황찬웅_T5252 개인회고

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가**?**

- 우리 팀의 목표

각 개인이 분석프로젝트 end to end 경험해보기

다양한 모델 시도해보기

- 나의 목표

서버 개발환경 & 깃 익숙해지기

분석 프로젝트를 하면서 딥러닝 모델을 꼭 다뤄보기

- 목표 달성 과정

분석 프로젝트의 처음부터 끝까지 한번은 경험해보기가 우리 팀의 가장 큰 목표였지만 프로젝트 경험이 있던 저로써는 두번째 목표인 다양한 모델을 다뤄보는 데에 더 집중을 했던 것 같습니다. 기존에는 ML로만 대회를 나가거나 프로젝트를 진행하였기에 이번에는 boostcamp에서 배웠던 딥러닝 모델 중 한 개라도 꼭 사용해보기를 개인 목표로 설정했습니다. 먼저 베이스라인 코드로 주어진 모델 들의 코드를 분석하였고, 그 중 딥러닝 모델들은 특히 더 집중해서 읽어보았습니다. 이를 통해 최종분석에 사용된 DL+ML 하이브리드 모델까지 만들어 보았습니다.

또한 기존에 해보았던 방법들도 리마인드 했습니다. 주로 정형데이터를 다루는 캐글이나 데이콘 등에서 자주 사용하였던 부스팅 기반 모델들 (xgb, lgb, cat) 을 데이터셋에 맞게 수정하였고, 'Optuna' 라이브러리를 통한 하이퍼 파라미터 튜닝도 진행해보았습니다. 비록 이번에도 라이브러리를 가져와서 사용하기는 하였지만, 예전에는 단지 성능이 좋아서 사용했더라면, 이번에는 각 모델들의 특징과 작동 방식 등을 함께 공부하였고, 어떤 모델이 어떠한 이유로 해당 종류의 데이터 셋에서 성능이 올라간다는 것을 알게 되었습니다.

처음에는 서버환경에서의 개발에 적응하는 것도 시간이 꽤 오래 걸렸습니다. 기존에서는 'Colab'이나 로컬환경에서의 주피터 노트북에서만 개발하였기 때문에 서버 환경에서의 개발경험은 매우 새로운 경험이었습니다. 개인 별 할당된 서버를 VS code와 SSH를 연결해서 사용하였고, 각자의 결과물을 팀 git의 각자의 branch에 업로드 하는 방식으로 프로젝트를 진행하였습니다. 처음해보는 것이어서 이런 과정을 혼자 하기는 버거웠고, 팀원들의 도움을 받아 안정적으로 개발환경을 구축할 수 있었습니다. 앞선 과정들이 낯설기는 하였지만, 기존의 방식보다 훨씬 협업하는데 용이하였고, 버전관리도 손 쉬웠던것 같습니다.

2. 나는 어떤 방식으로 모델을 개선했는가**?**

프로젝트 기간에 제가 집중적으로 담당했던 모델은 catboost와 hybrid model (FFM+DCN) 입니다.

모델 공통적으로 K-Fold validation을 통해 train data에서 valid set을 만들어 모델의 일반화와 과적합을 방지했습니다. Target feature인 rating feature의 히스토그램을 살펴본 결과 1~10 중 7~10에 점수가 편향되어 있음을 알 수 있었습니다, 따라서 K-Fold 방식을 Standard 방식이 아닌 각 분할에서의 클래스 비율이 동일하도록 데이터를 분할하는 방법인 Stratified 방식을 채택했습니다.

Catboost는 모델의 특징에 맞게 데이터의 수치형 변수를 범주형 변수로 변환하는 전처리를 진행해 주었고, Optuna라는 하이퍼파라미터 튜닝 라이브러리를 이용하여 grid나 random search보다 효율적으로 튜닝을 진행하였습니다.

마지막으로 두 모델의 결과를 stacking 방식으로 가중평균을 취하는 앙상블을 진행하여 최종 예측 성능을 높이고자 노력했습니다.

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

같은 모델이라도 데이터에 어떤 처리를 해주는지에 따라 결과가 달라질 수 있음을 알게 되었습니다. 처음 K-Fold를 진행했을 때는 일반적인 Standard 방식으로 진행했지만, Stratified 방식으로 변경하자 모델의 train loss 와 더 빠르게 개선되었고, 이를 통한 성능개선도 있었습니다. 또한 catboost의 경우에는 예전에는 범주형 feature만 사용가능한 모델이라한다면 numerical한 feature은 drop해주었을 텐데 별도의 처리를 통해 모델에 함께 넣어주면 더 좋은 결과를 얻을 수 있음을 알게 되었습니다.

구조가 단순한 기본 모델이라도 데이터를 알맞게 전처리하고 모델을 조정하면 만족스러운 성능을 보여줄 수 있지만, 상대적으로 최근에 개발된 고도화된 모델이라도 적합한 데이터가 사용되지 않는다면 좋은 성능을 보여주는 것이 어려울 수도 있다는 것을 알게 되었습니다.

4. 전과 비교하여 새롭게 시도한 변화는 무엇이고**,** 어떤 효과가 있었는가**?**

기존에는 분석 프로젝트나 대회에서 머신러닝 모델만으로 진행했었습니다. 그래서 이번 기회에 딥러닝 모델을 꼭 사용해보고 싶었습니다. 베이스라인 모델들 중 딥러닝 모델들이 상대적으로 성능이 좋지 않았는데, 아무래도 데이터가 정형데이터이면서 피처의 수도 적도 절대적인 양이 많지 않다 보니 그런 것 같았습니다. 단순히 'DL모델을 사용해봤다'에 의의를 둘 수 있었지만, 성능도 포기하고 싶지 않았습니다. 따라서 ML과 DL 각각의 장점을 살린 하이브리드 모델에 관심을 갖게 되었고, 결과적으로 FFM의

분해과정에서 생기는 latent factor를 DCN의 인풋으로 넣는 하이브리드 모델을 설계하게 되었습니다. 이 결과 기존 DCN의 성능을 뛰어넘었을 뿐 아니라 정통적으로 정형데이터 분석에 강세를 보이는 부스팅 모델들 과도 성능을 나란히 할 수 있었습니다.

5. 마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**

실력의 문제로 구현을 다 하지 못해 Tabnet의 성능 확인을 못해본 것이 아쉽습니다. 처음 하이브리드 모델 아이디어를 얻은 곳도 Tabnet을 소개하는 글이었고, 마스터클래스 발표에서 어떤 캠퍼 분도 해당 모델 사용은 고려해보지 않으셨냐는 질문을 해주셨습니다. 실제로 Tabnet 모델을 우리 데이터에 맞게 수정을 시도 했지만, 결국 완성하지는 못했습니다. 현재의 실력으로는 베이스라인 코드의 모델들을 수정해서 사용하는 수준에 그쳤고 새로운 외부 모델의 구현에는 한계가 있었습니다.

목표 중 하나였던 깃을 잘 사용하지 않았던 것 같습니다. 구인구팀, 오프라인 모임 등 생각보다 프로젝트 기간이 빠듯하였고, 모델링에 집중하다 보니 전처리나 다른 부분에 소홀했던 것 같습니다. 특히 git 활용은 앞으로도 중요한 요소 중 하나이면서, 다루는 데 익숙하지 않아 꼭 하고 넘어갔어야 하는 부분인데, 미루고 미루다 보니 기본적인 것도 많이 놓친 것 같습니다.

모델링에 집중하느라 전처리 부분에 신경을 많이 못쓴 것 같다. 팀원들 중 전처리에 관심을 갖고 열심히 하는 팀원이 있어서 기본적인 전처리 이후 과정은 전적으로 맡긴 것 같은데, 마스터님의 피드백을 통해 아웃라이어 처리나, feature engineering 등을 고려해보지 않았다는 것을 알게 되었습니다. 그 점에 대해 분명 알고 있었지만, 지금 하고 있는 것에 집중해 놓치고 말았습니다. 직접 하기 힘들었다면 조언을 해 줄수도 있었을 것 같은데, 그 마저도 생각하지 못했습니다. 분명 전처리를 통해서도 모델의 성능을 더 높일 수 있었는데 이런 점을 놓쳤다는 것이 아쉬웠습니다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

다음 프로젝트에서도 딥러닝 모델을 한개 이상 다뤄보고 싶습니다. 이번에는 모델 구현정도로만 그쳤지만 다음에는 딥러닝모델의 튜닝도 해보고 싶습니다. 베이스라인코드를 단순히 수정하는것에 넘어서 에폭조절이나 hidden layer의 추가뿐 아니라 마스터님이 피드백 해주신 attenion layer를 적용해서 모델의 성능을 높여보고 싶습니다. 또한 이번에는 인과과정이 바뀌기는 했지만, 발표를 준비하면서 제가 했던 분석에 대해 모델 선택 이유와 아이디어 도출 과정 등을 생각해보면서 좀 더 깊이 있는 공부를 할 수 있었습니다. 다음 프로젝트에도 발표를 하게 될지는 모르겠지만, 제 분석에 대한 근거를 찾는 행위에서 얻을 수 있는 것이 많았던 것 같습니다. 이 과정에서 고정관념처럼 해오던 것들이 해당 경우에서는 적절하지 못한것이 있었고, 더 좋은 방안을 찾았던 적도 있었습니다. 따라서 다음에는 정석대로 분석을 하면서 근거를 찾는 방식으로 진행해볼 것입니다.

프로젝트 일정관리를 크게 하지 않아 한 부분에 너무 집중하여 피처엔지니어링 등의 다른 부분을 놓쳤는데, 다음에는 전반적인 일정과 데드라인을 설정하여 늦어도 그 기한을 넘기지는 않음으로서 (꼭 필요한게 아니라면 충분한 시간을 갖고도 잘 진행되지 않는다면 과감히 포기하는것도 괜찮은 방법일것 같습니다.) 다른 프로세스에도 지금보다는 좀 더 관심을 갖아야 겠다고 생각했습니다.