

빅데이터 기반 AI활용 앱&웹 개발자 취업캠프

Paraway

지하철 혼잡도 예측

팀원 소개



팀장
정서윤



팀원
김은준



팀원
이창준



팀원
전감성



ParaWay

Paradise + Subway

현재 혹은 미래의 지하철 혼잡도를 예측 함으로써

사용자가 쾌적한 시간대의 지하철을 이용 할 수 있는

선택의 길에 도움을 주는 앱을 기획



CONTENTS

SECTION 01

프로젝트 개요 4~7p

- 1-1. 프로젝트 배경
- 1-2. 프로젝트 목적
- 1-3. Project Timeline

SECTION 02

EDA 8~17p

- 2-1. 데이터 수집 과정
- 2-2. 분석 범위 설정
- 2-3. 데이터 시각화

SECTION 03

데이터 분석과정 18~29p

- 3-1. 데이터 전처리
- 3-2. 모델 학습 / 예측 결과
- 3-3. 모델평가
- 3-4. 모델 함수 사용
- 3-5. 승차 인구수 시각화
- 3-6. 하차 인구수 시각화
- 3-7. 평일과 공휴일
- 3-8. 혼잡도 예측

SECTION 04

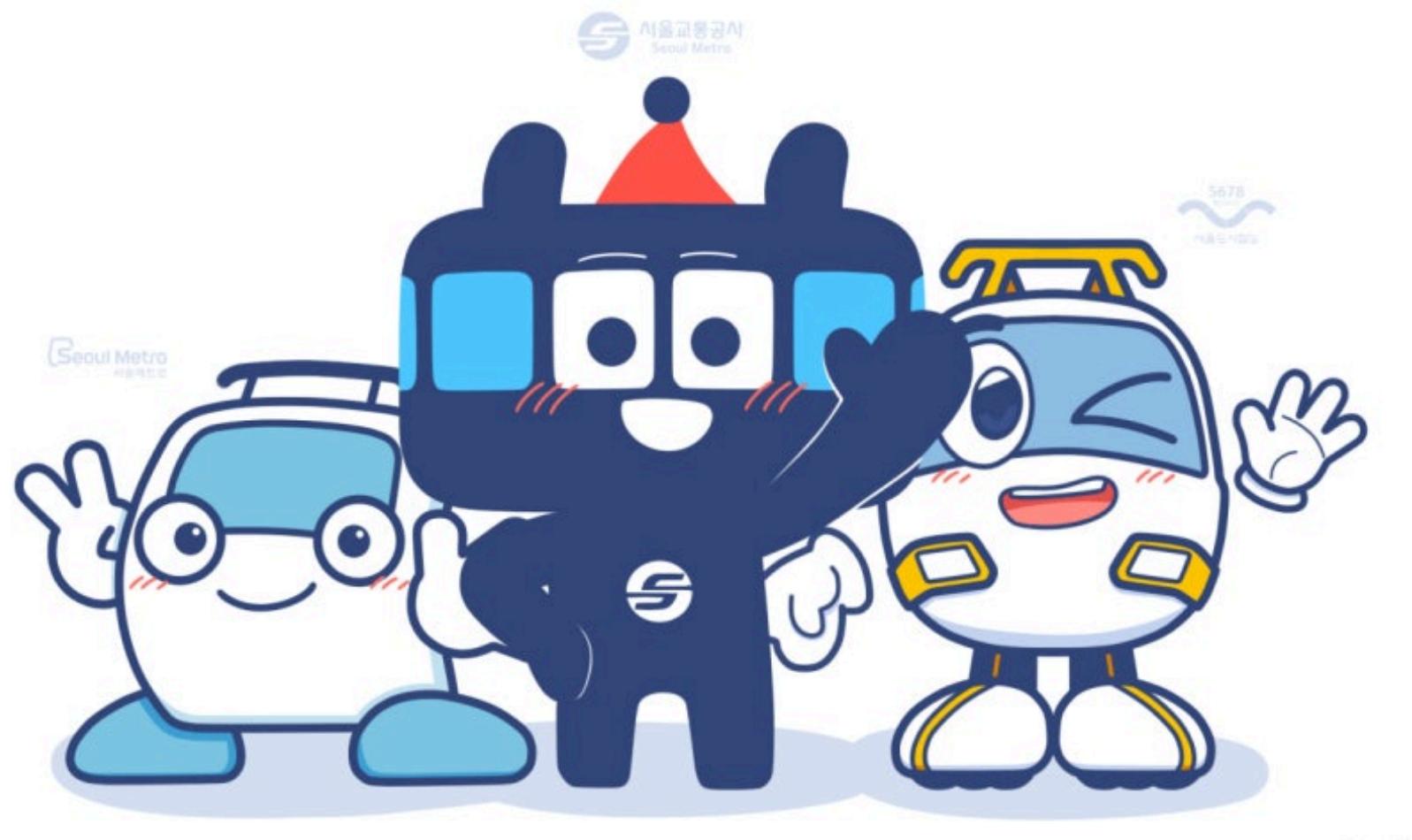
앱 개발과정

- 4-1. 개발 환경 및 기술 스택
- 4-2. 사용패키지
- 4-3. Git log
- 4-4. System Flow
- 4-5. 핵심기능과 주요기능
 - 혼잡도예측
 - 혼잡도상세페이지
 - 즐겨찾기, 뉴스
- 4-6. 자체평가



SECTION 01

프로젝트 개요



01

지하철 혼잡 문제인식

출퇴근 시간대 지하철 혼잡도는 매우 심각한 수준으로 혼잡한 시간대를 피하고 싶지만, 혼잡도에 대한 실시간 정보는 부족하고 예측하기 어렵다.

02

혼잡도 예측 필요성

혼잡도 예측이 가능하다면 사람들이 덜 봄비는 시간을 선택할 수 있다. 혼잡도 정보를 기반으로 시간을 최적화하고 또한 안전하게 운행 가능.

03

아이디어 도출

다양한 데이터를 활용하여 혼잡도 사전 예측할 수 있는 가능성이 커졌다. 이를 통해 머신러닝 기반 예측 모델을 만들 수 있을 것이라고 판단.



지하철 승차 인구 예측 모델 구축

지하철 하차 인구 예측 모델 구축

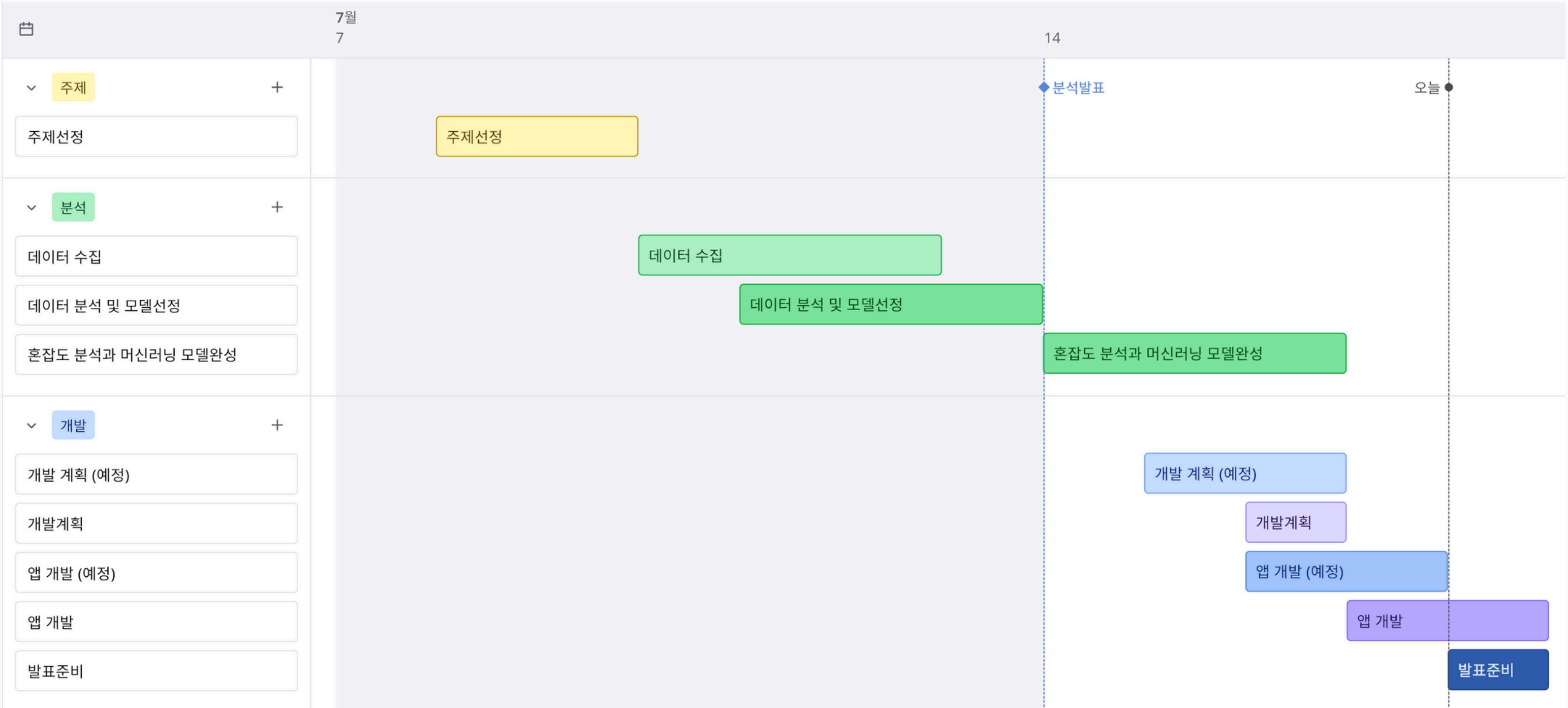
지하철 혼잡도 예측 모델 구축

지하철 혼잡도를 **시간대·역별**로 예측하는 모델을 구축하여,
사용자에게 더 나은 이동 선택지를 제공하는 것을 목표로 합니다.



Project Timeline

1-3



SECTION 02

EDA

서울 열린 데이터 광장

지하철 혼잡도 정보

지하철 호선별 역별 승하차 인원 정보

서울시 상권분석 서비스(직장인구)

서울시 상권분석 서비스(점포)

서울교통공사_지하철혼잡도정보

서울교통공사 1-8호선 30분 단위 평균 혼잡도로 30분간 지나는 열차들의 평균 혼잡도(정원대비 승차인원으로, 승차인과 좌석수가 일치할 경우를 혼잡도 34%로 산정)입니다.(단위: %).

서울교통공사 혼잡도 데이터는 요일구분(평일, 토요일, 일요일), 호선, 역번호, 역명, 상하선구분, 30분단위 별 혼잡도 데이터로 구성되어 있습니다. (2024년부터 분기별 제공됩니다.)

서울시 지하철호선별 역별 승하차 인원 정보

교통카드(선후불교통카드 및 1회용 교통카드)를 이용한 지하철호선별 역별(서울교통공사, 한국철도공사, 공항철도, 9호선) 승하차인원을 나타내는 정보입니다. (일단위)

※ Sheet 서비스는 마지막 한달치 데이터만 서비스 합니다. (* 데이터 적재는 매일 3일전 데이터를 갱신합니다.)

-
-
-



기상청 기상자료개방포털

기간을 정하여 일별데이터의 강수량을 csv파일로 받음

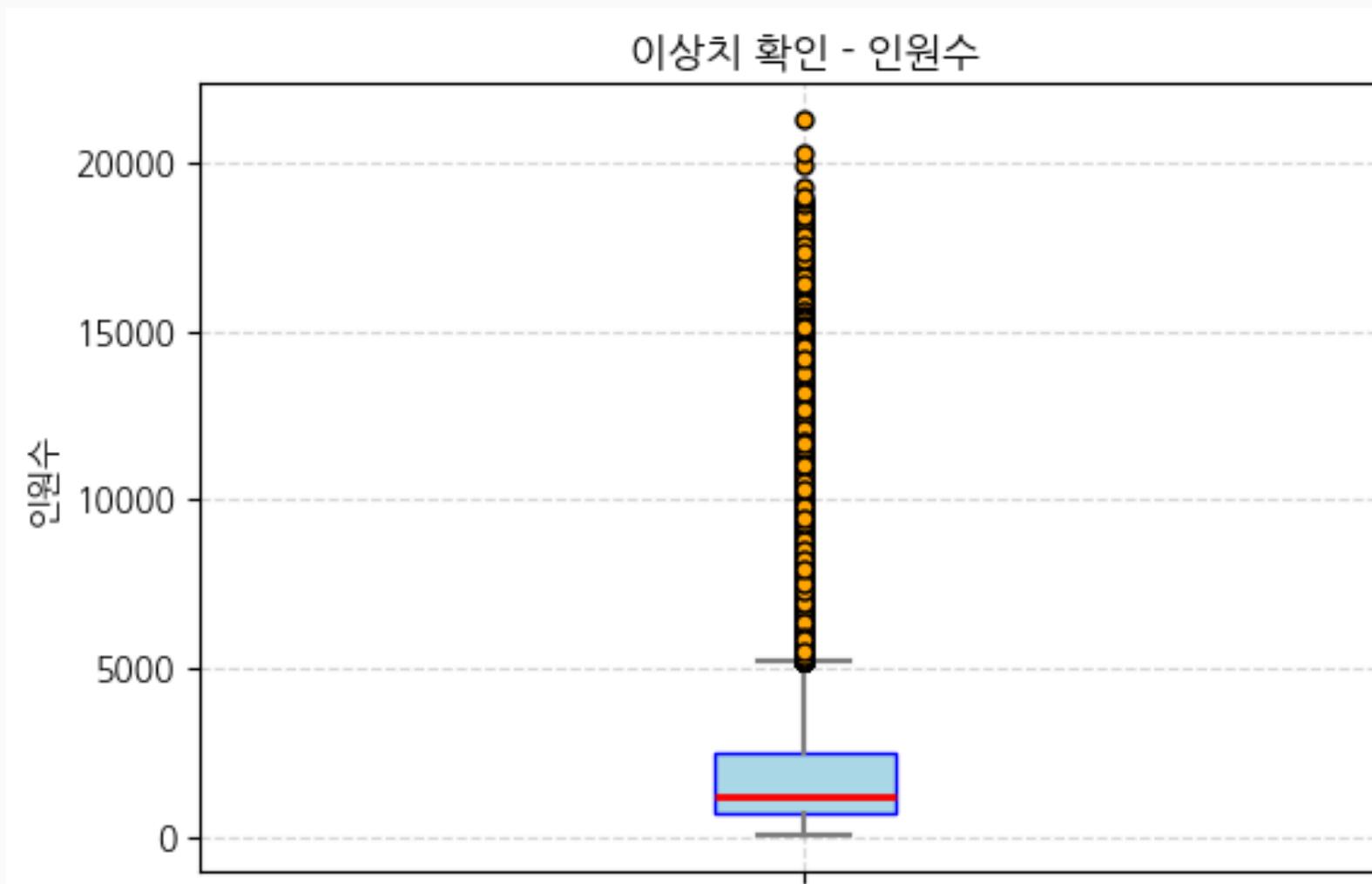
The screenshot shows the Korea Meteorological Administration Open Data Portal. At the top, there is a search bar with the placeholder '관측' (Observation) and a magnifying glass icon. Below the search bar, there are several navigation tabs: 데이터 (Data), API, 기후통계분석 (Climate Statistics Analysis), 간행물 (Published Materials), 소통과 참여 (Communication and Participation), and 기상현상증명 (Weather Phenomenon Certification). The '기후통계분석' tab is active. On the left side, there is a sidebar with various analysis options: 평년값 (Average Value), 통계분석 (Statistics Analysis), 조건별통계 (Conditional Statistics), 기온분석 (Temperature Analysis), 강수량분석 (Precipitation Analysis), 다중지점통계 (Multi-point Statistics), 24절기 (24 Solar Terms), 순위값 (Ranking Values), 장마 (Plum Rain), and 기상현상일수 (Weather Phenomenon Days). The '통계분석' option is currently selected. The main content area displays the '강수량분석 - 그래프' (Precipitation Analysis - Graph) section. It includes a graph tab and a map tab (선포도). Below the tabs, there is a '자료설명' (Data Description) section with information about regional precipitation analysis. It mentions that regional precipitation analysis is provided for 62 regions across the country, with 4 regions being Jeju Island. There is also a note about the use of national and regional average values from 1973. The '검색조건' (Search Conditions) section allows users to specify parameters such as data type (일, 기본), date range (20250613 ~ 20250712), and location (서울). A large green arrow points to the right at the bottom right corner of the page.

이용객이 가장 많은 2호선 라인중 분석범위선정

2호선 사당 - 잠실나루까지의(12개역) 범위를 정하고
이역들을 기준으로 혼잡도를 분석



이상치 확인

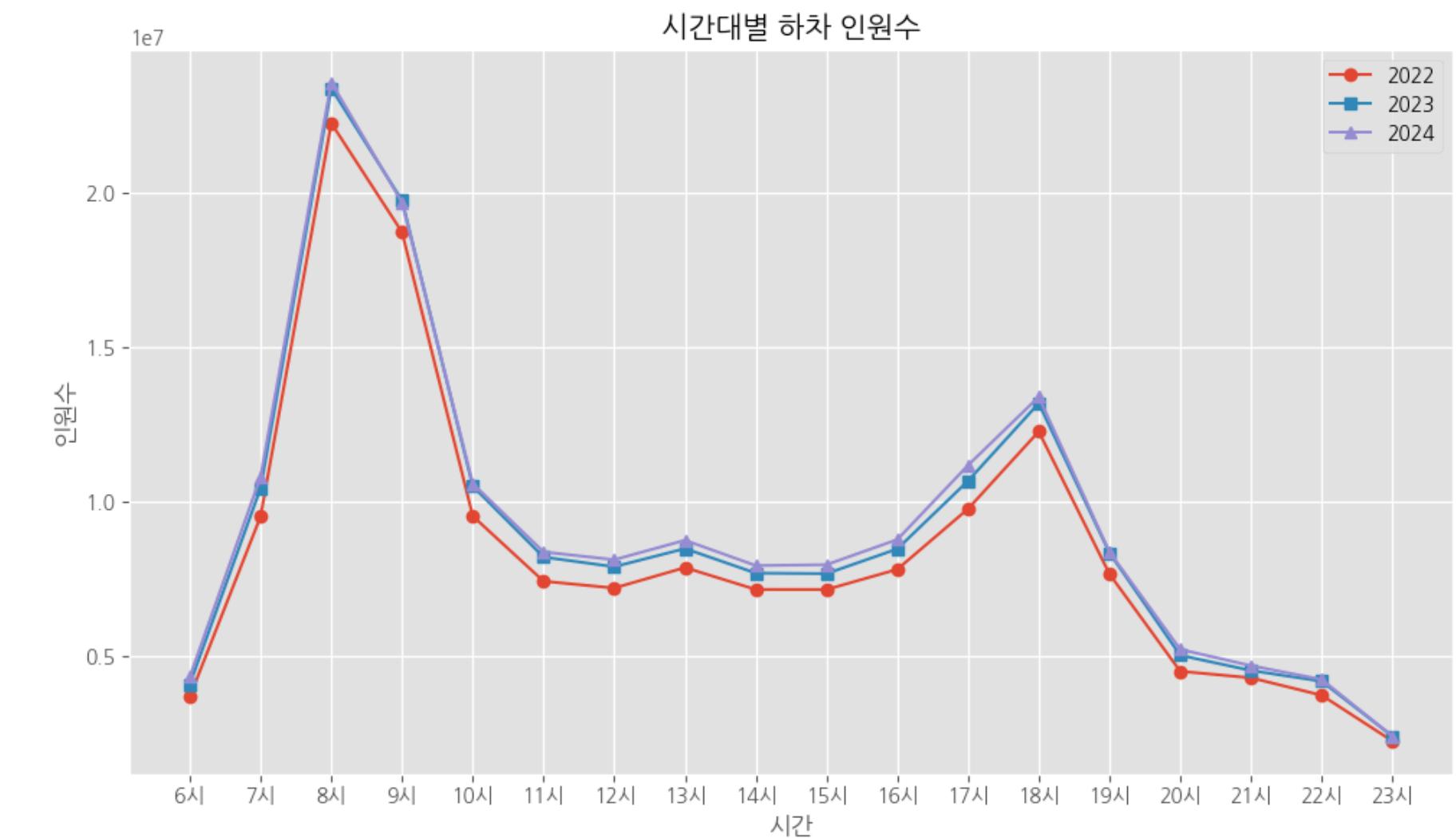
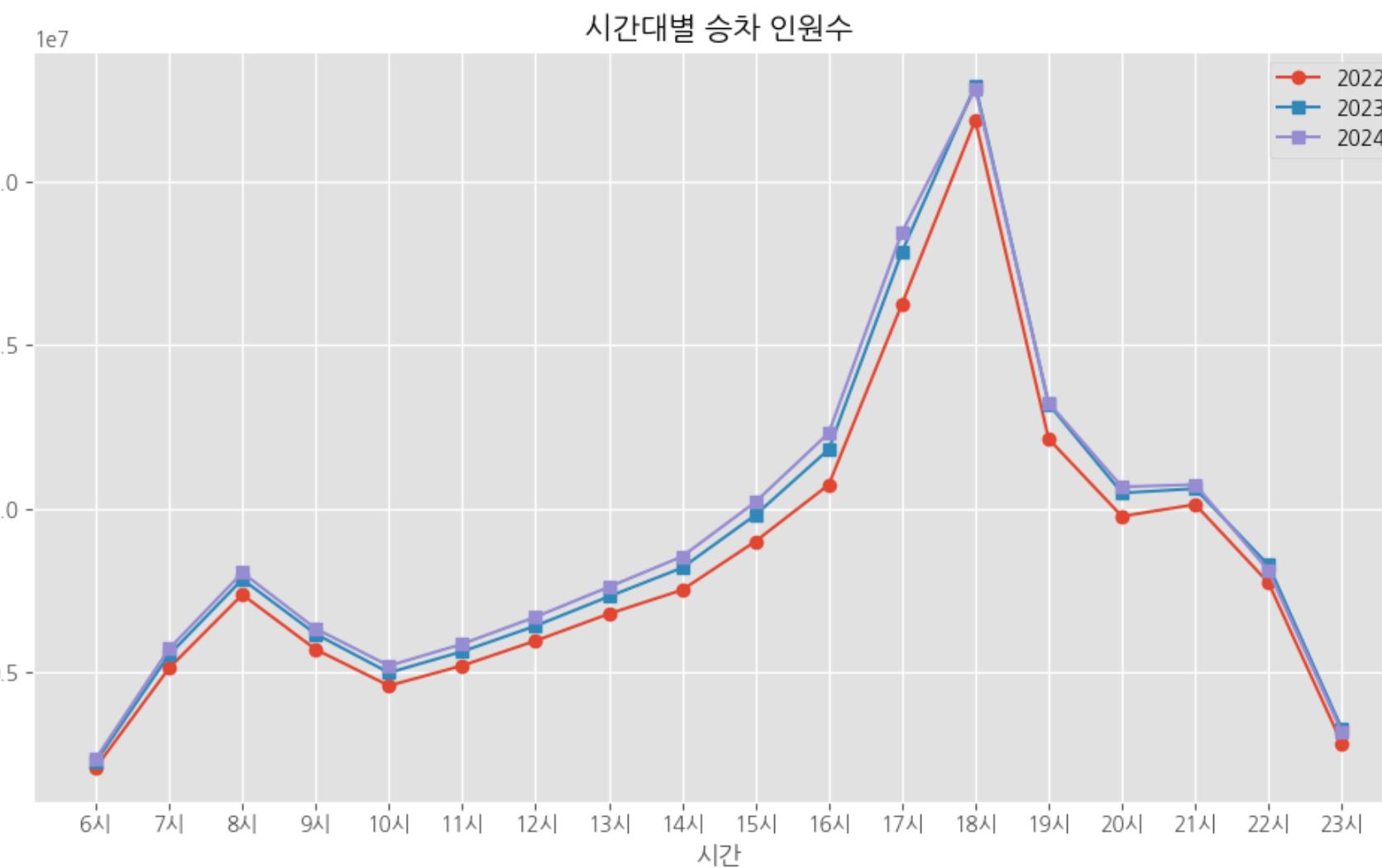


총 데이터 개수 : 473,472

승하차 인구수 이상치 확인하고
출근시간과 퇴근시간 분류함



연도 별, 시간대 별 승하차 인원 수



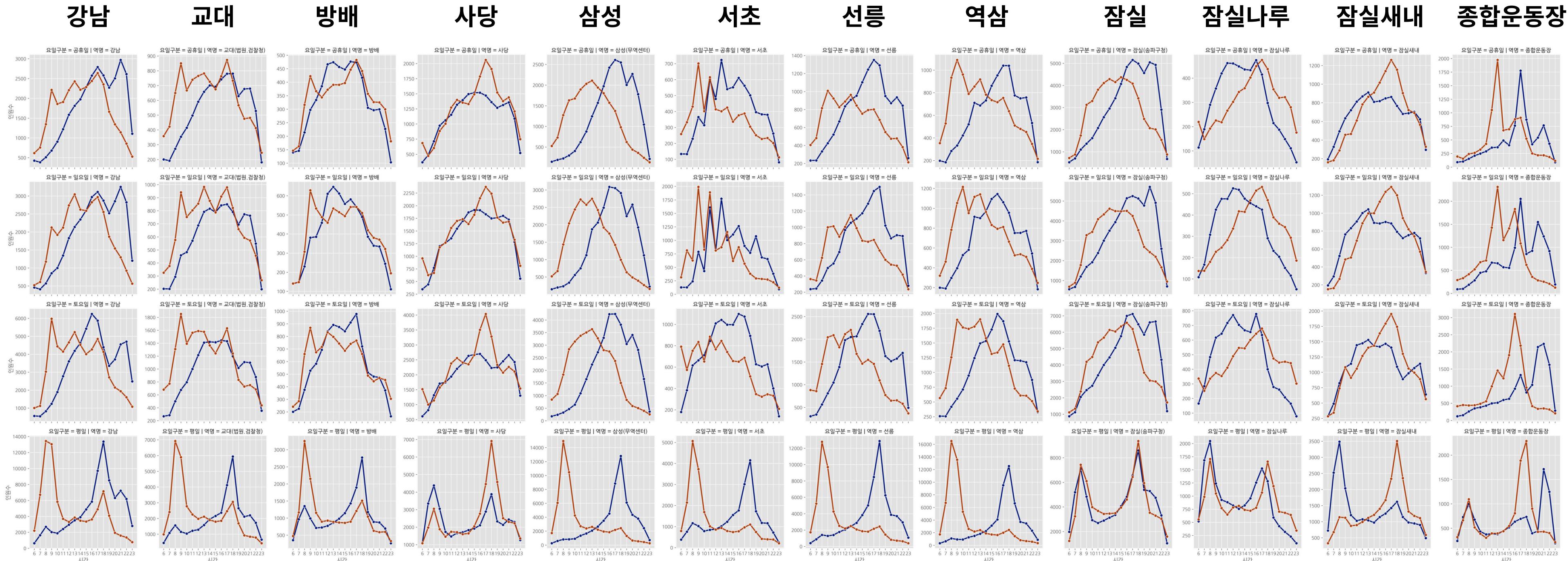
승차인원 차이의 최대값 : 17시, 2,196,489
승차인원 차이의 최소값 : 6시, 272,703

하차인원 차이의 최대값 : 17시, 1,412,886
하차인원 차이의 최소값 : 23시, 157,391

2023년 10월 7일 교통카드 기준 1,250원 → 1,400원 (150원 인상)



시간대 별, 각 역의 승하차 인원 변화 양상



구분

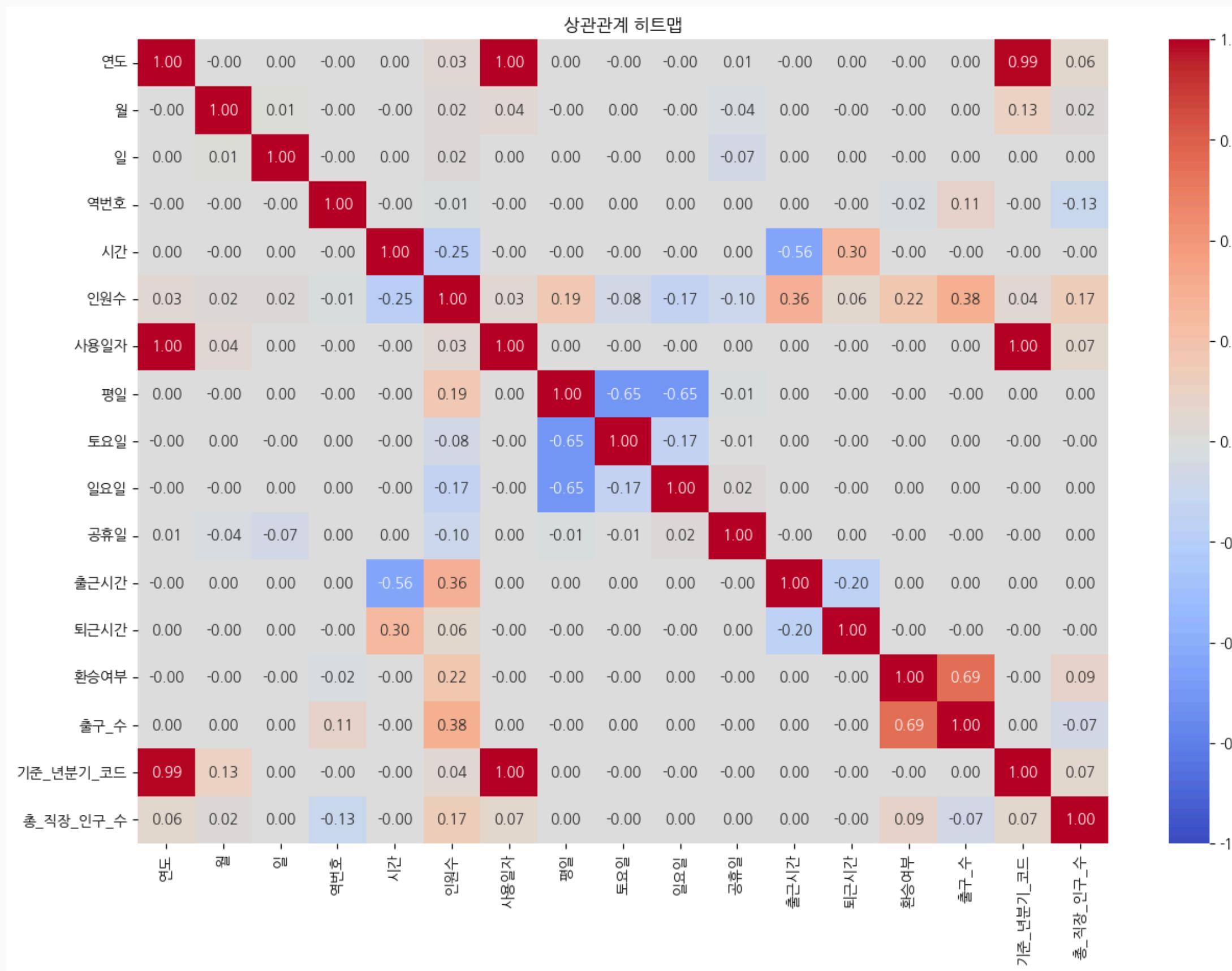
- 승차
- 하차

요일 구분 : 공휴일, 일요일, 토요일, 평일 순서



데이터 시각화

상관관계



인원수 1.000000

출구_수 0.424607

퇴근시간 0.400408

환승여부 0.266855

시간 0.216812

평일 0.202091

총_직장_인구_수 0.187828

기준_년분기_코드 0.037506

사용일자 0.035418

연도 0.034380

월 0.024828

일 0.017607

역번호 -0.035603

토요일 -0.087732

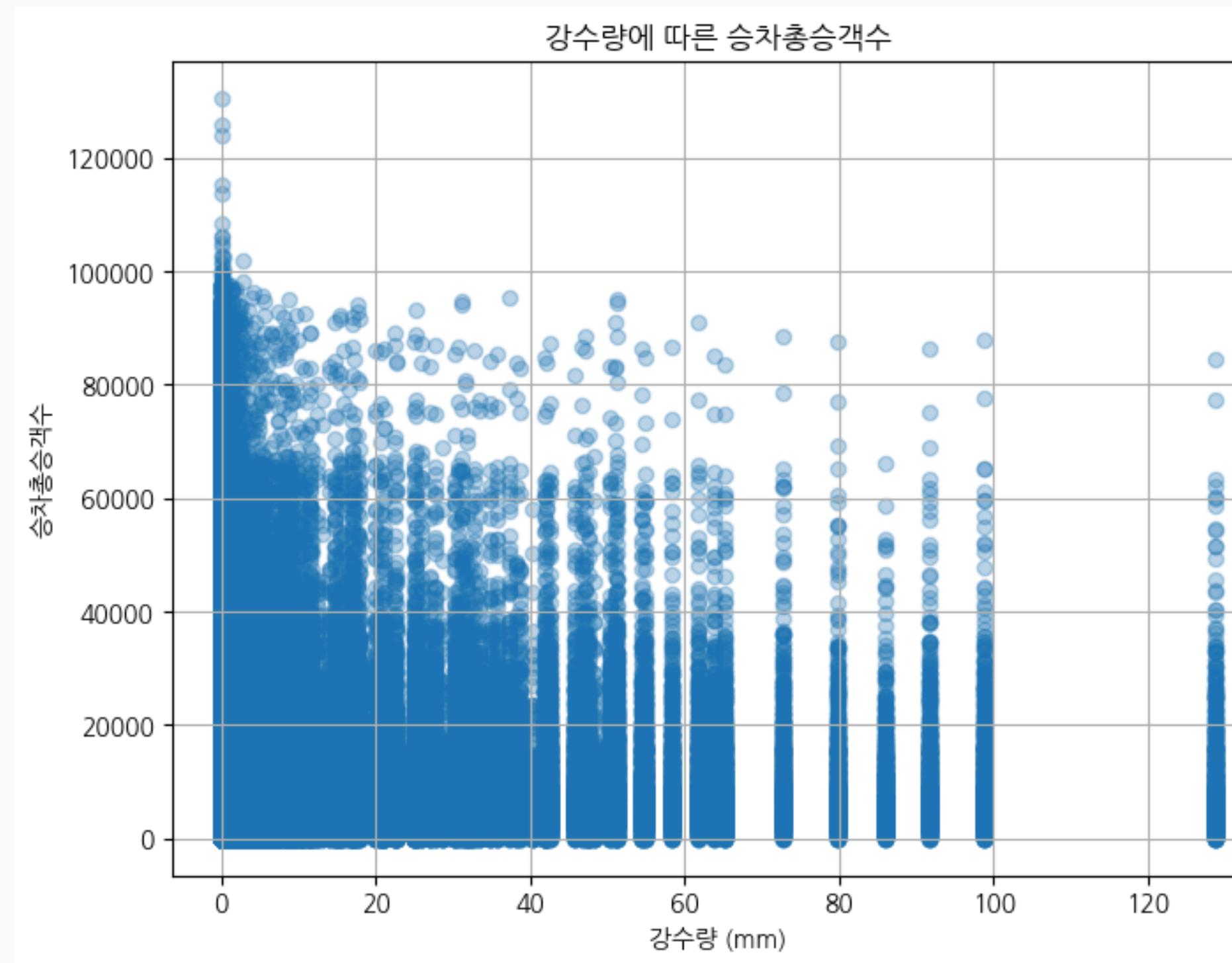
공휴일 -0.109096

출근시간 -0.112084

일요일 -0.173083

Name: 인원수, dtype: float64



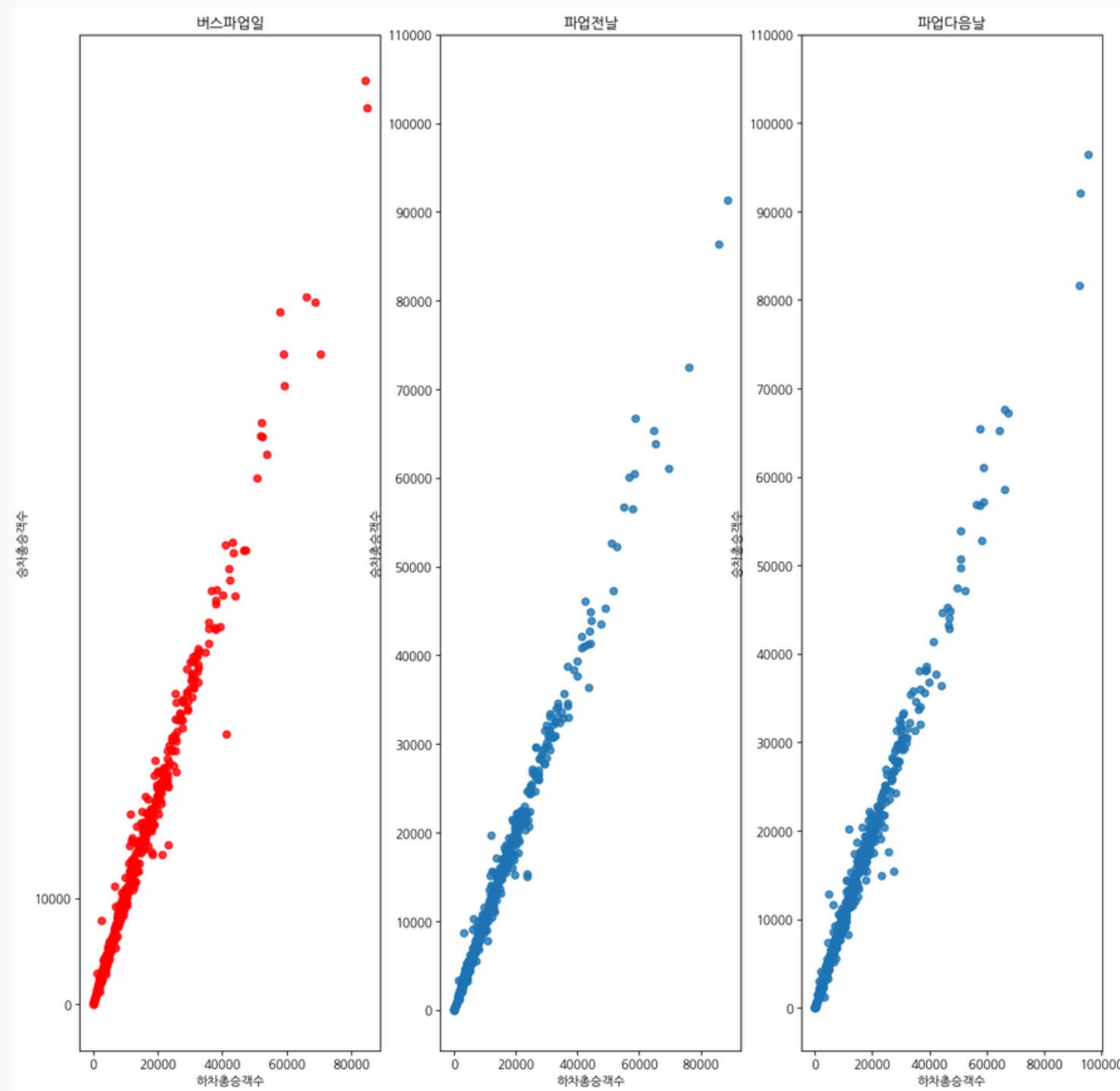


데이터 부적합

강수량이 증가하면 그에따라 승차총승객수도 증가할 것을 예상하고 산점도를 확인해 보았는데 강수량이 0.0(mm)인 경우가 대다수였고 강수량이 증가하여도 승객수 변화는 거의 없었음.



데이터 시각화



버스파업일 최대승차인원: 189795
20240327 최대 승차인원: 179903
20240329 최대 승차인원: 191790

버스파업일

버스파업일이 24년 3월 28일 데이터가있어 전일,다음날 기준으로 데이터를 봤을때 버스파업시간에따라 이상치가보이나 총 이용객수는 변함이없습니다.

결론

버스파업일 당일에 전체 이용객 수에는 큰 변화가 없었지만, 파업시간대에 맞춰 특정시간대와 구간에 승객이 집중 되는 경향이 나타났습니다. 즉, 버스파업이 총수요를 증가시키기보다는 기존 수요를 특정시간과 노선으로 몰리게하는 분산,집중효과를 유발한것으로 판단됩니다.

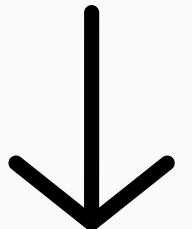


SECTION 03

데이터 분석

데이터 전처리

날짜	호선	역 번호	역명	구분	06:00 이전	06:00 ~ 07:00	07:00 ~ 08:00	08:00 ~ 09:00	09:00 ~ 10:00	...	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	24:00 이후
2020-01-01	1호선	150	서울역	승차	356	280	313	774	923	...	2972	3194	3342	3002	2857	2311	2523	1830	987	25
연번	날짜	호선	역번호	역명	구분	06시 이전	06시-07시	07시-08시	08시-09시	...	15시-16시	16시-17시	17시-18시	18시-19시	19시-20시	20시-21시	21시-22시	22시-23시	23시 이후	합계
1	2021-01-01	1	150	서울역	승차	86	111	157	306	...	837	796	839	658	579	479	510	445	101	9339



연도	월	일	역번호	역명	구분	시간	인원수	평일	토요일	일요일	공휴일	출근시간	퇴근시간
2022	1	1	215	잠실나루	승차	6	62	False	True	False	True	False	False

연도 별 데이터의 데이터 속성은 일치하지만 컬럼명이 다른 경우 통합하여 연도별 데이터를 하나의 데이터 프레임으로 합침

column 삭제

- 06:00 이전
- 24:00 이후

호선이 여러 개인 경우 2호선의 역 번호 data만 사용

분리

- 날짜 → 연도, 월, 일
- 시간 → melt 함수를 이용하여 column -> row

추가

- 평일, 토요일, 일요일, 공휴일 column
- 출, 퇴근 시간 column



Feature

연도, 월, 일, 역번호, 시간, 사용일자,
평일, 토요일, 일요일, 공휴일,
출근시간, 퇴근시간, 환승여부, 출구수,
기준_년분기_코드, 총_직장_인구_수

Target

인원수

연도	월	일	역번호	역명	구분	시간	인원수	사용일자	평일	...	일요일	공휴일	출근시간	퇴근시간	연도분기	점포수	환승여부	출구수	총_직장_인구_수	지하철간격
2022	8	1	222	강남	승차	6	588	20220801	True	...	False	False	False	False	20223	18550	True	12	83897	7
2022	8	1	222	강남	하차	6	2056	20220801	True	...	False	False	False	False	20223	18550	True	12	83897	7

승차인구수 모델 예측 결과

RandomForest
Train R² Score : 0.9945
Test R² Score : 0.9555
MSE : 221905.85
MAE : 184.17

XGBoost
Train R² Score : 0.9694
Test R² Score : 0.9608
MSE : 195780.05
MAE : 196.39

Linear
Train R² Score : 0.4673
Test R² Score : 0.4634
MSE : 2677683.77
MAE : 1117.25

GradientBoosting
Train R² Score : 0.8465
Test R² Score : 0.8374
MSE : 811217.02
MAE : 524.77

하차인구수 모델 예측 결과

RandomForest
Train R² Score : 0.9957
Test R² Score : 0.9703
MSE : 175654.57
MAE : 175.43

XGBoost
Train R² Score : 0.9766
Test R² Score : 0.9733
MSE : 158130.11
MAE : 196.81

Linear
Train R² Score : 0.4067
Test R² Score : 0.4076
MSE : 3505885.26
MAE : 1165.86

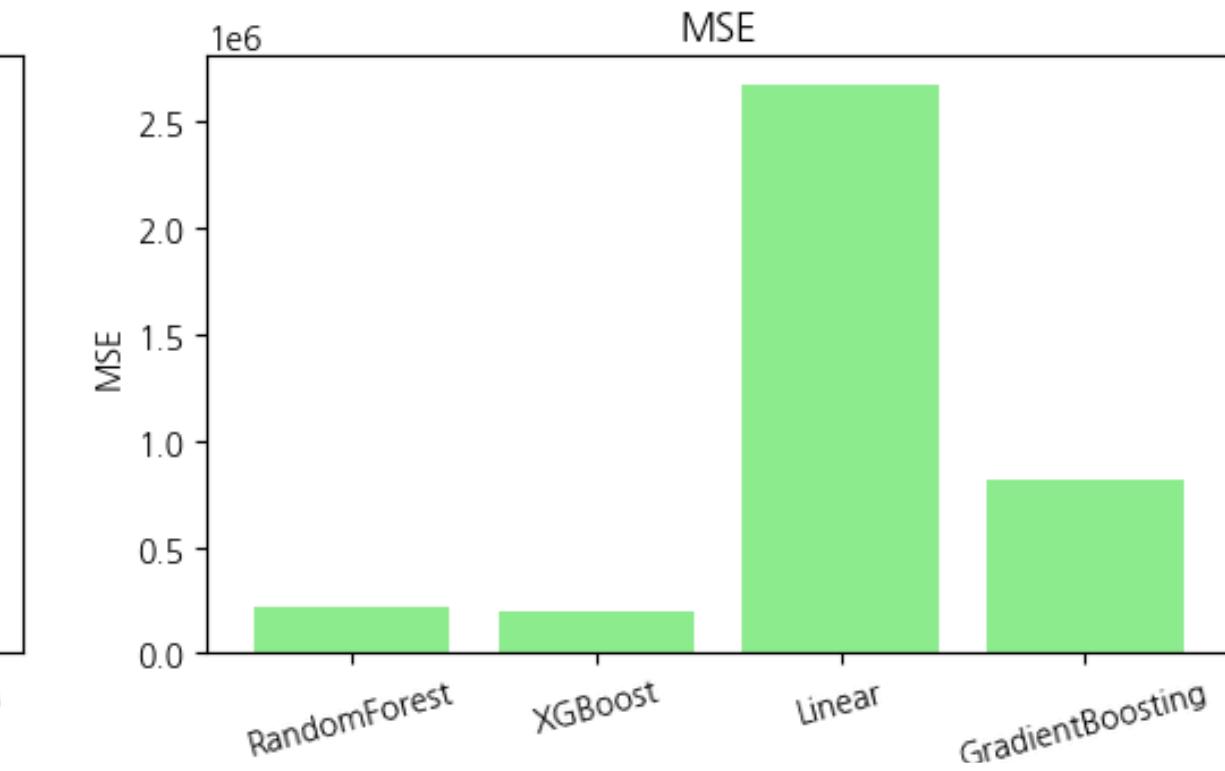
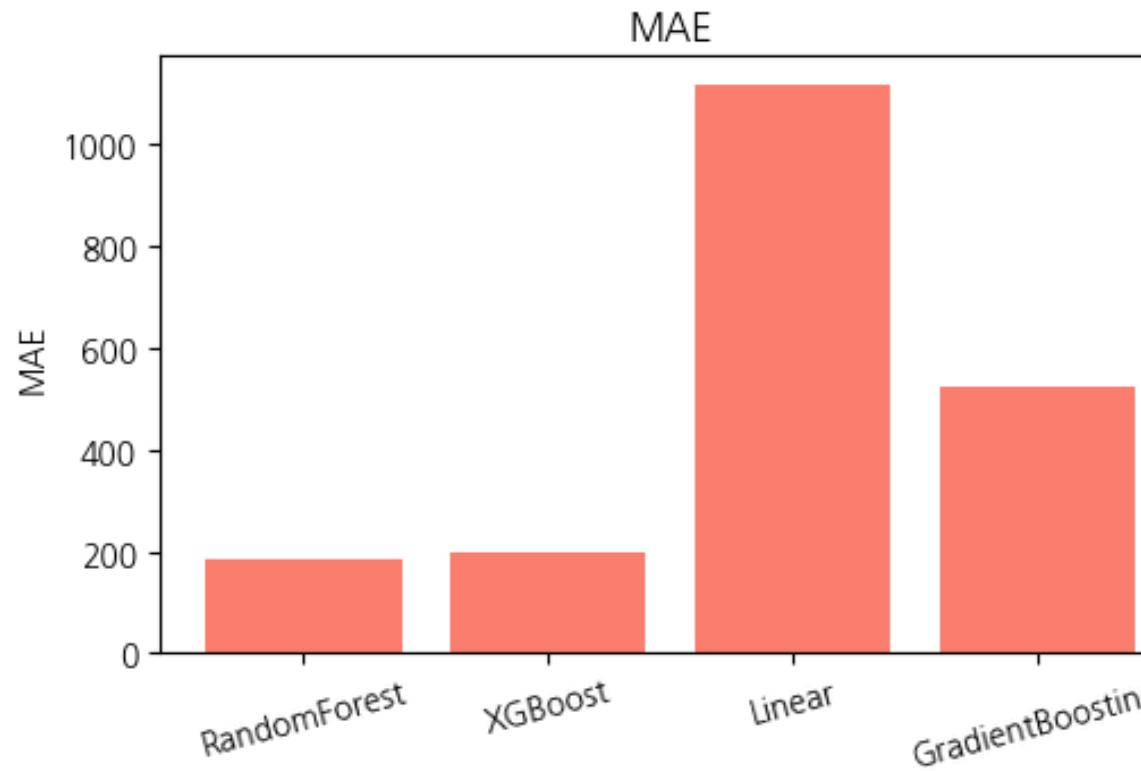
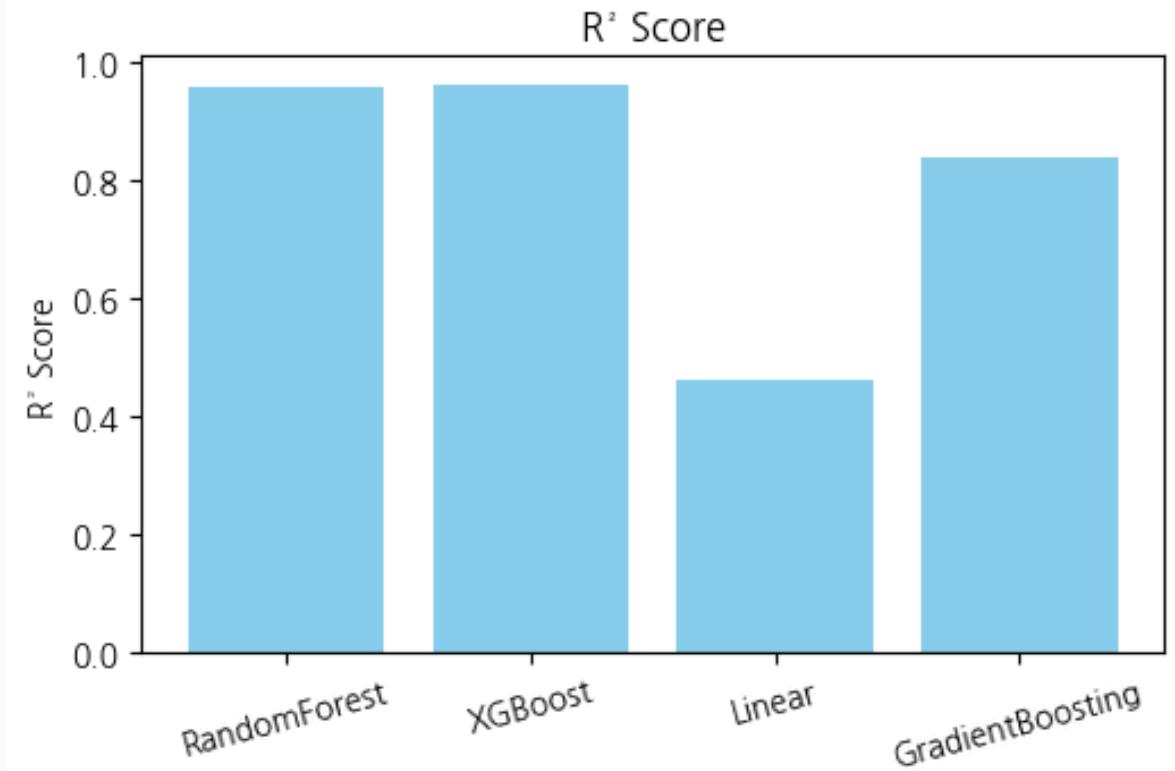
GradientBoosting
Train R² Score : 0.8494
Test R² Score : 0.8460
MSE : 911383.80
MAE : 540.41



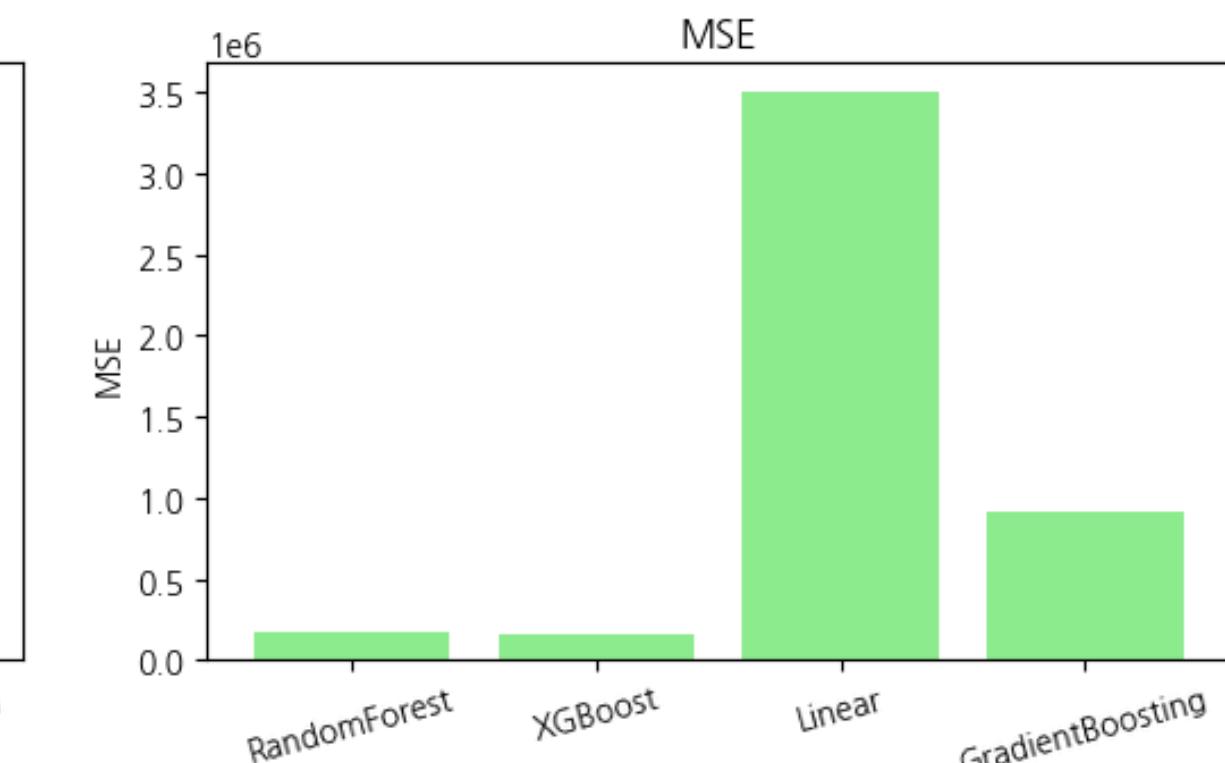
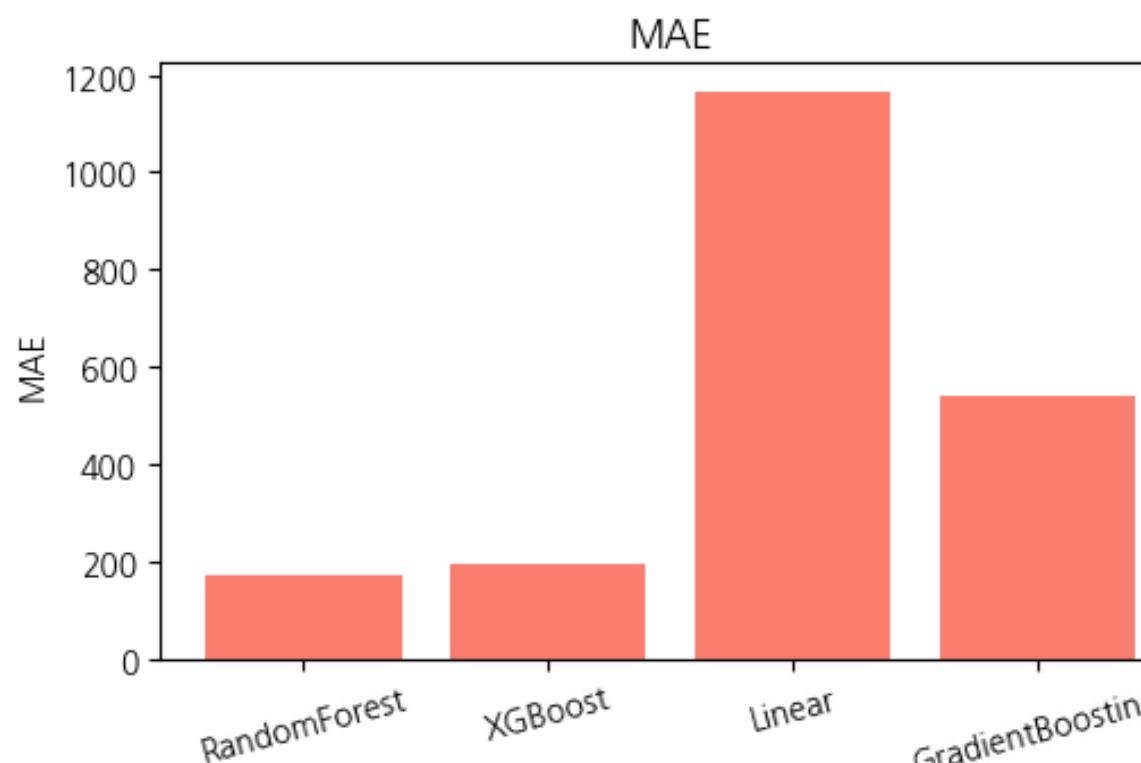
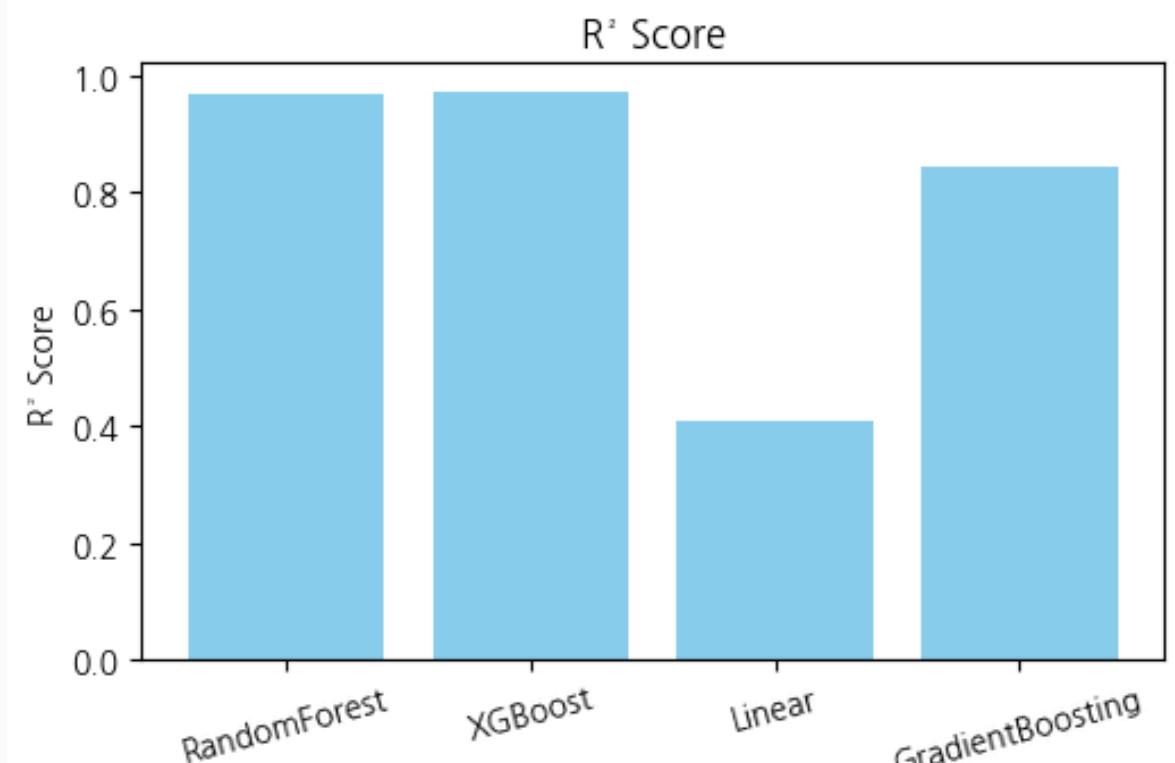
모델 평가

3-3

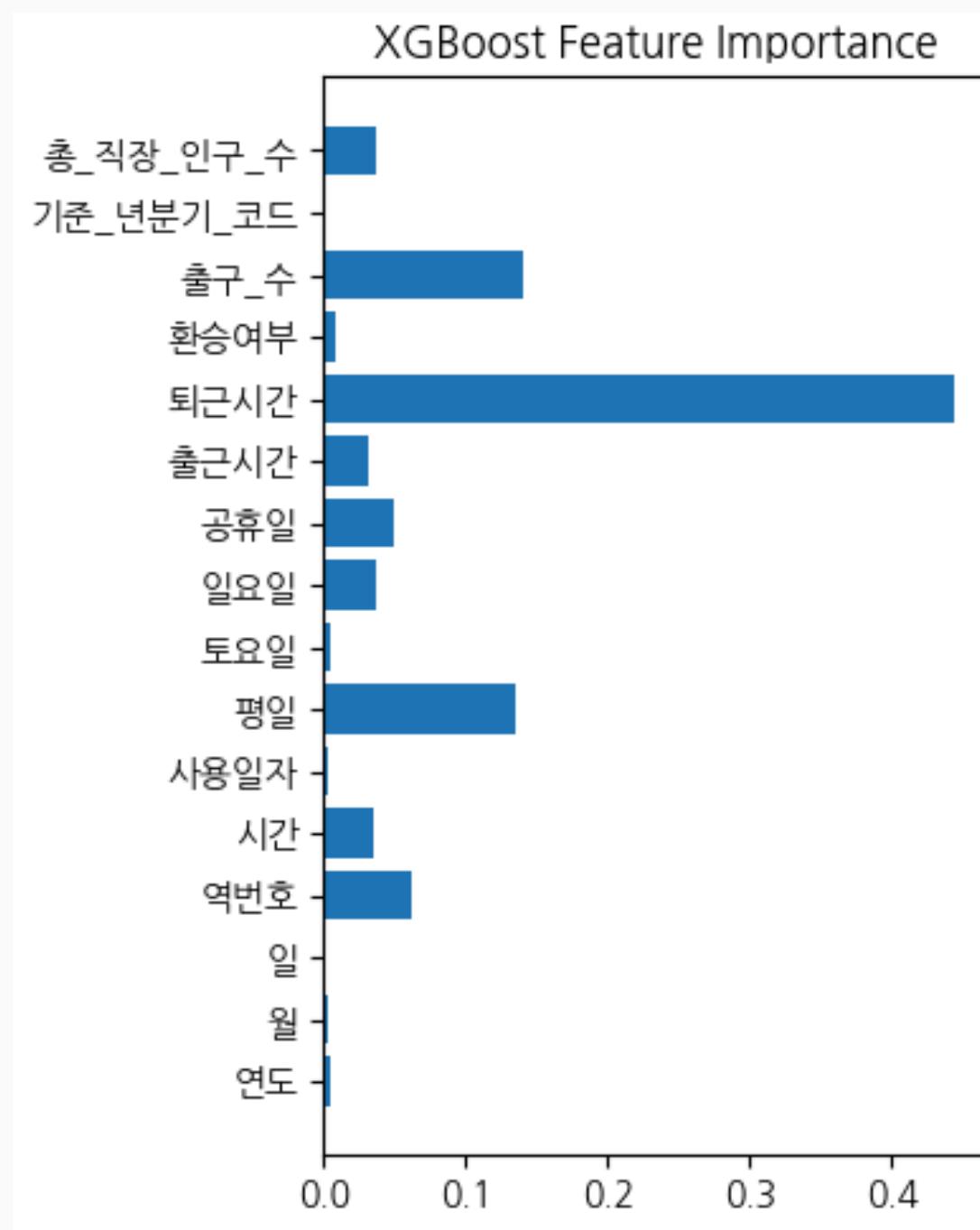
모델별 성능 비교



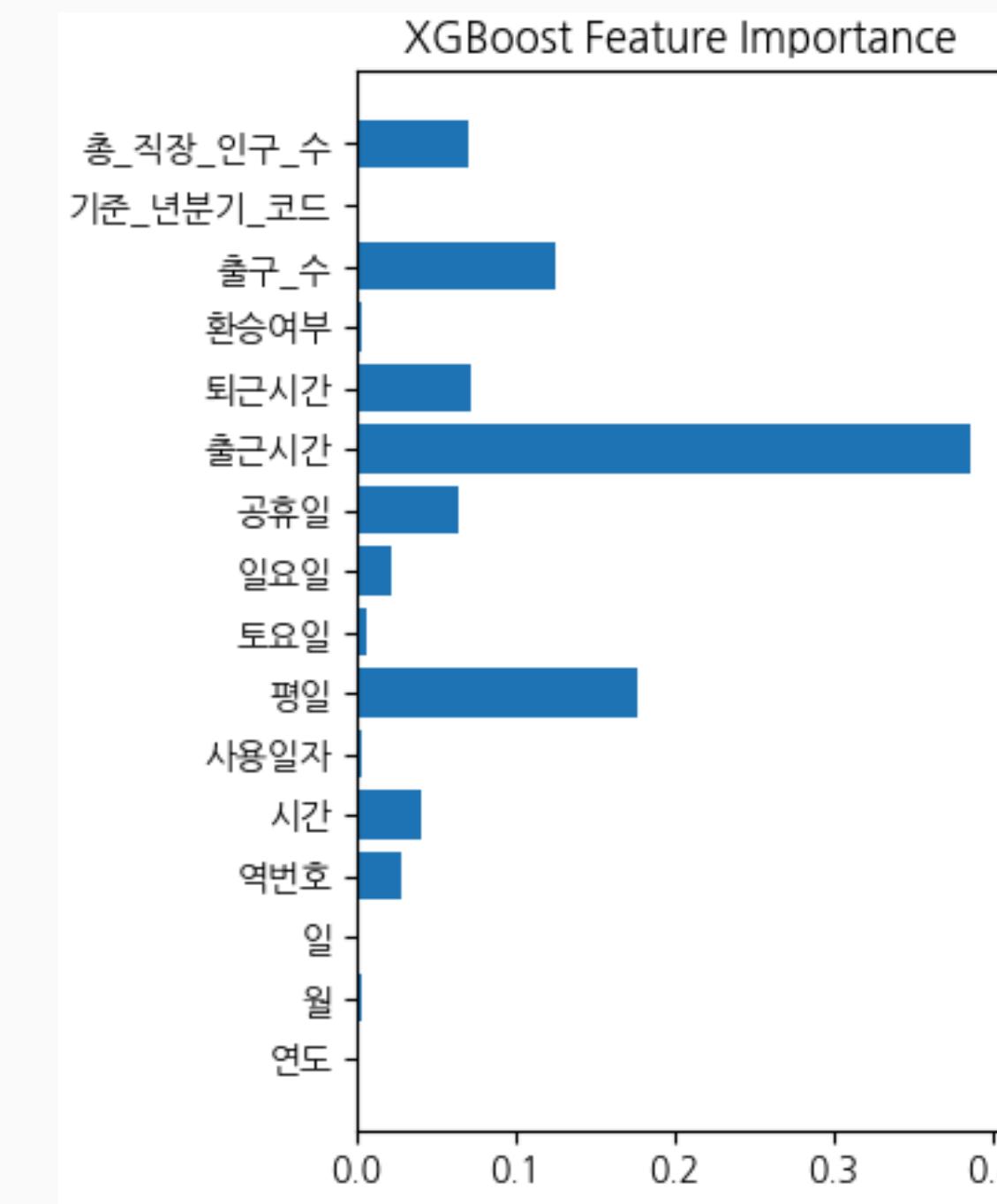
하차인구수 예측



승차인구수 데이터셋 Feature 중요도



하차인구수 데이터셋 Feature 중요도



승차인구수 예측

```
data = subway_on[['연도', '월', '일', '역번호', '시간', '사용일자', '평일', '토요일',
                  '공휴일', '출근시간', '퇴근시간', '점포수', '환승여부', '출구_수',
                  '총_직장_인구_수', '지하철간격']]
target = subway_on.인원수
machine.learning(data=data, target=target, model=XGBRegressor, dept=7)
```

✓ 6m 13.0s

Python

```
train_valid train : 0.9895948171615601 train_valid valid : 0.9707696199417114
train : 0.9882051944732666
test : 0.9495148062705994
```

모델 함수를 만들어서 GridSearchCV로
최적의 파라미터를 찾아 학습하여
예측값을 출력

하차인구수 예측

```
data = subway_off[['연도', '월', '일', '역번호', '시간', '사용일자', '평일', '토요일',
                   '공휴일', '출근시간', '퇴근시간', '점포수', '환승여부', '출구_수',
                   '총_직장_인구_수', '지하철간격']]
target = subway_off.인원수
machine.learning(data=data, target=target, model=XGBRegressor, dept=7)
```

✓ 7m 47.4s

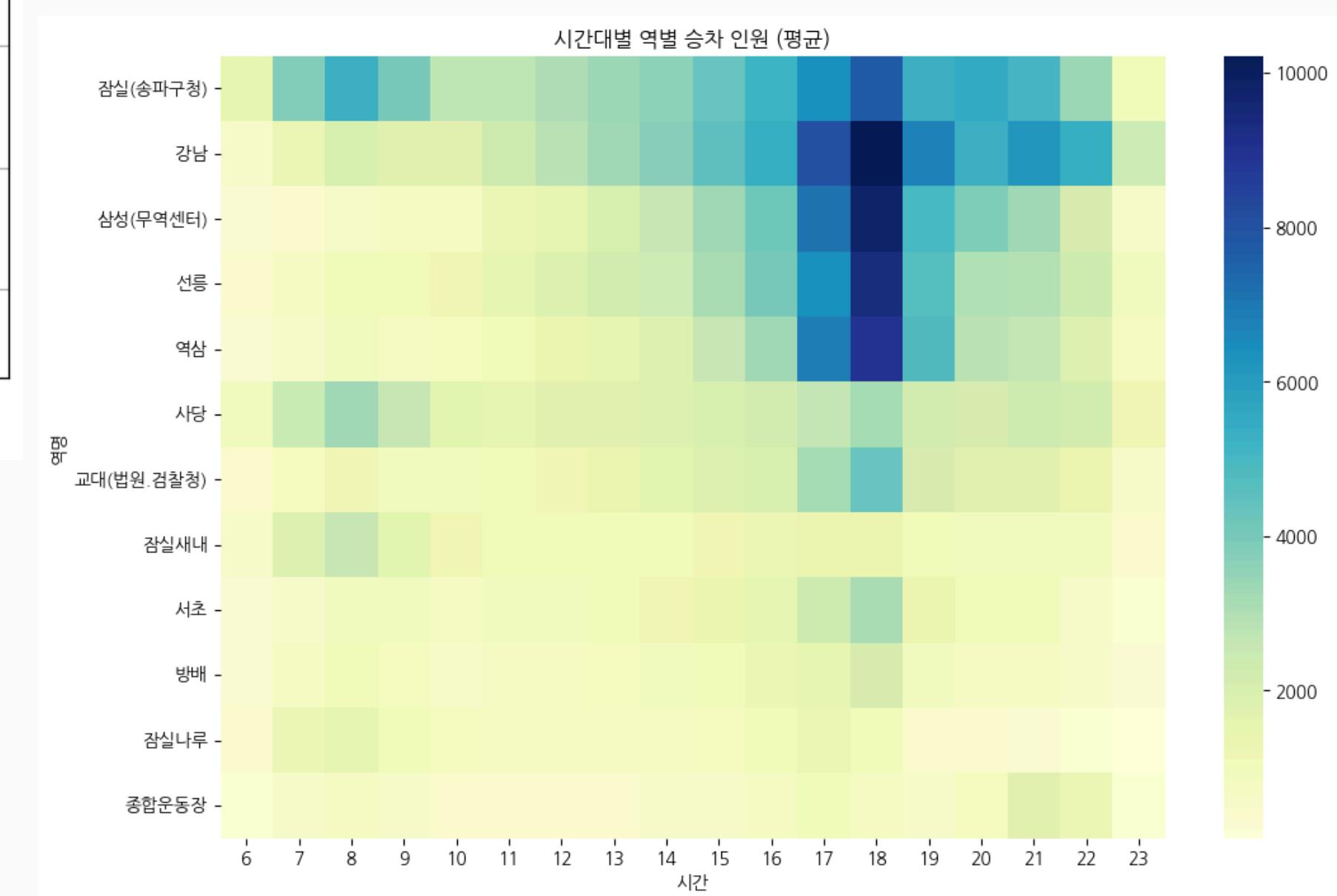
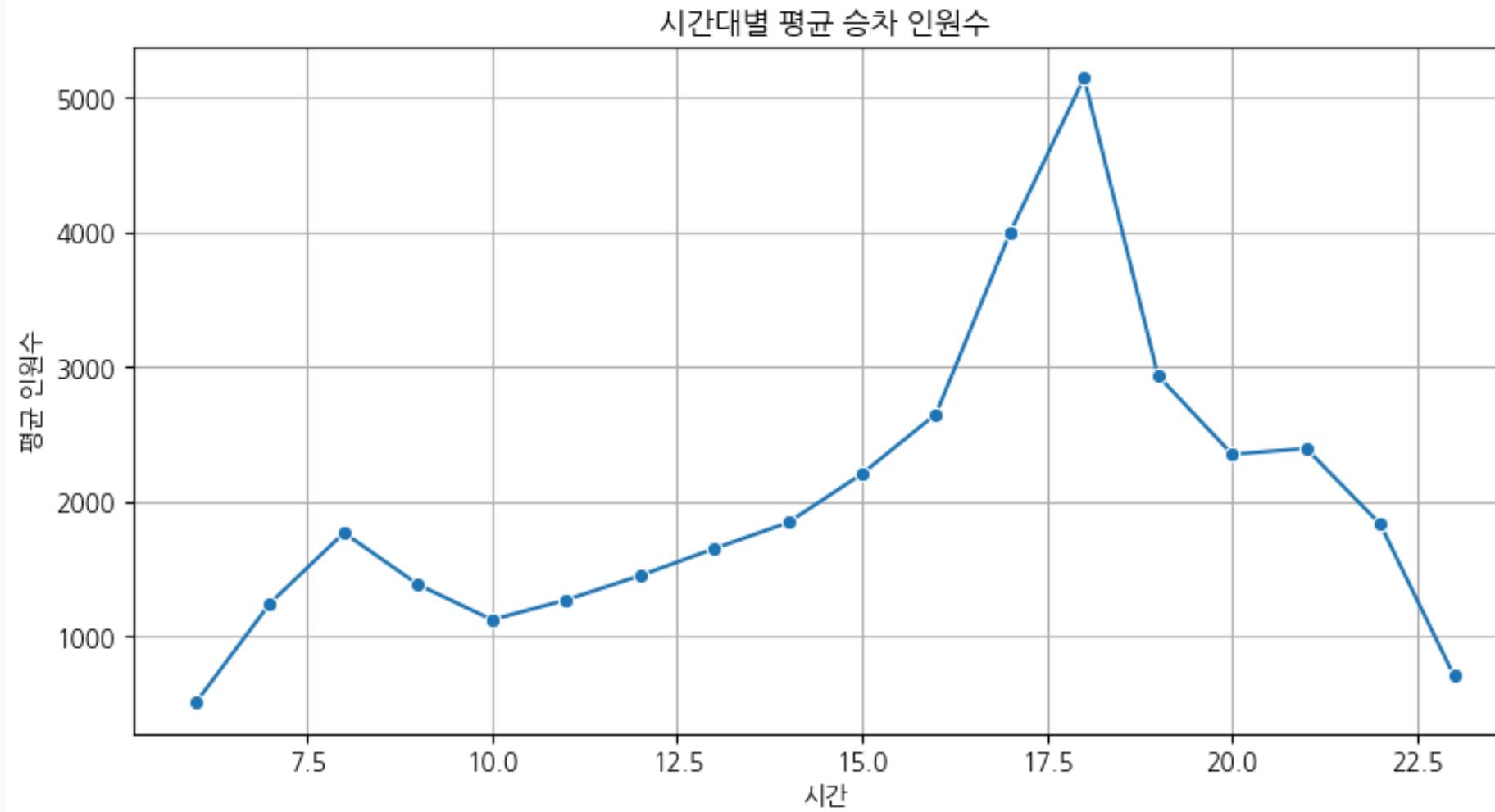
Python

```
train_valid train : 0.9949634194374084 train_valid valid : 0.9830569267272949
train : 0.9943293929100037
test : 0.9538089632987976
```

XGBRegressor

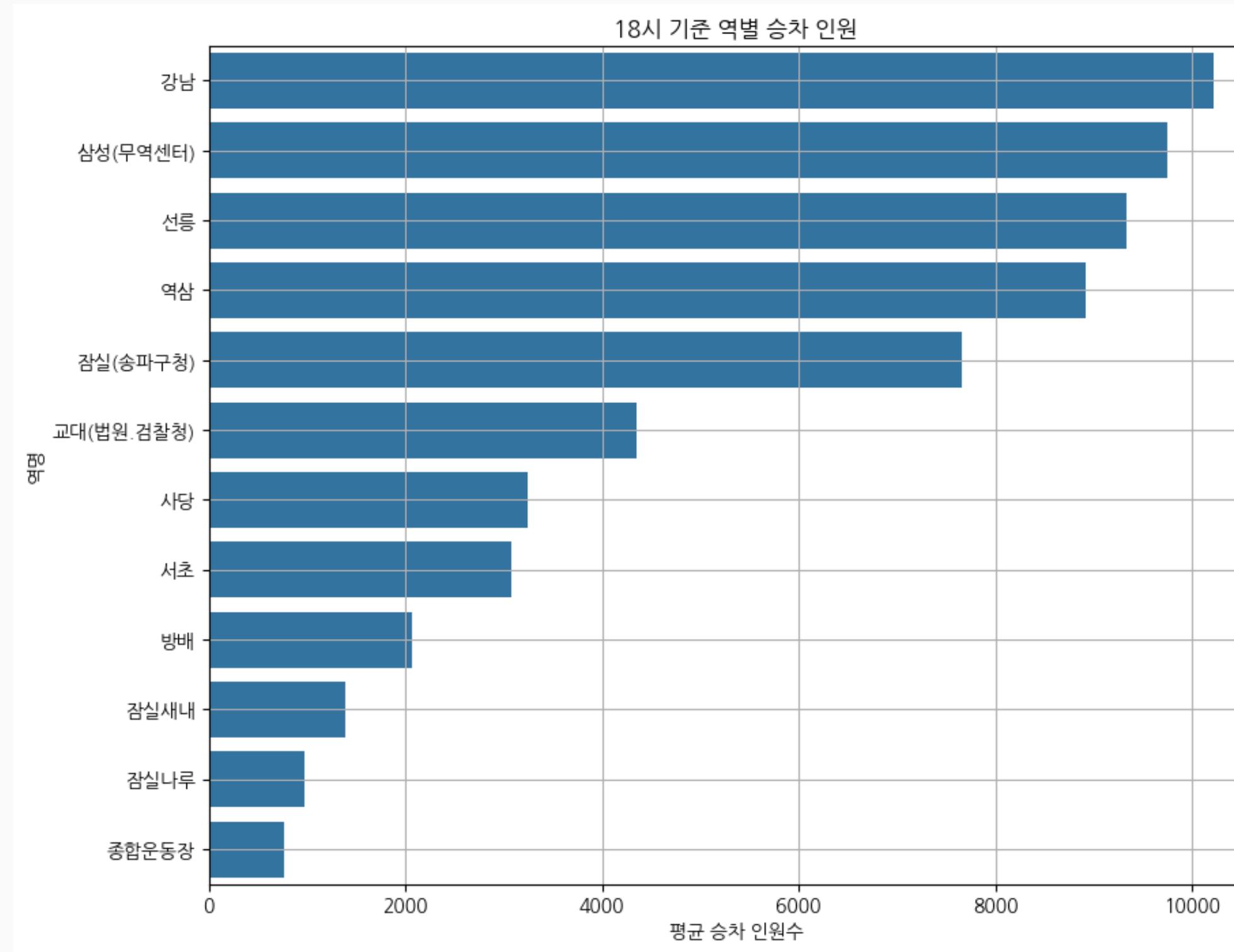
승차 인구수 시각화

3-5



승차 인구수 시각화

3-5

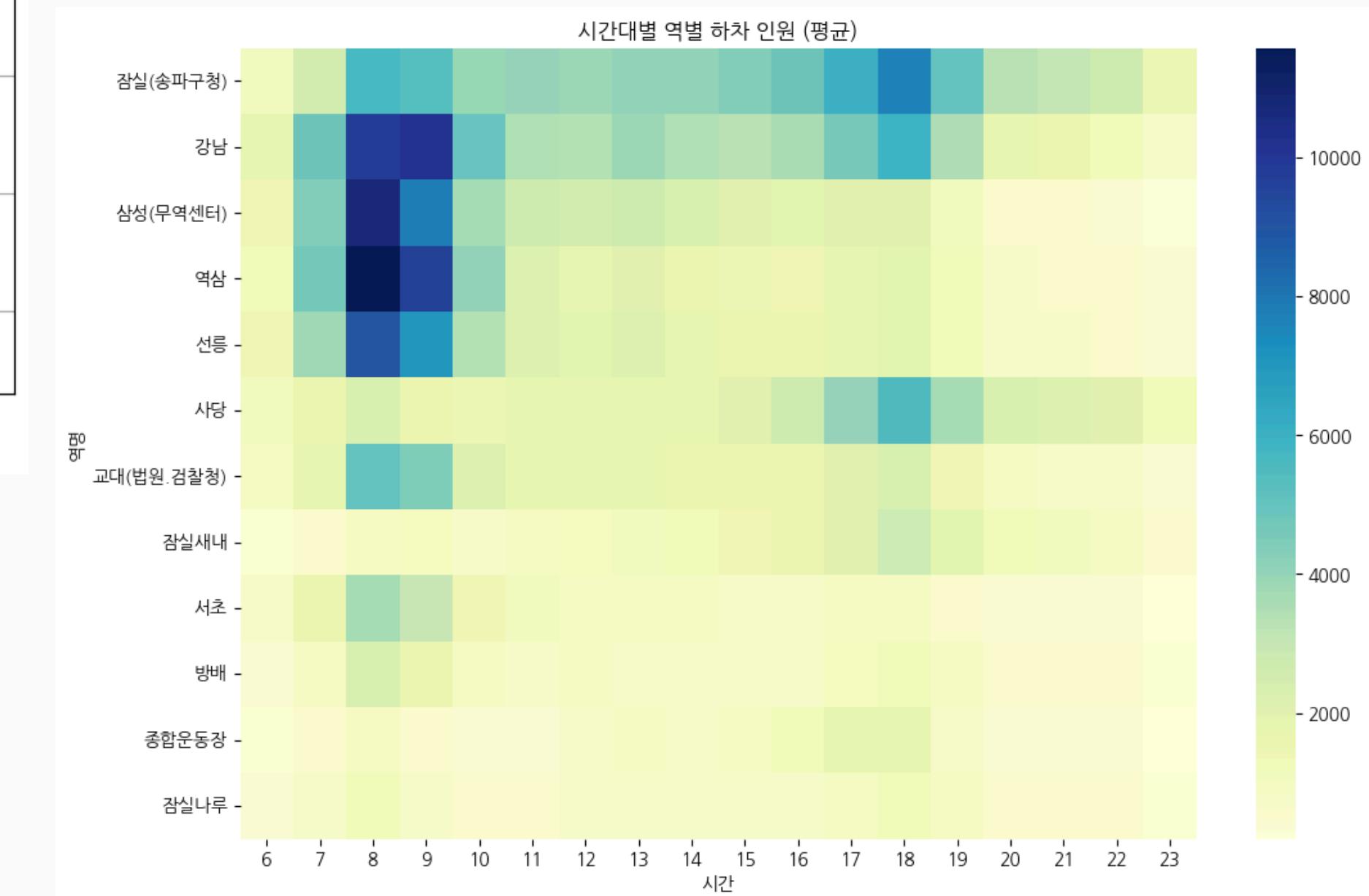
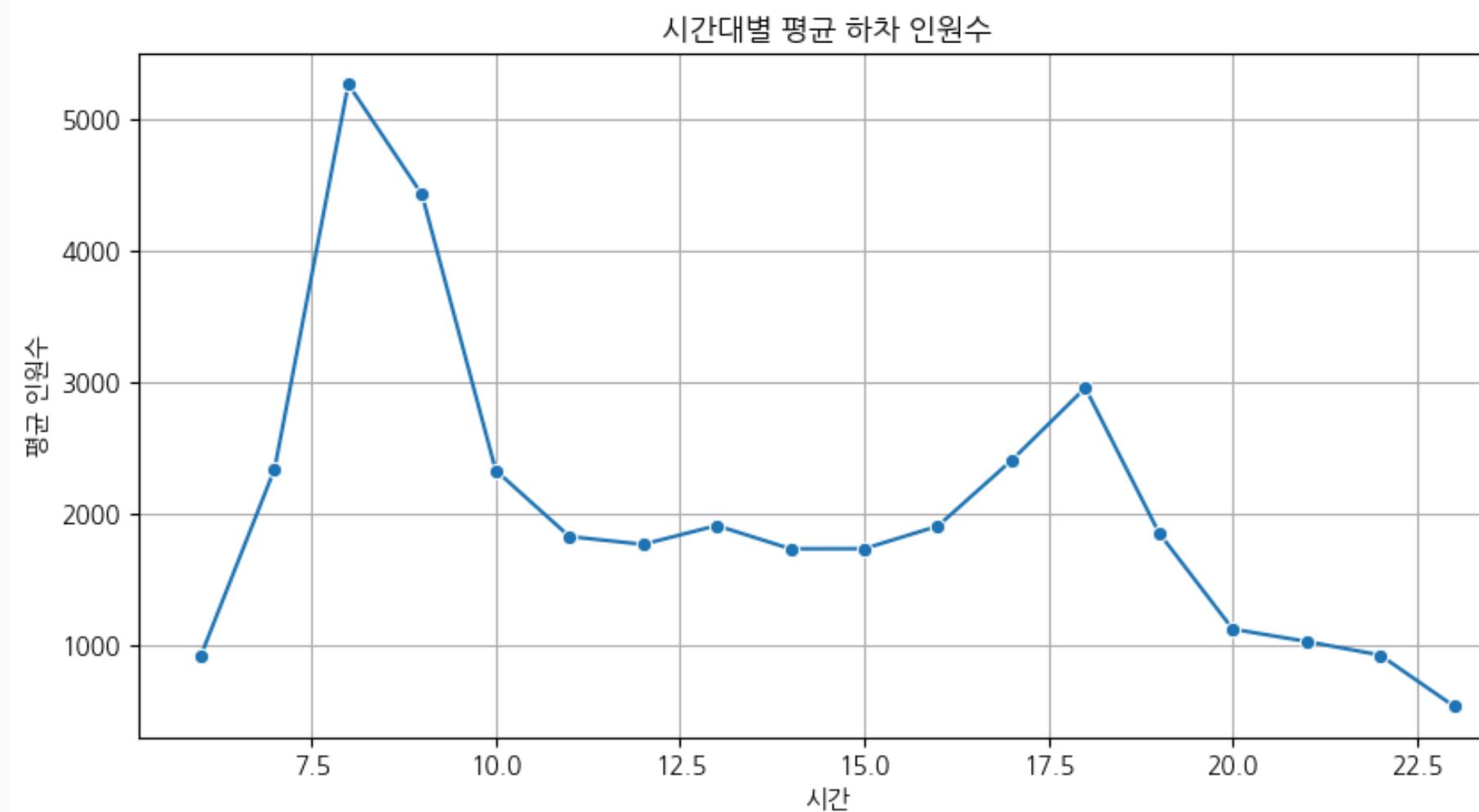


평균 승차 인원수가 가장 높은 시간대인 **18시**를
기준으로 역별 승차인원을 시각화 해보았을 때
강남역에 가장 많은 인원이 승차함.



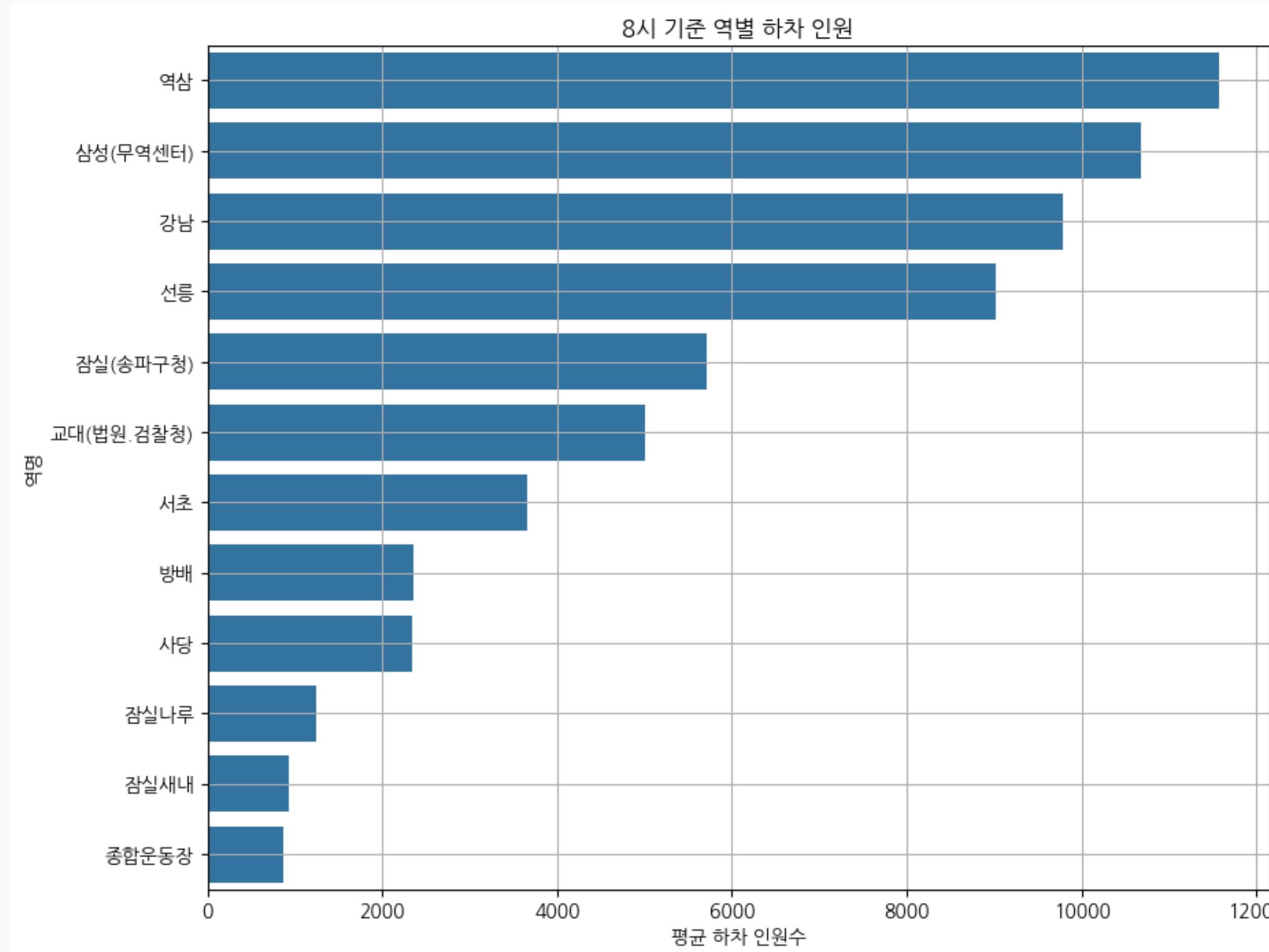
하차 인구수 시각화

3-6



하차 인구수 시각화

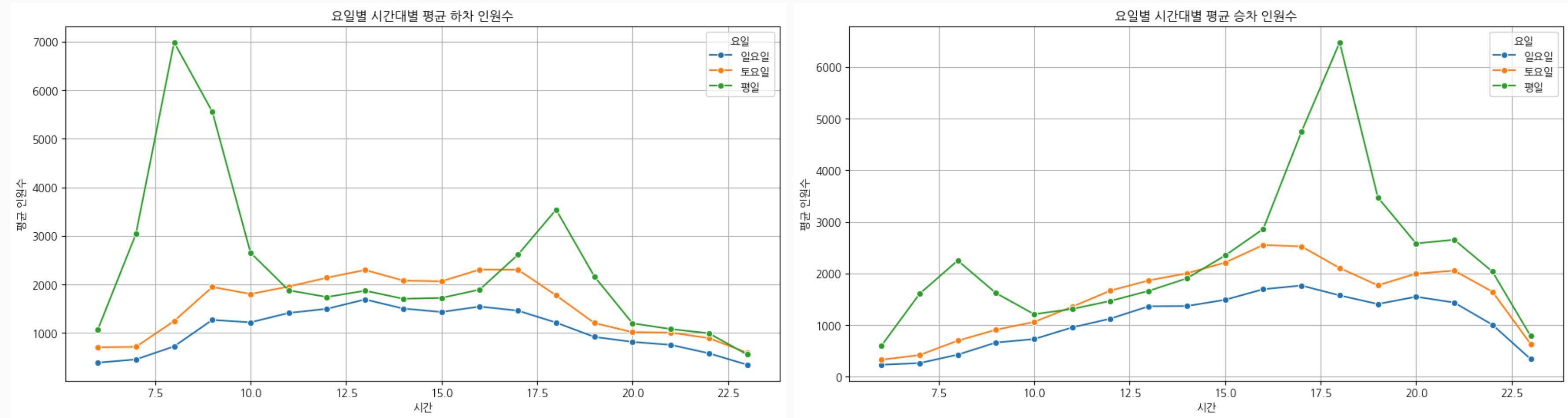
3-6



평균 하차 인원수가 가장 높은 시간대인 **8시**를
기준으로 역별 하차인원을 시각화 해보았을 때
역삼역에 가장 많은 인원이 하차함.



평일과 공휴일



```
train_valid train : 0.9940940141677856 train_valid valid : 0.9702078104019165
train : 0.9929844737052917
test : 0.9408504962921143
```

```
train_valid train : 0.9775237679481507 train_valid valid : 0.9459813952445983
train : 0.9758181571960449
test : 0.9107158184051514
```

평일

공휴일

평일, 토요일, 일요일/공휴일 분리하려 했으나 과대 적합이 발생함.
평일 출퇴근 시간에만 사람이 몰린다고 판단하여 평일 출퇴근 시간과 아닌 시간을 분리하여 모델 생성



평일 출퇴근 승하차 인원 예측

평일 출퇴근 o (기본 컬럼 : '역번호', '시간', '월요일', '화요일', '수요일', '목요일', '금요일' test size = 0.3) | train, test 차이
전체 컬럼 : ['연도', '월', '일', '역번호', '시간', '사용일자', '출근시간', '퇴근시간', '점포수', '환승여부', '출구_수', '총_직장_인구_수', '지하철간격']

- 컬럼 전체 추가 : 25.6% (99.8, 74.2) <- 과대적합
- '점포수', '환승여부', '출구_수' : 22.7% (91.7, 69) <- 과대적합
- '출근시간', '퇴근시간' : 22.6% (91.2, 68.6) <- 과대적합
- '출근시간', '퇴근시간', '총_직장_인구_수' : 22.8% (91.6, 68.8) <- 과대적합
- '출근시간', '퇴근시간', '총_직장_인구_수', '점포수', '환승여부' : 22.7% (91.8, 69.1) <- 과대적합

: 전부 과대적합으로 0.35로 test size를 증가시켜 다시 시도

평일 출퇴근 o (기본 컬럼 : '역번호', '시간', '월요일', '화요일', '수요일', '목요일', '금요일' test size = 0.35) | train, test 차이
전체 컬럼 : ['연도', '월', '일', '역번호', '시간', '사용일자', '출근시간', '퇴근시간', '점포수', '환승여부', '출구_수', '총_직장_인구_수', '지하철간격']

- 컬럼 전체 추가 : 12.7% (99.8, 87.1) <- 과대적합
- '점포수', '환승여부', '출구_수' : 10.6% (91.7, 81.1) <- 과대적합
- '출근시간', '퇴근시간' : 10.6% (91.2, 80.6) <- 과대적합
- '출근시간', '퇴근시간', '출구_수' : 10.6% (91.26, 80.63) <- 과대적합
- '출근시간', '퇴근시간', '총_직장_인구_수', '점포수', '환승여부' : 10.8% (91.81, 81.01) <- 과대적합

: 전부 과대적합으로 0.4로 test size를 증가시켜 다시 시도

평일 출퇴근 o (기본 컬럼 : '역번호', '시간', '월요일', '화요일', '수요일', '목요일', '금요일' test size = 0.4) | train, test 차이
전체 컬럼 : ['연도', '월', '일', '역번호', '시간', '사용일자', '출근시간', '퇴근시간', '점포수', '환승여부', '출구_수', '총_직장_인구_수', '지하철간격']

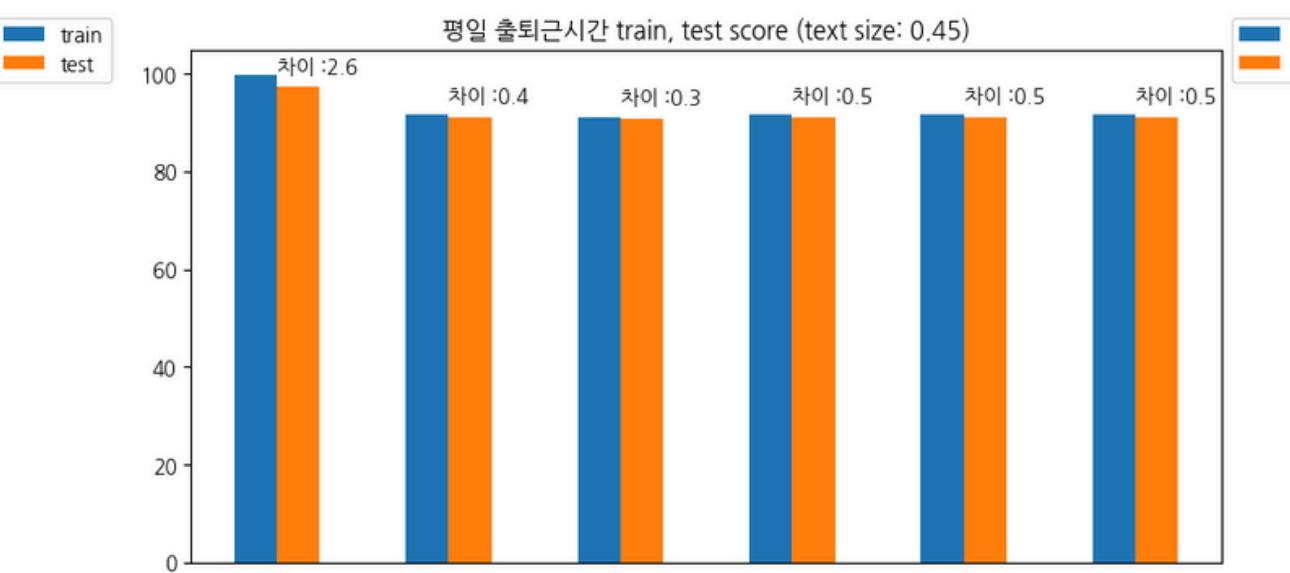
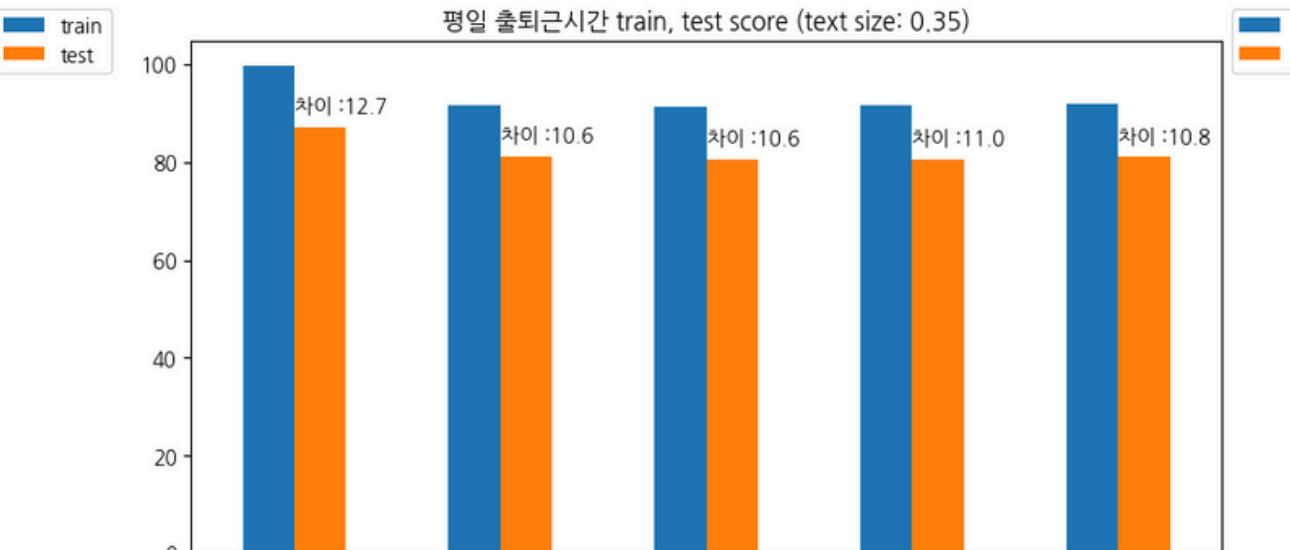
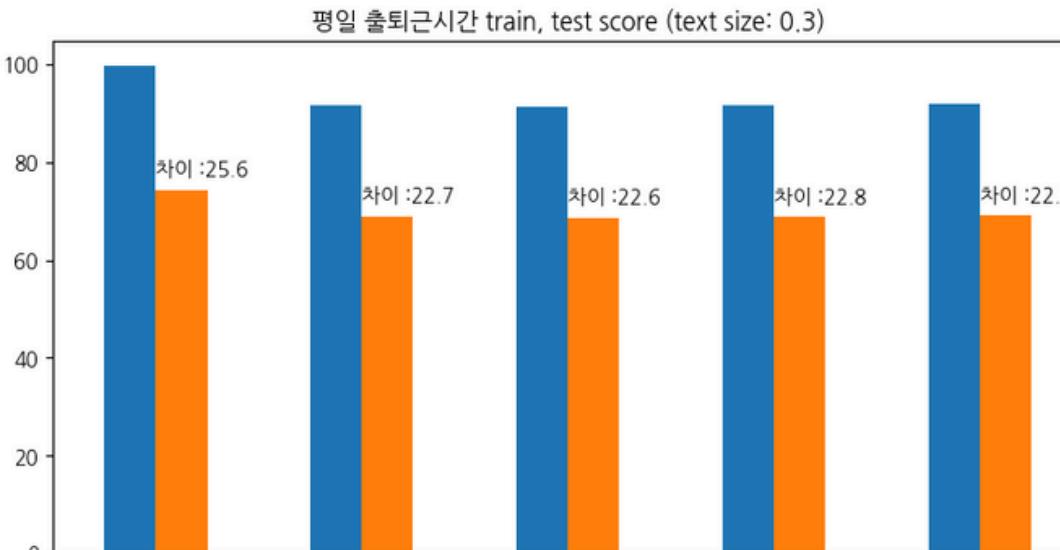
- 컬럼 전체 추가 : 5.32% (99.83, 94.55) >- 과대적합에서 벗어남
- '점포수', '환승여부', '출구_수' : 3.81% (91.82, 88.01)
- '출근시간', '퇴근시간' : 3.8% (91.82, 88.05)
- '출근시간', '퇴근시간', '출구_수' : 3.7% (91.29, 87.59) <- 적정 컬럼
- '출근시간', '퇴근시간', '점포수', '출구_수' : 3.7% (91.78, 88.09) <- 적정 컬럼
- '출근시간', '퇴근시간', '점포수', '출구_수', '총_직장_인구_수' : 3.8% (91.81, 88.05)

: train, valid의 차이는 4 ~ 1% 차이
: 최소 차이값을 구하기 위한 0.45로 test size를 증가시켜 다시 시도

평일 출퇴근 o (기본 컬럼 : '역번호', '시간', '월요일', '화요일', '수요일', '목요일', '금요일' test size = 0.45) | train, test 차이
전체 컬럼 : ['연도', '월', '일', '역번호', '시간', '사용일자', '출근시간', '퇴근시간', '점포수', '환승여부', '출구_수', '총_직장_인구_수', '지하철간격']

- 컬럼 전체 추가 : 2.58% (99.86, 97.28) >- 차이값이 감소
- '점포수', '환승여부', '출구_수' : 0.42% (91.55, 91.13)
- '출근시간', '퇴근시간' : 0.34% (91.10, 90.76)
- '출근시간', '퇴근시간', '출구_수' : 0.35% (91.10, 90.75)
- '출근시간', '퇴근시간', '점포수', '출구_수' : 0.49% (91.61, 91.12)
- '출근시간', '퇴근시간', '점포수', '출구_수', '총_직장_인구_수' : 0.51% (91.65, 91.14)
- '출근시간', '퇴근시간', '점포수', '출구_수', '총_직장_인구_수', '환승여부' : 0.46% (91.59, 91.13)

{'colsample_bytree': 1, 'max_depth': 7, 'n_estimators': 500, 'subsample': 0.5}



과대적합을 피하기위해 test size를 키워가며 진행



평일 출퇴근 제외 승하차 인원 예측

평일 출퇴근 o (기본 컬럼 : '역번호', '시간', '월요일', '화요일', '수요일', '목요일', '금요일' test size = 0.3) | train, test 차이
전체 컬럼 : ['연도', '월', '일', '역번호', '시간', '사용일자', '출근시간', '퇴근시간', '점포수', '환승여부', '출구_수', '총_직장_인구_수', '지하철간격']

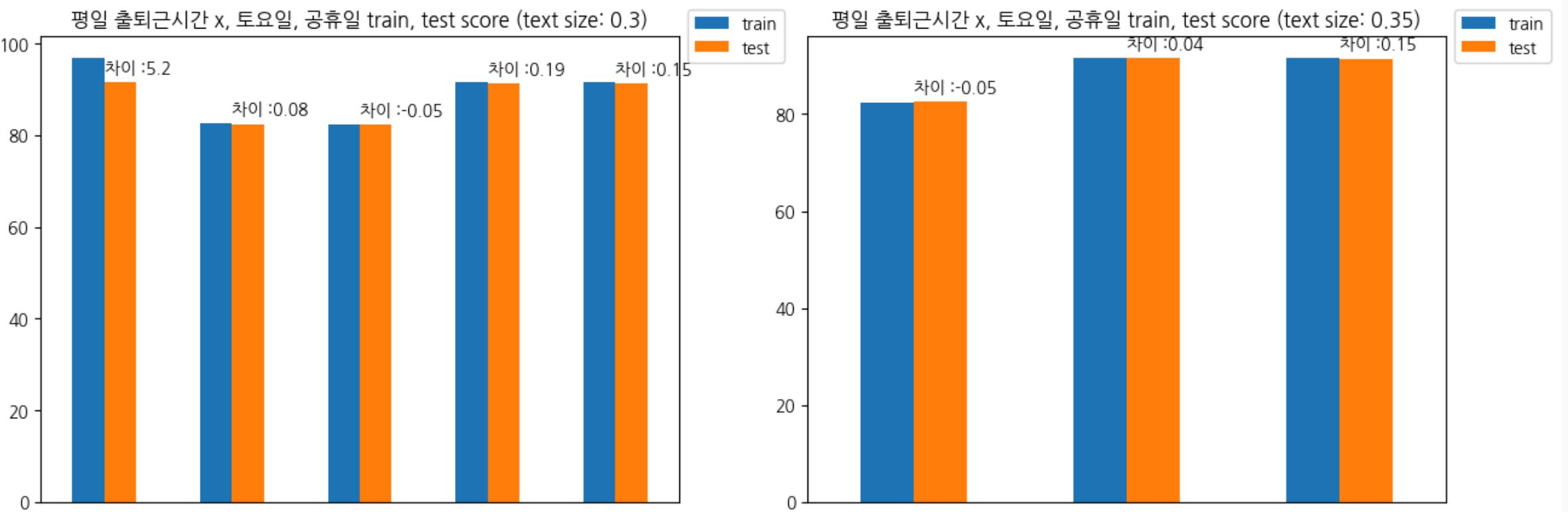
- 컬럼 전체 추가 : 25.6% (99.8, 74.2) <- 과대적합
- '점포수', '환승여부', '출구_수' : 22.7% (91.7, 69) <- 과대적합
- '출근시간', '퇴근시간' : 22.6% (91.2, 68.6) <- 과대적합
- '출근시간', '퇴근시간', '총_직장_인구_수' : 22.8% (91.6, 68.8) <- 과대적합
- '출근시간', '퇴근시간', '총_직장_인구_수', '점포수', '환승여부' : 22.7% (91.8, 69.1) <- 과대적합

: 전부 과대적합으로 0.35로 test size를 증가시켜 다시 시도

평일 출퇴근 o (기본 컬럼 : '역번호', '시간', '월요일', '화요일', '수요일', '목요일', '금요일' test size = 0.35) | train, test 차이
전체 컬럼 : ['연도', '월', '일', '역번호', '시간', '사용일자', '출근시간', '퇴근시간', '점포수', '환승여부', '출구_수', '총_직장_인구_수', '지하철간격']

- 컬럼 전체 추가 : 12.7% (99.8, 87.1) <- 과대적합
- '점포수', '환승여부', '출구_수' : 10.6% (91.7, 81.1) <- 과대적합
- '출근시간', '퇴근시간' : 10.6% (91.2, 80.6) <- 과대적합
- '출근시간', '퇴근시간', '출구_수' : 10.6% (91.26, 80.63) <- 과대적합
- '출근시간', '퇴근시간', '총_직장_인구_수', '점포수', '환승여부' : 10.8% (91.81, 81.01) <- 과대적합

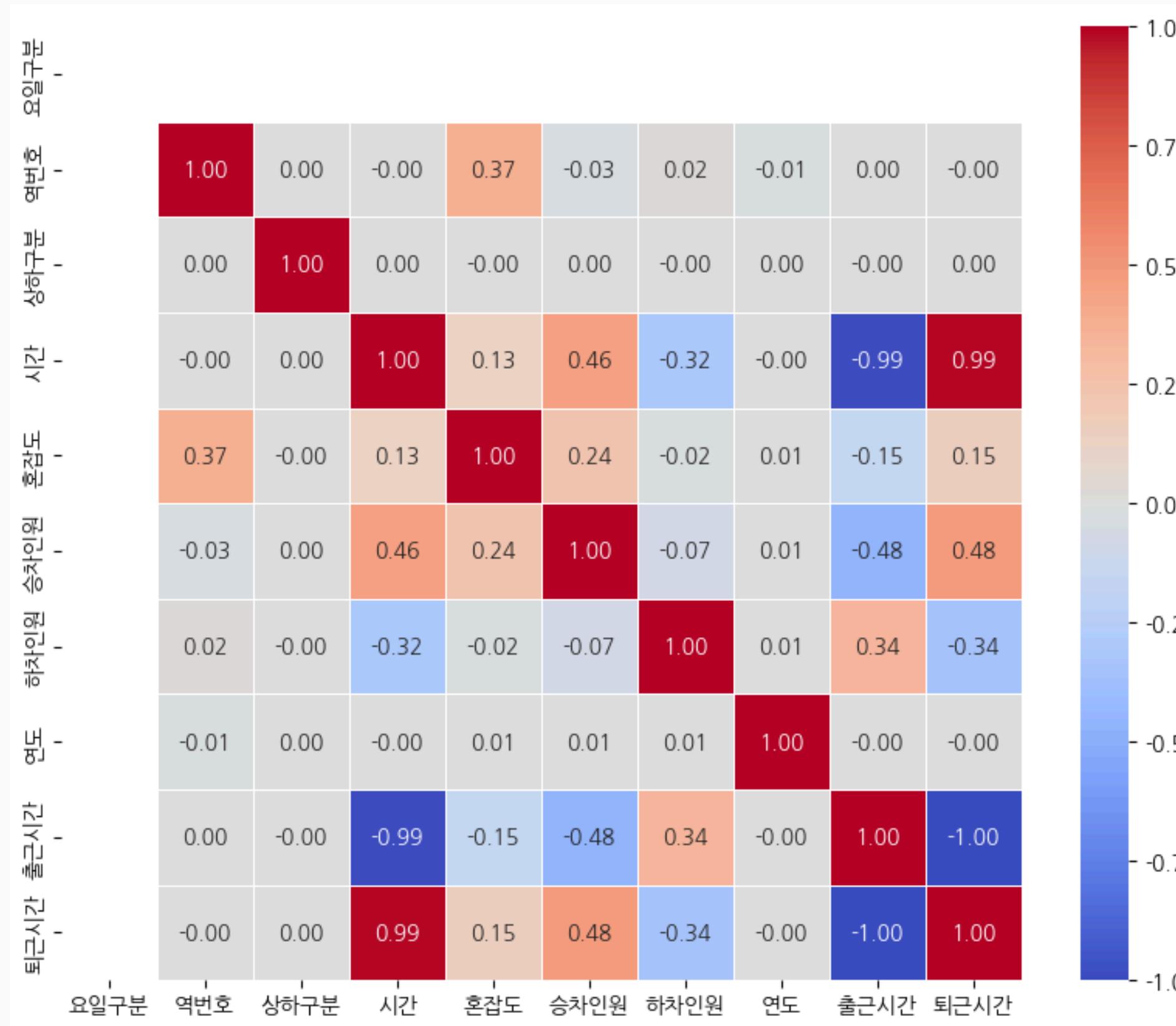
: 전부 과대적합으로 0.4로 test size를 증가시켜 다시 시도



과대적합을 피하기위해 test size를 키워가며 진행



상관관계



혼잡도 1.00
 역번호 0.37
 상하구분 -0.00
 시간 0.13
 승차인원 0.24
 하차인원 -0.02
 연도 0.01
 출근시간 -0.15
 퇴근시간 0.15
 Name: 혼잡도, dtype: float64



평일 출퇴근

```
data = subway_complex_work[['역번호', '상하구분', '시간', '승차인원', '하차인원', '출근시간', '퇴근시간']]  
target = subway_complex_work.혼잡도  
machine.learning(data=data, target=target, model=XGBRegressor, size=0.48)
```

```
{'colsample_bytree': 1, 'max_depth': 7, 'n_estimators': 1000, 'subsample': 0.8}  
train_valid train : 0.9986476018614652 train_valid valid : 0.9566477012634278  
train : 0.9976657548614351  
test : 0.9444166421890259
```

과대적합을 피하기위해 test size를 키워가며 진행
승하차 인원 예측과 동일하게 진행

평일 출퇴근 제외

```
data = subway_complex_normal[['요일구분', '역번호', '상하구분', '시간', '승차인원', '하차인원']]  
target = subway_complex_normal.혼잡도  
machine.learning(data=data, target=target, model=XGBRegressor, size=0.48)
```

```
{'colsample_bytree': 1, 'max_depth': 7, 'n_estimators': 70, 'subsample': 0.7}  
train_valid train : 0.9949692487716675 train_valid valid : 0.9608590364456177  
train : 0.9950538873672485  
test : 0.9545461535453796
```



SECTION 04

App

Flutter	3.32.0	
프론트 프레임워크		
Dart	3.7.2	
프론트엔드 언어		
VScode	1.100.0	
개발도구		
Provider	^6.1.5	
상태관리 라이브러리		

FastApi	0.1.0 ver	
백엔드 프레임워크		
Python	3.12.7 ver	
서버 언어		
SQLite	8.0.42 ver	
데이터 베이스		
GitHub	2.48.1 (Git)	
프로젝트 저장소		
Folk	2.54.3	
프로젝트 유ти리티		

1. State 관리

provider: ^6.1.5

2. Network

http: ^1.4.0

3. Database

sqflite: ^2.4.2

4. 날짜,시간

flutter_datetime_picker_plus: ^2.2.0

5. Image

flutter_svg: ^1.1.2

liquid_progress_indicator_v2: ^0.5.0

6. utility

url_launcher: ^6.3.2

flutter_link_previewer: ^4.0.0

webview_flutter: ^4.13.0

path: ^1.9.1

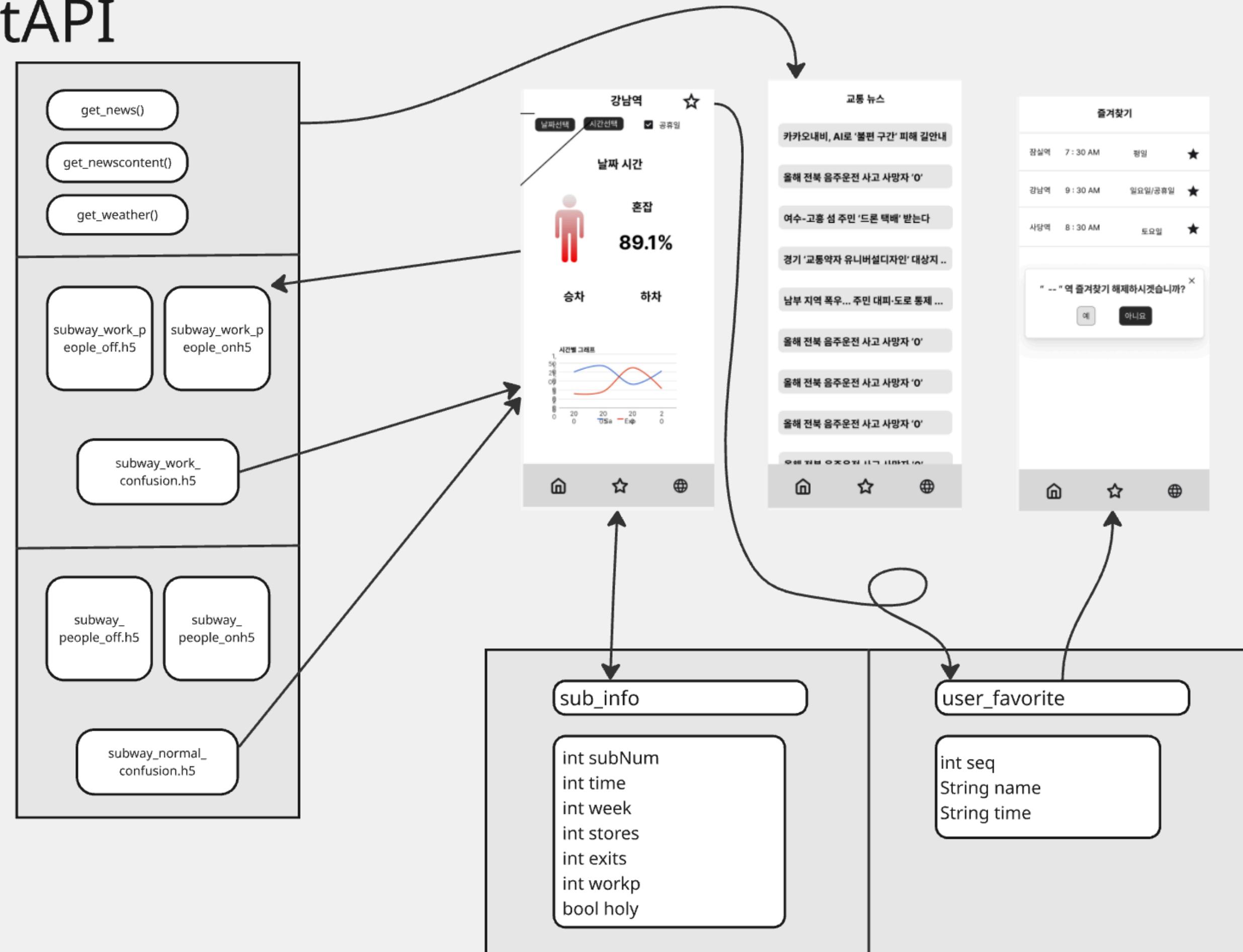


Git log

Merge remote-tracking branch 'origin/eunjun'	...
 seoyoony pushed 3 commits to develop · 1d65565...005b69e · 1 hour ago	...
디자인	...
 seoyoony pushed 4 commits to seoyun · 48e900e...9ceb5a1 · 1 hour ago	...
즐겨찾기 설정	...
 Eunjun1 pushed 1 commit to eunjun · 79b09c5...9d8f90b · 1 hour ago	...
Merge remote-tracking branch 'origin/chagjun'	...
 seoyoony pushed 2 commits to develop · 79b09c5...1d65565 · 4 hours ago	...
person 혼잡도 완성과 즐겨찾기 수정, 조회기능 수정, 크롤링 앱 실행할때 같이 실행 (07. 18 03 : 16)	...
 seoyoony pushed 1 commit to develop · 48e900e...79b09c5 · 4 hours ago	...
person 혼잡도 완성과 즐겨찾기 수정, 조회기능 수정, 크롤링 앱 실행할때 같이 실행 (07. 18 03 : 16)	...
 Eunjun1 pushed 12 commits to eunjun · 915d0a3...79b09c5 · 11 hours ago	...
CJ_Work_05	...
 ChangJun0716 pushed 1 commit to chagjun · af6f30d...107ca7f · 13 hours ago	...
혼잡도 페이지 값넣음	...
 seoyoony pushed 2 commits to develop · 1f272f6...48e900e · 17 hours ago	...
혼잡도 페이지 값넣음	...
 seoyoony pushed 1 commit to seoyun · 2fda3c5...48e900e · 17 hours ago	...
상선외선추가	...
 seoyoony pushed 10 commits to seoyun · 6f3a7f0...2fda3c5 · 19 hours ago	...
수정완료	...
 seoyoony pushed 10 commits to develop · ab1615b...1f272f6 · 20 hours ago	...
CJ_Work_04	...
 ChangJun0716 pushed 5 commits to chagjun · b37e543...af6f30d · 20 hours ago	...
star.dart	...
 seoyoony pushed 5 commits to seoyun · 010fb6...6f3a7f0 · 20 hours ago	...
예측 함수 view에 적용 (07.17 17:56)	...
 Eunjun1 pushed 10 commits to eunjun · e66321b...915d0a3 · 20 hours ago	...

FastAPI

System Flow



SQLite

오전9시의 혼잡도는?

80% 미만

여유



120% 미만

주의



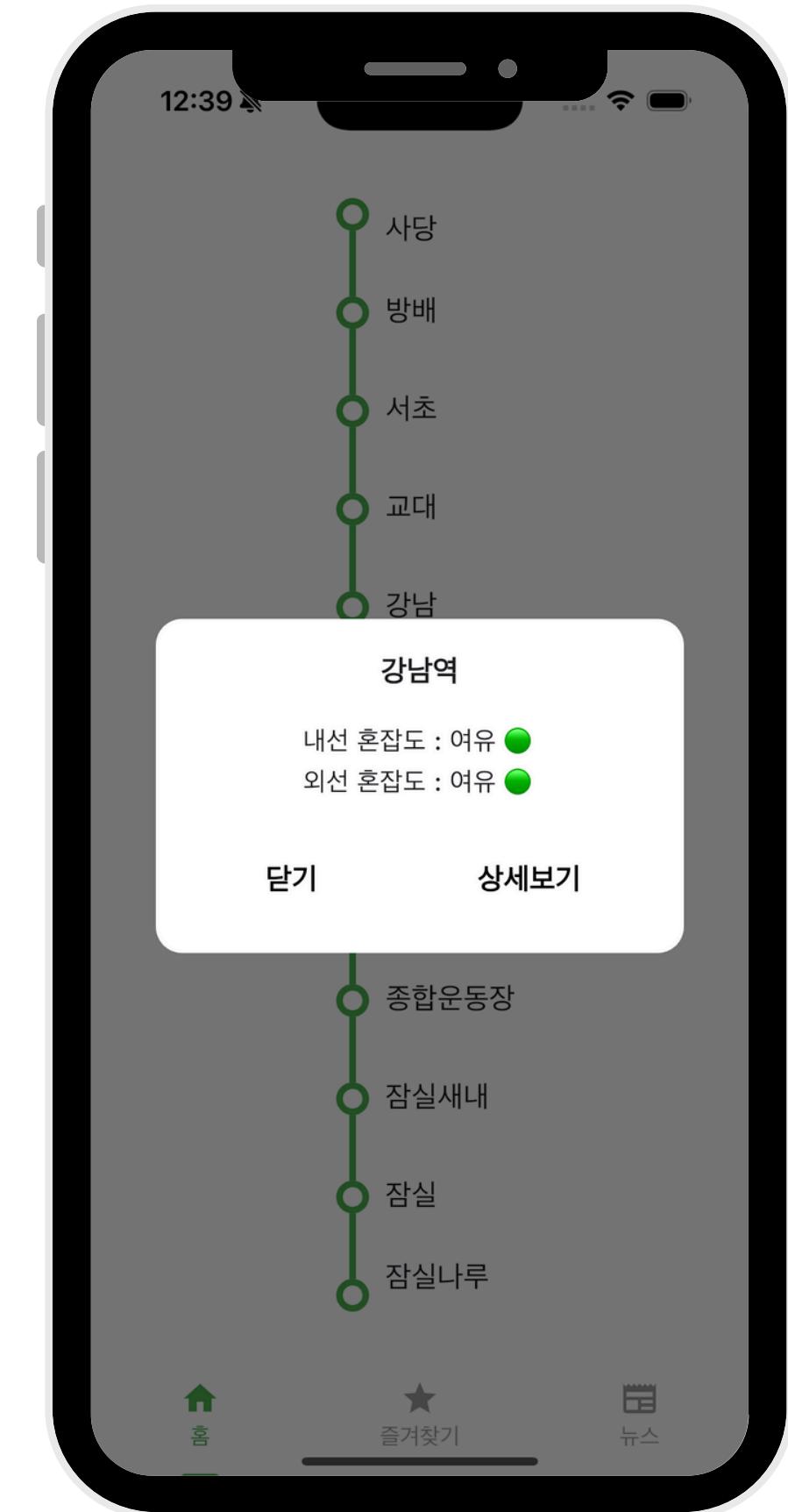
120 ~

혼잡



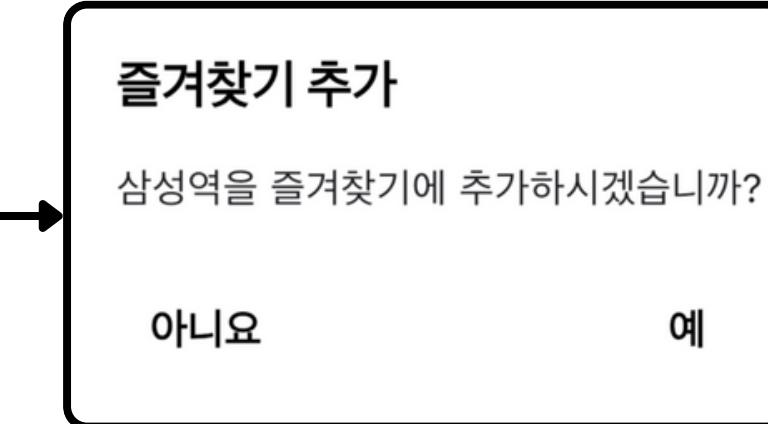
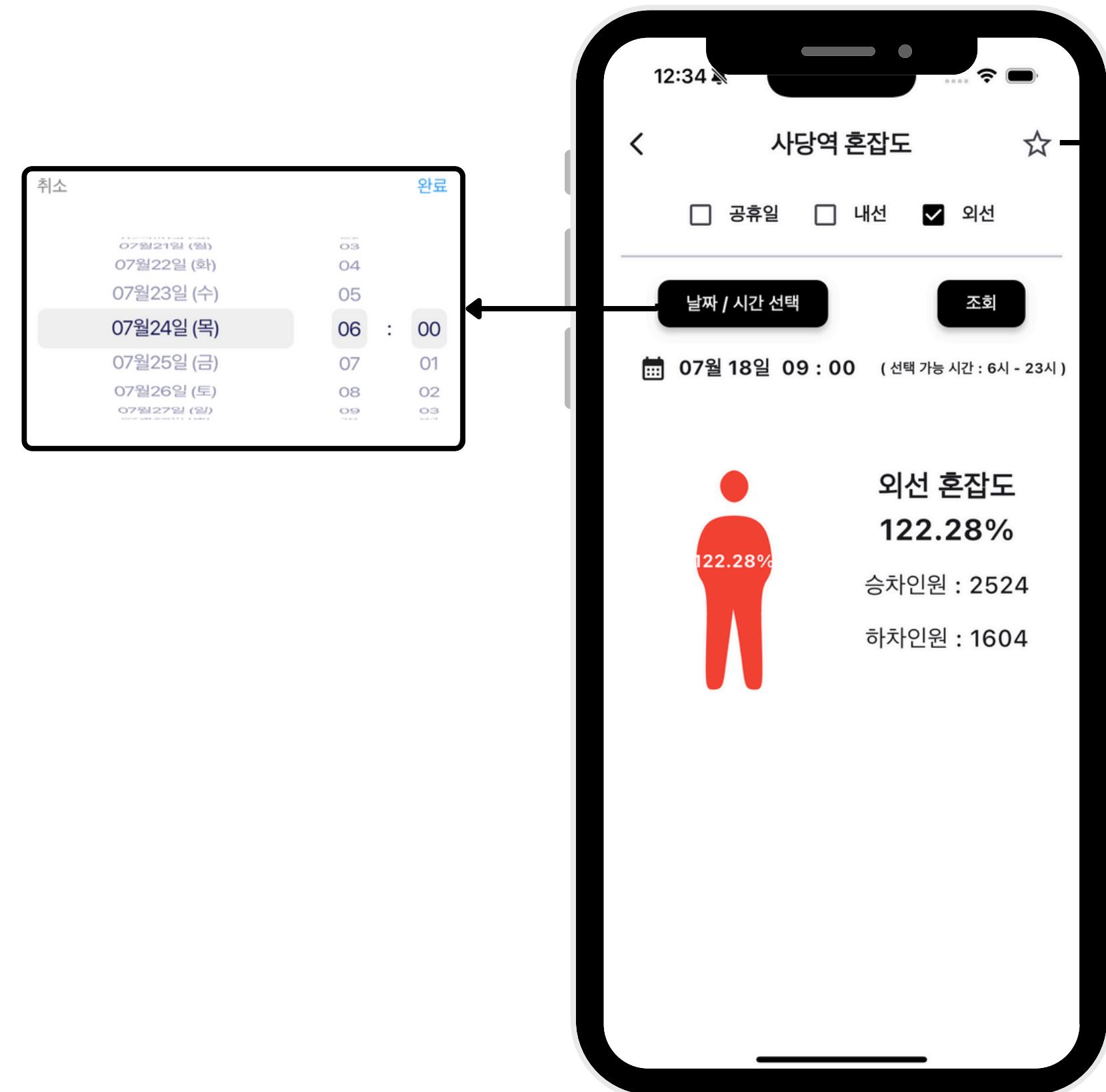
※ 지하철 열차 내 혼잡도 기준

구분	여유	보통	주의	혼잡1	혼잡2
혼잡도	80% 이하	80~130%	130%~150%	150~170%	170% 이상
차내 상태	대부분 착석, 통로 여유	여유롭게 이동	이동 시 부딪힘	열차 내 이동불가	



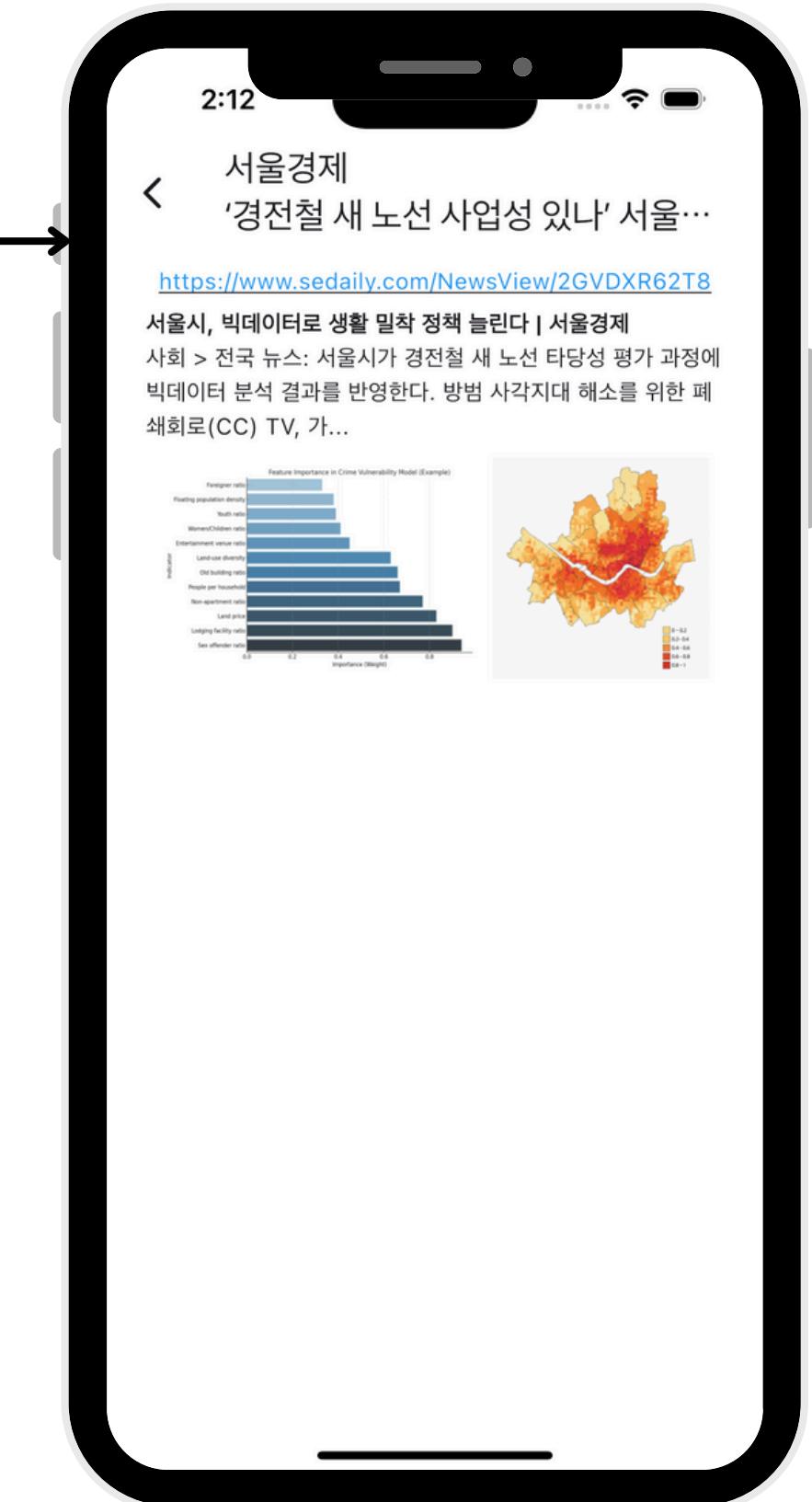
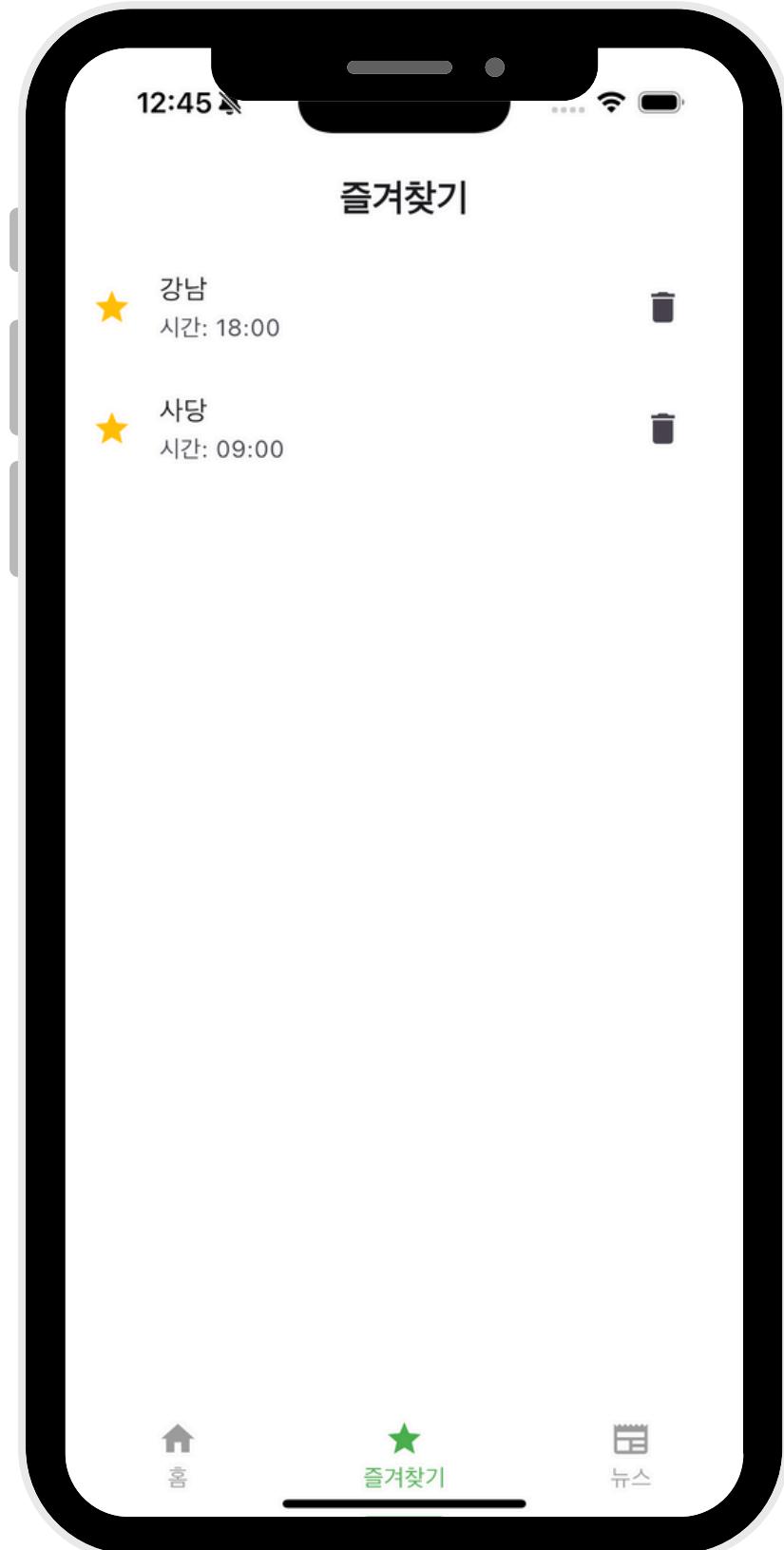
혼잡도 상세 페이지

- 사당
- 방배
- 서초
- 교대
- 강남
- 역삼
- 선릉
- 삼성
- 종합운동장
- 잠실새내
- 잠실
- 잠실나루



즐겨찾기 / 뉴스

외부 앱으로 이동(Safari)



서윤

하나의 주제에 대해 분석을 할 때 생각보다 연관된 많은 것들을 찾아보고 분석해야 한다는 것을 깨달았습니다. 분석을 마치고 앱개발까지 했지만 하나의 호선만 가지고 해봐서 다른 호선들을 해보고 싶습니다. 이번 기회로 분석에 대해 잘 알게 되었고 평소에 궁금했던 주제로 분석과 개발까지 해봐서 재밌었습니다. *^^*

은준

분석을 하면서 정확한 예측을 하기 위해서는 정말 많은 시간의 투자가 필요하고 예측력을 늘리기 위해 생각 해야할 것들이 정말 많다고 느꼈습니다. 또한 개발에 투자할 시간이 적어서 앱의 퀄리티를 높게 만들지 못한것에 아쉬움이 남았습니다. 이번 프로젝트를 바탕으로 앱을 기획 할때 좀 더 많은 것을 체크하고 생각할 수 있게 된것 같습니다.

창준

- 처음으로 팀장이 아닌 팀원으로 참여한 프로젝트였기에 팀원의 입장을 많이 배우게 되었습니다.
- 무언가를 예측한다는 것이 얼마나 어려운 것인지 배우는 계기가 되었으며 분석에서의 식견을 넓힐 수 있는 경험이 되었습니다.

감성

훌륭한 팀원들 덕분에 항상 즐겁고 책임감 있게 작업할 수 있었습니다. 프로젝트 분석 과정 중 막히는 부분이 있어도 서로 도와주고 배려해주는 분위기가 정말 좋았어요. 프로젝트 도중 몸이 좋지 않아 흐름을 잘 따라가지 못했지만, 그럴 때마다 도와준 은준님, 창준님 감사합니다. 팀장님의 뛰어난 통솔력과 리더십 덕분에 프로젝트를 마무리할 수 있었습니다.



THANK YOU