2023.08.28

LG aimers 8등 solution

TEAM 치킨마요덮밥

목차

01. EDA 및 데이터 전처리

01 EDA 및 데이터 전처리

1) train 데이터 특징

15890개의 고유ID의 459일치 판매량으로 이루어진 데이터.

	ID	제품	대분류	중분류	소분류	브랜드	2022-01-01	2022-01-02		2023-04-03	2023-04-04
0	0	B002-00001-00001	B002-C001-0002	B002-C002-0007	B002-C003-0038	B002-00001	0	0		0	0
1	1	B002-00002-00001	B002-C001-0003	B002-C002-0008	B002-C003-0044	B002-00002	0	0		2	0
2	2	B002-00002-00002	B002-C001-0003	B002-C002-0008	B002-C003-0044	B002-00002	0	0	...	0	0
3	3	B002-00002-00003	B002-C001-0003	B002-C002-0008	B002-C003-0042	B002-00002	0	0		0	0
4	4	B002-00003-00001	B002-C003-0042	B002-C002-0001	B002-C003-0003	B002-00003	0	0		0	0

⋮

01 EDA 및 데이터 전처리

2) 데이터 정제(LSTM)

판매량이 일정 기간 연속으로 0인 구간을 확인 후

이상 요인(품절, 공급부재, 미등록 등)의 이유로 판단되어 제거

	ID	제품	...	2022-01-01	...	2023-04-03	2023-04-04
0	0	B002-00001-00001		0		0	0
1	1	B002-00002-00001		0		2	0
2	2	B002-00002-00002		0		0	0
3	3	B002-00002-00003		0		0	0
4	4	B002-00003-00001		0		0	0

⋮

판매량 30일 이상 0인 구간 제거

```
fin_list = []
for k in tqdm(range(train_data.shape[0])): # 15890개에 대해 반복
    a = train_data.iloc[k, 4:] # 각 제품에 대해
    period = CFG['PERIOD'] # 0인 기간 30
    fir_list = []
    i = 0 # 첫 시작
    while i < len(a)-1: # 전체 길이 동안
        sec_list = []
        if a[i] == 0: # 해당 값이 0이면
            sec_list.append(i)
            while a[i] == 0:
                i += 1
                if i == 459:
                    break
            sec_list.append(i-1)
            if (sec_list[1] - sec_list[0]) >= period:
                fir_list.append(sec_list)
        i+=1
    fin_list.append(fir_list)
```

01 EDA 및 데이터 전처리

2) 데이터 정제(ML)

2022-01-03	143218
⋮	
2022-10-06	762097
2022-10-07	751579
⋮	
2023-02-23	19422
2023-02-24	17532
⋮	
2023-03-28	7527
⋮	
2023-04-04	159393

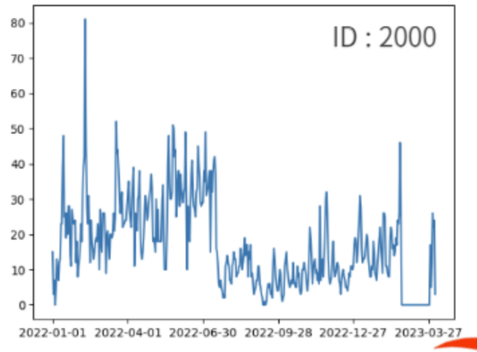
일자별 판매량 총합 계산 .

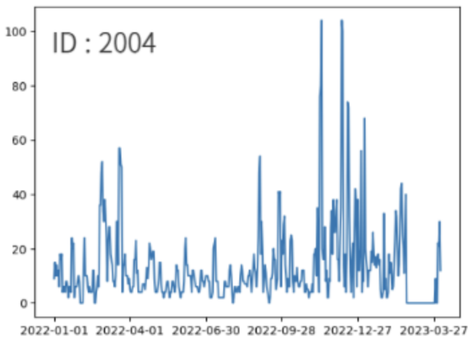
- 일자별 판매량 총합 계산 후 일정 수준에 미달하는 경우 일자 제거
- 10만개 미만, 70만개 초과인 경우 해당 일자 제거

	ID	제품	...	2022-01-01	...	2023-04-03	2023-04-04
0	0	B002-00001-00001		0		0	0
1	1	B002-00002-00001	...	0	...	2	0
2	2	B002-00002-00002		0		0	0
3	3	B002-00002-00003		0		0	0
4	4	B002-00003-00001		0		0	0
⋮				⋮			

01 EDA 및 데이터 전처리

2) 데이터 정제(ML)





ID별 판매량 차이가 커 ID별로 예측하는 것이 예측력을 높일 수 있다 판단

데이터셋 형태를 다음과 같이 변환

	ID	COUNT	제품	대분류		Year	Month	Day		COUNTS_56	COUNTS_57	COUNTS_58	COUNTS_59
0	0	0	B002-00001-00001	B002-C001-0002		2022	3	1		0	0	0	0
1	0	0	B002-00001-00001	B002-C001-0002		2022	3	2		0	0	0	0
2	0	0	B002-00001-00001	B002-C001-0002		2022	3	3		0	0	0	0
3	0	0	B002-00001-00001	B002-C001-0002	...	2022	3	4	...	0	0	0	0
4	0	0	B002-00001-00001	B002-C001-0002		2022	3	5		0	0	0	0
⋮													
5783959	15889	0	B002-03799-00010	B002-C001-0002		2023	4	4		0	0	0	0

01 EDA 및 데이터 전처리

3) 파생변수 생성

(1) 주말, 휴일, 공휴일 변수 생성 (ALL)

- 주말 및 공휴일이 판매량에 영향을 미칠 것으로 예상하여 주말, 공휴일, 휴일 변수 생성

	Year	Month	Day	Weekday	Weekday_Name	주말여부	공휴일여부	휴일여부
2022-01-01	2022	1	1	5	토	Y	Y	Y
2022-01-02	2022	1	2	6	일	Y	N	Y
2022-01-03	2022	1	3	0	월	N	N	N
2022-01-04	2022	1	4	1	화	N	N	N
2022-01-05	2022	1	5	2	수	N	N	N

01 EDA 및 데이터 전처리

3) 파생변수 생성

(2) 변동량 차/합 변수 생성(LSTM)

- 전일 판매량 대비 당일 판매량의 변동 정도를 표현하기 위해 판매량 변동량 / 전일 + 당일 판매량 합 변수 생성

$$V1_t = \begin{cases} \dfrac{y_t - y_{t-1}}{y_t + y_{t-1}}, & t : 2022-01-02 \text{ 이후} \\ 0, & t : 2022-01-01 \end{cases}$$

(3) 판매량 변동량 역수 변수 생성(LSTM)

- 판매량의 급한 변동에 대한 학습효과를 완화시키기 위해 전일 대비 판매량 변동량의 역수를 취한 변수를 생성

$$V2_t = \begin{cases} \dfrac{1}{y_t - y_{t-1}}, & t : 2022-01-02 \text{ 이후} \\ 0, & t : 2022-01-01 \end{cases}$$

02. 모델 학습

02 모델 학습

모델 검증 방법

- Rolling window 방법 이용(ML : size = 60, LSTM : size = 75)

모델 설정

- 모델 : LSTM,
  XGBOOST(n_estimator=1000),
  RandomForest(n_estimator=1000),
  ExtraTree(n_estimator = 200) 사용

LSTM

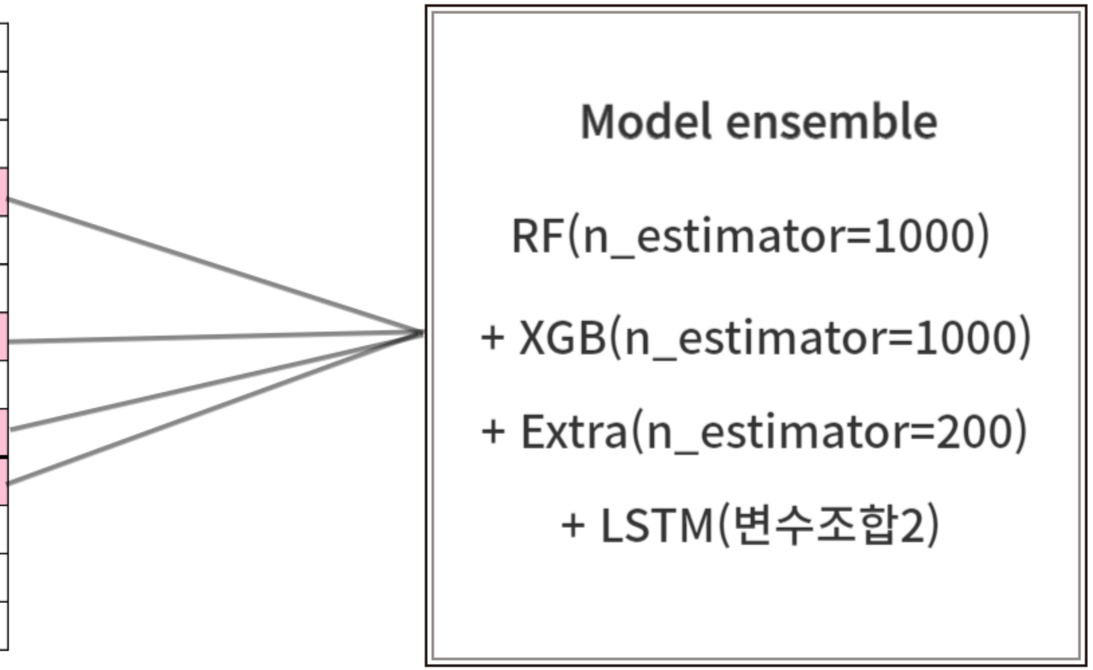
```
BaseModel(
  (lstm): LSTM(7, 512, batch_first=True)
  (fc): Sequential(
    (0): Linear(in_features=512, out_features=256, bias=True)
    (1): SiLU()
    (2): Linear(in_features=256, out_features=128, bias=True)
    (3): SiLU()
    (4): Dropout(p=0.2, inplace=False)
    (5): Linear(in_features=128, out_features=21, bias=True)
  )
  (actv): SiLU()
)
```

02 모델 학습

모델 결과 비교

- Rolling window 방법 이용(ML : size = 60, LSTM : size = 75)

Model	Loss
RF(n_estimator = 100)	17.26
RF(n_estimator = 200)	16.77
RF(n_estimator = 1000)	16.73
XGB(n_estimator = 200)	18.21
XGB(n_estimator = 500)	18.21
XGB(n_estimator = 1000)	18.21
LGBM(n_estimator = 1000)	17.82
EXTRA(n_estimator = 200)	16.17
LSTM(변수조합2)	0.02488
LSTM(변수조합1)	0.02492
LSTM(변수조합3)	0.02489
LSTM(변수조합4)	0.02490



02 모델 학습

최종 모델 Parameter

LSTM

- TRAIN_WINDOW_SIZE : 75
- EPOCHS : 10
- LEARNING_RATE : 0.0001
- BATCH_SIZE : 2048
- SEED : 9909
- PERIOD : 30

ML

- train_window_size : 60

RandomForest

- n_estimators = 1000, random_state = 41

XGBRegressor

- n_estimators = 1000, random_state = 41

ExtraTree

- n_estimators = 200, random_state = 41

03. 추론

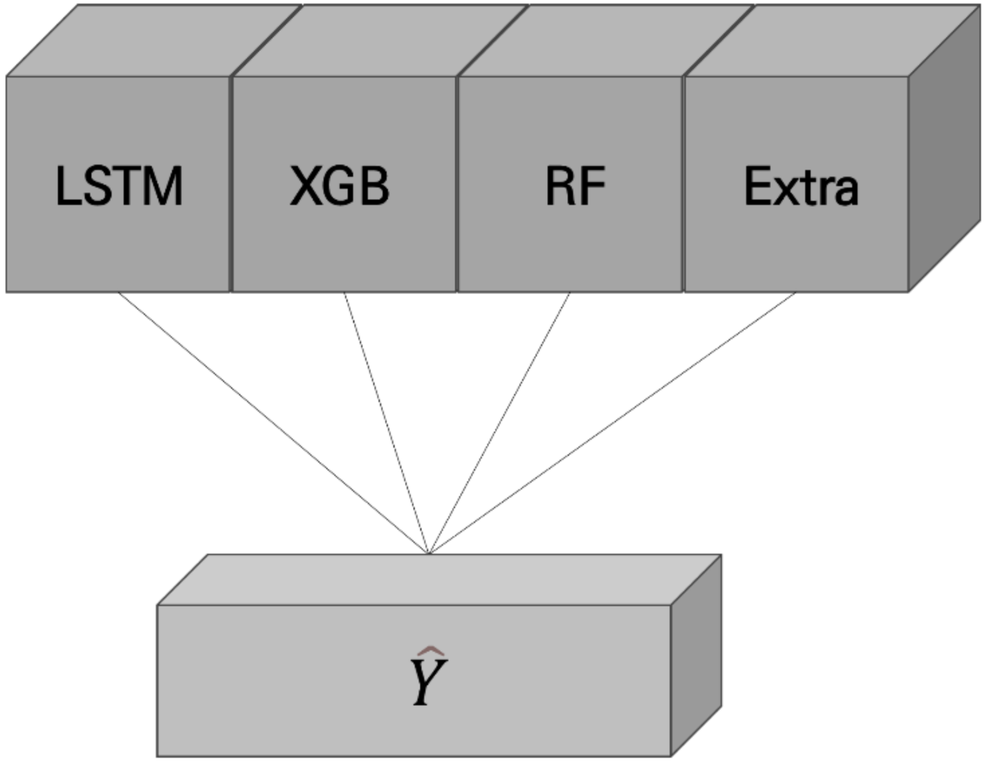
03 추론

예측

4개의 모델(LSTM, XGB, RF, Extra)을 앙상블

이후 4개 모델의 예측값의 평균을 사용

반올림하여 정수 형태로 반환



03 추론

평가 지표

최소한의 예측 가능성을 확보하고자 최소 판매량을 1로 설정

대분류 별 Pseudo 예측 정확도: $PSFA_m = 1 - \frac{1}{n}\sum_{day=1}^{n}\sum_{i=1}^{N}\left(\underbrace{\left(\frac{|y_i^{day} - p_i^{day}|}{\max(y_i^{day}, p_i^{day})}\right)}_{\text{오차}} \times \underbrace{\frac{y_i^{day}}{\sum_{i=1}^{N} y_i^{day}}}_{\text{(판매)비중}}\right)$

- m: 대분류 index
- i: (대분류 내에서) 제품 index
- y_i^{day}: i번째 제품의 day일의 판매량
- p_i^{day}: i번째 제품의 day일의 예측량

전체 Pseudo 예측 정확도: $PSFA = \frac{1}{M}\sum_{m=1}^{M} PSFA_m$

Thank you.