

## 비지도 학습

- 군집, 비지도 차원 축소 등.

군집 (Clustering) : 비슷한 특징을 가지는 데이터 인스턴스들끼리 그룹화.

비지도 차원 축소 :

대표적 예) 시각화를 위해 데이터셋을 2차원으로 변경

## [ 차원 축소 ]

### 주성분 분석 ( PCA )

비지도 학습을 사용해 데이터를 변환하는 이유는 여러가지입니다.

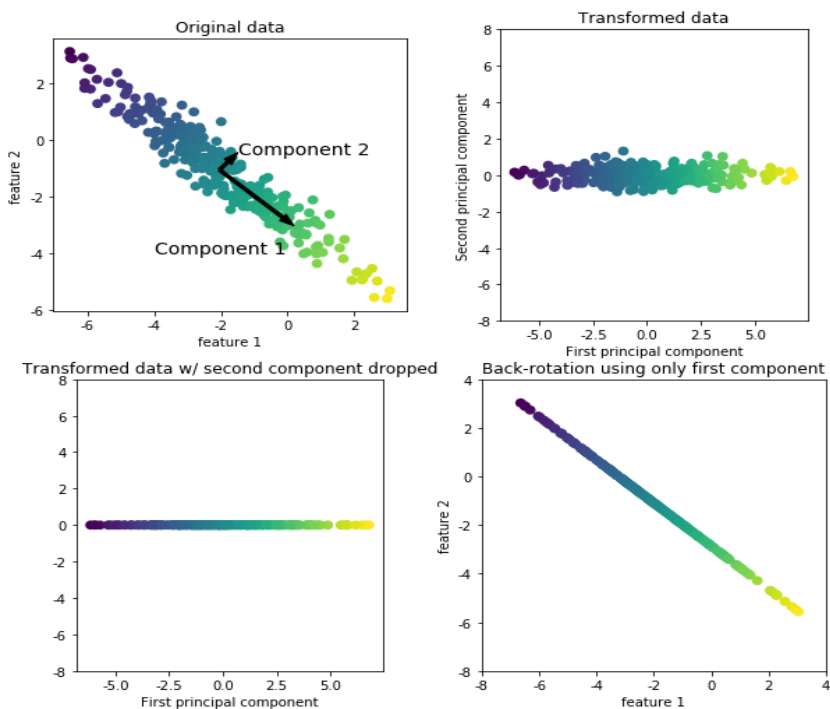
가장 일반적인 동기는 시각화하거나, 데이터를 압축하거나, 추가적인 처리를 위해 정보가 더 잘 드러나는 표현을 찾기 위해서 입니다.

이런 용도로 가장 간단하고 흔히 사용하는 알고리즘은 주성분 분석입니다.

ex22.py      PCA 처리 방식 설명

```
import mglearn
import matplotlib.pyplot as plt

mglearn.plots.plot_pca_illustration()
plt.show()
```



왼쪽 위 그래프는 원본 데이터 포인트를 색으로 구분해 표시, 이 알고리즘은 먼저 'component1'이라고 쓰여 있는,

Variant(분산)이 가장 큰 방향을 찾습니다. 이 방향(벡터)가 가장 많은 정보를 담고 있는 방향이고 ==> 특성들의 상관관계가 가장 큰 방향입니다.

그 다음에 이 algorithm은 첫 번째 방향과 직각인 방향 중에서 가장 많은 정보를 담은 방향을 찾습니다.

2차원에서는 직각 방향은 하나지만 3차원이상부터는 무수히 많은 직각 방향이 존재하고, 이 그래프에서 화살표 머리방향은 의미가 없습니다.

이런 방법으로 찾은 방향이 데이터의 주된 Variant(분산)의 방향이라 해서 principal component(주성분)이라 합니다. 일반적으로 원본 특성 개수만큼의 주성분 존재합니다.

오른쪽 위 그래프는 주성분1과 2를 각각 x축과 y축에 나란하도록 회전시켰습니다.

회전하기전에 데이터에서 평균을 빼서 중심을 원점에 맞춥니다.

PCA에 의해 회전된 두 축은 독립이므로 변환된 데이터의 correlation matrix(상관관계 행렬)이 대각선 방향(자기자신)을 제외하고는 0이 나옵니다.

PCA는 주성분의 일부만 남기는 차원 축소 용도로 사용할 수 있습니다. 왼쪽 아래 그림은 첫 번째 주성분만 유지하려고 하며 2차원 데이터셋이 1차원 데이터셋으로 감소하지만 단순히 원본 특성 중 하나만 남기는 것이 아니라, 첫 번째 방향의 성분을 유지하도록 데이터를 가공합니다.

마지막으로 데이터에 다시 평균을 더해서 반대로 회전(오른쪽 아래 그림)

이 데이터들은 원래 특성 공간에 놓여 있지만 첫 번째 주성분의 정보만 담고 있음

이 변환은 데이터에서 노이즈를 제거하거나 주성분에서 유지되는 정보를 시각화 하는데 사용됩니다.

PCA가 가장 널리 사용되는 분야는 고차원 데이터셋의 시각화 영역입니다.

보습학원 사례의 데이터셋은 특성이 5개나 되므로 산점도가 시각적으로 직관적일 수 없습니다.

breast cancer와 같은 데이터셋은 특성이 30개나 있어서 30개중 2개를 택하는 경우의 수인 435개의 산점도를 그려야하므로 단순한 시각화가 비효율적입니다.

사례) breast cancer 데이터 셋 시각화하기 1) 이 데이터 셋은 30개의 특성을 가지고 있다. 이렇게 많은 그래프는 이해하기는 커녕 자세히 들여다 볼 수도 없습니다.

ex23.py

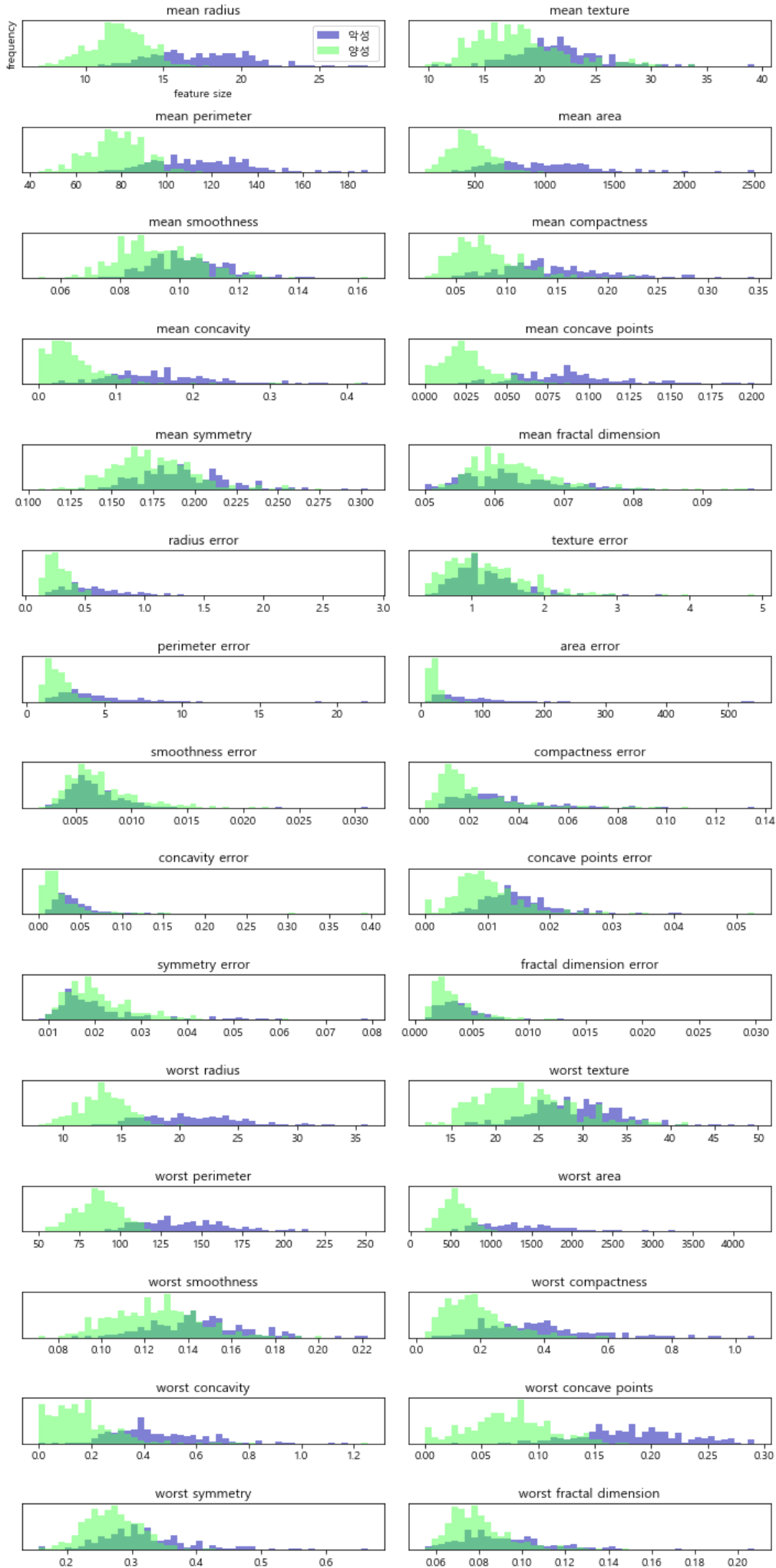
```
import mglearn
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
from sklearn.datasets import load_breast_cancer
import numpy as np

cancer = load_breast_cancer() #breast_cancer sample 데이터

# 특성이 30개이므로 15X2 set의 plot 객체 생성합니다.
fig, axes = plt.subplots(15, 2, figsize=(10,20))

malignant = cancer.data[cancer.target == 0] #악성 데이터
benign = cancer.data[cancer.target == 1] #양성 데이터
target_set = np.array([malignant, benign])
ax = axes.ravel()

for i in range(30) :
    _, bins = np.histogram(cancer.data[:,i], bins=50)# bins: histogram 간격
    ax[i].hist(malignant[:,i], bins=bins,color=mglearn.cm3(0), alpha=0.5)
    ax[i].hist(benign[:,i], bins=bins,color=mglearn.cm3(2), alpha=0.5)
    ax[i].set_title(cancer.feature_names[i])
    ax[i].set_yticks(())
ax[0].set_xlabel('feature size')
ax[0].set_ylabel('frequency')
ax[0].legend(['악성 ', '양성 '], loc="best" )
fig.tight_layout()
```



사례) PCA를 적용하여 차원 축소 후, breast cancer 데이터 셋 시각화하기 2)

ex24\_PCA.py

1) PCA 적용 전 각 특성의 평균이 0, 분산이 1이 되도록 데이터의 스케일을 조정

```
#####
#[Code 1 ]
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer() #breast_cancer sample 데이터

#각 특성의 분산이 1이 되도록 데이터의 스케일조정
scaler = StandardScaler()
scaler.fit(cancer.data)
X_Scaled = scaler.transform(cancer.data)
#####
```

ex24\_PCA.py (계속)

2) PCA 적용. 두 개의 주성분만 유지하는 데이터 변환 (차원축소)

```
#####
#[Code 2 ]
from sklearn.decomposition import PCA
pca = PCA(n_components=2) # 두 개의 주성분만 유지 .
pca.fit(X_Scaled) # PCA 모델

X_pca = pca.transform(X_Scaled) # 두 개의 주성분만 유지하는 데이터 변환 (차원축소)
print( X_Scaled.shape ) #(569, 30) 569행 30열
print( X_pca.shape ) #(569, 2) 차원 축소 후 569행 2열

#####
```

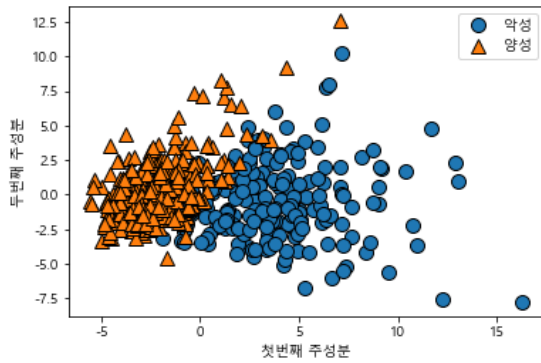
ex24\_PCA.py (계속)

3) 산점도 그래프(첫, 두번째 주성분으로 산점도 그래프, cancer.target 0 악성, 1 양성데이터에 따라 다른 색표현

```
#####
#[Code 3 ]
import mglearn
import matplotlib.pyplot as plt

mglearn.discrete_scatter(X_pca[:,0],X_pca[:,1], cancer.target)
plt.legend(["악성","양성"],loc="best")
plt.xlabel("첫번째 주성분")
plt.ylabel("두번째 주성분")
plt.show()

#####
```

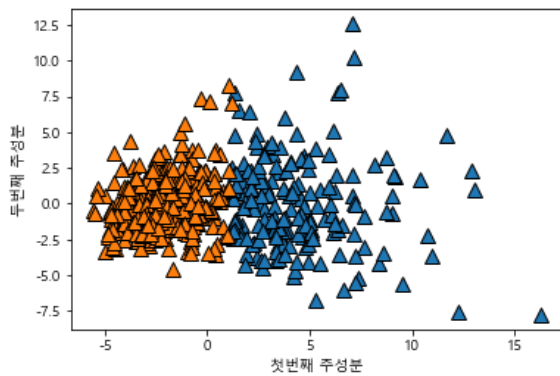


ex24\_PCA.py (계속)

4) cancer.target 데이터를 활용하지 않고, 첫번째 두번째 주성분으로 군집화

```
##### 군집화

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2)
kmeans.fit(X_Scaled)
mglearn.discrete_scatter(X_pca[:,0],X_pca[:,1],kmeans.labels_,markers='^')
plt.xlabel("첫번째 주성분")
plt.ylabel("두번째 주성분")
plt.show()
```



cancer.target 데이터와 유사하게 군집.