정형 데이터 분석

Wrap up Report

T1160 이정현

Contents

1. 기술적인 도전

정형 데이터 분석은 다른 데이터 분석과 다르게 나에게 익숙하지 않은 다양한 기술들이 모여있어 모두 불가능해 보였던 도전에 가까웠습니다.

큰 틀에서는 EDA, 데이터 전처리, 피처 엔지니어링, 모델 선택, 앙상블 과 같은 순서로 진행했기 때문에 다른 프로젝트와 일정 진행이 비슷하였습니다. 하지만 데이터가 가지는 숨은 의미를 피처로 이끌어내는 것이 굉장히 도전적이었습니다.

- 리더보드 점수 및 순위

  LB 점수 0.8537, 51등

- 검증(Validation) 전략

  K-stratified Fold Strategy (10 fold)

  10개의 train+valid 셋으로 나누어 train 정확도가 100 round동안 증가하지 않으면 정지하는 early stopping방식을 사용하였습니다.

- 사용한 모델 아키텍처 및 하이퍼 파라미터

  여러 모델을 사용하였으나 결과적으로 제 LightGBM 과 XGBoost를 선택하여 제출에 활용하였습니다.

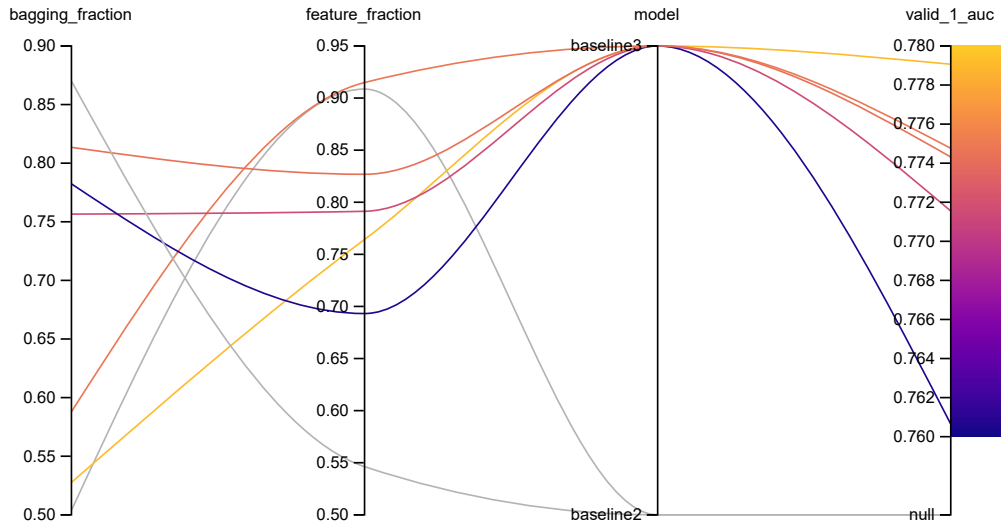
  1. 아키텍처: **LightGBM**

     a. LB 점수: **0.8537**

     b. 하이퍼 파라미터

```
- feature_fraction: 0.7643
- bagging_fraction: 0.5216
- bagging_freq: 1
- n_estimators: 10000
- early_stopping_rounds: 100
- verbose: -1
- n_jobs: -1
```

c. Parameter 튜닝: **wandb**의 sweeping기능을 통해 제한된 수의 학습시도 내에서 최적의 feature_fraction과 bagging_fraction을 찾고, 이를 적용하여 리더보드에 제출한 결과입니다.



d. 주요 피처:

- 주문건당 총지출, 주문량 등 주요 컬럼에 대한 대표 통계 피처 (sum, mean, min, max, std,.. etc)
- 데이터 row별 time difference
- 고객을 기준으로 총지출의 합계를 구한 값에 대한 quantile cut (고객 등급)
- 2012-11기준 1, 2, 3, 5, 7, 12, 20, 23개월 전으로 고정된 시점에서부터 기준 피처를 누적 각각에 대한 대표 통계 피처

2. 아키텍처: **LightGBM**

a. LB 점수: **0.8528**

b. 하이퍼 파라미터

- feature_fraction: 0.8
- bagging_fraction: 0.8
- bagging_freq: 1
- n_estimators: 10000
- early_stopping_rounds: 100
- verbose: -1
- n_jobs: -1

c. 각 데이터간의 주문시각의 차이값을 계산하여 피처로 추가하였습니다.

3. 아키텍처: **LightGBM**

a. LB 점수: **0.8526**

b. 하이퍼 파라미터

- feature_fraction: 0.8
- bagging_fraction: 0.8
- bagging_freq: 1
- n_estimators: 10000
- early_stopping_rounds: 100
- verbose: -1
- n_jobs: -1

c. Time series feature를 누적하여 피처로 추가하였습니다. 시계열 데이터로 확장하여 특정 기간동안의 total 관련 피처들을 대표 그룹함수로 종합한 형태입니다.

4. 아키텍처: **LightGBM + XGBoost** (Emsemble)

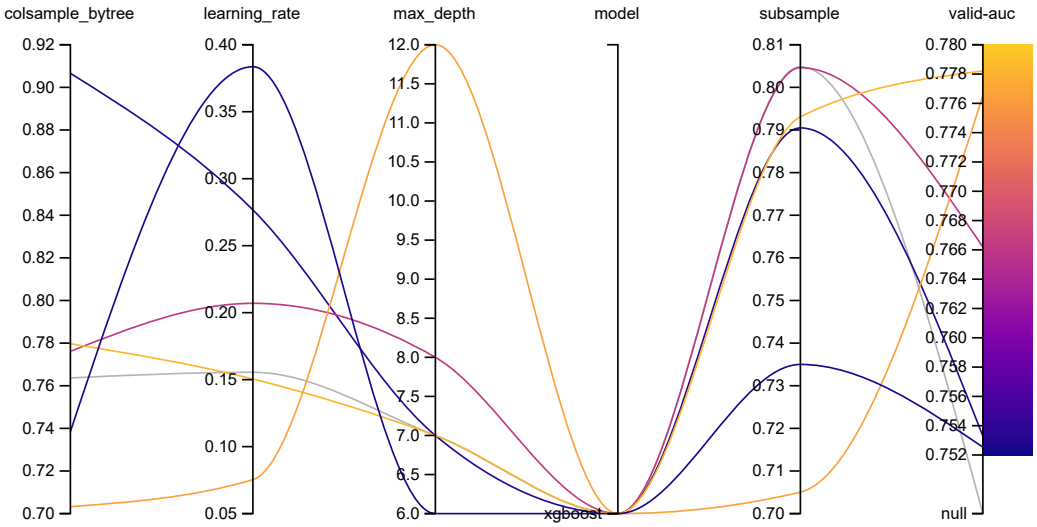
a. LB 점수: 0.8519

b. 하이퍼 파라미터

LightGBM	**XGBoost**
LB Score: 0.8537	LB Score: 0.8434
feature_fraction: 0.7643	learning_rate: 0.1505
bagging_fraction: 0.5216	max_depth: 7
bagging_freq: 1	colsample_bytree: 0.7797
n_estimators: 10000	subsample: 0.7931
early_stopping_rounds: 100	
verbose: -1	
n_jobs: -1	

c. 피처들을 종합한 상태에서 가장 점수가 양호한 두 모델의 결과를 앙상블하였습니다. probability를 계산한 두 결과를 평균을 내어 이용하였습니다. 최고점수를 받은 모델의 성능과 하위점수의 다른 모델의 결과를 종합하여 이용한 경우 새롭게 나오는 결과는 더 나은 결과를 제공하지 못했습니다.

d. XGBoost Parameter 튜닝: **wandb**의 sweeping기능을 통해 제한된 수의 학습시도 내에서 최적의 파라메터 조합(위의 표 참조)을 찾고, 이를 적용한 결과를 앙상블에 활용하였습니다.



5. 아키텍처: **LightGBM**

a. LB점수: 0.8365

b. 하이퍼 파라미터

- feature_fraction: 0.8
- bagging_fraction: 0.8
- bagging_freq: 1
- n_estimators: 10000
- early_stopping_rounds: 100
- verbose: -1
- n_jobs: -1

c. 피처엔지니어링 시도: 고객의 총 구매액을 기준으로 등급을 quantile하게 나누어서 A~E의 다섯개 등급을 부여하였습니다. Feature 중 **가장 영향력이 큰 Feature**에 해당하는 것을 발견하였기 때문에 기록할만한 가치가 있다고 보았습니다. (0.8100 > 0.8434)

- 앙상블 방법
  - 단순 평균 방법을 사용하였습니다.

- 시도했으나 잘 되지 않았던 것들
  - 코드 리팩토링
    - 데이터에 피처를 추가하는 과정에서 적절히 코드를 재사용하는데 어려움을 겪었다.

      애초에 계획했던 것은 'customer_id'인덱스 혹은 첫 컬럼에 존재하면서 피처를 추출하는 각 함수에서 반환된 데이터 프레임들을 merge를 통해 새로이 추가된 피처들을 선택적으로 추가할 수 있도록 하고자 하였다. 하지만 merge의 경우가 default인 inner 말고도 index-wise하게 left/right을 설정하거나 outer, left, right와 같은 how파라미터에 따라 다양한 경우가 준재했고, 데이터를 세밀하게 가공하여 중복되는 column을 고려하여 작성하는 부분이 어려웠고 따라서 커스텀하게 함수를 코딩하는 부분이 큰 어려움으로 다가왔다.
  - Auto ML 셋팅 (Wandb)
    - Sweeping 부분의 Wandb를 셋팅하는데 어려움을 겪었다. 파라미터 검색 방식(Random, Baysian, Grid), 파라미터의 타입이나 범위의 분포에 따라 distribution을 구체적으로 명세하는 경우(Random 이나 Bayse는 필수로 작성)도 있었다.
    - 프로젝트내에서 model간의 성능을 시각적으로 비교하는 방법으로 하나의 그래프에서 메트릭을 확인하는 것을 시도하였다. 그런데 auc 스코어로 표시되는 부분이 valid_auc, valid_1_auc등으로 다르게 명명되어 그래프가 분리되어 그려졌다. 직접적인 영향은 없었지만 개선해야 할 부분이라고 생각한다.
  - CV Score 별로 Catboost, LightGBM, XGBoost 간의 모델적합성 비교하기
    - wandb내에서 catboost의 feature importance를 그려주는 부분을 찾아보긴 하였지만 대시보드 내에서 적용될 수 있도록 코드를 수정하는 자료를 찾지 못했다. 제외하면서 다른 모델들을 찾고 서로 비교하는 방법으로 실험을 셋팅하였다.

- 학습과정에서의 교훈
  - EDA와 데이터 시각화의 중요성

    데이터를 탐색하고 전처리를 통해서 데이터의 의미를 파악하는 과정이 있어야 좋은 피처를 추출할 수 있음을 알게 되었습니다.
  - 대회과정에서의 프로세스에 부여한 계획은 바뀌는 것이고, 실행하면서 이 오차를 줄이는 것

    미리 충분한 시간을 새로운 시도 및 도전에 활용하고 토론을 활발하게 이용하면서 다른 시야를 갖춘 참가자들과 넓고 깊게 소통하는 것이 꼭 필요하다는 것을 깨달았습니다.

- 마주한 한계와 도전과제
  - 판다스 프레임워크의 러닝커브를 빠르게 올리려면 어떻게 해야하는지 깨달았습니다.

    이론 학습에서 나오는 데이터보다 실제 문제에 활용되는 데이터를 실습으로 손을 더럽히면서 코딩하면 빨라진다는 것이었습니다. 이것은 판다스와 비슷한 라이브러리를 배울때도 유효할 것이라고 생각합니다. 캐글등의 컴피티션이 가진 장점이기도 합니다. 실제 문제에 뛰어들어 해결하는 습관을 들이면서 더 성장하고 싶습니다.
  - 시계열 Validation을 구현하지 못했습니다.

피처 엔지니어링을 거친 데이터는 index가 customer_id로 이루어진 데이터와 피처들의 데이터프레임이었습니다. 다른 피처엔지니어링을 적용해서 Time Series로 인덱싱한 데이터들을 가지고 K-fold validation을 했다면 좀더 실제적인 문제접근 방법을 구현할 수 있었을 것입니다.

- TabNet을 활용하지 못한 아쉬움

  딥러닝이 적용된 TabNet을 문제에 적용하지 못해 아쉬움이 있었습니다. 다음 컴피티션에서 적용해보거나 이번에 주어진 데이터를 가공하여 Validation 스코어를 확인해볼 수 있을 것입니다.

Reference

- https://docs.wandb.ai/guides/sweeps/configuration#parameters
- Picking the best model: A Whirlwind Tour of Model
- Time Series Nested Cross Validation