프로젝트 진행상황

AI0팀

Task, Pipeline

Task : **고객이 했던 주문정보를 보고 다음 주문을 예측하는 다중 레벨 분류**

Pipeline :
- **오토인코더로 주문 제품을 임베딩 벡터로 변환**
-**임베딩 벡터와 데이터셋으로 프레딕터 학습**

현재 진행상황

1. 오토인코더로 제품 인베딩 벡터 변환 - 완료
2. 프레딕터 CNN->GRU->(예측)구조
   - 훈련 후 평가 완료
3. XGBoost 와 비교
   - 진행중

CNNGRU predictor 성능평가

개선점 : CNN **벡터압축에서** dropout**이 있었는데 예측률이 떨어진다 판단하여** dropout**은 사용하지 않는 것으로 변경**

하이퍼파라미터

- CNN **출력 차원** : 128
- GRU **은닉 상태 차원** : 256
- **드롭아웃 비율** : 0.1
- **학습률** : 0.001
- **구매로 판단할 확률 임계값** : 0.4
- CNN **커널 크기** : 4
- GRU **계층 수** : 2
- **예측 헤드 중간 차원** : 256
- **배치 크기** : 32
- pos_weight : 10

CNNGRU predictor 성능평가

결과

Loss : 0.0089
F1_micro_overall : 0.1297
F1_macro : 0.0009
Precision@10 : 0.1386
Recall@10 : 0.1219
Precision@20 : 0.1386
Recall@20 : 0.1219

제품 성능을 높이기 위한 시도(predictor)

- 임베딩 벡터 차원 조절해보기

차원	F1_micro_overall	F1_macro	Precision@10	Recall@10	Precision@20	Recall@20
16	0.1295	0.0012	0.1329	0.1262	0.1329	0.1262
32	0.1331	0.0011	0.1467	0.1218	0.1467	0.1218
64	0.1297	0.0009	0.1386	0.1219	0.1386	0.1219

=> 유의미한 향상이 있지는 않아서 32차원 사용결정

제품 성능을 높이기 위한 시도(predictor)

- CNN만으로는 kernel_size의 한계가 있다는 의견
- CNN 대신에 Attention으로 학습
- 각 주문을 CNN대신 Multi-Head Self-Attention + Attention Pooling으로 압축 후 GRU -> 최종 예측 진행
- 결과
  F1_micro_overall: 0.1013
  F1_macro: 0.0002
  Precision@10: 0.1245
  Recall@10: 0.0854
  Precision@20: 0.1245
  Recall@20: 0.0854

=> CNN이 더 높은 성능

제품 성능을 높이기 위한 시도(predictor)

- 계획과 달리 제품 정보만 넣고있었던 부분에 초기계획대로 주문 정보도 추가
- 주문 정보도 GRU에 투입

- 주문 정보에
  order_dow_sin/cos(주문요일을 sin/cos 범주로 표현),
  order_hour_of_day_sin/cos(주문시간을 sin/cos 범주로 표현),
  days_since_prior_order(마지막 주문으로부터 경과일) 를 넣고
  days_since_prior_order 가 첫 주문일때에는 -1을 대입

제품 성능을 높이기 위한 시도(predictor)

- **결과**
  F1_micro_overall: 0.1204
  F1_macro: 0.0007
  Precision@10: 0.1324
  Recall@10: 0.1103
  Precision@20: 0.1324
  Recall@20: 0.1103