수상작리뷰
[Private 1st - 0.10554] PolynomialFeatures, Round+ Ensembles - DACON

[칼로리 소모량 예측 AI 해커톤]

◈ 주제

👉 생체 데이터를 이용해 칼로리 소모량 예측하는 회귀 모델 만들기

◈ 데이터

- train.csv : ID, 운동시간(분), 체온, 심박수, 키, 몸무게, 체중상태, 성별, 나이, 칼로리 소모량
- test.csv : ID, 운동시간(분), 체온, 심박수, 키, 몸무게, 체중상태, 성별, 나이
- sample_submission.csv : ID , 칼로리 소모량
- 데이터명세.xlsx : 고객 금융활동 데이터에 대한 명세

◈ 코드 리뷰

(1) 라이브러리 임포트, 시드 고정, 데이터 불러오기

- autogluon 설치해 사용

(2) 데이터 전처리

- train, test 데이터세트에서 x, y 분리
- 연속형 x (운동시간, 심박수, 나이, 몸무게) 간의 히트맵 시각화
- 라벨 인코딩: 성별 (명목형 변수)
- 학습을 위한 데이터 분리 * train_x, val_x, train_y, val_y

(3) **회귀 모델링** + 검증

🔷 PolynomialFeatures 변환
🔷 선형회귀 학습
⇒ MSE: 0.2840 / 0.0966 (예측값 반올림 시)
🔷 릿지회귀 학습
⇒ MSE: 0.2836 / 0.1155 (예측값 반올림 시)

(4) 회귀 모델링 (검증X)

🔷 PolynomialFeatures 변환
🔷 선형회귀 학습
🔷 릿지회귀 학습

(5) 스태킹 수행 # autogluon 사용

[코드]
모델 생성

```
stacking = TabularPredictor(label='Calories_Burned',
eval_metric='rmse', problem_type='regression').fit(new_train,
presets=['best_quality'], num_stack_levels=0)
```

모델 학습

```
ld = stacking.leaderboard(silent=True)
LD = ld[['model', 'score_val']].rename({'score_val':'RMSE'},axis=1)
LD['RMSE'] = -LD['RMSE']
LD.head()
```

(6) 최종 예측

① Voting:
릿지, 라쏘의 반올림한 예측값에 가중치를 곱해 최종 예측치 결정 !
② Stacking

◈ 배울점

- PolynomialFeatures 변환 시, 적절한 차수를 선택하는 것이 중요함.
- 피처 가공보다 관련없는 피처를 드롭하는 것이 생각보다 성능 향상에 매우 중요.
- 보팅, 스태킹 알고리즘 생성 시, 가중치를 설정한 것이 인상깊었음.