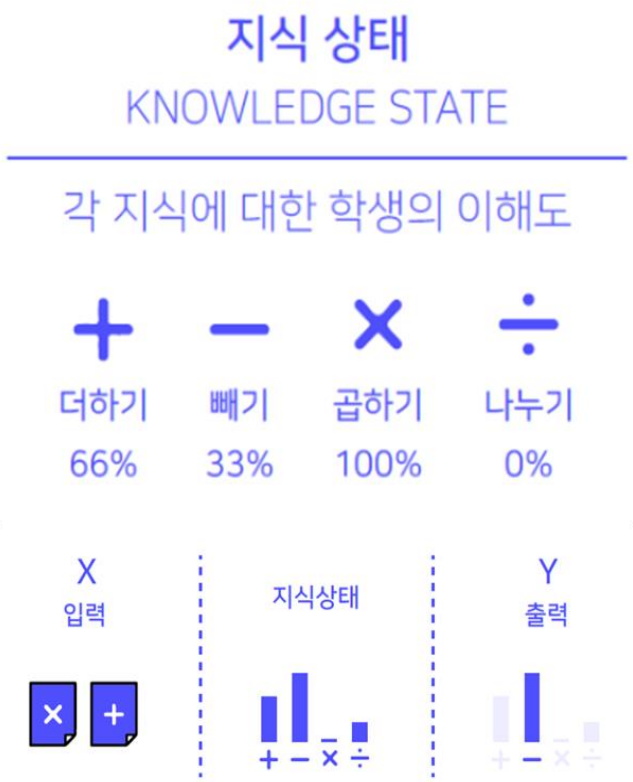
# Deep Knowledge Tracing Competition Wrap up Report

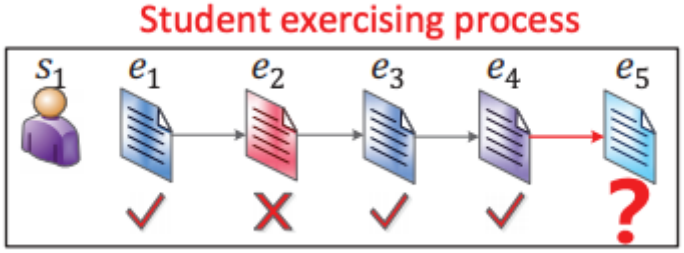
Recsys-13 (Re?LU)

김원섭(T3044), 김진수(T3058), 민태원(T3080), 이상목(T3146), 조민재(T3204)

## 1. 프로젝트 개요




"DKT는 Deep Knowledge Tracing의 약자로 우리의 '지식 상태' 를 추적하는 딥러닝 방법론입니다."




i-Scream 데이터셋을 통해 DKT모델을 구축합니다. 각 학생이 푼 문제 리스트와 정답 여부가 담긴 데이터를 받아 최종 문제의 정오답 여부를 예측하는 것이 목표입니다.

### 1.1 활용 장비 및 개발환경

OS : Ubuntu 18.04.5 LTS, Windows

IDE : VS Code

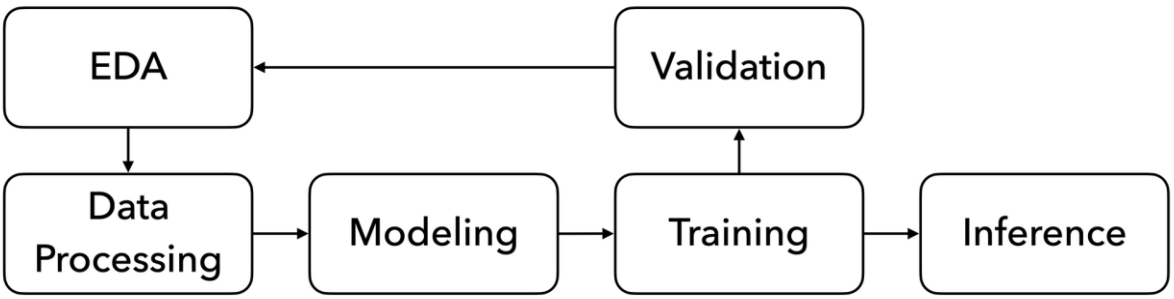
GPU : Tesla V100 *Boostcamp로부터 제공받은 서버

주 사용 언어 : Python 3.8.5

Frameworks : Pytorch, wandb,

Co-op tools : github, notion, slack

### 1.2 프로젝트 및 사용 데이터셋의 구조




| | userID | assessmentItemID | testId | answerCode | Timestamp | KnowledgeTag |
|---|---|---|---|---|---|---|
| 0 | 0 | A060001001 | A060000001 | 1 | 2020-03-24 00:17:11 | 7224 |
| 1 | 0 | A060001002 | A060000001 | 1 | 2020-03-24 00:17:14 | 7225 |
| 2 | 0 | A060001003 | A060000001 | 1 | 2020-03-24 00:17:22 | 7225 |
| 3 | 0 | A060001004 | A060000001 | 1 | 2020-03-24 00:17:29 | 7225 |
| 4 | 0 | A060001005 | A060000001 | 1 | 2020-03-24 00:17:36 | 7225 |
| ... | ... | ... | ... | ... | ... | ... |
| 2266581 | 7441 | A030071005 | A030000071 | 0 | 2020-06-05 06:50:21 | 438 |
| 2266582 | 7441 | A040165001 | A040000165 | 1 | 2020-08-21 01:06:39 | 8836 |
| 2266583 | 7441 | A040165002 | A040000165 | 1 | 2020-08-21 01:06:50 | 8836 |
| 2266584 | 7441 | A040165003 | A040000165 | 1 | 2020-08-21 01:07:36 | 8836 |
| 2266585 | 7441 | A040165004 | A040000165 | 1 | 2020-08-21 01:08:49 | 8836 |

```
RangeIndex: 2266586 entries, 0 to 2266585
Data columns (total 6 columns):
 #   Column            Non-Null Count    Dtype
---  ------            --------------    -----
 0   userID            2266586 non-null  int64
 1   assessmentItemID  2266586 non-null  object
 2   testId            2266586 non-null  object
 3   answerCode        2266586 non-null  int64
 4   Timestamp         2266586 non-null  object
 5   KnowledgeTag      2266586 non-null  int64
dtypes: int64(3), object(3)
```

- userID (int) : 사용자의 고유번호. 7,442명의 고유 사용자가 존재.
- assessmentItemID (str) : 문항의 고유번호. 9,454개의 고유 문항이 존재.
- testID (str) : 시험지의 고유번호. 1,537개의 고유한 시험지가 존재.
- answerCode (int) : 문항을 맞췄는지 여부. 0은 틀린 것, 1은 맞춘 것.
- Timestamp (date) : 문항을 풀기 시작한 시점.
- KnowledgeTag (int) : 문항 당 하나씩 배정되는 지식 태그. 912개의 고유 태그가 존재.

## 2. 프로젝트 팀 구성 및 역할

최종 프로젝트를 시작하기 전 마지막 대회인 만큼 그동안 배웠던 이론과 방법론들을 스페셜 미션과 강의들을 통해 정리하고 대회를 통해 실험하였습니다. 또한 Github의 경우 이전까지는 폴더별로 나누어 작업 후 notion을 통한 공유 방식으로 협업했지만, 이번 대회에서는 보다 현업에서의 개발과정과 유사하게 유기적으로 협업하기 위해 Issue 및 Pull Request를 적극 활용하는 git-flow 전략을 도입하여 branch에 기반한 방법으로 진행하였습니다.

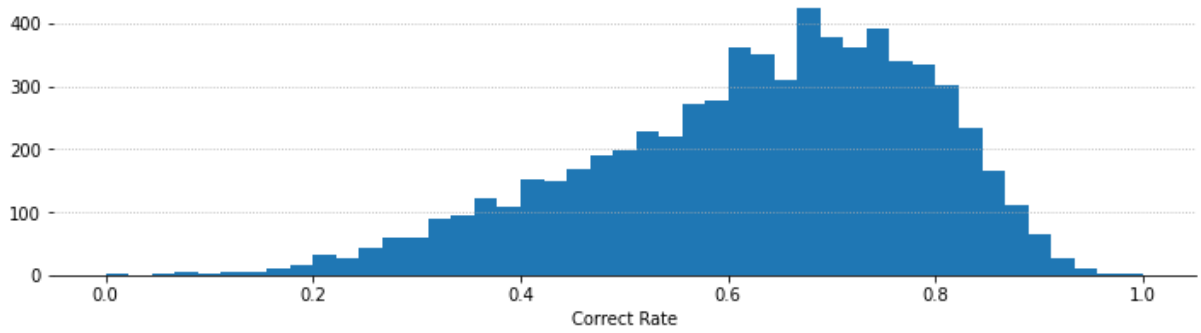
## 3. 프로젝트 수행 절차 및 방법

### 3.1 EDA 및 데이터 전처리

**학생별 정답률 분포**

| | |
|---|---|
| count | 6698.000000 |
| mean | 0.628912 |
| std | 0.159637 |
| min | 0.000000 |
| 25% | 0.527000 |
| 50% | 0.652000 |
| 75% | 0.751000 |
| max | 1.000000 |

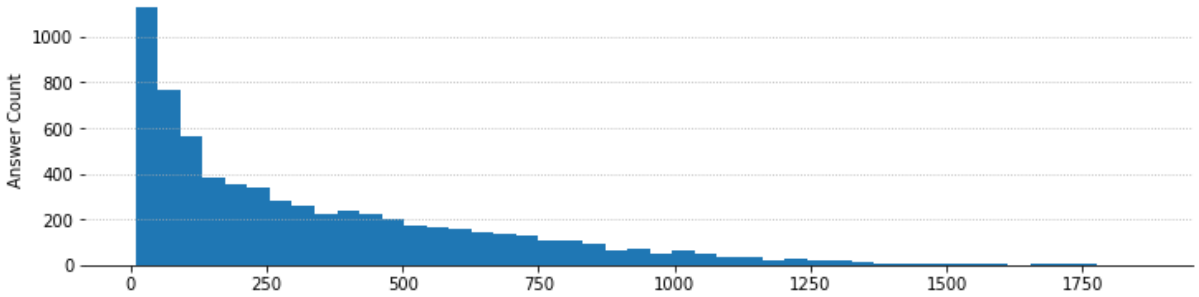
평균 정답률이 약 0.63이고 정규분포와 유사한 형태를 나타내고 있습니다. 특이점으로는 문제를 다 틀린 학생과 다 맞춘 학생이 존재한다는 점 입니다.



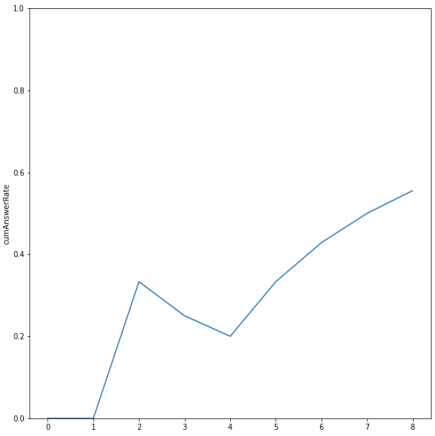

**학생별 푼 문제 수 분포**

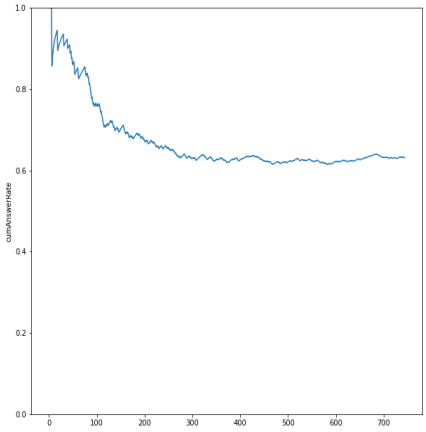
| | |
|---|---|
| count | 6698.000000 |
| mean | 338.397432 |
| std | 321.331429 |
| min | 9.000000 |
| 25% | 78.000000 |
| 50% | 231.500000 |
| 75% | 513.000000 |
| max | 1860.000000 |

6698명의 학생 중에서 문제를 가장 많이 푼 학생은 1860 문제를 풀었고, 가장 적게 푼 학생은 9문제로 차이가 많이 나는 것을 알 수 있습니다.



## 정답률과 난이도에 대한 인사이트




랜덤하게 샘플링 한 두 유저의 누적 정답률입니다. 처음에 1 혹은 0에서 시작하며 문제를 푼 데이터가 많이 쌓일수록 대체로 특정 값에 수렴하는 경향을 보입니다.

| difficulty | answer_rate | user_count |
|---|---|---|
| 010 | 0.800876 | 272082 |
| 020 | 0.737593 | 268327 |
| 030 | 0.702238 | 273762 |
| 040 | 0.684056 | 267323 |
| 050 | 0.658208 | 275773 |
| 060 | 0.709232 | 264434 |
| 070 | 0.521876 | 279164 |
| 080 | 0.502598 | 246336 |
| 090 | 0.449948 | 119385 |

시험지 ID에서 앞 세 개의 숫자가 난이도와 상당부분 연관이 있는 것으로 보입니다.

- difficulty 숫자가 높아질수록 answer_rate가 낮아지는 경향
- 단, (050 -> 060)에는 예외로 answer_rate가 높아지는 현상 확인

| item_num | answer_rate |
|---|---|
| 001 | 0.749916 |
| 002 | 0.720062 |
| 003 | 0.687773 |
| 004 | 0.663364 |
| 005 | 0.599134 |
| 006 | 0.555685 |
| 007 | 0.515399 |
| 008 | 0.457156 |
| 009 | 0.481729 |
| 010 | 0.527892 |
| 011 | 0.480609 |
| 012 | 0.370370 |
| 013 | 0.200743 |

한 시험지 내에서는 뒤에 있는 문제일수록 대체로 난이도가 상승하는 것을 확인 가능

### 3.2 Feature Engineering

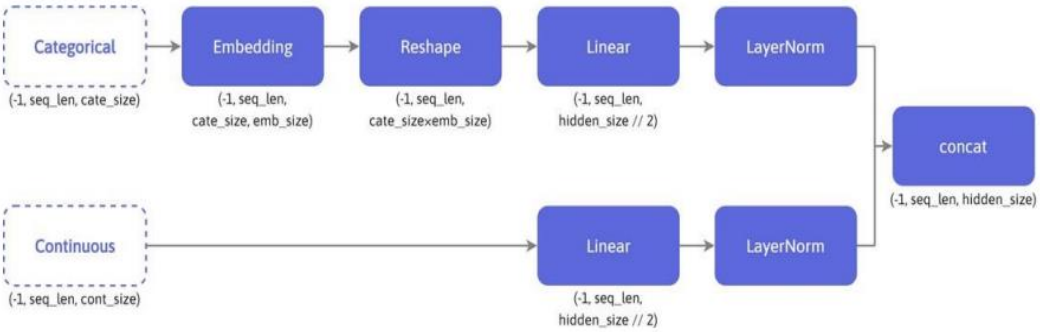
EDA를 통해 확인한 바와 같이, 정답률과 같은 정보로 문제에 대한 난이도에 추정해 볼 수 있는 등 문제 풀이 기록이기에 이와 관련된 추가적인 정보를 획득할 여지가 다수 존재합니다. 이러한 정보들을 활용하기 위해 feature engineering이 유효할 것이라 판단하게 되어 진행하였습니다.

- clusterHour : 집중력이 좋은 시간대가 있을 것으로 가정하고 법정 시간대를 기준으로 아침(5~9), 낮(9~17), 저녁(17~21), 밤(21~24), 새벽(0~5)으로 구분하였습니다,
- tagRate : 태그별 정답률입니다.
- answerRate : 문제별 정답률입니다.
- elapsedTime : 현재 문제를 푸는데 걸린 시간입니다. 다음문제를 풀기 시작한 시간에서 현재 문제를 풀기 시작한 시간을 뺀 값입니다. 단, 문제 풀이에 시간이 일정 기준 이상(1000초) 소요된 경우는 예외로 판단하고 처리합니다.
- cumAnswerRate : 유저별 누적정답률입니다. 유저의 문제 풀이 기록에 기반하여 생성합니다.
- pretrained_item : RNN 및 self-attention based model과 같은 sequential 모델들은 아이템 간 관계를 무시하는 경향이 있습니다. 이를 해결하기 위해 그래프 기반 딥러닝 모델인 LightGCN에서 학습한 아이템 임베딩을 활용합니다.

### 3.3 Modeling

a. augmentation
    i. 데이터가 문제 풀이 기록으로 된 시퀀스 데이터이기 때문에, 사용자의 풀이 기록에 대해 stride를 적용하여 sliding window 방식으로 학습용 데이터를 증강시켜 활용합니다.
b. 모델에서 문제의 표현 방법
    i. 문제 풀이 기록은 푼 문제들에 대한 시퀀스로 이루어져 있습니다. 하지만 문제라는 객체를 이루고 있는 특징들로는 문제의 고유번호와 시험지 정보 등과 같은 범주형 정보, 그리고 문제의 풀이 시간 정보처럼 수치형 정보들이 혼재되어 있습니다.
    ii. 범주형 데이터의 경우 Embedding 통해 데이터로 표현 및 LSTM입력을 위해 (seq_len, feature_n)으로 차원을 맞출 필요가 있습니다.

iii. 범주형 데이터 임베딩과 연속형 데이터를 각각 Linear, LayerNorm 레이어를 통과시킨 후 이어 붙여 LSTM과 같은 sequential 모델에 투영시켜 학습할 수 있게 됩니다.




c. k-fold cross validation

i. 데이터를 5개블록으로 분할한 후, 1개의 검증데이터 블록과 4개의 학습데이터 블록으로 구분하여 Out-Of-Fold로 모델을 평가합니다.

ii. inference 과정에서는 학습된 5개의 모델을 종합할 방법이 필요하다. CV세트 기반의 스태킹을 적용하기 위해 각기 모델의 AUC-ROC 점수를 통한 weighted ensemble을 적용하였으며, 이 경우 최종 inference 결과가 모든 train set의 데이터를 활용한 결과가 됩니다.

## 4. 프로젝트 수행 결과

프로젝트 시작 이후 제출 별 점수 변화를 나타낸 그래프입니다. 리더보드 최종 점수는 AUC-ROC = 0.8013, Accuracy = 0.7419입니다.

최종 결과물은 clusterHour, tagRate, cumAnswerRate 등 **3.2 Feature Engineering**에서 언급한 모든 feature들을 추가하고, CV세트 기반 스태킹 앙상블을 사용한 모델입니다.

## 5. 자체 평가 의견

이전 대회에서의 피드백으로 보다 유기적인 협업을 위해 commit convention 등 여러 규칙을 정하고, Issue와 Pull Request를 적극 활용하였습니다. 또한 git-flow 전략에 따라 코드 리뷰를 하며 진행한 것 등 모두 코드에 대해 이해하며 진행할 수 있어 효과적으로 다가왔습니다.

현 대회 모델링과 더불어 어떠한 모델링에도 활용할 수 있는 code-base를 제작하는 목표는 성공적이었으며, 이를 통해 체계적인 협업을 체험할 수 있었던 대회였습니다. 하지만 짧은 기간의 대회에서 다양한 목표를 설정하고, 동시에 최종프로젝트도 준비하다 보니 우선순위를 낮게 설정하였던 feature engineering에 투자한 시간이 상위권 팀에 비해 부족하였단 점은 부정하기 힘듭니다. 이 부분에 충분한 고민과 실험을 진행하지 못해 마지막 대회형 프로젝트에서 상위권에 이름을 남기지 못한 점은 아쉬움으로 남습니다.

앞으로의 일정은 최종 프로젝트만이 남아 있으며, 지금까지 공부한 지식들을 모두 녹여낸 결과물을 완성하기 위해 달릴 예정입니다.

## 6. 팀원 개인 회고

### 김원섭(T_3044)

**목표**

- (팀) 시퀀스 task에 대한 팀 전체의 기량을 향상시키기 위해, 다양한 심화 이론들에 대한 모델링을 각자 진행하고, 시도한 방법과 결과를 노션과 깃허브에 공유하는, 대회와 스터디의 장점을 융합한 형태의 협업을 진행한다.
- (개인) 시퀀스 모델 및 컴피티션 관련 심화 논문 등을 읽고 이해하며, 이를 구현할 수 있는 능력을 함양하기 위해 나만의 모델링 코드를 구축하고, 다양한 실험 사이클(가설-검증)를 구성한다.

**나는 내 목표를 달성하기 위해 무엇을 어떻게 했는가?**

- **EDA 및 데이터 분석 적용**
  - 시험지 번호 및 문제 번호와 난이도(정답률) 사이의 상관관계를 발견 및 모델 feature로 적용
  - 시험지 번호 앞 3자리는 난이도와 직접적인 관계가 있음
  - 문제 번호가 커질수록 난이도는 어려워짐
- **모델링**
  - LGBM: 다양한 피처를 고려할 수 있는 LGBM 모델을 구현 및 실험. 유저별 시퀀스를 고려하기 위해 timestamp 특성을 피처로 사용.
  - LightGCN: 문제(아이템)와 유저의 관계를 잘 포착하기 위한 graph모델 구현 및 실험.
  - LSTM with Pretrained embedding: 기존 Sequence 모델들은 아이템 간 관계를 무시하는 경향이 있음. 이를 해결하기 위해 LightGCN 모델에서 학습된 임베딩을 LSTM모델의 feature로 추가하여 학습을 진행.

**스스로 칭찬할 점은 무엇인가?**

- 필요한 태스크에 대해 정확한 문제 정의를 진행하기 위해 노력하였다.
- pretrained embedding을 하기 위해 LSTM과 LightGCN 모델들을 뜯어보고, 적절한 실험 설계를 하기 위해 노력하였다.

**마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

- 대회 기간에 해야 할 것들이 많이 겹쳐서 스스로 시간 등의 자원을 효율적으로 분배하지 못한 것 같다 아쉽다.
- 체력 등의 이유로 더 많은 실험 설계를 하지 못해 아쉬웠다.

- 여러 competition이 지속되다보니 목적을 명확히 하지 않고, 기계적으로 참여했던 것 같다.

**한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇인가?**

- 현재 여러가지 일들을 하는데 내 자원(시간, 노력 등)을 효율적이게 사용하지 못하고 있는 것 같다. 해야 할 것들을 하기에 앞서, 하고 있는 모든 것들에 대한 재점검과 자원의 재분배가 필요할 것으로 보인다.
- 어떠한 task가 주어졌을 때, 정확한 문제 정의를 먼저 하기위해 노력했던 태도를 다시 갖추기 위해 노력해야겠다.

김진수(T_3058)

**목표**

- 전체적인 AI 프로세스를 이해하고 하지못한 실험 시도해보기
- Recbole 라이브러리 활용법 익히기
- 협업툴(github, notion, wandb) 사용 익숙해지기

**목표 달성을 위한 노력**

- **Cross Validation 만들기** : 그동안 CV를 위해 데이터를 일일이 나누는 방식으로 구현했지만, 대회외적으로도 공부를 하여 sklearn을 활용해 간단하게 CV를 구현하고 CV점수와 대회점수가 함께 증가하거나 감소하여 간단히 잘 작동하는 CV를 구현할 줄 알게 되었음.
- **Small Feature Engineering** : 간단한 EDA 및 Feature Engineering으로 크지는 않지만 점수 향상을 이루어 낼 수 있었고, LSTM에 범주형 데이터와 수치형 데이터를 임베딩 하는 방식을 구현 및 활용하였음.
- **협업툴 사용** : 이고잉님의 강의에서 배웠던 부분과 멘토님이 알려주신 현업에서의 Git 활용 방식을 대회를 통해 실습해볼 수 있었음. Issue, PR, Branch, Commit 규칙 등 여러가지 Git 동작을 충분히 연습하고 익힐 수 있었음.

**아쉬운 점**

- **Deep Feature Engineering** : Special Mission에 매우 다양한 Feature 들이 예제로 나와있었으나, 충분한 실험을 진행해 보지 못했던 점이 아쉬움.

- **Recbole 라이브러리 활용 :** 베이스라인코드 중 Recbole을 활용한 LightGCN 모델이 있었으나 충분히 분석을 해보지 못해 아쉬움. 부스트캠프가 끝나면 Recbole을 다루는 방법을 다시한번 분석해볼 계획임.

### 민태원(T_3080)

#### 목표

- 데이터를 중심으로 인사이트 발견하기
- 모델에 대한 이해 및 기능 구현 역량 키우기
- 협업툴(github, notion, wandb) 사용 익숙해지기

#### 목표 달성을 위한 노력

- **EDA 를 통한 가설 세우기 :** 범주형 feature 들을 다양하게 조합해서 feature를 생성. 주어진 시간 데이터에서 유의미한 정보를 얻기 위해 시(hour)를 분류한 범주형 feature 생성, 초(second)를 이용해 문제 푸는 시간 을 나타내는 feature 생성. 이러한 feature들과 정답률을 조합한 변수들을 이용해 다양한 실험을 하며 유의미한 feature를 발견하고자 함.

  (결과) 유의미한 feature 를 발견하기 위해 다양한 실험을 했지만, 각각의 feature에서는 좋은 성능을 얻지 못했음. 하지만 여러 feature를 한번에 사용하면 유의미한 상승 효과를 가져옴.

- **모델 구조에 대한 분석과 추가 기능 구현해보기 :** Sequential model 의 구조에 대해서 생각해보았고 이러한 특성을 반영할 수 있는 feature 들을 추가하면서 embedding에 대한 생각 및 방법을 팀원들과 공유하면서 전반적인 딥러닝에 대한 이해력을 키움.
- **협업툴의 적극적인 사용 :** 특히 github의 강의와 더불어 팀원들과 규칙을 가지고 github를 사용한 결과 상당히 익숙해진 것을 느낌. wandb의 경우는 팀적으로 활용을 하지 않았지만, 개인적으로 사용해본 결과 간단하고 편리해서 만족스러웠음. 다른 팀에서 많이 사용하는 Jira도 사용해볼 예정.

#### 아쉬운 점

- **여전히 구현이 어려움 :** 기능을 구현하는데 있어서 어려움이 많았음. 특히, 차원에 대한 생각이 아직 정리가 안됨. 수치형 feature를 embedding 하는데에 여러가지 방법을 사용해 봤지만 차원 에러를 잡지 못해 구현하는 것은 결국에는 성공하지 못함.

- **Graph 기반 모델을 사용해 보지 못함 :** 추천시스템에서 활발하게 연구되고 있는 graph 이론을 이용한 모델을 구체적으로 분석하지 못함. 이론적으로는 이해한것 같으나 실제 코드로 봤을때, 이해가 가지 않는 부분이 많았음.

**앞으로 방향성**

- 기본이 중요하다는 것이 와닿고 있음. 딥러닝, 추천시스템 이론은 한번 정리를 할 계획. 간단한 베이스라인 코드를 수월하게 만들 수 있을 정도의 역량이 필요한 것 같음.
- github의 경우에는 이전에 비해서는 상당히 사용하는데 어려움이 없어졌지만, branch를 자유자재로 다루는 데에는 아직 어려움이 있음. 사용하지 않다보면 잊어버릴 수 있으므로 주기적으로 연습 필요.
- kaggle 과 데이콘과 같은 플랫폼을 이용해 다양한 데이터를 사용해보면서 data handling 역량을 더 키우겠다고 다짐.

이상목(T_3146)

**목표**

- DKT task 및 이에 어울리는 데이터와 모델에 대한 이해
- 어떠한 모델링에도 활용할 수 있는 형태의 base code 완성

**성장한 점**

- **협업 툴의 적극적인 활용 :** Github의 Issue 및 PR기능을 직접 다뤄보면서 쓰면 좋지만 어떤 때 써야 하는지, 어떤 이득이 있는지를 체감하지 못하고 있었는데 이를 확실히 체감할 수 있었다. 특히 팀원들과 진행 상황을 공유하면서 동시에 문서화도 가능하다는 점 등이 매우 매력적인 포인트라고 생각한다.
- **Sequence 모델에 대한 더 깊은 이해 :** DKT는 사실 추천이라기 보다는 시퀀스 모델링이라 보아도 무방하지 않았나 생각한다. 과 양방향 LSTM, self-Attention, transformer 등 다양한 Sequential 모델을 만져 보았는데, self-Attention기반 알고리즘들이 강력하지만 데이터가 부족한 상황에서는 왜 성능이 더 낮을 수 있는지 등에 대해 고민해 볼 수 있었다.

**성장할 점**

- **GCN에 대한 더 깊은 이해:** GCN 부분은 내가 맡은 파트가 아니다 보니, 개념은 이해하였지만 코드로는 아직 와닿지가 않아서 추가적인 공부가 필요하다.
- **서빙 인프라에 대해 더 공부하기 :** 프로젝트를 준비하면서, MLOps와 서빙과 같은 인프라에 더더욱 관심이 간다고 느껴져서 해당 부분을 더 공부해보려 한다.

조민재(T_3204)

**목표**

DKT Task를 이해하고 데이터로부터 다양한 Feature를 탐색하기

**실행 내용**

- mission에 제공된 것 외의 유의미한 feature 추가 탐색

- ground rule 설정과 다양한 도구를 활용한 유기적인 협업 과정 설계

- knowledge tag 별 누적 풀이에 따른 경험치 적용 시도

**회고**

대회 시작시점부터 최종 프로젝트를 위한 아이디어를 탐색하고 설계를 구체화 하는 것에 관심이 쏠려, 이번 DKT competition에는 지나치게 소홀하였다. 대회 막바지에 늦게나마 공부하면서 흥미가 생겨 무척 아쉬움이 남는다. 부스트 캠프 종료 이후에 남겨진 자료들로 다시 한번 공부하고자 한다.