* **차원의 저주**

**차원이 커질수록** 🡪 데이터 포인트들간 거리가 크게 늘어남

🡪 **데이터가 희소화(Sparse)** 됨

🡪 피처가 많을 경우 **개별 피처 간의 상관관계가 높아** 선형회귀와 같은 모델에서는 다중 공선성 문제로 **모델의 예측 성능이 저하**될 가능성이 높음

* **차원 축소의 장점**

수십-수백개의 피처들을 작은 수의 피처들로 축소한다면?

* 학습 데이터크기를 줄여 **학습 시간절약**
* 불필요한 피처들을 줄여 **모델 성능 향상**에 기여 (주로 **이미지 관련** 데이터)
* 다차원의 데이터를 차원 축소(3차원이하)를 통해 시각적으로 보다 **쉽게 데이터 패턴 인지**
* **피처 선택과 피처 추출**

**피처 선택(Feature Selection)** : 특정 피처에 **종속성이 강한 불필요한 피처는 아예 제거**,

데이터의 특징을 잘 나타내는 **주요 피처만 선택함**

**피처 추출(Feature Extraction)** : 기존 피처를 저차원의 중요 피처로 압축해서 추출

(새로운 피처를 추출하는 것.)

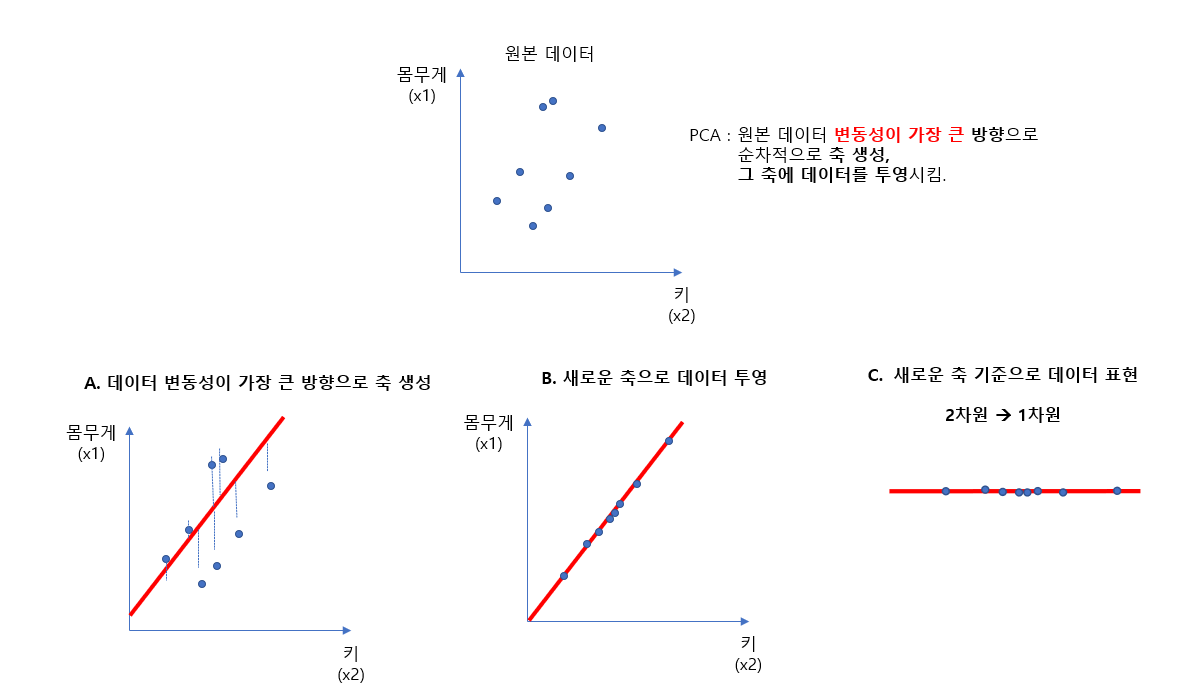
* **차원 축소의 의미**

데이터를 더 잘 설명할 수 있는 **잠재적인( Latent ) 요소를 추출**하는 것.

* 추천엔진 / 이미지 분류 및 변환 / 문서 토픽 모델링
* **PCA(Principal Component Analysis)의 이해**

고차원의 원본 데이터를 저차원의 부분공간으로 **투영**하여 데이터를 축소하는 기법

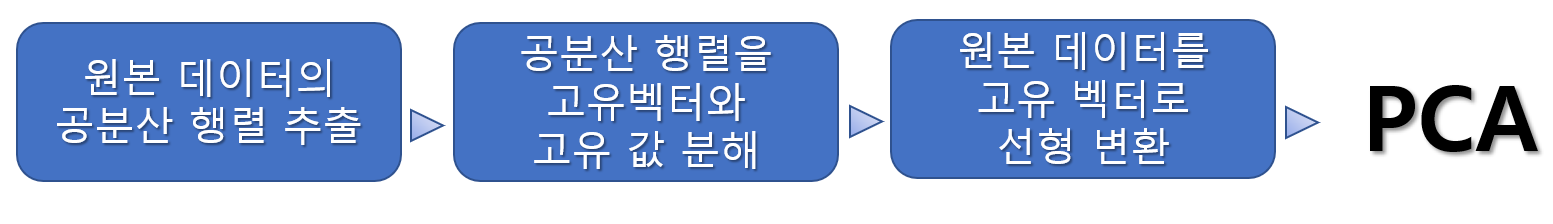
**데이터 변동성**을 **가장 중요한 정보**로 간주하며 이 변동성을 기반하여 원본 데이터를 투영시킴



가장 큰 데이터 변동성(Variance)을 기준으로 첫 번째 벡터 축을 생성하고, 두 번째 축은 첫 번째 벡터와 **직각**이 되는 벡터 축이다. 세 번째 벡터 축 역시 2번째 벡터 축에 **직각**인 벡터 축이다.

**PCA, 즉 주성분 분석**은 이처럼 원본 데이터의 피처 개수에 비해 매우 작은 주성분으로 **원본 데이터의 총 변동성을 대부분 설명**할 수 있는 분석법임

* **PCA변환**



**고유벡터 :**  PCA의 주성분 벡터로 **입력 데이터의 분산이 큰 방향**을 나타냄

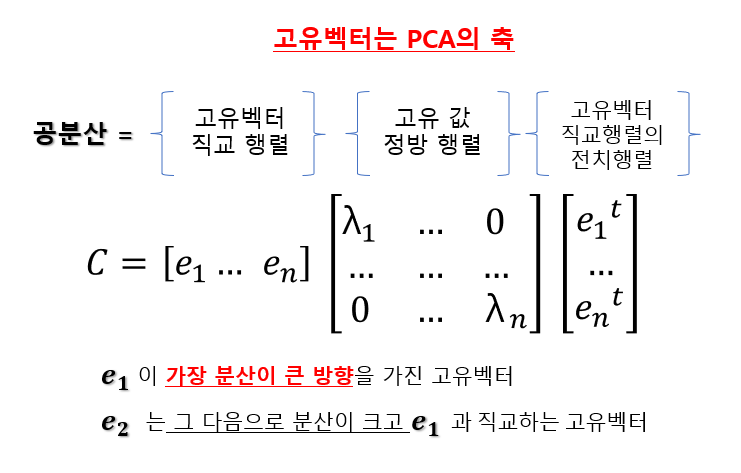
**고유 값(eigenvalue) :** 고유벡터의 크기를 나타내며, 동시에 **입력 데이터의 분산**을 나타냄

* **공분산 행렬**

공분산은 두 변수 간의 변동을 의미 // Cov(X, Y) >0 🡪 X가 증가할 때 Y도 증가함

테이블이(가) 표시된 사진

자동 생성된 설명



* **PCA변환**

입력 데이터의 **공분산 행렬이 고유벡터와 고유 값으로** 분해

* 고유벡터를 이용해 입력 데이터를 선형 변환하는 방식
* **PCA변환 수행 절차**

1. 입력 데이터 세트의 **공분산 행렬 생성**
2. 공분산 행렬의 **고유벡터와 고유 값 계산**
3. 고유 값이 가장 큰 순으로 K개(PCA 변환 차수)만큼 **고유벡터 추출**
4. 추출된 고유벡터를 이용해 새롭게 **입력 데이터 변환**

* **사이킷런 PCA**

***sklearn.decomposition.PCA (n\_components = None, copy = True, whiten = False, svd\_solver = ‘auto’, tol = 0.0, iterated\_power = ‘auto’, random\_state = None)***

n\_components : PCA 축의 개수 = 변환 차원

PCA 적용 전 개별 **feature 스케일링이 필수** **( 평균 0 분산 1인 표준 정규 분포로 변환 )**

전체 변동성에서 개별 PCA 컴포넌트 별로 차지하는 변동성 비율 🡪 ***explained\_variance\_ratio\_***