.179200                0.061540  …     14.970000
75%         0.195700                0.066120  …     18.790000
max         0.304000                0.097440  …     36.040000

       worst texture  worst perimeter   worst area  worst smoothness  \
count     569.000000       569.000000   569.000000        569.000000
mean       25.677223       107.261213   880.583128          0.132369
std         6.146258        33.602542   569.356993          0.022832
min        12.020000        50.410000   185.200000          0.071170
25%        21.080000        84.110000   515.300000          0.116600
50%        25.410000        97.660000   686.500000          0.131300
75%        29.720000       125.400000  1084.000000          0.146000
max        49.540000       251.200000  4254.000000          0.222600

       worst compactness  worst concavity  worst concave points  \
count         569.000000       569.000000            569.000000
mean            0.254265         0.272188              0.114606
std             0.157336         0.208624              0.065732
min             0.027290         0.000000              0.000000
25%             0.147200         0.114500              0.064930
50%             0.211900         0.226700              0.099930
75%             0.339100         0.382900              0.161400
max             1.058000         1.252000              0.291000

       worst symmetry  worst fractal dimension
count      569.000000               569.000000
mean         0.290076                 0.083946
std          0.061867                 0.018061
min          0.156500                 0.055040
25%          0.250400                 0.071460
50%          0.282200                 0.080040
75%          0.317900                 0.092080
max          0.663800                 0.207500

[8 rows x 30 columns]
```

[53]:
```
scaler= StandardScaler()
scaler.fit(data[['mean perimeter']])
```

[53]:
```
StandardScaler()
```

[54]:
```
data['mean perimeter'] = scaler.transform(data[['mean perimeter']])
```

[58]:
```
scaler= StandardScaler()
scaler.fit(data[['worst radius']])
data['worst radius'] = scaler.transform(data[['worst radius']])
```

[59]:
```
scaler= StandardScaler()
scaler.fit(data[['worst texture']])
data['worst texture'] = scaler.transform(data[['worst texture']])
```

[60]:
```
scaler= StandardScaler()
scaler.fit(data[['worst perimeter']])
data['worst perimeter'] = scaler.transform(data[['worst perimeter']])
```

[61]:
```
scaler= StandardScaler()
scaler.fit(data[['worst area']])
data['worst area'] = scaler.transform(data[['worst area']])
```

[62]:
```
data.describe()
```

[62]:

	mean radius	mean texture	mean perimeter	mean area	\
count	569.000000	569.000000	5.690000e+02	5.690000e+02	
mean	14.127292	19.289649	-1.272171e-16	-1.900452e-16	
std	3.524049	4.301036	1.000880e+00	1.000880e+00	
min	6.981000	9.710000	-1.984504e+00	-1.454443e+00	
25%	11.700000	16.170000	-6.919555e-01	-6.671955e-01	
50%	13.370000	18.840000	-2.359800e-01	-2.951869e-01	
75%	15.780000	21.800000	4.996769e-01	3.635073e-01	
max	28.110000	39.280000	3.976130e+00	5.250529e+00	

	mean smoothness	mean compactness	mean concavity	mean concave points	\
count	569.000000	569.000000	569.000000	569.000000	
mean	0.096360	0.104341	0.088799	0.048919	
std	0.014064	0.052813	0.079720	0.038803	
min	0.052630	0.019380	0.000000	0.000000	
25%	0.086370	0.064920	0.029560	0.020310	
50%	0.095870	0.092630	0.061540	0.033500	
75%	0.105300	0.130400	0.130700	0.074000	
max	0.163400	0.345400	0.426800	0.201200	

	mean symmetry	mean fractal dimension	…	worst radius	\
count	569.000000	569.000000	…	5.690000e+02	
mean	0.181162	0.062798	…	1.232757e-15	
std	0.027414	0.007060	…	1.000880e+00	
min	0.106000	0.049960	…	-1.726901e+00	
25%	0.161900	0.057700	…	-6.749213e-01	
50%	0.179200	0.061540	…	-2.690395e-01	
75%	0.195700	0.066120	…	5.220158e-01	
max	0.304000	0.097440	…	4.094189e+00	

	worst texture	worst perimeter	worst area	worst smoothness	\
count	5.690000e+02	5.690000e+02	5.690000e+02	569.000000	
mean	-4.532598e-16	-4.015534e-16	-2.848727e-17	0.132369	

```
std     1.000880e+00     1.000880e+00  1.000880e+00          0.022832
min    -2.223994e+00    -1.693361e+00 -1.222423e+00          0.071170
25%    -7.486293e-01    -6.895783e-01 -6.421359e-01          0.116600
50%    -4.351564e-02    -2.859802e-01 -3.411812e-01          0.131300
75%     6.583411e-01     5.402790e-01  3.575891e-01          0.146000
max     3.885905e+00     4.287337e+00  5.930172e+00          0.222600

       worst compactness  worst concavity  worst concave points  \
count         569.000000       569.000000            569.000000
mean            0.254265         0.272188              0.114606
std             0.157336         0.208624              0.065732
min             0.027290         0.000000              0.000000
25%             0.147200         0.114500              0.064930
50%             0.211900         0.226700              0.099930
75%             0.339100         0.382900              0.161400
max             1.058000         1.252000              0.291000

       worst symmetry  worst fractal dimension
count      569.000000               569.000000
mean         0.290076                 0.083946
std          0.061867                 0.018061
min          0.156500                 0.055040
25%          0.250400                 0.071460
50%          0.282200                 0.080040
75%          0.317900                 0.092080
max          0.663800                 0.207500

[8 rows x 30 columns]
```

[63]:
```
from sklearn.model_selection import train_test_split
```

[64]:
```
x= data
```

[66]:
```
x.head()
```

[66]:
```
   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0        17.99         10.38        1.269934   0.984375          0.11840
1        20.57         17.77        1.685955   1.908708          0.08474
2        19.69         21.25        1.566503   1.558884          0.10960
3        11.42         20.38       -0.592687  -0.764464          0.14250
4        20.29         14.34        1.776573   1.826229          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  …      1.886690      -1.359293         2.303601
1                 0.05667  …      1.805927      -0.369203         1.535126
2                 0.05999  …      1.511870      -0.023974         1.347475
3                 0.09744  …     -0.281464       0.133984        -0.249939
4                 0.05883  …      1.298575      -1.466770         1.338539

   worst area  worst smoothness  worst compactness  worst concavity  \
0    2.001237            0.1622             0.6656           0.7119
1    1.890489            0.1238             0.1866           0.2416
2    1.456285            0.1444             0.4245           0.4504
3   -0.550021            0.2098             0.8663           0.6869
4    1.220724            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]
```

[67]:
```
x.describe()
```

[67]:
```
       mean radius  mean texture  mean perimeter     mean area  \
count   569.000000    569.000000    5.690000e+02  5.690000e+02
mean     14.127292     19.289649   -1.272171e-16 -1.900452e-16
std       3.524049      4.301036    1.000880e+00  1.000880e+00
min       6.981000      9.710000   -1.984504e+00 -1.454443e+00
25%      11.700000     16.170000   -6.919555e-01 -6.671955e-01
50%      13.370000     18.840000   -2.359800e-01 -2.951869e-01
75%      15.780000     21.800000    4.996769e-01  3.635073e-01
max      28.110000     39.280000    3.976130e+00  5.250529e+00

       mean smoothness  mean compactness  mean concavity  mean concave points  \
count       569.000000        569.000000      569.000000           569.000000
mean          0.096360          0.104341        0.088799             0.048919
std           0.014064          0.052813        0.079720             0.038803
min           0.052630          0.019380        0.000000             0.000000
25%           0.086370          0.064920        0.029560             0.020310
50%           0.095870          0.092630        0.061540             0.033500
75%           0.105300          0.130400        0.130700             0.074000
max           0.163400          0.345400        0.426800             0.201200
```

```
       mean symmetry  mean fractal dimension  …  worst radius  \
count     569.000000              569.000000  …  5.690000e+02
mean        0.181162                0.062798  …  1.232757e-15
std         0.027414                0.007060  …  1.000880e+00
min         0.106000                0.049960  … -1.726901e+00
25%         0.161900                0.057700  … -6.749213e-01
50%         0.179200                0.061540  … -2.690395e-01
75%         0.195700                0.066120  …  5.220158e-01
max         0.304000                0.097440  …  4.094189e+00

       worst texture  worst perimeter    worst area  worst smoothness  \
count   5.690000e+02     5.690000e+02  5.690000e+02        569.000000
mean   -4.532598e-16    -4.015534e-16 -2.848727e-17          0.132369
std     1.000880e+00     1.000880e+00  1.000880e+00          0.022832
min    -2.223994e+00    -1.693361e+00 -1.222423e+00          0.071170
25%    -7.486293e-01    -6.895783e-01 -6.421359e-01          0.116600
50%    -4.351564e-02    -2.859802e-01 -3.411812e-01          0.131300
75%     6.583411e-01     5.402790e-01  3.575891e-01          0.146000
max     3.885905e+00     4.287337e+00  5.930172e+00          0.222600

       worst compactness  worst concavity  worst concave points  \
count         569.000000       569.000000            569.000000
mean            0.254265         0.272188              0.114606
std             0.157336         0.208624              0.065732
min             0.027290         0.000000              0.000000
25%             0.147200         0.114500              0.064930
50%             0.211900         0.226700              0.099930
75%             0.339100         0.382900              0.161400
max             1.058000         1.252000              0.291000

       worst symmetry  worst fractal dimension
count      569.000000               569.000000
mean         0.290076                 0.083946
std          0.061867                 0.018061
min          0.156500                 0.055040
25%          0.250400                 0.071460
50%          0.282200                 0.080040
75%          0.317900                 0.092080
max          0.663800                 0.207500

[8 rows x 30 columns]
```

[68]:
```
y = target
```

[69]:
```
y.head()
```

[69]:
```
   target
0       0
1       0
2       0
3       0
4       0
```

[70]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2,␣
↪random_state = 2)
```

[71]:
```
print(x_train.shape, x_test.shape, x.shape)
```

```
(455, 30) (114, 30) (569, 30)
```

[72]:
```
from sklearn.decomposition import PCA
```

[83]:
```
from sklearn.linear_model import LogisticRegression
```

[84]:
```
model = LogisticRegression()
```

0.1 before PCA

[85]:
```
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.7/site-packages/sklearn/utils/validation.py:993:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

[85]:
```
LogisticRegression()
```

[86]:
```
ypred= model.predict(x_test)
```

[87]:
```
from sklearn.metrics import accuracy_score
```

[88]:
```
accuracy_score(y_test,ypred)
```

[88]:
```
0.9385964912280702
```

0.2 after PCA

[89]:
```
from sklearn.decomposition import PCA
```

[90]:
```
data.shape
```

[90]:
```
(569, 30)
```

[91]:
```
pca= PCA(n_components= 0.5)
```

[92]:
```
pca.fit(x_train)
```

[92]:
```
PCA(n_components=0.5)
```

[93]:
```
pca.fit(x_test)
```

[93]:
```
PCA(n_components=0.5)
```

[94]:
```
pca_x_train = pca.transform(x_train)
```

[95]:
```
pca_x_test= pca.transform(x_test)
```

[96]:
```
print(pca_x_train.shape, pca_x_test.shape)
```

```
(455, 1) (114, 1)
```

[97]:
```
model.fit(pca_x_train, y_train)
```

```
/usr/local/lib/python3.7/site-packages/sklearn/utils/validation.py:993:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[97]:
```
LogisticRegression()
```

[98]:
```
ypred= model.predict(pca_x_test)
```

[99]:
```
accuracy_score(y_test,ypred)
```

[99]:
```
0.8421052631578947
```

we got 93% accuracy with 30 features and 84% accuracy with only 1 feature

[ ]: