[Recsys 8조 EXIT] Final Project Wrap Up Report

1.1 프로젝트 개요

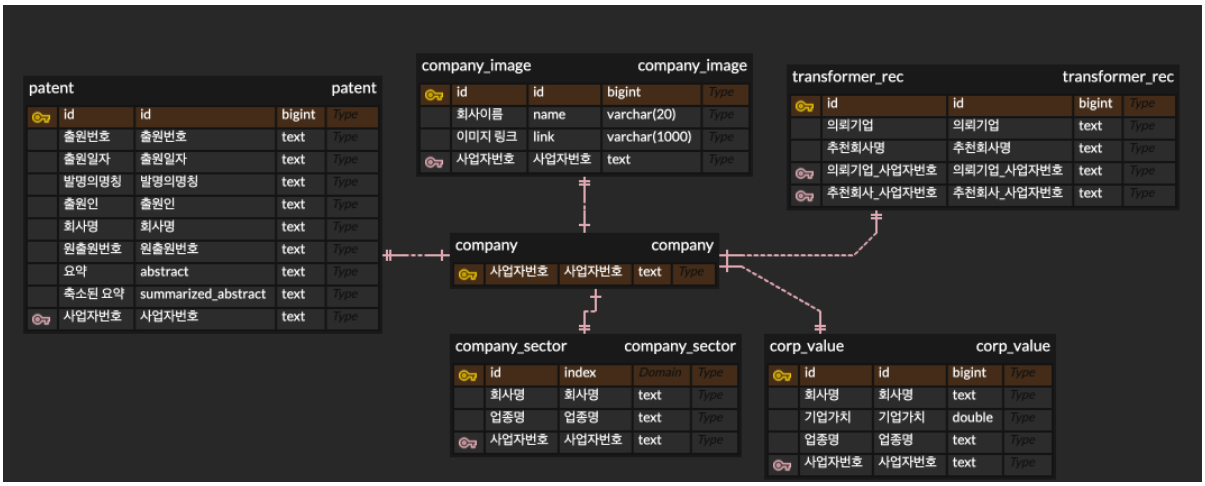
프로젝트명

> EXIT - 특허 기반 전략적 기업 파트너 추천

프로젝트 주제

기술적 성장을 도모하는 기업의 사업자 번호를 입력받아 `특허 정보`를 이용해 `전략적 파트너 기업`을 추천하는 프로젝트

데이터 개요 및 구조



데이터 설명	데이터 활용 Task	데이터 개수	출처
특허 데이터	`Summarization`, `Top-K Recommendation`	500만건	키프리스(https://www.kipris.or.kr/)
기업 관련 데이터	`Enterprise-Valuation`, `Retrieval`	100,000건	Dart(https://dart.fss.or.kr/)
Interaction Data	`Top-K Recommendation`	747건	Dart(https://dart.fss.or.kr/)
벤처 기업 관련 데이터	`Enterprise-Valuation`, `Retrieval`	30,000건	SMES(https://www.smes.go.kr/venturein/pbntc/searchVntrCmp)
상장 기업 업종 데이터	`Retrieval`	10,000건	네이버증권(https://finance.naver.com/)
상장 기업 주가 데이터	`Enterprise-Valuation`	10,000건	KRX(https://github.com/sharebook-kr/pykrx)

데이터 설명	데이터 활용 Task	데이터 개수	출처
비상장 기업 관련 데이터	`Enterprise-Valuation`	150건	KOTC(https://www.k-otc.or.kr/)
논문 요약 Data	`Summarization`	170,000건	AI 허브(https://aihub.or.kr/)

실험 환경 및 사용한 툴

- (팀 구성 및 컴퓨팅 환경) 5인 1팀, 인당 V100 서버를 VSCode와 SSH로 연결하여 사용
- (협업 환경) GitHub, Notion
- (의사 소통) Slack, Zoom

1.2 프로젝트 팀 구성 및 역할

김지우_T5063	데이터 크롤링 및 전처리, Top-K Recommendation
박수현_T5085	Product Manager, Business Embedding
석예림_T5110	데이터 크롤링 및 전처리, Top-K Recommendation
임소영_T5172	프로젝트 기획 및 제안, Enterprise Valuation, Streamlit 개발
전증원_T5185	Enterprise Valuation, Database 구축

1.3 프로젝트 수행 절차 및 방법

1.3.1 Business Embedding

Summarization

모델

- Pre-trained Model → SKT-AI KoBART
- Fine Tuning → 논문요약 데이터(AI HUB)

모델 선정 이유

- Summarization Task 특성상 Seq2Seq 구조가 요구되기 때문에 Encoder + Decoder 형태의 BART가 적절하다고 판단함
- SKT-AI KoBART를 통해 Summarization을 구현한 사례가 많아 초심자 입장에서 참고할만한 사이트가 많았음
- 다양한 데이터로 사전 학습된 모델들을 제공하여 실험을 통해 취사선택 하고자 했음
    - Example> kobart-base-v1 / kobart-base-v2 / kobart-summarization ⋯

Measure

- ROUGE-L
    - 다음 단계인 Sentence Embedding이 이루어지기 때문에 Output의 Syntax가 정확한 것보다는 Input과 Output 간에 Semantic information이 잘 보존 되는 것이 중요한 상황이었음
    - 이러한 Semantic Information을 잘 보존하는지를 평가할 수 있는 지표로 ROUGE라고 생각했으며, 문장을 구성하는 단어의 순서까지 반영한 ROUGE-L을 선정하게 되었음

프로젝트 수행결과

Version	Performance (Rouge-L)	Application
4	0.6109	Project Start
6	0.6089	(),[]를 통해 부연 설명되는 전문 용어 제거
7	0.6238	0.2 <= ratio_tokens = (output_tokens / input_tokens) <= 0.5 인 경우만 사용
8	0.6245	Pre-trained Model 변경(kobart-base-v2 -> kobart-summarization)

Sentence Embedding

모델

- Pre-trained Model : SKT-simCSE KoBERT
- Fine-Tuning : X

모델 선정 이유

- 프로젝트 기획 초기부터 해당 태스크에는 Fine-Tuning을 고려하지 않았기 때문에 사전 학습으로 부터 표현력이 적절할 것으로 기대되는 Contrastive Learning based Model을 선정하고자 했음
- 몇가지 데이터를 샘플링하여 임베딩의 벡터를 확인해본 결과 표현력이 준수했음
- DiffCSE와 simCSE를 Measure를 기준으로 비교한 결과 simCSE가 성능이 더 좋았음

Measure

- Cosine Similarity
  - 임베딩 벡터 간의 거리보다는 방향성이 중요했음
  - 앞의 단계인 Summarization에서 사용했던 Measure와 Align 되는 경향이 있었음

프로젝트 수행결과

Version (= Summarization)	BERT-based simCSE	RoBERTa-based simCSE
4	0.6705	0.6838
6	0.6715	0.6847
7	0.6743	0.6881
8	0.6745	0.6876

Retrieval

수행내용

- 다음 과정인 Enterprise Valuation의 사용 목적으로, 1개의 비상장회사를 기술적 관련있는 K개의 상장회사들과 매핑함

구현방식

- 특허가 있는 상장/비상장 기업들의 경우 Summarization에 의해 생성된 축소된 요약 문장, 특허가 없는 비상장 회사의 경우 벤처중소기업청에서 크롤링한 주생산품, 특허가 없는 상장 회사의 경우 네이버 증권에서 크롤링한 테마 & 업종을 사용하여 Sentence Embedding을 생성함

- 특허를 여러개 보유하고 있는 상장/비상장 회사의 경우 clusteroid를 통해 해당 기업을 대표할 수 있는 특허 1개를 선정해 Sentence Embedding을 진행함
- Cosine Similarity를 기준으로 임베딩 벡터들의 유사성을 통해 1개의 바상장회사에 K개의 상장회사들과 매핑을 진행함

진행결과

- 100물류(물류 솔루션 및 물류창고 관리업) ↔ [선광, 한익스프레스, 동방, 상일, 인터지스, 태웅로직스, CJ대한통운 ⋯]
- 힘멜(사무용 가구) ↔ [현대리바트, 오하임아이엔티, 진영, 퍼시스, 태양, 시디즈, 대상 ⋯]

1.3.2 Enterprise Valuation

현금 흐름 할인법 (DCF)

1️⃣ 기업의 beta 계수 구하기

Input

- 시장 월별 종가 (2012 ~ 2022 KOSPI / KOSDAQ)
- 해당 시장에 상장된 기업의 월별 종가 (2012 ~ 2022)

Model

Linear Regression

Output

Weight = 상장기업의 beta 계수 (Levered Beta)

2️⃣ 기업의 할인율 구하기

Rule base

비상장기업의 `Unlevered Beta` 를 구함

- Levered Beta : 위 Retrieval 에서 매핑된 `비상장1 : 상장K` 을 기준으로, 상장 K 개의 Levered Beta 값의 평균값
- Tax Rate : 비상장기업의 매출액에 따른 법인세율
- Debt / Equity : 비상장기업의 재무제표에 따른 부채총계, 자본총계

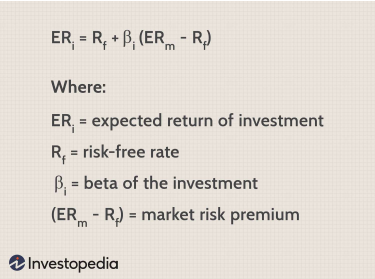
Wall Street Prep

$$\text{Levered Beta} = \text{Unlevered Beta} \times [1 + (1 - \text{Tax Rate}) \times (\frac{\text{Debt}}{\text{Equity}})]$$

$$\text{Unlevered Beta} = \frac{\text{Levered Beta}}{[1 + (1 - \text{Tax Rate}) \times (\frac{\text{Debt}}{\text{Equity}})]}$$

Rule base : CAPM

비상장기업의 할인율 `ER_i` 을 구함



- Risk free rate : 국채 이자율, `3.5` (2023.07 기준)
- beta of the investemnt : 위에서 구한 기업의 `Unlevered Beta`
- ER_m : 시장의 연평균 성장률

3️⃣ 기업의 성장률 예측

Process

상장기업의 3개년치 재무제표의 성장률로 학습된 모델에, 비상장기업의 3개년치 재무제표를 넣어 성장률을 계산

Input

상장기업의 3개년치 재무제표의 8가지 지표 (매출액, 부채총계, 자본총계, 당기순이익, 영업이익, 현금 및 현금성 자산, 감가상각비) 의 성장률

Model

`RNN` / `LSTM`

: sequence 길이가 짧았기에, Long Term memory 에 대한 추가 연산이 필요하지 않다고 생각해 선정

Output

비상장기업의 8가지 지표 (매출액, 부채총계, 자본총계, 당기순이익, 영업이익, 현금 및 현금성 자산, 감가상각비) 의 성장률

> 프로젝트 수행 결과는 하단 `주가 regression` 에 기재함.

4️⃣ 기업의 가치 예측

Rule base

1. 가장 최근 연도의 실제 금액에 예상 성장률을 곱해 예상 금액을 구함
2. 예상 매출액을 기준으로 법인세를 매김
3. 예상 세후 영업이익 = 예상 영업이익 * (1 - 법인세)
4. 예상 현금흐름 = (예상 세후 영업이익 + 예상 감가상각비) * (기존 EBITDA 배율)
5. 예상 기업가치 = 예상 현금흐름 / CAPM

주가 regression

모델

- `LGBM`

모델 선정 이유

- 데이터의 수가 1300개 정도로 적다고 생각되어 일반적으로 우수한 성능이 보장되는 부스팅 계열 모델을 사용함

구현 방식

- **모델**: LightGBM 라이브러리를 활용함

Measure

- **loss 함수**: 주식 가격은 이상치가 존재하기 때문에 이상치에 강한 RMSE를 loss 함수로 사용함
- **평가 지표**: 주식의 가격 예측을 직관적으로 파악할 수 있는 MAE를 사용하여 평가함

data preprocessing

- **train(상장기업), test(비상장기업) 데이터 불일치**: 비상장회사와 상장회사의 매출액을 비교한 후 비상장회사의 가장 큰 매출액보다 큰 상장회사 데이터를 제거함
- **skewness**: `Box-Cox 변환`을 통해 `skewness` 10.105 → 0.235

프로젝트 수행 결과

설명	validation RMSE	kotc MAE(장외 비상장 주식 30개의 주가총액)	기타
시장가치법	Rule base이기 때문에 loss가 존재하지 않음	90,510,037,323 (905억)	
base LGBM	652,619,000,000 (6500억)	272,791,384,272 (2700억)	여러 모델 (`linear_regression`, `ridge`, `lasso`, `elasticnet`, `decision_tree`, `random_forest`, `xgb`, `lgbm`) 중 실험을 통해 가장 성능이 좋게 나온 lgbm을 선정
train, test 데이터 불일치 해결	153026000000 (1500억)	105,017,135,914 (1050억)	

설명	validation RMSE	kotc MAE(장외 비상장 주식 30개의 주가총액)	기타
skewness 해결했을때	5.64659	74,535,555,470 (740억)	Box-Cox 변환을 통해 skewness를 해결, loss는 데이터가 변환되어 줄어듦
optuna 최적화	9.733409515218751	71,294,112,261 (712억)	
현금할인법	-	77,589,265,668 (775억)	`RNN`
	-	70,147,243,268 (701억)	`LSTM` (volume estimation 적용한 retrieval, top 20)
	-	70,121,800,399 (701억)	`LSTM` (volume estimation 적용하지 않은 retrieval, top 5)
앙상블(LGBM + 현금할인법)	-	61431246864 (614억)	

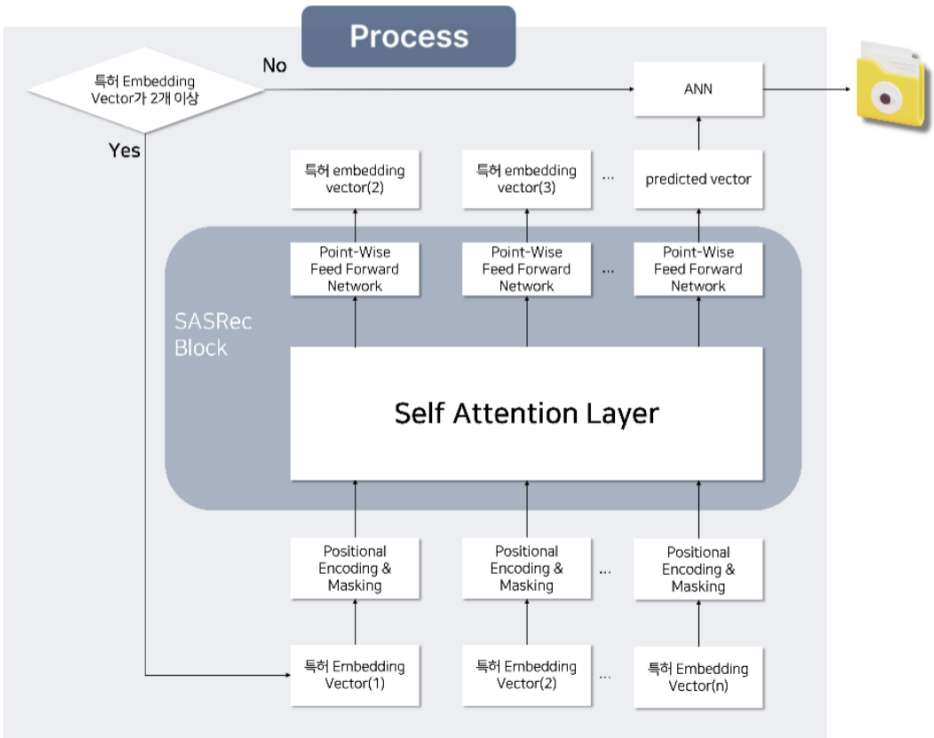
1.3.3 Top-K Recommendation

Retrieval based on Patent Similarity

특허를 가진 기업들의 특허 요약 문장 embedding vector를 Input으로 사용하여 의뢰 기업이 보유한 특허들의 순차적인 패턴을 학습하고, 의뢰기업이 출원할 다음 특허 벡터를 예측함

모델

`SASRec` `ANN`



SASRec

모델 선정 이유

> 💡 먼저, Sequential한 데이터를 Input으로 사용하기 위해 Transformer 모델을 고려했으며, Transformer 모델 중에 masked attention을 사용한 Decoder 구조를 가진 모델을 고려해야함.
> 동시에, 한 회사에 대해서 순차적으로 출원된 특허의 경향성을 파악하여 다음 특허 벡터를 예측하는 Task에 적합한 Sequential Recommendation 계열의 SASRec 모델을 사용하기로 결정함.

구현 방식

- Dataset, Dataloader, Model : PyTorch로 직접 구현함

모델 Custom

> 💡 **Loss Function Custom**
> 먼저, 기존에 사용되던 loss function의 logit을 커스터마이징 진행함.
> 기존의 logit은 sigmoid 함수를 사용하여 0과 1사이로 범위를 고정해 주지만, SASRec의 모델의 출력으로 얻은 임베딩 벡터의 차원이 너무 크기 때문에 로짓이 0 또는 1로 고정되어 제대로 학습되지 않는 문제가 발생함.
> 따라서 0에서 1사이의 값을 가지도록 cosine similarity를 shift하고 scaling한 shifted cosine similarity와 angular distance를 각각 logit으로 정의하고 학습을 진행함.

평가 지표

> 💡 Measure : **Interaction이 발생한 의뢰기업의 Data 로 예측한 특허 벡터와 Interaction이 일어난 상대 회사의 특허 벡터 간의 유사도**
>
> 의뢰 기업의 특허를 기반으로 예측한 벡터와 유사한 특허를 가진 기업을 찾아서 추천하는 것이 Task이기 때문에, 위에서 정의한 유사도가 높을 경우, interaction이 발생할 가능성이 높은 회사를 추천한 것이라고 판단함.

수행 결과

설명	실험	성능 변화
Loss Function 정의	shifted cosine similarity	0.7206
	angular distance	0.7178
Max Sequence Length	512	0.7175
	1024	0.7227

ANN

선정 이유

- Candidate Generation이 필요한 이유
  - 특허를 가지고 있는 기업의 Corpus가 너무나 크기 때문에 확장성 있는 추천을 하기 위해 도입함
  - 모든 기업에 대해서 순위를 매기는 것보다 후보 생성 네트워크에서 추천할 기업의 수를 줄이고 랭킹 네트워크에서 순위를 매기는 것이 훨씬 빠름

- ANNOY의 장점
  - 빠른 검색 속도
    - 빠른 검색 속도를 가지고 있어 대용량의 데이터셋에서도 빠르게 근사적인 최근접 이웃을 찾음
  - 효율적인 메모리 사용
    - 데이터를 메모리 기반으로 처리하기 때문에 메모리 사용량을 효율적으로 관리함
  - 고차원 데이터 처리
    - 데이터를 저차원으로 투영하여 고차원 데이터에서도 효과적으로 작동함
  - index와 vector 정보만으로 동작
    - index - value 형태로 데이터베이스에 따로 저장을 해두고 활용 가능함

구현 방식

- ANN 알고리즘을 이용하여 예측한 다음 벡터와 유사한 벡터를 검색하여 유사한 벡터 n개 추출함

Measure

- 유사한 벡터 100개를 검색하는 속도 : `0.7187 sec`

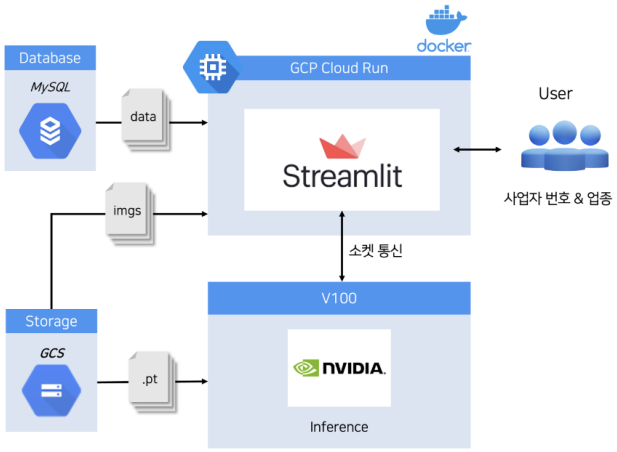
Ordering & Filtering

Retrieval의 결과에 대해서, Ordering과 Filtering을 통해 단순히 다음 특허 벡터와의 유사도가 높은 순으로 기업을 추천하는 것이 아닌 자체적인 Scoring 체계를 가지고 대상 기업의 조건을 한번 검토한 후 최종적인 추천 결과를 제공하고자 함.

구현 방식

- SASRec과 ANN을 통해 나온 특허 리스트를 대상으로 ordering을 진행함
- 특허 리스트 내에 기업 등장 빈도수를 기준으로 ordering을 하고, 동일 빈도수에 대해 더 높은 유사도를 지닌 특허를 보유한 순으로 정렬함
- 의뢰 기업과의 기업 가치 비교를 통해 필터링된 파트너 기업 추천을 제공함
- 기업 가치에 대한 정보가 없거나, 의뢰 기업보다 기업 가치가 낮은 경우 제거함
- 필터링을 거친 이후, 의뢰 기업에게 추천할 최종 K개의 기업을 선정함

1.3.4 Serving



Streamlit & Docker

- User 와 직접 맞닿아있는 부분으로, 사업자 번호와 선호하는 업종을 받아 추천 결과를 띄워줌
- mui module 을 이용하여 UI 면을 개선하였음
- cache 를 사용하여, 이전 추천 결과는 빠르게 로딩할 수 있도록 만듦
- MySQL 과 연결되어, 특허 데이터를 불러옴
- GCS 와 연결되어, 기업 이미지 데이터를 불러와 띄워줌

Google Cloud Platform

- Google Cloud Run
  - docker container image 를 dockerize 하여 cloud 상에서 동작하도록 함
- Google Cloud Storage
  - 기업의 로고 이미지를 담고 있음
- Google MySQL
  - 기업 추천에 사용할 메타데이터를 담고 있음

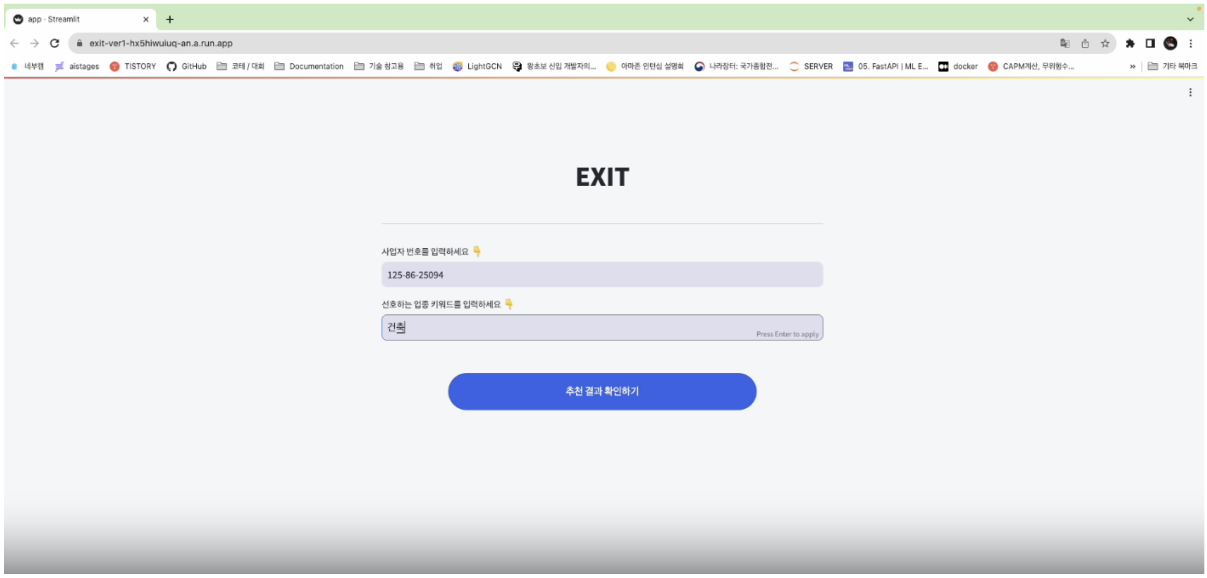
V100

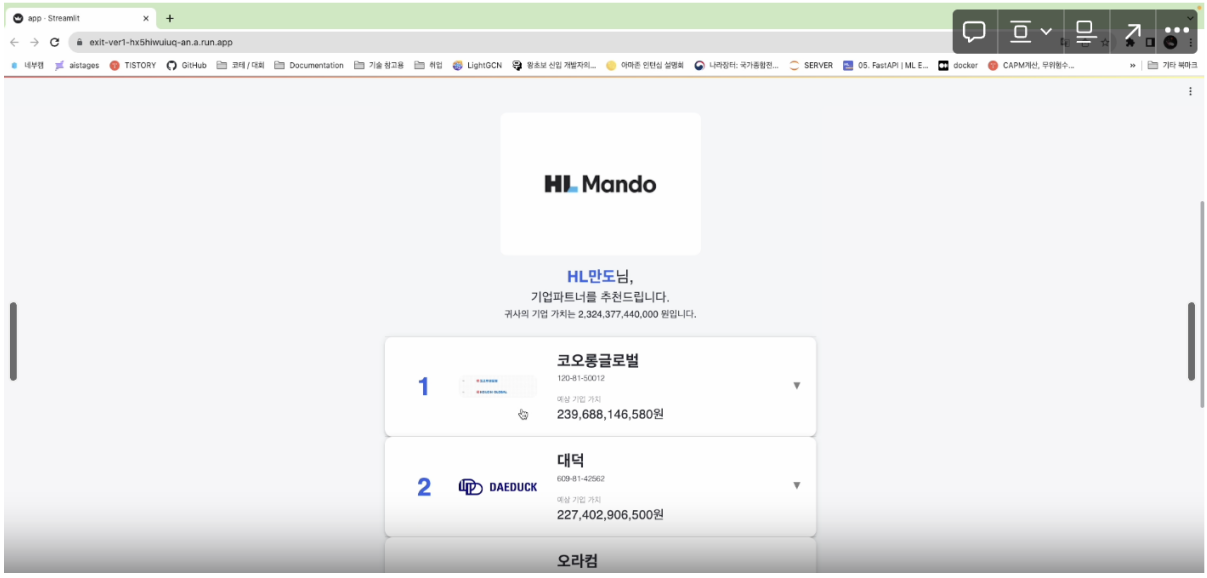
- Inference 속도를 개선할 수 있도록 V100의 대용량 램 및 GPU를 활용하고자 Deploy 서버와 V100서버 간에 소켓통신을 구현하였으며, Inference는 V100에서 이루어질 수 있도록 하였음

1.4 프로젝트 수행 결과

Streamlit

https://exit-ver1-hx5hiwuiuq-an.a.run.app/





1.5 자체 평가 의견

👍 칭찬할 점

백엔드 고도화	1차 제출 시점에서 인지했던 `Latency가 큰 점을 개선`하고자 Inference가 실시간으로 이루어질 수 있도록 변경함.
도메인 지식 습득	부족한 도메인 지식을 채우기 위해 다방면으로 노력함(현직자 인터뷰, 관련 서적, 유튜브).
독창성	여러 모델의 결합을 통해 하나의 플로우를 완성함.
	프로젝트 선례가 없었음에도 자체적인 방법을 만들어 문제를 해결함.
개발 외	이활석 마스터님께 중간피드백을 받고 프로젝트를 더 나은 방향으로 발전시킴.
	각자 모델 개발을 하나씩 맡아 부스트캠프의 취지를 살림.
	팀원들의 문제점을 함께 고민하고 해결방안을 찾음.

😢 한계점

백엔드 통신 속도	Streamlit에서 CloudSQL에 있는 특허 정보를 가져오는 속도가 여전히 느려 Latency가 큰 문제 발생함.
데이터 소통	Task의 Input/Output을 명확히 하지 않아 (인터페이스 표준화를 하지 않아) 협업과정에서 어려움을 겪음.
	각 데이터 소스나 시스템에서 사용하는 기업 식별자가 다양하여, 특정 기업을 유일하게 식별하는 공통된 ID가 없었음. 이로 인해 데이터 간의 일관된 매칭이 어려워졌고, 서로 다른 ID 형식을 사용하는 데이터들을 효과적으로 통합하는 데 어려움이 발생함.
Top-K Recommendation	Interaction 데이터의 부족으로 인하여 직접적인 추천 평가 지표(Recall@K 등) 측정이 어려움.
Enterprise Valuation	수집할 수 있는 비상장 기업의 과거 데이터에 한계가 있어서 Sequential한 패턴을 충분히 학습하지 못함.
	재무제표의 정보만으로는 주가를 표현하지 못하여 예측의 정확도가 떨어짐.
시간 관리	시간 관리를 잘못해서 점점 급하게 프로젝트를 진행함.

🛠️ 개선할 점

Cold Start 대응	사업자 번호만을 받는 구조를 개선해야 함.
	업종을 통해 popular 추천을 제공해야 함.
모델 최신화	빠른 속도로 특허가 출원되기에 정기적인 웹스크래핑의 구현을 통해 모델 배치학습을 시도하려 함.
사용자 편의성	UI/UX 를 개선해야 함.

2.1 개인회고

김지우_T5063

- **나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?**
  - 프로젝트의 End-to-End를 경험하기 위해, 기획부터 데이터 수집, 모델링 파트에 참여했으며, 서비스 파이프라인을 어떻게 구성하면 좋을지 함께 고민했다.
  - 수집된 데이터를 바탕으로 의뢰 기업에게 대상 기업을 추천해주는 프로세스를 완성시킬수 있도록, 전체적인 프로세스를 구성하고, 코드를 작성했다.
- **내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?**
  - 기획부터 데이터 수집, 모델링까지 참여하면서 어떠한 목표가 있고, 해당 목표를 달성하기 위해 세운 계획들이 한번에 착착 진행되는 경우는 거의 없다는 것을 깨달았다. 생각했던 것과 데이터의 질이나 Task의 난이도가 달라서 추가적인 전처리가 많이 필요하거나 불가피하게 계획을 수정해야하는 경우가 많았다. 그래서 더더욱 MVP 모델을 빠르게 뽑고, 프로젝트 사이클 한바퀴를 돌리는 것이 중요하다는 것을 깨달았다.
  - 그동안은 ML/DL 프로젝트라고 하면 모델링에 국한되고 직접적으로 유저에게 전달되는 End-to-End를 경험해보지 못했는데, 이번에 프로젝트를 통해 Latency 문제가 중요하고, 그 문제를 해결하기 위해서 여러 방면에서의 고민이 필요하다는 것을 알게 되었다. 데이터 전처리나 inference 방식 등 데이터나 모델 단에서 최대한 속도를 고려하여 효율적으로 코드를 짜는 게 중요하다는 것을 다시 한번 느끼게 되었다.
- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**
  - Interaction 데이터의 부족으로 인하여 직접적인 추천 평가 지표(Recall@K) 측정이 어려웠다. 평가 지표를 선정하긴 했지만 일반적인 지표가 아니어서 아쉬웠다.
  - 처음에 설계한 파이프라인을 많이 수정하고, Airflow 및 CI/CD를 적용해보지 못했다.
  - SASRec 모델의 성능을 고도화하기 위한 시도를 많이 해보지 못한 점이 아쉽다.

- **한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?**
  - Airflow를 공부하여, 주기적인 Batch 작업(데이터 크롤링 및 모델 학습)을 자동화 해보고 싶다.
  - 안정된 배포를 위해 CI/CD를 적용해보고 싶다.
  - Interaction의 종류를 M&A뿐만 아니라, 다양한 형태(기술적 협약)를 고려하여 추가적으로 수집하여 직접적인 추천 평가 지표를 측정하고 싶다.
  - SASRec Model의 성능을 고도화하기 위해 이런 저런 실험을 해보고 싶다. 또한 여러가지 모델을 추가적으로 고민하여 모델끼리의 성능 비교도 진행해보고 싶다.

박수현_T5085

- **나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?**
  - RecSys 도메인 이외에 NLP 관련 지식 습득 및 해당 지식을 활용해 더욱더 풍부한 예측을 이끌어 내길 희망했다. 그래서 문장 또는 문서에 대해 전처리 하는 방법 및 모델 트렌드를 빠르게 습득해 이번 프로젝트에 적용할 수 있었다.
  - 개인적인 경쟁력을 높이고자 Product Serving을 주도적으로 진행했다. 서비스 파이프라인 구현 중에 Inference 속도를 높이고자 자주 불리는 데이터의 경우 사전에 RAM에 로딩 및 GPU를 활용하고자 대용량 램 및 GPU를 보유하고 있는 V100을 사용할 수 있도록 소켓 통신을 통해 해당 플로우를 완성시켰다.
- **내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?**
  - 이번 프로젝트에서 진행한 태스크들이 시간순으로 엮어있던 만큼, 빠르게 MVP 모델을 개발해서 다음 태스크를 개발해내는 팀원에게 Output을 제공해야 했음을 깨달을 수 있었다.
  - 서비스 파이프라인 구축을 위해 전체 태스크들을 통합하는 과정에서 각 태스크들의 Input/Output을 명확히 할 뿐만 아니라 Type 또한 명확히 해야하는 것을 깨달을 수 있었다.
- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**
  - Git을 제대로 활용하지 않았으며, 그래서 CI/CD를 구현하지 못했다.
  - 무엇보다 Airflow를 통해 Data Engineering 경험을 꼭 해보고 싶었는데, 시간적인 문제로 이뤄내지 못해 아쉽다.
  - LLM의 Fine Tuning에 대한 노하우가 부족해 전처리에만 매몰되어서 아쉬웠다.
  - 랩업리포트를 작성 중에 논리적인 오류가 존재하는 것을 알게 되었다.
- **한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?**
  - LLM의 Fine Tuning의 경우 Pre-trained Model의 Structure를 수정하거나 Vocab의 unused token를 활용하는 것과 같은 다양한 방법을 적용해보고 싶다.
  - 이번 프로젝트를 통해 Data Storage와 Database가 따로 존재하는지를 깨달을 수 있었으며, Data Warehouse 또한 구축해서 차별점 공부해보고 싶다.
  - Git에 조금 더 익숙해져보고 싶으며, 이를 통해 CI/CD 또한 경험해보고 싶다.
  - Airflow를 경험해 Data Engineering에 대한 지식도 넓혀보고 싶다.

석예림_T5110

- **나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?**
  - 데이터 크롤링 부터 전처리, 모델링까지 전체적인 프로세스 구성하고, 이를 구현해 보며 하나의 서비스를 제작했다.

    - 일반적인 성능을 내지 못하는 Task에서 적절한 지표를 만들어 평가하기 위해 고민하고, 부족하지만 모델의 성능을 확인할 수 있는 지표를 제작했다.
- **내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?**
    - 항상 주어진 데이터로 진행했었는데, 크롤링 후 진행한 데이터 전처리에서 예상했던 데이터의 품질과 달리 Raw한 데이터들은 품질이 매우 떨어지는 것을 알게 되었다.
    - 프로젝트 Task에 맞는 모델, loss함수 등 고민하면서 최적의 모델을 만드려고 고민하였다. 그 사이에서 loss 함수를 custom 하는 방법을 배울 수 있었다.
- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**
    - 성능을 측정하기 어려웠기 때문에 Interaction 데이터가 부족한 점이 가장 큰 아쉬웠다.
    - 짧은 프로젝트 기간이다 보니 빠르게 프로토타입을 만들고 진행해야했지만, Task가 sequential하여 앞선 Task에서 지연에 의해 뒷 부분을 빠르게 진행하지 못한 점이 아쉬웠다.
- **한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?**
    - Interaction 데이터의 범위를 넓혀 일반적으로 사용하는 추천시스템의 성능지표를 사용해 보고 싶다.
    - 모델을 구현에서 프로젝트가 마무리 되어, Airflow로 스케쥴링, 모니터링 진행해 보고 싶다.
    - Sequential한 Task라도 서비스의 틀이나 모델의 input, output부분을 결정하여 빠르게 Beta 버전을 먼저 만든 후 개선해 나가는 방식으로 진행해 보고 싶다.

임소영_T5172

- **나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?**
    - 도메인 (금융) 지식이 많이 없었기 때문에, 용어 및 방법 공부에 많은 시간을 사용하였다.
    - 처음으로 raw data 를 수집하고, 이를 정제하여 모델링 과정에서 사용하였다.
    - 데이터의 형태에 맞는 모델을 선정하고, 학습시켰다.
    - 처음으로 Streamlit 을 공부하고 활용하였고, 기본 UI 로는 부족하다고 생각하여, React 기반의 mui module 을 import 하여 디자인적 요소도 함께 고려하였다.
- **내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?**
    - 도메인 지식 습득의 한계점은 어쩔 수 없다는 생각이 들었고, 앞으로 프로젝트를 진행할 때 기획 단계에서 도메인 전문가와의 충분한 토론과 논의를 진행해야겠다는 생각이 들었다.
    - 데이터 이상치를 제거하면서, 별다른 기준 없이 임의로 삭제했던 적이 있었는데 팀원분께서 통계적 지식을 알려주셔서, IQR 을 이용하여 이상치를 제거하였다. 이를 통해 데이터 전처리 및 분석에도 통계적 지식을 이용할 수 있도록 통계를 더 깊이 공부해야겠다는 생각이 들었다.
    - Streamlit 으로 구현된 웹사이트를 dockerize 하여 google cloud 에 run 하였다. 처음으로 end to end 를 직접 구성해보면서 앞으로 큰 두려움 없이 다른 프로젝트에도 적용할 수 있을 것 같다.
- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**
    - data 의 sequence length 가 길지 않아서, RNN 과 LSTM 이 잘 학습시켰을지 의문이다. 최소한 5개년치라도 구할 수 있었으면 좋았을텐데 DART 나 smes 에서 제공하는 데이터의 양이 제한적이다보니 아쉬웠다.
    - 통계적 지식의 부재로 인해, EDA 를 원활히 진행하지 못한 것이 아쉬웠다.
    - Streamlit 과 MySQL 을 연동하는 부분에서, data 를 로딩하는 과정에서 시간이 오래걸려서 아쉬웠다.
- **한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?**
    - 통계 지식 습득 후, 논리적으로 EDA 를 진행해보고 싶다.

- Streamlit - MySQL loading 시간을 단축해보고 싶다.

전증원_T5185

- **나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?**
  - 데이터를 수집하기 위해 웹 스크래핑에 대해 공부하였다.
  - 웹 스크래핑의 시간을 단축시키기 위해 멀티 쓰레딩, 멀티 프로세싱에 대해 공부하였다.
  - ML 모델을 사용하기 위해 데이터 EDA를 하기 위해 노력하였다.
- **내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?**
  - 파이썬은 GIL이라는 것 때문에 멀티 쓰레딩이 파이썬이라는 언어의 철학과 맞지 않는다는 것을 알게 되었다.
  - 웹 스크래핑의 시간을 단축하기 위해 멀티 쓰레딩, 멀티 프로세싱을 공부하며 멀티 쓰레딩의 효과로 스크래핑에 걸리는 시간을 단축시켰다.
  - ML을 공부하며 통계적인 지식, 데이터의 특성을 잘 파악해야한다는 것을 깨달았다.
- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

아쉬운 점	개선 방향
`재무제표의 정보`만으로는 `주가를 표현하지 못하는 것 같아` 예측의 `정확도가 떨어지는것` 같았다.	`뉴스 기사 데이터`나 `특허 데이터`등을 활용하여 `데이터의 수를 늘려` 예측의 정확도를 올려보고 싶다.
AI의 `기초적인 지식이 부족`하여 모델이나 feature engineering을 적용할때 명확한 이유를 알지 못했다.	동영상 강의를 복습하고, 기본적인 서적을 공부해야겠다.

- **한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?**
  - AI의 기초적인 부분을 확실히 공부해서 어떤 선택을 할때 이유있는 선택을 하고 싶다.
  - 다음 프로젝트에서는 ML뿐만 아니라 DL을 사용하는 프로젝트를 해보고 싶다.