6장 차원 숙소

실습 및 실습에 대한 설명은 .ipynb에 있습니다

01 차원 축소 개요

정의

- 매우 많은 피처로 구성된 다차원 데이터셋 → 차원 축소 → 새로운 차원의 데이터셋 생성
- 단순 데이터 압축이 아닌, 데이터를 잘 설명할 수 있는 잠재적인 요소 추출

차원 축소가 필요한 이유

- 차원이 증가 → 데이터 포인트 간 거리 증가 → sparse한 구조
- 수백 개 이상의 피처 → 예측 신뢰도 감소, 개별 피처 간에 상관관계가 높을 확률 큼
- 선형 모델은 입력 변수 간의 상관관계가 높을수록 다중 공선성 문제 → 예측 성능 저하

차원 축소 장점

- 직관적으로 데이터 해석 가능
- 학습데이터 크기 감소 → 학습에 필요한 처리 능력 감소

종류

- feature selection
  - 특정 피처에 종속성이 강한 불필요한 피처 제거
  - 데이터의 특징을 잘 나타내는 주요 피처 선택
  - 새롭게 추출된 중요 특성은 기존 피처와 완전 다른 값
- feature extraction
  - 피처를 함축적으로 잘 설명할 수 있는 또 다른 공간으로 매핑해 추출
  - 함축적인 특성 추출은 기존 피처가 인지하기 어려웠던 Latent Factor 추출

알고리즘

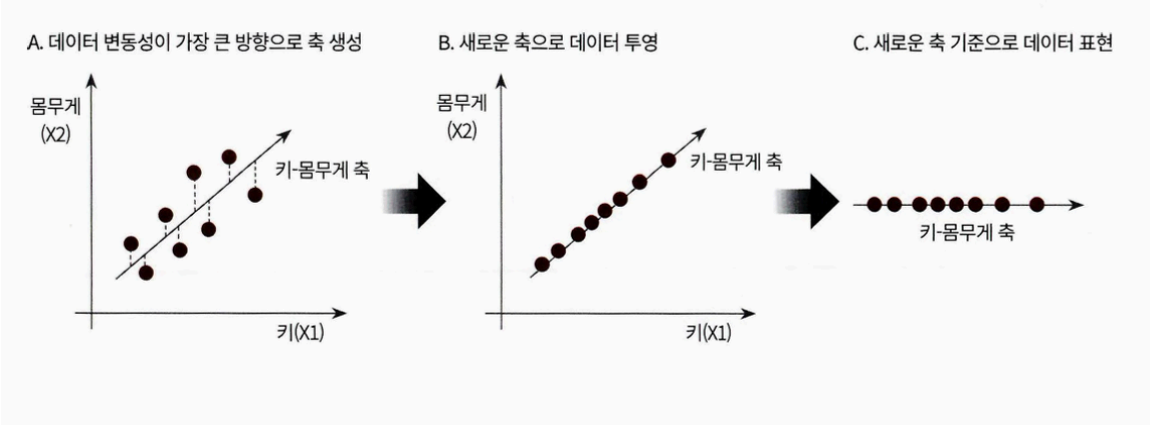
- 종류

    - PCA
    - SVD
    - NMF
- 이미지
    - 매우 많은 픽셀로 이뤄진 이미지 데이터 → 잠재된 특성을 피처로 도출 → 함축적 형태의 이미지 변환, 압축
    - overfitting 영향력이 작아져 예측 성능 상승
    - 차원 수 너무 많으면 약간의 픽셀 차이도 잘못된 예측을 유발할수 있어서 필요
- 텍스트
    - 텍스트 문서의 숨겨진 의미 추출
    - SVD, NMF가 시맨틱 토픽 모델링을 위한 기반 알고리즘으로 사용됨

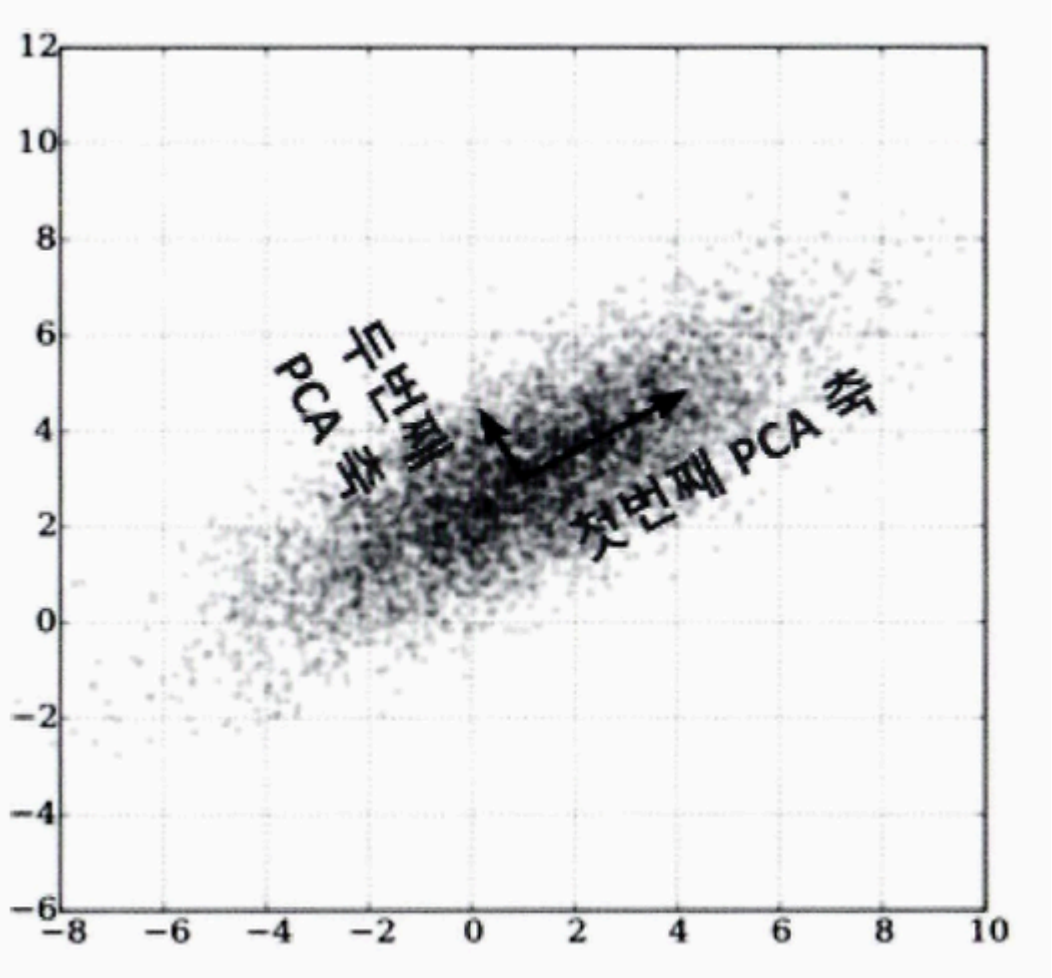
02 PCA

개요

- 여러 변수 간에 존재하는 상관관계를 이용해 이를 대표하는 주성분 (Principal Component)을 추출해 차원 축소
- 기준 데이터 정보 유실 최소화를 위해 가장 높은 분산을 가지는 데이터의 축으로 차원 축소 (PCA의 주성분)
    - 데이터 변동성이 가장 큰 방향으로 축 생성 → 새롭게 생성된 축으로 데이터 투영
    - 원본 데이터의 피처 개수에 비해 매우 작은 주성분으로 원본 데이터의 총 변동성 대부분 설명
    - 차원 축소 방법 (사진 참고)



생성된 벡터 축에 원본 데이터를 투영하면 벡터 축 개수만큼의 차원으로 원본 데이터가 차원 축소됨



선형 대수 관점

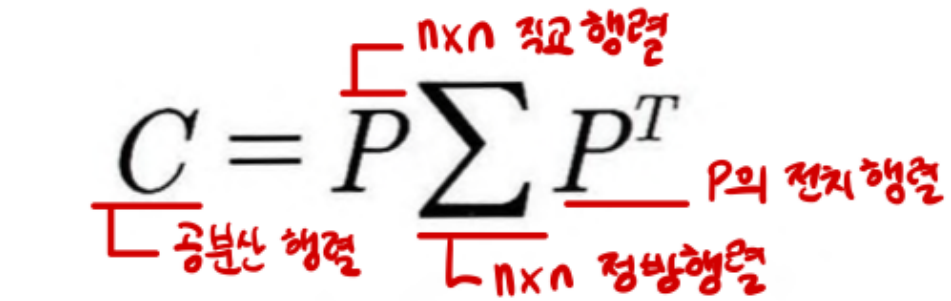
- 입력 데이터의 Covariance Matrix을 eigenvalue 분해 → eigen vector 입력 데이터를 선형 변환
- eigen vector: PCA의 주성분 벡터, 입력 데이터 분산이 큰 방향을 나타냄
- eigenvalue: 고유벡터의 크기, 입력 데이터의 분산
- 선형대수학 개념
  - 분산: 1개의 변수의 데이터 변동
  - 공분산: 두 변수 간의 변동
  - Cov(X, Y)>0: X가 증가할 때, Y도 증가한다
  - 공분산 행렬: 여러 변수와 관련된 공분산을 포함하는 정방형 행렬

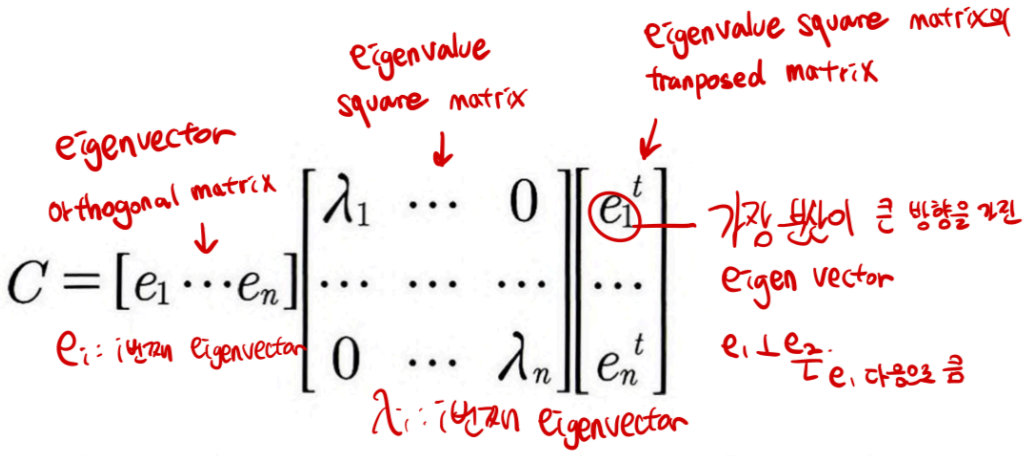
	X	Y	Z
X	3.0	-0.71	-0.24
Y	-0.71	4.5	0.28
Z	-0.24	0.28	0.91

가능한 모든 변수 쌍간의 공분산

분산

- 고유 벡터: 행렬 A를 곱해도 방향이 변하지 않고 크기(스칼라값)만 변하는 벡터. 행렬 분해에서 사용됨
  - 정방행렬 (Square Matrix): n x n
  - 대칭행렬 (Symmetric Matrix): 정방행렬 중에서 대각 원소를 중심으로 원소 값이 대칭되는 행렬. `A^T = A` 항상 eigen vector → orthogonal matrix, eigen value → square matrix 가능
- 공분산 행렬 분해





입력 데이터의 공분산 행렬이 eigen vector, eigen value로 분해 가능

PCA: 분해된 eigen vector 이용해 입력 데이터 선형 변환하는 방식

PCA 수행 step

1. 입력 데이터셋의 Covariance Matrix 생성
2. Covariance Matrix의 eigen vector, eigenvalue 생성
3. eigenvalue가 가장 큰 순으로 PCA 변환 차수 만큼의 eigen vector 추출
4. eigenvalue가 가장 큰 순으로 추출된 eigen vector를 이용해 새롭게 입력 데이터 변환

실습

- PCA는 여러 속성 값을 연산해야함 -> 속성 스케일에 영향 받음
  해결방법: PCA로 압축하기 전에 각 속성값들을동일한 스케일로 변환
- 사이킷런의 StandardScaler: 평균=0, 분산=1인 표준 정규 분포로 iris 데이터셋 속성 값 변환
- `PCA 클래스` : 생성 파라미터로 n_components(PCA로 변환할 차원 수) 입력받음
- fit(입력 데이터셋), transform(입력 데이터셋)을 호출해 PCA 변환 수행
- explained_variance_ratio: 전체 변동성에서 개별 PCA 컴포넌트별로 차지하는 변동성 비율 제공

활용 분야

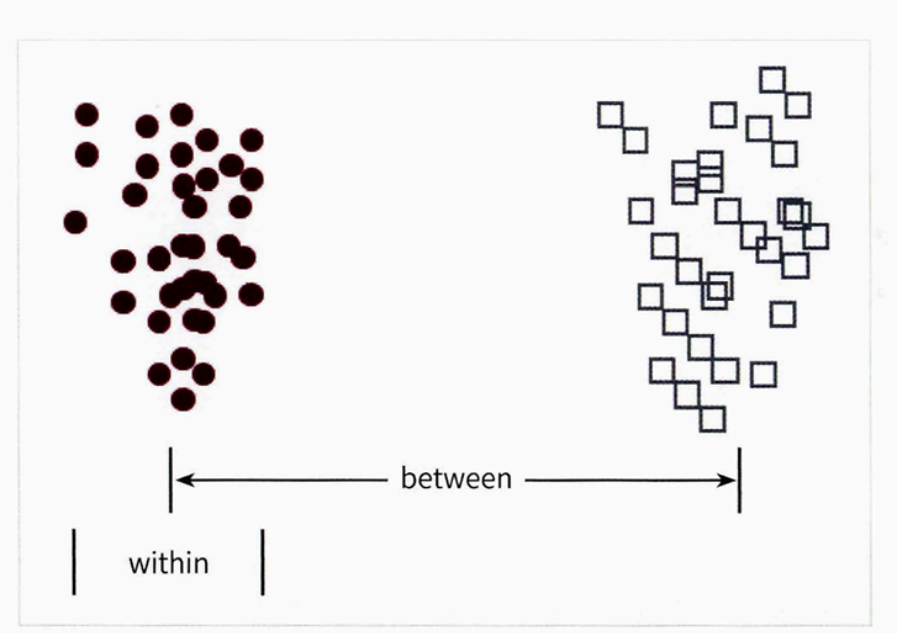
- 쉬운 데이터 인지
- 컴퓨터 비전; Eigen-face (얼굴 인식)

03 LDA

개요

- Linear Discriminant Analysis; 선형 판별 분석법
- 입력 데이터셋을 저차원 공간에 투영해 차원을 축소하는 기법 (PCA와의 공통점)
- Classification에서 사용하기 쉽도록 개별 class를 분별할 수 있는 기준을 최대한 유지하면서 차원 축소 (PCA와의 차이점)
  - 클래스 간 분산과 클래스 내부 분산의 비율을 최대한 방식으로 차원 축소 → 특정 공간상에서 클래스 분리를 최대화하는 축 찾기

- 클래스 간 분산: 크게
- 클래스 내부 분산: 작게



LDA Step

1. 클래스 내부와 클래스 간 분산 행렬 구함. 두 개의 행렬은 입력 데이터의 결정값 클래스별로 개별 피처의 mean vector를 기반으로 구함
2. 아래 식을 통해 두 행렬을 eigen vector로 분해

$$S_W^T S_B = [e_1 \ \cdots \ e_n] \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} e_1^T \\ \cdots \\ e_n^T \end{bmatrix}$$

클래스 간 분산행렬
클래스 내부 분산 행렬

3. eigenvalue가 가장 큰 순으로 LDA 변환 차수 만큼 추출
4. eigenvalue가 가장 큰 순으로 추출된 eigen vector를 이용해 새롭게 입력 데이터 변환

실습

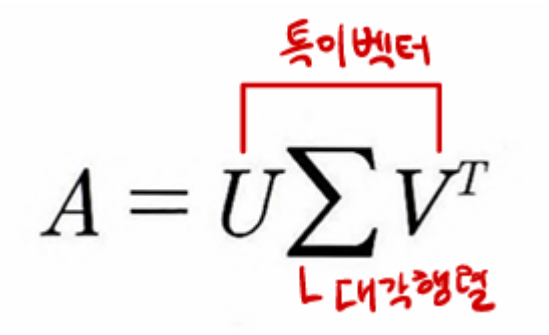
- 사이킷런은 LDA를 LinearDiscriminantAnalysis 클래스로 제공

- 유의할 점: supervised learning 이므로 클래스 결정값이 LDA 변환할 때 필요함

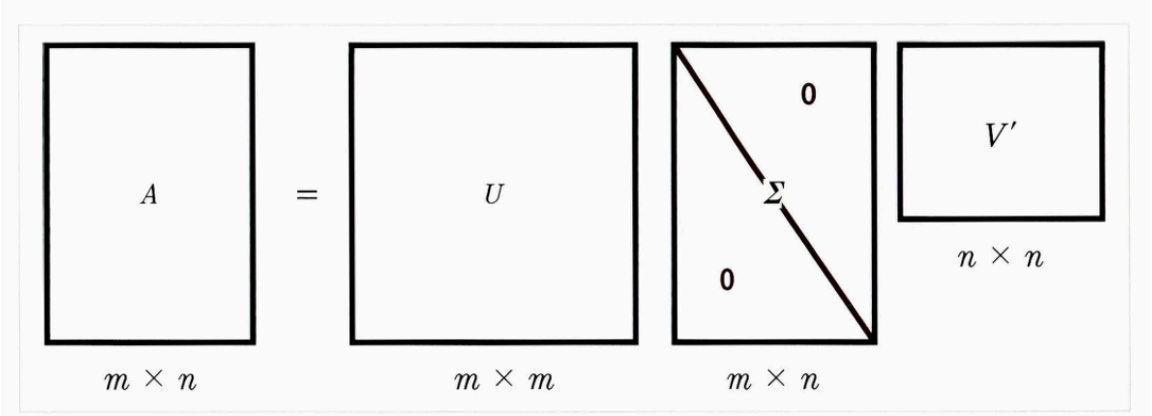
04 SVD

개요

- 정의: m x n 크기의 행렬 A를 아래와 같이 분해하는 것; 특이값 분해

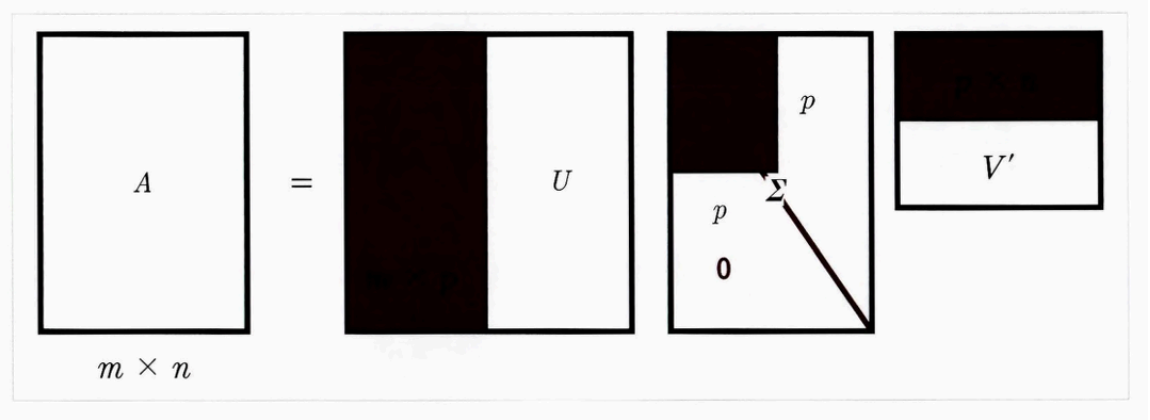


  - A의 특이값 = 대각행렬이 위치한 0이 아닌 값
- 정방행렬뿐만 아니라 행과 열의 크기가 다른 행렬에도 적용 가능
- 넘파이, 사이파이 라이브러리 이용
- 차원이 m x n일 때, U차원 m x m, 시그마 차원 m x n, V^T 차원 n x n으로 분해



- 일반적인 방법
  - 시그마의 비대각인 부분과 대각원소 중에서 특이값이 0인 부분 모두 제거
  - 제거된 시그마에 대응되는 U, V의 원소도 같이 제거해서 차원을 줄인 형태로 SVD 적용

⇒ A 차원 m x n, U차원 m x p, 시그마 차원 p x p, V^T 차원 p x n



- Truncated SVD
  1. 시그마의 대각원소 중에서 상위 몇개만 추출
  2. 1에 대응하는 U, V의 원소 제거
  3. 차원을 줄인 형태로 분해

실습

- 랜덤 행렬 생성 이유: 행렬의 개별 로우끼리의 의존성 없애기
- Truncated SVD는 상위 몇개만 추출해서 원본 행렬을 정확하게 다시 원복하는 것은 불가능

  하지만, 상당한 수준으로 원본 행렬 근사 가능 (원래 차원의 차수가 가깝게 truncate할수록)
- Truncated SVD는 사이파이에서만 지원됨.. `scipy.sparse.linalg.svds`
- PCA는 Dense Matrix에 대한 변환만 가능
- SVD는 Sparse Matrix에 대한 변환도 가

활용 분야

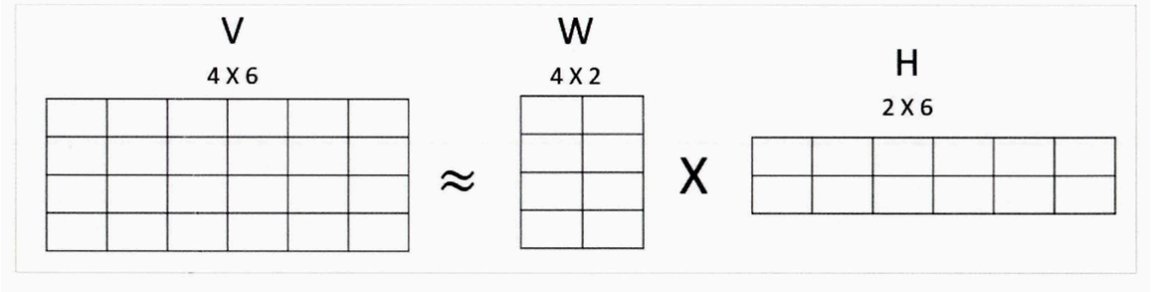
- 컴퓨터 비전 영역
  - 이미지 압축을 통한 패턴인식, 신호처리
- 텍스트의 토픽 모델링 기법 LSA

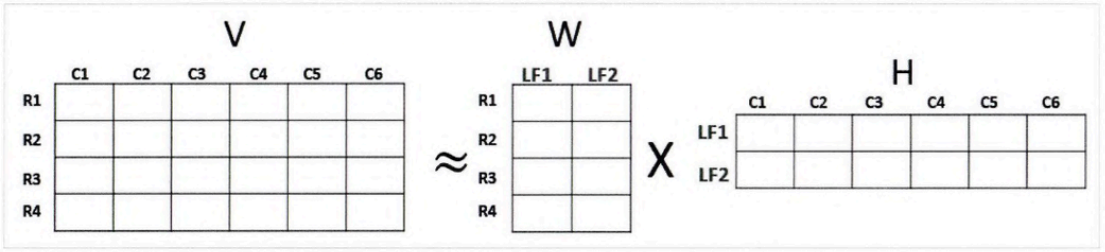
05 NMF

개요

- Non-Negative Matrix Factorization

- 낮은 랭크를 통한 행렬 근사 (Low-Rank Approximation)
- 원본 행렬의 모든 원소값이 모두 양수라는게 보장되면, 간단하게 2개의 양수 행렬로 분해하는 기법





  - W: 길고 가는 행렬 (원본 행렬의 행 크기와 같고 열 크기 보다 작은 행렬), 잠재 요소값이 얼마나 되었는가?
  - H: 작고 넓은 행렬 (원본 행렬의 행 크기보다 작고 열 크기와 같은 행렬), 잠자 요소가 원본 열(속성)로 어떻게 구성되었는가?
  - 분해된 행렬은 잠재 요소 특성을 가짐

활용 분야

- 패턴인식
- 텍스트 토픽 모델링 기법
- 문서 유사도
- 클러스터링
- 추천시스템; 잠재 요소(Latent Factoring) 기반의 추천 방식
  1. 사용자의 상품(영화) 평가 데이터 세트인 사용자-평가 순위(user-Rating) 데이터 세트를 행렬 분해 기법 통해 분해
  2. 사용자가 평가하지 않은 상품에 대한 잠재적인 요소를 추출
  3. 평가 순위(Rating)를 예측, 높은 순위로 예측된 상품을 추천