

DKT Project

Recsys15_RecOne

1. 프로젝트 개요

1-1. 프로젝트 주제

초등학교, 중학교, 고등학교, 대학교와 같은 교육기관에서 우리는 시험을 놀 봐왔다. 시험 성적이 높은 과목은 우리가 잘 아는 것을 나타내고 시험 성적이 낮은 과목은 반대로 공부가 더욱 필요함을 나타낸다. 하지만, 시험에는 한계가 있다. 우리가 돈을 들여 과외를 받지 않는 이상 우리는 우리 개개인에 맞춤 피드백을 받기가 어렵고 따라서 무엇을 해야 성적을 올릴 수 있을지 판단하기 어렵다. 이럴 때를 위하여 DKT(Deep Knowledge Tracing)를 구축하여 본다.

1-2. 프로젝트 구조

- **입력:** 약 7000명의 사용자들의 문제 풀이 내역
- **출력:** test_data 사용자들의 마지막 문제의 정답여부
- **구조**
 - 모델 LightGBM, XGBoost에 feature engineering한 feature를 적용하여 시행
 - 이후 LightGBM, XGBoost 각각의 결과를 합친 Ensemble 구조

2. 프로젝트 팀 구성 및 역할

- 소경학 : 데이터 EDA
- 노건웅 : Data EDA, LSTM·Transformer·머신러닝 모델 성능 실험
- 허민영 : Data EDA, feature 생성, LightGBM XGBoost CatBoost 모델 실험, 하이퍼 파라미터 최적화 실험
- 하동준 : LGBM feature engineering, 앙상블 및 overfitting 방지
- 임경연 : Data EDA, LSTM 모델 실험

3. 프로젝트 수행 절차 및 방법

3-1. 프로젝트 사전 기획 및 문제 정의

DKT는 지식 상태를 딥러닝을 통해 추정하는 방법론으로 각 지식에 대한 학생의 이해도를 딥러닝 모델을 통해 파악하고 취약한 부분을 보완할 수 있도록 도와주는 교육 AI이다. 본 대회는 학생의 과거 공부 이력을 바탕으로 마지막 문제를 맞췄을 지, 틀렸을 지 예측하는 Binary Classification 문제였기 때문에 Sequential Modeling에 적합한 Transformer(Encoder)·RNN(LSTM) 같은 딥러닝 모델과 부스팅 계열 모델·전통적인 머신러닝 모델을 통해 문제를 해결하고자 하였다.

3-2. 프로젝트 수행 절차 및 과정

기간	활동 내용
2022.04.18 ~ 2022.04.22	강의 수강, Data EDA, Baseline 코드 분석, Feature Engineering
2022.04.25 ~ 2022.04.29	Baseline 코드를 이용한 ML Pipeline 구축, Sequential 모델 · GBM 모델 실험
2022.05.02 ~ 2022.05.06	외부 라이브러리를 이용해 딥러닝 모델·GNN 모델·전통적인 머신러닝 모델 추가 실험
2022.05.09 ~ 2022.05.13	앙상블을 이용한 성능 향상

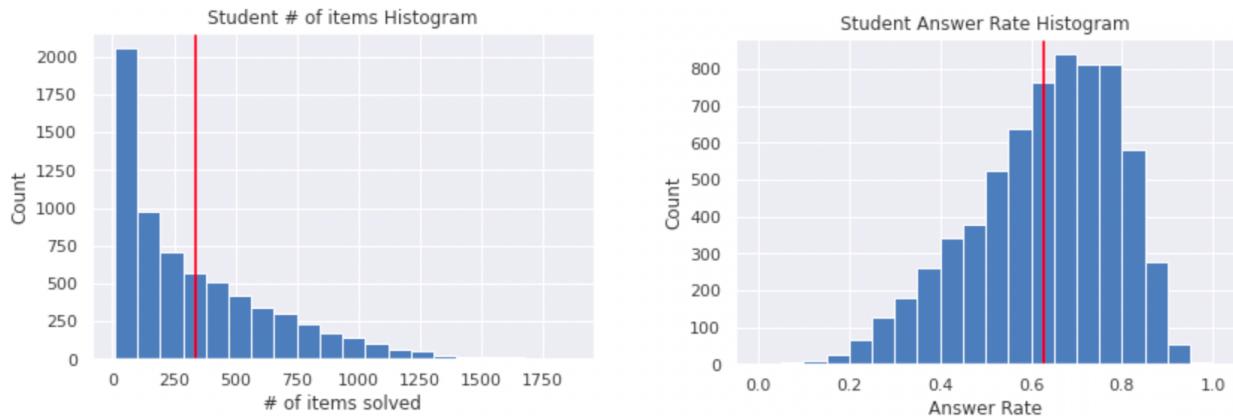
3-3. 탐색적 자료 분석 (EDA) 및 전처리

데이터는 UserID (유저 ID), AssessmentItemID (문제 ID), TestId (시험 ID), answerCode(정답여부), Timestamp, KnowledgeTag(문제 분류)로 구성되어 있다. 여기서 Target은 answerCode로, answerCode는 0 (오답), 1 (정답), -1 (예측)로 되어있다. 여기서 기초 정보를 살펴보자면 아래 표와 같다.

userID (명)	assessmentItemID (개)	testID (개)	평균 정답률 (%)	knowledgeTag (개)
6698	9454	1537	65.44	912

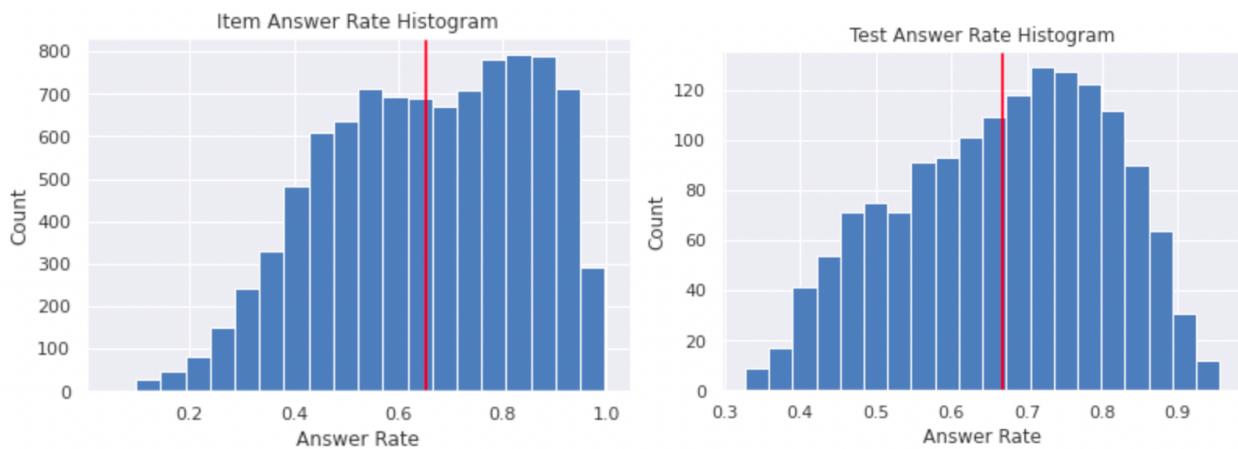
[사용자 분석]

UserID 를 기준으로 각 데이터 분포를 확인해보자면, 아래와 같다. 학생들은 평균적으로 338 개의 문제를 풀었고 대다수의 학생은 250 개 이하의 문제를 풀었다. 또한 학생별 평균 정답률은 0.62이며 대다수의 학생들은 문 문제의 절반 이상 문제를 맞췄다.



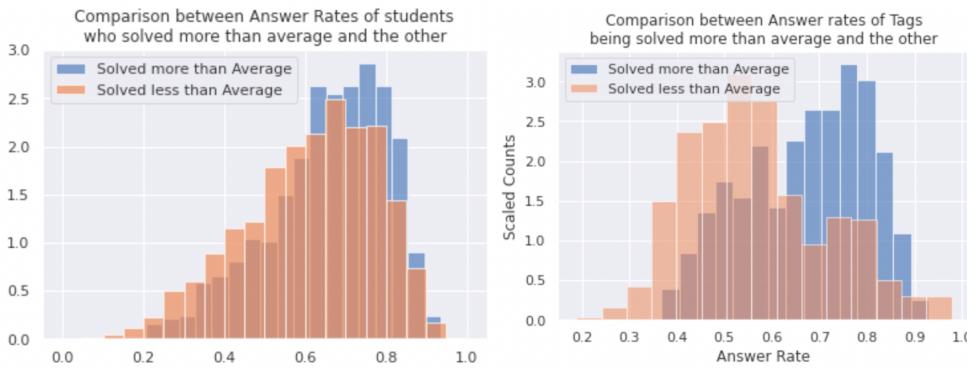
[문항/시험지 분석]

문항별 시험지별 히스토그램을 살펴보면 아래와 같다. 평균 정답률은 각각 0.65, 0.67로 큰 차이가 없으며, 아래 분포도 큰 차이를 보이지 않는다.



[기타 EDA]

일반적으로 생각해보자면, 학습량이 많아 많은 문제를 푼 학생의 경우, 정답률이 높을 것으로 생각한다. 따라서 상관관계가 있는지 확인해보자면 아래와 같다 (좌측). 평균보다 많이 푼 학생, 적게 푼 학생으로 나눠 비교해보면 큰 차이가 없다는 것을 알 수 있다. 실제 correlation 은 0.168로 아주 약한 상관관계를 보인다.



문제별 태그 정보를 활용해 동일한 태그 문제를 풀어본 학생이 비슷한 태그 문제를 잘 맞출 수 있는지 살펴보자면, 위 그림과 같다 (우측). 실제로 비슷한 태그 문제를 많이 풀어본 경우, 더 높은 정답률을 보인다. 실제 correlation 은 0.376 으로 약한 상관관계를 보인다.

3-4. 모델 평가 및 개선 과정

부스팅 계열 모델 중, LightGBM, XGBoost, CatBoost 세 가지 모델로 실험을 진행하였다. CatBoost 모델은 나머지 두 모델에 비해 상대적으로 성능이 좋지 않았으므로 제출에 포함하지 않았다.

[feature Engineering]

- feature Engineering 을 통해 총 90 가지 정도의 feature 를 생성했고, 그 중 정답 여부 예측에 도움이 되었던 ['유저 이전 문제별 오답률', '문제별 오답률', '유저 이전 문제별 상대적 오답률', '시차'] 들을 사용해 성능을 향상 시켰다.

[LightGBM]

- LightGBM 은 정답률 feature 에서 높은 성능을 보여주었고 위의 4 개의 feature 에 대해 다양한 조합으로 실험하여 ['유저 이전 문제별 정답률', '문제별 정답률', '시차'] 를 사용했을 때 Auc 0.8022, Acc 0.7500 로 가장 높았다.
- verbose_eval = 100, num_boost_round = 500, early_stopping_rounds = 100

[XGBoost]

- XGBoost 은 오답률 feature 에서 높은 성능을 보여주었고 위의 4 개의 feature 에 대해 다양한 조합으로 실험하여 ['유저 이전 문제별 오답률', '문제별 오답률', '문제별 정답률', '시차'] 를 사용했을 때 Auc 0.8032, Acc 0.7366 로 가장 높았다.
- verbose_eval = 100, num_boost_round = 500, early_stopping_rounds = 100

[양상블]

- 최종적으로 LightGBM 모델(Auc 0.8022, Acc 0.7500)과 XGBoost 모델(Auc 0.8032, Acc 0.7366)의 양상블을 진행하였고, 이를 통해 최종 리더보드 점수가 (Auc 0.8086, Acc 0.7366) 로 향상되었다

4. 프로젝트 수행 결과

	model	Verbose_eval	num_boost / early stopping	Features	LB 점수 (AUC ACC)
1	LightGBM	100	500, 100	유저 이전 문제별 정답률 문제별 정답률 시차	0.8022 0.7500
2	XGBoost	100	500, 100	유저 이전 문제별 오답률 문제별 오답률 유저 이전 문제별 상대적 오답률 시차	0.8032 0.7366
3	Ensemble			LightGBM_output['prediction'] * 0.5 + XGBoost_output['prediction'] * 0.5	0.8086 0.7366

4	3번 결과의 prediction 값이 0.45미만인 문제라면 0.08만큼 더해주어서 정답으로 예측할 확률 높임	0.8093 0.7446
---	---	-----------------

Feature Engineering을 통해 생성한 feature는 총 90개이며, 크게 ‘정답률’, ‘오답률’, ‘상대적 정답률’, ‘상대적 오답률’, ‘정답 유저 수’, ‘오답 유저 수’, ‘유닉스 시간’, ‘시차’로 나눌 수 있다.

우선 feature들 각각에 대해 실험해보았을 때, ‘유저 수’에 관련된 feature는 valid Auc, valid Acc 값이 좋지 못한 경향을 보였기 때문에 사용할 feature에서 제외시켰다.

유의미한 결과를 가져온 feature들은 다음과 같으며, 꽤 안의 값들은 해당 feature만 모델링 해보았을 때 valid Auc, valid Acc 값이다.

유저 이전 문제별 정답률 (0.63 | 0.55), 문제별 정답률 (0.71 | 0.65), 시차 (0.66 | 0.63),

유저 이전 문제별 오답률 (0.63 | 0.55), 문제별 오답률 (0.71 | 0.65), 유저 이전 문제별 상대적 오답률 (0.64 | 0.53)

위의 feature들을 토대로 다양한 실험을 진행해보았을 때 LightGBM 모델에서는 (‘유저 이전 문제별 정답률’, ‘문제별 정답률’, ‘시차’) 3가지 feature 조합이 높은 성능을 나타냄을 알 수 있었고, XGBoost 모델에서는 (‘유저 이전 문제별 오답률’, ‘문제별 오답률’, ‘유저 이전 문제별 상대적 오답률’, ‘시차’) 4가지 feature 조합이 높은 성능을 나타냄을 알 수 있었다.

마지막으로 가장 높은 두 가지 모델을 양상을 시켜서 (0.8032 | 0.7366) → (0.8086 | 0.7366)만큼의 성능 향상을 볼 수 있었고, 해당 결과의 prediction 값이 0.45 미만인 문제에 대해서 0.08만큼 더해주는 방식을 통해 정답률을 예측할 확률을 높여 줌으로써 (0.8086 | 0.7366) → (0.8096 | 0.7446)만큼의 성능 향상을 볼 수 있었다.

5. 자체 평가 의견

- 하동준: feature engineering 을 새롭게 접하였고 같은 모델을 사용해서 이를 통해 얼마나 성능 개선이 가능한지, 얼마나 중요한 과정인지 많은 깨달음을 얻게 된 대회였다.
- 노건웅: 데이터 전처리부터 시작해 모델을 이용해 결과까지 구하는 pipeline을 구축하는 방법과 Feature Engineering을 이용해 다양한 형식의 feature들을 사용해보고 딥러닝 모델 · 머신러닝 모델을 이용해 실험을 체계적으로 진행할 수 있어서 좋았다. 실제 대회에 적용할 수 있는 여러가지 전략들을 배울 수 있었던 시간이었다.
- 허민영 : feature engineering을 통해 최대한 다양한 피처들을 생성하고 모델링에 적용시켜보면서 모델별로 적합한 피처 조합이 다르다는 것을 알게 되었다. 더 많은 모델을 사용해보지 못한 것에 아쉬움이 남지만 본 대회를 발판으로 삼아 다양한 대회에 참여해서 실력을 더 키워 나가고 싶다고 느꼈다.
- 임경연 : 이번 대회에선 다음 최종 프로젝트 준비를 위해서 많이 참가하지 못해서 아쉬움이 있다. 하지만 머신러닝이 아닌 AI에서도 feature engineering이 중요하다는 점과 머신러닝 모델이 AI모델보다 더 좋은 성능을 보일 수 있다는 점이 새로웠다.
- 소경학 : 프로젝트에 제대로 집중하지 못한 점이 아쉬웠고, 평소 생각만해오던 프로젝트를 진행할 수 있어서 다양하게 도움이 된 것 같다.

Wrap Up Report

소경학_T3110

1. 이번 프로젝트에서 목표는 무엇이었는가?

이번 프로젝트의 목표는 모델의 성능을 높이기 위한 방법들을 연구함에 있었다. 데이터 전처리부터 하이퍼 파라미터까지 EDA를 통해 얻은 정보들을 토대로 다양한 방법들을 시도해보고 설계해보는 것이 주된 목표였다. 학습한 내용을 토대로 모델을 A부터 Z까지 설계하는 과정을 이해하는 것에도 목적이 있었다.

2. 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

- 하이퍼 파라미터를 다양하게 바꿔보며 최적의 성능을 내는 수치를 찾아봄
- 여러 모델을 데이터셋에 걸맞게 수정하고 실험을 진행함
- 다양한 모델을 앙상블을 해보고 결과를 기록하고 분석함
- EDA를 통해 문제 정의에 초점을 두고 목적에 맞게 모델을 설계함

3. 나는 어떤 방식으로 모델을 개선했는가?

- 다양한 피처를 만들어보고 그것들을 이용해 모델을 훈련시켜 성능을 개선함
- 높은 성능을 기록한 모델을 앙상블을 통해 성능을 개선함
- 여러 모델의 결과를 토대로 성능을 분석해보고 라벨에 따른 앙상블을 시도함

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

· 기존에 있던 피처를 통해 새로운 피처를 만들어보고 이를 통해 다양한 실험을 해봄. 모델을 설계하기전에 데이터를 분석하는 과정이 무엇보다 중요한 것을 깨달았다. 데이터 전처리를 할 때에도 우리가 목표로 하는 것이 무엇인지 확실히 인지하고 했을 때 더욱 효과적인 방법으로 성능을 개선할 수 있는 것을 알게 되었고, 시간 또한 절약할 수 있다는 것을 알게 되었다.

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

· 문제 정의에 좀 더 시간을 투자했다. 문제 정의를 하지 않고 접근하였을 때 보다 실험을 하는데에 있어서 목적이 뚜렷하게 생겼고 실험을 통한 결과를 분석할 때도 좀 더 효과적으로 할 수 있게 되었다.

6. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 모델간의 차이를 뚜렷하게 밝혀내지 못한 것이 아쉬웠다. 제출횟수에 한계가 있어서 다양한 모델을 여러 경우의 수로 나누어 실험하지 못한 부분이 아쉬웠다.
- 실험결과에 대한 분석을 제대로 하지 못한 것이 아쉬웠다. 성능이 좋게 나온 이유와 안좋게 나온 이유를 뚜렷하게 밝혀내지 못하였다.
- 찾을 수 있는 모델들을 데이터 셋에 걸맞게 변환시키지 못하였다.

7. 한계/교훈을 바탕으로 다음 P-Stage에서 스스로 새롭게 시도해볼 것은 무엇일까?

- 문제를 올바르게 정의하고 진행할 방향을 정하는데 시간을 더욱 투자해보고 싶다.
- 마구잡이인 피처를 이용하지 않고 문제에 걸맞는 방향으로 모델을 직접 설계해보고 싶다.
- 주제에 대한 파악을 심도 있게 하고, 올바른 방향을 정하고 설계해보고 싶다.
- 베이스라인 코드를 명확하게 분석해보고 코드를 다양하게 바꿔가며 내 것으로 만들어 보고 싶다.

Wrap Up Report

하동준_T3229

1. 이번 프로젝트에서 목표는 무엇이었는가?

이번 프로젝트의 목표는 다양한 종류의 모델을 시도하고 최적의 성능을 내는 feature engineering 을 진행하는것과 categorical, continuous 데이터를 특정 모델에 맞게 임베딩하는 것에 있었다. 또한 최종 submission 파일을 성능 향상시키는 앙상블 기법 또한 주된 목표 였다.

2. 나는 내 학습목표를 달성하기 위해 어떤 방식으로 모델을 개선했는가?

- baseline 으로 주어진 LGBM, LSTM, BERT(양방향), GNN 등의 모델을 사용하여 최적의 성능을 찾아봄
- riiid competition BERT 코드를 대회 데이터에 맞게 수정
- transformer 를 사용하여 성능을 올리는데 데이터 크기에 한계가 있었고, LSTM 의 feature engineering 과정을 통해 점차적으로 성능을 개선시킴.
- RFE/RFECV/feature importance 를 이용하여 feature 선별
- 맞았다고 예측한 데이터들이 틀렸다고 예측한 데이터보다 두배 가량 많았기 때문에 0.45 ~ 0.5 로 예측한 경우 추가적으로 0.05 ~ 0.08 정도 값을 더해줌

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

- LGBM baseline 코드를 활용하여 feature 를 하나씩 제거하며 최적의 feature 를 선정하였고 auroc 0.7466 을 기록하였다. 최적의 feature 를 이용해 다시 5-fold cross validation 을 하였고 5 개의 submission.csv 를 voting 하고 0.45 ~ 0.49 로 예측한 값들에 한정해 0.051 을 더해주었고 auroc 0.7479 의 유의미한 성능 개선을 얻어낼 수 있었다. 이를 통해 앙상블의 중요성과 함께 overfitting 을 방지하는 것의 중요함을 깨달을 수 있었다.

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

- 주어진 데이터의 feature 만 활용했던 이전과 달리 과정을 통한 최적의 feature engineering 과 새로운 feature 를 만들었다. 이를 통해 데이터를 주어진대로만 활용하지 않고 모델에 따라 새로운 feature 를 추가하는 과정이 중요함을 깨달았다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 우리 팀에서는 높은 성능을 내지 못한 transformer 계열의 모델을 상위 기록 팀들은 활용 했다는 것에 놀라움과 아쉬움을 느꼈다. feature engineering 의 중요성을 알게 된 대회이지만 주어진 baseline 코드 만을 이용해 단순 추가, 삭제 를 반복한 것 같아 아쉬웠고 대회 종료 이후에도 추가적인 실험을 해보고 싶다고 생각을 했다.

6. 한계/교훈을 바탕으로 다음 P-Stage에서 스스로 새롭게 시도해볼 것은 무엇일까?

- 주어진 task 와 데이터를 이용해 단순 성능 개선을 위해 4주를 보낸 이전의 대회들과 달리 최종 프로젝트에는 내가 직접 선정한 주제를 한정된 시간 내에 다양한 툴과 사용자가 실제 사용할 때 부족한 점이 없도록 디버깅 및 완성도에 집중해볼 생각이다. 또한 만들어낸 결과물이 해결할 문제점과 차별성에도 집중할 것이다.

Wrap Up Report

허민영_T3235

프로젝트 목표

- 주어지는 Baseline Code를 습득하여 내 것으로 만들고 최대한 많은 것을 실험해보기
- 주어진 데이터에 대해 최대한 꼼꼼하게 EDA를 진행하기
- 피처 엔지니어링을 통해 최대한 다양한 피처들을 생성해보고 실험에 적용시키기

나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

개인 학습 측면

- 처음에는 피처 엔지니어링을 어떻게 해야 할지 감이 잡히지 않아서 Iscream 데이터셋과 유사한 Riid 데이터셋으로 진행되었던 Special Mission의 코드를 꼼꼼히 공부하면서 Iscream 데이터셋에 적용시키려 노력하였다.
- 피처 엔지니어링을 통해 이미 많은 피처가 생성되었음에도 불구하고, 모델링에 도움이 될 만한 새로운 피처를 더 생성하기 위해 계속해서 생각했다.
- 내가 생각했을 때는 좋은 피처가 아닐지라도 성능이 좋게 나오면 그 모델에 대해서는 좋은 피처라는 강사님의 말씀을 바탕으로 생성한 모든 피처들을 모델링에 적용시켜서 하나하나 관찰해보는 과정을 진행하였다.
- 부스팅 계열 모델을 최대한 활용해보고자 LightGBM, XGBoost, CatBoost 세 가지 모델을 중점으로 실험을 진행하였다.

공동 학습 측면

- 서로 질문이 있을 시 Slack, Zoom 그리고 카카오톡을 활용하여 언제든 도움을 주고받았다.
- Wandb를 통해 Train 과정을 공유하였다.
- 피처 엔지니어링을 통해 생성해낸 피처들을 공유하였다.
- 좋은 결과물을 종합해 양상별하는 과정을 진행하였다.

나는 어떤 방식으로 모델을 개선했는가?

- 피처 엔지니어링을 통해 약 90개의 피처를 생성하였으며, 생성된 피처들 각각을 모두 실험해보며 그 중 어떤 피처들이 유의미한 결과를 가져올지 판단하여 모델링에 적용시킬 피처를 선택하였다.
- 선택된 6가지 피처들로 다양한 조합을 만들어 모델링에 적용시켰다.
- LightGBM 모델에서는 ('유저 이전 문제별 정답률', '문제별 정답률', '시차') 3가지 피처 조합으로 유의미한 결과를 얻을 수 있었다.
- XGBoost 모델에서는 ('유저 이전 문제별 오답률', '문제별 오답률', '유저 이전 문제별 상대적 오답률', '시차') 4가지 피처 조합으로 유의미한 결과를 얻을 수 있었다.

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

- 처음에는 순전히 내가 생각했을 때 좋은 피처라고 생각되는 피처들의 조합으로 모델링을 진행해보았는데, 기대한 것에 비해 성능이 너무 좋지 못했다.
→ 내가 생각했을 때는 좋은 피처가 아닐지라도 성능이 좋게 나오면 그 모델에 대해서는 좋은 피처라는 강사님의 말씀이 맞았음을 깨닫고 피처들 각각을 모델링에 적용시켜보면서 최종적으로 모델링에 적용시킬 피처를 선별했다.
- 많은 피처들을 모델링에 적용시키면 성능이 무조건적으로 향상될 줄 알았으나, 좋은 피처 조합이 아니라면 피처를 적게 사용하는 것보다 못하다는 것을 깨닫게 되었다..
- LightGBM 모델에서는 정답률을 관련 피처가 좋은 성능을 보였고, XGBoost 모델에서는 오답률을 관련 피처가 좋은 성능을 보이는 것을 직접 확인해봄으로써 모델별로 적합한 피처가 있다는 것을 깨닫게 되었다.

마주한 한계와 아쉬웠던 점은 무엇인가?

- 더 다양한 모델을 사용해보지 못한 것에 아쉬움이 남는다.
- 하이퍼 파라미터 최적화를 통해서 최적의 하이퍼 파라미터 값을 찾으려고 노력했으나, 기존의 하이퍼 파라미터 값에 비해서 더 좋은 결과를 가져온 하이퍼 파라미터 조합을 찾지 못했다.
더 많은 실험을 진행하였다면 찾을 수 있었을 거라고 생각하는데 그러지 못했음에 아쉬움이 남는 것 같다.

한계/교훈을 바탕으로 다음 스테이지에서 새롭게 시도해볼 것은?

- 시계열 데이터를 활용하여 더 다양한 피처를 생성해보지 못한 것에 아쉬움이 남는다.
다른 대회에서 시계열 데이터를 활용할 일이 생긴다면 '연도, 월, 일, 시, 분, 초' 별로 다양한 피처를 생성해보고 싶다.

개인 회고

노건웅_T3073

1. 이번 프로젝트에서 목표는 무엇이었는가?

- 데이터 전처리부터 모델을 이용해 결과를 구하는 것까지 직접 pipeline 구축해보고 실제 대회에서 쓰일 수 있는 다양한 방법과 전략들을 경험해보고 공부하는 것이 목표였다

2. 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

- 데이터 전처리부터 시작해 모델을 이용해 결과를 구하는 것까지 직접 pipeline 구축
- Feature Engineering을 통해 여러가지 Feature들을 추가·제거해보며 다양한 실험 진행
- Huggingface를 이용해 Transformer 계열 모델 실험
- LSTM, RNN 모델 실험
- LightGBM 및 sklearn 라이브러리를 이용한 전통적인 머신러닝 모델 실험
- hyperopt, ray tune을 이용한 하이퍼 파라미터 실험
- 앙상블

3. 나는 어떤 방식으로 모델을 개선했는가?

- 직접적으로 모델을 개선했기보다 Feature Engineering을 이용해 다양한 Feature를 제작·추가·제거하면서 성능을 올리려고 노력했다. 마지막에 hyperopt,, ray를 이용해 하이퍼 파라미터 실험을 하며 성능을 개선하였다.

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

- 다양한 조건들을 설정하고 실험하면서 모델의 성능을 개선하려고 노력했지만 실제적으로 큰 성능 향상폭이 없어 결과적으로 사용하지 못하였다. 대회 마지막 날 다른 조에서 발표한 것을 들어보니 체계적으로 실험 관리를 못한 것과 다양한 모델을 앙상블 실험을 하지 못한 것이 아쉬웠다. 더 다양하게 Feature Engineering을 하지 못한 것도 아쉬웠다.

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

- 데이터 전처리부터 시작해 모델을 이용해 결과를 구하는 것까지 직접 pipeline을 구축해본 것
- hyperopt, ray 같은 라이브러리를 이용하여 하이퍼 파라미터 실험을 진행해본 것
- 딥러닝 모델 · 머신러닝 모델을 이용해 앙상블을 시도해보고 실험해본 것
- 딥러닝·머신러닝 대회에 쓰일 수 있는 다양한 전략들을 배우고 직접 체험해본 것
- 이를 통해 실제적으로 쓰일 수 있는 다양한 방법들을 배워보고 코드를 직접 구현해보면서 코딩에 관한 부분이 조금 늘었던 것 같다. 그리고 외부 라이브러리를 사용하는데 부담감이 없다는 생각이 들었다.

6. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- Weight & bias 같은 실험 관리 도구를 이용해 체계적으로 실험을 정리하지 못했던 것이 어렵다. 무엇보다 성능 향상에 크게 기여하지 못한 부분이 가장 아쉬웠다. 앞으로는 어떤 부분이 부족하고 어떤 부분을 발전시킬 수 있을지 검토해보고 가설을 설정해 실험할 수 있도록 노력해야겠다는 생각이 들었다.

개인 회고

임경연_T3245

1. 이번 프로젝트에서 목표는 무엇이었는가?

Feature Engineering 을 집중적으로 하고자 했다. 우선 대회가 Kaggle 대회와 비슷한 구조로 되어있어, Kaggle 과 같은 경우 Feature engineering 이 중요하기 때문에 해당 feature engineering 을 중점적으로 진행했다.

2. 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

Feature 를 생성하고 변경하면서 모델 학습시켰다. 또한 기존 Baseline 코드인 LSTM 모델을 뜯어보면서 구조 학습을 했다.

3. 나는 어떤 방식으로 모델을 개선했는가?

기존 Baseline 을 토대로 하이퍼 파라미터를 변경하면서, 새로운 feature 를 생성하면서 모델 성능 개선을 했다.

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

Feature engineering 을 시도한 것이 제일 큰 시도였다. 기존에 알고있던 지식은 AI는 스스로 feature engineering 을 하기 때문에 하지 않아도 된다는 생각이 많았지만, 이번에는 feature engineering 이 주요 주제였고 많은 것을 배울 수 있었다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

우선 이번 프로젝트에 완전 몰입해서 참가하기 힘들었다. 교육 기간이 길어지면서 나타해진 면도 있었고, 최종 프로젝트가 바로 앞에 있어, 주제 선정과 같은 준비단계가 있어 많이 참가하지 못해서 아쉬웠다. 또한 항상 느끼는 것은 약간 가설 -> 검증이 아닌 “그냥 한번 돌려보자”라는 행동이 반복되는 것 같아 이것도 아쉬웠다.