3주차 결과보고서

 전공: 컴퓨터공학과 학년: 2학년 학번: 20221519 이름: 구현모

1. 이번 대회에 사용한 예측 모델은 무엇이며, 선정한 기준은 무엇인가요?

이번 대회에서 사용한 예측모델은 XGBoostClassifier 모델이다.

이 모델을 선택한 이유는 우선 대회의 목표가 Clicked가 0인지 1인지를 예측하는 이진 분류 문제이기 때문이다. XGBoostClassifier 모델은 분류 문제에서 좋은 성능을 내는 모델임으로 이진 분류 문제를 푸는데 적합한 모델이다.

XGBoost는 트리 기반 모델의 앙상블을 통해 학습한다. 복잡한 패턴을 학습할 수 있고 과적합을 방지하기 위한 정규화와 pruning 같은 기법이 포함되어 있다. 그렇기 때문에 데이터가 비선형적이고 복잡할 때도 예측을 잘 한다.

Hyper-parameter를 세세하게 조절할 수 있다. 실제로 Hyper-parameter를 다르게 한다면 결과도 크게 달라지기 때문에 optuna 등으로 적절한 Hyper-parameter를 찾을 수 있는 XGBoostClassifier모델이 적합하다고 생각했다.

또한 XGBoostClassifier모델은 특성 중요도를 제공하기 때문에 어떤 데이터가 중요하게 사용되었는지 확인할 수 있다. 이러한 모델의 특징은 데이터 전처리와 feature engineering에 도움이 될 것이기 때문에 XGBoostClassifier모델을 선정하였다

2. 데이터 전처리를 수행했다면 어떤 기준으로 데이터를 전처리했나요? 데이터 전처리가 예측 성능 향상에 도움이 되었나요?

데이터가 학습에는 큰 도움이 되지 않고 복잡도를 올린다면 삭제하고 학습에 도움이 되는 데이터에 결측치가 있다면 최대한 정확하게 채우는 방법을 찾아서 결측치를 채워주는 것을 기준 삼아 데이터 전처리를 수행하였다.

click\_id의 경우 단지 각 클릭 로그의 ID이기 때문에 학습을 한다면 복잡도를 올리고 좋은 영향은 기대하기 힘들기 때문에 제거하였다. Country의 경우 데이터가 특정 국가에만 치우쳐져 있기 때문에 학습에 대해서 편견을 만들 수 있기 때문에 제거하였다.

page1의 데이터는 학습에 중요하게 쓰이고 결측치는 page2를 보고 유추할 수 있기 때문에 page2를 기반으로 결측치를 채워주었다. Page2는 page1과 연관이 크기 때문에 page2의 결측치의 경우 page1에서 같은 분류에 해당하는 최빈값으로 결측치를 채워주었다.

이렇듯이 복잡도를 올리고 학습의 도움이 되지 않을 만한 데이터들은 삭제하였고 학습의 도움이 될 만한 데이터의 경우 결측치를 채워주었다.

실제로 별다른 데이터 전처리를 하지 않은 기본 제공된 코드의 경우 정확도가 0.664정도가 나왔지만 여러 데이터 전처리를 해준 뒤 모델을 학습시킨 경우는 정확도가 높아졌다.

3. feature engineering을 수행하였나요? 어떤 방법을 적용했을 때 성능 향상이 있었나요? feature engineering의 효과가 컸나요?

date의 경우에는 주기성을 확인하기 위하여 월, 화, 수, 목, 금, 토, 일, 요일를 각각 표시해주었으며 보통 주말에 더 많이 쇼핑을 할 것이라고 생각하여 평일과 주말로 나누어 주었다. 단순한 날짜의 나열보다 요일별 데이터가 학습에 더 도움이 될 것이라고 생각하였다.

order의 경우 특정 구간에서 clicked가 집중된 모습을 보여서 1~2, 3~6, 그 이상으로 구간화 하여 학습에 도움이 되도록 하였다. 이는 clicked와 order 관계를 더 뚜렷하게 만들어서 모델이 패턴을 학습할 가능성을 높이는데 도움을 주었을 것으로 예상된다.

Location의 경우 너무 많이 나누어져 있어 위, 아래로 나누어 데이터를 단순화해주었다. 이는 복잡성은 줄이면서 중요한 정보는 학습할 수 있도록 하는데 도움이 되었을 것이라고 생각한다.

Page 데이터의 경우에도 구간별로 나누어 복잡성을 줄여주었다.

feature engineering을 하기 전에는 0.68정도의 정확도가 나왔지만 feature engineering을 한 이후 0.69정도까지 정확도가 높아졌다. 특히 Order에 대해 feature engineering을 한 후 정확도가 크게 올랐다. feature engineering의 효과가 컸다.

4. Hyper-parameter tuning의 효과는 어땠나요? 적절한 방식을 수행했나요? 최적의 hyper-parameter 조합을 찾았다고 생각하나요?

기본값을 사용하여 모델이 데이터를 학습하였을 때는 데이터 전처리와 feature engineering을 한 상태에도 불구하고 0.646정도로 정확도가 낮게 나왔다. 하지만 optuna를 이용하여 Hyper-parameter tuning해 모델에 적합한 Hyper-parameter를 찾아 주어 그 파라미터로 모델을 학습시켰더니 0.69정도가 나왔다. 또한 Train ACC의 값만 높은 과적합되는 모델이 아니라 Train ACC와 test ACC의 차이가 적은 일반화가 잘 되는 Hyper-parameter을 찾아 주었다.

Hyper-parameter를 하나하나 직접 바꾸어 가며 비교하는 것보다 optuna를 이용하여 최적의 hyper-parameter 조합을 찾는 것이 정확도와 속도 면에서 월등이 좋기 때문에 적절한 방식이다.

데이터가 같은 상태에서 Hyper-parameter tuning만을 통하여 정확도가 크게 상승하였고 일반화도 잘 되었기 때문에 최적의 hyper-parameter 조합을 찾았다고 생각한다.

5.Public Score와 Private Score가 많이 차이가 나나요? 차이가 크다면 왜 차이가 크다고 생각하나요?

Public Score와 Private Score의 차이가 크지 않다.

(Public Score)

(Private Score)

각각 Public Score와 Private Score이다. 등수는 Public에 비해서 Private가 1 떨어졌으며 Score의 경우 Public에 비해서 Private가 0.00242만큼 올랐다. 완벽하게 최적화되지 않았기 때문에 이런 차이가 발생했다고 볼 수 있다. 하지만 Public Score와 Private Score의 차이가 그다지 크지 않기 때문에 본인이 만든 머신러닝 모델은 최적화가 잘된 편이라고 볼 수 있다고 생각한다.