1. Inner Join과 Outer Join에 대해서 조사하고 이를 비교해봅시다.

\* Join 이란? 한 데이터베이스 내의 여러 테이블의 레코드를 조합하여 하나의 열로 표현한 것 /

Join을 하면 서로 의미 있는 데이터들끼리 집합이 형성되기 때문에 각 테이블끼리 관계를 가지게 되고, 효율적인 검색과 결과를 얻을 수 있다.

**- Inner Join (내부 조인)** : 가장 흔한 결합 방식이며 쉽게 말해 ‘교집합’ 이라고 할 수 있다.

**- Outer Join (외부 조인)** : Left, Right, Full 조인으로 나눌 수 있으며 Full 을 제외하고는 각 왼쪽 테이블, 오른쪽 테이블 기준으로 데이터를 보여준다. 쉽게 말해 ‘합집합’ 이라고 할 수 있다.

만약 왼쪽 테이블을 기준으로 한다고 할때, 오른쪽 테이블에 해당하는 값이 없다면 Null로 채워진다.

2. Python의 set\_trace기능을 살펴보고 이에 대한 예를 들어주세요.

\* set\_trace 란?

python 전용 디버거인 pdb 에 속한 기능이다. 크게 두 가지 방법이 있지만 더 흔한 방법은 pdb를 import 한 후, pdb.set\_trace() 를 중단하고 싶은 위치에 추가하는 것이다. 그러면 이 코드 윗 부분까지 실행된 후 해당 부분의 변수가 무엇인지, 어떤 값이 들어오는지 등을 알 수 있다.

예시) a와 b의 합과 곱 구하기

import pdb

→ pdb 모듈 불러오기

def add\_and\_multiply(a, b):

sum\_value = a + b

pdb.set\_trace()

product\_value = a \* b

return sum\_value, product\_value

→ add\_and\_multiply 라는 함수 정의, a와 b 두 개의 인자를 받는다.

함수 내부에서 a와 b의 합을 계산해 sum\_value 라는 변수에 저장한다.

pdb.set\_trace() 를 실행하면 프로그램이 이 시점에서 멈추고 디버거 모드로 들어간다.

(디버거 모드에서는 변수 값을 확인하거나 코드를 한 줄씩 실행할 수 있다.)

디버거에서 나와 a와 b의 곱을 계산해 prdouct\_value에 저장한다.

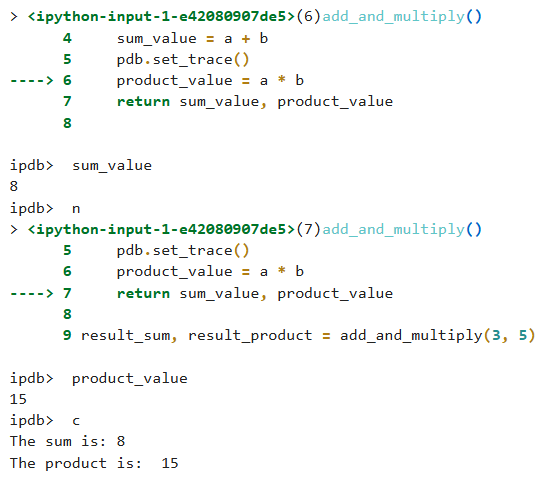
result\_sum, result\_product = add\_and\_multiply(3, 5)

print("The sum is:", result\_sum)

print("The product is: ", result\_product)

→ add\_and\_multiply 함수에 a=3, b=5 를 인자로 전달한다.

함수가 실행되면서 해당되는 값들이 result\_sum, result\_product에 저장된다.



3. https://github.com/lumiamitie/TIL/blob/master/python/recsys\_in\_python\_101.md

실제 분석하고 해당 코드를 수행해봅니다.

=> 협업필터링 vs 컨텐츠 기반 모형을 비교 설명해주세요. 그리고 이를 합친 하이브리드모형과 각각 성능면에서 비교해주세요

**- 협업필터링**

유사한 사용자가 좋아하거나 평가한 항목을 추천하는 방식

메모리 기반 :

→ 사용자 기반 필터링 : 유사한 취향을 가진 사용자를 찾아 그 사용자가 좋아하는 항목 추천

→ 아이템 기반 필터링 : 사용자가 이전에 좋아한 아이템과 유사한 아이템 찾아 추천

모형 기반

\* 장점 = 도메인 지식 불필요 / 자동으로 사용자의 관심 반영 가능

\* 단점 = 콜드 스타트 문제 (새로운 사용자나 아이템에 대한 정보 부족할 때, 추천하기 어려운 상황) / 희소성 문제 (사용자가 항목에 대한 평가를 적게 했을 경우 정확도 낮아질 가능성)

**- 컨텐츠 기반 모형**

사용자가 선호한 아이템의 특성을 분석하여 유사한 특성을 가진 다른 아이템 추천하는 방식

주로 아이템의 메타데이터(영화의 장르, 감독, 배우 정보 등) 을 활용

\* 장점 = 콜드 스타트 문제에 강함 / 사용자의 명시적 선호도에 대한 개인화된 추천 가능

\* 단점 = 도메인 지식 필요, 아이템의 특성 잘 정의 / 새로운 유형의 아이템 추천이 어려울 수 있다.

**- 하이브리드 모형**

협업 필터링 + 컨텐츠 기반 필터링

→ 가중 평균 모델 : 협업 필터링과 컨텐츠 기반 필터링의 결과를 가중 평균으로 결합

→ 스위칭 모델 : 상황에 따라 더 적합한 방식 선택해 사용

→ 혼합 모델 : 협업 필터링과 컨텐츠 기반 필터링의 결과를 동시에 보여줌

**- 결론 -**

협업 필터링은 사용자 데이터가 풍부하고, 사용자가 유사한 취향을 가진 그룹을 형성할 때 효과적

컨텐츠 기반 필터링은 특정 항목에 대한 데이터만으로 추천이 가능하고, 사용자 취향이 명확히 드러날 때 효과적

하이브리드 모형은 두 방식을 결합하여 개별 모델의 단점을 보완하므로, 추천 시스템의 전반적인 성능을 높이는 데 유리. 특히 콜드 스타트 문제를 해결하고, 다양한 사용자와 항목에 대해 안정적인 추천을 제공.