PROJECT WRAP UP, RecSys-5

boostcamp

알아서 잘 딱 깔끔하고 센스있게.

RecSys-05

Team 소개

알아서 잘 딱 깔끔하고 센스있는 엔지니어 팀

> Naver Connect BoostCamp AI-Tech 3기
> 추천시스템 5조 알잘딱깔센
>
> P-stage-2 프로젝트
> 2022.04.18 ~ 2022.05.12 19:00 (4주)

우리팀의 목표

" 성능보다 성장, 혼자보다는 함께 "

- 프로젝트를 진행하면서 **협업 역량**을 키운다.
- 학습 내용 공유를 통해 **함께하는 성장**을 이룬다.
- 추천 시스템에 필요한 **엔지니어적 역량**을 기른다.
- 필요한 지식 및 기술을 찾아 **적용하는 역량**을 갖춘다.

목차

저희는 알아서 잘 딱 깔끔하고 센스있게 추천하는 엔지니어 팀입니다!

🚀 프로젝트 개요

주제 : DKT(Deep Knowledge Tracing)

유저의 문제 풀이 Sequence를 이용하여 유저의 지식 상태를 추론해, 유저가 마지막에 푼 문제를 틀릴지 맞출지를 예측하는 것

데이터 개요

- 7,442명의 유저 (user)
- 9,454개의 문항 (assessmentItemID)
- 시간 순서로 나열된 유저의 문제 풀이 Sequence
- 유저의 문제 풀이 수는 최소 9 ~ 1,860개
- 하나의 문항에 대해 시험지, 지식 태그, 문제를 푼 시간, 정답 여부 등의 정보가 주어짐

활용 장비 및 재료

개발환경
- Ubuntu 18.04.5 LTS
- GPU Tesla V100-PCIE-32GB

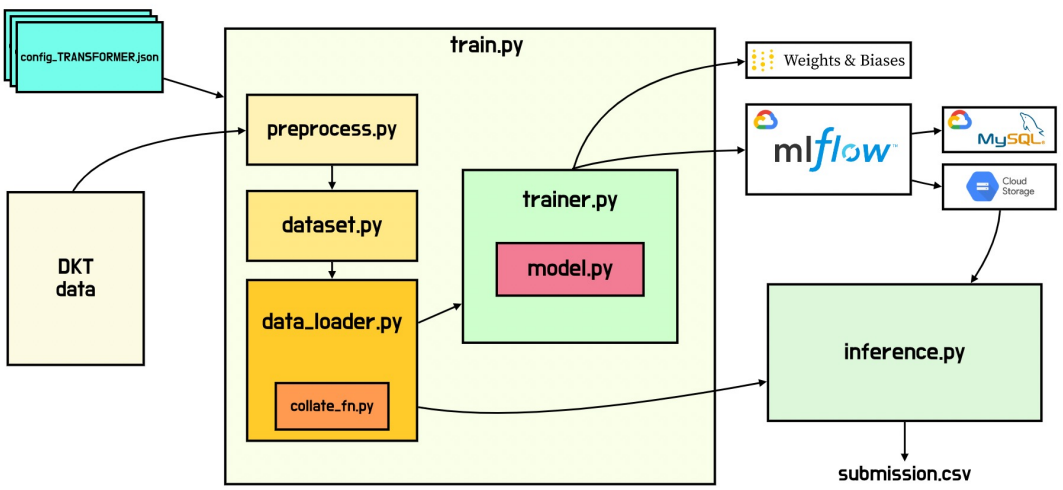
Tools
- python 3.8.5
- pytorch 1.10.2
- MLflow 1.24.0
- Weights & Biases

Collaborate
- Github
- Jira
- Gather Town

👨🏻‍🦽 프로젝트 팀 구성 및 역할

김건우: EDA를 통한 feature 탐색, Project Template 탐색
김동우: 베이스 모델 탐색
박기정: EDA를 통한 Feature 탐색, Project Template 설계 및 제작, MLflow model registry 구축
심유정: EDA를 통한 Feature 탐색, 베이스 모델 탐색, LGBM 모델 학습 및 Ensemble
이성범: 데이터 패턴 분석, 모델 설계 및 학습, Ensemble, Project Template 제작

⚒️ 프로젝트 템플릿 제작



https://github.com/victoresque/pytorch-template 기반으로 제작

- 학습 환경의 경우 pytorch-template(https://github.com/victoresque/pytorch-template)을 이용하여 **DKT 학습환경에 맞추어 리팩토링을 진행함**
- 각 모듈들은 **추상클래스를 이용하여 객체지향적으로 구현**
- 학습 결과를 **wandb를 이용하여 시각화** 하도록 함
- **학습한 모델의 경우 mlflow를 이용하여 버저닝 관리** (backend - MySql 이용, artifact - Google Cloud Storage 이용)

프로젝트 절차 및 방법

✧ EDA

- **문항별 풀이 시간** : 풀이 시간이 짧을 수록 틀릴 확률이 더 높아짐
- **유저의 문제 풀이 경향성** : 해당 어플을 장기간 이용하는 사용자는 그렇게 많지 않은 것으로 보임
- **문제를 푸는 시간대별 정답률** : 7시부터 14시까지 증가하다 점심(14시) 이후 감소하며, 11시 이후 다시 증가
- **최근 문제 정답률에 대한 가중치 반영** : 최근 문제 정답률에 대한 윈도우 사이즈가 커질 수록 마지막 문제에 대한 정답 여부를 예측하기 어려워짐
- **같은 문제 시간을 두고 중복해서 푼 사람들 존재** : 같은 문제를 시간을 두고 반복해서 풀이한 사람 존재 (최대 3번) 다시 풀었을 때 (오답->정답) (정답->정답) (오답->오답) (정답->오답) 등 모든 경우 존재

✧ 프로젝트 방향성

- **모델의 특성에 기반하여 데이터의 패턴을 분석함**
- 분석한 결과를 바탕으로 데이터의 패턴을 효과적으로 표현할 수 있는 **Model Architecture를 설계**함
- 유저 데이터 부족 문제를 해결하기 위해 **모델의 일반화 성능을 높이기 위한 학습 방법을 고민함**

✧ 검증 전략

- 유저를 기준으로 학습과 검증 데이터셋을 구축하여 8:2의 비율로 교차 검증을 진행
- 검증 데이터의 경우 유저의 마지막 문항에 대한 정보 만으로 구성하여, 학습 시에는 검증 데이터를 제외한 모든 유저의 학습 정보를 사용
- **유저를 기준으로 데이터셋을 분리함으로써 일반화된 환경에서 모델 평가가 가능해짐**

✧ 모델을 활용한 데이터 패턴 분석

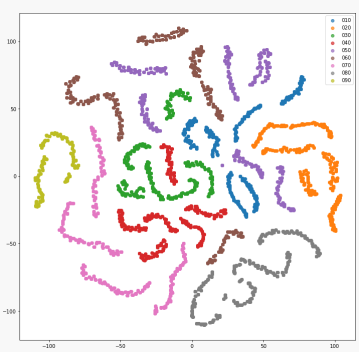
GMF

- 유저 임베딩을 사용하면 모델이 과적합이 발생한다는 것을 확인
- 문항에 대한 변수를 추가할 수록 모델의 성능이 향상된다는 것을 확인
- 즉, **유저보다는 문항 정보를 활용하는 것이 더 중요**

LightGCN

- 유저-문항을 그래프 형태로 표현하여, 유저-유저 연관성을 기반으로 데이터를 표현할 시에 성능이 좋지 않다는 것을 확인
- 즉, **유저-유저의 연관성을 바탕으로 본 데이터를 표현하는 것은 매우 어려움**

Item2Vec



Word2Vec 학습 결과

- 유저의 문제 풀이 내역을 기반으로 문항을 임베딩 했을 때 특정 군집을 이룸
- 즉, **유저의 문제 풀이 내역에는 특정한 패턴이 존재**한다는 것을 알 수 있음

⚙️ 프로젝트 절차 및 방법

✧ 모델을 활용한 데이터 패턴 분석

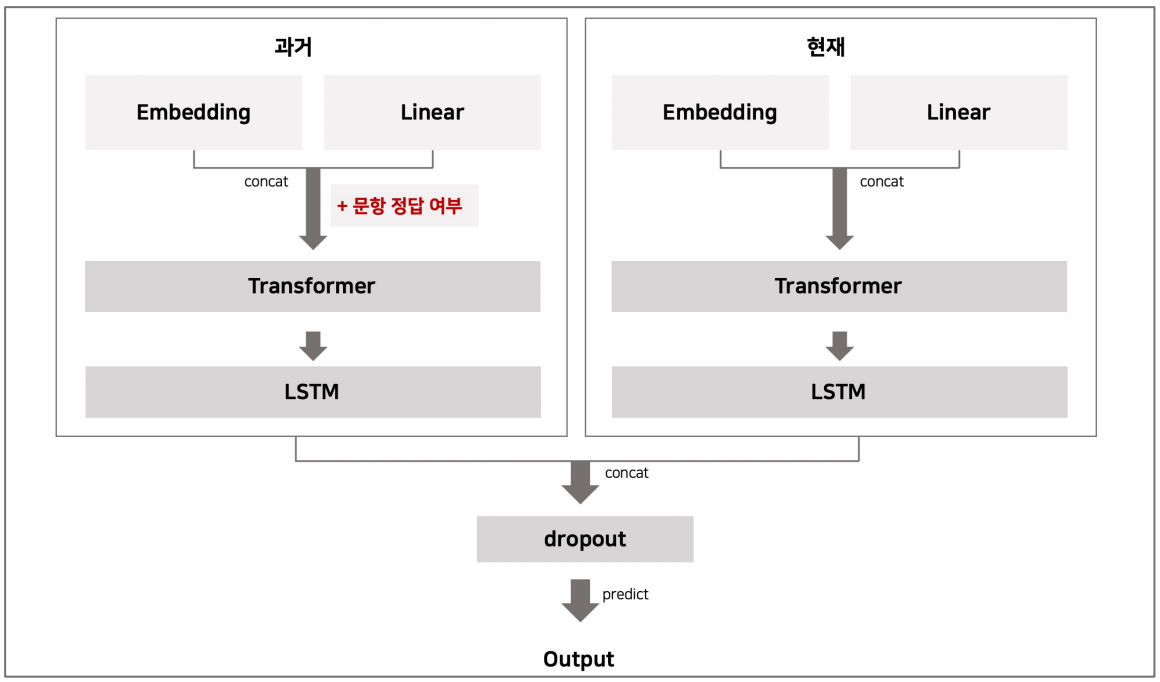
BERT(0.78) vs Transformer(0.83)

- 문제 풀이 내역을 양방향으로 학습하는 BERT보다 단반향으로 학습을 하는 Transformer의 성능이 더 좋음
- Trasformer을 가지고 단순히 문항 만을 이용하여 다음에 등장할 문항을 예측 하더라도 높은 ACC를 보임
- 즉, **유저의 문제 풀이 순서는 매우 중요한 패턴이 내재**되어 있다는 것을 알 수 있음

💡 insight

- 유저를 표현할 수 있는 데이터의 수가 매우 제한적이기 때문에, 유저를 임베딩 하는 것보다 **최대한 문항 정보를 활용하여 parameterized function을 만드는 것이 중요**
- 유저의 문제 풀이 순서에는 매우 중요한 패턴이 내재되어 있기 때문에, **시간적 순서를 효과적으로 표현할 수 있는 Model Architecture 설계가 중요**

✧ Model Architecture 설계



- 과거 풀이 정보의 경우 정답 여부를 알 수 있지만, 현재 풀이 정보의 경우 정답 여부를 모르기 때문에, 이에 표현될 수 있는 정보가 다를 수 있다고 **생각하여 과거와 현재 풀이 정보의 연관성을 표현할 수 있는 Model Architecture를 설계함**
- 범주형 변수의 경우 Embedding Layer로, 수치형 변수의 경우 Linear Layer를 이용해 Embedding하고, 두 변수를 concat해 문항에 대한 Embedding을 구함
- **시간적 순서를 효과적으로 표현하기 위하여 Transformer와 LSTM을 활용함**
- **과거 풀이 정보와 현재 풀이 정보를 서로 다른 Embedding을 활용해 학습**시킴으로써 non-convex 한 목적 함수를 조금 더 convex하게 만듦

프로젝트 절차 및 방법

✧ 모델의 일반화 성능을 높이기 위한 학습 방법

Representation Learning

- 데이터의 표현을 효과적으로 학습할 수 있는 Model Architecture를 설계하여 **최소한의 변수로 최대의 성능을 이끌어냄**
- 범주형 - 문항, 시험지, 태그, 시험지 대분류, 시간, 요일
- 숫자형 - 정답률의 평균 / 표준편차, 풀이 시간의 평균 / 표준 편차

Loss

- 마지막 문제에 대해서만 Loss를 계산하면, 모델은 한번 학습시 유저 개수 만큼의 데이터 밖에 활용할 수 없음
- 이에 **전체 Time-step에 대하여 Loss를 계산하여 유저 데이터 부족 문제를 해결**

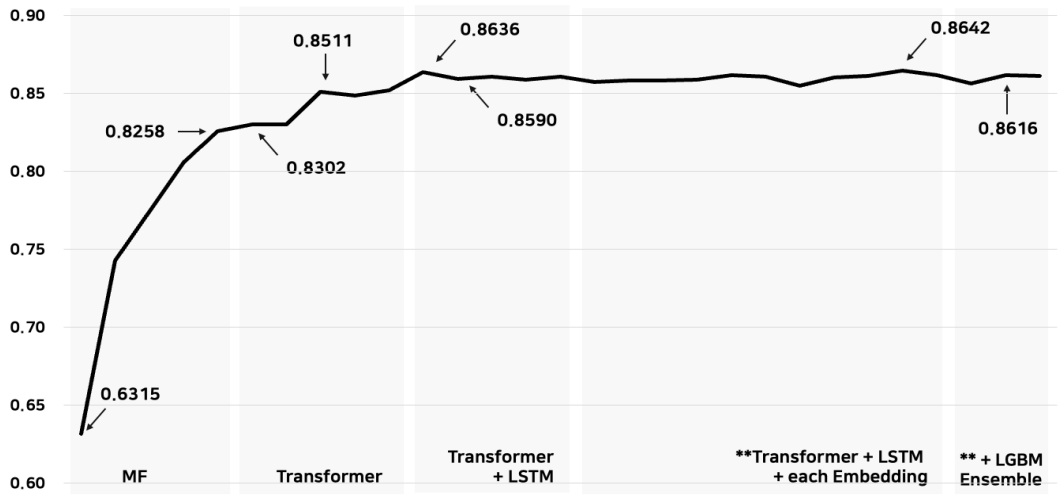
Padding

- Max-len과 mean-len의 차이가 크기 때문에, Max-len으로 동일하게 Padding을 하게 되면 의미 없는 학습과 모델 학습 시간이 매우 늘어난다는 단점을 가짐
- 이를 해결하고자 **배치 마다 서로 다른 크기의 padding을 두어 모델을 학습 시킴**

Ensemble (soft-voting)

- Transformer에 사용된 변수에 **elo score을 추가하여 LGBM 학습**
- Transformer와 LGBM의 경우 학습 방법이 다르기 때문에, 결과에 다양성을 향상시킬 수 있다고 생각하여 Ensemble을 진행

프로젝트 수행 결과 (3위)



*최종 점수 기준

- **유저와 문항 정보를 함께 사용한 GMF 모델**을 통해서 0.7429의 성능을 얻음
- **문항에 대한 정보만을 활용한 GMF 모델**을 통해서 0.8258의 성능을 얻음
- **시간적 순서를 고려한 Transformer**을 통해서 0.8302의 성능을 얻음
- **Transformer와 LSTM을 함께 사용**하여 0.8511의 성능을 얻음
- **과거 정보에 정답에 대한 Embedding을 추가**하여 0.8636의 성능을 얻음
- **과거 정보와 현재 정보를 같이 Modellig** 하여 0.8590의 성능을 얻음
- **과거 정보와 현재 정보를 서로 다른 Embedding을 활용**해 학습하여 0.8642의 성능을 얻음
- Head Ensemble을 진행한 **Transformer와 LGBM의 결과를 soft-voting**하여 0.8616의 성능을 얻음

🙋‍♀️ 자체 평가 의견

✧ 잘한 점들

- '성능 보다는 성장, 혼자보다는 함께'를 목표로 **성장과 협업에 중점을 두고 프로젝트를 진행**
- 검증된 프로젝트 템플릿을 이용하여 **더 나은 프로젝트 구조를 설계함**
- **EDA를 팀원 모두가 진행하고 공유**하며 데이터를 탐색함

✧ 시도했으나 잘 되지 않았던 것들

- sequential한 정보를 포함하는 그래프 모델인 SURGE를 적용해보려 하였으나, 템플릿화 된 코드가 복잡해서 적용하지 못하였음
- LGBM을 더 오래 학습시킬 수록 성능이 향상됨을 알았지만, 너무 늦게 시도하여 시간적 여유가 부족했음

✧ 프로젝트를 통해 배운 점 또는 시사점

- 본격적인 모델링 전에 **EDA를 통한 데이터 탐색이 중요**하다는 것을 알게됨
- **잘 설계된 모델 Architecture를 통해 Feature Engineering을 많이 하지 않고도 비슷하거나 더 뛰어난 성능을 보일 수 있다는 점**
- **Weight & Bias** 툴을 사용하는 경우 MLFlow에서 그래프로 모델간 성능 비교를 할 수 없다는 점과 Tensorboard의 실험 내용 공유를 할 수 없다는 점을 보완할 수 있다는 점

✧ 아쉬웠던 점들

- Jira나 Confluence와 같은 새로운 일정 관리 도구를 잘 사용하지 못한 점이 아쉬움
- 최종 프로젝트에 할당된 기간이 짧아 DKT 대회에서 다양한 시도를 해보지 못함

Recsys5 – 알잘딱깔센 개인 회고

T3015 김건우

내가 시도한 기술적인 도전

○ 다 같이 성장하는 프로젝트를 만들기 위해 팀원들간 정리와 공유를 활발하게 하기 위해 노력하였다.

○ 파이토치 템플릿을 분석하며 전체적인 머신러닝 파이프라인 프로세스를 경험하려고 노력하였다.

○ EDA를 통한 데이터 탐색을 중점으로 하여 다양한 피처들을 만들어내는데 노력하였다.

- 최근 문제 정답률에 대한 가중치 반영 (최근에 문제를 많이 맞췄으면 마지막 문제를 더 잘 맞췄을까?)
  - 최근 문제 정답률에 대한 윈도우 사이즈가 커질 수록 마지막 문제에 대한 정답 여부를 예측하기 어려워진다. 하지만 윈도우 사이즈 5에서 gt_50_answerCode_1이 가장 큰 것으로 보아 적정 윈도우 사이즈가 5일 가능성이 있다.
- 마지막 문제의 태그가 주는 영향력
  - 가장 많이 풀린 태그와 비교했을 때 마지막 문제와 같은 태그를 갖는 문제들이 풀린 횟수와 마지막 문제의 정답률 사이에는 상관관계를 찾을 수 없었다.
  - 기준을 바꿔 가장 많이 푼 태그와의 비교가 아닌 절대적인 개수로 비교했을 때도 딱히 어떤 상관관계를 찾기는 힘들었지만 25개일 때 정답률이 높아지는 현상이 관찰되어 이를 탐색하였다.
  - 해당 데이터를 탐색해보았더니 이는 단지 푼 문제의 대략 90%가 마지막 문제의 태그와 같았기 때문이었다.
  - 따라서 마지막 문제와 같은 태그를 지닌 문제를 많이 풀었다고 정답여부에 영향을 준다기보다는 보다 전체적으로 보았을 때 특정 문제와 같은 태그를 많이 풀수록 해당 문제를 더 잘 맞힐 가능성이 있다

학습과정에서의 교훈

○ 이번 대회에서 가장 크게 느낀 부분은 공유와 멘탈 관리였다.

○ 공유

- 초반 깃허브 브랜치 전략을 논의하여 실험과 개발 브랜치로 나누어서 코드를 관리하였다.
- 이번에는 새로운 도구인 Jira를 사용하여 일정을 관리하였다.
- 특히 각자가 찾은 지식과 정보를 Confluence에 기록하며 정보를 공유를 조금 더 원활히 하였다.
- 스프린트나 칸반과 같이 애자일 방법으로 일정을 관리할 수 있어서 좋았다.
- 하지만 우리가 진행하고 있는 태스크의 크기가 작아서인지 제대로 활용하지 못해 아쉬웠다.
- 개인적으로는 AiStage 토론 게시판에 EDA 한 것을 잘 정리하여 많은 좋아요 수를 기록하였다.

○ 멘탈 관리

- 저번 프로젝트에서 너무 참여도가 적었다는 생각에 이번 프로젝트에선 큰 기여를 해보자는 생각으로 대회에 임했다.
- 초반 1~2주는 EDA를 통해 토론 게시판에 공유하기도 하고 팀원들과 많은 토의를 하면서 피처를 발전시키기 위해 노력하였다.
- 하지만 본격적으로 파이토치 템플릿을 이용하여 파이프라인을 구축하면서부터 집중력이 급격히 떨어져서 제대로 된 기여를 할 수 없었다.
- 이 시기 부스트캠프에 대한 회의가 몰려와서 약 2주동안 코딩을 놓았었다.
- 팀원들에게 배울 수 있는 것을 모두 배우자는 생각으로 슬럼프에서 조금 씩 벗어날 수 있었던 것 같다.
- 이를 바탕으로 얻은 가장 큰 교훈은 팀원들과 함께하는 성장을 중요하게 생각하면서 항상 배우는 자세를 지니자는 것이다.
- 항상 배우는 자세를 통해 모든 것에 호기심을 가지고, 팀원들을 통해 새로운 것을 흡수하려고 한다면 힘든 과정도 즐기면서 갈 수 있다는 것을 깨달았다.

Recsys5 – 알잘딱깔센 개인 회고

T3025 김동우

나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

○ JIRA를 통해 일정을 관리하며 프로젝트를 진행하였으며, confluence를 통해 실험 내용과 insight를 공유하였습니다

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

○ 특정 방법을 시도해 보기 전에 자료를 정리해서 팀원들과 공유하고 발표 하는 방식을 통해, 배운 것을 공유할 수 있었고, 생각의 근거가 있는지, 어떤 방식으로 시도하면 좋을지 피드백을 듣고 개선할 수 있었습니다.

나는 어떤 방식으로 모델을 개선했는가?

○ 베이스라인 모델(Light-gcn)에서 주요 파라미터를 변경 시키며 성능을 측정하는 방식으로 모델을 분석 하였습니다.
○ 기존 베이스라인 모델을 개선한 모델의 논문을 읽고 모델을 적용시켜 성능을 개선하려고 시도 하였습니다.
○ 게임에서 주로 사용되는 ELO라는 상대적인 실력을 점수로 나타낸 지표가 주요 feature로 사용될 수 있다고 생각해서 모델에 적용하려 하였습니다.

마주한 한계는 무엇이며,아쉬웠던 점은 무엇인가?

○ 특정 모델과 방법을 적용해서 성능을 개선 시킬 수 있을 지도 중요하지만, 내가 실제로 완성해서 적용할 수 있는 난이도인지, 기간 안에 완성할 수 있는지 파악하는 것이 중요하다고 느꼈습니다.
○ ELO를 주요 feature로 적용하는 것의 타당성과 어떻게 적용할지 대략적으로 분석 하였으나, 스스로 직접 적용하지는 못했던 점이 아쉬웠습니다.

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?

○ light gcn 기반의 그래프 모델이 transformer 기반 모델에 비해서 성능이 잘 나오지 않는 것을 바탕으로, DKT 프로젝트 에서 문제의 sequential한 정보가 중요한 정보라고 생각 하였습니다.
○ sequnetial한 정보까지 포함하는 그래프 모델인 SURGE를 적용해 보려 하였으나, 모델이 light-gcn에 비해서 많이 복잡하고, 템플릿화 된 코드가 너무 복잡해서 적용하지는 못하였습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

○ small-to-large 방식은 진리라는 것을 다시 한번 느꼈습니다.내 현재 실력에 비해서 해야할 목표가 어려울수록, 적용할 수 있고, 간단한 방법의 코드부터 일단 만들고, 점차적으로 복잡하게 해서 완성시켜야 한다고 느꼈습니다.

Recsys5 – 알잘딱깔센 개인 회고

T3084 박기정

나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

○ 팀의 개발 능력을 향상 시킬 수 있도록 좋은 베이스라인 코드를 작성하고 이에 모델을 하나씩 추가해 가면서 각자 객체지향적 사고 능력이 상승 할 수 있도록 노력하였다.
○ 지난 mlflow를 실험 결과 공유만 할 수 있도록 하였는데 이번에는 모델 버저닝까지 될 수 있도록 하였다.

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

○ 지난 번엔 내가 직접 템플릿을 제작하였는데 이번에는 기존의 검증된 템플릿을 사용하여 metric이나 collate function 등을 추가적으로 변경 가능한 부분을 추가하여 좀 더 유연하고 완성도 높은 템플릿을 사용할 수 있도록 하였다.
○ Mlflow 모델 레지스트리를 이용하여 모델 버저닝이 가능하도록 하여 좀 더 Mlops 기능을 향상시켰다.

나는 어떤 방식으로 모델을 개선했는가?

○ 단순히 프로젝트 템플릿을 그대로 사용하는 것이 아니라 우리 프로젝트에서 사용하는 모델에 따라 템플릿을 맞추어 리팩터링을 진행하였다.
○ 또한 불필요한 요소들을 제거하여 템플릿의 복잡성을 낮추었다.

마주한 한계는 무엇이며,아쉬웠던 점은 무엇인가?

○ DKT 프로젝트와 최종프로젝트를 동시에 진행하여 좀 더 DKT 프로젝트에 집중할 수 없었던 점이 너무 아쉬웠다.

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?

○ 프로젝트 템플릿을 잘 작성함에 팀원들의 코드리딩이 빨라지고, 다양한 툴을 추가함에 추상클래스에 추가함으로서 모델의 범용성을 높이고 코드 중복성을 낮추었다.

한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

○ 프로젝트 템플릿 뿐 아니라 전반적인 mlops 와 머신러닝 아키텍처를 완성도 있게 진행해 보고 싶다.

Recsys5 – 알잘딱깔센 개인 회고

T3122 심유정

나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

○ 우리 팀과 나의 학습목표는 무엇이었나?
- 리더보드 1등보다는 많은 것을 시도해보고, 성장하기
- 함께 협업하기

○ 개인학습 측면
- 데이터에 대한 다각도의 탐색(EDA)
- 모델이 예측하는 결과의 다양성을 위해 LGBM 모델 앙상블

○ 공동 학습 측면
- 그라운드 룰을 정해 체계를 갖추기
- 서로 학습한 내용 공유를 통해 모두가 이해하기
- 대면 회의 및 주기적인 의사소통으로 방향성 잡기
- Jira와 Confluence를 통한 일정, 자료 관리

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

○ 이전 프로젝트에서 아쉬웠던 부분 두 가지를 개선하였음

○ 조금 더 다각도의 EDA를 진행하기
- 이전 대회들에 비해 EDA를 제대로 해보고자 하였으며, 팀원들 모두가 각자 탐색한 결과들을 공유함으로써 데이터를 더 잘 이해할 수 있었음

○ 특성이 다양한 모델을 앙상블하기
- 지난 대회 때 static한 모델만을 앙상블해서 sequence를 잡지 못했던 아쉬움이 있는데, 이번에는 Feature를 최소화하여 학습한 Transformer와 다양한 Feature를 학습시킨 LGBM을 앙상블 해보았으며, 리더보드 상 성능이 향상됨을 확인함

나는 어떤 방식으로 모델을 개선했는가?

○ 여러 파생변수를 만들고, 그래프를 통한 데이터 탐색

○ 베이스 모델로 주어진 LGBM에 다양한 Feature들을 추가하여 모델 성능 향상

마주한 한계는 무엇이며,아쉬웠던 점은 무엇인가?

○ 최종 프로젝트의 기간이 여유롭지 않아 대회 기간 중반 이후에는 최종 프로젝트에 집중하였음

○ 이에 따라 다양한 시도를 해보지 않았음이 아쉬웠고, 이에 따라 깃허브나 Jira도 제대로 사용하지 못한것 같아 아쉬움이 남음

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?

○ Transformer와 LGBM 앙상블을 통해 리더보드 점수가 향상함을 확인할 수 있었고, 이를 통해 처음에 가정했던 예측 결과의 다양성을 주었을 것이라고 예상함

한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

○ 이전 프로젝트들과 다르게 각자의 파트와 롤을 명확하게 나누었다는 점과 주어진 task를 경쟁 형태로 수행하는 것이 아니라는 점에서 조금 더 맡은 역할에 책임감을 가지고 다양한 시도를 해보고자함

○ 전체적인 흐름을 알고, 그 흐름과 일정들을 관리하기

Recsys5 – 알잘딱깔센 개인 회고

T3152 이성범

나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

○ 우리 팀은 '성능 보다는 성장, 혼자보다는 함께'를 목표로 성장과 협업에 중점을 두고 프로젝트를 진행했고, 개인적으로도 정리와 공유를 목표로 잡았다.
○ 나는 다 같이 성장하기 위해서 내가 알고 있는 지식을 최대한 공유할 수 있도록, 깃허브의 PR과 Issue를 통해서 내가 실험한 내용을 정리했고, 이 내용을 발표함으로써 모두가 같이 성장할 수 있도록 노력했다.
○ 이번에는 철저히 모델 중심적인 사고를 가지고 프로젝트를 진행했고, 모델을 활용하여 이번에 주어진 문제를 해결할 수 있도록 노력했다.

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

○ 모델을 활용하여, 데이터의 패턴을 분석했고, 본 결과를 바탕으로 모델을 설계하는 모델 중심 적인 사고로 본 프로젝트를 진행했고, 다른 팀보다 더욱더 빠른 속도로 좋은 모델을 만들 수 있었다.

나는 어떤 방식으로 모델을 개선했는가?

○ 각 모델의 특성을 활용해 데이터의 패턴을 분석하여 본 Task에 맞는 모델 Architecture를 설계했고, 실험에 근거하여 모델 Architecture를 변형해 나가며 모델의 성능을 개선했다.
○ 전체 Time-step에 대하여 Loss를 계산해 유저 데이터 부족 문제를 해결하여 모델의 성능을 개선했다.
○ 배치마다 서로 다른 padding을 두어 학습 속도와 모델 성능 개선에 기여했다.

마주한 한계는 무엇이며,아쉬웠던 점은 무엇인가?

○ 팀원 간의 모델 역량에 차이가 존재해 모델 실험 시에 많은 아이디어를 공유하지 못한 것이 아쉬움

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?

○ 모델 중심적인 사고를 바탕으로 본 프로젝트를 진행했고, 주로 Ensemble과 feature engineering에 대부분의 시간을 투자한 다른 팀과 다르게, 최소한의 변수로 최대의 성능을 얻는 최고의 단일 모델 Architecture를 만들었다.
○ 데이터를 제대로 표현할 수 있는 모델 Architecture를 만든다면, 충분히 적은 변수로도 최대의 성능을 낼 수 있다는 것을 배웠다.

한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

○ 내가 초반부터 너무 빠르게 모델을 만들고 공유했을 수도 있다고 생각해서, 다음 부터는 팀 전체의 속도를 고려하여 모델을 개발하고 공유해볼까 생각 중이다. 혼자 너무 앞서면 다른 사람들이 따라오기 벅찰 수도 있다는 생각도 들고, 나라도 쉽게 지칠 것 같다.