PCA 과제

1. sklearn.datasets.load_breast_cancer에 PCA를 적용하여 2D space로 데이터 차원 축소

2. PCA components의 feature importance visualization

PCA는 안쓰이는 컬럼을 버린다기 보다, 데이터 손실 최소화하여 차원을 줄이고 feature의 중요성을 확인할 수 있다.

In [1]:
```
#데이터 확인

from sklearn.datasets import load_breast_cancer
import numpy as np
import pandas as pd


raw_data = load_breast_cancer()
print('key 값들 ',dir(raw_data))# raw data가 딕셔너리 형태라서 dir로 key값들 확인
print('데이터 ',raw_data.data.shape)
print('feature ',raw_data.feature_names)
DATA=pd.DataFrame(raw_data.data)
DATA.columns=raw_data.feature_names
DATA.head() # pca는 데이터가 정규화가 필요할까?
#공분산은 데이터의 크기에 영향을 받기 때문에 해야된다고 생각.
```

```
key 값들  ['DESCR', 'data', 'feature_names', 'filename', 'frame', 'target', 'target_names']
데이터  (569, 30)
feature  ['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']
```

Out[1]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	wo rad
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	25
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	24
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	22

5 rows × 30 columns

데이터 정규화

min-max scaler 이용

최대값으로 나눠도 되지만,

값을 0~1 비율로 바꿔주기 위해, 분자에서는 최소값을 빼고

분모에서는 최대 - 최소를 해준다.

In [2]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
DATA_norm = scaler.fit_transform(DATA[:])
DATA_norm = pd.DataFrame(DATA_norm)
DATA_norm.columns = raw_data.feature_names
DATA_norm.head()
```

Out[2]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	..
0	0.521037	0.022658	0.545989	0.363733	0.593753	0.792037	0.703140	0.731113	0.686364	0.605518	..
1	0.643144	0.272574	0.615783	0.501591	0.289880	0.181768	0.203608	0.348757	0.379798	0.141323	..
2	0.601496	0.390260	0.595743	0.449417	0.514309	0.431017	0.462512	0.635686	0.509596	0.211247	..
3	0.210090	0.360839	0.233501	0.102906	0.811321	0.811361	0.565604	0.522863	0.776263	1.000000	..
4	0.629893	0.156578	0.630986	0.489290	0.430351	0.347893	0.463918	0.518390	0.378283	0.186816	..

5 rows × 30 columns

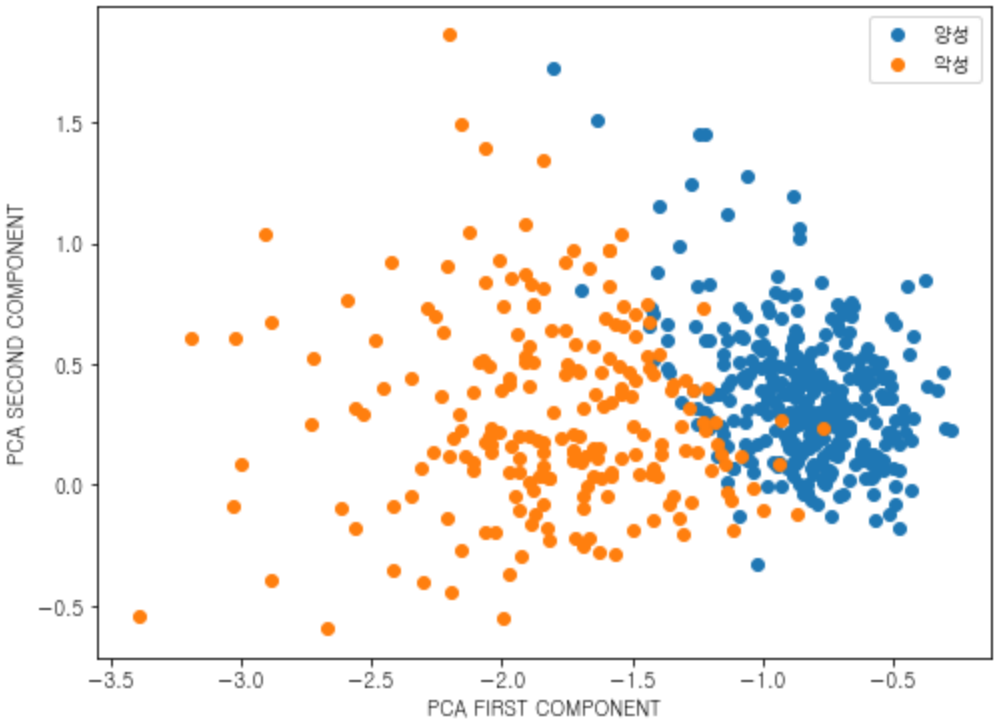
1. 2D SPACE 축소

변환 행렬 U와 데이터의 행렬곱을 통해, 데이터 축소.

In [267…
```
# PCA 구현
# 1. 2D SPACE 축소
from matplotlib import font_manager, rc
font_path = "C:/Windows/Fonts/NGULIM.TTF"
font = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font)

import numpy as np
import pandas as pd
from matplotlib import pyplot as plt


DATA_COV=np.cov(DATA_norm.T)
print('eigenvector 수: ',DATA_COV.shape[1])

#decomposition 후, eigen value가 높은 두 벡터 선택.
## 추후, 자동으로 두개 선택하도록 고쳐야함.
eig_vals, eig_vecs=np.linalg.eig(DATA_COV)
print("데이터 유지 비율: ", np.sum(eig_vals[0:2])/np.sum(eig_vals))
U=eig_vecs[:,0:2]

#데이터 사영
DATA_PCA=DATA_norm.dot(U)


DATA_PCA.columns=['PCA1','PCA2']

fig =plt.figure(figsize=(8,6))
```

```
benign= DATA_PCA[raw_data['target']==1] #양성
malignant = DATA_PCA[raw_data['target']==0] #악성

plt.scatter(benign.iloc[:,0],benign.iloc[:,1],label='양성')
plt.scatter(malignant.iloc[:,0],malignant.iloc[:,1],label='악성')

plt.xlabel('PCA FIRST COMPONENT')
plt.ylabel('PCA SECOND COMPONENT')
plt.legend()
plt.show()

# eigen vector는 상수배를 하면 달라지기 때문에, 방향이 다를 수 도 있다.
```

```
eigenvector 수:  30
데이터 유지 비율:  0.7038117901347682
```



In [261…

```
DATA_PCA.head()
```

Out[261…

	PCA1	PCA2
0	-2.590770	0.762974
1	-1.666057	-0.220868
2	-2.158371	0.226378
3	-2.204565	1.861168
4	-1.830577	0.033608

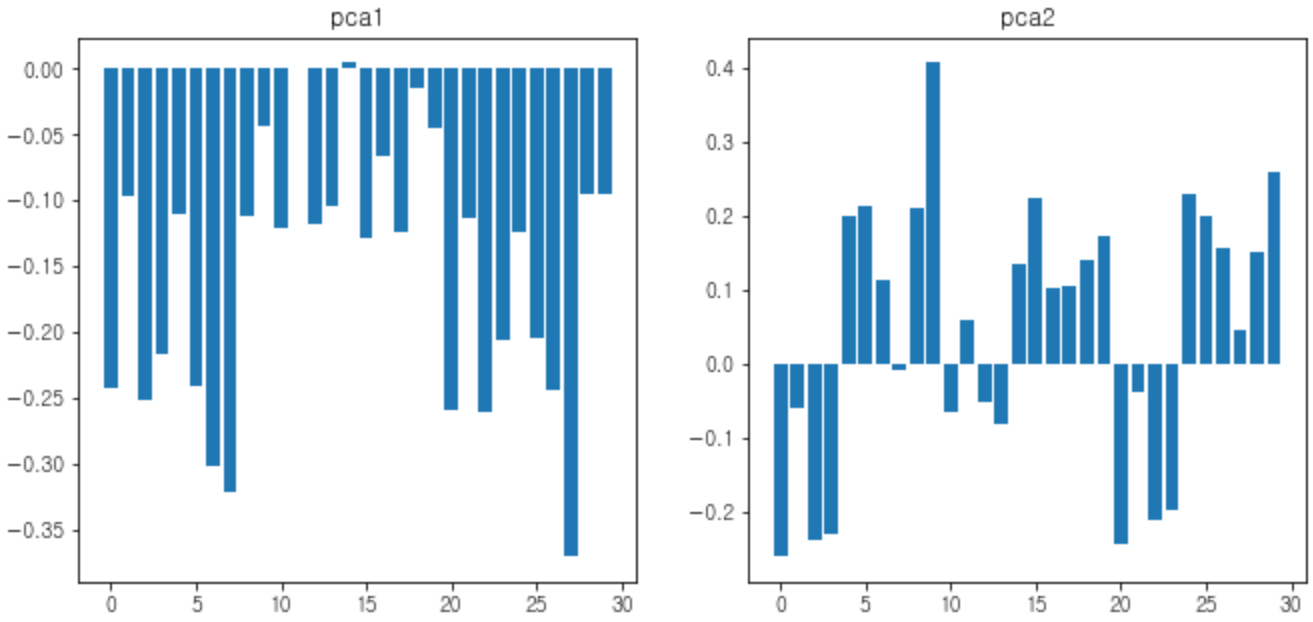
2. PCA components의 feature importance visualization

U행렬. 주성분의 값을 활용해, feature importance 확인.

In [268…

```
#feature visualization
import seaborn as sns

U_df=pd.DataFrame(U.T)
U_df.columns=raw_data.feature_names
```

```
U_df.index=['PCA1','PCA2']
U_df

fig =plt.figure(figsize=(11,11))

plt.subplot(221),plt.title("pca1")# 2행 2열중 1번째
plt.bar(np.arange(30),U_df.loc['PCA1',:])

plt.subplot(222),plt.title("pca2")# 2행 2열중 1번째
plt.bar(np.arange(30),U_df.loc['PCA2',:])

plt.show()
```



In [264…

```
U_df
```

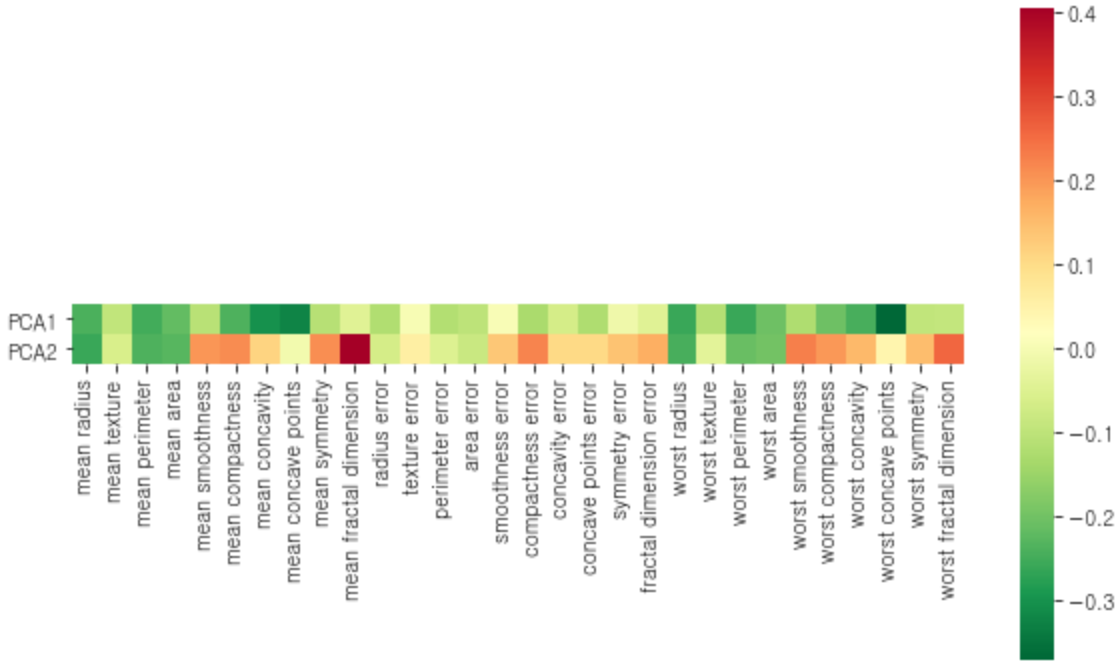
Out[264…

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fr dimer
PCA1	-0.242676	-0.096479	-0.25255	-0.216495	-0.109695	-0.240398	-0.301914	-0.322475	-0.111432	-0.04
PCA2	-0.261317	-0.059058	-0.23859	-0.231107	0.199884	0.213915	0.113811	-0.008312	0.211115	0.40

2 rows × 30 columns

In [269…

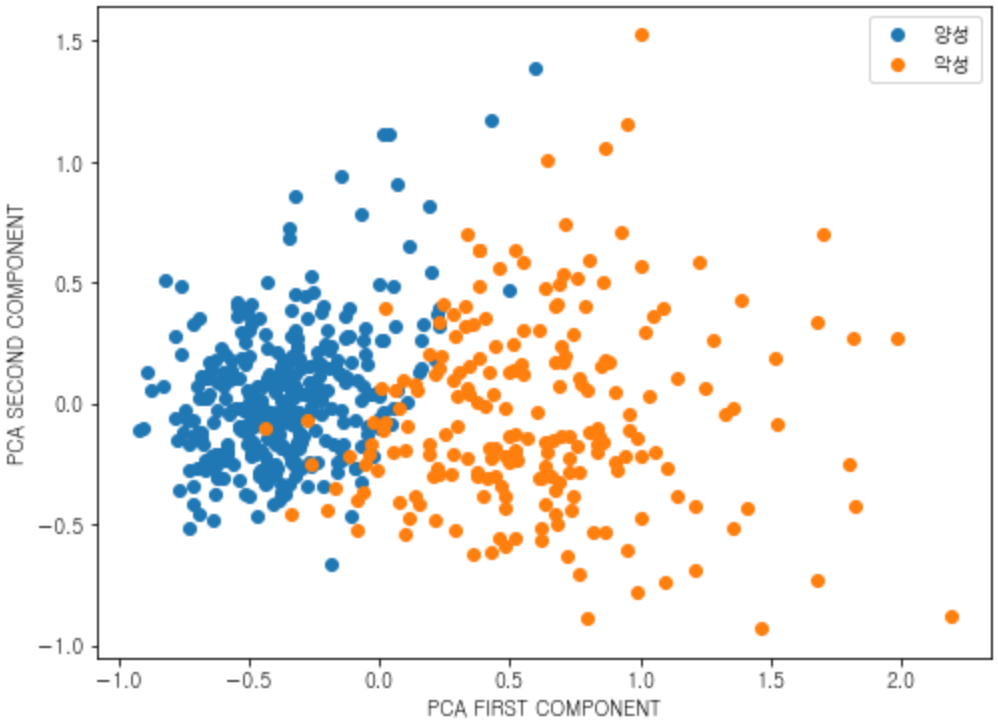
```
fig =plt.figure(figsize=(10,6))
ax = sns.heatmap(U_df,
                 cmap='RdYlGn_r',# cmap Color
                 annot=False,             # Value Text
                 square=True,
                )
#PCA1과 PCA2 모두 각자 다른 feature가 높다.
```

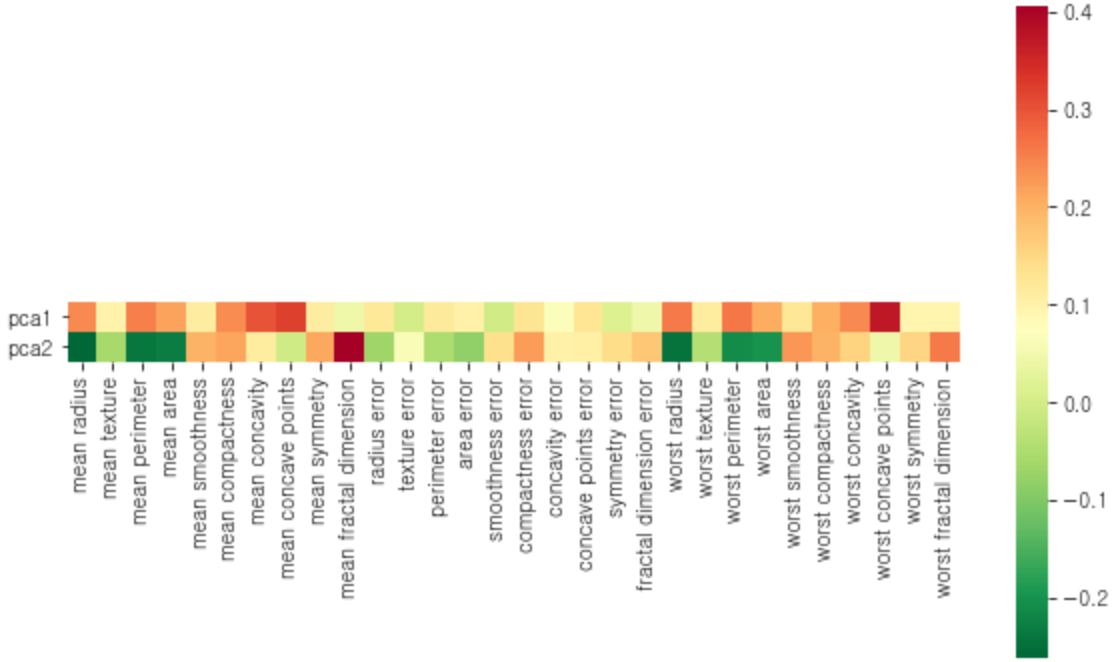


PCA 라이브러리 활용

In [229…

```
# pca 라이브러리 활용.
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA

df2 = pd.DataFrame(raw_data['data'],columns=raw_data['feature_names'])
scaler=MinMaxScaler()
scaler.fit(df2)
scaled_data=scaler.transform(df2)
pca2=PCA(n_components=2)
pca2.fit(scaled_data)

pca2_data=pca2.transform(scaled_data)
pca2_df=pd.DataFrame(pca2_data,columns=['PCA1','PCA2'])

fig =plt.figure(figsize=(8,6))
benign2= pca2_df[raw_data['target']==1] #양성
malignant2 = pca2_df[raw_data['target']==0] #악성

plt.scatter(benign2.iloc[:,0],benign2.iloc[:,1],label='양성')
plt.scatter(malignant2.iloc[:,0],malignant2.iloc[:,1],label='악성')

plt.xlabel('PCA FIRST COMPONENT')
plt.ylabel('PCA SECOND COMPONENT')
plt.legend()
plt.show()
pca2_components_df=pd.DataFrame(pca2.components_,columns=raw_data['feature_names'],index=

fig =plt.figure(figsize=(10,6))
ax = sns.heatmap(pca2_components_df,
                 cmap='RdYlGn_r',# cmap Color
                 annot=False,            # Value Text
                 square=True,
                )
```





In [230…

```
pca2_df.head(5)
```

Out[230…

	PCA1	PCA2
0	1.387021	0.426895
1	0.462308	-0.556947
2	0.954621	-0.109701
3	1.000816	1.525089
4	0.626828	-0.302471

In [11]:

```
print('데이터 ',raw_data.DESCR)
```

데이터 .. _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
        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/

.. 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.
```

참조.

https://losskatsu.github.io/machine-learning/sklearn/#%EA%B0%80%EC%83%81-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EB%9E%9C%EB%8D%A4%EC%9C%BC%EB%A1%9C-%EC%83%9D%EC%84%B1%ED%95%98%EA%B8%B0

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html

https://jfun.tistory.com/108

https://blog.naver.com/pmw9440/221843085568 : pca의 역할

https://blog.naver.com/pmw9440/221843886378 :왜 정규화를 해야하는지. 각 원소별 load

https://soo-jjeong.tistory.com/122 :min-max scaler 설명

https://www.youtube.com/watch?v=QdBy02ExhGI : 시각화 https://rfriend.tistory.com/419 : sns heatmap의 cmap 종류들