Context 태깅 모델 &
개인-정책 추천 모델 구축

코드스테이츠 Project 2 - 기업 협업 with 웰로

AI_06_강지호
AI_06_이남준

01. 프로젝트 개요

01. 프로젝트 개요 기업소개

웰로

- 사용자: 흩어진 정책/혜택을 찾아주는 **개인화 정책 추천**-신청 솔루션 제공
- 기업/기관: 맞춤형 타겟알림, 신청 오퍼레이션, 수요 및 설문조사 솔루션 제공

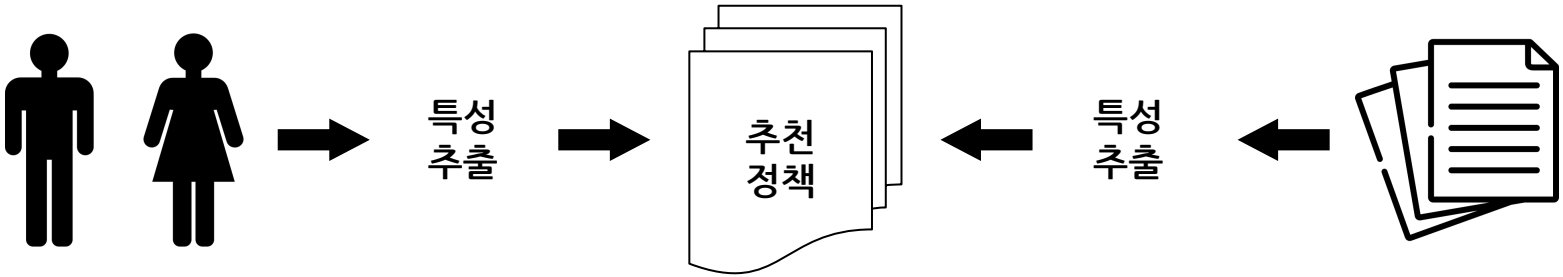
01. 프로젝트 개요 배경 및 목적

- **배경**

  대한민국에는 수많은 정책이 있음에도, 정책 대상자들은 해당 정책의 존재를 모른다.
  기관 입장에서는 정책 대상자를 선별하여 홍보하는 과정이 번거롭다.

- **목적**

  정책 공고문 context에서 정책 특성을 자동으로 추출하고, 이를 토대로 전국의 정책/지원 사업을 유저의 프로필 특성에 맞게 추천해준다.



02. 프로젝트 진행

02. 프로젝트 진행 데이터 소개

추천 모델

- **유저 데이터 2만개**

성별, 나이, 거주지역(시도, 시군구), 관심지역(시도, 시군구), 학력, 직장, 가구원 유형, 결혼, 자녀, 자녀 수, 자녀 정보, 특수상황, 관심상황특성, 장애 상황, 보훈대상 상황, 예정 상황, 소득 정보, 관심 정책

↓ 태깅 모델

13개) 성별, 나이, 시도, 시군구, 학력, 직장, 가구원 유형, 결혼, 자녀, 자녀상세, 대상특성, 관심상황특성, 중위소득, 관심정책

- **정책 데이터 8.8만개**

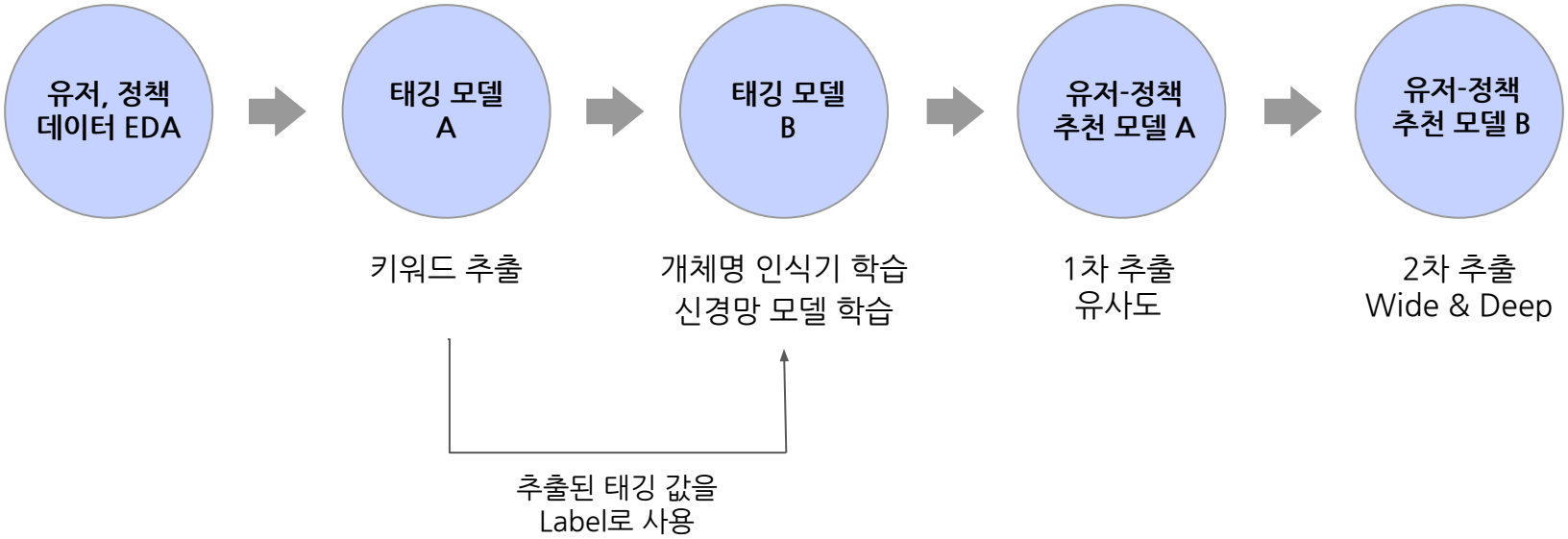
정책ID, 정책서비스ID, 서비스명, 소관기관, 소관기관유형, 생애주기, 신청절차, 선정기준, 지원유형, 서비스목적, 지원내용, 지원대상

↓ 태깅 모델

20개) 소관기관유형, 지원유형, 신청절차, 성별, 대상연령시작, 대상연령끝, 시도, 시군구, 학력, 직장, 직장상세, 가구원 유형, 결혼, 자녀, 자녀상세, 대상특성, 대상특성상세, 관심상황특성, 중위소득, 관심정책

02. 프로젝트 진행 프로세스

- 기간: 4주
- 비고: 총 2가지 모델 (태깅을 위한 자연어처리 모델, 유저-정책 매칭을 위한 추천시스템 모델)을 만들어야 한다. 서비스화 이전이라고 가정하므로 각 모델에 대한 라벨값이 없다. (답지는 존재하지만 성능 평가로만 활용)



02. 프로젝트 진행 - 태깅 모델

02. 프로젝트 진행 태깅 모델

- **진행방향**

예시1.
○서비스명 : 국민취업지원제도
○선정기준 :
(I유형) 중위소득 50% 이하, 재산 3억 이하이면서, 최근 2년 이내 100일 또는 800시간 이상의 경험이 있는 분
＊취업경험이 없는 비경활, 청년은 예산 범위 내 선발지원 (청년은 소득요건 120% 이하)
(II유형) I유형에 해당하지 않는 가구단위 중위소득 100% 이하 (청년은 모두 무관)

=> 성별: 무관 | 직장 : 구직자/실업자 | 중위소득 : 중위소득 100% 이하 | ……

예시2.
○서비스명 : 의료급여수급자 만 6세 미만 영유아 건강검진 지원
○선정기준: 만 6세 미만 영유아(의료급여수급권자)

=> 성별: 무관 | 나이 : 0~6세 | 자녀: 있음 | ……

- **고려사항**

- context가 복잡하고 길다. -> feature별로 태깅을 추출해야 한다.
- 자연어처리 모델을 학습하기 위한 **라벨값이 없음.**

02. 프로젝트 진행 태깅 모델

모델 A. 키워드 추출 (진행완료)

라벨값이 없는 문제를 해결할 수 있으며, 베이스라인모델이 될 수 있음.

- **방법**

  feature별로 카테고리의 class를 대표하는 키워드를 설정하고, 해당 키워드가 n개 이상 존재하면 class 추출

  ex)

  1. '대상특성' feature에는 19개의 class가 있다. 그 중 '농축수산인' class를 대표하는 키워드를 설정한다.
  2. 하나의 정책의 context에 해당 키워드들이 n개 이상 포함될 경우 '농축수산인'이라고 태그를 달기로 한다. (n개를 정하는 기준) 답지에서 '농축수산인' class의 context에 해당 키워드가 몇 개 들어있는지 파악한다.

  ㄴ> 답지를 성능 평가로만 활용해야 하기 때문에 적절하지 못한 방향

```
# 농축수산인
cha_words = ['농업', '농축', '어업' '원예', '축산', '임업']
if sum(text.count(x) for x in cha_words) >= 2:
    cha.append('농축수산인')
```

02. 프로젝트 진행 태깅 모델

모델 A. 키워드 추출 (진행완료)

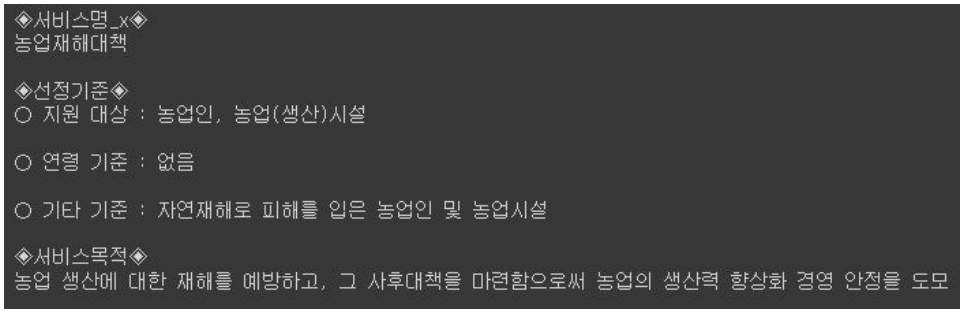
라벨값이 없는 문제를 해결할 수 있으며, 베이스라인모델이 될 수 있음.

- **방법**

  feature별로 카테고리의 class를 대표하는 키워드를 추출하고, 해당 키워드가 N개 이상 존재하면 태깅 추출

  ex)

  1. '대상특성' feature에는 19개의 class가 있다. 그 중 '농축수산인' class를 대표하는 키워드를 설정한다.
  2. 하나의 정책의 context에 해당 키워드들이 n개 이상 포함될 경우 '농축수산인'이라고 태그를 달기로 한다. (n개를 정하는 기준) 답지에서 '농축수산인' class의 context에 해당 키워드가 몇 개 들어있는지 파악한다.



→ 대상특성:농축수산인 으로 태깅됨

02. 프로젝트 진행 태깅 모델

모델 A. 키워드 추출 (진행완료)

- **결과**

답지와 비교한 피처별 태깅을 정확하게 맞춘 비율

지역(시도), 지역(시군구)	99%, 83%
대상특성, 대상특성상세	42%, 68%
관심상황특성	65%
소관기관유형	99%
지원유형	14%
신청절차	32%

성별	76%
학력	65%
가구원	81%
결혼	63%
자녀, 자녀상세	80%, 82%
직장, 직장상세	64%, 0.4%
대상연령 시작, 끝	54%, 70%

* 정확도 낮은 요인
  a. 답지의 답이 틀린 경우가 다수 존재
  b. 하위태그(~상세)일수록 답지의 오답률 ↑
  c. 정확하게 일치해야 정답
     ex. 정답: 영유아,성인 → 오답
         모델: 성인,영유아

* 정확도가 높은 요인
  a. null값이 다수인 피처도 존재

- **보완점**

결과값이 모델B의 **라벨**로 사용되기 때문에 모델A의 **정확도**가 매우 중요하다.
태깅이 쉬운 피처는 정확도가 높게 나왔으나, 복잡한 context를 가진 피처는 정확도가 낮게 나왔다.
정확도를 높이기 위해 **알고리즘을 보완**해야 할 필요가 있음.

모델 B. 개체명 인식기 학습 & BERT 모델 (진행중)

- ## 개체명 인식기 (Named Entity Recognition, NER)

  이름을 가진 개체를 인식하는 모델

  ex. 지호와 남준이는 4주동안 웰로와 함께 프로젝트 진행했다. 점심에 샐러드도 같이 먹었다.
  => 지호(사람)와 남준(사람)이는 4주(시간)동안 웰로(조직)와 함께 프로젝트를 진행했다. 점심(시간)에 샐러드(음식)도 같이 먹었다.

- ## BERT (Bidirectional Encoder Representations from Transformers)

  구글에서 공개한 사전 훈련된 자연어처리 신경망 구조

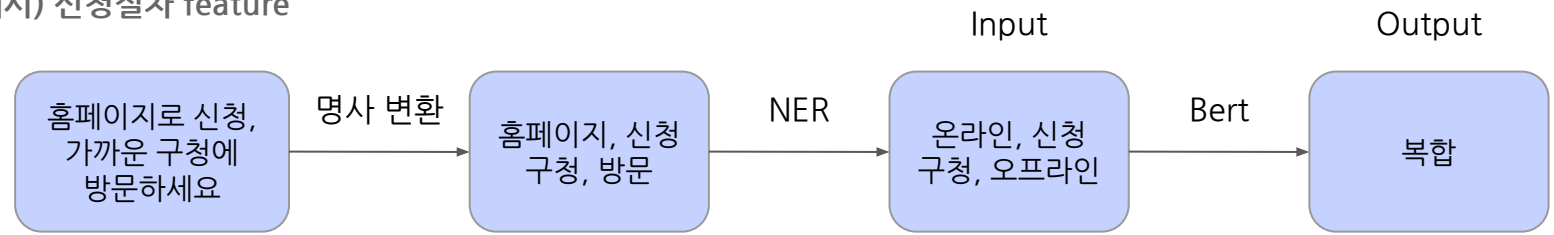
  장점1. 문장이 양방향 학습이 가능해 문맥 파악에 용이하다.
  장점2. 대용량의 코퍼스로 학습한 pretrained 모델이기 때문에 목적에 따라 파인튜닝에 용이하다

모델 B. 개체명 인식기 학습 & BERT 모델 (진행중)

- **가설**
  1. 모델 A에서 feature별로 설정한 키워드로 **개체명 인식기(NER)를 학습**시킬 수 있다.
  2. 학습된 NER로 정책 context 속에서 **키워드를 찾아 특정 문자로 치환**한 모델이 그렇지 않은 모델보다 성능이 더 좋을 것이다.
  3. 모델 A에서 추출한 태깅을 label로 주어 BERT모델 학습을 통해 **복잡한 정책 context에서 태깅 추출**이 가능할 것이다.

- **방법**
  1. 형태소 분석기 (mecab 등)을 사용해서 **명사로 변환**
  2. **학습된 NER**를 이용해 **키워드를 치환**
  3. 모델 A에서 추출한 태깅값을 label로 사용
  4. Bert 모델 학습 진행

  예시) 신청절차 feature



02. 프로젝트 진행 - 추천 모델

02. 프로젝트 진행 추천 모델

- **진행방향**

  예시.
  ○유저 정보
  - 성별: 여성
  - 나이: 20대 중반
  - 지역: 경기도, 파주시
  - 학력: 대학(원) 졸업
  - 직장: 구직자/실업자
  - 가구원 유형: 무주택 세대원
    …
  - 관심 정책: 취업 지원

  => 구직자이므로 취업 지원과 관련된 정책, 무주택자이므로 주택 청약 등의 정책, 경기도 정책, …

- **고려사항**

  unique 정책이 8.8만개이며, 모델의 서비스화를 생각한다면 정책의 갯수는 증가.
  정책 데이터가 많기 때문에 1차 모델에서 **추천 정책 후보군**을 추린 다음, 2차 모델에서 추천해야 함.
  유저가 평가한 정책에 대한 데이터**(라벨값)이 없음**.

02. 프로젝트 진행 추천 모델

모델 A. 유사도 (진행완료)

- **방법**
  유저와 정책 데이터에 공통적인 피처를 추출, 모든 피처를 One-Hot 으로 만들어서 (유저*피처 matrix) x (피처*정책 matrix)의 곱을 진행 - 각 피처마다 가중치를 줄 수 있음.

  예시) 범위를 넓게 잡을 경우
  : 추천 정책의 갯수 多, 1차 모델로 사용 가능

	여성	남성	자녀0	자녀X	구직자/실업자
A	1	0	0	1	1
B	0	1	0	1	1
C	1	1	1	0	1

X

	취업 지원	양육비 지원	여성 안심귀가
여성	0	0	1
남성	0	0	0
자녀 0	0	1	0
자녀 X	0	0	0
구직자/실업자	1	0	0

=

	취업 지원	양육비 지원	여성 안심귀가	
A	1	0	1	2
B	1	0	0	1
C	1	1	1	3

02. 프로젝트 진행 추천 모델

모델 A. 유사도 (진행완료)

- **방법**
  유저와 정책 데이터에 **공통적인 피처**를 추출, 모든 피처를 One-Hot 으로 만들어서 **(유저*피처 matrix) x (피처*정책 matrix)의 곱**을 진행 - 각 피처마다 가중치를 줄 수 있음.

  예시) 범위를 좁게 잡을 경우
  : 추천 정책의 갯수 小, 2차 모델로 사용 가능

	여성	남성	자녀0	자녀X	구직자/실업자
A	1	0	0	1	1
B	0	1	0	1	1
C	1	1	1	0	1

X

	취업 지원	양육비 지원	여성 안심귀가
여성	1	1	1
남성	1	1	0
자녀 0	1	1	1
자녀 X	1	0	1
구직자/실업자	1	0	0

=

	취업 지원	양육비 지원	여성 안심귀가	
A	3	1	2	1
B	3	1	1	1
C	3	3	2	2

02. 프로젝트 진행 추천 모델

모델 A. 유사도 (진행완료)

- 결과

유저 정보

성별: 여성
나이: 27
시도: 경기도
시군구: 파주시
학력: 대학(원)졸업
직장: 구직자/실업자
가구원:
결혼: 미혼
자녀: 없음
자녀상세: None
중위소득: 중위소득 40% 이하,
중위소득 40 ~ 60% 사이
관심정책: 취업 지원

추천된 정책

직업훈련생계비 대부
국민취업지원제도
구직촉진수당
청년 미취업자 대학생에 대한
무료법률구조
직업훈련생계비 융자신청
글로벌 현장학습 프로그램

- 보완점

유저 정보와 정책 context의 질에 따라 다른 모델을 구축해야 한다.
ex) 유저가 프로필을 가득 채운 경우 더 specific한 모델로, 정보의 질이 좋지 않을 경우 general한 모델 나누기

02. 프로젝트 진행 추천 모델

모델 B. Wide and Deep model (진행중)

- **선정이유**
  정책 도메인 특성상 추천될 정책과 유저의 조건이 부합하는 것이 중요하면서도, 새로운 정책을 추천할 수 있어야 한다. Wide part는 linear한 부분으로, 정책과 유저의 특성이 정확히 **일치**하는 부분에 대해 **기억**한다. 이 경우 상세화된 예측 결과를 제시하기 때문에 **과적합**이 발생할 수 있다. 이를 해결하기 위해 Deep part가 존재한다. Deep part는 non-linear한 부분으로, 유저와 정책 특성을 **일반화**하여 추천한다. 이 경우 과적합을 방지할 수 있어 유저에게 **새로운 정책**을 추천할 수 있다.

- **가설**
  1. Wide and Deep 모델은 Wide part에서 유저와 정책의 정보가 일치(부합)하면서도 Deep part를 통해 더 다양한 정책을 제시할 수 있다.
  2-1. 1차 모델이 유사도기반이라면, 1차 모델 후 추려진 후보군 중에서 wide&deep model을 통해 가장 적합한 정책을 추천해줄 수 있다.
  2-2. 1차 모델로 Wide and Deep을 사용한다면 다양한 후보군을 추릴 수 있다. (2차 모델: 유사도)

02. 프로젝트 진행 추천 모델

모델 B. Wide and Deep (진행중)

- **방법**
  - Label 생성
    a. 유저의 '관심정책' feature 사용하여 1,0 binary하게 생성

〈유저 데이터의 '관심정책'〉

num	mb_15
11726	
21745	취업 지원,근로자 지원,문화생활 지원,주택-부동산 지원
13327	
23542	의료 지원,보육지원(만0~7세),성인교육지원,개인금융지원,기업금융지원,근로자 지원,문화생활 지원,주택-부동산 지원
14442	교육지원(만8~19세)
10640	교육지원(만8~19세)
23936	교육지원(만8~19세)
9083	의료 지원,성인교육지원,개인금융지원,창업 지원,취업 지원,근로자 지원,문화생활 지원,주택-부동산 지원

〈'관심정책' feature의 11개 클래스〉

근로자 지원, 취업 지원, 개인금융지원, 문화생활 지원,
주택-부동산 지원, 교육지원(만8~19세), 성인교육지원,
의료 지원, 창업 지원, 기업금융지원, 보육지원(만0~7세)

↓

정책 데이터에 해당되는 클래스 기입

↓

label: 유저와 정책이 공통된 클래스를 가지면 1, 아니면 0

02. 프로젝트 진행 추천 모델

모델 B. Wide and Deep (진행중)

- **방법**

  - Label 생성

    a. 유저의 '관심정책' feature 사용하여 1,0 binary하게 생성
    단점: 1. '관심정책' feature는 모델 성능 개선에 기여할 수 있지만, label값으로 사용됨
    2. 정책마다 라벨 생성을 위한 라벨링을 새로 만들어야 하므로 공수가 많음

    b. 유저와 정책의 공통된 feature 에서 겹치는게 n개 이상일 경우 1, 아닐 경우 0 binary하게 생성
    단점: 1. feature가 겹친다고 해서 유저가 원하는 정책인지는 의문

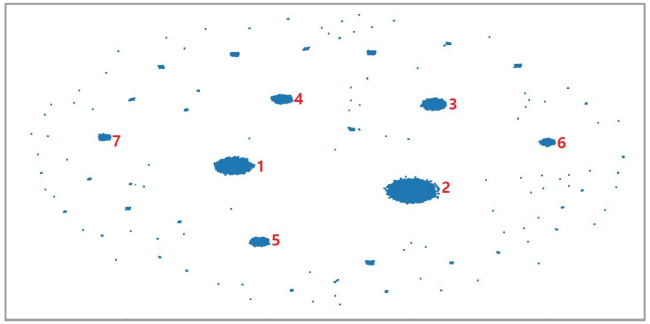
  - Wide part(cross product transformation) & deep part input 진행 및 학습

02. 프로젝트 진행 추천 모델

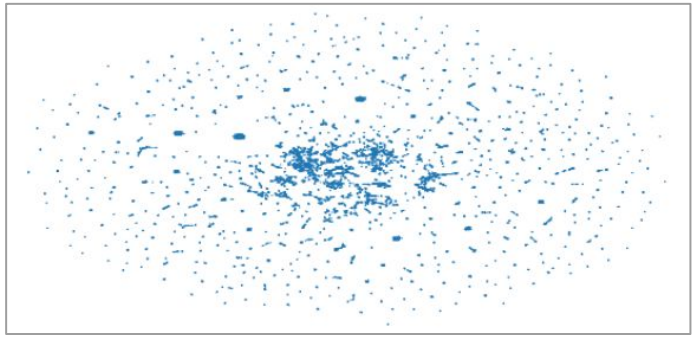
모델 B. Wide and Deep (진행중)

- **이슈**
  훈련을 위한 train set: (유저x정책)의 형태. (유저2만개 * 정책8.8만개 = 8억8천만 샘플)
  학습량이 많기 때문에 유저 데이터와 정책 데이터 모두 **다운샘플링** 선행이 필요하며, 유저와 정책의 feature 갯수가 많기 때문에 다양한 방법에 대한 고려가 필요함.

- **What we've tried**
  클러스터링으로 군집을 만들고, 해당 군집에서 대표 n개를 추출하는 방식으로 다운샘플링을 시도.
  => 피쳐 갯수 4개로 클러스터링을 진행할 경우 군집화가 잘 되었으나, 모든 피처를 사용하면 군집화가 어려움



<4개의 feature만 사용>



<모든 feature를 사용>

03. 프로젝트 회고

03. 프로젝트 회고

- ## 딥러닝 모델 환경 구축의 문제

  프로젝트 데이터에 맞는 딥러닝 모델을 선정하기 위해 다양한 모델을 공부했으나, 이론적으로 모델에 대해 아는 것과 실제 데이터로 모델을 다루는 것은 많이 다르다는 것을 깨달음. 모델의 Input 데이터에 맞춰 우리의 데이터를 엔지니어링하는 부분에서 실패와 보완을 반복함.

- ## 강지호

  추천시스템에 관심이 있었고, RecSys와 NLP는 함께 공부해야 한다 생각해서 시작한 프로젝트였는데, 모델을 개념적으로 아는 것과 직접 데이터를 가공하여 모델에 접목시키는 것은 많이 다르다는 것을 깨달았다. 다양한 모델을 더 깊이 알고, 간단한 데이터로 모델을 돌려본 경험이 있어야 비로소 raw 데이터에 맞는 모델을 빠르게 선정할 수 있고, 여러 모델을 비교하면서 추천시스템을 구축할 수 있다는 생각이 들었다.

- ## 이남준

  추천과 자연어처리에 관심을 두고 다양한 구현을 해봤음에도 실무 데이터를 이용해 실제 활용하는 것이 어렵다는 것을 느낄 수 있었다. 데이터 활용과 모델링의 깊이있는 공부 및 사용 경험이 적어 시간이 부족했고, 그로인해 원했던 딥러닝 모델을 적용하지 못한 것이 아쉽다. 모델들의 발전을 좀더 깊이있게 공부하면서 이론상에서 놓쳤던 부분을 재숙지하고, 이 모델을 서비스에서 어떻게 응용할 수 있을지 더 많이 고민해야겠다.

감사합니다.