

## 1. 다양한 성능 실험

- train에 age\_gender\_info를 포함 / 단지 별 평균 / drop\_duplicate / baseline
  - 123.2816326531
- 나이대를 10으로 나누어서 위와 동일하게 적용
  - 116.8657142857
- standardScaler 적용하여 스케일 맞춰보기
  - 116.7528571429 (0.1 정도 낮아짐.)

## 2. 고안한 3가지 전처리 방안들의 조합 성능 검증

[1] Train에서 '공급유형'이 장기전세, 공공분양, 공공임대(5년)인 example을 Drop

[2] 각 단지코드 내에 '임대건물구분'

- '아파트'만 존재하면 1로,

- '상가'와 '아파트'가 모두 존재하면 0으로 매핑

[3] '자격유형'별 평균임대료 feature를 추가

구분	Neg mae
[1] 적용	-150.1562
[2] 적용	-150.4797
[3] 적용	-149.6817
[1]&[2] 적용	-149.6715
[1]&[3] 적용	-148.1634
[2]&[3] 적용	-150.0975
모두 적용	-149.8280

[베이스라인 모델 LB 성능]

모두 적용 (LB) -> 121.0857    /    [1]&[3] 적용 (LB) -> 122.0808

### 3. 'age\_gender\_info' 추가하기(ver 3(모두 적용 ver.)에 추가)

	1st	1gender	2nd	2gender	3rd	3gender
지역						
경상북도	50	0	40	0	60	0
경상남도	60	0	50	0	40	0

위와 같이 지역별로 상위 3개 그룹의 정보를 Train에 지역에 맞춰서 대입

[베이스라인 모델 LB 성능]

age\_gender\_info 추가 (LB) -> 199.2286

### 4. 더 나은 성능의 Model 찾기

[test 후보 및 각 성능(neg mae)]

- RandomForest (Baseline) : **-129.66053879429143**
- RandomForest (GridSearch) : **-129.66053879429143**
- XGBoost : **-134.52866928479915**
- Lightgbm : **-128.65809393573224**
- Ridge : **-130.7595200519923**
- Lasso : **-128.2108916930039**

[automl LB 성능]

- automl 110.9089242269

[lasso 모델 LB 성능]

- 전처리 ver 4 (age\_gender\_info)를 적용

Lasso\_with\_version\_4 (LB) -> 111.2101