

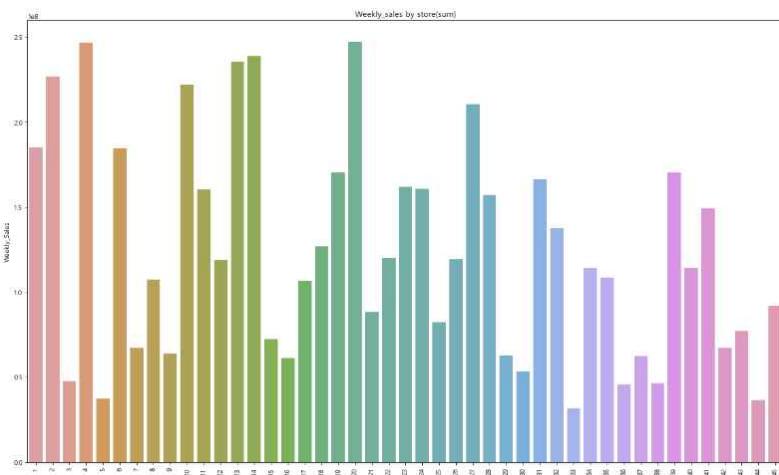
1. 주제 및 데이터

- 쇼핑몰 지점별 매출액 예측 AI 해커톤
- train 데이터: id(샘플 아이디), Store(쇼핑몰 지점), Date(주 단위 날짜), Temperature(해당 쇼핑몰 주변 기온), Fuel_Price(해당 쇼핑몰 주변 연료 가격), Promotion1~5(해당 쇼핑몰의 비식별화된 프로모션 정보), unemployment(해당 쇼핑몰 지역의 실업률), IsHoliday(해당 기간의 공휴일 포함 여부), Weekly_Sales(주간 매출액: 목표 예측값)
- test 데이터: id(샘플 아이디), Store(쇼핑몰 지점), Date(주 단위 날짜), Temperature(해당 쇼핑몰 주변 기온), Fuel_Price(해당 쇼핑몰 주변 연료 가격), Promotion1~5(해당 쇼핑몰의 비식별화된 프로모션 정보), unemployment(해당 쇼핑몰 지역의 실업률), IsHoliday(해당 기간의 공휴일 포함 여부)

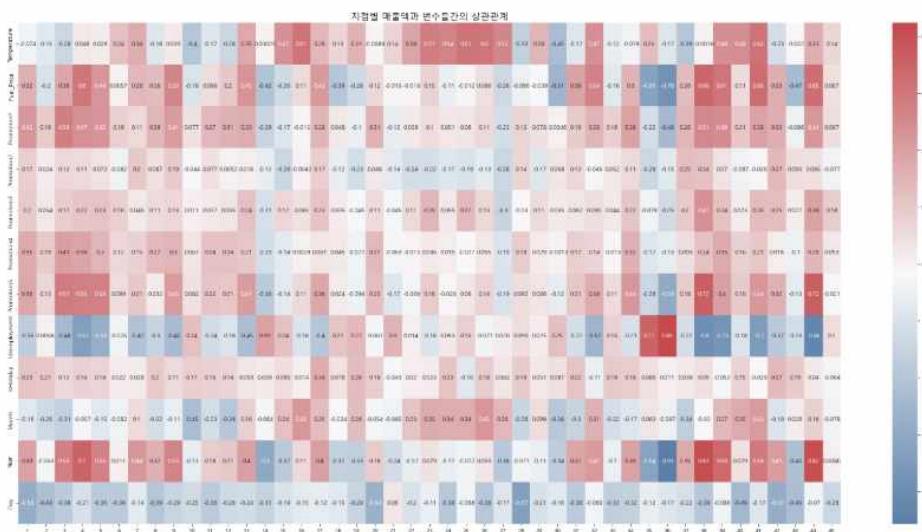
2. 코드 리뷰

1) EDA

(1) 지점별 주간 매출액 합 비교



(2) 지점별로 매출액의 차이가 크므로 지점별 매출액과 변수들의 상관관계 확인



- Store별로 Promotion과의 상관계수 중 0.4 이상인 Feature들만 선택

2) 전처리

(1) Date를 년/월/일로 나눠주기

(2) Promotion 값 유지

- Promotion 값들의 편차가 굉장히 크고, Promotion2와 Promotion3에서 음수값 존재함
- Promotion에 대한 해석이 다양하게 존재할 수 있고 음수값을 처리할만한 충분한 근거를 찾지 못해 그대로 둠
- 이후에 Feature selection 과정에서 Promotion2가 사용되지 않았으므로 음수값 처리가 큰 영향을 미치지는 않는 것으로 판단함

(3) Promotion에만 결측치를 0으로 처리

- 2011년 11월 11일 이전에도 Promotion이 진행되었지만 기록이 11월 11일 이후부터 되었다고 판단하고, 2011년 11월 11일 이전의 결측값들은 0으로 채워주고 이후의 결측값들은 선형보간법을 이용해 채워주려고 함
- 하지만 선형보간법을 적용해 결측치를 채운 모델보다 그냥 모든 결측치를 0으로 채운 모델의 성능이 더 높기 때문에 0으로 채우는 것을 선택함

(4) 2010.11월과 12월, 2011.11월과 12월 변수 삭제

- test data를 확인해보면 2012년 10월을 예측하는 것임
- 2010과 2011 모두 11월과 12월에 추수감사절, 블랙프라이데이, 크리스마스, 연말 연초 등이 포함되어 극단값을 가지는 것으로 확인되므로 이를 삭제함

3) 모델링

여러 가지 모델들과 Voting, Stacking등 여러 양상을 기법을 사용해보았지만 CatBoost 단일 모델의 성능이 가장 좋았기 때문에 이를 선택함

(1) 지점별 모델 학습

```
models = []
for store in range(1,max(train.Store)+1):
    if (store == 38):
        train_store = train[train.Store==store]
        model = CatBoostRegressor(**model_params)
        model.fit(train_store[features_1_3_5], train_store.Weekly_Sales)
        models.append(model)

    elif (store == 3):
        train_store = train[train.Store==store]
        model = CatBoostRegressor(**model_params)
        model.fit(train_store[features_1_4_5], train_store.Weekly_Sales)
        models.append(model)

    ...
    ...
```

(2) 지점별 모델 예측

```
pred = []
for store in range(1, max(test.Store)+1):
    if (store == 38):
        test_store = test[test.Store==store]
        y_pred = models[store-1].predict(test_store[features_1_3_5])
        pred += y_pred.tolist()

    elif (store == 3):
        test_store = test[test.Store==store]
        y_pred = models[store-1].predict(test_store[features_1_4_5])
        pred += y_pred.tolist()

    ...

```

3. 차별점 및 배울 점

2010년과 2011년 모두 11월, 12월에는 각종 연휴들이 많기 때문에 극단값이 많다는 것을 파악하여 10월까지의 데이터만 활용하여 예측 성능을 높인 점이 인상깊었다. 시장 특성에 따라 일정한 기준을 가지고 데이터를 전처리하는 것이 중요하다. 보팅, 스태킹 등 여러 앙상을 기법을 사용했지만 캡부스트 단일 모델의 성능이 가장 높았음을 보면 무작정 많은 모델을 사용한다고 예측 성능이 높아지는 것은 아니라는 것을 알 수 있다. 여러 조합을 통해 예측 성능을 살펴볼 필요가 있다.