



Wrap-up Report

RecSys_3조 (냉장고를부탁해)

박영균 🏰, 강성택, 김다빈, 김윤경, 김희수, 노근서

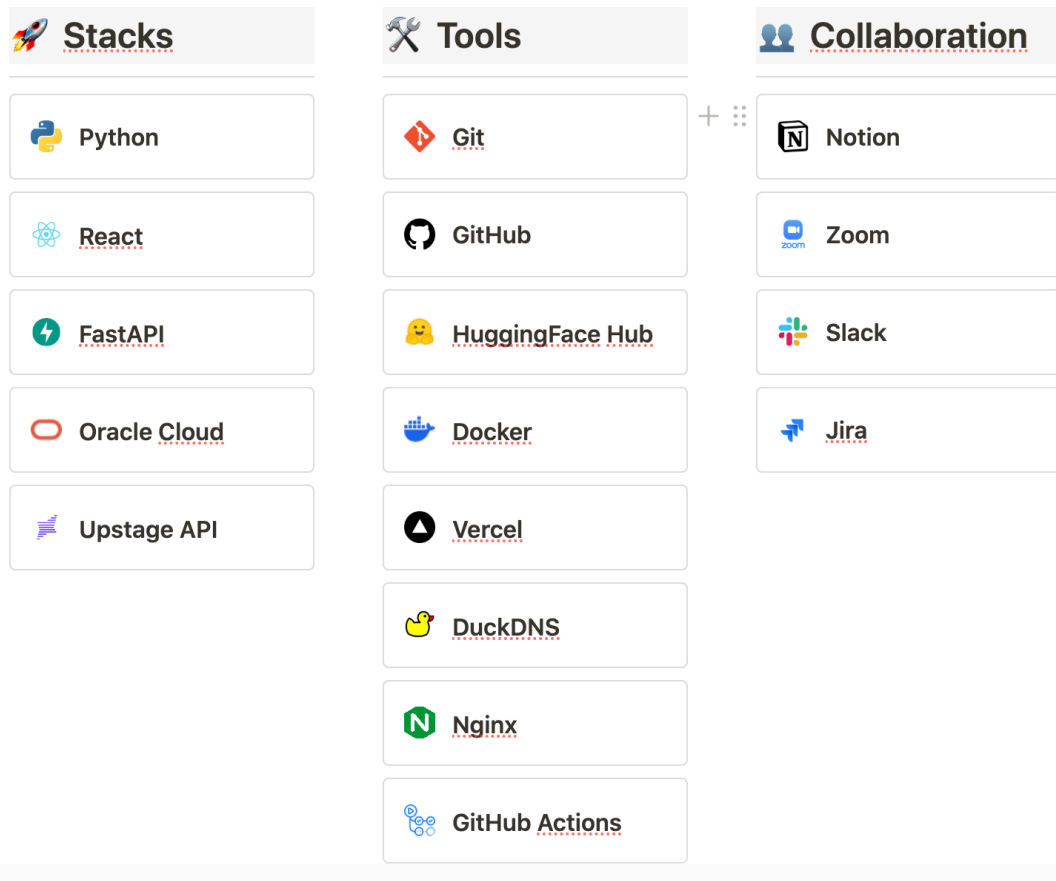
1. 프로젝트 개요

1.1 개요

- 기업은 프로젝트를 성공적으로 수행하기 위해 적합한 프리랜서를 찾아야 하지만, 개인의 역량을 검증하는데 많은 시간과 비용이 든다.
- AI 기반 추천 시스템을 활용해 기업과 프리랜서 간의 최적 매칭을 자동화하고, 더 빠르고 정확한 인재 매칭을 가능하게 만든다.
- 매칭 정확도를 높여 기업의 인재 탐색 비용을 절감시키고, 프리랜서에게 더 적합한 프로젝트 기회를 제공하여 양측의 만족도를 극대화하는 것이 목표이다.

1.2 환경

- 팀 구성: 6인 1팀
- 컴퓨팅 환경
 - Oracle Cloud (OCI) 서버: 백엔드 서버
 - Oracle Cloud (OCI) DB: 데이터베이스
 - V100 서버 (SSH 연결): 모델 학습 환경
- 배포 및 네트워크 환경
 - Vercel: 프론트엔드 배포
 - DuckDNS: 동적 DNS 설정
 - Nginx: Reverse Proxy, 로드 밸런싱 및 서버 관리
- 협업 환경: Notion, GitHub, Jira
- 의사 소통: Zoom, Slack



2. 프로젝트 팀 구성 및 역할

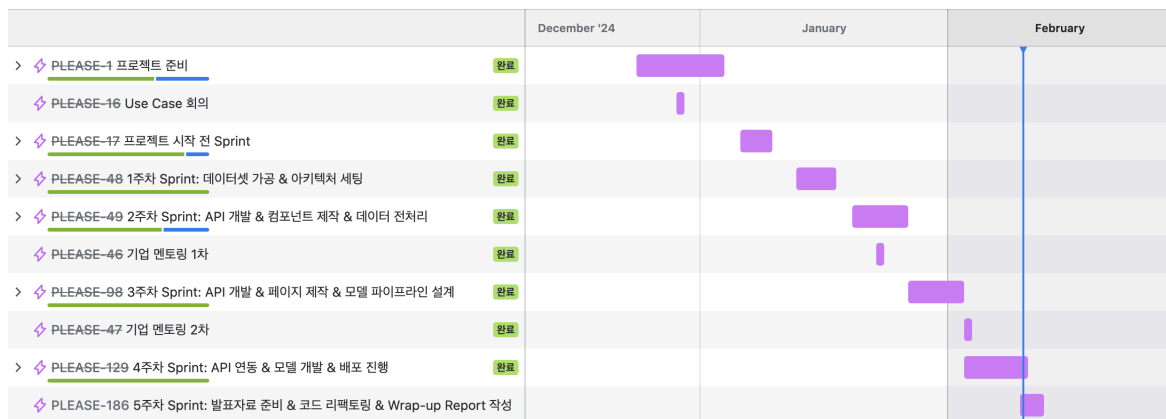
	Frontend	Backend & DevOps	Model
강성택	-	API 개발, 모델 및 서비스 배포	데이터 크롤링, 데이터 전처리(타겟 변수 생성) CB 모델 개발 (XGBoost, CatBoost, Logistic Regression)
김다빈	-	API 개발, DB 설계	모델 파이프라인 설계 LightGCN, Wide&Deep 모델 개발, 앙상블
김윤경	웹페이지 개발, API 연동	-	모델 파이프라인 설계 EASE, SLIM 모델 개발, 앙상블
김희수	웹페이지 개발	-	EDA, 데이터 전처리(타겟 변수 생성, PCA)
노근서	웹페이지 개발, API 연동	-	데이터 전처리(프리랜서, 인코딩/임베딩) FM 모델 개발
박영균 (팀장 🍕)	웹페이지 개발, API 연동	-	데이터 전처리(프로젝트), CB 모델 개발 (XGBoost, CatBoost, Logistic Regression)

3. 프로젝트 수행 절차 및 방법

프로젝트 타임라인

- 진행 기간: 2025.01.10 (금) 10:00 ~ 2025.02.12 (수) 14:00

- 주 단위 스프린트 계획 및 백로그 작성



[세부 내용]

1. 프로젝트 규칙 설정: Project Rule 설정, GitHub/Coding Convention(PEP8 기반) 설정
2. 개발 환경 구축: 서버 세팅, GitHub 템플릿 적용, Jira-GitHub Issues 연동
3. 요구사항 정의 및 설계: 서비스 요구사항 정의, 서비스 아키텍처 설계, 와이어프레임 제작, ERD 설계
4. 데이터셋 준비와 FE/BE 병행
 - ML: 데이터셋 크롤링 및 가공, 모델 설계 및 파이프라인 구축
 - FE: 공통 컴포넌트 개발 | BE: API 설계 및 DB 설계
5. 데이터 전처리와 FE/BE 병행
 - ML: 데이터 전처리 및 피처 엔지니어링
 - FE: 웹 페이지 개발 | BE: API 개발
6. 모델 구현과 FE/BE 병행
 - ML: 모델 학습 및 추론 기능 개발, 하이퍼파라미터 튜닝, 앙상블 설계
 - FE: 웹 페이지에 API 연동 | BE: API 개발
7. 서비스 배포: 웹 페이지 배포 및 최종 테스트

협업 방식

- GitHub Issues 관리
 - Jira와 자동으로 연동되도록 Issue Template 작성(개요, Ticket Number, DoD, To-Do 작성)
 - Issue와 관련된 자세한 작업 내용은 Notion 데이터베이스 보드로 확인
- GitHub/Code Convention에 따라 GitHub Repository 관리

4. 프로젝트 수행 결과

프로젝트 핵심 기능

- 기업이 프로젝트 내용을 제시 → Upstage Chat API 활용

- 등록할 프로젝트에 대한 **요약** 정보 및 **필요 스킬**에 대한 태그 생성
- 기업의 공고 내용과 프리랜서의 정보를 **AI 모델로 학습**
 - 기업에게 추천 결과를 **매칭 점수**의 형태로 제공
- 프리랜서의 **프로젝트 지원 현황, 스킬 역량(점수)** 정보를 추가로 제공
 - 프로젝트 종료 후 **기업 피드백**을 반영하여 프리랜서의 보유 **스킬 숙련도** 점수 조정

데이터셋 설명

- 프리랜서 데이터
 - Stackoverflow 사용자 대상 설문조사(2024)를 바탕으로 한 개발자 정보 데이터
 - 65,437개의 데이터 중 프리랜서에 적합한 조건을 가진 5488개 데이터 추출
- 프로젝트 데이터
 - 위시켓(wishket) 사이트에 기업이 게시한 개발 관련 프로젝트 공고 데이터
 - 총 22892개의 외주, 기간제 프로젝트 데이터
- 두 데이터셋으로 학습 가능한 과거 이력을 생성하기 위해 새로운 함수를 정의한다.

타겟 변수 y 생성

[매칭 점수 S 정의]

- 프리랜서가 과거 참여한 프로젝트 이력(Implicit Feedback)을 생성할 수 있게 매칭 점수를 정의해 기업이 과거 선호한 프리랜서 목록이 있다는 전제로 타겟 변수를 사용한다.
- 기업마다 (또는 프로젝트마다) 필요한 프리랜서의 요구사항이 다르다는 가정을 위해, 다음과 같이 매칭 점수를 계산하는 함수를 정의한다.

$$S_i = \sigma(\alpha \cdot \text{Skill Overlap} + \beta \cdot \text{Skill Temperature} - \gamma \cdot \text{Cost} + \delta \cdot \text{Category Match}) \times 100$$

- $\alpha, \beta, \gamma, \delta$: 각 항에 대응하는 가중치(하이퍼파라미터).
- σ : 시그모이드 함수
- Skill Overlap: 프로젝트 요구 스킬을 프리랜서가 가질 비율.

$$\text{Skill Overlap} = \frac{|\text{Project Skills} \cap \text{Freelancer Skills}|}{|\text{Project Skills}|}$$

- Skill Temperature: 경력이 높을수록 스킬 숙련도는 우상향하게 반영한다. 초깃값은 2로 설정하고 노이즈를 추가한다. 마찬가지로 Min-Max 스케일링 적용

$$\text{Skill Temperature} = 2.0 + \left(\frac{\text{Experience Years}}{\max(\text{Experience Years})} \times 2.0 \right) + \text{Random Noise}$$

- Cost: 프리랜서 고용비 → Min-Max 스케일링 적용

$$\text{Cost} = \frac{\text{Freelancer Cost} - \min(\text{Freelancer Cost})}{\max(\text{Freelancer Cost}) - \min(\text{Freelancer Cost})}$$

- Category Match: 참여한 산업 분야의 겹치는 정도

$$\text{Category Match} = \begin{cases} 1, & \text{if Project Category} = \text{Freelancer Industry} \\ 0, & \text{otherwise} \end{cases}$$

[추천 이력 R 생성]

- 각 프로젝트 i 별로 매칭 점수(Matching Score, S_i)가 높은 상위 K 명의 프리랜서 선택
: 기업이 과거 프로젝트에 선호한 K 명의 프리랜서 정보를 의미

$$R_i = \{u_1, \dots, u_K\}, S_i(u_1) \geq \dots \geq S_i(u_K)$$

[타겟 변수 Y 생성]

- 각 프로젝트 i 에서 선호한 프리랜서의 집합을 R_i 라고 할 때, 프로젝트 i 에 선호한 프리랜서 j 의 Y 값은 1, 나머지 프리랜서의 Y 값은 0으로 표현한다.

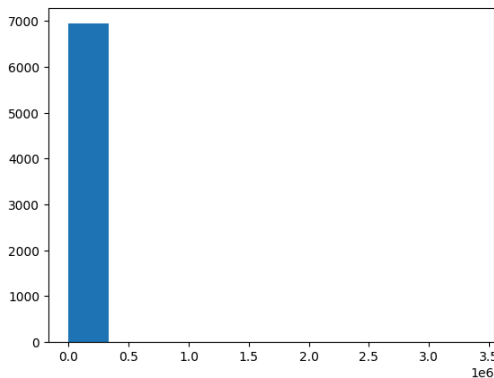
$$Y_{ij} = \begin{cases} 1, & j \in R_i \\ 0, & j \notin R_i \end{cases}$$

- 상위 K 명의 프리랜서의 매칭 점수 S_i 에 Softmax 함수를 적용하고, 상위 K 명 중 한 명을 소프트맥스 출력만큼의 확률로 프리랜서가 과거 프로젝트에 매칭된다.

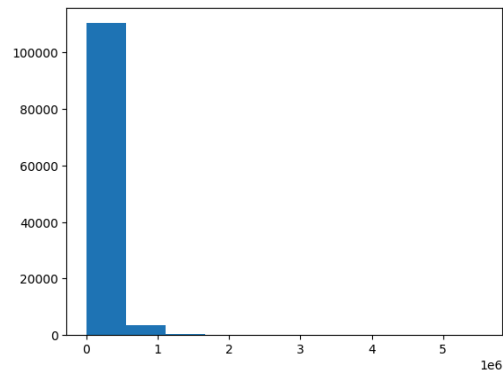
$$P(i) = \frac{e^{S(i)}}{\sum_{j=1}^K e^{S(j)}}$$

- Y 는 과거 프로젝트와 프리랜서의 Implicit feedback으로 활용할 수 있고, 실제 과거 프로젝트에 매칭된 프리랜서 정보도 활용 가능하도록 생성한다.

탐색적 데이터 분석(EDA)



<Figure 1> 프리랜서 가격의 히스토그램



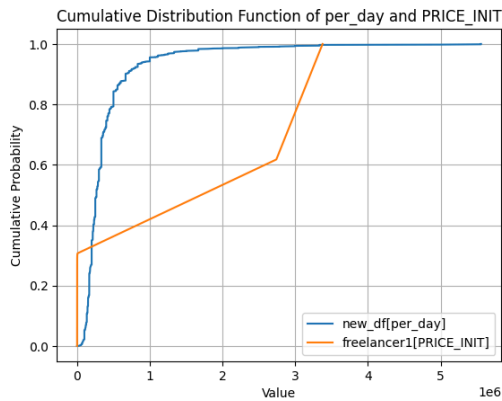
<Figure 2> 프로젝트 가격의 히스토그램

- 프리랜서의 가격은 원화(KRW), 프로젝트의 가격은 달러(USD)이고 가격의 간격이 다르므로 두 분포가 유사하다고 말하기 어렵다.

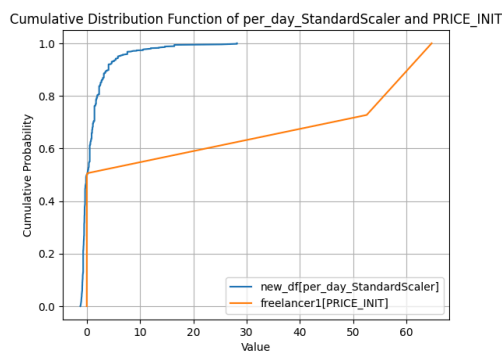
- 두 변수의 확률분포가 서로 동일한지 여부를 판단하기 위해 콜모고로프-스미르노프 검정(Kolmogorov-Smirnov Test, KS-Test)을 사용

<Table 1> 프리랜서와 프로젝트 가격에 여러 가지 정규화 방법을 적용한 KS 검정 통계량과 유의확률

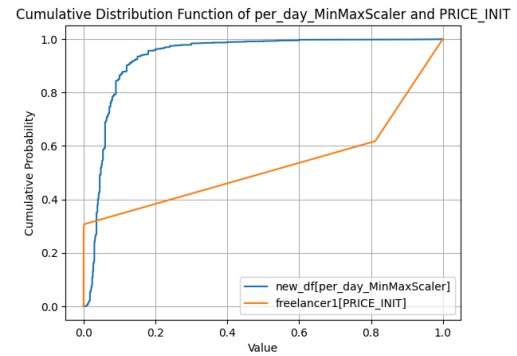
정규화 방법	Default	MinMaxScaler	StandardScaler	RobustScaler
KS 검정 통계량	0.9989	0.9984	0.6390	0.0869
유의확률	<.0001	<.0001	<.0001	<.0001



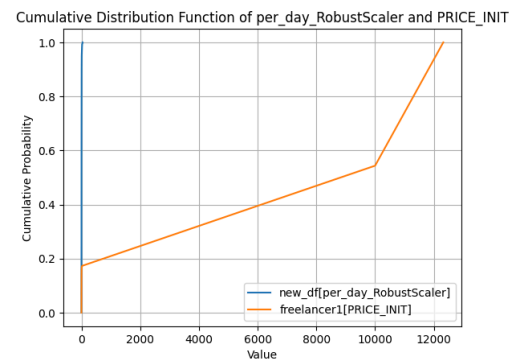
<Figure 3> Default인 누적분포함수



<Figure 5> StandardScaler를 적용한 누적분포함수

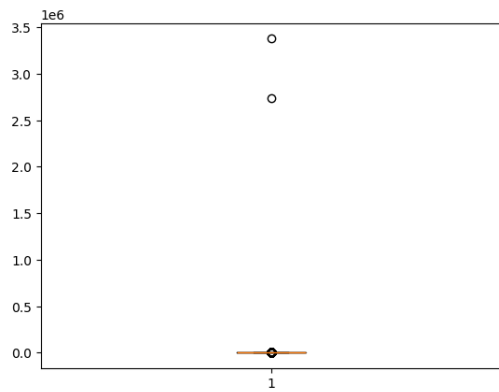


<Figure 4> MinMaxScaler를 적용한 누적분포함수

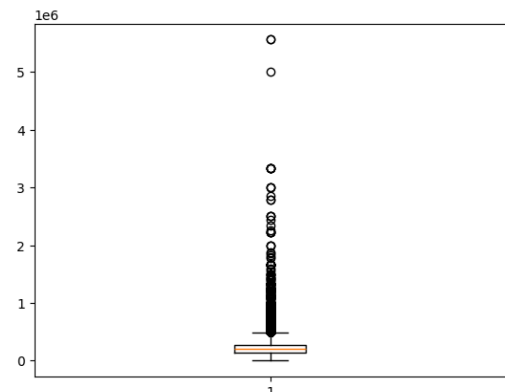


<Figure 6> RobustScaler를 적용한 누적분포함수

- 유의확률이 전부 0.0001 미만이므로 두 변수마다 각각 정규화 방법들을 적용했을 때, 모두 프리랜서와 프로젝트의 가격이 서로 다르다. 또한, <Figure 3 - 6>을 보면 두 변수의 누적분포함수가 서로 상이하다.



<Figure 7> 프리랜서 가격의 상자그림

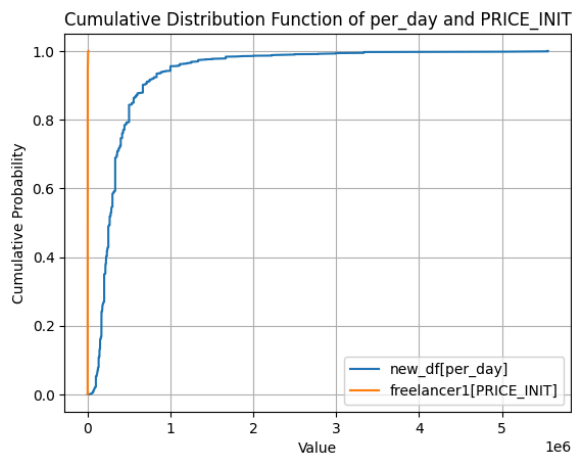


<Figure 8> 프로젝트 가격의 상자그림

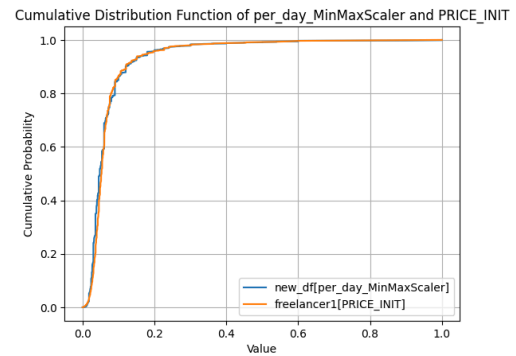
- 그 원인으로 프리랜서의 가격과 프로젝트의 가격에 높은 가격의 이상치가 있기 때문이다. 그래서 높은 이상치 일부를 제거하여 다시 분석해보았다.

<Table 2> 이상치 제거 후 프리랜서와 프로젝트 가격에 여러 가지 정규화 방법을 적용한 KS 검정통계량과 유의확률

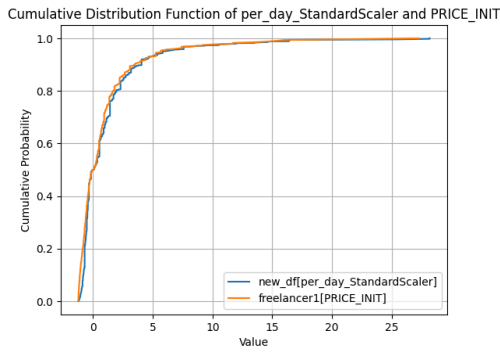
정규화 방법	Default	MinMaxScaler	StandardScaler	RobustScaler
KS 검정통계량	0.9992	0.1542	0.1114	0.0869
유의확률	<.0001	<.0001	<.0001	<.0001



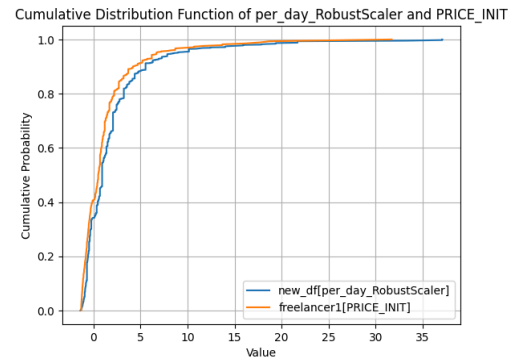
<Figure 9> 이상치 제거후 Default인 누적분포함수



<Figure 10> 이상치 제거후 MinMaxScaler를 적용한 누적분포함수

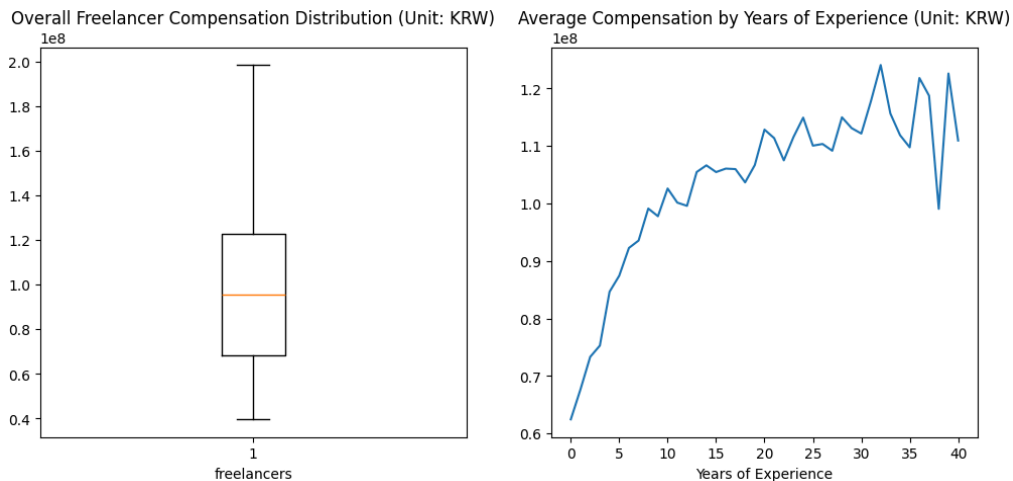


<Figure 11> 이상치 제거후 StandardScaler를 적용한 누적분포함수



<Figure 12> 이상치 제거후 RobustScaler를 적용한 누적분포함수

- 결과적으로 두 변수의 분포를 유사하다고 할 수 있는 통계적인 결과를 얻지는 못했다.
- 따라서 우리는 통계적으로 유의하지 않더라도, 두 변수 사이의 분포가 유사하다고 말할 수 있도록 여러 가정을 통해 변환한 변수를 사용해 데이터의 신뢰도를 최대한 보장할 수 있도록 했다.
 - 미국 개발자 시장의 평균 연봉과 한국 프리랜서 개발자 시장의 평균 연봉을 설문조사 시점(2024.05)에 맞춰 통계 자료 확인 후 스케일링 적용
 - 설문조사 시점(2024.05)에 맞춰 달러-원화 환율 적용
 - 현실적인 한국 프리랜서 개발자 시장의 최소, 최대 연봉의 경계를 4,000만 ~ 2억 원 사이로 선정 후 필터링



<Figure 13> 최종 변환한 프리랜서 가격 변수의 상자 그림과 경력-가격 선 그래프

데이터 전처리 및 피쳐 엔지니어링

[전처리 및 피쳐 엔지니어링]

- 프로젝트 금액, 프리랜서 연봉 : 일 단위로 통일
- 프리랜서 경력 : 최대 40년으로 범위 조정
- 프로젝트 산업 분야, 필요 스킬셋 전처리 후, 프리랜서 활동 분야, 보유 스킬셋 범주와 교집합 사용
- 프로젝트 내용 : Upstage Embeddings API 활용 후, PCA로 차원 축소 (4,096 → 256 또는 512)

[변수 선택]

- 수치형 변수

- 프리랜서: `경력`, `연봉` | 프로젝트: `금액`

- 범주형 변수

- 프리랜서: `보유 스킬셋`, `활동 분야` (멀티-핫 인코딩 또는 `torch.nn.Embedding` 활용)
- 프로젝트: `필요 스킬셋`, `산업 분야` (멀티-핫 인코딩 또는 `torch.nn.Embedding` 활용),

`프로젝트 내용` (`Upstage Embeddings` API 활용),

- 상호작용 변수: 프로젝트별 과거 선호 프리랜서 정보(positive)와 매칭 점수가 담긴 상호작용 행렬

모델 선정 및 평가

[프로젝트-프리랜서 매칭 Task의 특징]

- 희소한 상호작용 문제

- 프로젝트당 1명의 프리랜서만 배치되어, 기존 CF 기반 모델이 학습하기 어렵다. 또한 기존 유저 데이터(프로젝트)에 새로운 아이템(프리랜서)의 평가가 업데이트 되지 않는다.

- 콜드 스타트 문제

- 새로운 유저(프로젝트), 아이템(프리랜서) 등장 시 과거 상호작용 데이터를 활용하기 어렵다.
- 유저, 아이템 메타데이터를 활용할 수 있는 모델이 필요하다.

[모델 선정 및 이유]

- EASE

- 데이터 희소성에 적합

- 상호작용 데이터가 제한적인 경우, 행렬 기반 방법이 효과적
- Gram Matrix 기반 연산을 활용해 데이터가 적어도 학습 가능

- 빠른 계산 속도와 해석 가능성

- 대규모 데이터셋에서도 빠르게 학습 가능
- 아이템 간 유사도를 분석하는 방식이므로 해석이 쉽다.

- 단, 콜드 스타트 문제 해결이 어렵다.

- 프리랜서(아이템) 정보 없이 과거 유사도를 기반으로 추천하므로 신규 아이템 추천이 어려운 점은 단점

- FM

- Feature Interaction 학습 가능

- 프로젝트(유저)와 프리랜서(아이템)의 스킬, 경력, 산업 분야 등의 메타데이터 반영 가능
- Feature 간 상호작용을 학습해 관계를 고려한 추천 가능

- Explicit/Implicit Feedback 모두 활용 가능

- 기업이 특정 프리랜서에게 관심을 가진 정보를 활용해 Implicit Feedback 기반 학습 가능

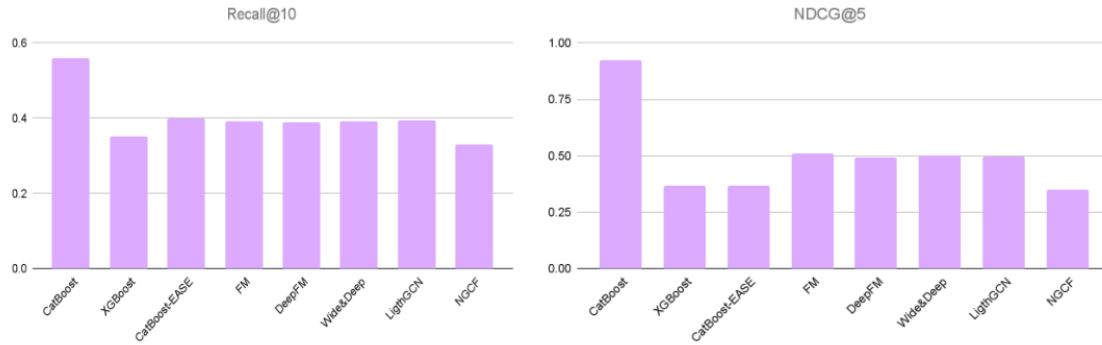
- BPR Loss와 같은 랭킹 기반 손실 함수도 적용 가능
- 단, 비선형적인 관계 학습에는 한계 존재
 - FM은 2차 상호작용까지만 학습이 가능해 더 복잡한 Feature 조합을 학습하려면 DeepFM 등 모델 필요
- Tree: CatBoost
 - 비선형적 Feature 관계 학습 가능
 - 선형적인 관계만 학습 가능한 FM 모델과 달리, 트리 모델은 더 복잡한 관계 학습 가능
 - 빠른 학습 속도 및 해석 가능성:
 - GPU 학습을 지원해 대규모 데이터에서도 빠르게 학습 가능
 - Feature Importance 제공으로 모델 추천 결과의 근거 분석이 가능
 - 콜드 스타트 문제 해결 가능
 - 프리랜서, 프로젝트의 메타데이터를 직접 입력값으로 사용 가능
 - EASE, FM이 과거 상호작용 기반 학습으로 콜드 스타트 문제에 취약한 반면, CatBoost는 Feature만으로 예측 가능 (Content-based)
 - 단, 상호작용 기반 추천에는 상대적으로 약세
 - Feature 기반 예측을 수행해 CF 기반 모델이 제공하는 유사도 정보는 상대적으로 부족

[모델 평가]

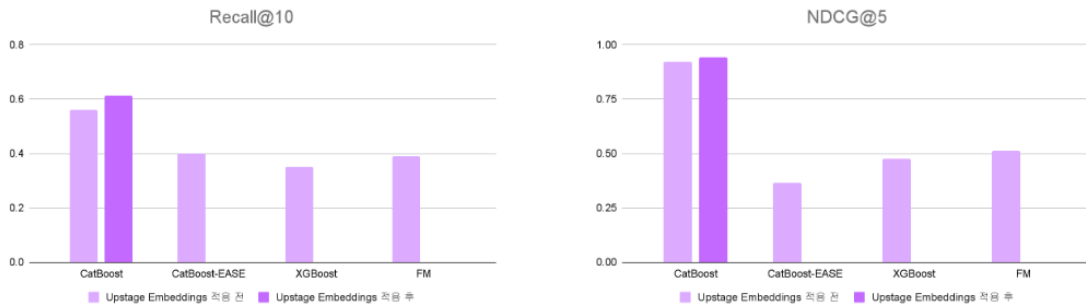
이외 다양한 모델(XGBoost, DeepFM, Wide&Deep, LightGCN, NGCF)을 활용해 Recall@10, NDCG@5 성능 비교

- Recall@10
 - 적절한 인재의 추천을 빠뜨리지 않는 것이 중요
 - 틀린 추천을 하지 않는 것도 중요하지만, 기업이 적합한 프리랜서를 찾지 못하는 것이 더 큰 손실이므로 Recall@10을 높이는 것이 핵심 목표
- NDCG@5
 - 매칭 점수 기준으로 추천 순위를 매기는 서비스의 특성 반영
 - 기업이 한 프로젝트에 채용하는 프리랜서 수는 대체로 소수
 - 상위권 추천이 실제 선택에 미치는 영향이 크며, 추천 순위 품질을 중점적으로 평가

모델 성능 및 결과



- CatBoost가 **Recall@10** 과 **NDCG@5** 모두 가장 높은 성능 달성
 - **Feature 기반 학습**으로 콜드 스타트 유저(신규 프로젝트)에서도 다른 모델보다 좋은 추천 가능
 - **Sparse & Structured 데이터에 강하고, 비선형적인 Feature 간 관계 학습이 가능**해 다른 모델보다 높은 성능 달성



- **프로젝트 내용** Upstage Embeddings 적용 후 추가 Feature로 사용 시, **Recall@10** : ▲ **9.5%**, **NDCG@5** : ▲ **2.2%**

시행 착오

Model

- 모델 설계 시 학습에 스킬 온도를 반영하고자 했으나 어떤 모델에서는 반영할 수가 없거나 성능이 떨어지는 문제 발생하여 사용하지 못했다.
 - CF 모델(EASE, LightGCN, NGCF)은 Item feature를 반영할 수 없어 사용 불가
 - 스킬셋과 스킬 온도가 서로 한 쌍으로 모델이 인식해야 하지만, 두 Feature의 타입이 달라 (스킬셋-범주형, 스킬 온도-수치형) 의도한 학습대로 이루어지지 않는 문제 발생
 - 이를 해결하기 위해 스킬셋을 멀티-핫 인코딩 또는 임베딩 후 스킬온도를 가중치로 반영했지만, 사용 시 성능이 떨어져 결국 **스킬 온도를 학습에서 제외**

Frontend

- 여러 명이 동시에 컴포넌트/페이지 개발을 진행하면서 css에서 동일한 클래스 이름이 중복되어, 스타일이 예상과 다르게 적용되는 문제 발생
 - css 파일이 전역 범위에서 적용되다 보니, 특정 컴포넌트의 스타일이 다른 컴포넌트에도 영향을 줌
 - **중복된 클래스명을 정리**하고, 동일한 디자인을 가진 컴포넌트는 **동일한 클래스명으로 통일**

Backend

- Solar에 현재 DB에 있는 프로젝트 데이터를 넘겨 등록하는 프로젝트와 유사한 프로젝트를 찾도록 했지만 데이터 양이 많아 문제 발생
 - **사전 필터링된 50개의 프로젝트만 사용**하여 유사한 프로젝트를 찾는 방식으로 변경 (카테고리, 예산, 스킬 태그 등을 고려하여 1차 필터링)
- 프로젝트 등록과 동시에 모델 추론 결과를 저장하는 방식을 사용하여 등록에 시간이 오래 걸린다는 문제 발생
 - 모델 추론 및 결과 저장을 **비동기 방식으로 처리**하고 결과 저장이 끝나면 WebSocket을 통해 **실시간 알림**을 제공
- 새로운 프로젝트의 텍스트 데이터를 추론할 때마다 전체 프로젝트 데이터를 다시 가져와 PCA를 실행해 추론 시간이 오래 걸리고 비효율적이라는 문제 발생
 - 모델 훈련에 사용하는 데이터 전처리 시에 **PCA 모델을 미리 학습하여 저장**하고, 추론 시 불러와 사용하도록 변경

DevOps

- Vercel을 활용한 배포 과정에서 사용한 레포지토리(level4-recsys-finalproject-hackathon-recsys-03-lv3)가 boostcampaitech7 organization에 속해있어 Pro 요금제 및 권한에 여러 사항 발생
 - 개인 repository에 fork를 하여 개인 repository와 연동하는 방식으로 진행하여 **Github Actions를 활용해 CD까지** 가능하도록 해결

기대효과

- **추천 시스템의 활용**으로 매칭에 소요되는 시간과 비용을 절감시키고 중간 인력의 부담을 줄일 수 있다.
- **사후 피드백을 반영한 스킬 점수**는 기업에게 프리랜서의 전문성에 대한 객관적이고 신뢰할 만한 정보를 제공
- 기업과 프리랜서 양측의 **인력 풀을 확장**함으로써 보다 다양한 선택지를 제공하고 협업 성공률을 높일 수 있다.

서비스 확장성

- 타 프리랜서 매칭 플랫폼과 같이 다양한 **도메인에 대해 범위를 확장**
- **LLM 추천 모델 추가**
- **프로젝트 등록 과정에 OCR 기술을 접목**해 프로젝트 기획서의 형태로 첨부하는 기능 추가
- 프리랜서 맞춤 기능 추가
- **프리랜서의 선호도 반영**: 프리랜서 소개글에서 선호 유형을 LLM으로 추출해 반영
- **프리랜서 성장 로드맵** 추천: 프리랜서가 원하는 프로젝트에 매칭되기 위해 보완할 역량 수준 제공
- 기업(프리랜서) 입장에서, **추천의 이유 설명**

5. 자체 평가 의견

잘한 점

- **다양한 모델 구현**: 여러 가지 모델을 시도하고 다양한 접근 방식을 실험함으로써 데이터의 특성을 고려하면서도 최적을 성능을 내기 위해 노력했다.
- **끝까지 최선을 다하는 태도**: 중간에 주제가 변경되어 일정이 촉박했음에도 불구하고, 모두가 최대한 시간 투자하며 기한 내에 프로젝트를 성공적으로 마무리 할 수 있었다.

- **적극적인 피드백 반영:** 두 멘토님(부스트캠프 & 업스테이지)으로부터 받은 피드백을 적극적으로 반영하여 놓친 부분을 보완해 프로젝트의 완성도를 높일 수 있었다.

아쉬운 점

- **구현 능력 부족을 LLM으로 대체:**
 - 기한 내 완성과 프로젝트 완성도를 위해 이전 프로젝트 대비 LLM 의존 빈도 증가
- **데드라인 및 일정 준수 미흡:**
 - 프로젝트 초기에 구체적 목표와 실현 가능성 등 프로젝트 청사진을 철저히 검증하지 못해 촉박해진 일정
 - 매주마다 스프린트 계획을 세우고 마감 날짜를 설정했지만, 예상 시간을 넘기는 작업이 지속적으로 발생하면서 일정이 지연되어 계획된 스프린트 진행에 차질을 빚었다.

이번 프로젝트를 통해 배운 점

3 GUNA!

1. 전체 파이프라인을 경험해보니, 어떤 프로젝트든 앞으로 잘 할 수 있겠**GUNA!**
2. 데이터 수집 단계부터 데이터 전처리까지 해보니, 양질의 데이터가 정말 소중하**GUNA!**
3. 우리 모두 해커톤에서 새로운 가능성을 발견했다. 뭐든지 일단 시도해 보는 것이 중요하**GUNA!**

다음 프로젝트에 적용해볼 점

HAJA!

1. 부캠은 끝나도 우리의 프로젝트는 끝나지 않는다. 시간 때문에 넣지 못했던 기능들 꼭 추가**HAJA!**
2. 앞으로의 커리어에 있어 너무나도 값진 이 경험을 모두 기록**HAJA!**
3. 일단 발표 끝나면 폭 쉬는 걸로 **HAJA!**

6. 개인 회고

강성택_T7501

이번 프로젝트에서 나는 데이터 크롤링, 데이터 전처리(타겟 변수 생성), CB 모델 개발(XGBoost, CatBoost, Logistic Regression), API 개발, 그리고 모델 및 서비스 배포를 담당했다. 이를 통해 추천 시스템을 구축하고, 실제 서비스에 적용하는 과정 전반을 경험할 수 있었다.

1. 나의 학습목표

이번 프로젝트에서 나는 보다 **실용적인 추천 시스템을 구축하는 것**을 목표로 했다. 단순한 규칙 기반 추천이 아닌 **데이터 기반의 학습 모델을 활용한 추천 시스템을 개발하고, 이를 실제 서비스에 적용**하는 것이 핵심이었다. 이를 위해 데이터 크롤링과 전처리를 통해 추천 모델이 학습할 수 있는 데이터를 구축하고, CatBoost, XGBoost, Logistic Regression 등 다양한 모델을 실험하며 최적의 성능을 찾는 과정에 집중했다. 또한 FastAPI를 활용해 모델을 API로 배포하고, 서비스 배포까지 진행하며 **추천 시스템의 실무적 운영 방식까지 경험하고자 했다.**

2. 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

기존 프로젝트에서는 주어진 데이터를 활용하여 모델을 학습하고 분석하는 데 집중했다면, 이번 프로젝트에서는 **실제 데이터를 직접 크롤링하고, 타겟 변수를 생성하여 추천 모델을 학습하는 방식**을 시도했다. 이를 통해 보다

현실적인 추천 시스템을 구축할 수 있었다.

또한 기존에는 모델을 로컬에 저장하고 직접 배포해야 했지만,

Hugging Face Hub를 이용해 모델을 저장하고 쉽게 로드할 수 있어 협업 및 배포가 한층 더 편리해졌다. 이를 통해 모델을 재사용하거나 다른 환경에서도 손쉽게 접근할 수 있는 방식을 적용할 수 있었다.

2. 개인 학습 측면

이번 프로젝트를 통해 크롤링과 데이터 전처리 기술을 실무적으로 활용하는 경험을 쌓을 수 있었다. 단순한 데이터 수집이 아니라,

추천 시스템에 적합한 타겟 변수를 생성하고 가공하는 과정이 핵심이었다. 이를 통해 지도 학습 모델을 적용하는데 필요한 데이터 준비 과정을 직접 경험할 수 있었다.

또한,

FastAPI, Vercel, Nginx, DuckDNS를 활용하여 추천 API와 서비스를 배포하는 과정도 새로운 경험이었다. 단순한 모델 배포가 아니라, 실제 사용자에게 서비스를 제공할 수 있는 환경을 구성하는 과정에서 **실무적인 배포 경험을 쌓을 수 있었다.**

3. 공동 학습 측면

협업 관련해서는

Jira를 활용하여 태스크를 관리하고, 팀원과 작업을 효과적으로 분배하는 방식도 시도했다. HuggingFace Hub에 모델을 공유하면서, **팀원들이 같은 모델을 활용할 수 있도록 중앙 집중형 모델 관리 환경을 구축할 수 있었다.**

4. 기억에 남는 시도

모델링 과정에서 CatBoost의 **범주형 변수 자동 처리 기능**을 활용한 결과, Recall@10 기준 0.3300 에서 0.5603으로 약 **69.7% 성능 향상**을 이끌어냈다. 이후, **Upstage의 임베딩 API**를 활용하여 프로젝트 내용을 임베딩한 Feature를 추가한 결과, 0.5603에서 0.6138로 약 **9.59% 성능 향상**을 이끌어냈다. 이 과정 속에서 단순히 성능이 좋은 모델을 선택하는 것이 아니라, **데이터의 구조와 특성에 맞는 최적의 모델을 선정하고, 그 선택에 대한 타당한 이유를 고민하는 과정이 필수적임을 다시 한번 깨달았다.**

5. 한계와 아쉬움

이번 프로젝트에서 아쉬운 점은

실제 사용자 피드백을 반영하지 못한 것이다. 추천 모델의 성능을 개선하기 위해서는 실제 사용자의 반응과 데이터를 지속적으로 반영해야 하지만, 이번 프로젝트에서는 제한된 데이터로만 평가를 진행했다. 또한, CD 단계까지는 구축하였지만 CI 단계까지는 진행하지 못하여 A/B 테스트를 해보지 못한 점이 아쉬웠다.

6. 앞으로의 도전

이번 프로젝트를 통해 나는

추천 시스템의 End-to-End 과정을 경험할 수 있었다. 데이터 수집부터 전처리, 추천 모델 개발, API 구현, 배포까지의 **전체 흐름을 수행**하며, 추천 시스템을 실무에서 어떻게 운영해야 하는지 배울 수 있었다. 이 경험을 잃지 않고 더욱 완성도 높은 추천 시스템을 개발하고자 한다.

김다빈_T7558

1. 나의 학습목표

실제 비즈니스 문제를 해결하는 실용적인 AI 기반 서비스를 만드는 것을 목표로 했다. 주제 선정할 때부터 **AI 기술이 비즈니스에서 어떤 가치를 창출할 수 있을지 고민**했으며, 단순히 추천 시스템을 만드는 것이 아니라 이를 실제 서비스로 운영하는 데 필요한 모든 과정을 경험하는 것을 목표로 했다.

2. 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

데이터 수집부터 전 과정을 진행하면서, 단순한 모델 학습이 아니라 평가지표 선정부터 평가 방법까지 모든 부분을 깊이 고민해야 했다. 이를 통해 단순한 AI 성능이 아니라, **실제 서비스에서 의미 있는 평가 기준**을 정할 수

있었고, **보다 현실적으로 유용한 모델**을 만들 수 있었다고 생각한다. 또한 FastAPI의 비동기(Async) 처리를 활용하면서, 기존에 학습했던 동기 방식(Spring)과의 차이를 이해하고, **성능 최적화 및 비동기 API 설계에 대한 경험**을 쌓을 수 있었다.

2. 개인 학습 측면

백엔드 관련 DB 설계부터 API 개발까지를 주로 담당하고, 모델 학습에 필요한 데이터 전처리 작업을 하며 현재 가지고 있는 데이터셋을 어떻게 활용해야 할지와 각기 다른 출처의 데이터셋의 관계를 설득력 있게 만들지를 고민하며 **데이터 기반의 의사결정, 다대다 관계 데이터 모델링, SQL 최적화, 임베딩 기법** 등을 경험할 수 있었다. 또한 모든 API를 실제 서비스 수준으로 설계하고 구현하기 위해 고민하며 개발했다. OAuth 인증, 에러 처리 등 단순한 기능 구현이 아니라 **깊이 있는 설계와 안정성을 고려하여 구축**했다.

3. 공동 학습 측면

이전 프로젝트들을 진행하면서

우리 팀만의 협업 방식이 정립되어 있었던 점이 큰 도움이 되었다. GitHub Issues와 Jira를 활용해 스프린트 일정 별로 각자 작업을 등록하고 관련해서 공유해야 하는 내용을 Notion 보드에 기록했다. 또한 매일 회의록에 작업 내용을 간단하게 기록해 연휴 동안 개인 작업을 하면서도 진행상황을 파악하며 협업할 수 있었다.

4. 기억에 남는 시도

사용자가 프로젝트를 등록한 후, 스킬 태그 및 요약 결과를 받아보고, 최종적으로 매칭 결과까지 확인하는 과정을 실제 서비스처럼 구현하기 위해 최적화에 많은 노력을 기울였다. 단순한 기능 개발이 아니라, **사용자 경험을 최우선으로 고려**하여, 프로젝트 등록 후 최대한 빠르게 결과를 제공할 수 있도록 모델 추론 속도를 개선하고, 비동기 처리를 도입하여 응답 지연을 최소화했다.

5. 한계와 아쉬움

차별화된 서비스를 구현하고 싶어 '스킬 온도' 개념을 제안했지만, 실제 모델 학습에 적용하기에는 어려움이 있어 결국 반영하지 못한 점이 아쉽다. 또한 실제 서비스를 만들겠다는 목표에 맞게 모델학습 배치 서버까지 구현하고 싶었지만 시간이 부족해 하지 못했다는 아쉬움이 남는다.

6. 앞으로의 도전

맨땅에서 아키텍처 설계부터 시작해서 서비스를 구축해나가는 과정 자체가 큰 도전이었다고 생각한다. 이번 프로젝트를 시작할 때 그동안 배운 기술들을 활용해야겠다고 생각했었는데, 비즈니스 문제를 해결하는 과정 속에서 문제를 해결해 나가는 과정 속에 **새롭게 배우는 기술들이 있고 이 경험들이 곧 실력이 된다는 것**을 배웠다. 앞으로도 적극적으로 도전하며, 끊임없이 배우며 성장해 나가겠다.

김윤경_T7511

1. 나의 학습목표는 무엇이었나?

이번 프로젝트에서는 전체적인 파이프라인을 경험하는 것이 목표였다. 하나의 서비스를 구축하는 데 필요한 모든 과정을 한 달이라는 시간 안에 마무리하고자 했다.

2. 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

서비스를 구축하기 위해 프론트엔드 작업을 처음으로 담당하게 되었다. 물론 처음 다뤄보는 언어로 와이어프레임에 맞게 구현해낸다는 점이 어렵기도 했지만, 결과가 눈에 보이는 작업인 만큼 흥미를 느끼게 되었다. 하지만 처음인 만큼 파일 구조나 컴포넌트, 클래스 이름 등 규칙을 명확하게 정하지 못해 충돌이 생기는 문제가 잦았던 것 같다. 앞으로도 기회가 된다면 프론트작업을 좀 더 배워서 이러한 문제가 생기지 않도록 공부해봐야겠다.

3. 개인 학습 측면

지난 프로젝트에서 마무리하지 못했던 Recbole 라이브러리를 성공적으로 사용할 수 있었다. 확실히 모델 실험하기에 편리했던 것 같다. 하지만 지난 번에 왜 안됐는지 명확한 원인을 찾지 못하기도 했고, 라이브러리를 다

루는 데 있어서 아직 부족한 점이 많다는 것도 느꼈다. 또한, 아직 다른 팀원에 비해 데이터를 다루고 모델링 하는 면에서 많이 부족하다는 것을 깨달았다. 캠프가 마무리 되는 대로 차근차근 다시 공부해봐야겠다.

4. 공동 학습 측면

이번 프로젝트에서는 주 단위로 스프린트 계획을 짜며 주어진 일을 차근차근 해내는 형태였는데 확실히 전체적인 스케줄 조정하는 데는 좋은 방식인 것 같았다. 중간에 주제를 바꾸게 되어 근 2-3주동안 다같이 영혼을 갈아 넣었고, 스프린트 계획이 쌓여갈수록 부담을 느끼기도 했지만, 이러한 예외 상황이 없다면 스프린트 계획 방식은 앞으로의 프로젝트에도 도입하면 좋을 것 같다.

5. 한계와 아쉬움

짧은 시간 내에 데이터 수집부터 배포까지 하다보니 모델 작업에는 많이 시간 투자하지 못했던 점과 계획대로 수행하지 못했던 점이 아쉬웠다. 특히 그 주의 계획을 마무리하지 못한 채 주말이 와서 매번 주말에도 프로젝트를 했었는데, 이에 따라 체력 관리도 무너졌던 것 같다. 앞으로 이렇게까지 밤샘 일이 있을까 싶긴 하지만 무엇을 하든 최대한 순집중 시간을 늘려서 그날의 일을 마무리할 수 있도록 노력해야겠다.

6. 앞으로의 도전

이번 프로젝트에서 전체적인 파이프라인을 경험했던 것을 바탕으로 앞으로도 크고작은 프로젝트들을 진행하고 싶다. 특히 나와 거리가 멀다고 생각했던 프론트 작업을 하면서 새롭게 도전하는 일을 회피하지 않아야겠다고 다짐했다. 또한, AI 관련해 부족한 부분을 최대한 채우기 위해 지금까지 배웠던 것을 복습하고 취업을 위해 달려 봐야겠다.

김희수_T7513

1. 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

프로젝트에 사용할 데이터를 얻어왔더라도 서로 다른 두 데이터를 연결시키는 것은 어려운 일이다. 두 데이터가 아무리 실제 데이터라고 해도 한 쪽 데이터를 기준으로 연결시켜버리면 연결되어버린 데이터는 신뢰성을 잃어버리곤 한다. 따라서 이번 학습목표는 데이터를 중점으로 삼아 프로젝트를 수행하였다.

2. 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

Kolmogorov-Smirnov 검정을 통하여 두 분포가 동일한지에 대한 검정을 하였고, PCA를 사용하여 고차원인 데이터를 저차원으로 변환하였다.

3. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

Kolmogorov-Smirnov 검정같은 경우는 분석에 사용하는 두 개의 가격에 대해 검정을 하여 두 분포가 동일하지 않다는 결론이 나왔다. 이를 동일하다는 결론이 나올 때까지 시간을 많이 소모하며 가공하였지만 결국 동일하지 않다는 결론으로 넘어가는 체로 끝나 상당히 아쉬웠다. 또한 Matching Score를 만들 때도 코딩 구현하는 실력이 부족하여 생각해놓은 부분을 반영하지 못했다.

4. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

다음 프로젝트를 하기 전까지는 코딩과 전공을 공부를 할 것이다.

노근서_T7514

1. 나의 학습 목표는 무엇이었나? 달성하기 위해 무엇을 어떻게 했는가?

부스트캠프에 처음 지원한 이유가 떠올랐다. **"End-to-End 경험"**. 업스테이지와의 기업 해커톤이 확정되면서 이를 실천할 기회가 다가왔다. 처음 해보는 일이 필연적으로 발생할 수밖에 없었고, 프론트엔드/백엔드 어떤 역할을 맡든 **"주제와 팀 내에서 1인분을 할 수 있도록 하는 것"**이 첫 번째 목표였다. 그 다음으로는 **"데이터나 모델을 탐구하는 데 소홀하지 않기"**였다. 역시 예상대로 처음 하는 일에는 시간이 더 많이 소요될 수 밖에 없었고, 가능하면 매일 AI 관련 작업을 지속할 수 있도록 병행 작업을 진행했다.

2. 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

살면서 처음으로 프론트엔드 개발을 시작해 보았다. React 프로젝트를 관리하며 프론트엔드 개발까지 처음 해 보니 혼돈 그 자체였지만, 걸음마를 떼듯 시작하다보니 어설피지만 구색은 갖춘 웹 페이지를 완성할 수 있었다.

2. 마주한 한계는 무엇이며 아쉬웠던 점은 무엇인가?

한 달이라는 기간 동안 데이터 수집부터 서비스 배포까지 End-to-End 과정을 경험하면서, 짧은 시간 내에 결과물을 도출하기 위해 스스로 만족도를 포기해야 하는 상황이 온다는 점을 배웠다. 어쩌면 trade-off 관계이거나, 내 능력 부족으로 인해 생긴 문제일 지도 모르겠다.

5. 내가 해본 시도 중 어떤 실패를 경험했는가? 실패 과정에서 어떤 교훈을 얻었는가?

“더 좋은 추천을 위해 어떤 점들이 반영돼야 할까”를 요구사항 정의부터 모델 설계까지 많이 고민했다.

스킬 점수라는 개념을 도입하면서 단순히 유저-아이템 스킬의 정량적 일치보다 정성적 일치를 보장하려 했다. 이를 위해 스킬을 인코딩/임베딩 후 가중치로 적용하려 했지만, 정작 설계한 FM 모델을 학습하는 단계에서 이를 효과적으로 활용하기 어려운 문제에 부딪혔을 때 유독 아쉬움이 남았다.

이전까지는 정제된 데이터를 받아서 대회 형식의 프로젝트를 진행했다면, 이번 프로젝트는 주제에 맞는 데이터셋을 탐색하고, 가공하고, 타겟 변수가 없어 정답지까지 만들어서 평가해야 하는 상황이었다. 가상 데이터셋과 정답지를 만든 후, 이를 학습하는 모델의 평가 지표를 어떻게 신뢰할 수 있는지 설득하는 것이 중요한 과제였다. 이를 뒷받침하기 위해선 위에서 고민했던 지점들이 잘 반영됐어야 했다고 느꼈다.

하지만 기업 멘토님의 피드백을 통해, 이런 문제는 현업에서도 자주 경험하는 영역이라는 점을 알게 됐다. 결국 실무에서 고민해야 할 지점을 미리 경험해 봤다는 점에서 의미있는 과정이었다고 위로받을 수 있었다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

End-to-End 프로젝트를 경험하면서 느낀 건 “내가 담당할 업무 외의 분야도 한 번 이상 경험해 볼 필요가 있다”는 것이다. 기존의 나는 한 우물을 깊게 들여다보고 탐구하는 사람이었다. 돌아보면 부스트캠프에 들어오면서 새로운 영역을 많이 마주하며 이전처럼 우물 안 개구리가 되지 않게 (나란 사람치고) 부단히 움직였던 것 같다. 이번 프로젝트에서도 마찬가지로 새로움의 연속이었다. 그럼에도 내가 아직 경험하지 못한 부분이 많기 때문에 다음 프로젝트에서도 또 다른 세계에 발을 들여볼 것이다.

박영균_T7520

1. 나의 학습목표는 무엇이었나?

뭔가 배워서 나가고 싶었다.

그게 꼭 AI 관련이 아니어도 된다고 생각했다. 오히려 이번 부스트캠프에서 AI와 관련해 내가 과연 이 분야에서 역경을 잘 헤쳐 나갈 수 있을지 고민을 많이 했었기 때문에 더욱 그런 것 같기도 하다. 적어도 해커톤을 진행하면서 남들만 못했다는 말은 듣지 말아야겠다 라고 생각했다. 내 모토처럼 “1.1인분 하는 사람”이 되려고 노력했던 것 같다.

2. 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

(아마 살면서 처음으로) 팀장 역할을 자처해서 해봤다.

내가 크게 리더십이 있다고 생각하는 편은 아니었지만, 어쨌든 기업은 리더십을 원하고, 내가 못 한다고 시도도 안 해버리면 될 것도 안 된다는 생각이 들었다. 끝나고 나서 생각해봤을 때 리더로서의 역할을 제대로 했는지 잘은 모르겠지만, 그래도 비전공자로서 처음 접하는 부트캠프에서 리더 역할을 맡아봤다는 것이 나에게는 새로운 경험이 될 것이고 앞으로의 자신감이 될 거라는 생각이 든다.

3. 개인적으로 어떤 것을 배웠는가?

내 커리어에 있어 방향이 하나가 아닐 수 있겠다는 생각을 하게 되었다.

부스트캠프를 AI Tech로 오기는 했지만 마지막 프로젝트에서 프론트엔드 개발을 해 보며 새로운 즐거움을 얻었다. 물론 거의 다 GPT를 이용해서 짠 코드이기는 했지만, 프로젝트가 끝나고 나서 제대로 프론트엔드 관련 공부를 해보면 어떨까 하는 생각도 들었다. 사실 내가 원했던 건 굳이 AI이어야만 했던 게 아니라 코딩을 하면서 뭔가를 시뮬레이션 돌려보고 원하는 대로 될 때까지 시도해보는 과정이었다는 것을 깨닫게 해 준 프로젝트가 이번 프로젝트였다.

4. 협업 측면에서 얻은 것이 있는가?

생각보다 사람은 잘 맞는다고 생각하는 사람도 사실은 그렇지 않을 수 있다는 것을 알았다.

누군가와 일적으로 잘 맞는다는 생각이 드는 것은 그 때 내가 여유로웠기 때문이다. 이번 프로젝트가 내 협업 경험 중에 가장 힘들다고 생각이 들었는데, 극한의 스트레스를 받는 상황이 되니 언성도 높아지고 나의 기질적인 면이 자주 나타나는 것이 느껴졌다. 이 사람은 왜 이럴까? 하며 생각하는 것보다 이 사람은 이런 상황에서 이렇게 행동하는 사람이구나 라고 생각하며 어떻게 하면 이 갈등을 해소할 수 있을지를 찾아보는 마음가짐으로 협업에 임해야겠다고 다짐했다.

5. 프로젝트 중 기억에 남는 것이 있는가?

기대보다 더욱 잘 따라쫓던 팀원들의 모습이 인상적이었다.

내가 생각하는 나는 꽤나 어리숙하고, 길을 제시하는 측면에 있어서는 늘 부족하다고 생각하고 있었다. 그런 면이 아쉬워서 이번에 팀장 역할을 맡아본 것도 있었다. 그런데 내가 운이 좋아서 그런 것일지는 모르지만, 생각보다도 팀원들이 자신의 역할을 열심히 해 준 것 같다. 왠지 내가 팀장이 아닌 것 같다는 생각이 들 정도로. 내가 팀에서 가장 막내뻘에 속하는 사람인데도 불구하고 그런 것에 대한 의식 없이(적어도 내가 느끼기에는) 내 지시에 선뜻 움직여주고 잘 해내주는 모습에서 자신감을 꽤나 얻었다. 또 그럼에도 내가 원하는 방향과 다른 의견을 가진 사람은 거리낌없이 이야기해주는 모습이 더욱 완성도있는 프로젝트를 위해 내가 할 일이 무엇인지를 일깨워주었다.

6. 프로젝트에서 아쉬웠던 점은?

무엇보다도 시간이 부족한 것이 매우 아쉬웠다. 약 한 달이라는 시간 안에 데이터 수집부터 배포까지 모든 것을 완료해야 했기 때문에 포기한 것들이 너무 많았다. 특히 모델과 관련해 너무 빈약하다고 생각이 들 정도로 신경을 쓰지 못했다. 하지만 돌이켜 생각해 보았을 때는 조금 달랐다. 우리가 애초에 목표로 했던 것이 End-to-End의 경험이었고, 그런 측면에서 보았을 때 우리는 그 목표를 누구보다도 잘 해냈다고 생각이 든다. 모델링에 힘을 쏟지 못한 기회비용과 내가 새롭게 배운 내용을 비교해봤을 때 내가 얻은 것이 훨씬 많다고 생각해 후회는 없다. 아쉬움은 아쉬움일 뿐이라고 생각한다.

7. 다음 프로젝트에서 시도해볼 것은?

앞으로 있을 프로젝트에서도 지금과 같은 마음가짐이면 될 것이라 생각한다. 말하자면 현상 유지만 해도 잘 해낼 수 있을 것이라고 본다. 확실히 부스트캠프 전보다 끝난 후의 나의 모습이 나아졌음을 느끼기 때문에, 앞으로 있을 프로젝트에서도 이렇게만 한다면 훨씬 더 좋은 결과물을 만들 수 있을 것이며 지금보다도 더 많이 성장할 수 있을 것이라 믿는다.