실시간 따릉이 잔여대수 예측을 통한 사용자 불만 보완 프로젝트

#Kubernetes # Kubeflow # EK #ML #Scikit-Learn

김인규 X 김진세 X 박영민 X 양주화

CONTENTS

CONTENTS

따릉이 소개 및 현황 | 서울시민이 뽑은 정책 1위

따르릉 따르릉 비켜나세요~ 예측모델 나갑니다 따르르릉~

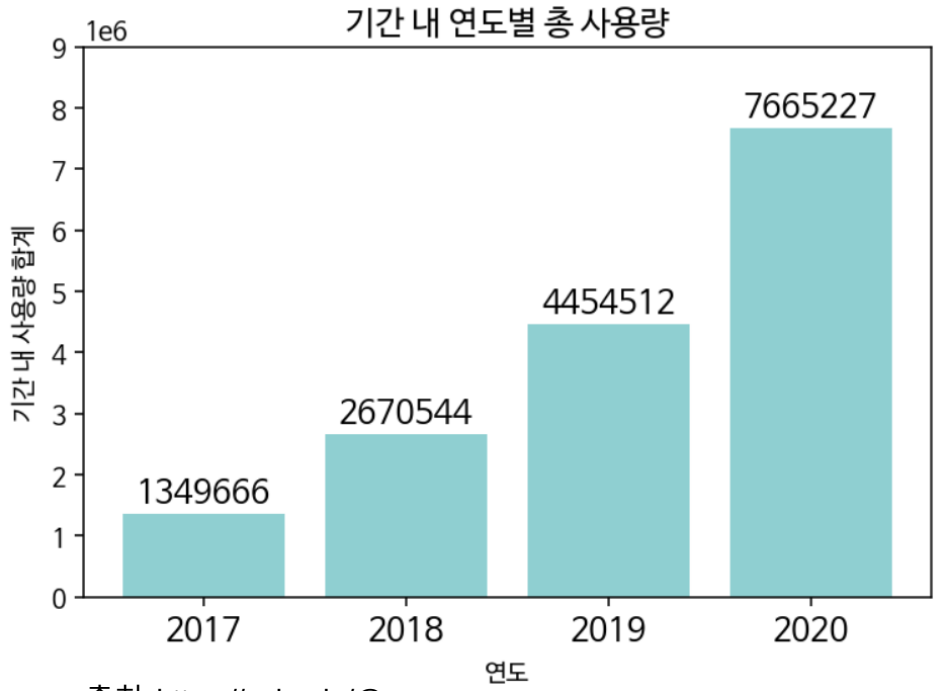
Do you know K-bike?

- 서울시에서 운영하는 무인 공공자전거 대여 서비스
- 크게 LCD형, QR형 2종류의 형태
- 웹이나 앱을 통해 이용권(정기권/일일권)을 구매하여 사용
- 2020. 10. 31 기준 2,090개의 대여소와 29,500개의 자전거가 가동 중

따릉이 소개 및 현황 | 급속히 증가하는 사용량

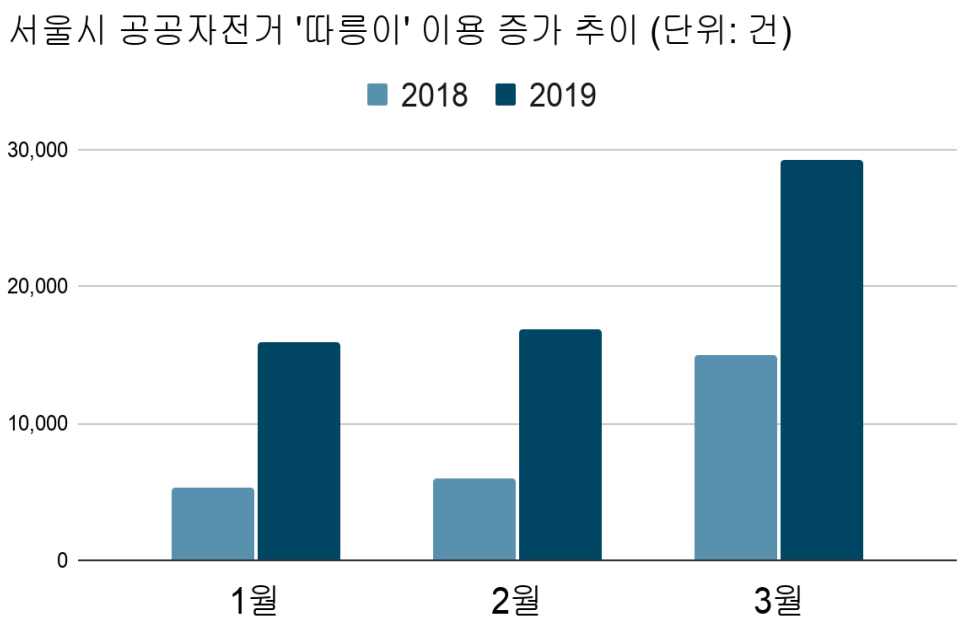
매년 따릉이의 수요는 성수기와 비수기 구분없이 **급격하게 증가**하는 추세

따릉이 사용량, 꾸준히 증가



출처: https://velog.io/@gayeon

비수기 이용건수 200% 증가

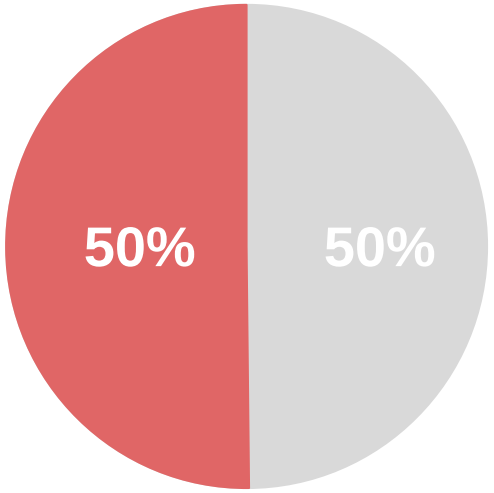


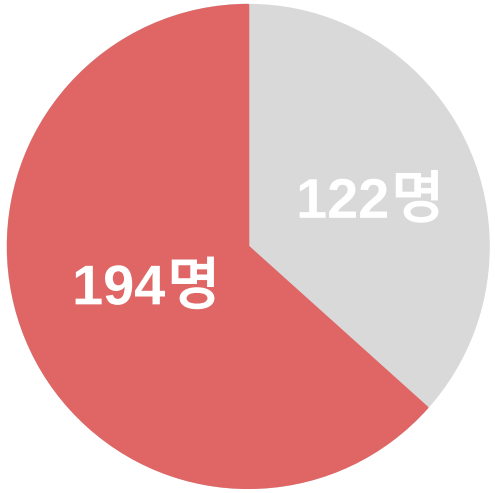
출처: 서울시 '서울열린데이터광장'

따릉이 소개 및 현황 | 수요 대비 부족한 인프라

- 따릉이의 사용량은 지속적으로 증가하는 반면, 따릉이의 인프라는 그 사용량을 따라가지 못하고 있는 상황
- 서울시에서는 인원을 늘릴 계획은 아직 없으며, 24시간 내내 배송원들이 재배치를 하여도 자전거의 불균형이 발생

▲ 비수기 인프라 축소





비수기의 목표 거치율은 성수기 대비 20%감소한 **50%**
비수기의 인력은 성수기 대비 122명 감소한 **194명**

▲ 2020 서울시의 따릉이 추가 배치 계획

구 분	사 업 명	주 요 내 용	추진계획	
			계획수립	실행
혁신 경영	공공자전거 양적 인프라 확대 (질적 인프라 개선)	○ 촘촘한 대여소 설치를 통한 대중교통과의 연계성 및 주거지역 접근성 강화 구 분 / 2019년 / 2020년 / 증 감 자전거 / 25,000대 / 40,000대 / 15,000대 대여소 / 1,540개소 / 3,040개소 / 1,500개소 ○ QR형 단말기 도입을 통한 실시간 위치 추적 – 미아 따릉이 최소화 ○ 전기자전거 500대 시범운영 방향 전환 – 시민대상 ➡ 공공기관 업무용 활용 추진	2~3월	4~11월

따릉이의 문제점 인식 | 사용자 불만 사항

자료: 서울특별시 따릉이 공식 홈페이지 FAQ

대여소마다 계속 0대
2020.05.01 ma******

저는 주로 저녁~밤 시간대에 따릉이를 서초,관악 일대에서 따릉이를 이용하는데
최근 주변의 모든 대여소마다 걸핏하면 따릉이가 0대입니다.

어쩌다 한 대여소에만 0대인 것이 아니라
주변 5~6개 대여소가 모두 0대라서
몇 km씩 걸어도 따릉이를 탈 수 없는 경우가 허다합니다.

대여소마다 자전거가 없어요.
2020.05.01 da******

정기권 구매하면 뭐하나요...
자전거가 0대에 타기는 하늘에 별따기에요.
대여소와 자전거 갯수 하루빨리 늘릴 필요가 있어요.
너무 불편하고 돈 아깝습니다.

따릉이가 없어요.
2020.04.27 ba******

6개월치 따릉이권을 구매해서 주변 따릉이를 빌리려고하면 정말.. 따릉이가 없습니다. 따릉이 수에 비해 이용자가 많은거같은데 이렇게 탈려고하면 매번 없는데 따릉이 수
를 늘리던가 아니면 이용자 수를 제한하던가 해야되는거 아닌가요? 매번 타고싶어도 없어서 못타고 있습니다. 제 위치에서 걸어서 30분 걸어야지 2,3대 있는 곳이있는데 막상 가면 고장나
서 열리지도 않고 타지도 않아요. 환불정책이나 답변 부탁드립니다.

자전거가 없어요
2020.04.07 lk****

최근 상암동에 자전거가 없어요
다 어디로 간건가요?
자전거 배치좀 해주세요
많던 자전거가 없으니, 출근하기가 너무 힘듭니다.

남부터미널 주변 따릉이가 없어요
2020.04.16 so******

요즘 날씨가 따뜻해져서 따릉이가 정말 없네요... 주변에 따릉이가 없으면 걸어가서 다른 곳에서라도 타는데 이건 없어도 너무 없어요... 자전거를 최대한 많이 공급해주세요..

따릉이가 없어요 (1346,1347 대여소)
2020.05.04 pa******

아침 9시에 출근 오후 5시 퇴근하는 사람인데요
집근처에 따릉이 정류소가 2개나 있는데 9시에는 한대도 없어요 1년 대여권 결재 했는데 9시에는 이용할수가 없어요
길음역까지 타고 가는데 9시15분경에 항상 따릉이랑 주변 청소 하시는 조록조끼 입으신 어르신분들이 계시던데 청소도 중요 하지만
출근시간부터 오전에는 따릉이를 주거지 근처로 옮겨주시는 작업이 시급 할 꺼 같습니다.
이용권 반납해야 하나 심각하게 고려중 입니다.
인력부족이 심각한거 뉴스통해서 들어서 알고 있지만 알바를 오전시간에만 써서라도 출근시간대 따릉이를 이용 할 수 있게 해주시면 좋겠습니다.
정류소 늘리는거와 함께 따릉이 이동 꼭 부탁 드립니다.~~

요즘 대여를 할수가없어요. 새로 만든 따릉이는 왜 방치하나요!?
2020.05.01 hw******

따릉이가 없어요
2020.03.31 hj****

있다고 해서 갔는데 없어요
2020.03.31 sd*****

어플에선 2대 있다고 했는데, 바로 갔는데도 0대예요...
누가 그렇게 빨리 가져가는건지.. 따릉이 한번 타는게 일이네요 ㅠㅠ

민원이 빈번하게 제기됨에도 불구하고
추가적인 인원 고용 예정 없음.
직접적인 재배치를 통한 문제해결이 어려운 상황

“따릉이 계속 0대”

“매번 따릉이가 없음”

“정기권을 끊어 놓고도 이용할 수 없음”

“찰나에 자전거 놓침”

주제 선정 | 주제 구체화

“실시간 따릉이 잔여대수 예측을 통한 사용자 불만제로 프로젝트”

사용자가 따릉이를 **대여할 시간대의 잔여대수**를 미리 알 수 있다면,
거치소에 갔을 때 이용 가능한 따릉이가 없는 **문제점을 사전에 예방** 가능.

큐브플로우와 **키바나**를 사용하여 우선 시범적으로 매 10분마다 10분 뒤를 예측하는 프로젝트를 기획.
예시) 12시 10분에 12시 20분의 서울 전역 거치소들의 잔여대수 예측

Kubeflow X  X kibana

Kubeflow를 이용한 **자동화**

따릉이 **실시간** OPEN API

Kibana를 이용한 **시각화**

주제선정 | 선행 연구 사례 및 차별점

선행 연구

- 특정 시간의 잔여대수를 예측
- 한개의 거치소만 잔여대수를 예측
- 실시간 데이터가 아닌 과거 대여 및 반납데이터만을 가지고 예측

**실시간 데이터를 반영하지 못하고,
분석 범위가 행정동 단위로 한정적**

본 프로젝트

- 서울특별시 전체 거치소의 잔여대수를 예측
- 실시간 따릉이 잔여대수 데이터로 모델링을 하여 기존 데이터와 서울시의 자전거 재배치까지 반영된 예측값 제시
- 실시간 날씨, 통합대기환경지수 데이터를 가져와 거치소의 잔여대수를 트렌디하게 예측

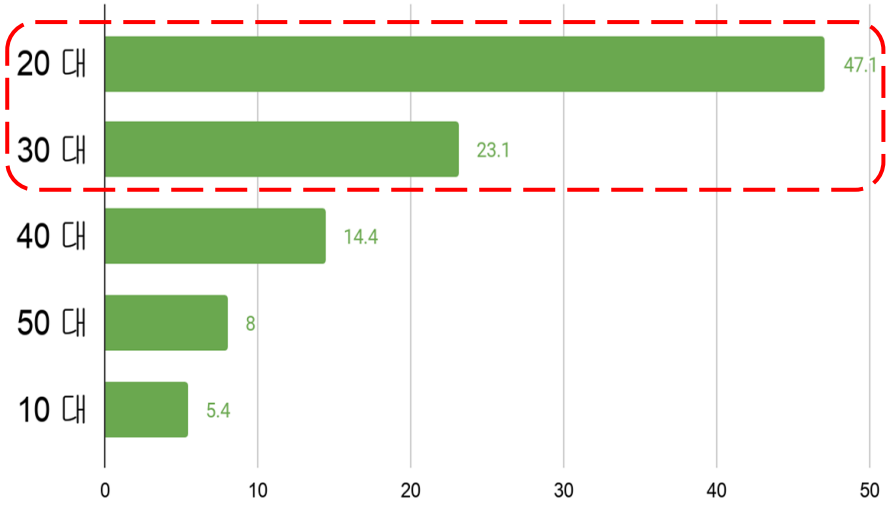
**거치소가 가지는 지역속성 변수를 추가하여
정확한 예측값을 실시간으로 사용자에게 제공**

CONTENTS

지역속성 변수 | 20대~30대 인구수

자치구별 20대와 30대의 인구수 변수 추가

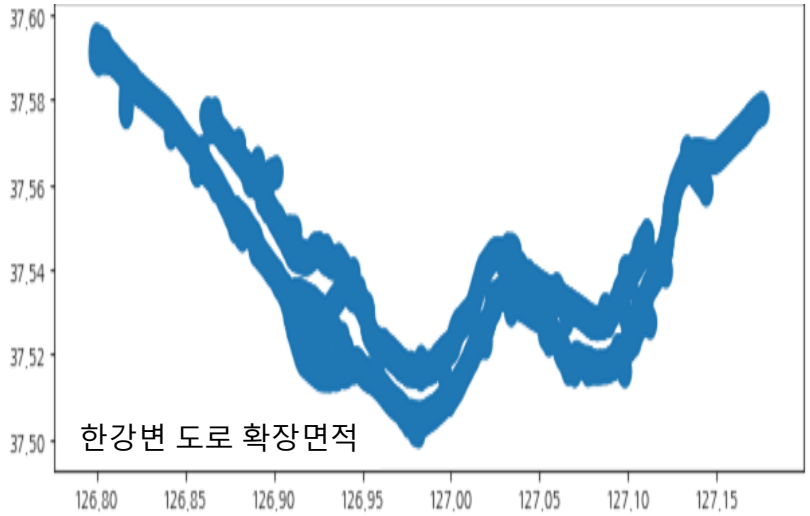
연령대별 따릉이 이용자 비율(단위 : %)

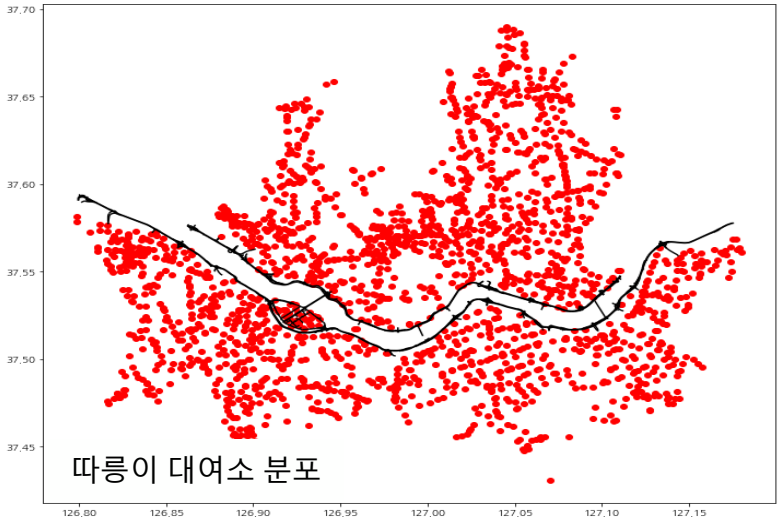


70% 이상

	대여소번호	자치구	20s	30s
0	1001	강동구	60955	71144
1	1002	강동구	60955	71144
2	1003	강동구	60955	71144
3	1004	강동구	60955	71144
4	1006	강동구	60955	71144
...	...	...	...	...
2038	992	은평구	66558	68357
2039	993	은평구	66558	68357

지역속성 변수 | 거치소 400m내 한강 위치 여부





- 한강변 도로 LineString(선) 안에 있는 점 데이터들을 400m의 범위를 가지도록 변환
- 변환된 데이터는 선이 아닌 Multipoligon(면적)의 값을 가진다.

- 각 점들의 위도, 경도값을 Point(점)으로 변환해 해당 점이 좌측 한강변 면적에 포함되는지 안되는지 변수로 생성

지역속성 변수 | 거치소의 400m 반경 내 시설물 포함개수

- **in400_bike:** “여기는 대여소가 응집되어 있는걸 보니 사람들이 많이 따릉이를 타나봐 ~”
- **in400_market:** “장보고 들고가기 힘드니~ 따릉이 바구니에 담고 집에가야지~”
- **in400_park:** “자전거 타고 공원을 돌아야겠다~”
- **in400_school:** “ 학교 끝나고 집갈때 자전거 타고 가야지!”
- **in400_culture:** “한가할때 즐기는 문화시설인데 자전거 타고 놀다 들어가야지~”
- **in400_bus:** “대중교통으로는 내 직장 앞까지 안가니깐 내려서 따릉이를 타고 가야겠다!”
- **in400_subway:** “우리집은 역세권이 아니라 집앞까지 가려면 따릉이를 타야만해..□ □ "

		또 다른 거치소	시장	공원	지하철	학교	문화 공간	버스 정류장
	대여소번호	in400_bike	in400_market	in400_park	in400_subway	in400_school	in400_culture	in400_bus
0	101	3	0	2	0	0	0	9
1	102	2	1	2	1	0	0	22
2	103	3	1	3	1	0	0	18
3	104	4	2	2	2	1	2	20

▲ 400m 반경 내 각 주요 시설물 포함 개수

지역속성 변수 | 거치소로부터 각 시설물별 최단거리(거치소포함)

거치소마다 다양한 시설과의 최단거리를 feature로 사용

거치소와 지하철 출입구, 공공시설 그리고
대학교까지의 최단 거리가 짧을수록 공공자전거 하루
평균 대여와 반납건수가 높은 것으로 나타나
공공자전거 대여소 주변의 접근성 특성이 공공자전거
이용에 강한 영향을 미치는 것으로 확인하였다.

- 서울시 공공자전거 이용에 영향을 미치는 물리적 환경 요인 분석 : 대여소별 거리에 따른 요인의 영향력 차이를 중심으로 사경은… ; 이수기…

- 400m안에 시설물들을 포함하고 있지 않은 거치소들은 서로 지역속성 비교가 어려움 => 최단거리 feature로 비교하고자 함
- 거치소와 지하철 출입구와 공공시설, 대학교까지의 최단거리가 짧을 수록 공공자전거 이용에 영향을 주는 것을 확인하여 추가적으로 시장, 공원, 문화시설, 버스, 또다른 거치소와의 거리를 feature로 사용

▲ 해당 거치소에서 각 주요 시설물과의 최단 거리

	대여소번호	market_shortest	park_shortest	subway_shortest	school_shortest	culture_shortest	bus_shortest	bike_shortest
0	1001	259.669	1702547.173	368.294	514.881	253.581	31.235	249.202
1	1002	417.241	1702162.130	685.778	181.586	146.577	70.992	150.577
2	1003	333.558	1702306.993	608.381	299.472	9.970	129.669	150.577

실시간 변수 | 시간대별 따릉이 거치소별 잔여대수

- 10분 뒤를 예측하는 모델을 만들기 위해 따릉이실시간 잔여대수를 API를 통해 10분단위로 수집
- 자전거 운송요원들이 24시간 3교대근무를 하면서 자전거 거치율을 유지하기 위해 재배치 진행 중 => 자전거가 채워지는 사이클 24시간
- 예측하는 시점대에 대한 패턴을 학습하기 위해서는 두번의 사이클이 적당하다고 판단(전날의 다른 특징적인 요인을 고려할 필요가 있다)
- 예측시점으로부터 48시간 전까지의 따릉이 잔여대수 실시간 데이터를 feature로 사용

… DB에는 10분 단위로 데이터를 자동으로 수집하는 중 …

호출데이터 _예측시점으로부터 48시간 전까지의 따릉이 잔여대수 실시간 데이터

모델이 전일 동일 시간대의 데이터값을 고려한 학습 가능

	대여소번호	2020-12-07 22:10 자전거	2020-12-07 22:20 자전거	2020-12-07 22:30 자전거	2020-12-07 22:40 자전거	2020-12-07 22:50 자전거	2020-12-07 23:00 자전거	2020-12-07 23:10 자전거	2020-12-07 23:20 자전거	2020-12-07 23:30 자전거	…	2020-12-09 21:30 자전거	2020-12-09 21:40 자전거	2020-12-09 21:50 자전거	2020-12-09 22:00 자전거	2020-12-09 22:10 자전거	2020-12-09 22:20 자전거	2020-12-09 22:30 자전거	2020-12-09 22:40 자전거	2020-12-09 22:50 자전거	2020-12-09 23:00 자전거
0	1001	2.0	2.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	…	2.0	3.0	3.0	2.0	3.0	4.0	6.0	6.0	6.0	6.0
1	1002	9.0	8.0	7.0	8.0	8.0	8.0	8.0	11.0	11.0	…	12.0	12.0	12.0	11.0	11.0	9.0	8.0	8.0	9.0	10.0
2	1003	4.0	4.0	3.0	3.0	4.0	4.0	4.0	4.0	4.0	…	4.0	4.0	4.0	4.0	5.0	5.0	6.0	6.0	6.0	6.0
3	1004	5.0	5.0	5.0	6.0	5.0	5.0	5.0	5.0	5.0	…	10.0	11.0	11.0	12.0	12.0	13.0	13.0	13.0	13.0	13.0

실시간 변수 | 시간별 통합대기환경 등급 변수

- 실시간 통합대기환경 데이터는 공공데이터포털api를 이용해 1시간 단위로 자치구별 통합대기환경 데이터를 수집
- 현재의 서울은 초미세, 미세먼지가 심하지는 않아 모델에 중요한 변수로 작용하지 않을 수 있으나, 심해지는 시기를 고려하여 추가
- 과거 미세먼지에 오래 시달리기도 했으며, 현재 코로나로 인해 다들 마스크를 착용하고 있어 어느정도의 미세먼지와, 초미세먼지에는 둔감해졌을것으로 판단.
- 위의 조건을 고려하여 통합대기환경을 두그룹으로 나누어 변수로 사용했습니다. 매우나쁨 = bad, (나쁨, 보통, 좋음) => good

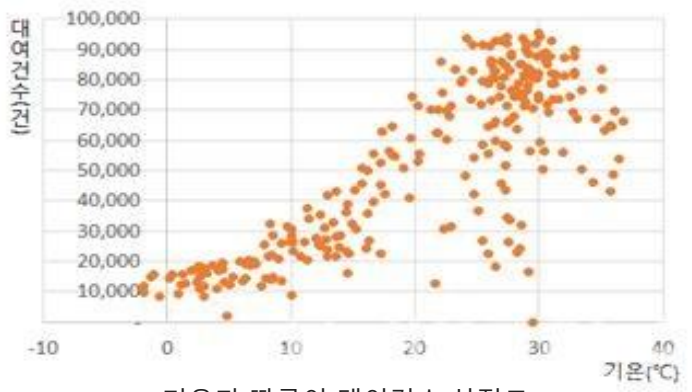
1시간 단위로 **통합대기환경** 데이터 수집

	관측일시	권역명	2020-12-07 17:00 통합대기환경 등급	2020-12-07 18:00 통합대기환경 등급	2020-12-07 19:00 통합대기환경 등급	2020-12-07 20:00 통합대기환경 등급	2020-12-07 21:00 통합대기환경 등급	2020-12-07 22:00 통합대기환경 등급	2020-12-07 23:00 통합대기환경 등급	2020-12-08 00:00 통합대기환경 등급	2020-12-08 01:00 통합대기환경 등급	...	2020-12-09 13:00 통합대기환경 등급	2020-12-09 14:00 통합대기환경 등급	2020-12-09 15:00 통합대기환경 등급	2020-12-09 16:00 통합대기환경 등급	2020-12-09 17:00 통합대기환경 등급	2020-12-09 18:00 통합대기환경 등급	2020-12-09 19:00 통합대기환경 등급	2020-12-09 21:00 통합대기환경 등급	2020-12-09 22:00 통합대기환경 등급	2020-12-09 23:00 통합대기환경 등급
0	강남구	good	good	good	good	good	good	good	good	good	good	...	good	good	good	good	good	good	good	good	good	good
1	강동구	good	good	good	good	good	good	good	good	good	good	...	good	good	good	good	good	good	good	good	good	good
2	강북구	good	good	good	good	good	good	good	good	good	good	...	good	good	good	good	good	good	good	good	good	good

```
# 매우나쁨 = bad
# 나쁨, 보통, 좋음 => good
```

실시간 변수 | 시간대별 체감온도와 시간대별 기후상태

기온과 대여건수의 상관관계 분석



기온과 따릉이 대여건수 산점도

출처 : 서울로컬뉴스(http://www.slnews.co.kr)

온도와 기후는 자전거 이용에 영향을 미친다

기상조건에 따른 영향은 평균 기온이 상승할수록 대여량이 늘어나는 것으로 분석되었으며, 강수량이 10mm 이상 되거나, 평균기온이 29도 이상으로 높아지는 경우, 풍속이 7m/s 이상 되는 경우에 대여량이 떨어지는 것으로 분석되었다.

- 이장호, 정경옥, 신희철. (2016). 기상조건과 입지특성이 공공자전거 이용에 미치는 영향 분석. 대한교통학회지, 34(5), 394-408.

- 실시간 날씨 데이터 DB를 생성하여 openweather-API를 이용해 1시간 단위로 자치구별 날씨 데이터를 수집
- 따릉이 자전거에서 48시간의 데이터를 사용하였기에 자치구별 체감온도 데이터와 기후상태 데이터를 가장최근으로부터 48시간의 데이터를 가져와 모델의 feature로 사용하였습니다

1시간 단위로 **체감온도** 데이터 수집

	관측일시	권역명	2020-12-07 23:00 날씨	2020-12-08 00:00 날씨	2020-12-08 01:00 날씨	2020-12-08 02:00 날씨	2020-12-08 03:00 날씨	2020-12-08 04:00 날씨	2020-12-08 05:00 날씨	2020-12-08 06:00 날씨	2020-12-08 07:00 날씨	...	2020-12-09 14:00 날씨	2020-12-09 15:00 날씨
0	강남구		-5.45	-5.29	-5.63	-5.85	-6.62	-6.74	-7.73	-8.51	-8.20	...	1.96	2.05

1시간 단위로 **기후상태** 데이터 수집

	관측일시	권역명	2020-12-07 23:00 날씨	2020-12-08 00:00 날씨	2020-12-08 01:00 날씨	2020-12-08 02:00 날씨	2020-12-08 03:00 날씨	2020-12-08 04:00 날씨	2020-12-08 05:00 날씨	2020-12-08 06:00 날씨	2020-12-08 07:00 날씨	...	2020-12-09 14:00 날씨	2020-12-09 15:00 날씨
0	강남구		1	1	1	1	1	1	1	1	1	...	1	1

clear, cloud == '1'

else == '0'

Feature 요약 | 변수 구분

지역속성 변수

- 자치구별 거치소의 20대 & 30대 인구수
- 거치소 기준 400m 내 한강포함 여부
- 거치소 기준 400m 내 타 거치소 및 시설물 개수
- 거치소 기준 다른 거치소 및 시설물까지의 최단거리

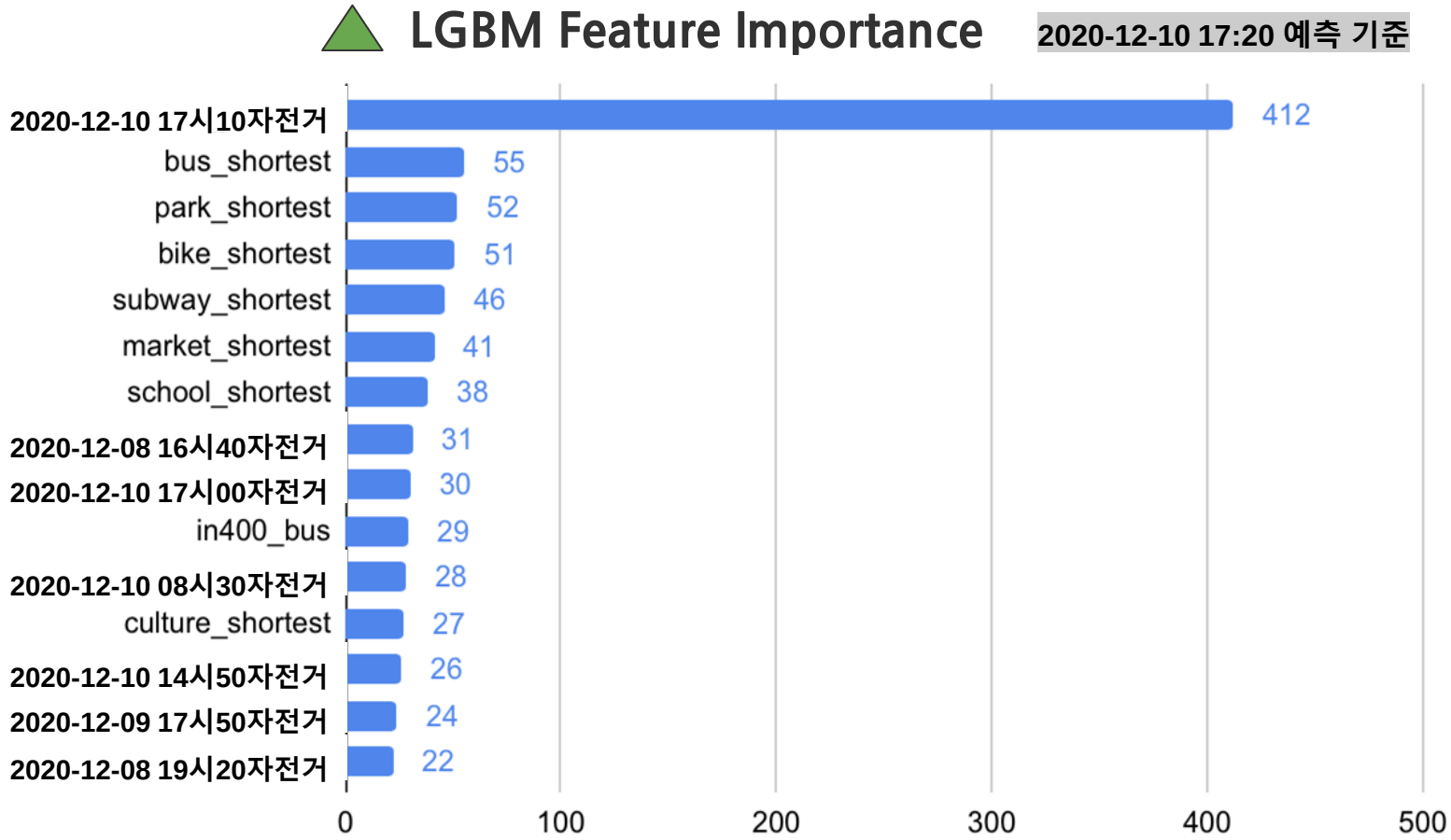
실시간 변수

- 따릉이 48H 전~ 현재 / 10 min 간격
- 날씨(체감온도) 48H 전 ~ 현재 / 1H 간격
- 날씨(기상상황) 48H 전 ~ 현재 / 1H 간격
- 미세먼지(통합대기환경) 48H 전 ~ 현재 / 1H 간격

Train Data

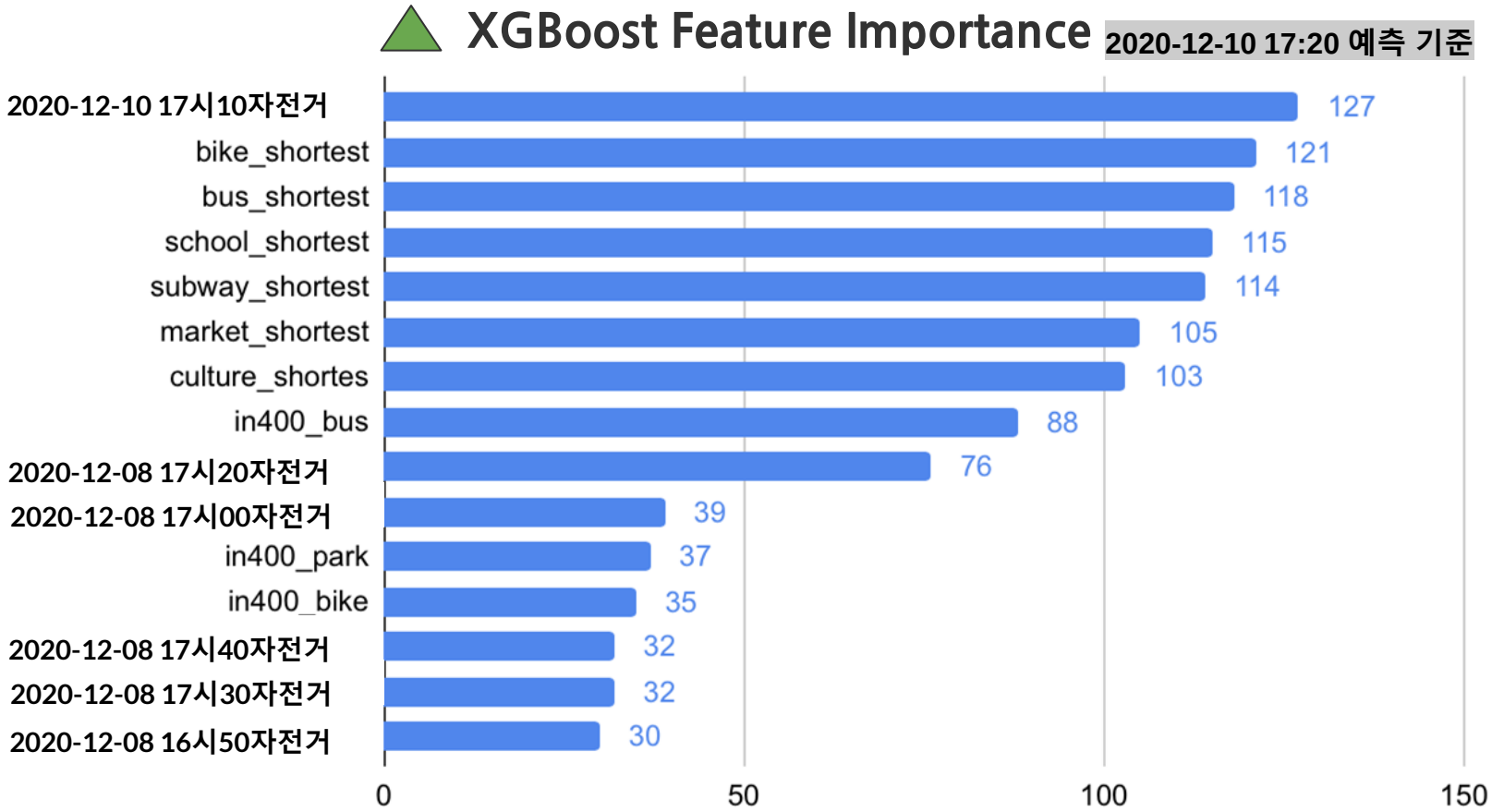
변수 중요도 검증 | LGBM feature importance 통한 검증

- lgbm의 경우 예측 시점으로부터 가장 최근의 시간대의 자전거 잔여대수가 가장 변수 중요도가 높았다.
- 그다음으로는 지역 속성 변수들이 중요도가 높았다.



변수 중요도 검증 | XGBoost feature importance

- xgboost의 경우 마찬가지로 가장 최근의 시간대 자전거 잔여대수가 가장 변수 중요도가 높음
- 그 다음으로는 지역 속성 변수들의 중요도가 높게 나타남



CONTENTS

모델링 준비 | 머신러닝 모델 및 rmse 평가지표 선택배경

머신러닝 선택 이유

- 단순히 시계열적인 데이터만 사용하는 것이 아닌 변하지 않는 고정값인 지역 속성 데이터를 학습에 사용하기 때문에 딥러닝으로 분석할 경우 지역 속성데이터가 의미를 잃는다는 점
- 딥러닝은 하나의 모델로 모든 거치소의 잔여대수를 예측해야하기 때문에 모든 거치소에 각각의 모델을 학습하기 어렵다는 점

↓

- 머신러닝중 대표적인 모델인 lgbm과 xgboost 이 두가지 모델의 예측값 평가
- 두가지 모델의 결과값을 앙상블 하여 평가

RMSE 선택이유

- 회귀모델의 평가지표로는 여러가지가 있지만 RMSE, RMSLE 이 두가지의 지표를 처음에 고려함
- rmsle는 이상치에 둔감하다는 특징을 가지고 있음
- 몇분사이에 갑작스럽게 많은 대여와 반납이 발생할 수 있는 따릉이 데이터를 반영하여 다음 시간의 잔여대수를 예측해야한다.

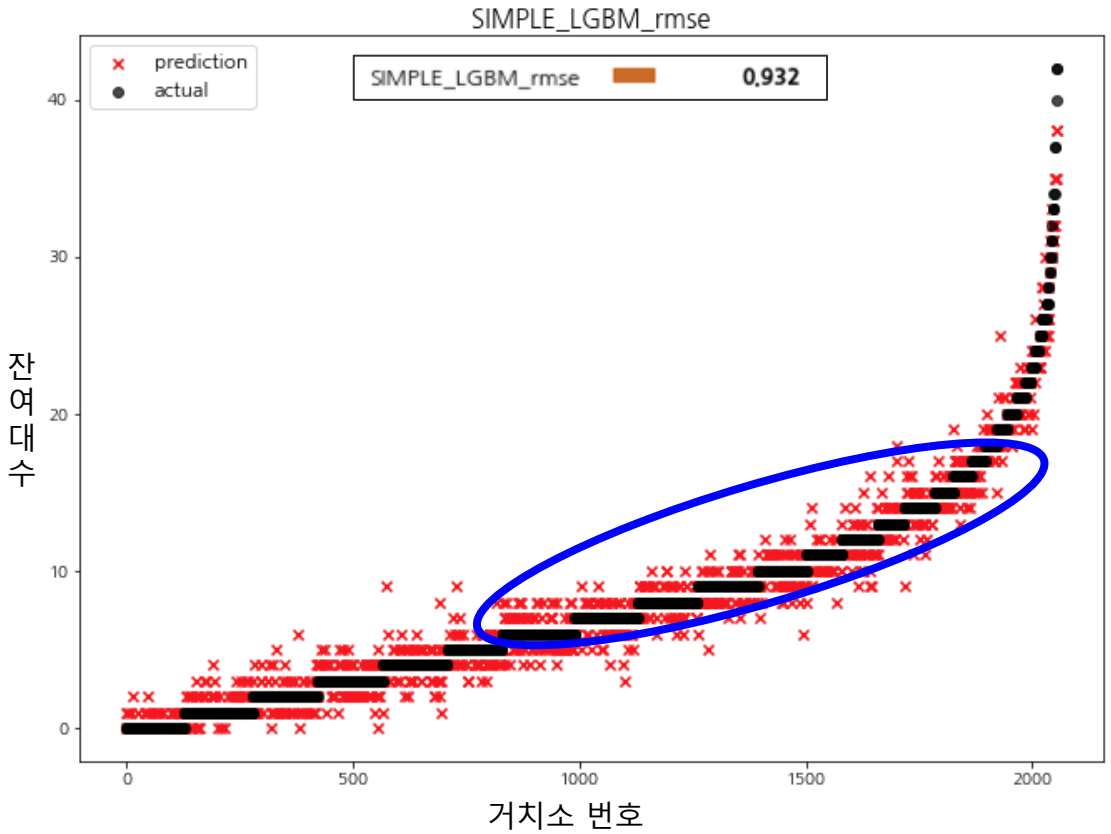
↓

- 이상치에 민감한 RMSE를 평가지표로 선정.

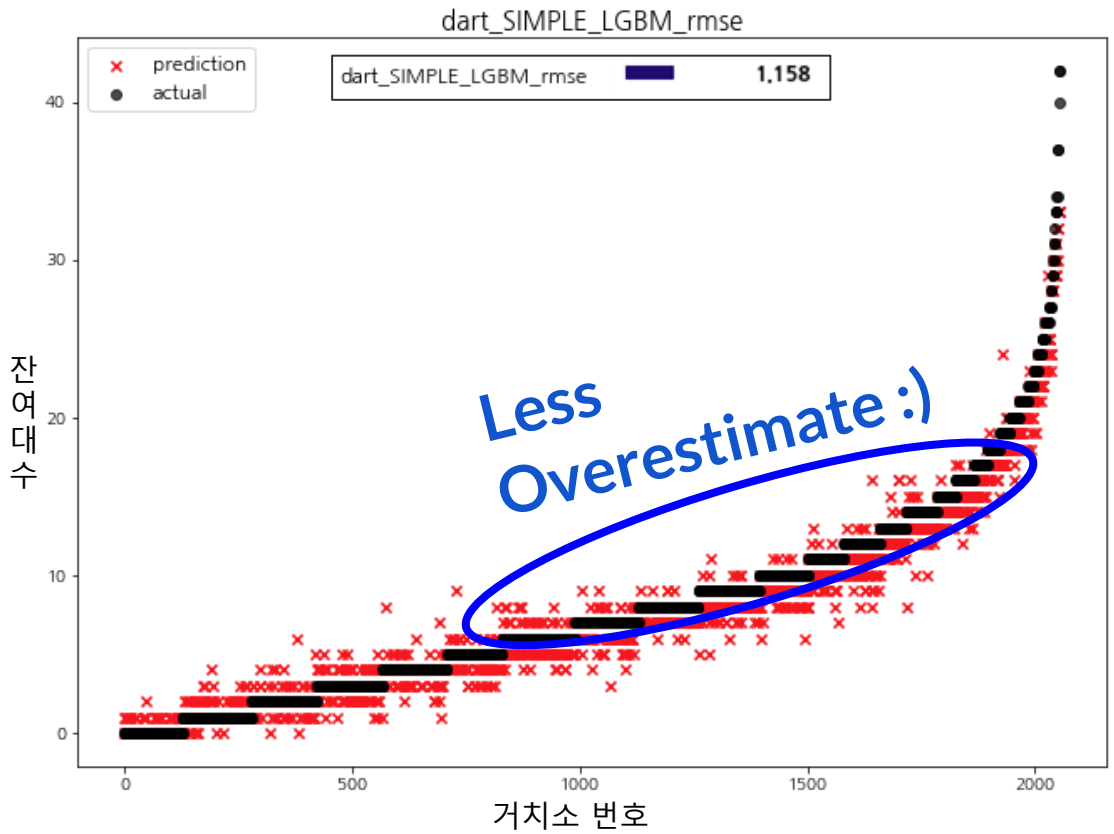
모델 선정 | LGBM regressors vs LGBM (boosting_type = DART)

- 회귀모델을 돌릴때 모델의 정확도를 높히기 위한 파라미터인 boosting_type = 'dart'를 준것과 비교를 해보았습니다.
- 그래프를 자세히 보면 파라미터로 boosting_type = 'dart'를 준 모델을 보면 simple lgbm에 비해 과대추정이 적어보이는 것을 알 수 있습니다.

Simple LGBM



Simple LGBM (boosting_type = DART)



모델 선정 | LGBM regressors (boosting_type = DART)을 선택한 이유

- 실제 잔여대수보다 높게 예측해버리면 실제로 사용자가 거치소에 갔을때 예측한 값보다 적어서 타지 못한다면 오히려 불만이 더 커질것으로 예상했습니다.
- 따릉이 사용자들의 불편 요소를 줄이기 위해서 정확도가 조금은 떨어지더라도 dart 파라미터를 주는 게 낫다고 판단했습니다.

Simple LGBM

```
1 eval_rmse_simple
```

	y_test	simple_lgbm_pred	simple_lgbm_pred_eval
0	8.0	9	overestimate
1	3.0	2	underestimate
2	0.0	1	overestimate
3	4.0	6	overestimate
4	1.0	2	overestimate
...	...	...	...
2048	3.0	5	overestimate
2049	7.0	6	underestimate
2051	8.0	9	overestimate
2053	1.0	2	overestimate
2055	5.0	4	underestimate

830 rows × 3 columns

N = 20	avg prec.
Under estimate	50%
Over estimate	50%

```
1 eval_rmse_simple.simple_lgbm_pred_eval.value_counts()
```

```
underestimate    423
overestimate     407
Name: simple_lgbm_pred_eval, dtype: int64
```

Simple LGBM (boosting_type = DART)

```
1 eval_rmse_simple_dart
```

	y_test	simple_lgbm_pred_dart	simple_lgbm_pred_dart_eval
1	3.0	2	underestimate
3	4.0	5	overestimate
4	1.0	2	overestimate
6	12.0	9	underestimate
7	9.0	10	overestimate
...	...	...	...
2050	10.0	9	underestimate
2052	8.0	7	underestimate
2053	1.0	2	overestimate
2054	17.0	16	underestimate
2055	5.0	4	underestimate

1146 rows × 3 columns

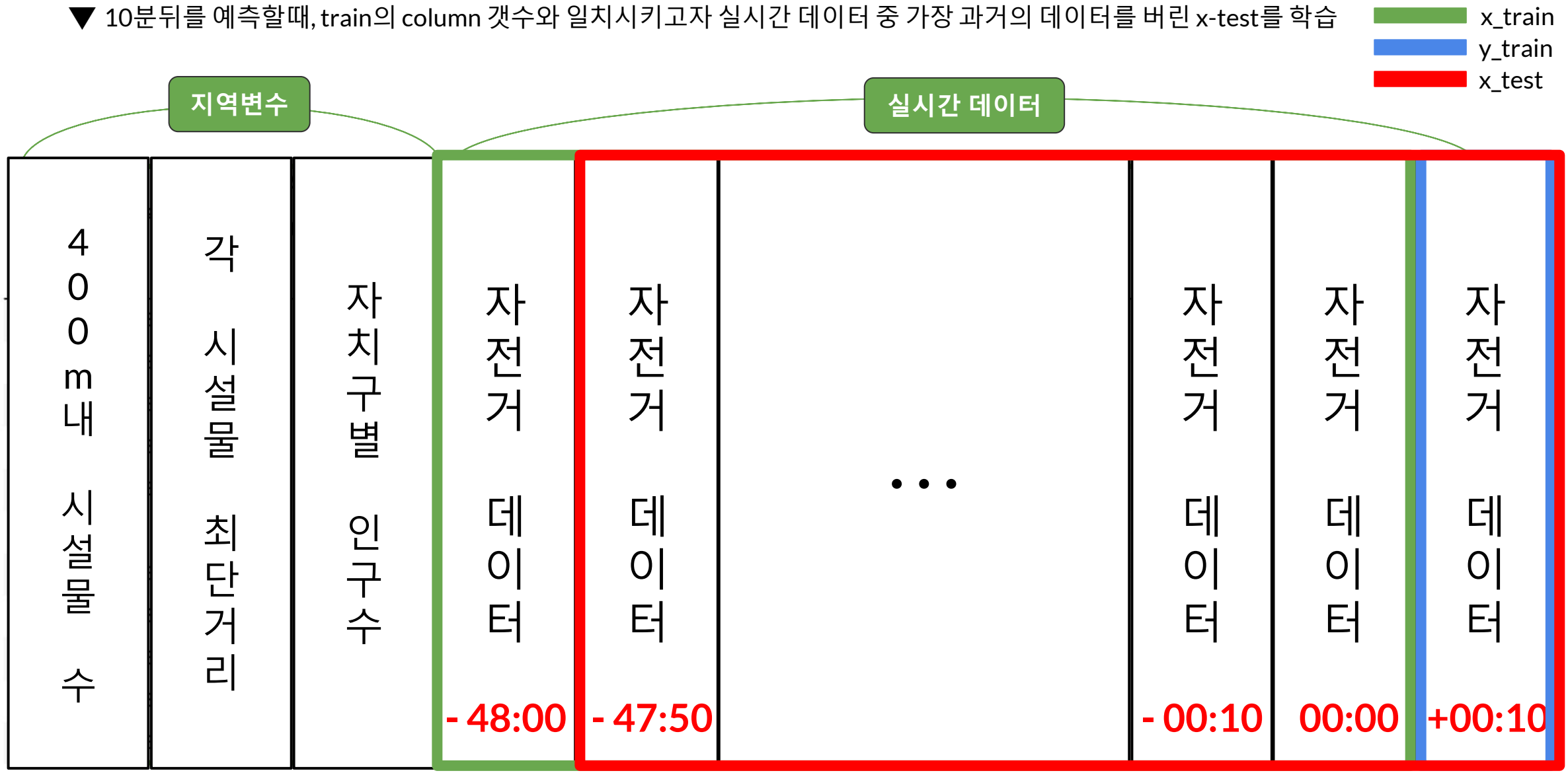
N = 20	avg prec.
Under estimate	77%
Over estimate	23%

```
1 eval_rmse_simple_dart.simple_lgbm_pred_dart_eval.value_counts()
```

```
underestimate    911
overestimate     235
Name: simple_lgbm_pred_dart_eval, dtype: int64
```

모델링 준비 | 학습 매커니즘

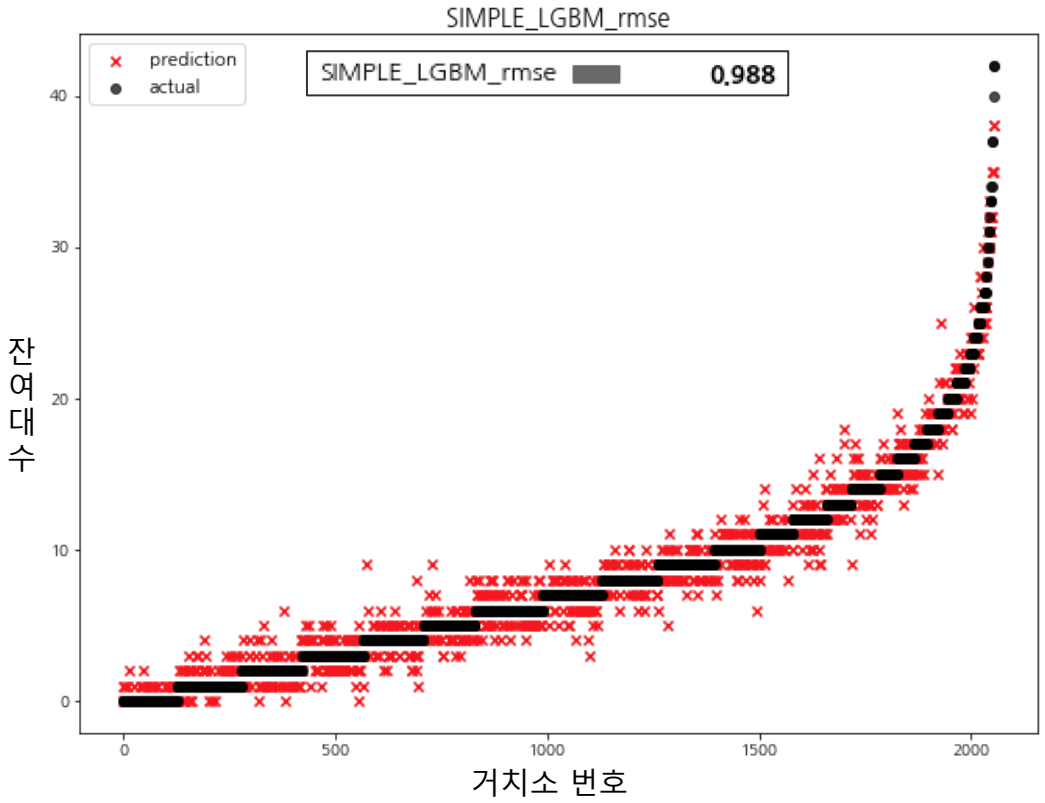
▼ 10분뒤를 예측할때, train의 column 갯수와 일치시키고자 실시간 데이터 중 가장 과거의 데이터를 버린 x-test를 학습



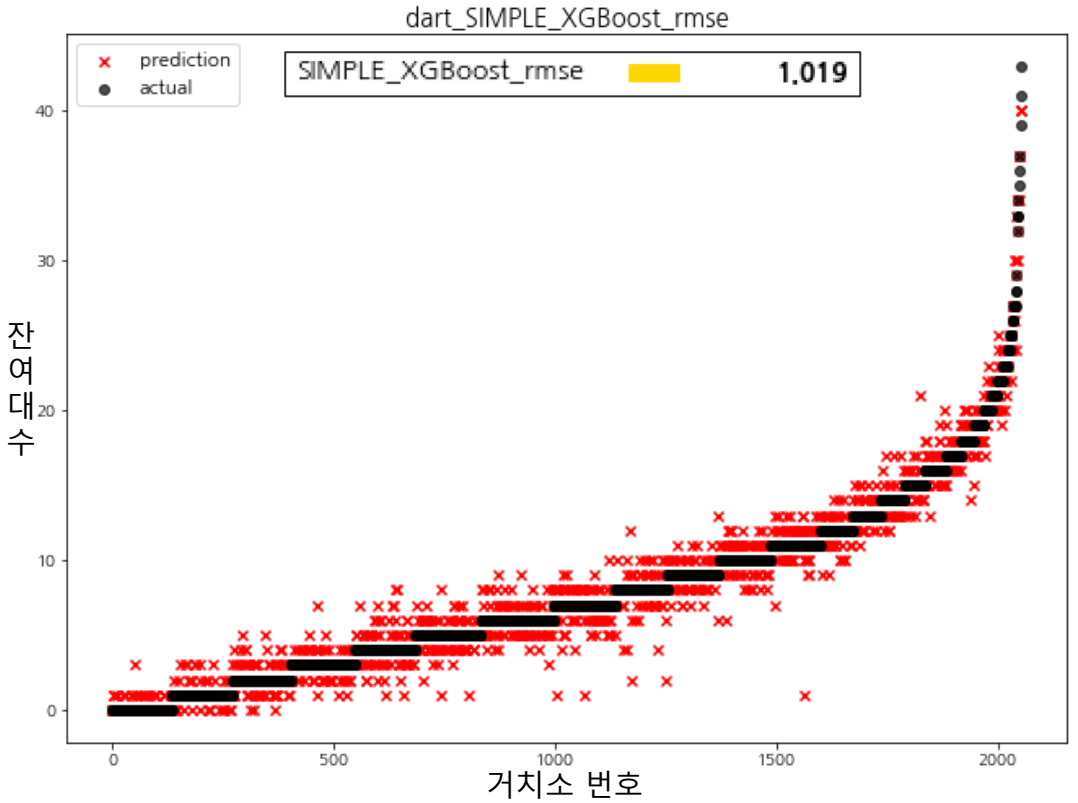
모델 선정 | LGBM vs. XGBoost

- lgbm의 rmse = 0.99 , xgboost의 rmse = 1.019 로 lgbm이 rmse값이 더 높음

LGBM



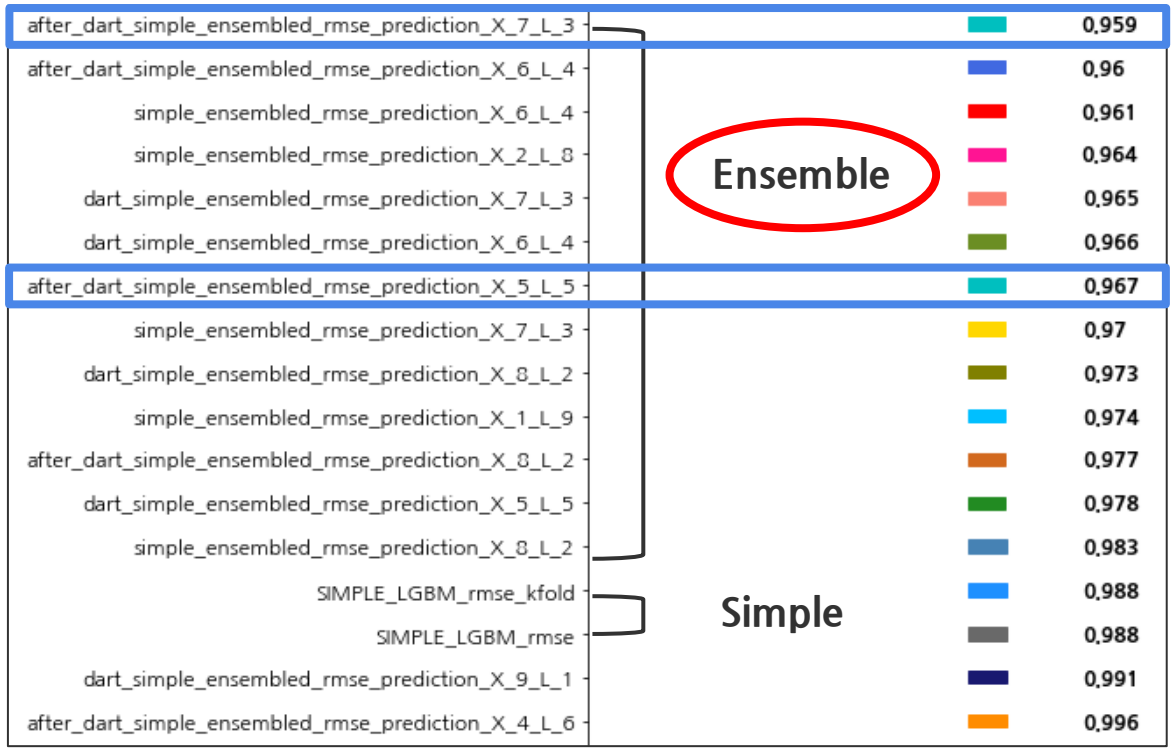
XGBoost



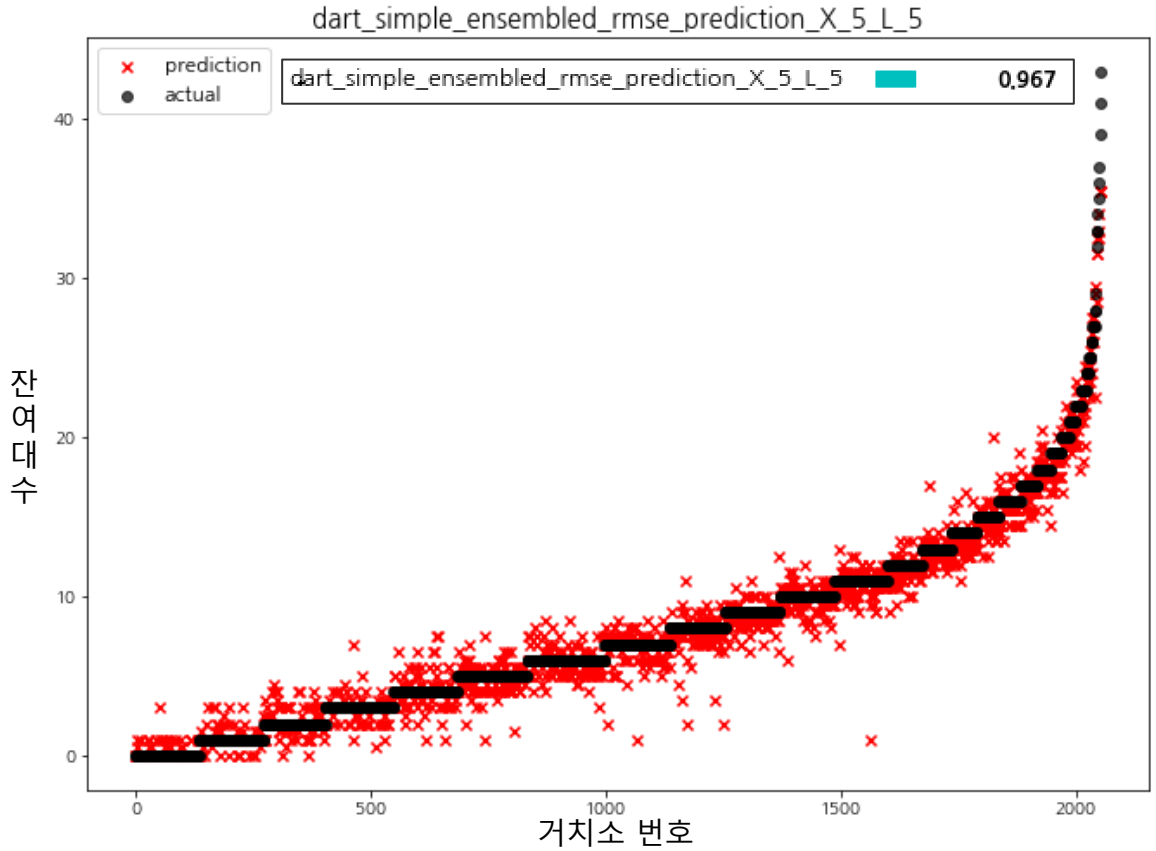
모델 선정 | LGBM & XGB => Ensemble

- 단일 모델에 비해 LGBM(dart) + XGB 를 앙상블한 모델의 성능이 Best!
- 최종 모델은 LGBM(dart)의 결과값에 0.5의 가중치와 XGB의 결과값에 0.5의 가중치를 주었고, 0보다 작은 값은 0으로, 실수는 반올림으로 처리하여 최종 예측값을 구함

평가지표를 통한 모델 성능 비교 | RMSE



Ensemble (DART LGBM + XGB)



최종모델 선정 | 테스트 결과

Ensemble 실제 테스트 결과
(DART LGBM*0.5 + XGB *0.5)

시간대별 예측 평가 (오전10 ~오후8시)

eval_df

	time	rmse
0	2020-12-09 09:50:00	0.992919
1	2020-12-09 10:00:00	1.040541
2	2020-12-09 10:10:00	1.029019
3	2020-12-09 10:20:00	0.923602
4	2020-12-09 10:30:00	0.896872
...	...	...
57	2020-12-09 19:20:00	1.020462
58	2020-12-09 19:30:00	1.068427
59	2020-12-09 19:40:00	1.070247
60	2020-12-09 19:50:00	1.033015
61	2020-12-09 20:00:00	1.110395

10시간 동안
60번의
예측 모델 오차
평균 기록
(eval_df)

```
1 eval_df.rmse.mean()
```

```
0.9933929680616311
```

매 단위 시간
약 1대 정도의 오차
(rmse.mean() = 0.9934)


```
1 eval_df.rmse.mean()
```

```
0.9933929680616311
```


```
1 data_under_3 = data[data.잔여대수 < 2]
2
3 np.sqrt(mean_squared_error(data_under_3.예측잔여대수, data_under_3.잔여대수))
```

```
0.5546626787111874
```

```
1 data_under_3 = data[data.잔여대수 < 3]
2
3 np.sqrt(mean_squared_error(data_under_3.예측잔여대수, data_under_3.잔여대수))
```

```
0.5934722315302949
```

```
1 data_under_3 = data[data.잔여대수 < 4]
2
3 np.sqrt(mean_squared_error(data_under_3.예측잔여대수, data_under_3.잔여대수))
```

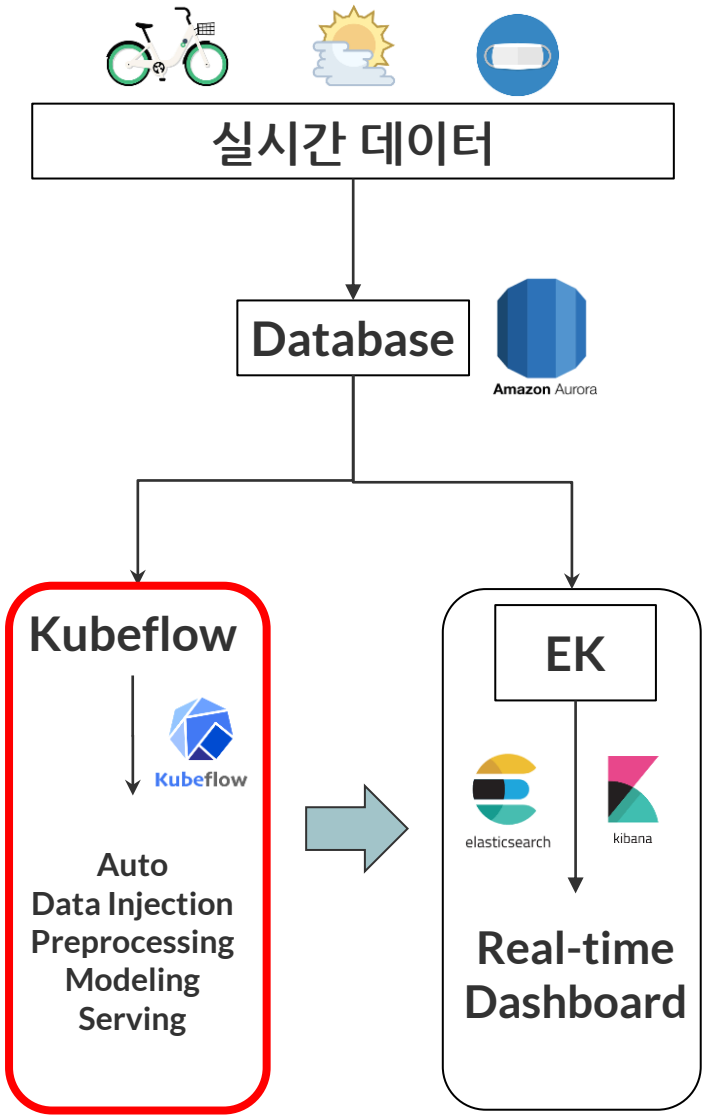
```
0.6311575023380444
```

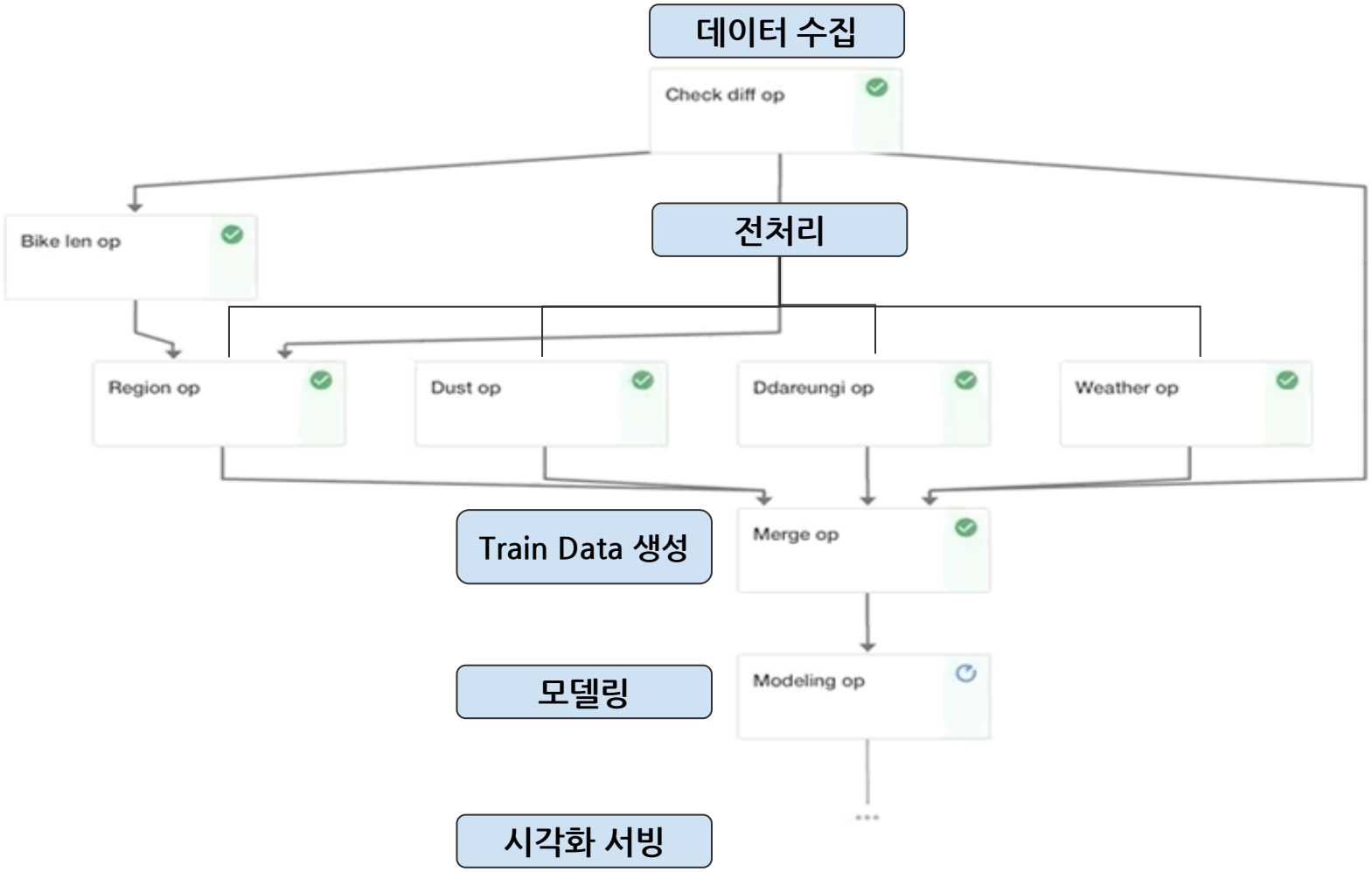
```
1 data_under_3 = data[data.잔여대수 < 5]
2
3 np.sqrt(mean_squared_error(data_under_3.예측잔여대수, data_under_3.잔여대수))
```

```
0.6603745772263538
```

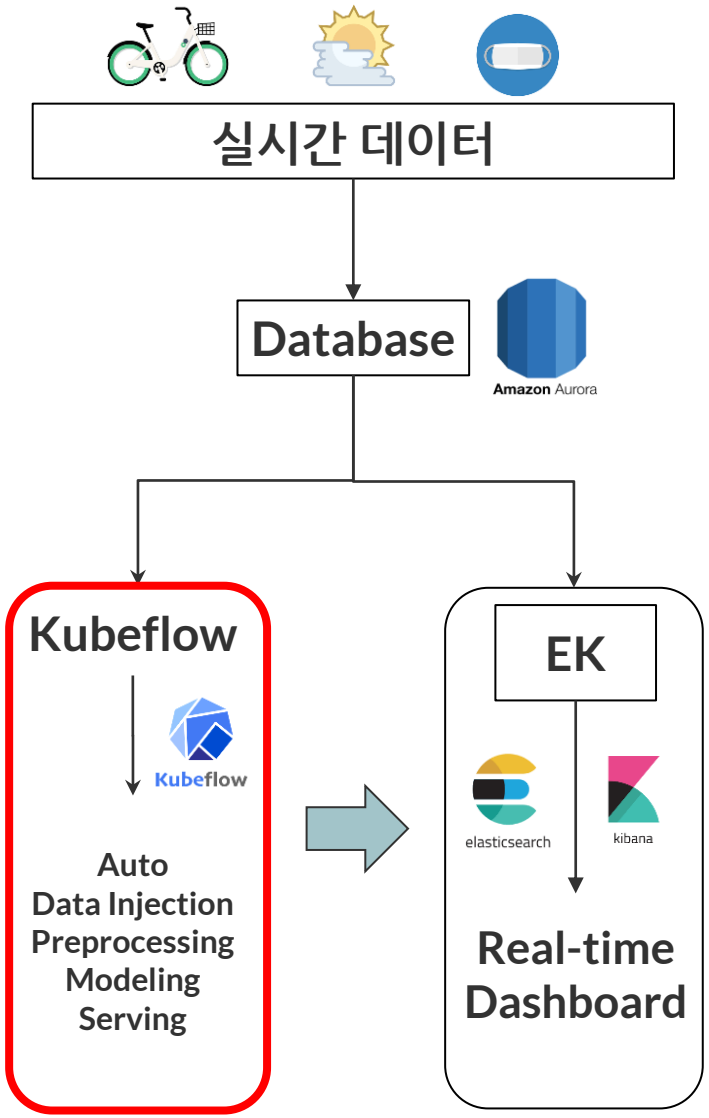
CONTENTS

Kubeflow Pipeline | 실시간 예측 모델링 자동화





Kubeflow Pipeline 장점 | 실시간 예측 모델링 자동화



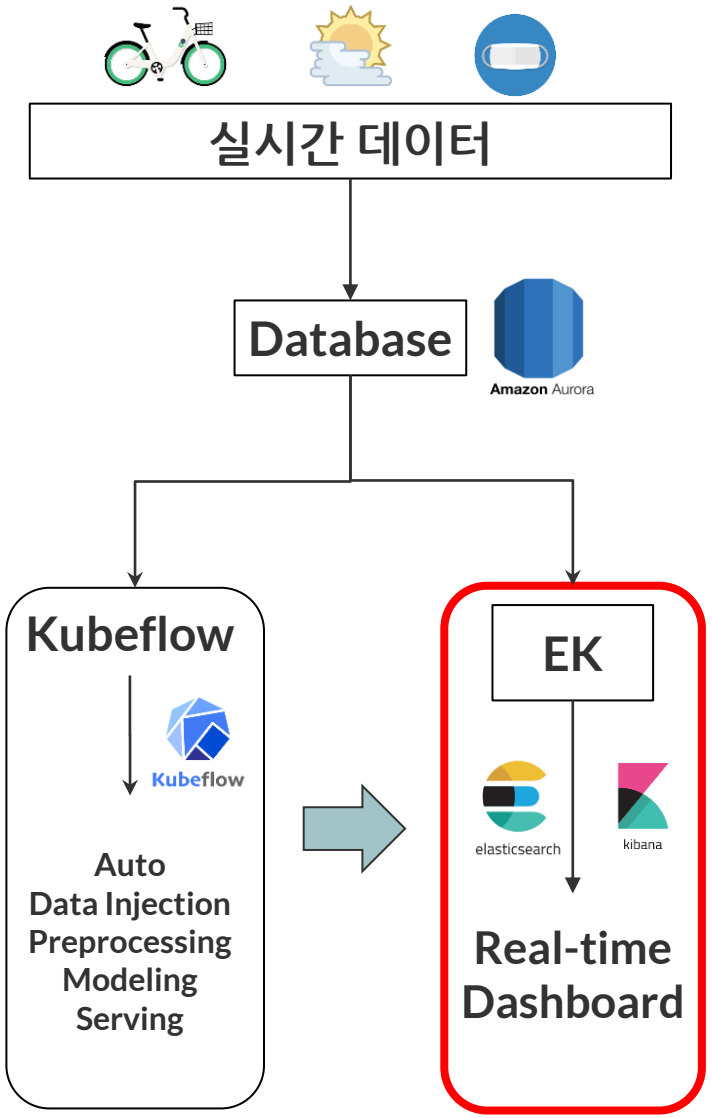
ML Workflow End-to-end Automation

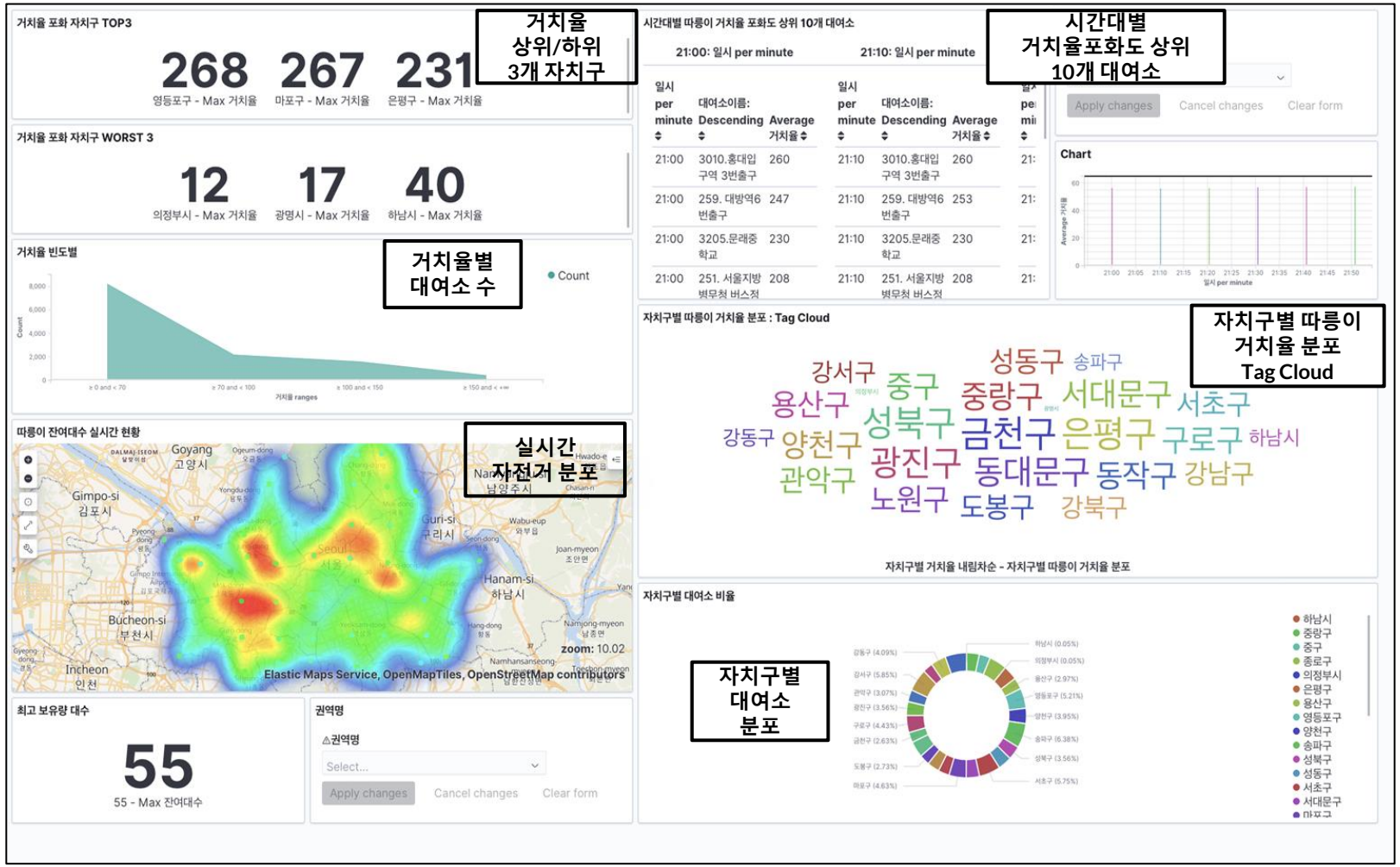
- 머신러닝 파이프라인 자동화
- 스케쥴링
- **실시간 ML 예측**
- **예측 결과 관리 용이**

Kubernetes 클러스터 환경

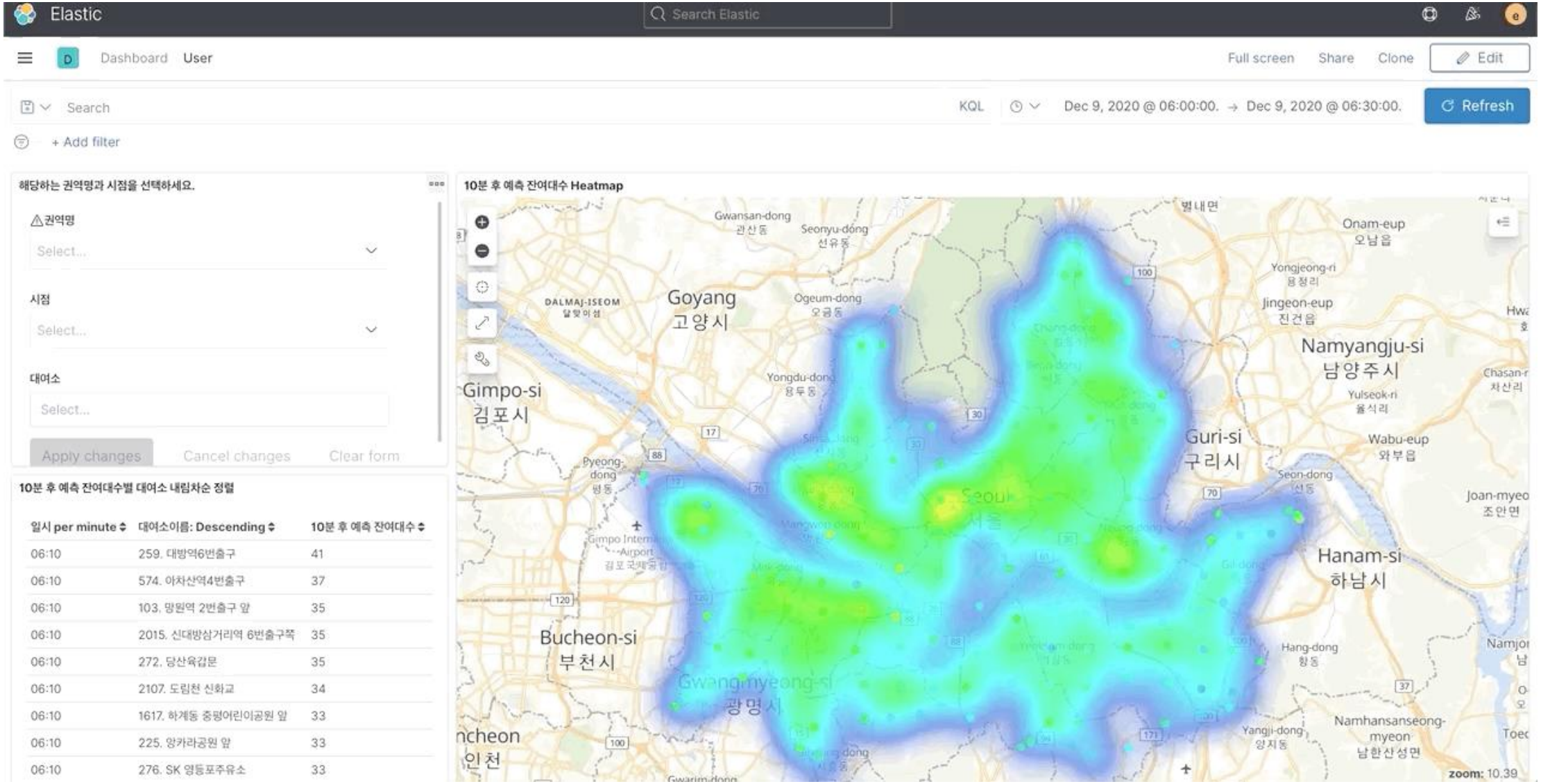
- **병렬 연산**과 배포 안정성 보장
- **빠르고 안정적으로 예측 결과 전달**

Kibana | 실시간 시각화 대시보드

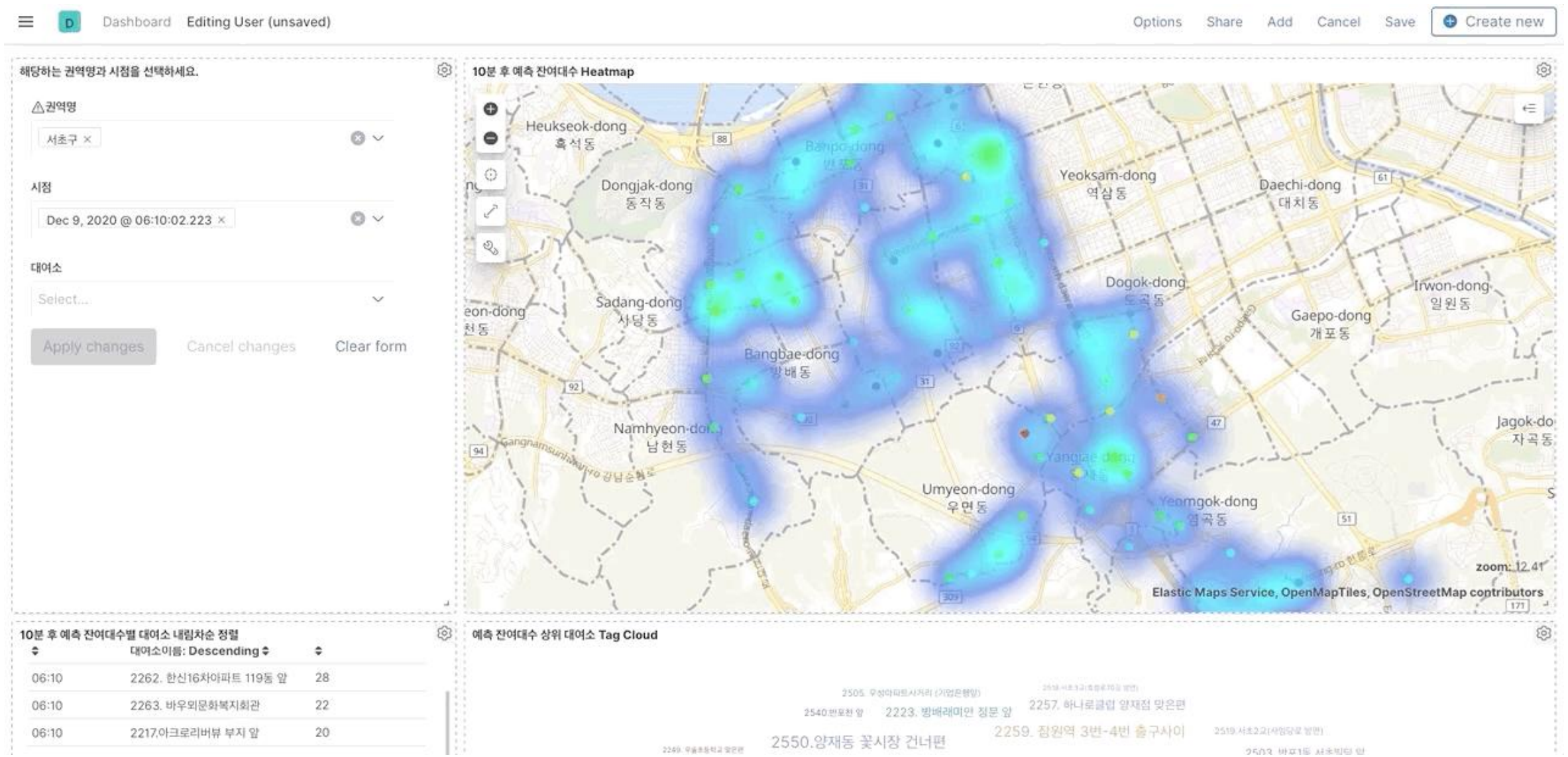




Kibana | 사용자 정보 입력



Kibana | 대시보드 구성



우리 서비스는!

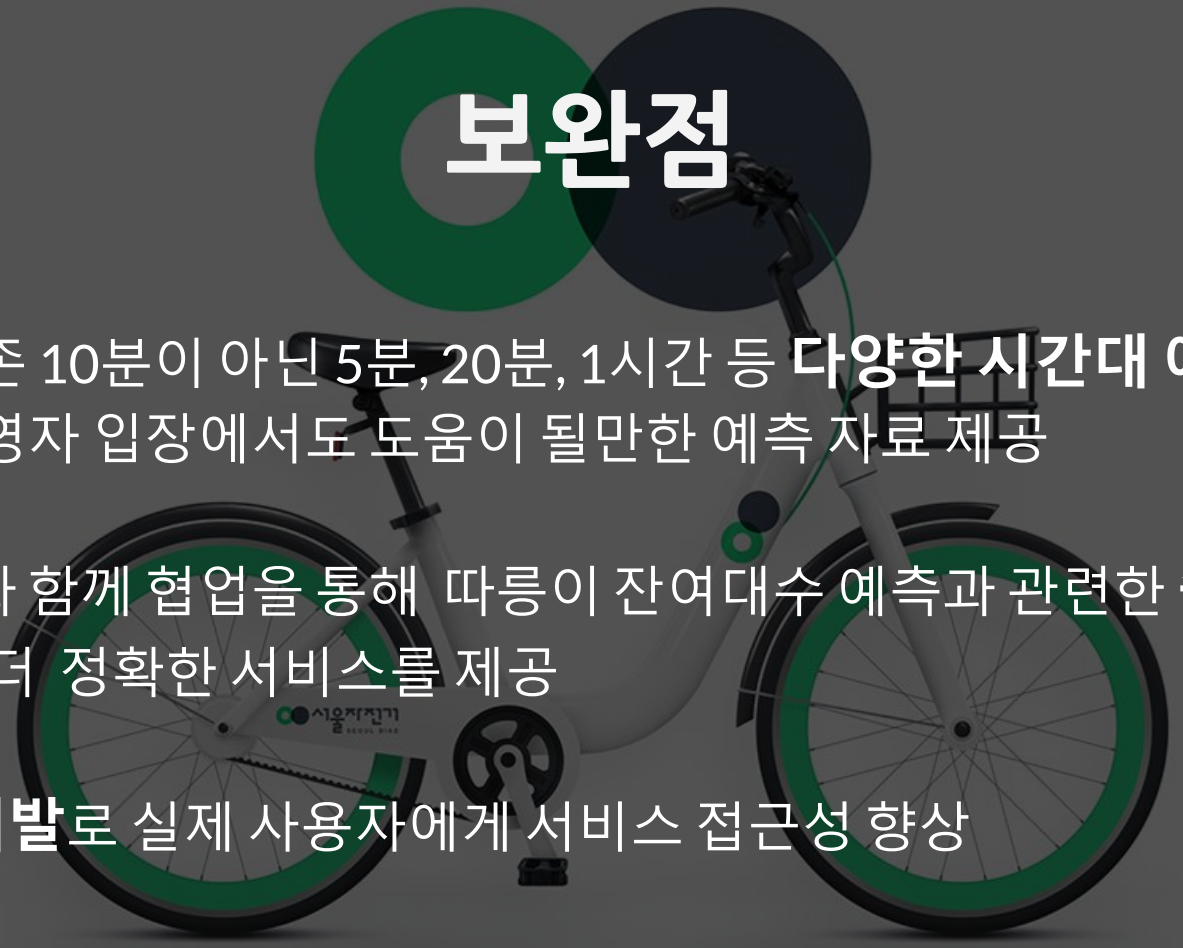
따릉이 사용자에게

빠르고 안정적인 배포가 가능한 Kubeflow와

실시간으로 특수한 상황까지 예측이 가능한 정확한 모델을 접목시켜

10분 뒤의 잔여대수를 제공하여, **따릉이 이용 불편을 해소**합니다.

보완점

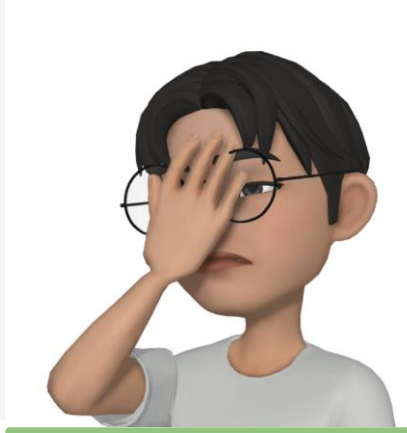


- 예측 시간을 기존 10분이 아닌 5분, 20분, 1시간 등 **다양한 시간대 예측**으로 사용자 뿐만 아니라 운영자 입장에서도 도움이 될만한 예측 자료 제공
- 따릉이 관계자와 함께 협업을 통해 따릉이 잔여대수 예측과 관련한 **중요변수를 추가**하여 더욱 더 정확한 서비스를 제공
- **Application 개발**로 실제 사용자에게 서비스 접근성 향상

CONTENTS

팀원 소개

박영민



- 데이터 수집
- 데이터 클렌징
- 데이터 전처리
- 모델 검증

김진세

- 데이터 수집
- 데이터 클렌징
- 데이터 전처리
- 데이터 시각화
- 예측 모델링

데이터 모델링

김인규

- AWS EKS 구축
- Kubeflow 구축
- ElasticSearch 구동
- Kibana 구동
- DB 자동화

양주화

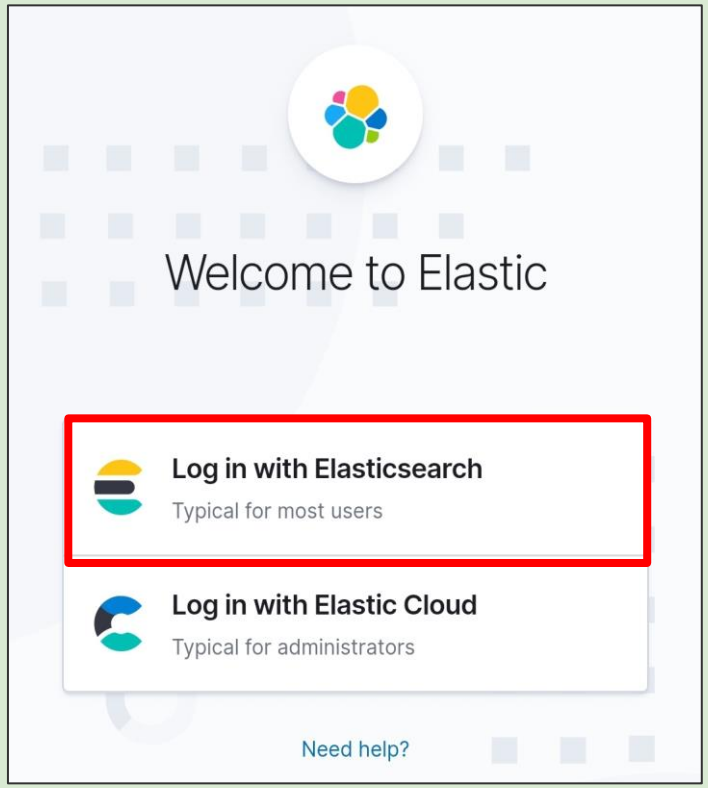
- AWS EKS 구축
- Kubeflow 구축
- AWS Aurora 구동
- DB 데이터 수집

인프라 구축

Q&A

SCANME.QR

ID : guest
PW : 123456



QR코드를 스캔하면 6조의 대시보드를 확인할 수 있습니다.

THANK YOU ALL!