시계열 회귀 모델을 활용한 미국 소기업 밀도 예측

2023.02.05

비타민 10기 자율 | 복습 프로젝트 발표

5조 | 이두진 임홍주 전민주 최대상

목차

PART 1.
프로젝트 소개

01 주제 선정 배경
02 데이터셋 소개
03 평가지표

PART 2.
분석 방법 1

01 교차 검증 방법 구축
02 전처리

PART 3.
분석 방법 2

01 모델 설계
02 하이퍼 파라미터 튜닝
03 모델 학습 및 전략

PART 4.
결과 해석

01 결과 해석
02 앞으로의 계획

PART 1.
프로젝트 소개

01 주제 선정 배경
02 데이터셋 소개
03 평가지표

Microbusiness Density Forecasting

! 지도학습모델

교차검증 / 그리드서치

이상치 탐지 / 특성공학

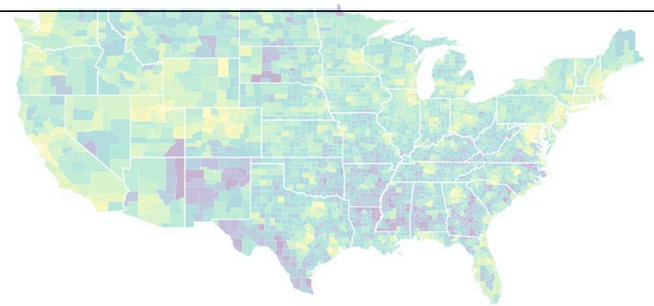
손실함수
/ 과대적합 제어 및 규제

? 시계열 데이터

지난 한 학기 동안 다룬 모든 내용의 응용
&
다루지 않은 시계열 데이터

II

시계열 데이터의 지도학습 예측 모델을 주제로 한
KAGGLE 경진대회 참가



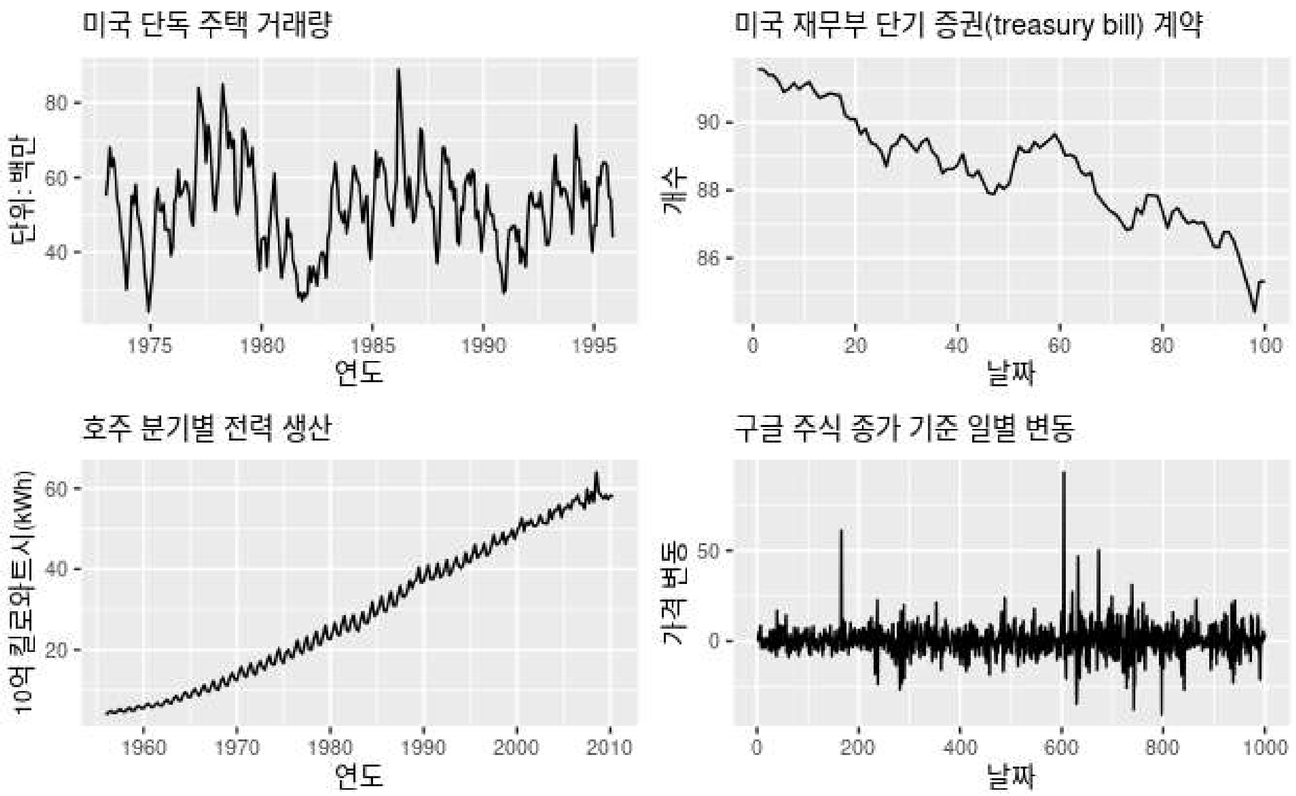
변수 정의

Features	row_id	56045_2021-11-01 | cfips + first_day_of_month
	cfips	56045 | 카운티 식별 번호, 앞 두자리는 주, 뒤 세자리는 카운티
	county_name	CountyWyomin | 주(state)의 하위 행정구역 단위 : 3135개
	state_name	Weston | 미국의 행정구역 단위 : 50개
	first_day_of_month	2021-11-01 | train의 경우 2021-11-01 부터 2022-10-01 까지 존재
	active	98.0 | 카운티에 존재하는 소기업 실제 빈도
Target	**microbusiness_density**	1.760374 | 소기업 밀도(타겟 변수), 18세 이상 인구 100명당 소기업 ** 계산에 사용된 인구 정보는 2년 전 집계

시계열 데이터의 특성

시계열 변동 요인

- 추세 (trend)
- 계절성 (seasonality)
- 주기성 (cycle)



비용함수 SMAPE (Symmetric Mean Absolute Percentage Error)

$$SMAPE = \frac{100}{n} \times \sum_{i=1}^{n} \frac{\left|Y_i - \widehat{Y}_i\right|}{\left(|Y_i| + |\widehat{Y}_i|\right)/2} \qquad MAPE = \frac{100}{n} \times \sum_{i=1}^{n} \left|\frac{Y_i - \widehat{Y}_i}{Y_i}\right|$$

회귀 모델의 전체 예측에 대한 평가 지표로 사용되는 비용 함수 → 낮을수록 성능 우수

😎 장점

➕ MAPE의 한계 : 실제 값이 0이면 계산 불가 → (완화) SMAPE는 실제 값과 예측 값이 동시에 0이면 계산 불가(단, 자명하게 0으로 정의 가능)

➕ 공역이 [0,200] → 200으로 나눠서 확률 해석 가능

➕ 데이터포인트 단위의 손실(Absolute Error)에 대한 합리적인 가중 평균을 사용하는 효과

ex. 실제 무게가 각각 10000, 10인 사물 A, B에 대하여 예측한 무게가 9999, 9인 경우

	y_true	y_pred	AE	APE
A	10000	9999	1	0.01
B	10	9	1	10.00

🤔 단점

➖ 실제 값 또는 예측 값 중 하나만 0인 경우 자동으로 손실의 최댓값(200)을 반환

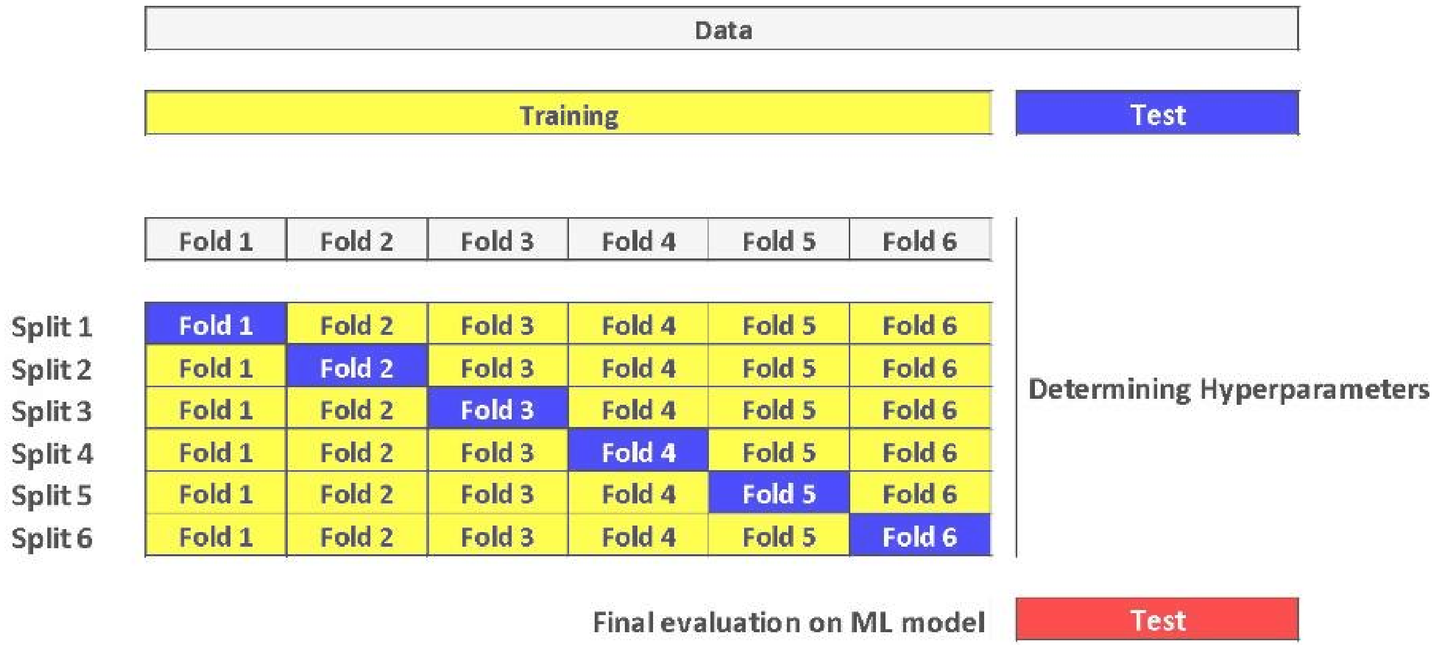
➖ 실제 값과 예측 값의 차이가 같을 때 대소 관계에 따라 손실이 다름

PART 2.
분석 방법 1

01 교차 검증 방법 구축
02 전처리

일반적인 K-Fold 교차 검증 방법은...

[ex. 6-fold 교차 검증 ⇒ 하이퍼파라미터 결정]



⚠️ 훈련 폴드와 검증 폴드의 분할에 순서 관계가 없음

한편 시계열 데이터는...

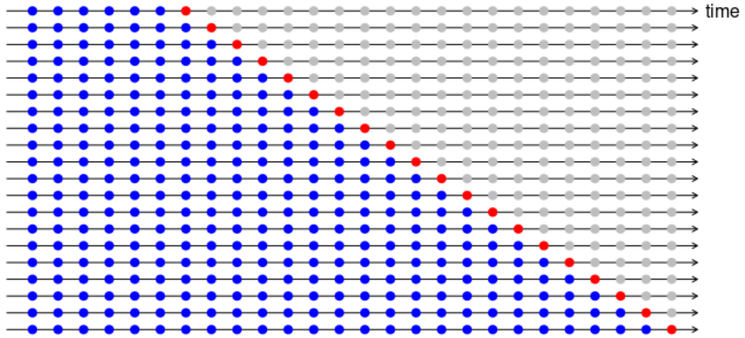
시계열 데이터 특성상

⛓️ 전후 데이터 간 상관성(auto-correlation) 존재

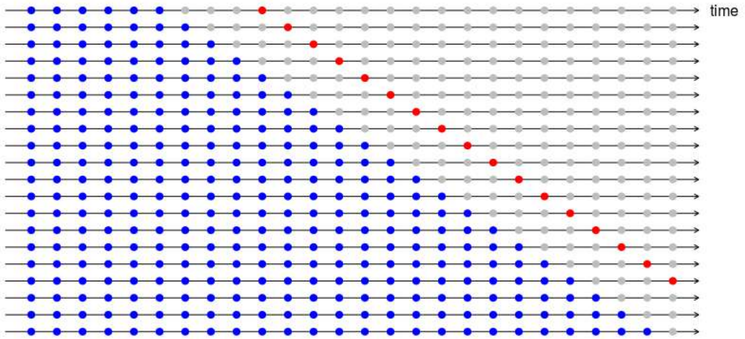
⟶ ⚠️ **순서 관계를 보존하는 것이 중요**

✔ **훈련 폴드는 항상 검증 폴드의 과거 시기를 할당**

✔ **그렇지 않을 경우 data leakage**



한 단계(1-step) 미래를 예측하는 시계열 교차 검증



네 단계(4-step) 미래를 예측하는 시계열 교차 검증

train, test를 한 데이터프레임에 합침

- 시계열 데이터 분석 과정에서 시차(lag)에 의해 파생되는 특성을 예측에 사용
  → 훈련 세트와 테스트 세트의 출처가 표시된 concat 필요

	row_id	cfips	county	state	first_day_of_month	microbusiness_density	active	istest
147329	56045_2022-03-01	56045	Weston County	Wyoming	2022-03-01	1.767542	99.0	0
147330	56045_2022-04-01	56045	Weston County	Wyoming	2022-04-01	1.767542	99.0	0
147331	56045_2022-05-01	56045	Weston County	Wyoming	2022-05-01	1.803249	101.0	0
⋮								
147342	56045_2023-04-01	56045	Weston County	Wyoming	2023-04-01	NaN	NaN	1
147343	56045_2023-05-01	56045	Weston County	Wyoming	2023-05-01	NaN	NaN	1
147344	56045_2023-06-01	56045	Weston County	Wyoming	2023-06-01	NaN	NaN	1

훈련 세트

테스트 세트

```
train['istest'] = 0
test['istest'] = 1
raw = pd.concat((train, test)).sort_values(['cfips','row_id']).reset_index(drop=True)
```

결측치 처리

- 훈련 세트와 테스트 세트의 concat 과정에서 스키마 불일치에 의해 결측치 발생
  → 결측된 스키마의 식별자 정보인 cfips를 기준으로 그룹화 후 결측치를 앞쪽 정보로 입력
- 💡 앞서 정렬된 row_id에 의해 cfips 그룹 내에서 시간 순으로 정렬되므로 가능

	row_id	cfips	county	state	first_day_of_month	microbusiness_density	active	istest
147329	56045_2022-03-01	56045	Weston County	Wyoming	2022-03-01	1.767542	99.0	0
147330	56045_2022-04-01	56045	Weston County	Wyoming	2022-04-01	1.767542	99.0	0
147331	56045_2022-05-01	56045	Weston County	Wyoming	2022-05-01	1.803249	101.0	0

훈련 세트

ffill()

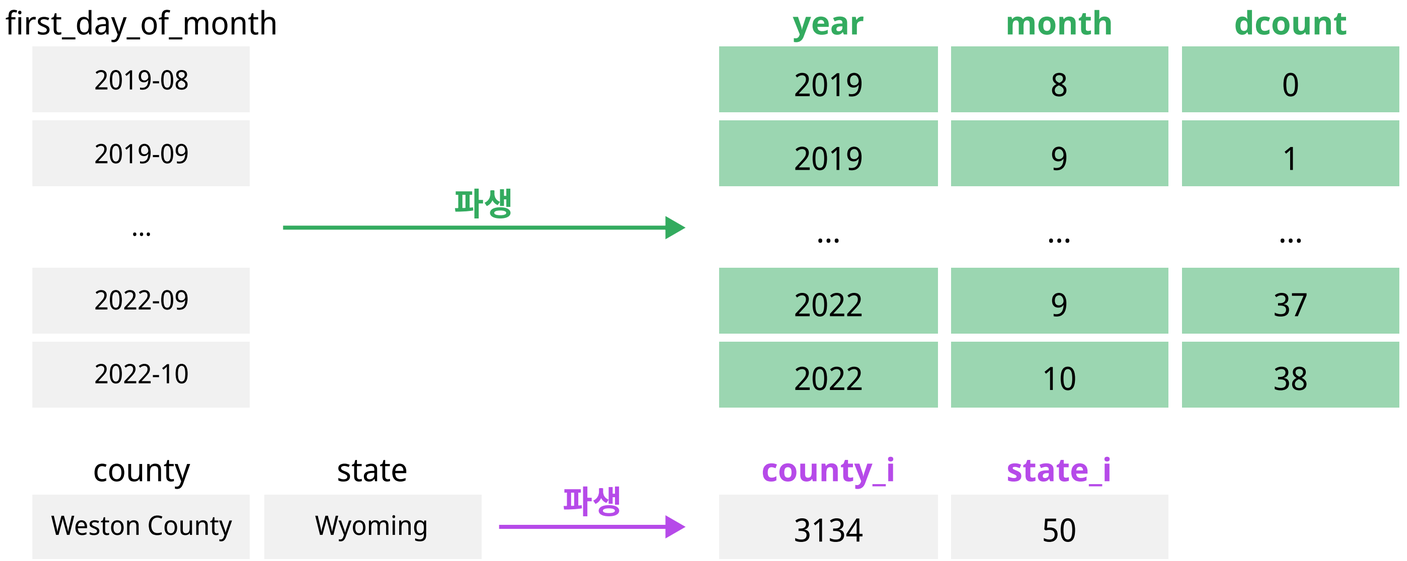
147342	56045_2023-04-01	56045	Weston County	Wyoming	2023-04-01	NaN	NaN	1
147343	56045_2023-05-01	56045	Weston County	Wyoming	2023-05-01	NaN	NaN	1
147344	56045_2023-06-01	56045	Weston County	Wyoming	2023-06-01	NaN	NaN	1

테스트 세트

```
raw['county'] = raw.groupby('cfips')['county'].ffill()
raw['state'] = raw.groupby('cfips')['state'].ffill()
```

기본적인 날짜 & 지역 관련 파생변수 생성

- EDA 및 분석의 전반에 걸쳐 필요한 인덱스 역할




```
raw['first_day_of_month'] = pd.to_datetime(raw['first_day_of_month'])
raw['year'] = raw['first_day_of_month'].dt.year
raw['month'] = raw['first_day_of_month'].dt.month
raw['dcount'] = raw.groupby(['cfips'])['row_id'].cumcount()
raw['county_i'] = (raw['county'] + raw['state']).factorize()[0]
raw['state_i'] = raw['state'].factorize()[0]
```

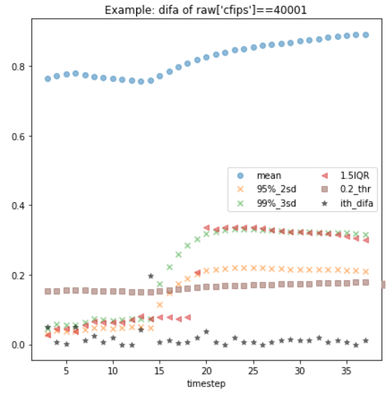
전반적인 추세가 변하지 않는 선에서 이상치 처리

- 이상치 탐지 방법 : 이번달 밀도와 지난달 밀도 차이가 특정 기준보다 크면 이상치로 판단
- 제한 조건 : 전반적인 추세(그래프의 모양)를 최대한 보존
- 상위권 솔루션의 기준을 사용했으며 남은 대회 기간 동안 다양한 방법을 시도해볼 계획

[이상치 탐지 방법]

- IQR Rule-based Anomaly Detection
- STL 분해
- 분류 및 회귀 트리 (CART)
- 클러스터링 기반 이상 탐지
- 오토 인코더
- 임의로 데이터의 분포를 활용한 기준 사용

[기준 = 지금까지의 밀도 평균의 20%]

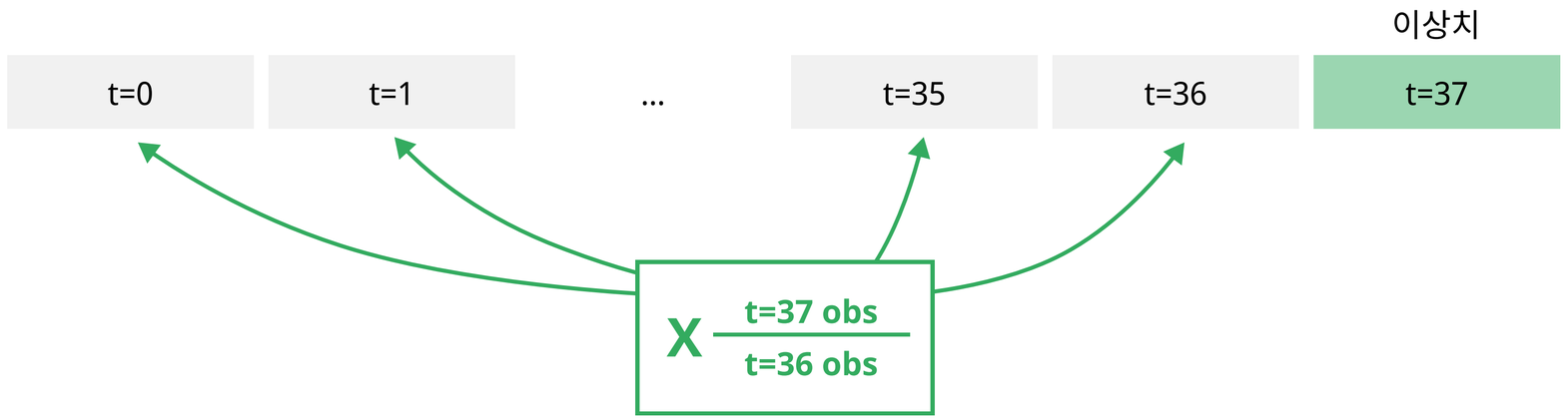



```
for i in range(37, 2, -1):
    thr = 0.20 * np.mean(var[:i])
    difa = abs(var[i]-var[i-1])
```

IQR, 신뢰구간 등 다른 방법에 비해
기준선이 일관되며 후하게 채점하는 편

이상치가 탐지되면 과거 관측값이 현재 추세를 반영하도록 보정

- t=i에서 이상치가 탐지되면 그 이전의 관측값을 i번째 값과 같은 수준으로 끌어올림
- 최근 관측값과 추세가 중요하기 때문에 이상치가 아닌 과거 관측값을 변경해 추세를 인위적으로 바꿔줌



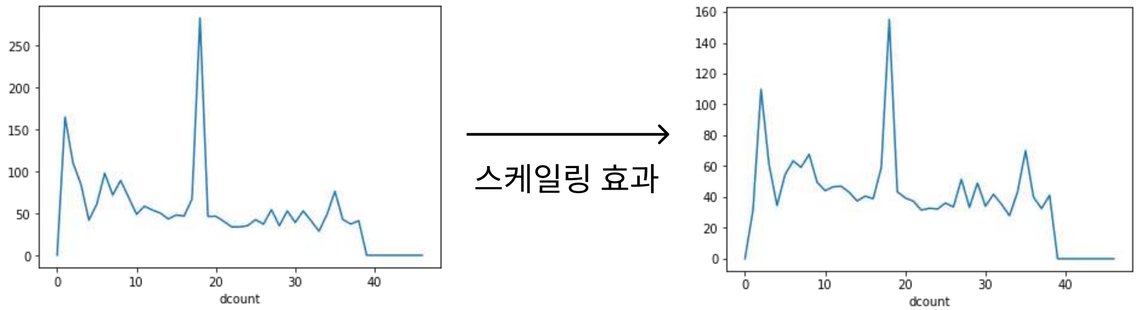

```
if (difa>=thr):
    var[:i] *= (var[i]/var[i-1])
```

*관측값 = microbusiness_density

이상치가 탐지되면 과거 관측값이 현재 추세를 반영하도록 보정

- 분산을 나름 균일하게 조정하는 스케일링 효과, 데이터의 일관성 높임
- 시계열 데이터에서 중요한 정상성 가정을 일정 부분 만족시키기 위함
- 이상치(outliers)이면서 “영향력 있는 관측값(influential points)”을 반영

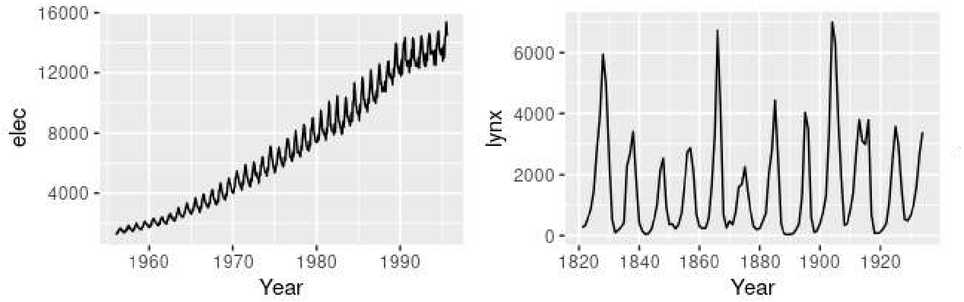
[지난달 대비 이번달 타깃값]



보정 전 범위 (0, 250)

보정 후 범위 (0, 160)

[시계열 데이터의 정상성]



정상성 X

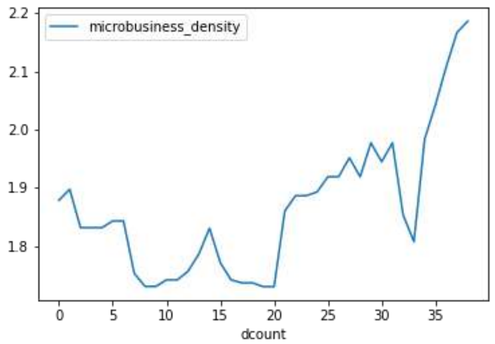
정상성 O

*영향력 있는 관측값: 회귀 모형의 추정된 계수에 큰 영향을 주는 관측값

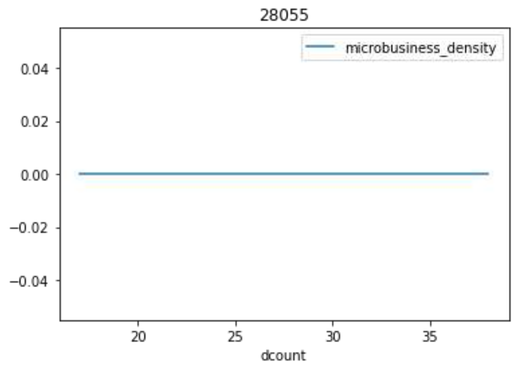
데이터 입력/수집 오류로 보이는 county 삭제

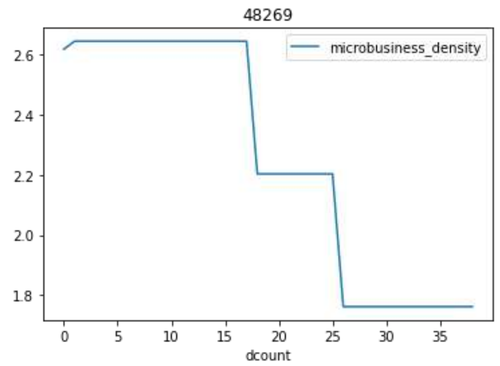
- mbd가 전부 NaN/0이거나 특정 값이 계속 반복되는 계단형인 경우 오류로 판단, 타깃 값에 0 대입

[일반적인 분포]



[삭제한 county의 분포]






```
raw.loc[raw['cfips']==28055, 'target'] = 0.0
raw.loc[raw['cfips']==48269, 'target'] = 0.0
```

평가지표의 특성을 반영해 예측 대상을 '변화율'로 변경

- SMAPE는 상대적인 지표이므로 절대적인 값인 '밀도'보다 상대적인 값인 '밀도의 변화율' 예측이 평가지표와 부합

[변경 전] 이번달 microbusiness_density

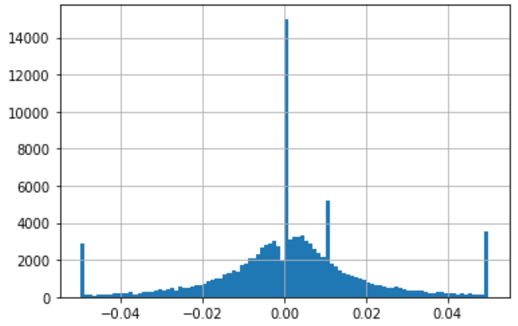
▼

[변경 후]

$$\frac{\text{다음달 microbusiness\_density} - \text{이번달 microbusiness\_density}}{\text{이번달 microbusiness\_density}} \times 100$$

= mbd 변화율

[타깃의 분포]




```
raw['target'] = raw.groupby('cfips')['microbusiness_density'].shift(-1) / raw['microbusiness_density'] - 1
```

시계열 예측을 위해 시차값(lagged value) 생성

- 시차 변수, 지연 변수인 lag 1, lag 2, lag 3 생성
- 데이터 특성상 '밀도의 변화율'을 예측할 때 과거의 '소기업 밀도'와 '소기업 개수'가 영향을 주기 때문

date	county id	(TARGET) **mbd 변화율**	**lag 1**	**lag 2**	**lag 3**
2019-08	A	a			
2019-09	A	b	a		
2019-10	A	c	b	a	
2019-11	A	d	c	b	a
2019-12	A	e	d	c	b
2020-01	A	f	e	d	c

```
raw[f'mbd_lag_{lag}'] = raw.groupby('cfips')[target].shift(lag)
raw[f'act_lag_{lag}'] = raw.groupby('cfips')[target_act].diff(lag)
```

시계열 예측을 위해 이동평균 변수 생성

- 일반적으로 시계열 데이터를 다룰 때 추세-주기를 측정하기 위해 이동평균 사용
- 이동평균: k 기간 안의 모든 관측값의 평균
- 평균이 데이터의 무작위성을 줄이고 매끄러운 추세-주기 성분만 남기는 효과

date	county id	mbd lag 1	2-이동평균	4-이동평균	6-이동평균
2019-08	A	a			
2019-09	A	b	(a+b) / 2		
2019-10	A	c	(b+c) / 2	a	
2019-11	A	d	c	b	a
2019-12	A	e	d	c	b
2020-01	A	f	e	d	c

window
“이동”하며
“평균”

```
lag = 1
for window in [2, 4, 6]:
    raw[f'mbd_rollmea{window}_{lag}'] = raw.groupby('cfips')[f'mbd_lag_{lag}'].transform(lambda s: s.rolling(window, min_periods=1).mean())
    feats.append(f'mbd_rollmea{window}_{lag}')
```

PART 3.
분석 방법 2

01 모델 설계
02 하이퍼 파라미터 튜닝
03 모델 학습 및 전략

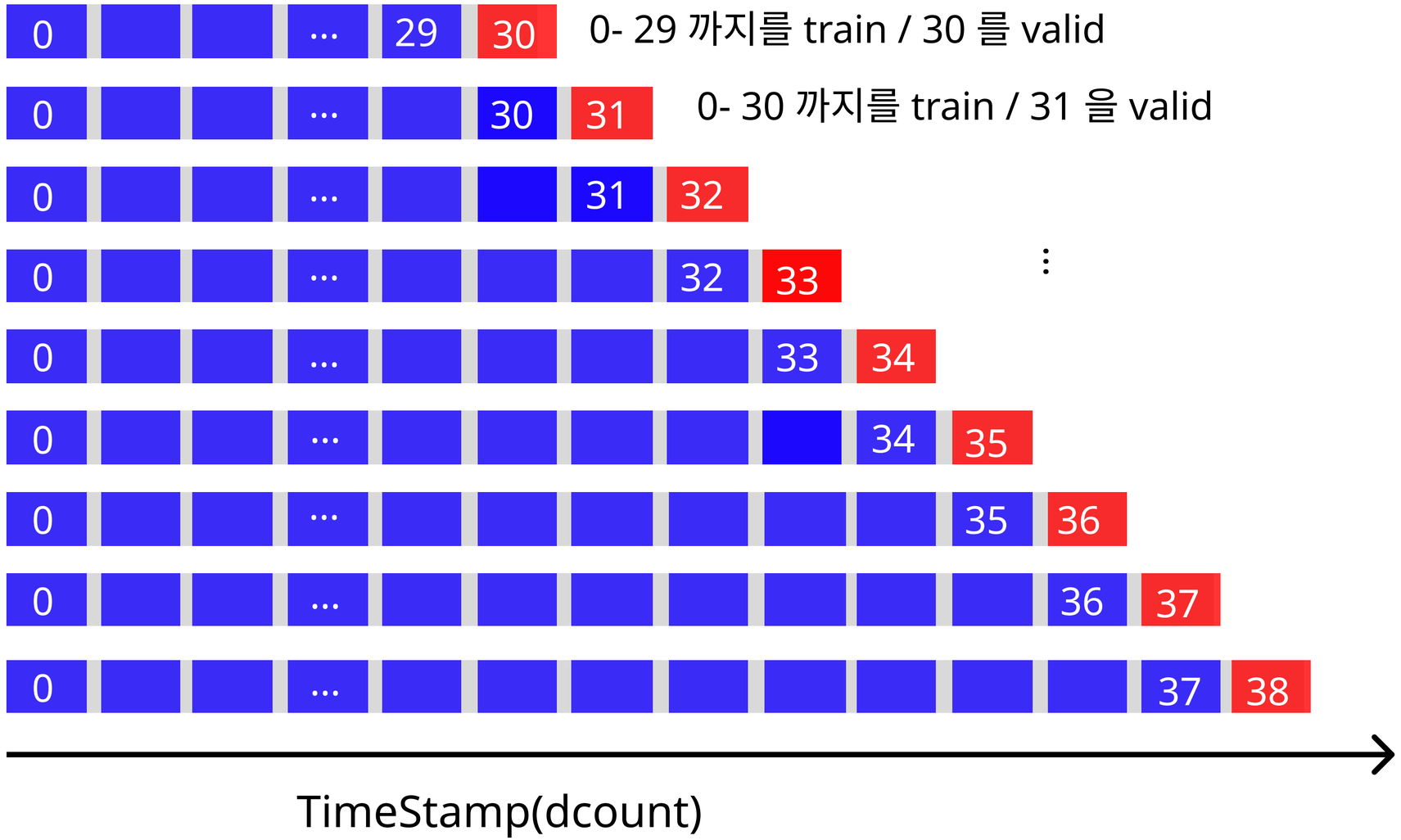
TimeSeriesSplit

항상 훈련 데이터가 검증 데이터보다 앞선 시간 값을
가지도록 교차검증 데이터 세트를 만든다.

for TS in range(29,38) :

학습데이터

예측값



Blacklist 설계 배경

문제 상황

1. **적은 양의 데이터**에 비해 **많은 양을 예측**해야 한다.

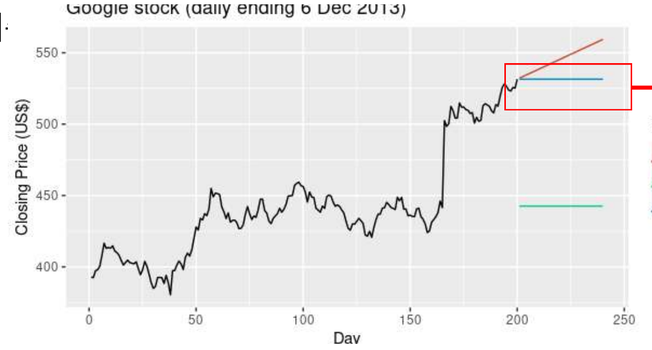
38 step → **모델** → **8 step**

2. 많은 지역을 한꺼번에 예측해야 하므로, **모든 지역의 시계열적 특성을 고려할 수 없다.**
   따라서, 지역에 따라서 모델의 성능이 다 다를 수 있다.

고안해 낸 해결 방안 | BlackList

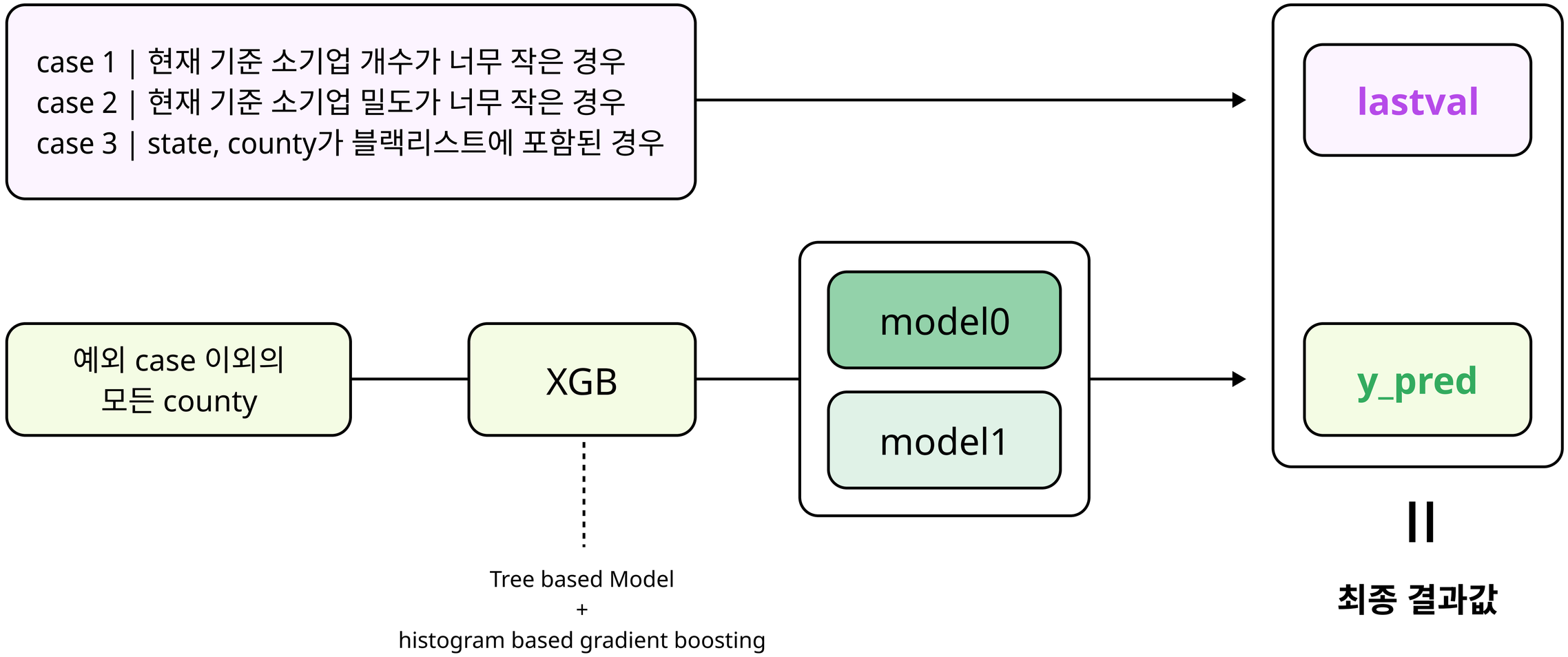
- 회귀의 가장 베이스가 되는 예측은 평균으로 예측하는 것인데, 시계열 데이터의 특성상 오래된 데이터까지 고려한 평균은 비효율적, **가장 최근의 값으로 예측하는 방법인, Naive Forecasting을 할 지역들을 선별해야한다.** → **Blacklist**

예시) 구글 주가 예



Naive Forecasting

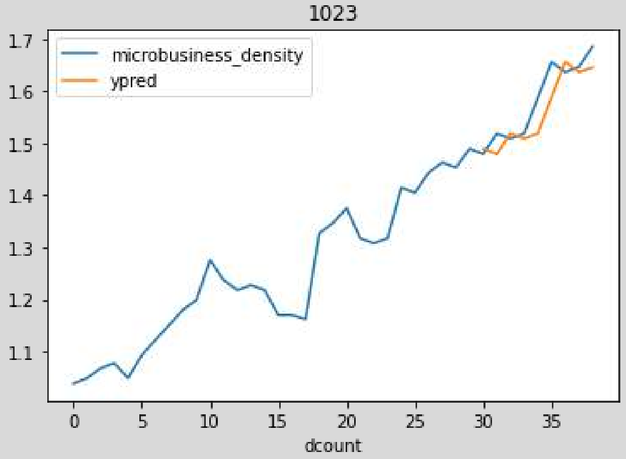
Blacklist 생성을 위한 모델링

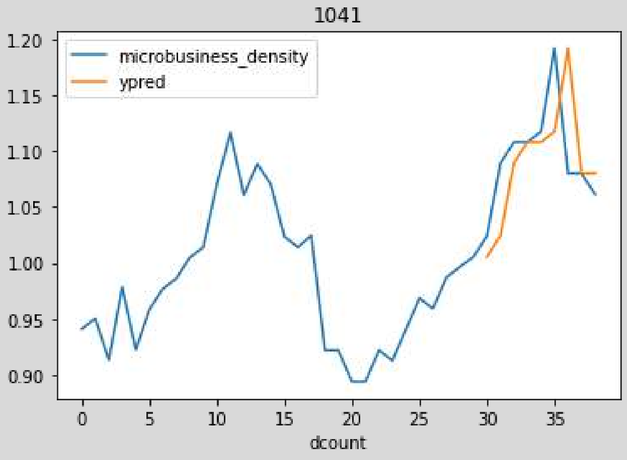


BlackList

만약 모델의 예측값이 최근값을 그대로 사용하는 것보다 못하다면?

blacklist county plot





xgb 모델을 통한 예측값 — xgb_pred

바로 이전 dcount의 밀도 — lastval

xgb_pred vsmape — error

lastval vsmape — error_last

error > error_last

모델 예측값에 대한 error 값이 바로 이전 밀도값에 대한 error 값보다 크다면,
모델 예측값이 아닌 바로 이전 밀도값을 그대로 적용하는 것이 적절하다.

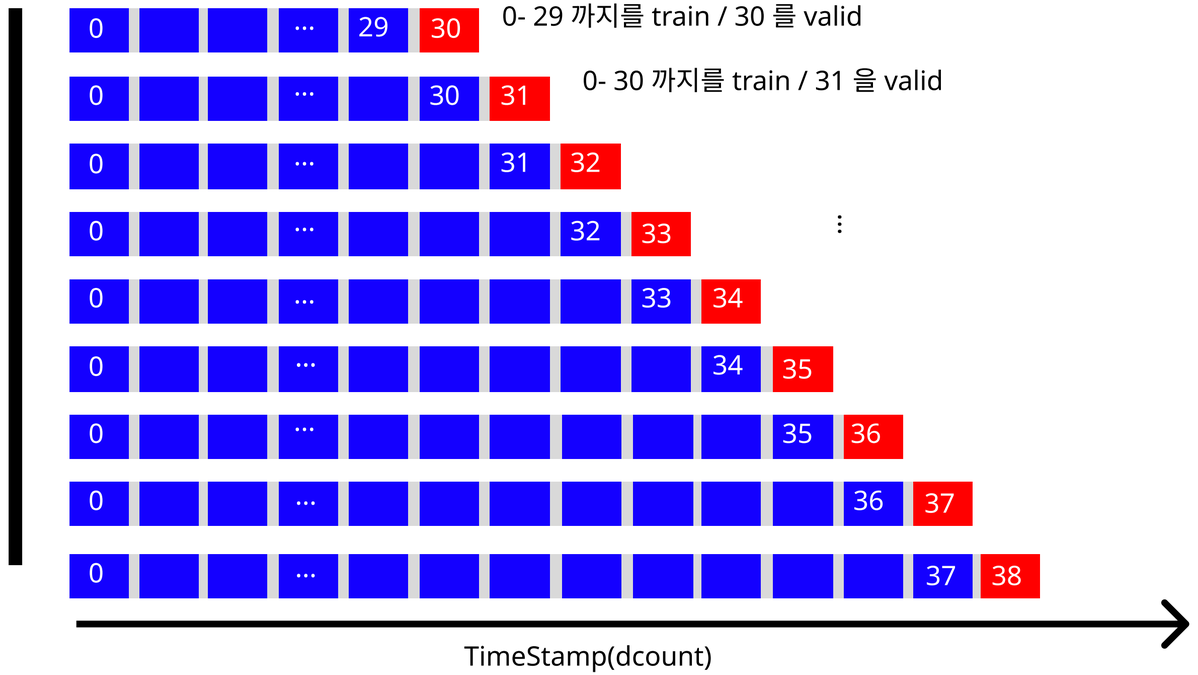
TimeSeriesSplit 를 통해 BlackList 생성하기

[1] valid set 이 30 - 38 까지 총 9 fold 에 대하여 반복

[2] 10 fold 반복하며 y_pred 와 lastval 도출

xgb_pred　　lastval

총 9 fold



[3] 마지막 fold 에서 error 값과 error_last 값을 비교

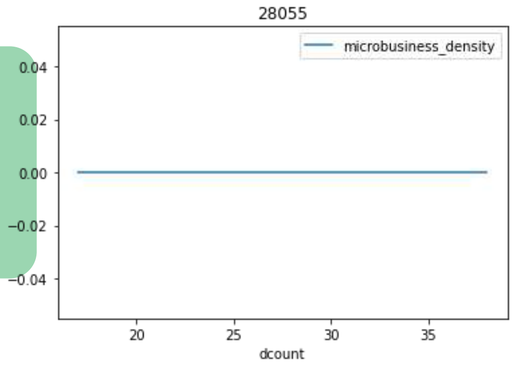
error　>　error_last

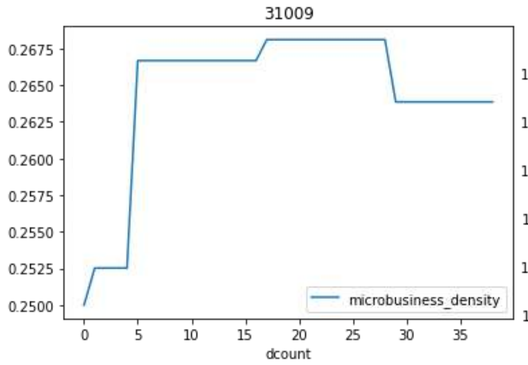
[4] error > error_last 인 경우 blacklist 에 append

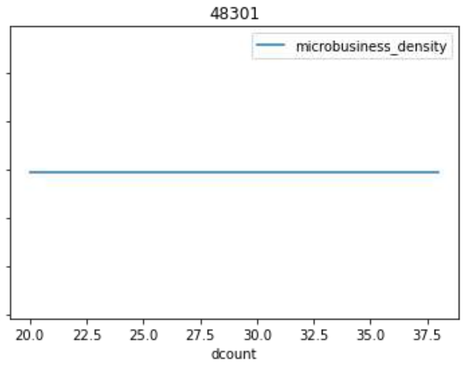
총 10번의 검증을 통해서 평균적으로 모델을 통한 error가 더 큰 지역들은 blacklist에 추가 후에 예측값도 최근값으로 대체

이외에도 lastval 을 적용하는 case

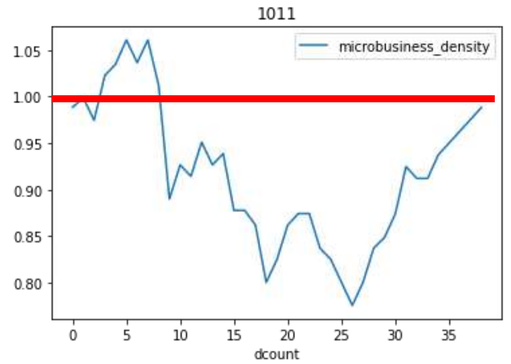
1) lastactive <= ACT_THR (1.8)
실제 소기업의 수가 일정 이하인 경우

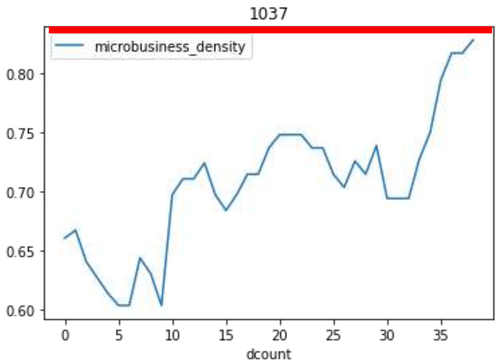


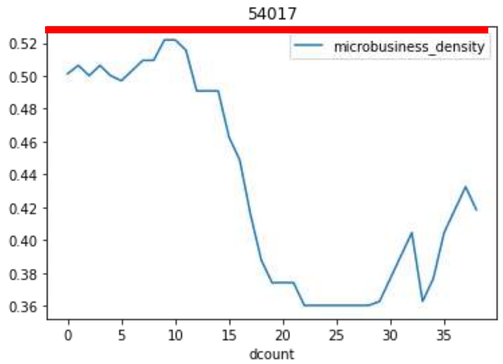




2) lastval <= ABS_THR (1)
소기업의 밀도가 일정 이하인 경우





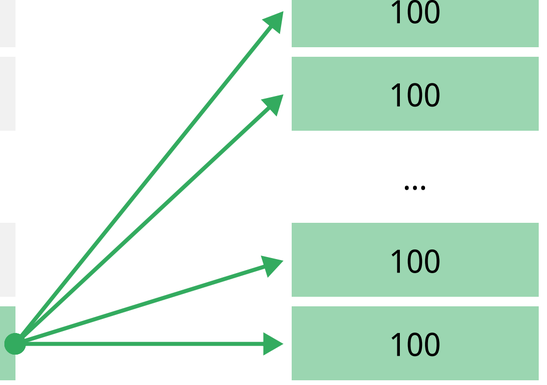


- 가장 최신의 소기업 빈도수와 밀도가 해당 임계치보다 낮은 경우 모델을 통한 예측의 정확도를 확신할 수 없다고 판단
- xgb_pred 대신에 lastval 을 적용한다

훈련 시 간편한 인덱싱을 위해 임시변수 생성

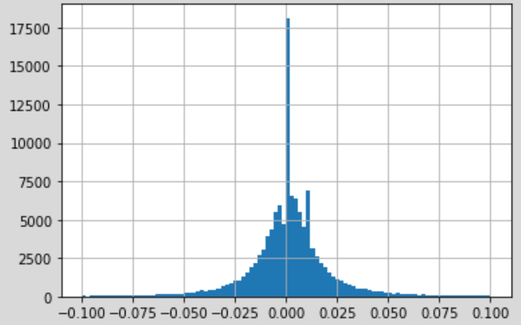
- 시계열 데이터의 특성 상, 피쳐들 간에 시간 상 순서가 있고, 가장 최근의 특성 값을 적절히 탐색할 필요가 있다
- 따라서 타깃과 연관된 변수에 한하여 주어진 데이터 중 가장 최근 값을 저장하는 변수 생성했다

date	county id	active	last active
2019-08	A	102	100
2019-09	A	99	100
...	...	...	...
2022-09	A	104	100
2022-10	A	100	100

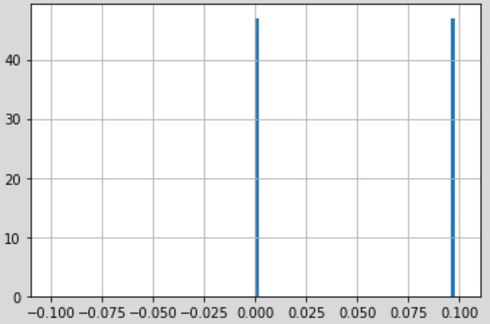


- 같은 county라면 같은 값을 저장
- **last active**: 우리가 알 수 있는 가장 최근 시기 active 값 (t=38, 2022-10 소기업 개수)
- **last target**: 첫번째 train 마지막 시기 target 값 (t=28, 2021-12 소기업 밀도 변화율)

- 기존 target 값 분포



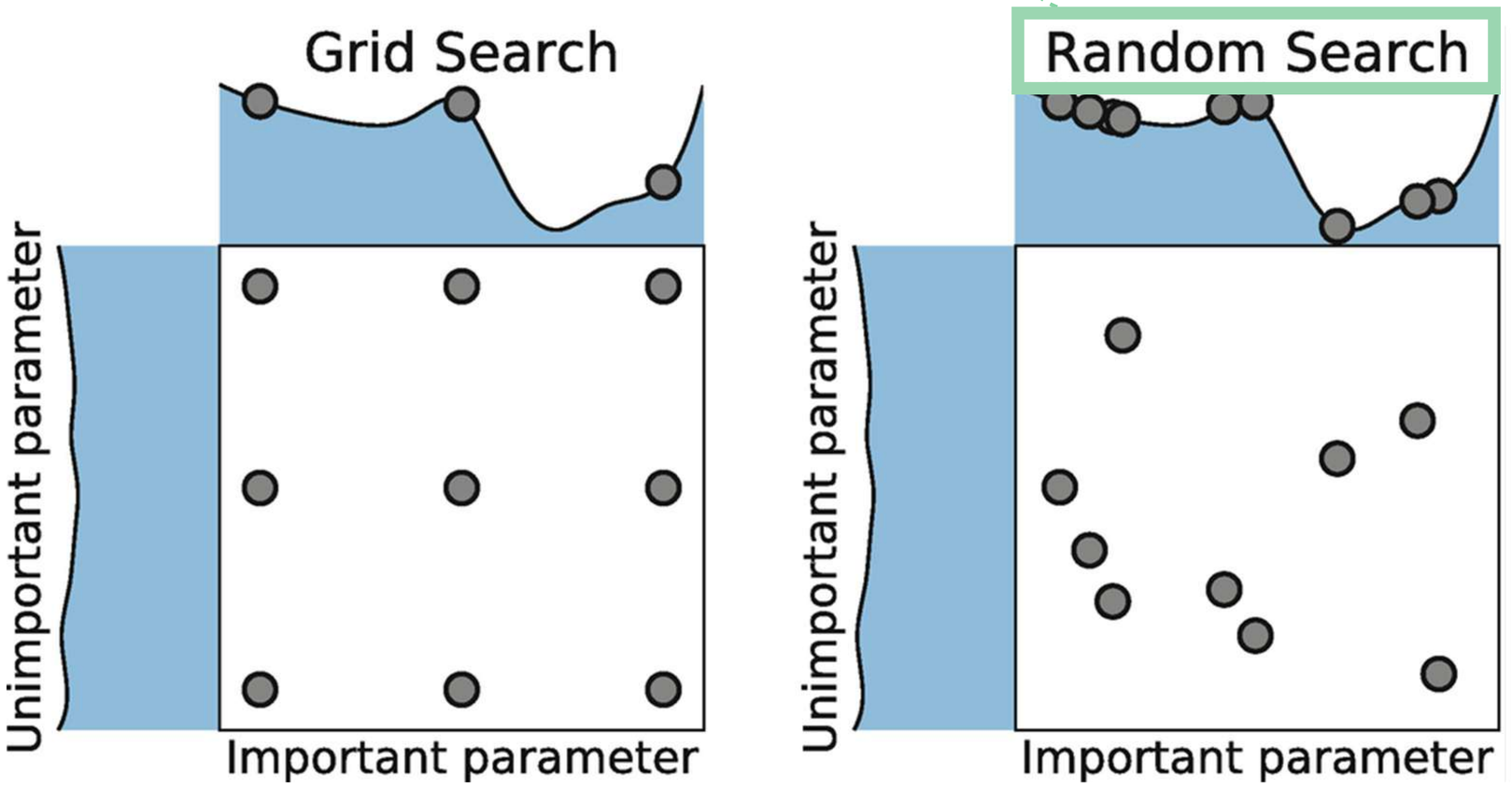
- 새로 생성한 lasttarget 값 분포



- t=28 값만을 저장했으므로 불연속적이다

하이퍼 파라미터 튜닝

하이퍼 파라미터 조합에 대한 시간 자원의 효율성을 위해
RandomSearchCV 적용



하이퍼 파라미터 튜닝 결과

```
from sklearn.model_selection import RandomizedSearchCV
import time
for TS in range(29, 38):
    print(TS)
    model = xgb.XGBRegressor(tree_method='gpu_hist', gpu_id=0,

        objective='reg:absoluteerror',
        eval_metric='mae',
        early_stopping_rounds=70)

    param_grid = {
        'eta': [0.05, 0.1, 0.3],
        'max_depth': range(3,10),
        'subsample': np.arange(0.3,1,0.1),
        'colsample_bytree': np.arange(0.1,1,0.1),
        'n_estimators' : np.arange(100,5000,100),
        'learning_rate': np.arange(0.001,0.01,0.001)
                    }

    grid_xgb_cv = RandomizedSearchCV(model, return_train_score=True,
                                param_distributions = param_grid,
                                verbose=0,
                                scoring = "neg_mean_absolute_percentage_error",
                                n_jobs=-1)

    train_indices = (raw.istest==0) & (raw.dcount  < TS) & (raw.dcount >= 1) & (raw.lastactive>ACT_THR)  & (raw.lasttarget>ABS_THR)
    valid_indices = (raw.istest==0) & (raw.dcount == TS)

    grid_xgb_cv.fit(
        raw.loc[train_indices, features],
        raw.loc[train_indices, 'target'].clip(-0.0043, 0.0045),
        eval_set=[(raw.loc[valid_indices, features], raw.loc[valid_indices, 'target'])],
        verbose=0
    )

    print('최적의 매개변수 조합: ', grid_xgb_cv.best_params_)
    print('최고의 교차 검증 점수: ', grid_xgb_cv.best_score_)
```

RandomizedSearchCV 사용

- scoring = ‘neg_mean_absolute_percentage_error’

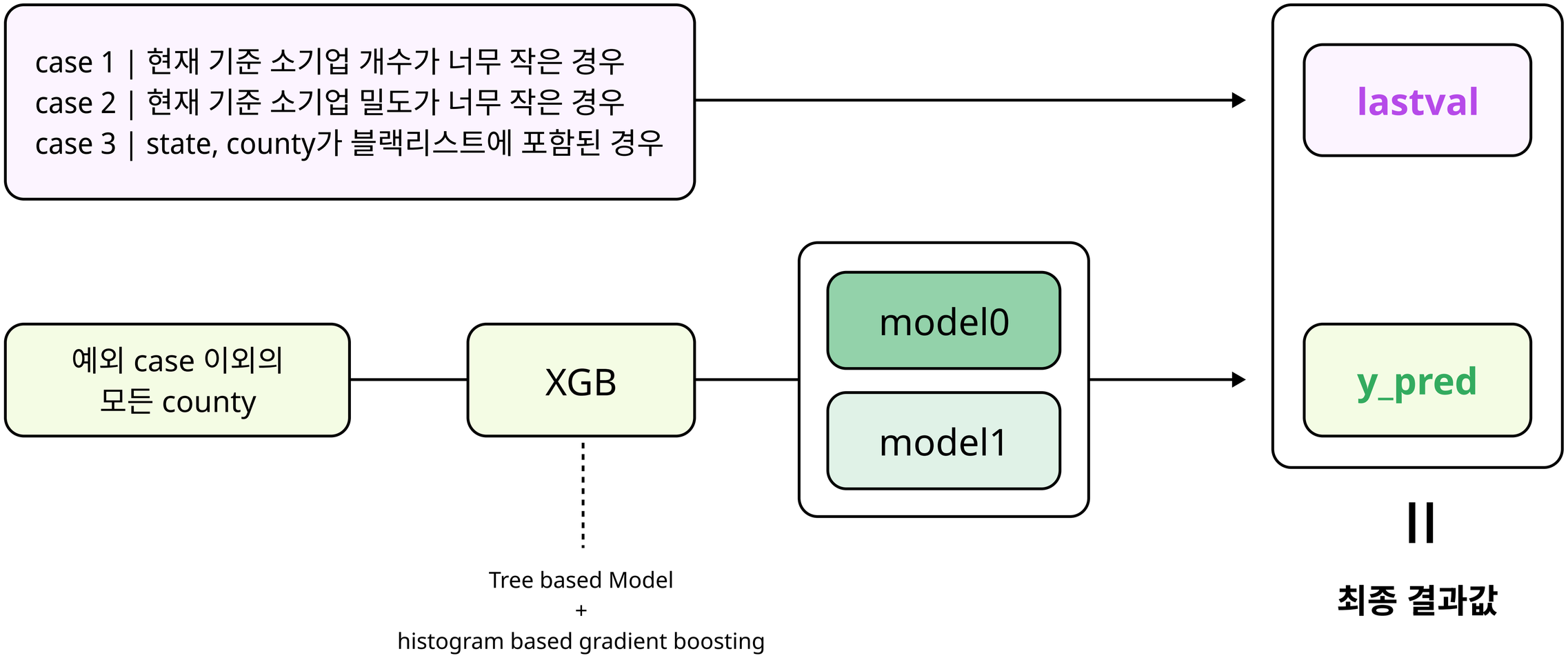
  모델 성능 평가 지표로 smape 와 동일한 flow를 가지는 지표 사용

교차검증 결과 TS=37에서의 최적 하이퍼 파라미터 도출

- subsample : 0.4
- n_estimators : 2200
- max_depth : 6
- learning_rate : 0.001
- eta : 0.1
- colsample_bytree : 0.2

Last Value SMAPE: 1.101119095956366
XGB SMAPE: 1.0769803704614893

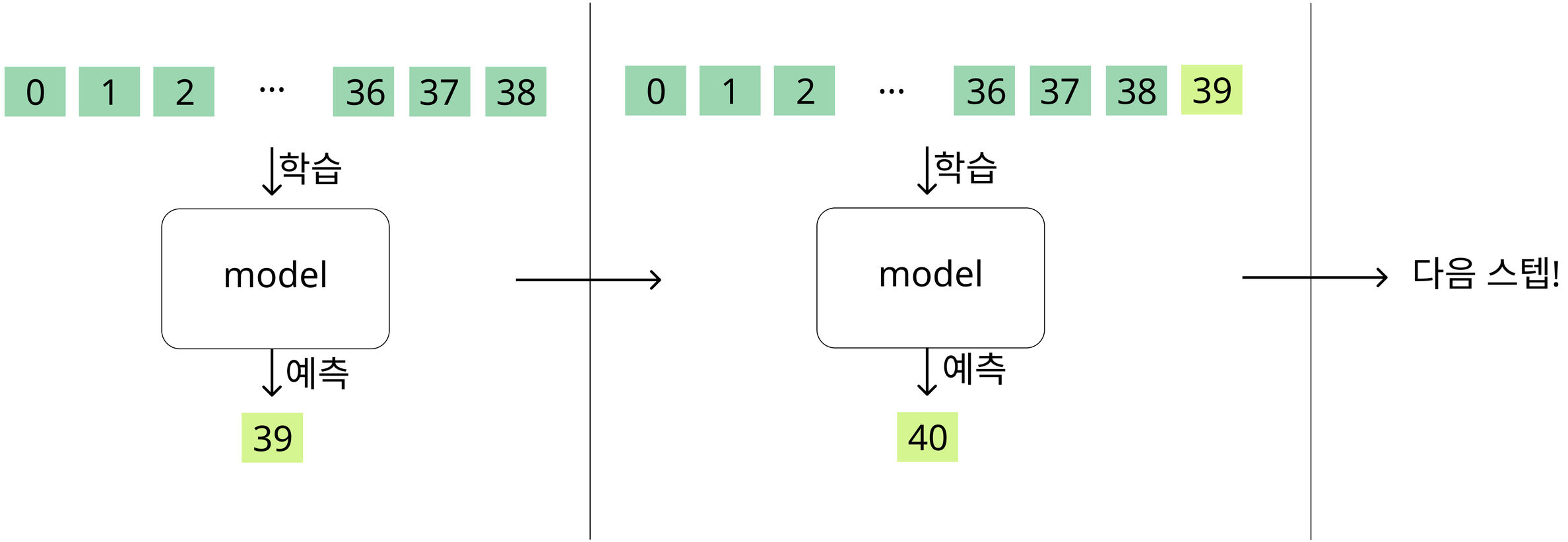
모델 학습 및 결과 값 도출


case 1 | 현재 기준 소기업 개수가 너무 작은 경우
case 2 | 현재 기준 소기업 밀도가 너무 작은 경우
case 3 | state, county가 블랙리스트에 포함된 경우
lastval
예외 case 이외의
모든 county
XGB
model0
model1
y_pred
=
최종 결과값
Tree based Model
+
histogram based gradient boosting

모델 전략

- **Direct-recursive hyprid model(**

  주어진 time step이 총 39개인데 비해 8개의 step을 예측해야함
  한꺼번에 예측하는 모델은 성능이 떨어질 위험이 있으므로, 다음의 모델을 사용하기로 결정



PART 4.
결과 해석

01 결과 해석
02 앞으로의 계획

결과 도출 flow

데이터 클렌징

결측치 처리 | 이상치 탐지 및 보정

파생변수 생성

단순 | 시차 변수 | 이동평균변수

→

모델 활용 case 분류

blacklist | xgb_pred vs lastval

→

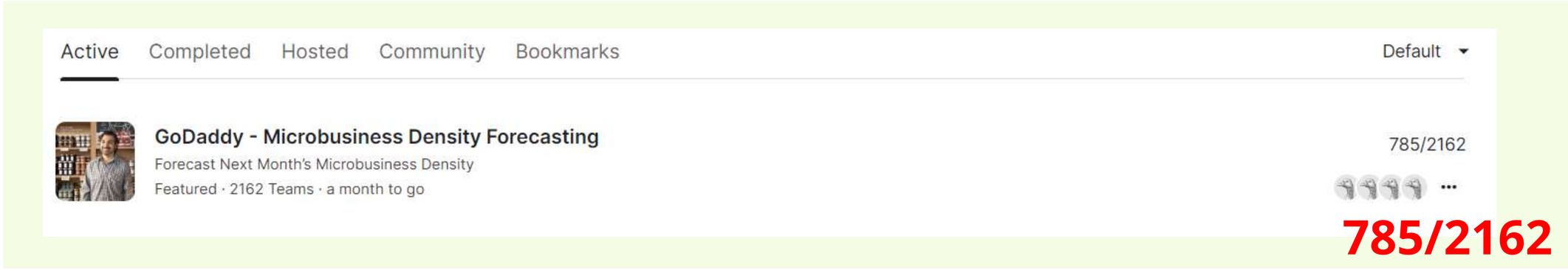
모델 학습 [1]

XGB+랜덤서치 | 하이퍼파라미터튜닝

모델 학습 [2]

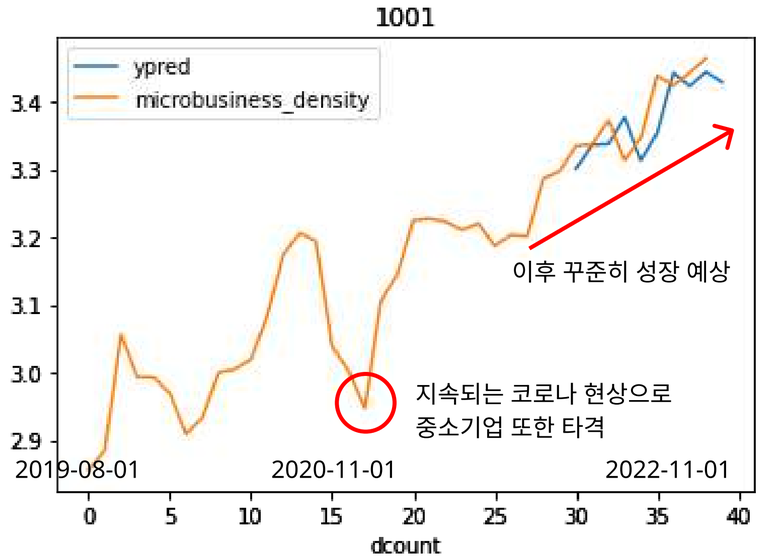
Direct- recursive hyprid 전략

현재 랭킹

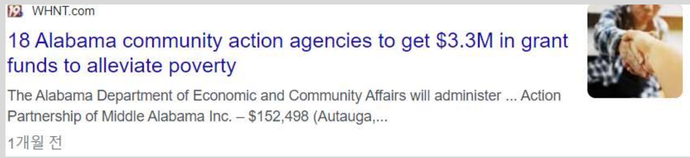


결과 해석 (증가 추세 county)

Alabama state / Autauga County



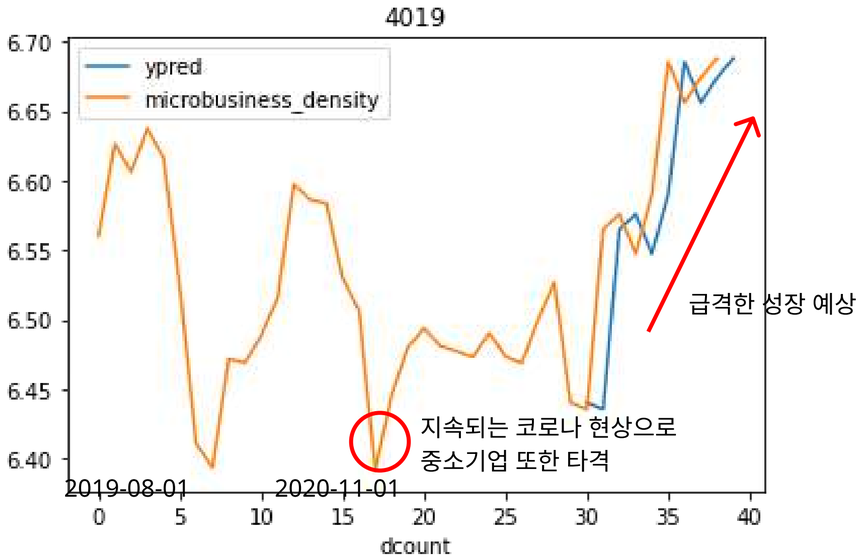
- 해당 county 경제 관련 뉴스



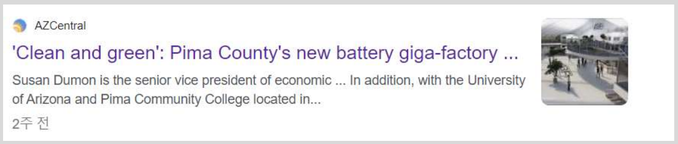
- 인구수 (2021) | 59,095
- 인구밀도 | 97/sq mi (38/km2)

- 최근 큰 사업 보조금을 받아 성장 가능성 O
- 인구수와 인구밀도, 개인 소득 또한 alabama state 내에서 상위권에 속함
- 앞으로 중소기업의 성장 가능성 매우 크다

Arizona state / Pima County



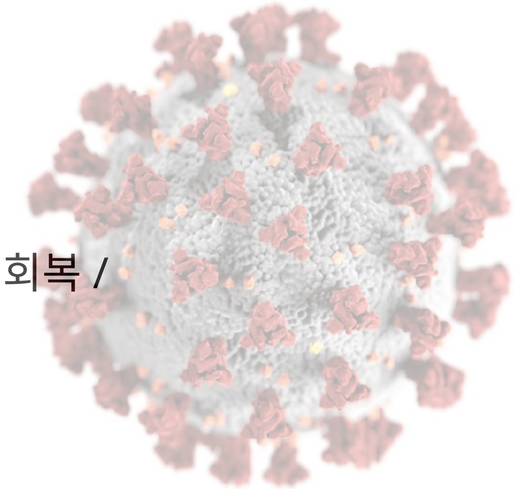
- 해당 county 경제 관련 뉴스



- 인구수 (2021) | 1,052,030
- 인구밀도 | 110/sq mi (44/km2)

- 다양한 공장 및 산업이 활성화 되어 있는 county
- 인구수와 인구밀도, 개인 소득 또한 매우 높음
- 앞으로 중소기업의 성장 가능성 매우 크다

도출된 결과를 통한 전반적 해석

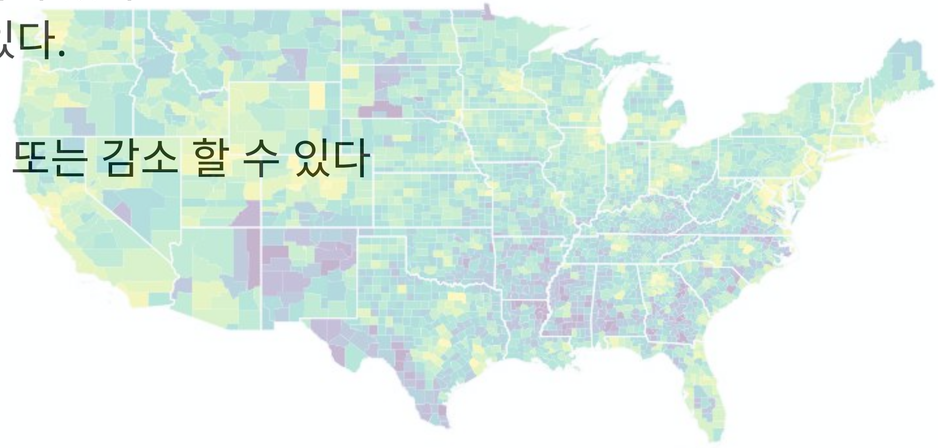


[1] 코로나로 인한 타격

- 상당수의 county 는 중소 기업 밀도가 코로나 이전의 추세를 회복 /
  오히려 상회하는 것으로 파악

[2] 타겟값에 영향을 미치는 다양한 요인

- 미국은 50개의 state, 3000개 이상의 county 로 구성되어 있다
  즉, state 별, county 별로 매우 다양한 특성을 가지고 있다.
- 사업보조금, 활성화된 산업, 인구수, 인구밀도 등
  다양한 요인들에 따라 중소기업 밀도가 영향을 받아 증가 또는 감소 할 수 있다



- **앞으로 시도해볼 것들**

외부 데이터 활용

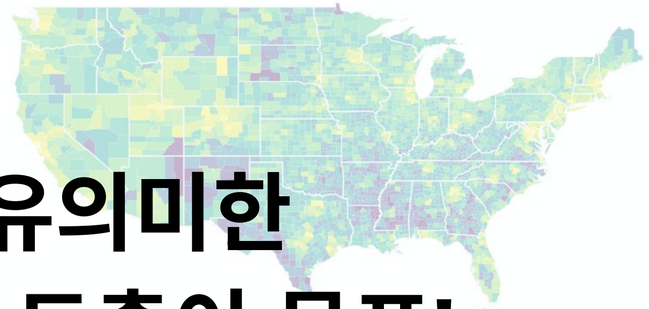
트리 기반 모델과 시계열 전용 모델 성능 비교

다양한 이상치 처리 방법 시도

블랙리스트 기준 설정

앙상블 시도

→



유의미한
성과 도출이 목표!

감사합니다