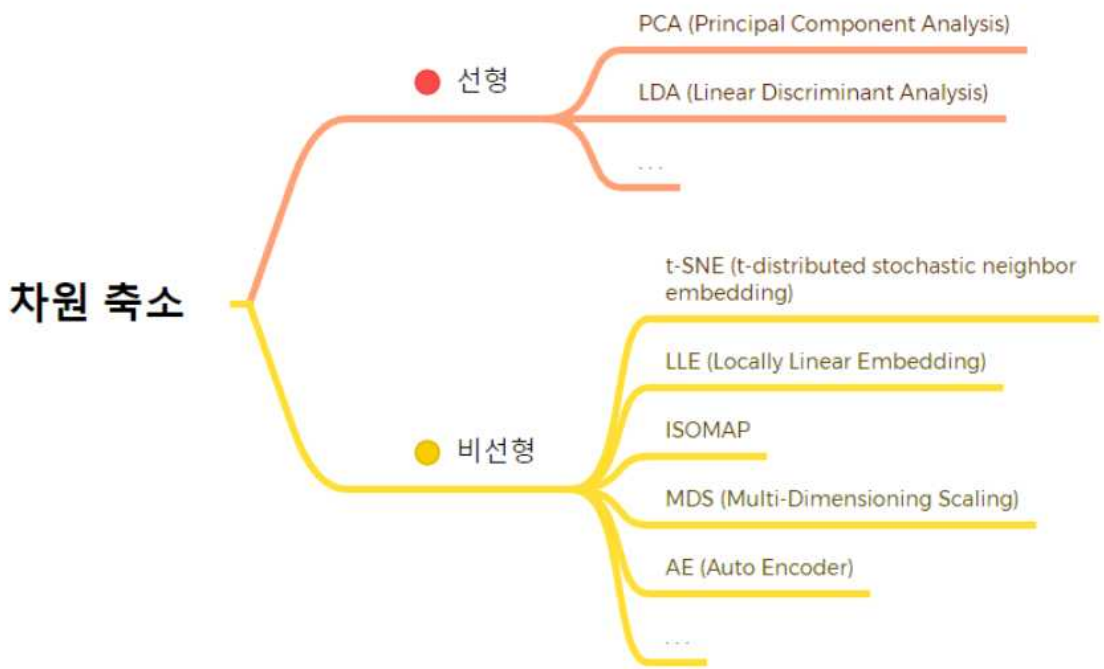
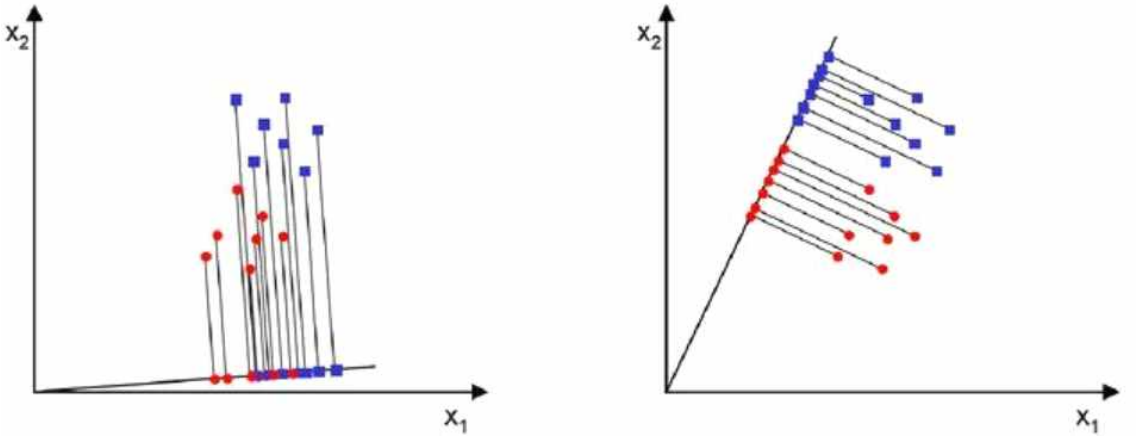


1. 주성분 분석(PCA)

주성분 분석(Principal Component Analysis, PCA)은 가장 널리 사용되는 차원 축소 기법 중 하나로, 원 데이터의 분포를 최대한 보존하면서 고차원 공간의 데이터들을 저차원 공간으로 변환한다. PCA는 기존의 변수를 조합하여 서로 연관성이 없는 새로운 변수, 즉 주성분(principal component, PC)들을 만들어 낸다. 첫 번째 주성분 PC1이 원 데이터의 분포를 가장 많이 보존하고, 두 번째 주성분 PC2가 그 다음으로 원 데이터의 분포를 많이 보존하는 식이다. 앞서 언급한 11차원의 데이터의 경우 기존의 변수들을 조합하여 같은 개수(11개)의 주성분을 만들 수 있는데, 만일 PC1, PC2, PC3가 원 데이터의 분포(성질)의 약 90%를 보존한다면, 10% 정도의 정보는 잃어버리더라도, 합리적인 분석에 큰 무리가 없으므로, PC1, PC2, PC3만 택하여 3차원 데이터로 차원을 줄일 수 있다.

2. 선형판별분석(Linear Discriminant Analysis, LDA)
PCA와 마찬가지로 축소 방법 중 하나다. (구글에 LDA라고 치면 토픽 모델링 기법인 Latent Dirichlet Allocation이 주로 나오는데 이와는 다른 개념이다.) LDA는 PCA와 유사하게 입력 데이터 세트를 저차원 공간으로 투영(project)해 차원을 축소하는 기법이지만, PCA와 다르게 LDA는 지도학습의 분류(Classification)에서 사용된다.



LDA는 바로 오른쪽과 같이 분류를 해주는 기법이다. 투영 후 두 클래스 간 분산은 최대한 크게 가져가고, 클래스 내부의 분산은 최대한 작게 가져가는 방식이다. 클래스 간 분산이 최대가 된다는 것은 각 클래스의 중심(평균)이 서로 멀어지도록 분류한다는 것이다. 클래스 내부의 분산이 작아진다는 것은 하나의 클래스끼리는 오밀조밀하게 뭉쳐있다는 뜻이다. 클래스 간 분산이 최대가 되고 클래스 내부 분산이 최소가 되면 [(클래스 간 분산) / (클래스 내부 분산)]은 최대가 된다. 다시 말하자면 LDA는 특정 공간상에서 클래스 분리를 최대화하는 축을 찾기 위해 클래스 간 분산(between-class scatter)과 클래스 내부 분산(within-class scatter)의 비율을 최대화하는 방식으로 차원을 축소한다.

3. Manifold Learning

고차원 데이터가 있을 때 고차원 데이터를 데이터 공간에 뿌리면 샘플들을 잘 아우르는 subspace가 있을 것이라 가정에서 학습을 진행하는 방법이다. Manifold Learning은 차원축소를 위해 사용하며 이를 통해 고차원 데이터를 저차원에서도 잘 표현하는 공간인 manifold를 찾아 차원을 축소시킨다.