10월 11일 공유

1) 업데이트 내용

1. 2023년 데이터를 예측값으로 사용

In [31]:
```
# 결과 해석

result = pd.DataFrame([y_pred, y_test.Target]).T
result.columns = ['예측',"실제값"]
result = pd.concat([y_idx.reset_index(drop=True),re
result
```

Out[31]:

	연도	월	주차	예측	실제값
0	2023	1	1	38.185821	37.0
1	2023	1	2	35.942030	35.0
2	2023	1	3	35.492412	36.0
3	2023	2	1	37.077018	34.0
4	2023	2	2	36.773600	32.0

`기대효과. 더 많은 데이터를 학습하여 더 좋은 성능을 보일 것으로 예상`

2. 비율 변수(열)을 사용

   ⇒["Science_IT_P","Ecnomic_P","Global_P","North_Korea_P","Culture_P","Social_P","Issue_P", 'Politic_P', 'Sports_P', 'Local_P', 'Etc_P']

   `기대효과. 단순히 뉴스 보도량이 아닌 그 시기(해당 주차)에 어느 뉴스 카테고리에 집중되었는지 알 수 있다.`

3. 이전 달의 긍정, 부정, 잘모름 정도를 활용

   ⇒ ["긍정_과거(Positive_Past)","부정_과거(Negative_Past)","잘모름_과거(Non_Past)”]

   `기대효과. 시계열적인 특성을 활용`

4. 이전 달과 비교하여 뉴스 카테고리별 증감량 확인

   ⇒["Science_IT_ID","Ecnomic_ID","Global_ID","North_Korea_ID","Culture_ID","Social_ID","Issue_ID", 'Politic_ID', 'Sports_ID', 'Local_ID', 'Etc_ID’]

   `기대효과. 어느 카테고리가 증가 혹은 감소 할 시 긍, 부정에 영향을 끼치는지 알 수 있다.`

2) 모델링 결과

예측 결과 해석

	연도	월	주차	예측	실제값
0	2023	1	1	57.567362	37.0
1	2023	1	2	44.068627	35.0
2	2023	1	3	33.252183	36.0
3	2023	2	1	29.090435	34.0
4	2023	2	2	41.160612	32.0
5	2023	2	3	36.814931	35.0
6	2023	2	4	42.539644	37.0
7	2023	3	1	36.672715	36.0
8	2023	3	2	37.336986	34.0
9	2023	3	3	33.044060	33.0

이전 모델(BaseLine)

	연도	월	주차	예측	실제값
0	2023	1	1	38.185821	37.0
1	2023	1	2	35.942030	35.0
2	2023	1	3	35.492412	36.0
3	2023	2	1	37.077018	34.0
4	2023	2	2	36.773600	32.0
5	2023	2	3	32.269157	35.0
6	2023	2	4	37.062892	37.0
7	2023	3	1	37.070276	36.0
8	2023	3	2	34.772727	34.0
9	2023	3	3	34.467469	33.0

최신 모델(Ver2)

- 2023년 1월 1주차, 2월 4주차 값을 보면 실제로 이전 모델보다 더 긍정 정도를 잘 맞추는 모습이다.

변수 중요도 해석

	0	1
9	Local	193
7	Politic	169
10	Etc	165
11	SUM	164
6	Issue	162

이전 모델(BaseLine)

	0	1
0	Positive_Past	177
22	Politic_P	92
5	Global	78
10	Politic	73
12	Local	67

최신 모델(Ver2)

- 최신 모델의 변수 중요도를 살펴보면 확실히 **한달 전의 긍정 정도**가 높은 중요도를 보이는 것으로 확인
- 그리고 해당 주의 **정치 기사의 비중**이 긍정에 높은 중요도를 보인다.

성능 지표 확인

```
In [19]: ## 성능 지표 확인
         from sklearn.metrics import mean_absolute_error

         metric_score = mean_absolute_error(y_pred, y_test.Target)
         display(metric_score)

4.367878952501611
```

이전 모델(BaseLine)

```
In [33]: ## 성능 지표 확인
         from sklearn.metrics import mean_absolute_error

         metric_score = mean_absolute_error(y_pred, y_test.Target)
         display(metric_score)

2.2500202599125068
```

최신 모델(Ver2)

- 해당 성능 지표는 MAE라는 성능 지표로 회귀 모델에서 쓰이는 성능 지표이다.
- MAE는 결과값이 낮을수록 좋다.
- 이전 모델에 비해 최신 모델이 더 좋은 모습을 보여준다.

3) 추후 진행 예정 사항

1) 주제 * 주제 변수 투입 방안 고려