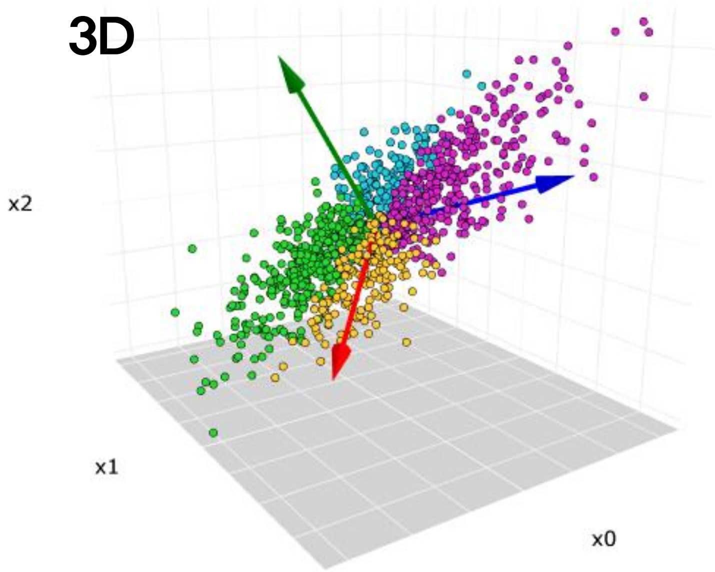
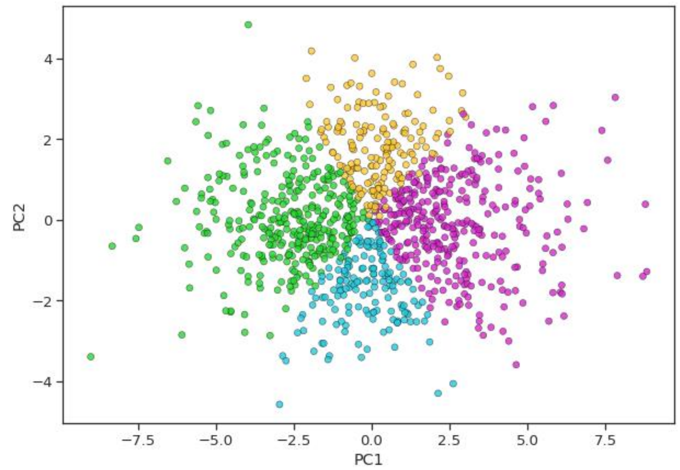
PCA(주성분분석, Principal Component Analysis)

3D



dimension reduction (차원축소)

2D



차원 축소 의미

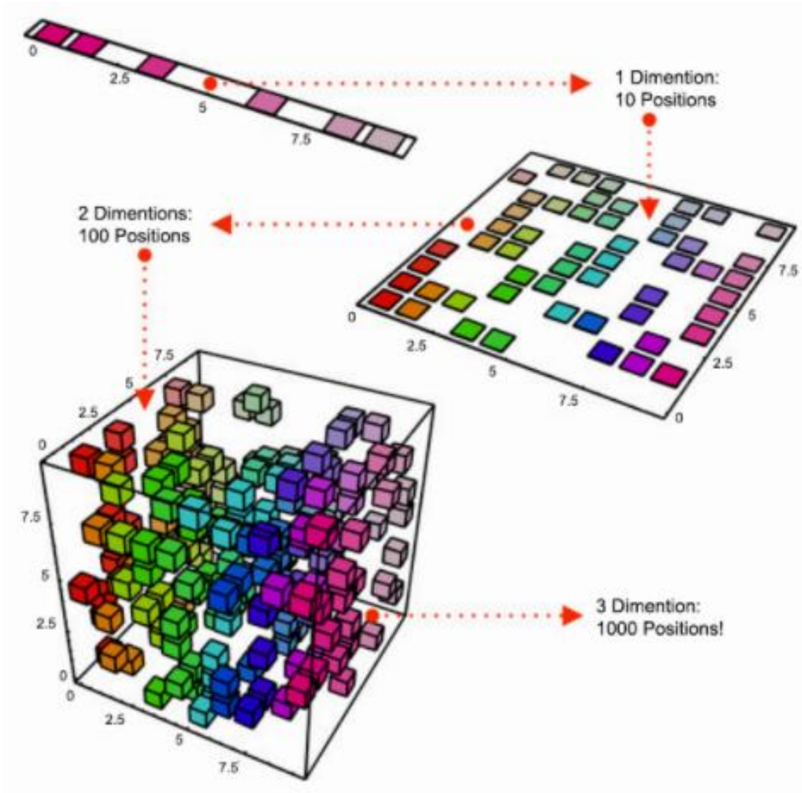
- 75%의 이미지를 제거해도 수자를 인식할 수 있음

https://towardsdatascience.com/

- Why Feature selection?
  - Memory
  - Time
  - Accuracy
  - Interpretability
  - Debugging

The curse of dimensionality

- 차원의 저주 : 데이터의 차원이 높아질 수록 알고리즘의 실행이 아주 까다로워지는 일

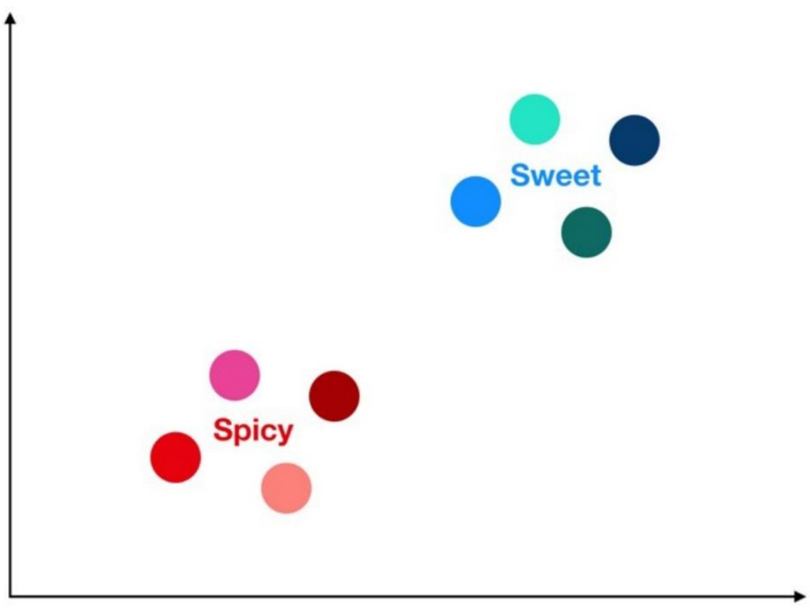


segment, square, cube (1D to 3D cubes)

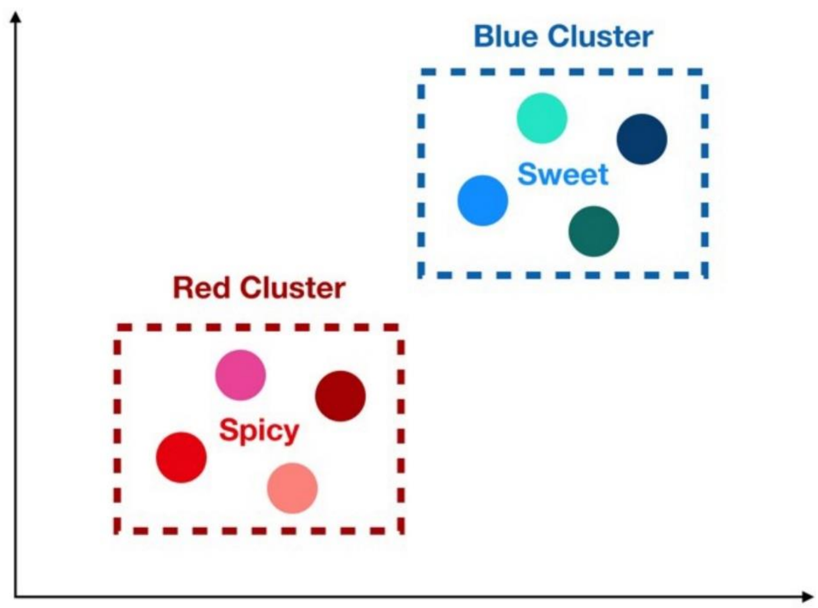
http://www.infme.com/curse-of-dimensionality-ml-big-data-ml-optimization-pca/

The curse of dimensionality

2차원 변수, 8개 instances의 데이터에 대한 군집(Clustering)



Our Two Candy Clusters



Our 2 Color Based Clusters of Candy Flavor

High cardinality

- 8차원 변수, 8개 instances의 데이터에 대한 군집(Clustering)에 문제 발생
  - 일반화(Generalization)이 아닌 memorization
  - 극단적 과적합(Overfitting) 발생, 과거 데이터에 너무 관심을 두고 학습

	Reddish	Bluish
	1	0
	1	0
	1	0
	1	0
	0	1
	0	1
	0	1
	0	1

Perfect Clusters

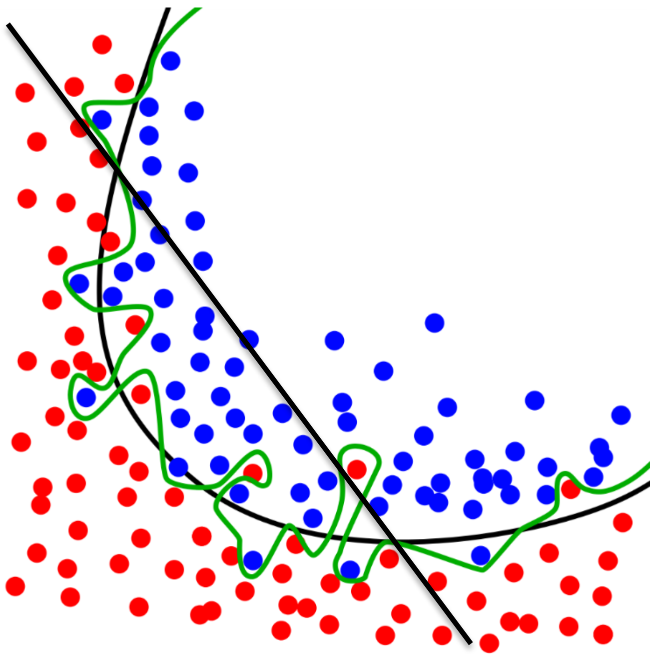
	Red	Maroon	Pink	Flamingo	Blue	Turquoise	Seaweed	Ocean
	1	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0
	0	0	1	0	0	0	0	0
	0	0	0	1	0	0	0	0
	0	0	0	0	1	0	0	0
	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	1

High Dimensional Data Makes Trouble For Clustering

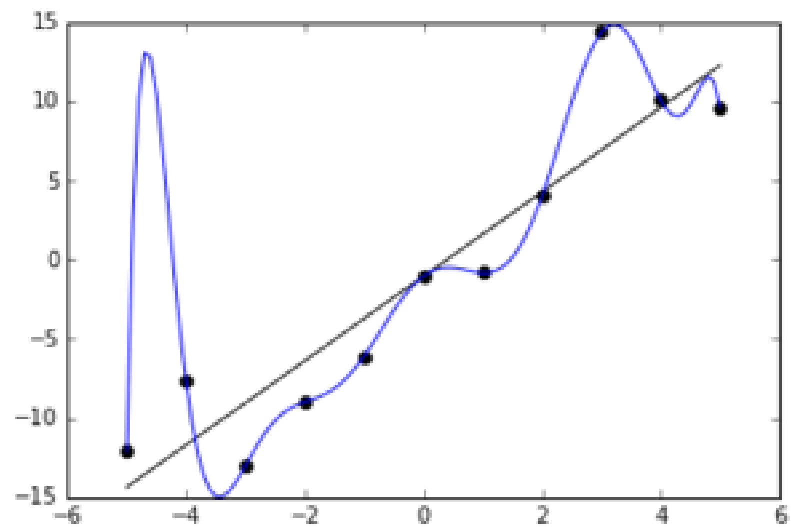
https://towardsdatascience.com/

과적합(Overfitting), Regularization(penalty)

- 다항식 함수가 완벽하게 적합하더라도 선형 함수가 더 잘 일반화된 것
- 두 함수를 사용하여 예측을 하게되면 선형 함수가 더 나은 예측을 하게 됨



https://en.wikipedia.org/wiki/Overfitting



차원 축소의 방법

- **projection (선형)**

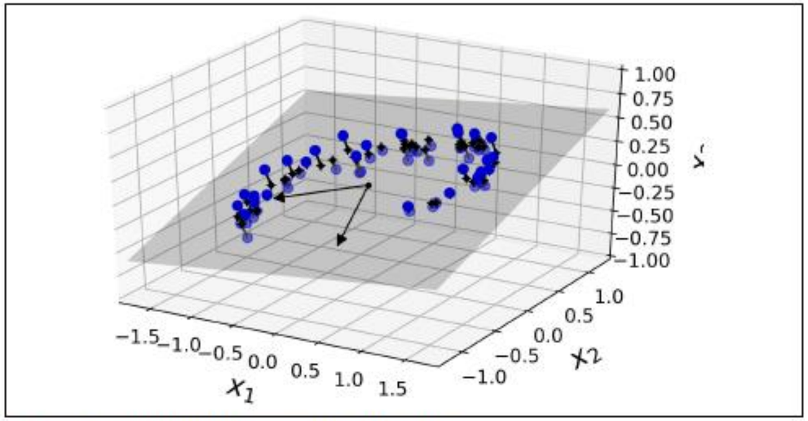


Figure 8-2. A 3D dataset lying close to a 2D subspace

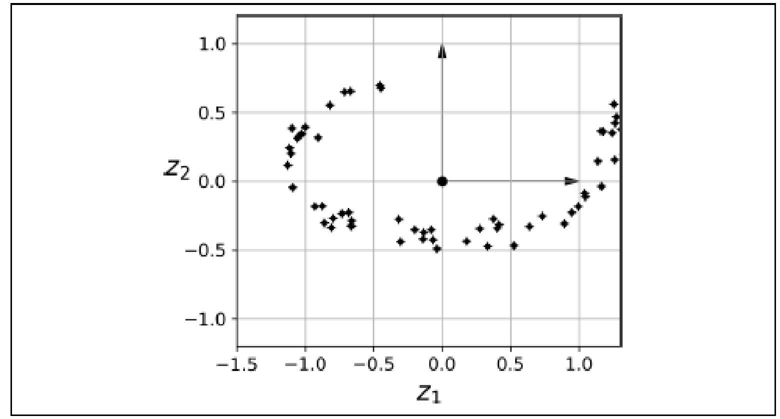


Figure 8-3. The new 2D dataset after projection

- **manifold (비선형)**

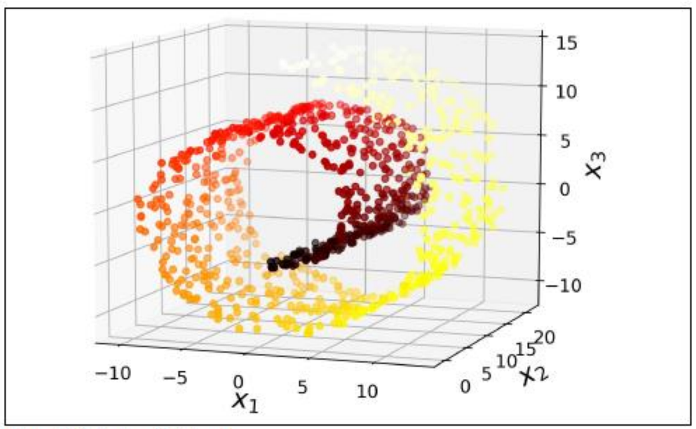


Figure 8-4. Swiss roll dataset

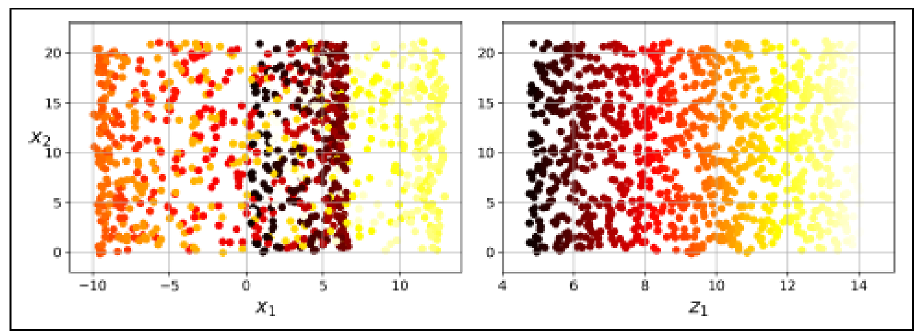


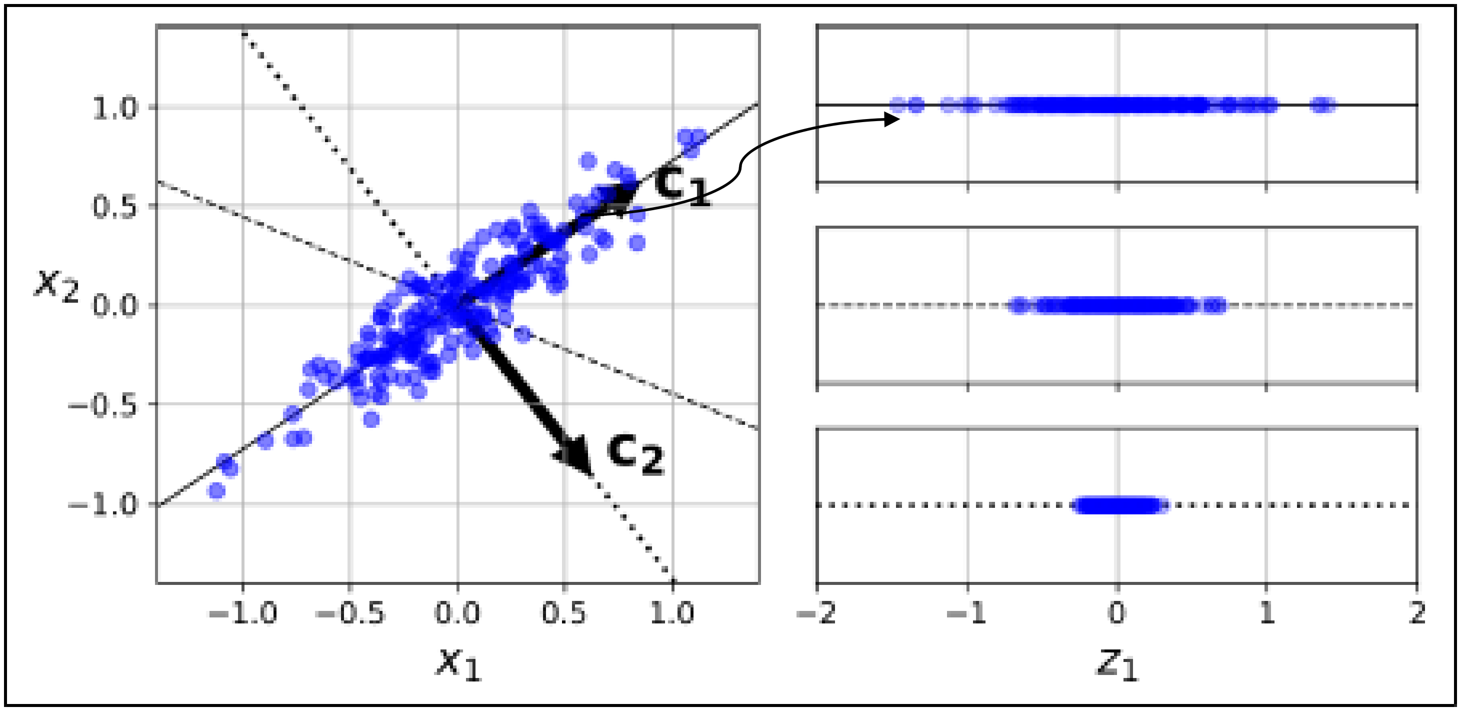
Figure 8-5. Squashing by projecting onto a plane (left) versus unrolling the Swiss roll (right)

Hands-on Machine Learning-with-Scikit-Learn, Keras & TensorFlow

PCA_총변동의 보존

- 주성분(principal components)
  - 총변동의 대부분을 설명하는 소수의 새로운 변수 (C2 벡터)

✓ The projection of dataset onto three axes.
✓ The Solid line(principal component) preserves the maximum variance.



Source : Hands on Machine Learning with Scikit Learn and Tensorflow

PCA represented by Matrix

- n 개의 관측치, k개의 변수를 아래 $n \times k$ 행렬로 표현

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1k} \\ X_{21} & X_{22} & \dots & X_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{nk} \end{bmatrix} = (X_1, X_2, \dots\dots\dots X_k)$$

(Mean centering)

$$\text{평균 조정된 X} = \begin{bmatrix} X_{11} - \overline{X_1} & X_{12} - \overline{X_2} & \dots & X_{1k} - \overline{X_k} \\ X_{21} - \overline{X_1} & X_{22} - \overline{X_2} & \dots & X_{2k} - \overline{X_k} \\ \vdots & \vdots & \dots & \vdots \\ X_{n1} - \overline{X_1} & X_{n2} - \overline{X_2} & \dots & X_{nk} - \overline{X_k} \end{bmatrix}$$

(Standardizing)

$$\text{표준화된 X} = \begin{bmatrix} \frac{X_{11} - \overline{X_1}}{S_1} & \frac{X_{12} - \overline{X_2}}{S_2} & \dots & \frac{X_{11} - \overline{X_k}}{S_k} \\ \frac{X_{21} - \overline{X_1}}{S_1} & \frac{X_{22} - \overline{X_2}}{S_2} & \dots & \frac{X_{21} - \overline{X_k}}{S_k} \\ \vdots & \vdots & \dots & \vdots \\ \frac{X_{n1} - \overline{X_1}}{S_1} & \frac{X_{n2} - \overline{X_2}}{S_2} & \dots & \frac{X_{n1} - \overline{X_k}}{S_k} \end{bmatrix}$$

PCA_총변동

$$X = (X_1, X_2, X_3) = \begin{bmatrix} 2 & 3 & 11 \\ 3 & 5 & 8 \\ 4 & 7 & 5 \\ 3 & 5 & 8 \end{bmatrix} \qquad \textit{평균 조정된 } X = \begin{bmatrix} -1 & -2 & 3 \\ 0 & 0 & 0 \\ 1 & 2 & -3 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\bar{X} = (3,\ 5,\ 8) \qquad \overline{X_1} \quad \overline{X_2} \quad \overline{X_3}$$

- *평균 조정된 X * transpose(평균 조정된 X)*
- 표본 공분산

$$\begin{bmatrix} -1 & 0 & 1 & 0 \\ -2 & 0 & 2 & 0 \\ 3 & 0 & -3 & 0 \end{bmatrix} \times \begin{bmatrix} -1 & -2 & 3 \\ 0 & 0 & 0 \\ 1 & 2 & -3 \\ 0 & 0 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0.667 & 1.3333 & -2 \\ 1.333 & 2.6667 & -4 \\ -2 & -4 & 6 \end{bmatrix}$$

총변동은 trace (trans(X)*X로 방행렬의 대각선 원소의합

$$\Sigma = \begin{bmatrix} \sigma_{00}^2 & \sigma_{01}^2 & \sigma_{02}^2 \\ \sigma_{10}^2 & \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{20}^2 & \sigma_{21}^2 & \sigma_{22}^2 \end{bmatrix}$$

PCA as geometry

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

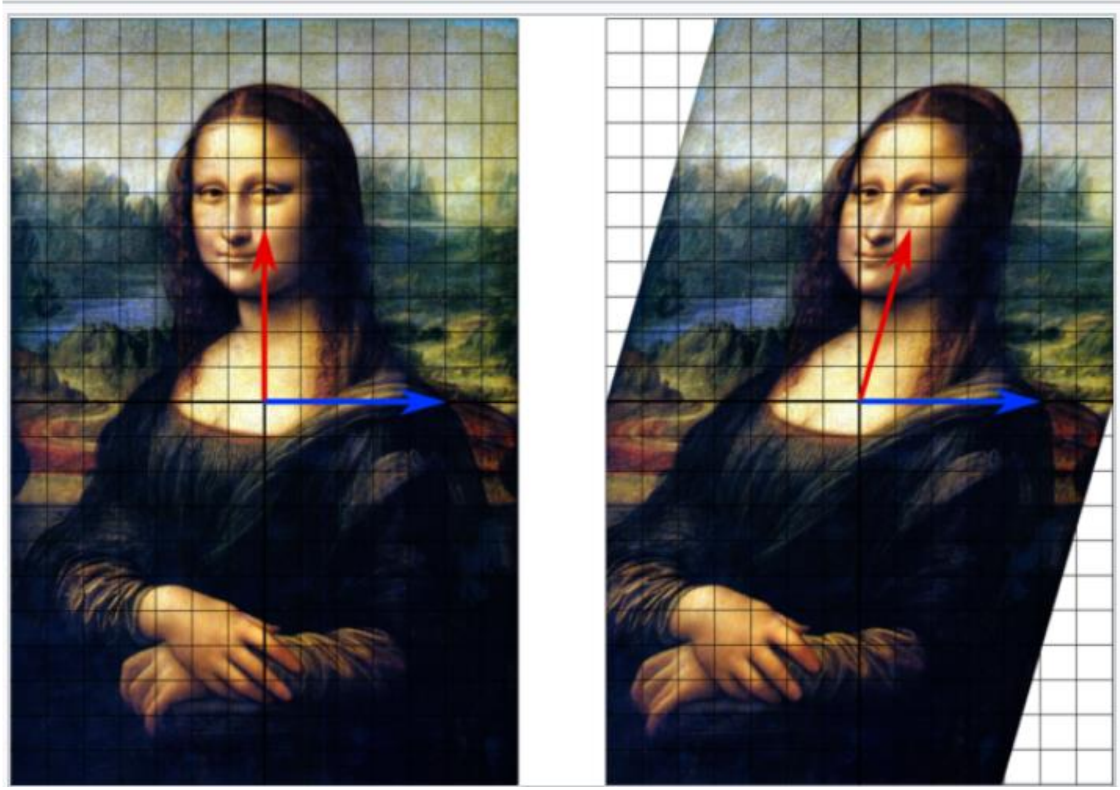
$$|A - \lambda I| = \left| \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix}$$
$$= 3 - 4\lambda + \lambda^2.$$

$$\mathbf{v}_{\lambda=1} = \begin{bmatrix} v_1 \\ -v_1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\mathbf{v}_{\lambda=3} = \begin{bmatrix} v_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

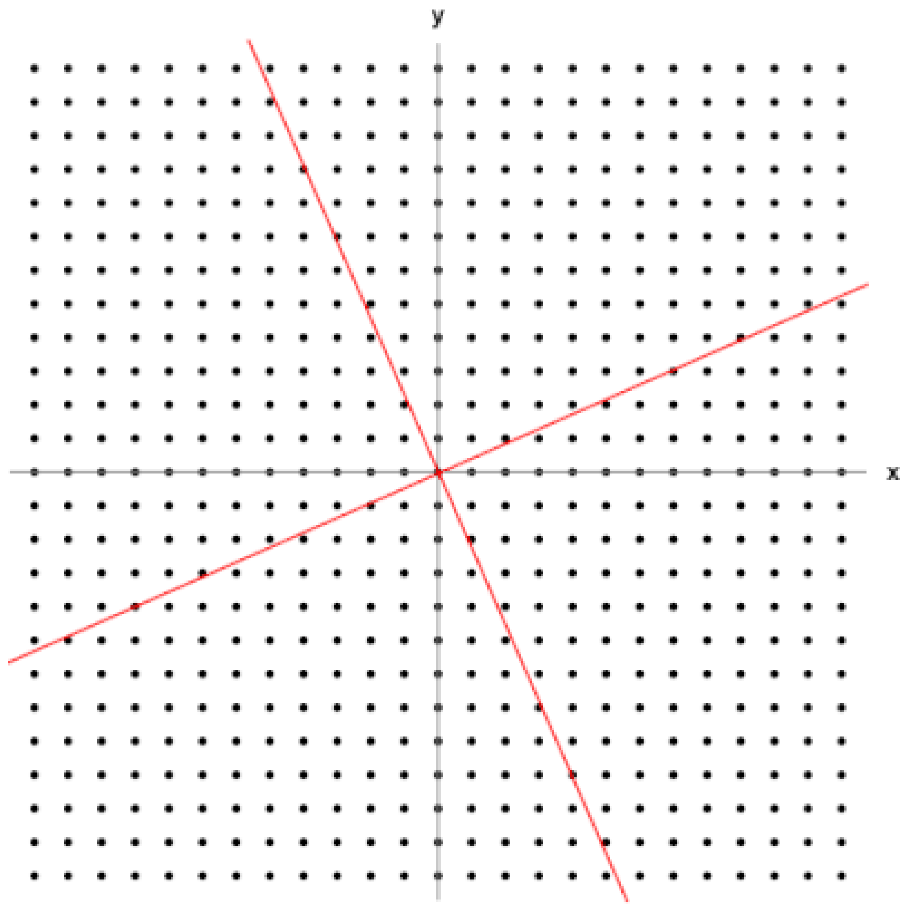
$$(\mathbf{A} - \lambda_i \mathbf{I})\,\mathbf{v} = 0.$$



https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

PCA as geometry

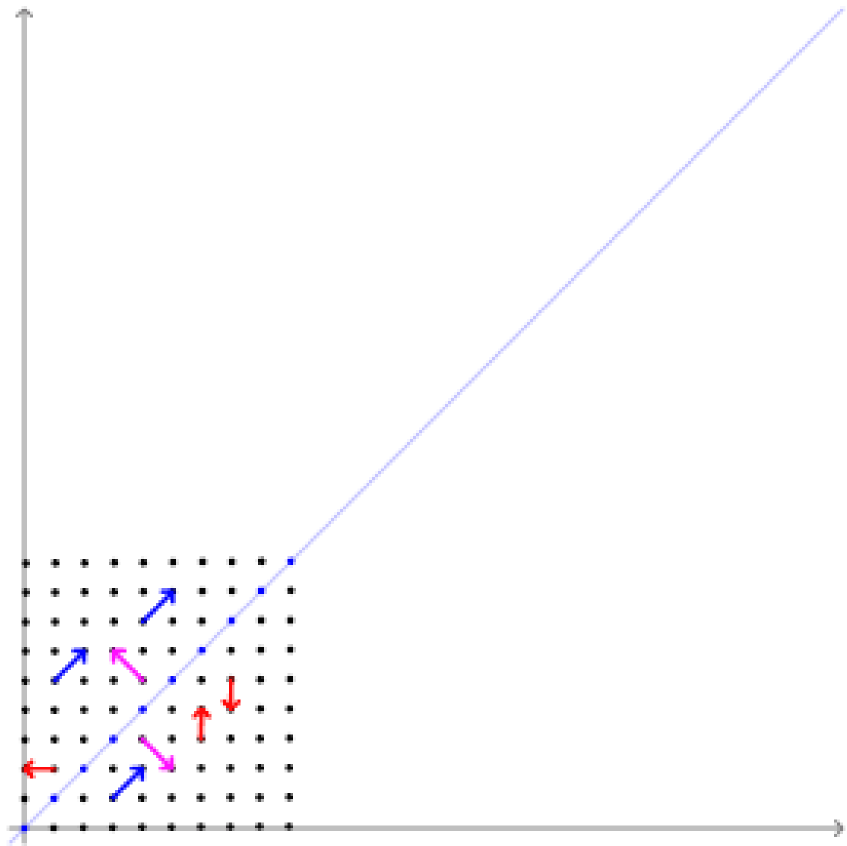
Matrix A does not acts eigenvectors.



https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

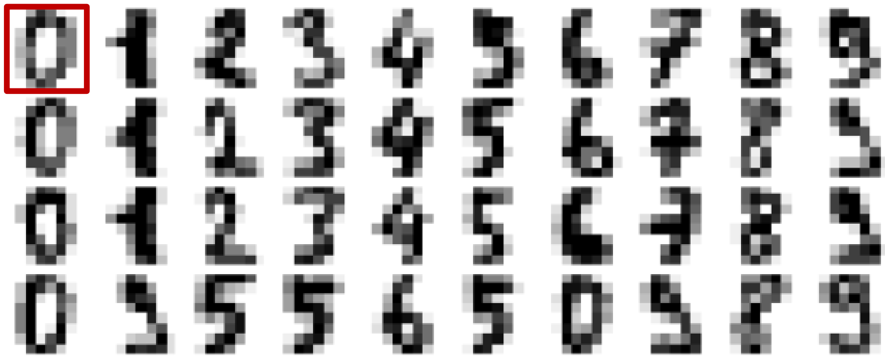
PCA as geometry

Matrix A does not acts eigenvectors.



https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

주성분의 의미



Source : Python Datascience Handbook

```
digits.data.shape
digits.data[0]
digits.data
```

```
(1797, 64)
```

```
array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
       15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
       12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
        0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
       10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.])
```

```
array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ..., 10.,  0.,  0.],
       [ 0.,  0.,  0., ..., 16.,  9.,  0.],
       ...,
       [ 0.,  0.,  1., ...,  6.,  0.,  0.],
       [ 0.,  0.,  2., ..., 12.,  0.,  0.],
       [ 0.,  0., 10., ..., 12.,  1.,  0.]])
```

- 이미지 훈련데이터를 64개의 pixel(변수)로 나타내는 벡터(vector, x)로 가정

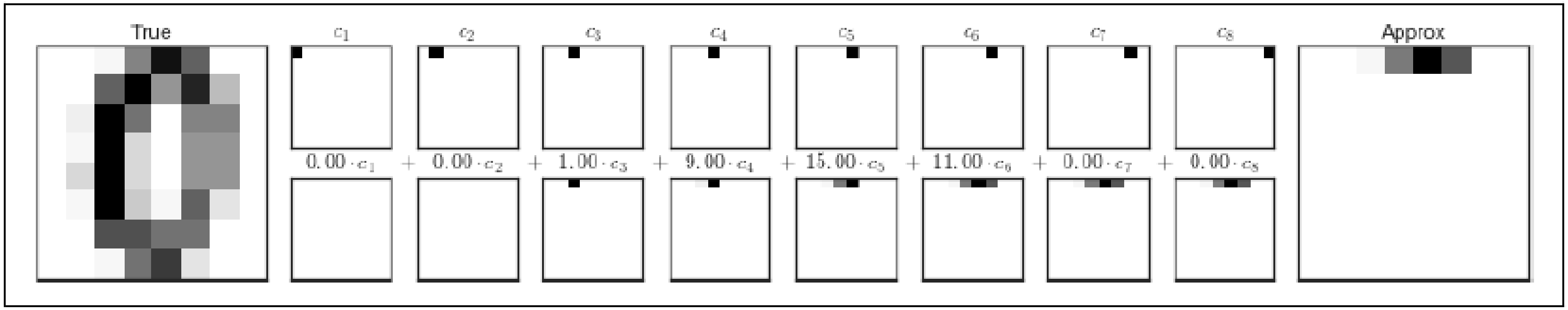
$$x = [x_1, x_2, x_3 \cdots x_{64}]$$

- 64개 차원의 기저(basis)를 갖는 함수로 개별 이미지를 선형결합으로 표현

$$\text{image}(x) = x_1 \cdot (\text{pixel } 1) + x_2 \cdot (\text{pixel } 2) + x_3 \cdot (\text{pixel } 3) \cdots x_{64} \cdot (\text{pixel } 64)$$

주성분의 의미

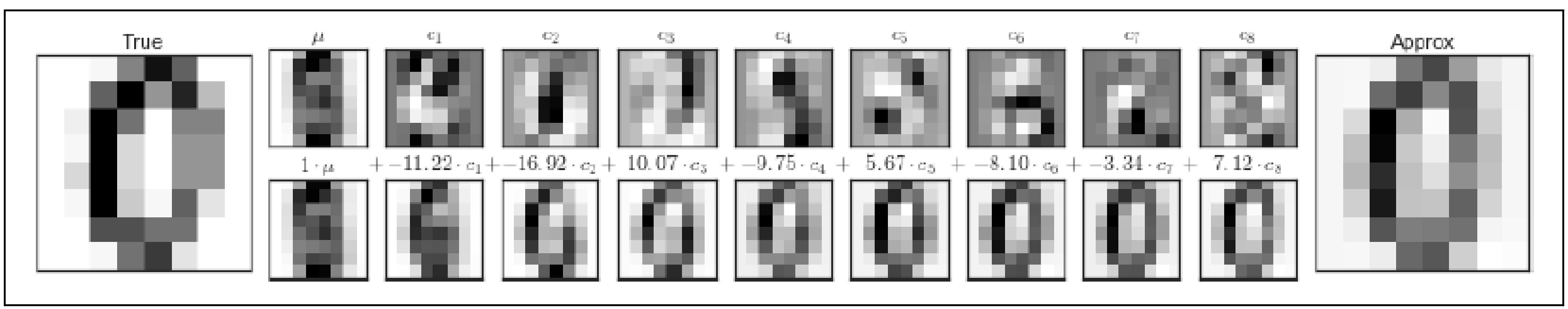
- 원본이미지(64개 변수)의 처음 8개 pixels(기저)을 basis로 하여 8차원으로 표현(projection) 하면 의미없는 이미지를 나타냄



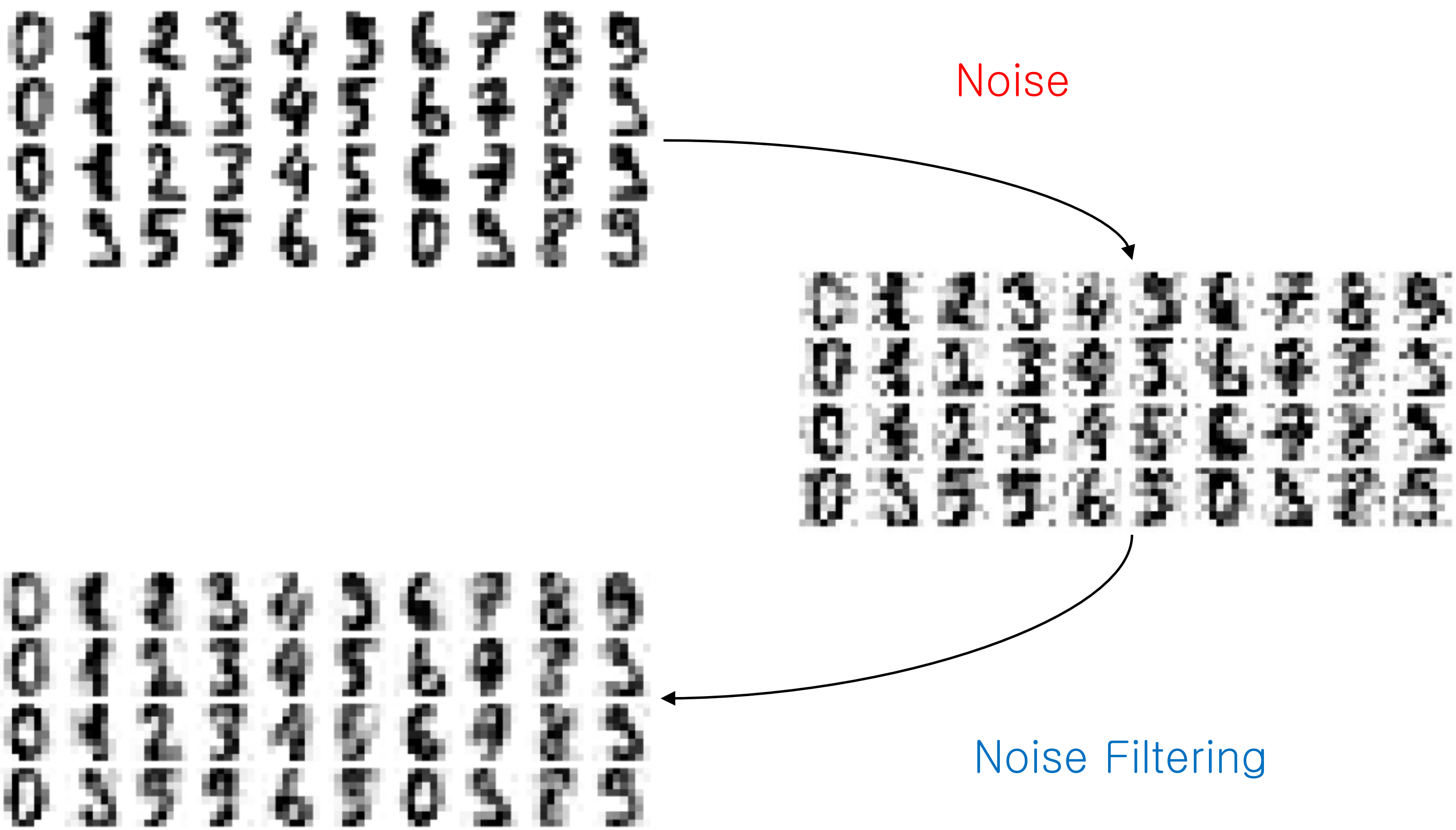
Source : Python Datascience Handbook

- 단순히 64개 pixel 중에서 임의로 몇 개를 골라 projectio하는 것은 의미가 없음

- 주성분분석을 통해 총변동량을 가장 많이 설명해주는 기저함수(basis functions, PCA basis function)로 구성하여 이미지를 나타내면 원본 이미지와 거의 동일함



PCA as Noise Filtering



Source : Python Datascience Handbook

PCA as image recognition

- 고차원 이미지데이터(62 * 47 pixel, 2,914 dimensionality)를 주성분 150개로 인지하여 원래 이미지로 복원

고차원 이미지데이터(62 * 47 pixel, 2,914 dimensionality)

150 Principal Components

Image reconstruction

Source : Python Datascience Handbook